

# Applied Data Science in Azure with Microsoft Fabric and Azure Machine Learning

Enrico & Tomaž

# Agenda

- Module 1 – Getting onboard with F&AML
- Module 2 – Storage / storing data and compute with F&AML
  - Exercise for F / AML
- Module 3 – Data Movement
  - Exercise for F / AML
- Module 4 – Data Exploration
  - Exercise for F / AML
- Module 5 - MLFlow
  - Exercise for F / AML
- Module 6 – E2E Solution
  - Exercise for F / AML

**Outlook of the day:**

**9.00 – 12.00**

**12.00-13.00 - LunchTime**

**13.00 – 17.00**

**2x Coffee Time**

# Material

[tomaztk/Fabric\\_Azure\\_Machine\\_Learning: Microsoft Fabric and Microsoft Azure Machine Learning \(github.com\)](#)

[tomaztk/Azure-Machine-Learning: Azure Machine Learning \(github.com\)](#)

# With you today – Enrico van de Laar

- Developer | data analyst
- Privacy Guru
- SQL Server, R, Python, C#
- 20years experience MSSQL, DEV, BI, DM
- Frequent community speaker, Author
- Privinity CEO
- MVP Alumni



# With you today – Tomaž Kaštrun

- BI Developer | data analyst | data scientist
- SQL Server, SAS, R, Python, C#, SAP, SPSS
- 20years experience MSSQL, DEV, BI, DM
- Frequent community speaker
- Avid coffee drinker & bicycle junkie



<http://tomaztsql.wordpress.com>

tomaz.kastrun@gmail.com

@tomaz\_tsqI

/in/tomaztsql

<http://github.com/tomaztk>

<https://mvp.microsoft.com/PublicProfile/5002196>



# Module 1

Starting with light general introduction to Microsoft Fabric and Azure Machine Learning Services (AML), and how you navigate these services.

- basic key concepts
- how they affect both offerings and subscriptions
- key differences and similarities between both services
- explore additional administrative settings

# Basic Concepts

- **Subscription:** Pay-as-you-go, Capacity, ToC purchase
- **Azure ML Studio:** A drag-and-drop interface for building ML models.
- **Azure ML Designer:** Visual interface to create ML pipelines.
- **Notebooks:** Support for Jupyter Notebooks for code-based model development.
- **Automated ML (AutoML):** Automatically trains and tunes models using hyperparameter tuning and model selection.
- **MLOps:** Supports integration with DevOps for model lifecycle management, version control, and continuous integration.

# Basic Concepts – Data Storage

- **Azure Data Lake Storage (ADLS)**: A highly scalable data lake for big data analytics. It stores both structured and unstructured data in its native format.
- **Azure Blob Storage**: Object storage for data, images, videos, and other binary formats.
- **Azure SQL Database**: A relational database service with built-in machine learning capabilities.
- **Azure Cosmos DB**: A globally distributed, multi-model database ideal for real-time data and fast analytics.
- .

# Basic Concepts – Cognitive services / Azure AI

- **Core Concept:** Pre-built APIs for AI capabilities without needing to build machine learning models from scratch.
- **Key Services:**
  - Vision:** Image recognition, object detection.
  - Speech:** Speech-to-text, text-to-speech, translation.
  - Language:** Text analysis, sentiment analysis, translation.
  - Document:** Intelligence on documents, slicing, tokenization, embeddings
  - Decision:** Personalization and anomaly detection.
  - OpenAI:** Access to LLM and Vector index

# Basic Concepts – Data movement, pipelines and orchestration

- **Azure Data Factory (ADF)**: A cloud ETL service that allows you to build data pipelines to move and transform data across services.
- **Azure Event Hubs**: A real-time data ingestion service for high-throughput data streaming and event-driven processing.
- **Azure Stream Analytics**: A real-time analytics service for processing data streams from IoT devices, social media, and more.

## Basic Concepts – Model deployment / consumption

- **Azure Kubernetes Service (AKS)**: A managed Kubernetes service to deploy machine learning models at scale. Enables scaling, load balancing, and model versioning.
- **Azure App Service**: Another platform for deploying web apps, including those that host machine learning models.
- **Azure Functions**: Serverless compute to run lightweight machine learning models.

## Basic Concepts – Model monitoring

- **Azure Monitor:** A comprehensive monitoring solution that includes logs, metrics, and alerts for deployed models and services.
- **ML Pipelines:** In Azure ML, these are reusable workflows for automating the training and deployment of models.
- **Model Versioning:** Keeping track of model versions and experiments within Azure ML or with MLOps.

## Basic Concepts – Security

- **Role-Based Access Control (RBAC):** Fine-grained access control to services and data.
- **Managed Identity:** Secure, managed identity for resources to authenticate and access other Azure services without storing credentials.
- **Data Encryption:** Built-in encryption for data at rest and in transit, ensuring security and compliance.

## **Basic Concepts – Integration with other Azure services**

- **Power BI:** Business intelligence tool for visualizing data and model results, tightly integrated with Azure.
- **Azure IoT Hub:** Integration for IoT devices to send data into Azure for real-time analytics and machine learning.
- **OpenAI:**

# **Offerings for Data Science**

- Fabric
- Azure Machine Learning
- Databricks
- HD Insight

Comprehensive Comparison of Data Engineering Platforms					
Feature / Product	Databricks	Microsoft Fabric	Azure Machine Learning	HDInsight	Overall Rating
Primary Offering	Unified platform for data engineering, ML, and analytics	Data Fabric platform for data integration, analytics, and governance	Machine learning platform for building, training, and deploying models	Managed cloud service for big data analytics (Hadoop, Spark, etc.)	4.8
Data Engineering	5/5: Strong in ETL with Apache Spark, notebooks, and ML	4/5: Good integration but lacks Databricks' power	3/5: ML focused, limited for ETL	4/5: Built for big data, but not focused on ML	4.5
Machine Learning	5/5: Supports scalable ML with MLflow, AutoML, Spark MLlib	3/5: Limited advanced ML capabilities, better for data integration	5/5: ML-centric, strong AutoML, pipelines, distributed training	3/5: Supports ML but lacks advanced tools like Databricks	4.2
Analytics	5/5: Great for exploratory data analytics, Spark SQL	4/5: Strong reporting and Power BI integration	5/5: More ML-oriented, not ideal for interactive analytics	4/5: Focused on big data processing, not analytics per se	4.4
Programming Languages	5/5: Supports Python, SQL, Scala, R, Notebooks functionality	4/5: Primarily SQL and Power Query, Python, Spark	5/5: Python, R, Scala and SDK support for deep learning	5/5: Supports Python, Scala, and others for big data	4.7
Compute Scaling	5/5: Auto-scaling clusters, Pools, Photon, Multi	4/5: Elastic pools, easy scaling, but less compute-intensive	5/5: Scalable compute for distributed ML training, AK8s	4/5: Auto-scaling but more oriented to big data than ML	4.6
Integration	4/5: Integrates well with Azure, data lakes, MLflow	5/5: Best for Microsoft ecosystem, Power BI, Azure Synapse, DWH	5/5: Deep Azure ecosystem integration, supports Git, MLOps	4/5: Integrates well with Azure and other data services	4.3
Ease of Use	4/5: Excellent for data scientists, data engineers, but complex	5/5: User-friendly, low-code/no-code tools	4/5: Powerful, but setup can be complex for non-experts	3/5: More complex for beginners, focused on big data engineers	4.1
Personal experience					
Costs	5/5	4/5	3/5	4/5	
Maturity	5/5 On Market almost 10y	3/5 Rapid development, changing focus	5/5 Mature with 10+Years	5/5 Mature	

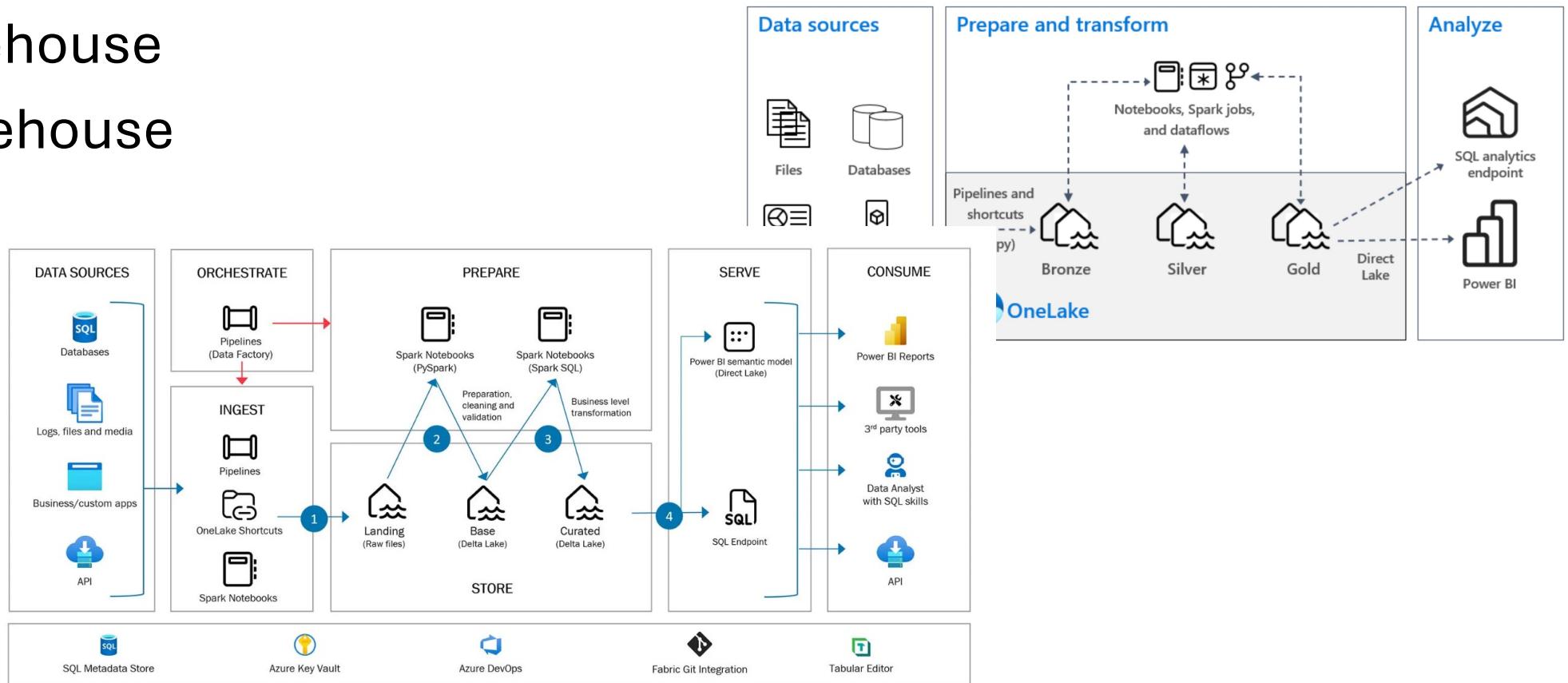
# General concepts - Architecture

Workspaces (Delta Lake; ETL; Medallion; Environments; CI/CD)

Data Lakehouse

Data Warehouse

Spark



# Data warehouse

- Unified data repository for storing large amounts of information from multiple sources
- Combines relational datasets from business apps, ERP, CRM, FICO
- Transactional data and master data
- Processes: extracting, clearing, versioning, consuming



# Data Warehouse PRO

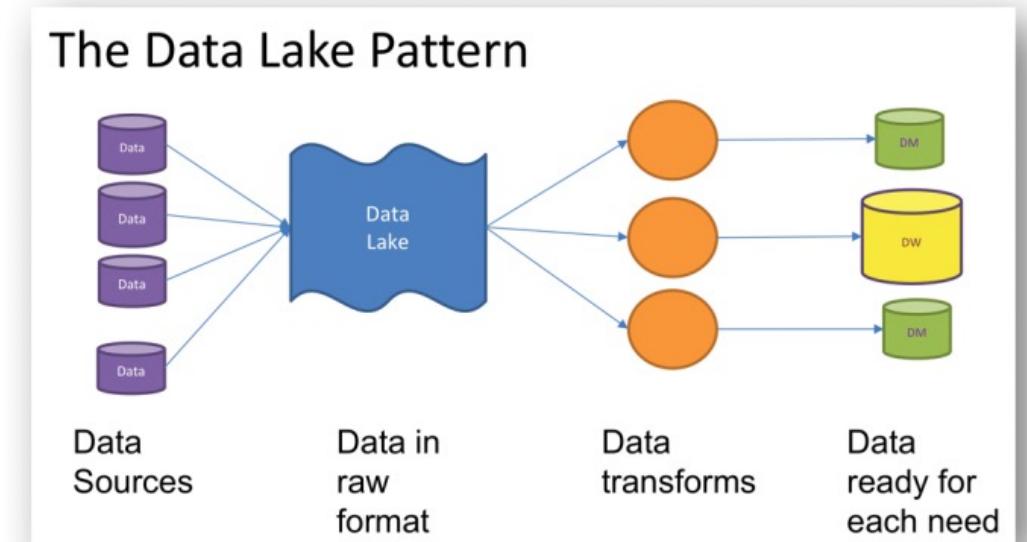
- Data canonization / standardization
- Improved data quality, consistency
- Improves decision-making business process and analytics

# Data Warehouse CON

- Lack of flexibility
- “Struggles” with semi- and un-structured data
- High OPEX (implementation, CI/CD, maintenance)

# Data Lake

- Centralized and highly flexible storage repository
- For structured, unstructured data; original / raw format
- Durable, cost-effective
- Schema on-read
- ETL for purposes and “ad-hoc”
- Machine learning, Shallow/Deep
- IoT devices, streaming data



# Data Lake PRO

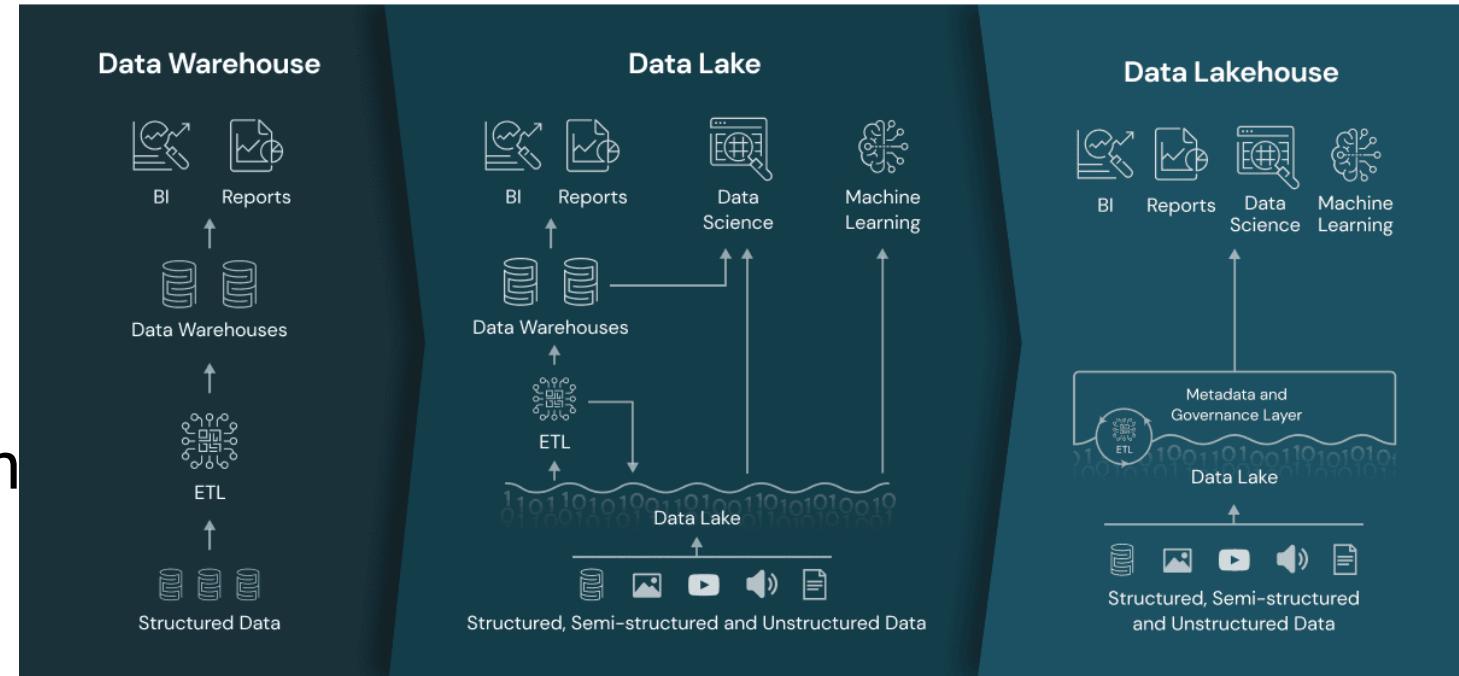
- Data consolidation  
(structured + unstructured)
- Improved flexibility
- Cost savings (object store)
- Improved and support for variety analytics, shallow and deep learning  
(machine learning)

# Data Lake CON

- Lack of management
- Enforcing data reliability
- Uniform security model  
(data formats)

# Data Lakehouse

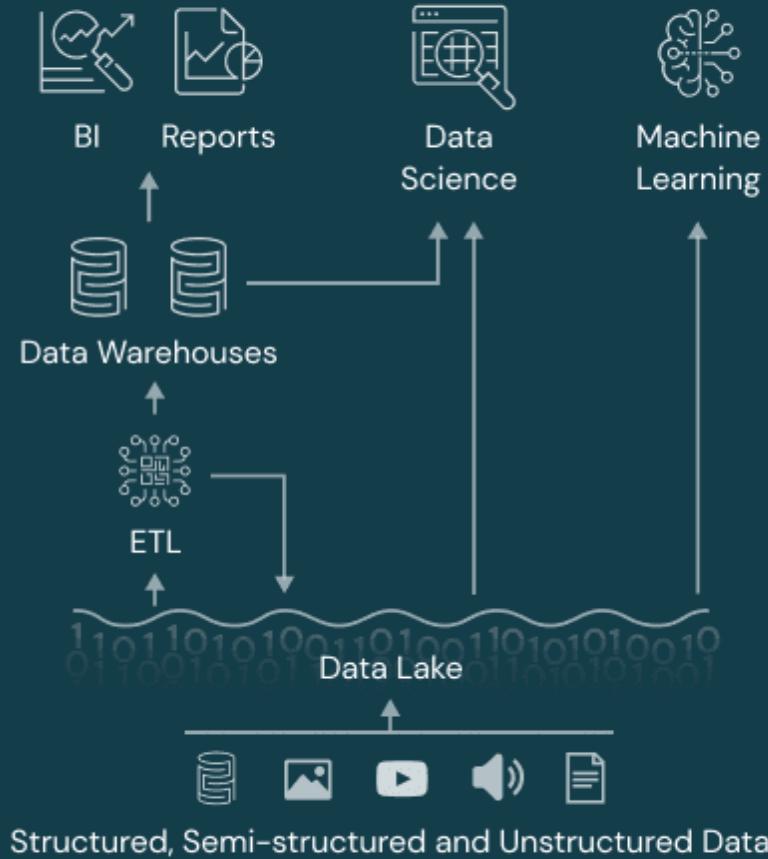
- Scalable data lakes
- Organized data management
- Organized ACID for:
  - Transactional data (DWH)
  - Improved analytics (DWH)
  - Improving ML, DL (DWH, Data Lake)
- Metadata layer for data lakes
- Improved Query engine
- Optimized access for data science



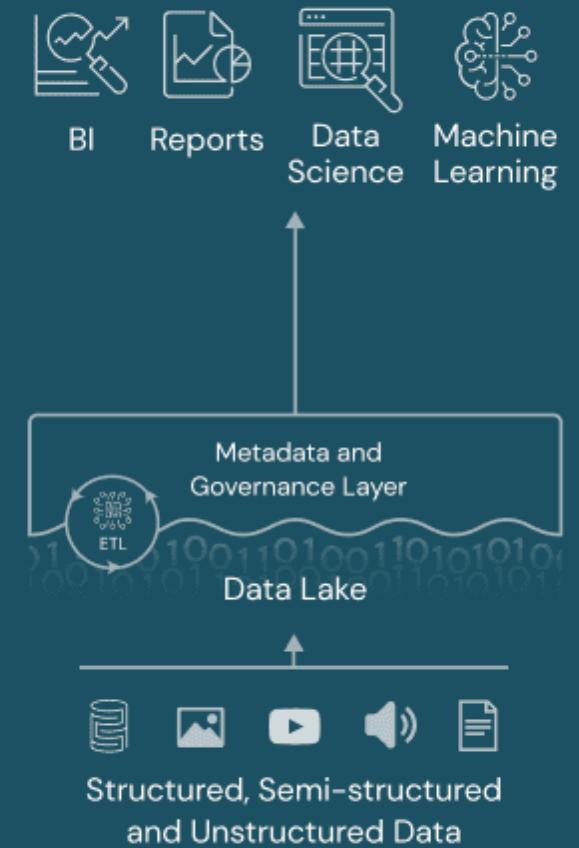
## Data Warehouse



## Data Lake



## Data Lakehouse



# Data lakehouse PRO Data lakehouse CON

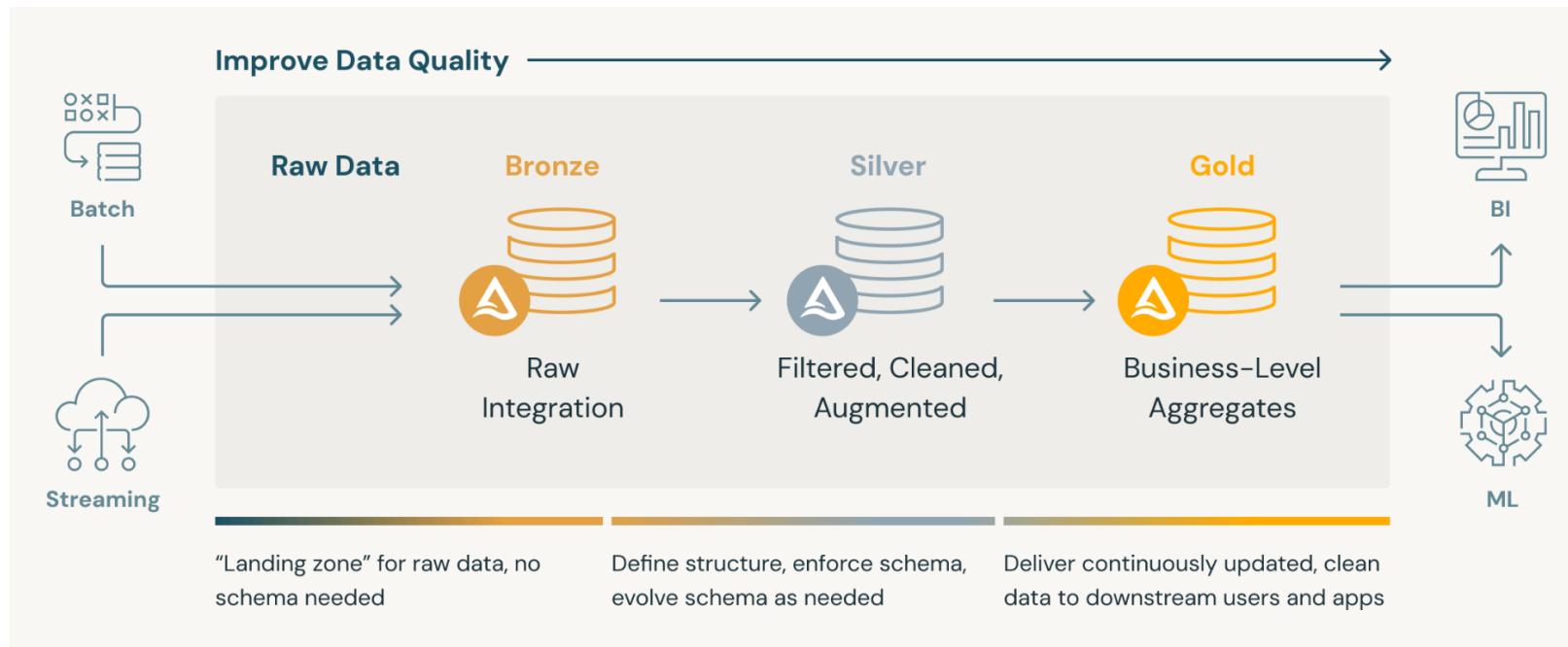
- Reduced data redundancy
  - Improved flexibility
  - Cost effectiveness (combination of DWH and Data Lake)
  - Improved data governance, data management, and security
  - Support for variety of workloads (analytics, open data formats (API, parquet))
  - Straightforward Machine Learnign
- Relatively new (concept)
  - Improved, yet immature technology
  - Offerings (Databricks)

# *Recap: DWH vs. DL vs. DLH*

	Data Warehouse	Data Lake	Data Lakehouse
<b>Storage Data Type</b>	Works well with structured data	Works well with semi-structured and unstructured data	Can handle structured, semi-structured, and unstructured data
<b>Purpose</b>	Optimal for data analytics and business intelligence (BI) use-cases	Suitable for machine learning (ML) and artificial intelligence (AI) workloads	Suitable for both data analytics and machine learning workloads
<b>Cost</b>	Storage is costly and time-consuming	Storage is cost-effective, fast, and flexible	Storage is cost-effective, fast, and flexible
<b>ACID Compliance</b>	Records data in an ACID-compliant manner to ensure the highest levels of integrity	Non-ACID compliance: updates and deletes are complex operations	ACID-compliant to ensure consistency as multiple parties concurrently read or write data

# General concepts – Data Lake

## Medallion concept (Databricks)



# ACID on Data Lake (files)

- ETL frameworks / ELT pipelines
- Assuring zones (architecture)
- Logging, storing, updating
- Delta Lake (Databricks)

# File Types supported in Spark

- .CSV – delimited text file using comma, semicolon to separate values.
- .TXT – delimited text file.
- .JSON – text file that stores simple data structures and objects. Standard for data interchange
- .Parquet – column storage file format used by hadoop systes, such as Pig, Spark, and Hive. It has binary presentation, is cross platform.
- .AVRO – open source data serialization system by Apache Hadoop. stores serialized data in binary format and schema in JSON format.
- .ORC – open source columnar data storage format. Similar to Parquet. Is cross platform
- ... and others

ORC – created by HortonWorks (feb 2013)

Apache Parquet – Created by Cloudera and twitter (may 2013)

# Avro vs. Parquet vs. ORC

BIG DATA FORMATS COMPARISON

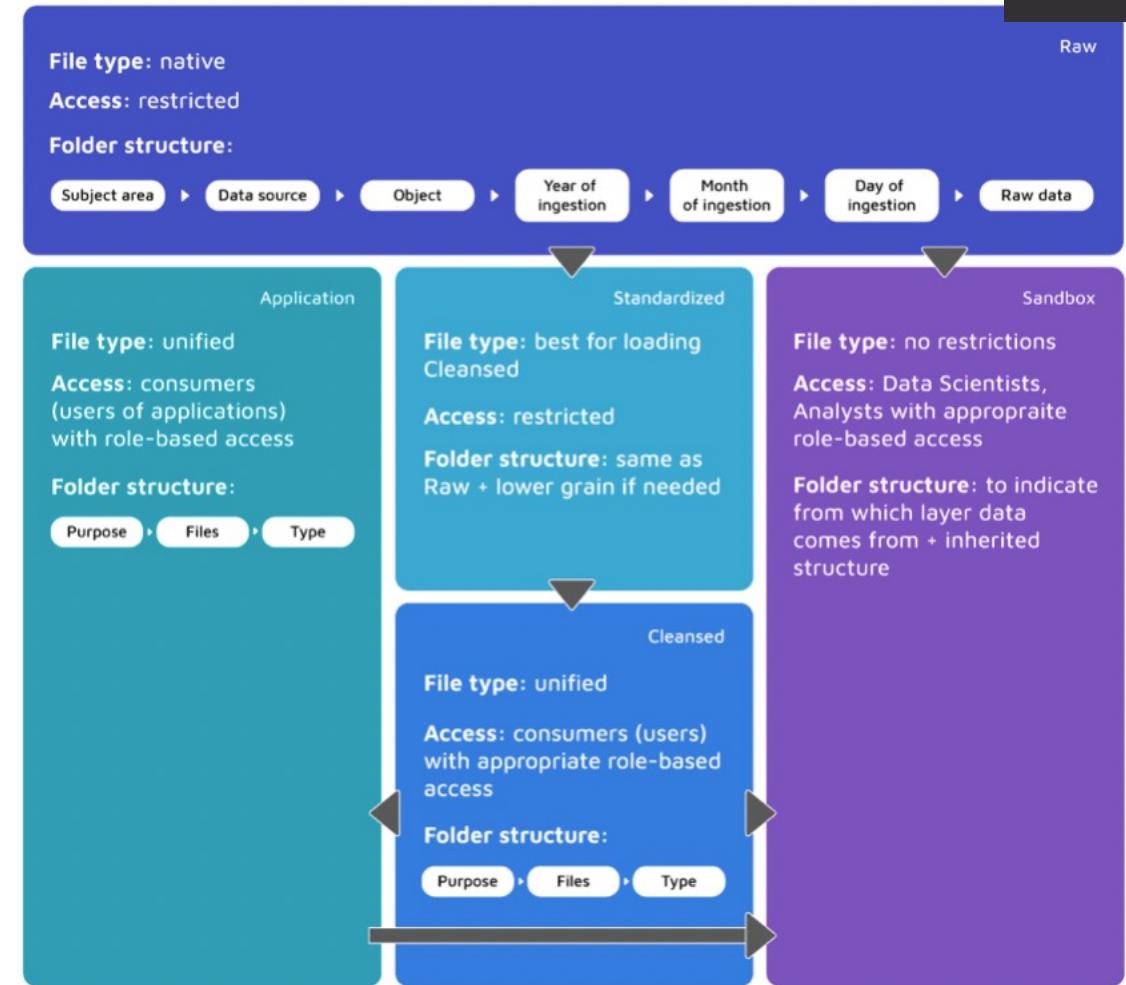


Source: Nexla analysis, April 2018



# Organizing data Lake layers

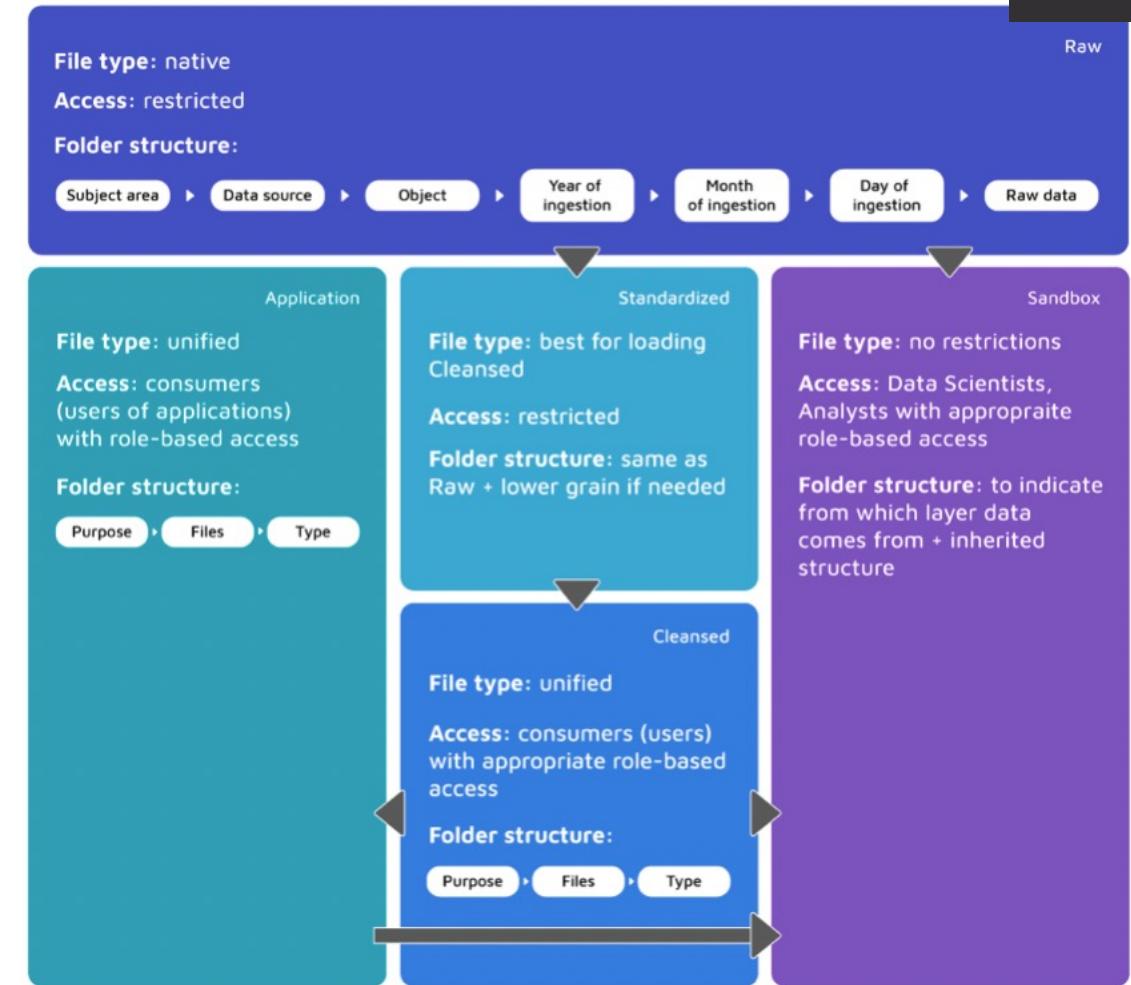
- Data Lake is a “single” repository
- Separating them into layers
- Architecture of layers / zones
- Each layer represents additional data manipulation
- Layers:
  - Raw
  - Standardized
  - Cleaned
  - Application
  - Sandbox
- Alternatives:
  - Raw > Curated > Enriched
    - Sandbox, Machine Learning
  - Ingest > Curated > Conformed > Production
    - Sandbox, Personal, Reporting, Machine Learning
  - Bronze > Silver > Gold
    - Sandbox, Work, Machine Learning





# Organizing data Lake folders

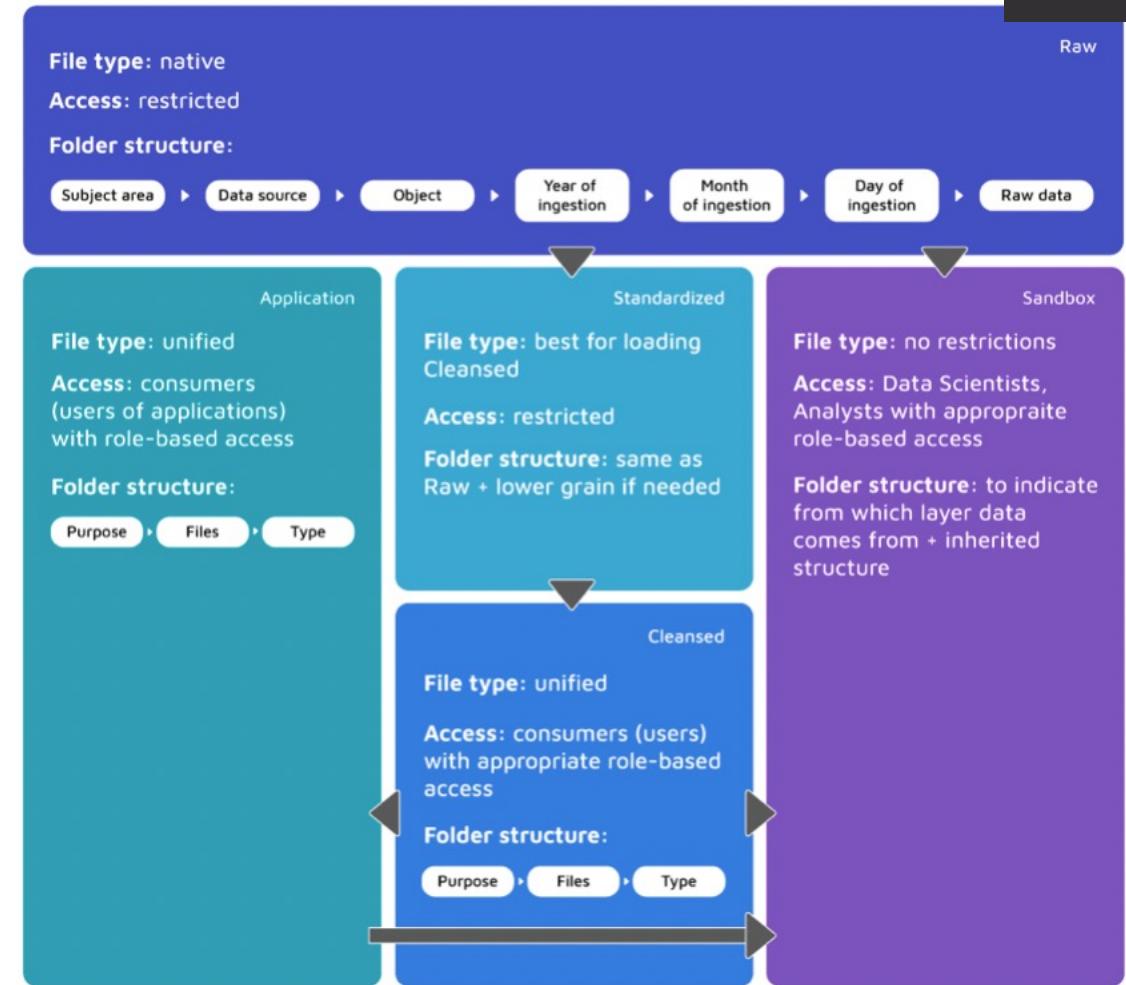
- Folder structure within Layer
- Folder by purpose:
  - Area
  - Source
  - Object
  - >>Key<<
  - Raw data
- Folders by structure
  - Purpose
  - Files
  - Type





# Organizing data Lake - Keys

- Keys are considered as “index”
- Keys help data organized
- Implement keys based on search
- Folder structure:
  - Year=2022
  - Year=2022&Month=03
  - GeoAreaID=100&Year=2022





# Files organization

Raw Layer

Sales

CustomerOrders

Year

Month

Raw\_Sales\_CustomerOrders\_2022\_03\_01.csv

Raw\_Sales\_CustomerOrders\_2022\_03\_05.csv

CustomerInvoices

Year

Month

Raw\_Sales\_CustomerInvoices\_2022\_03\_01.csv

# Structure, type, access, granulation

## Folder structure

## Data Types

Raw layer

Original format

Standardized layer

Files aggregated,  
still original format

Cleaned layer

Sparsed (faster)  
format

Application layer

Custom to  
application

## Access control

Data Engineers

Data Engineers, Data  
Scientists

Data Engineers, Data  
Scientists, Data  
Analysts

Data Engineers,  
Business people,  
Data Analysts

Granulation (e.g.:  
stream data)

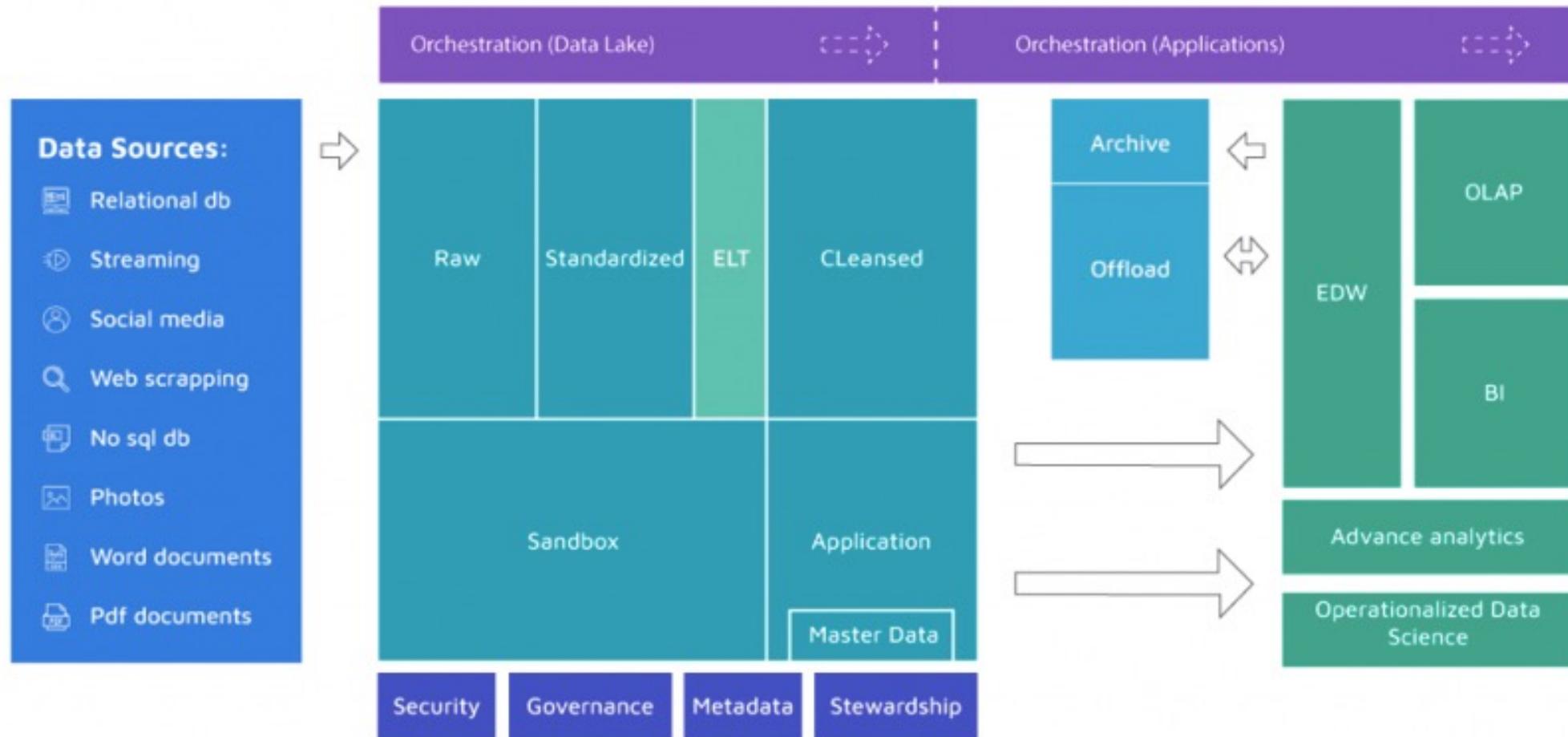
As is

Aggregate to higher  
level (e.g. from sec  
-> min)

Another abstraction  
layer above (e.g.:  
from sec -> hour)

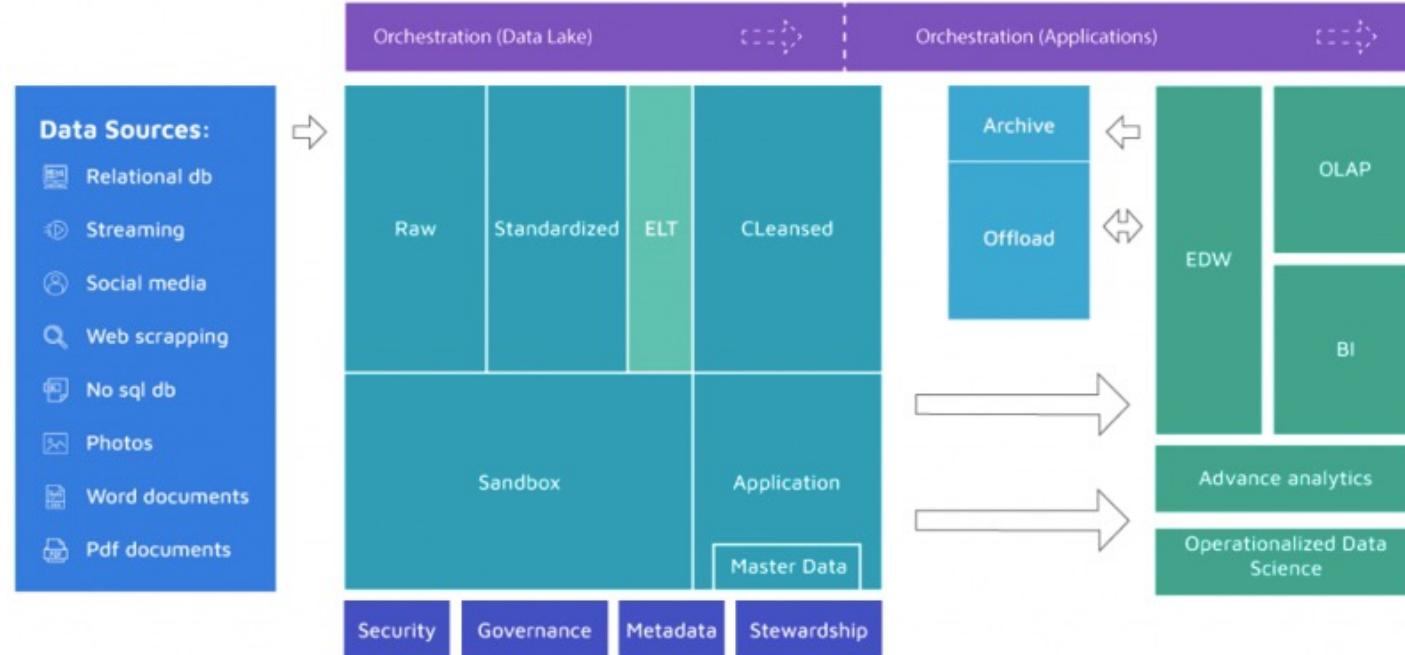
Higher granulation  
(e.g. Day, month,  
custom)

# Data Lake Architecture



# Data Lake Components

- Security
- Governance
- Metadata
- Stewardship
- Master data
- Archiving
- Orchestration



# Architecture designs

Architecture designs can be influenced by:

- Data load patterns (hot/cold, real-time, stream, increment/full)
- Time partitioning
- Data type (relational, time series, blob)
- Subject areas and sources
- Security and roles
- Retention policies (temporary, permanent, time-fixed)
- Business impact (Core, Critical, High, Medium, Low)
- SLA and regional regulations
- GDPR and confidential classifications (public use, internal use, financial, Government, supplier/partner confidential)
- Workspaces (Fabric)

# General concepts – Naming conventions



Keep in on organizational level

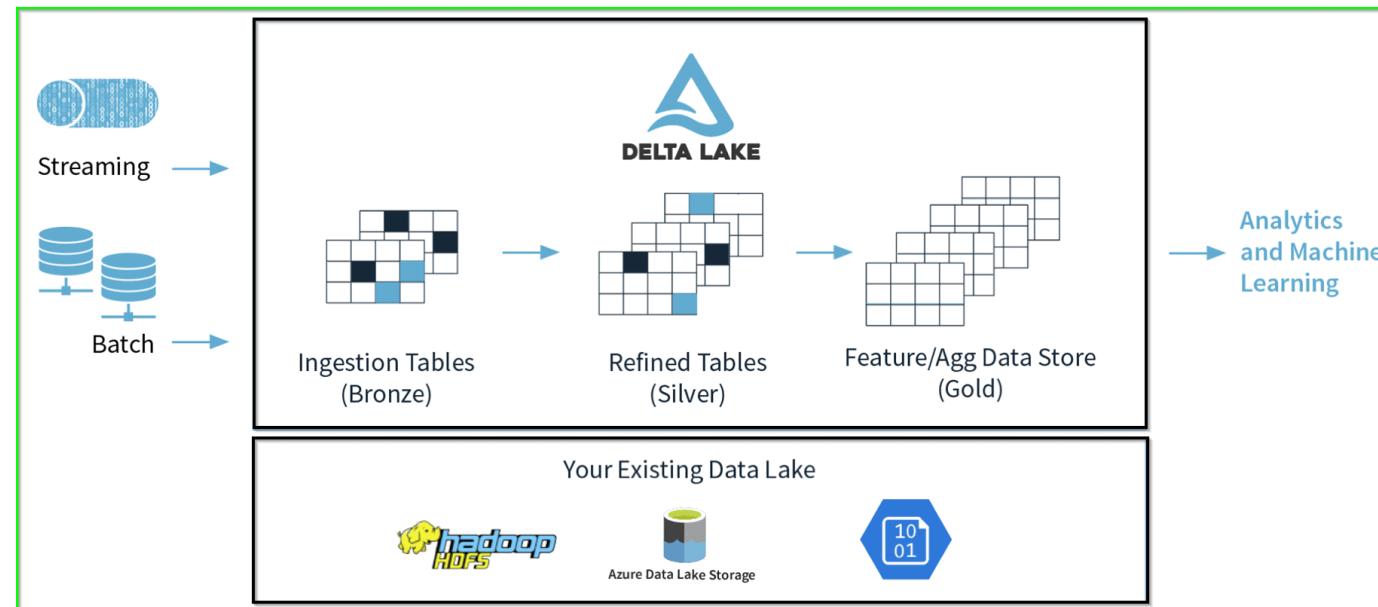
Granularity

RBAC access



# Delta lake

- Open source data storage layer that unifies ACID transactional and scalable metadata management and batch and/or streaming data processing. It also offers time travel and CDC, SCD, streaming upserts operations
- Sits on top of your existing Data Lake in tandem with Apache Spark APIs.

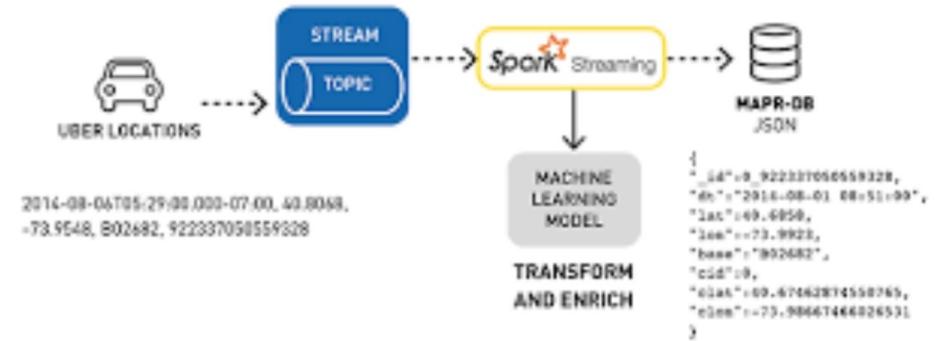


# General concepts – Spark

- Fast, expressive, general-purpose in-memory cluster computing framework compatible with Apache Hadoop and built around speed, ease of use and streaming analytics  
Faster and easier than Hadoop MapReduce\*  
Large community and 3rd party libraries
- Provides high-level APIs (Java, Scala, Python, R) Supports variety of workloads
  - interactive queries, streaming, machine learning and graph processing

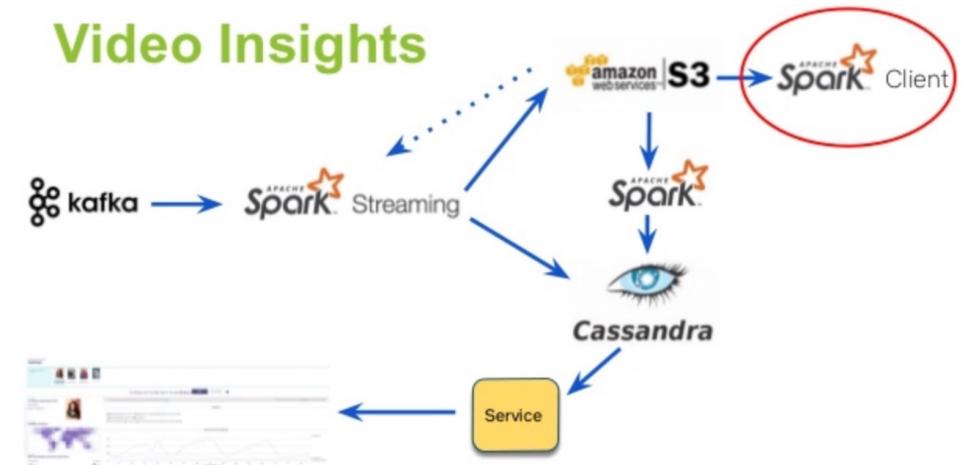
# Spark Use-Cases

- Logs processing (Uber)
- Event detection and real-time analysis
- Interactive analysis
- Latency reduction
- Advanced ad-targeting (Yahoo!)
- Recommendation systems (Netflix, Pinterest)
- Fraud detection
- Sentiment analysis (Twitter, X)



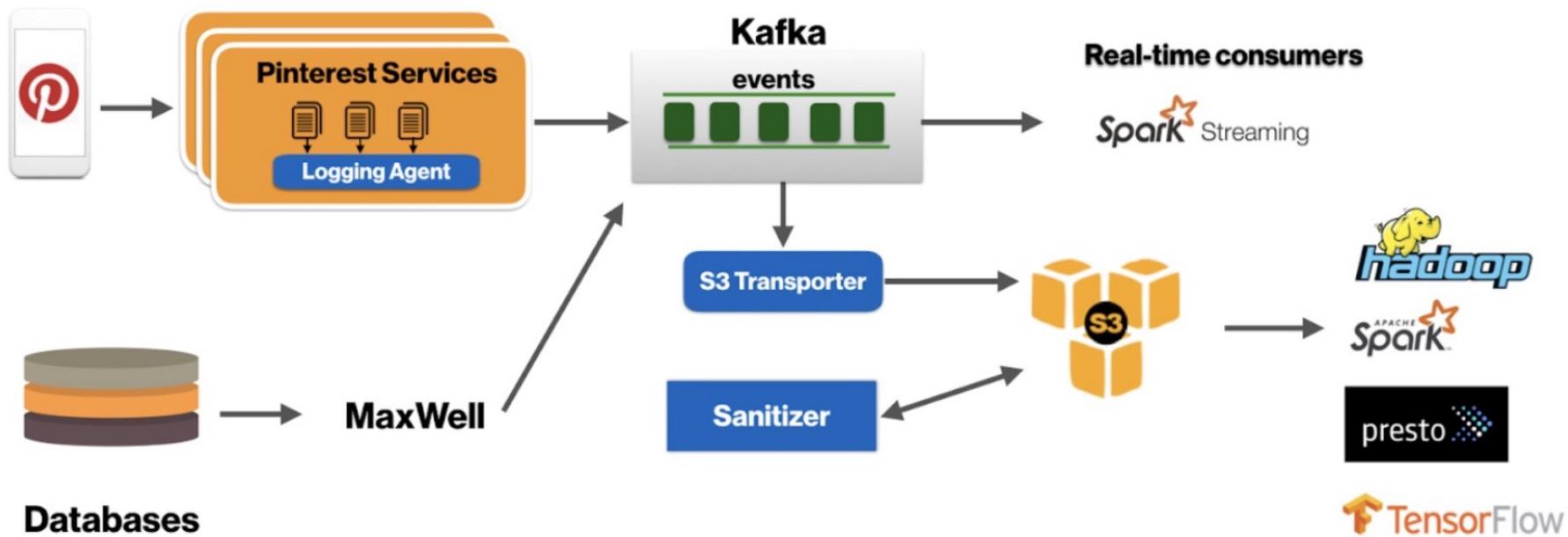
# Spark Use-Cases

- Logs processing (Uber)
- Event detection and real-time analysis
- Interactive analysis
- Latency reduction
- Advanced ad-targeting (Yahoo!)
- Recommendation systems (Netflix, Pinterest)
- Fraud detection
- Sentiment analysis (Twitter)



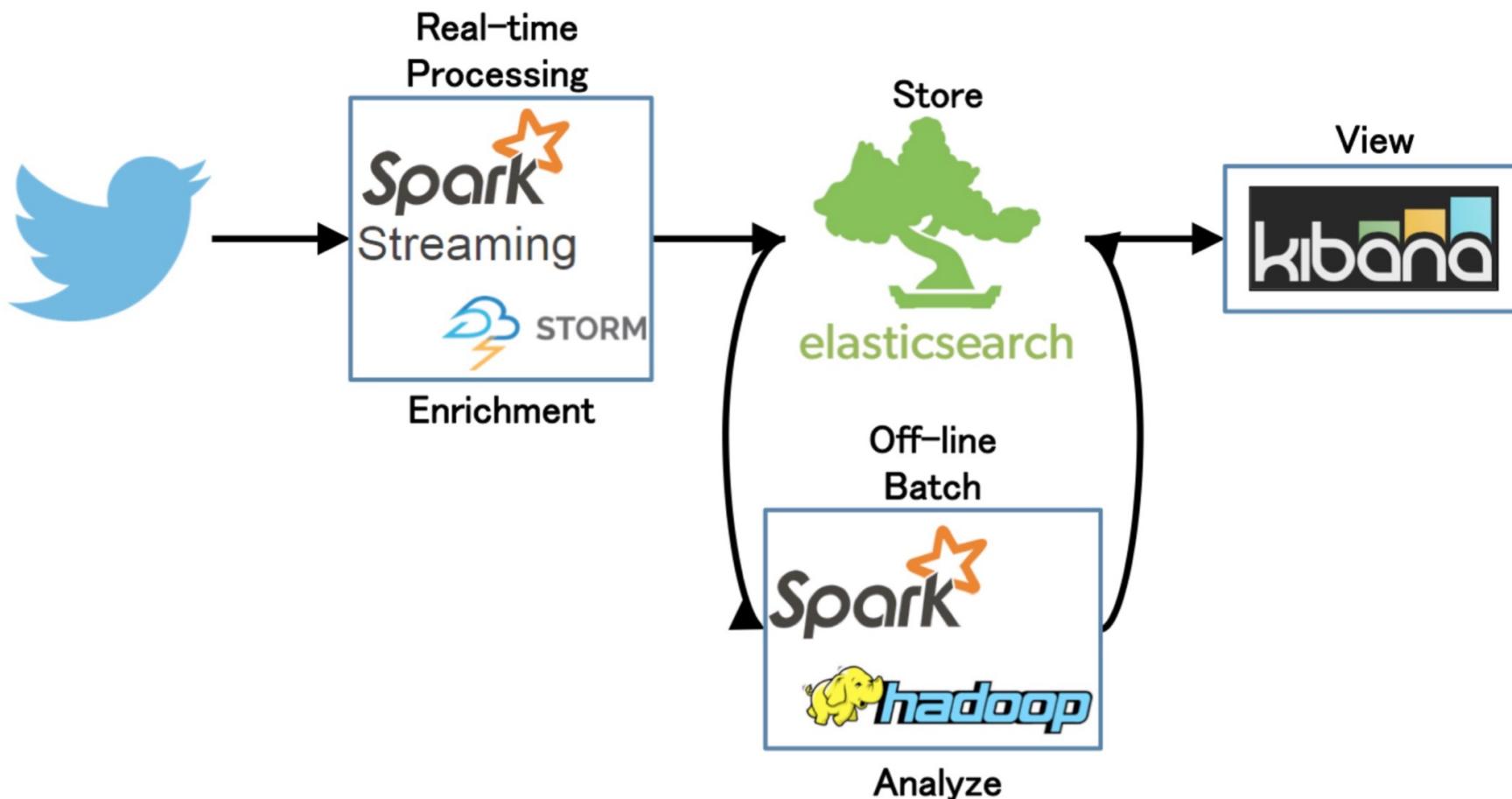
# Spark Use-Cases

- Recommendation systems (Netflix, Pinterest)



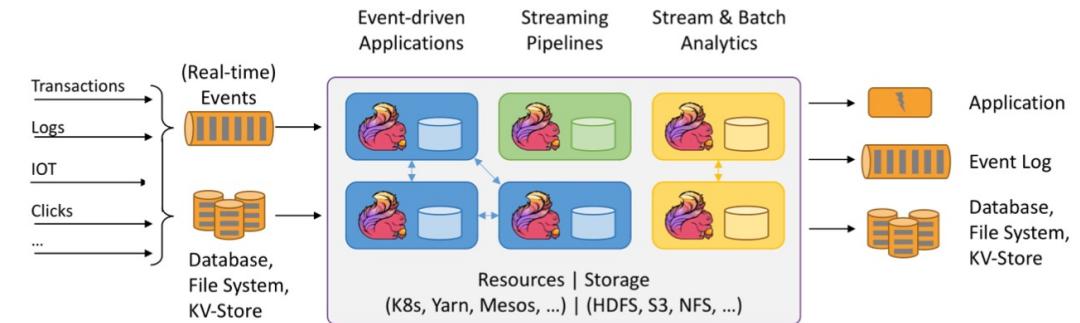
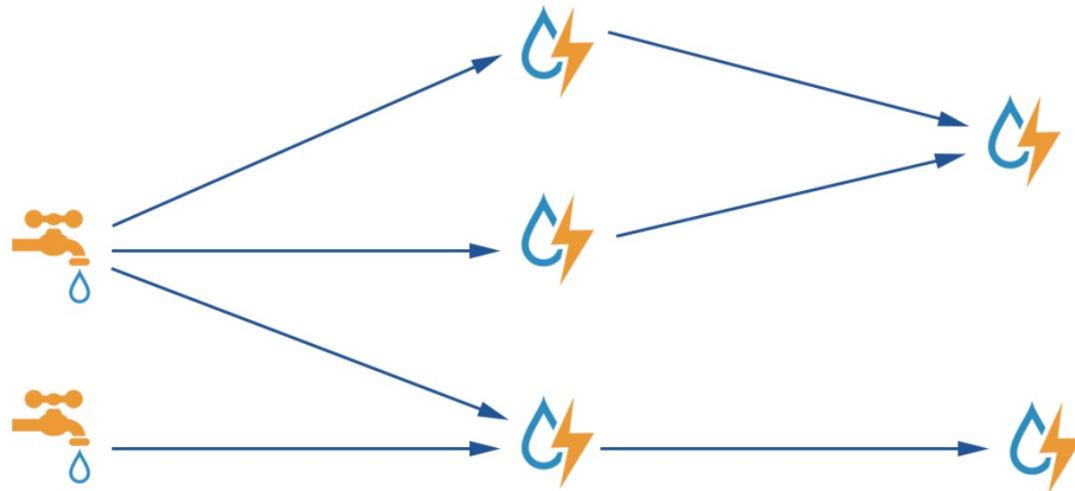
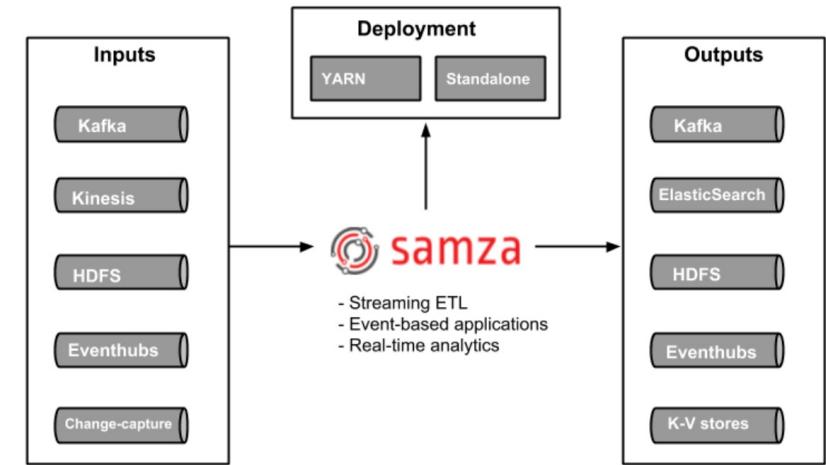
# Spark Use-Cases

- Sentiment analysis (Twitter, X)



# Apache Spark “real-time”

- Apache Samza (library/framework)
- Apache Storm (real-time stream processing)
- Apache Flink (native streaming support for all workloads)



# Apache Spark 3.x.x

- Latest stable release – v3.5.3. (Sep 24, 2024)
- Latest preview release – v4.0 (Sep 26, 2024)
- Improvements over Spark 2 (v2.4.1)
  - Python 2 deprecated
  - Adaptive execution of Spark SQL (merging intermediate results among workers)
  - Dynamic partition pruning
  - Support for deep learning (GPU support for Nvidia, AMD, Intel)
  - Better Kubernetes integration
  - Graph features (Morpheus as extension of Cypher, neo4j support)
  - ACID transactions (for Delta Lake storage)
  - Apache Arrow data format integration (columnar format for analytics)

# Hadoop MapReduce vs. Apache Spark

## Big data frameworks

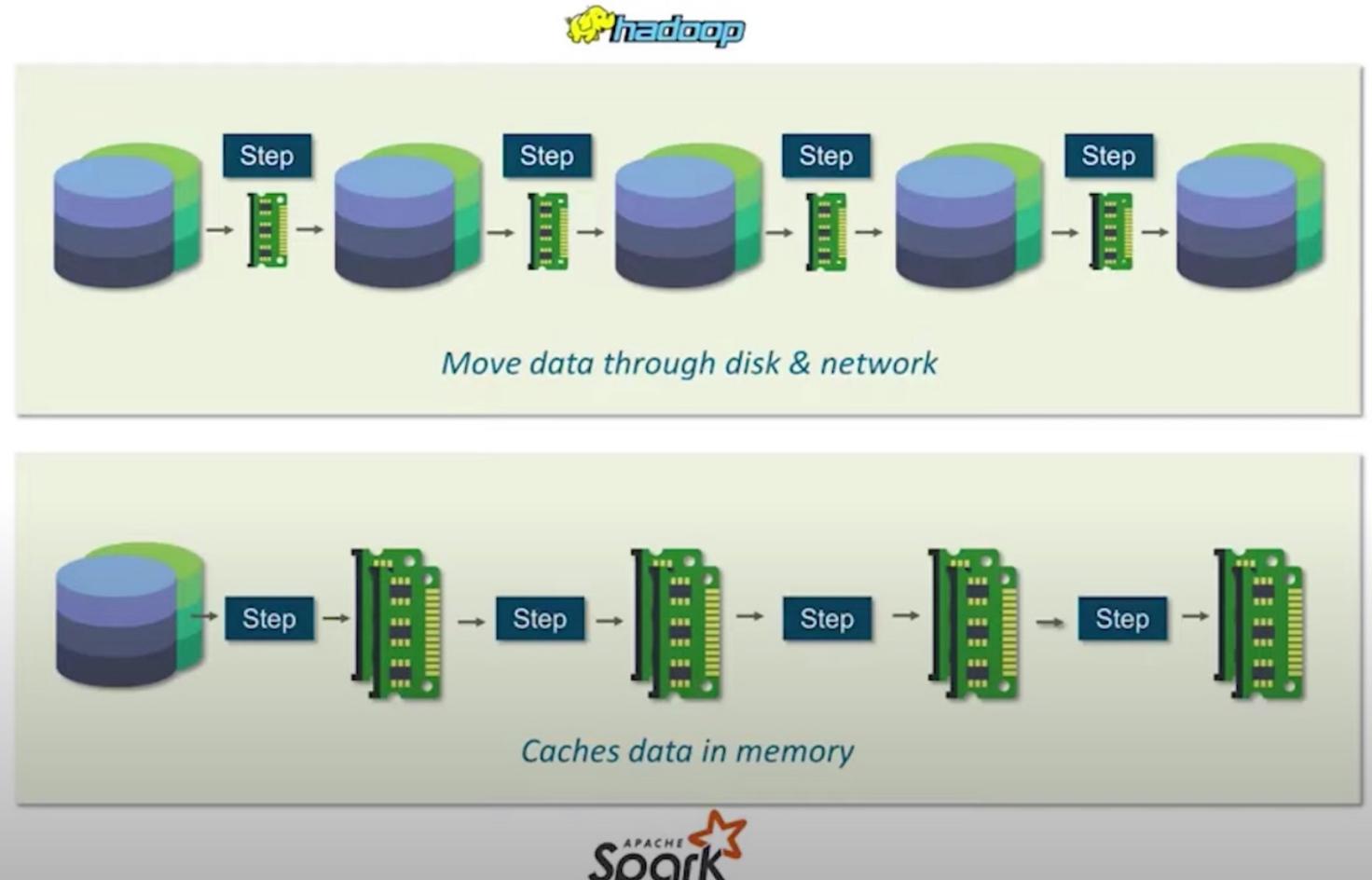
- Performance
- Ease of use
- Costs
- Data processing
- Fault tolerance
- Security

## Hadoop

Archival data analysis

## Spark

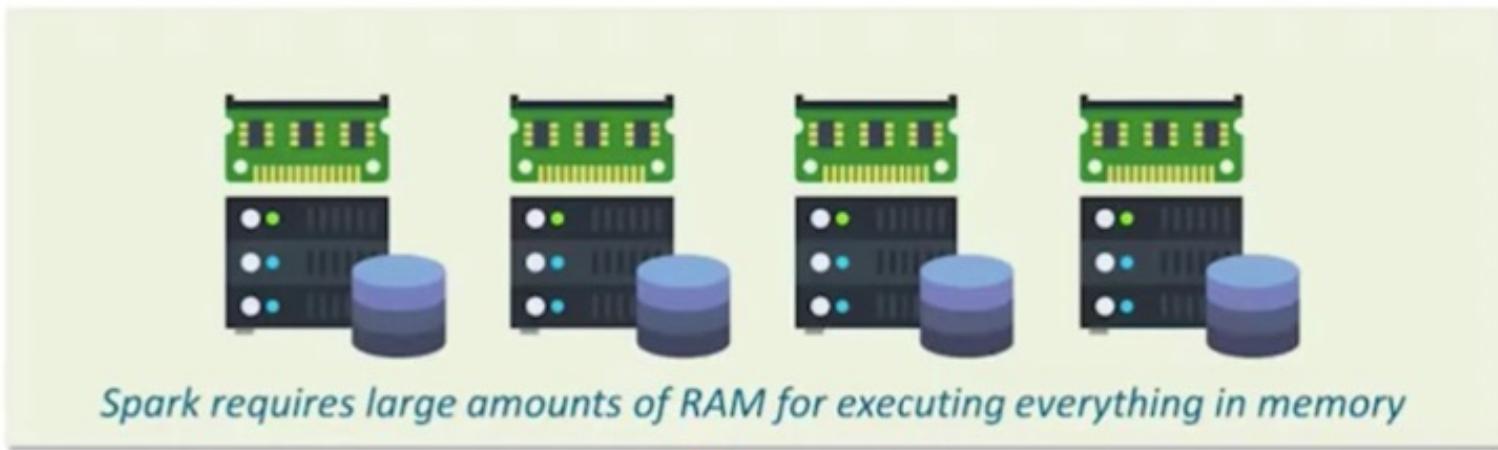
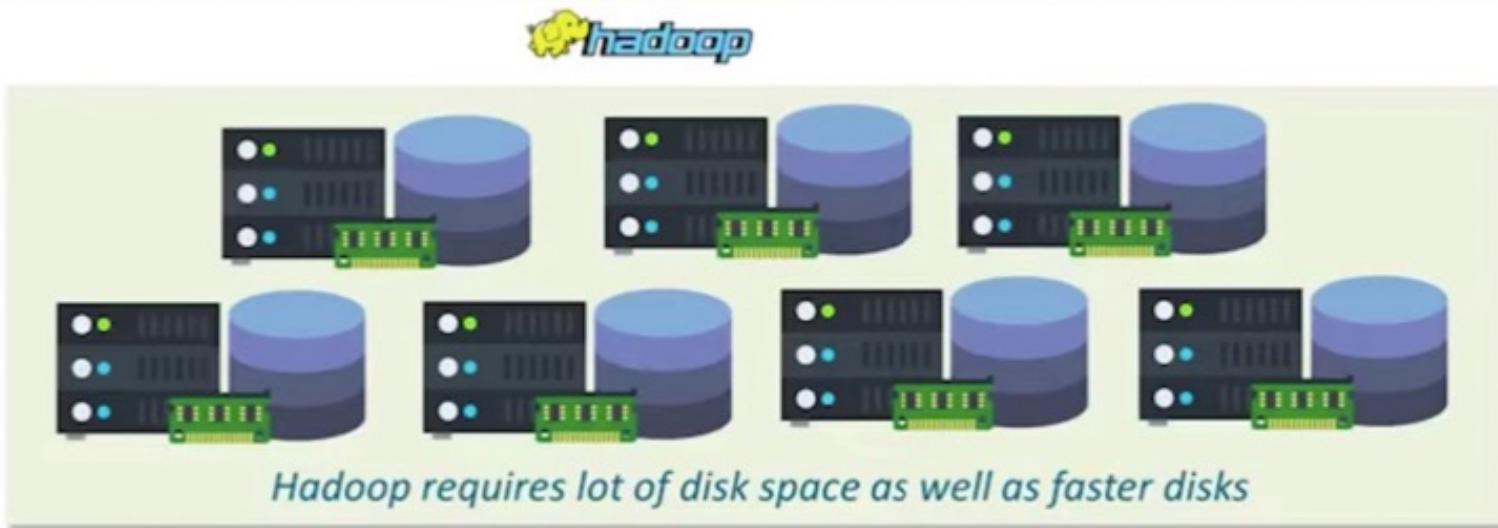
Real-time data analysis



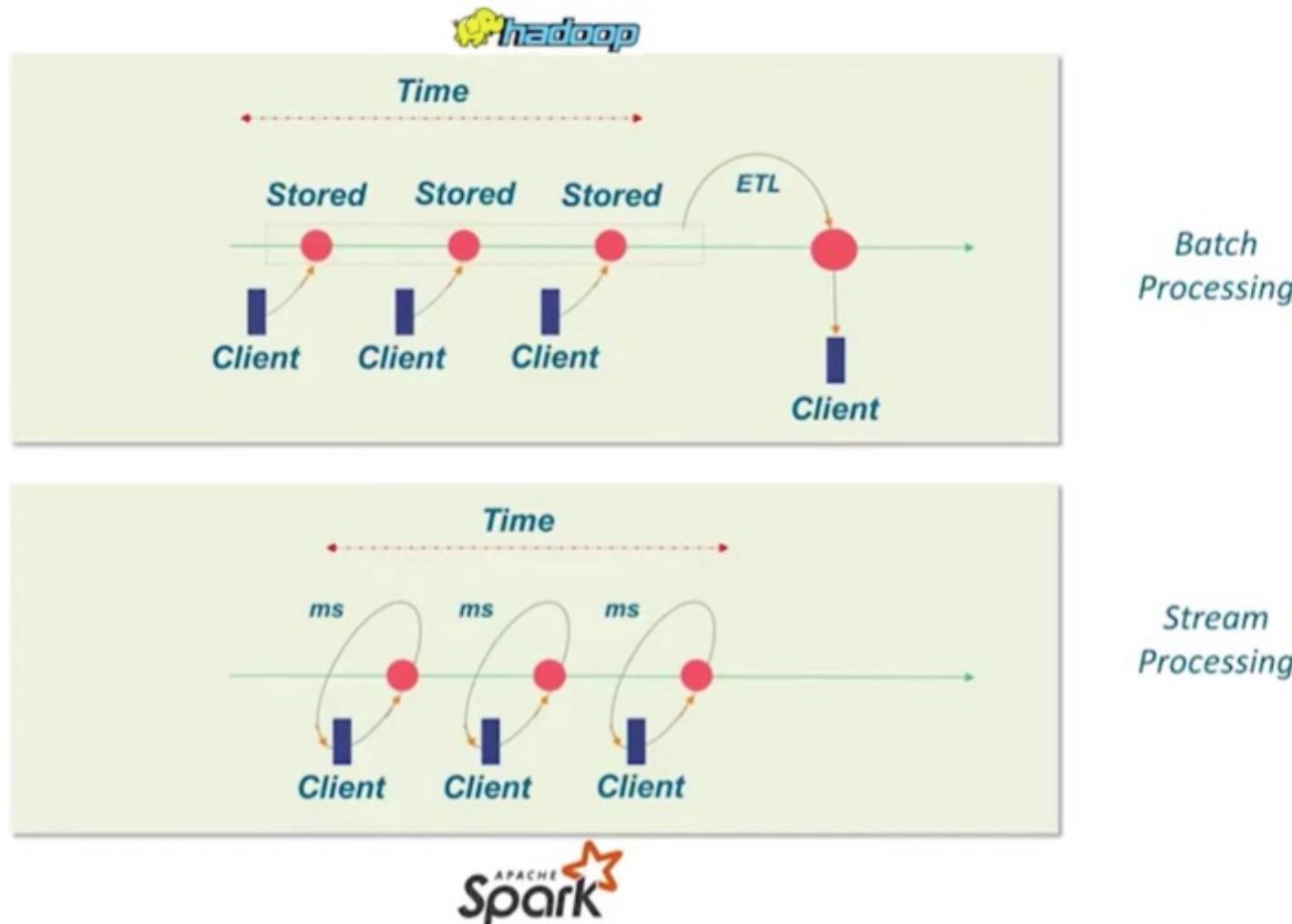
# Hadoop MapReduce vs. Apache Spark



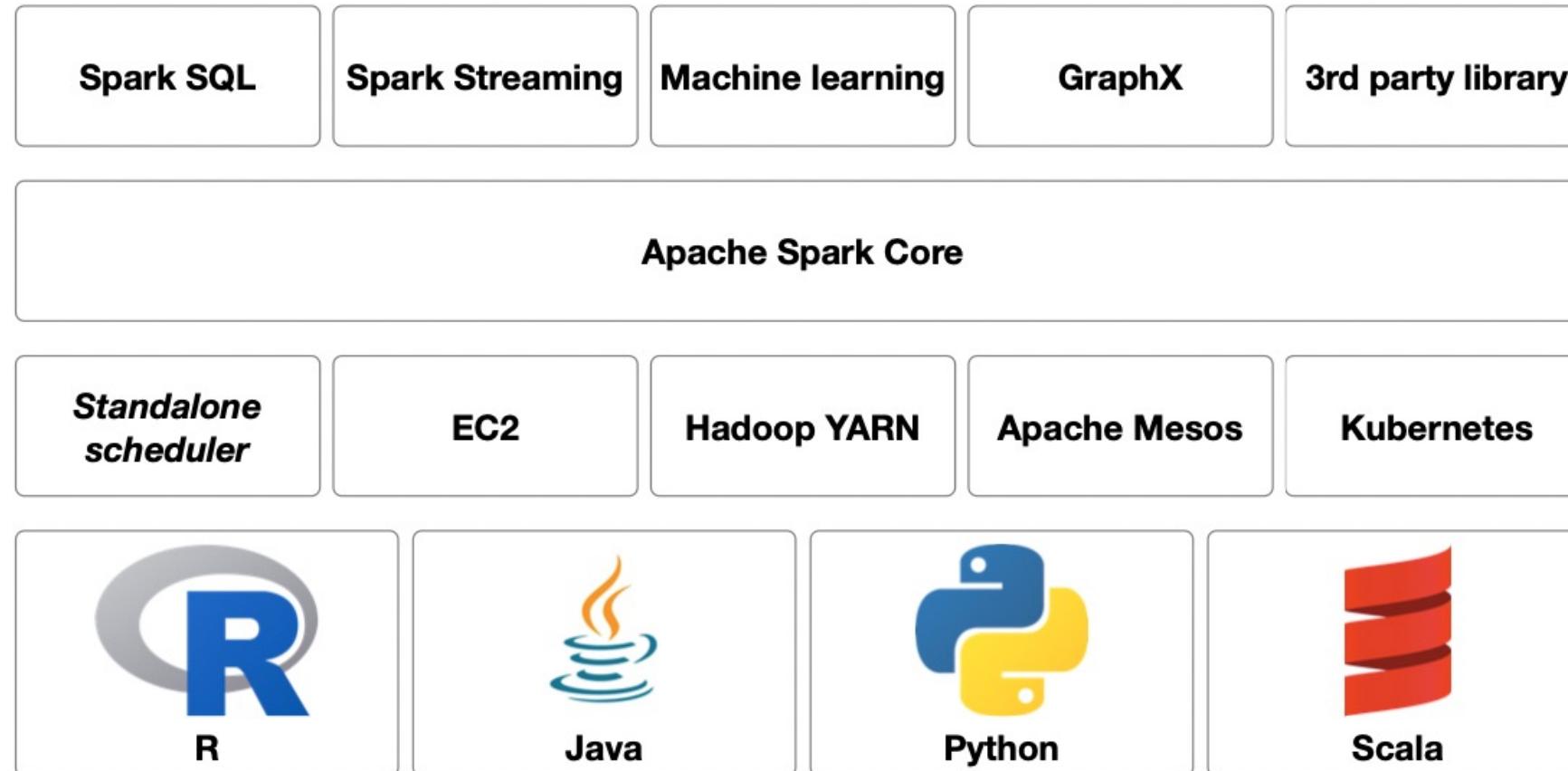
# Hadoop MapReduce vs. Apache Spark



# Hadoop MapReduce vs. Apache Spark



# Apache Eco-System



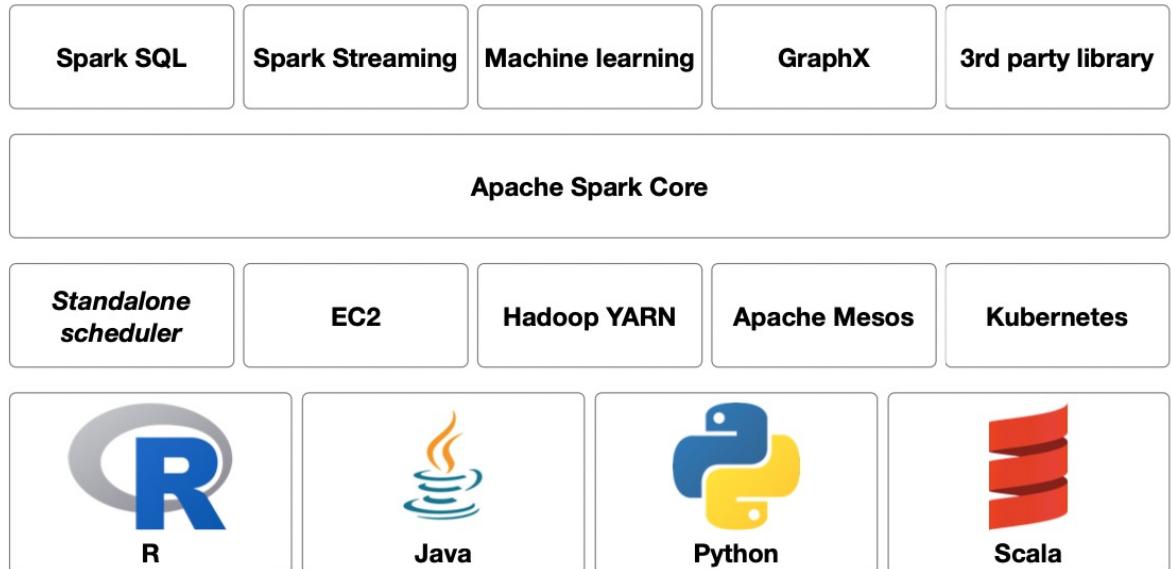
# Spark Core functionalities

## Core functionalities

- task scheduling
- memory management
- fault recovery
- storage systems interaction
- etc.

## Basic data structure definitions/abstractions

- Resilient Distributed Data sets (RDDs)  
main Spark data structure
- Directed Acyclic Graph (DAG)



# Ecosystem: Spark SQL

Structured data manipulation

- Data Frames definition

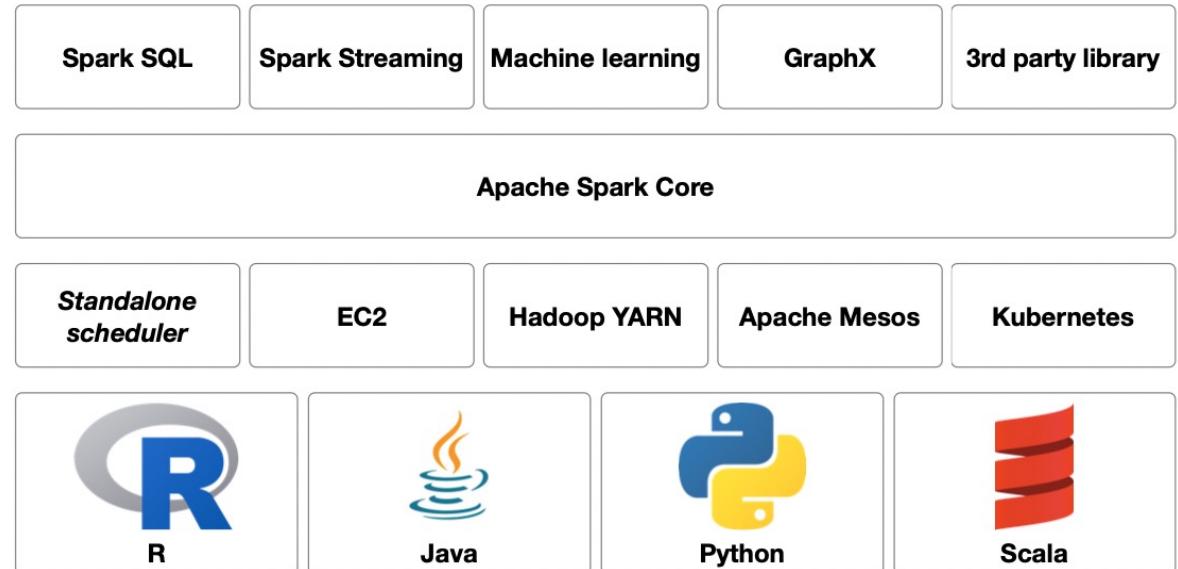
Table-like data representation

- RDDs extension
- Schema definition

SQL queries execution

Native support for schema-based data

- Hive, Parquet, JSON, CSV



# Ecosystem: Spark Streaming

Data analysis of streaming data

- e.g. tweets, log messages, SCADA

Features of stream processing

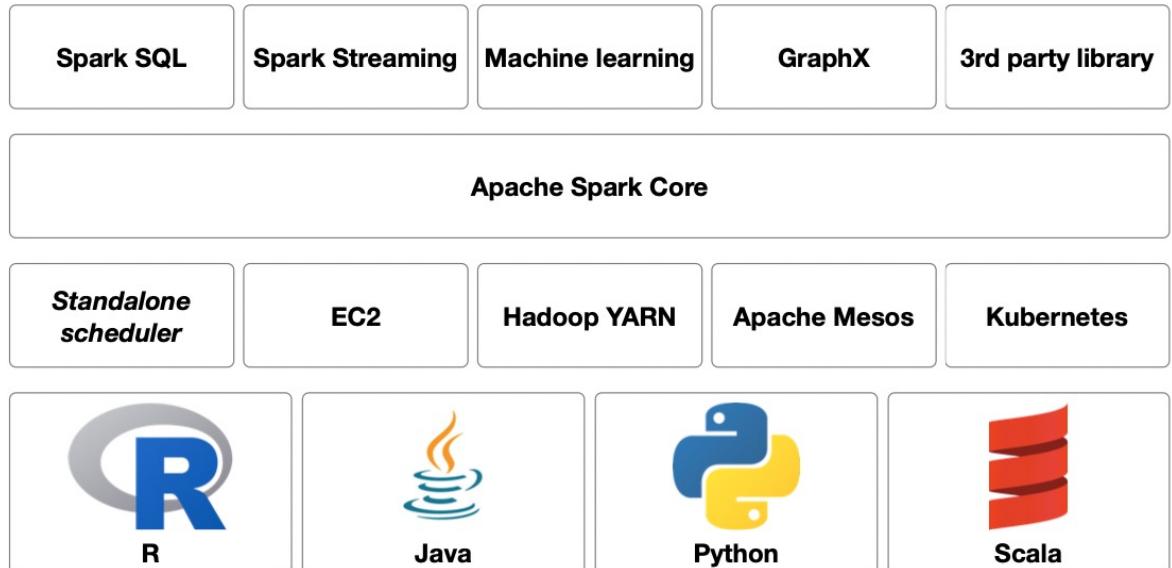
- High-throughput
- Fault-tolerant
- End-to-end
- Exactly once

High-level abstraction of a discretized stream

- Stream represented as a sequence of RDD

With Spark 2.3.x and above continuous processing

- End-to-end low latency (< 1ms)



# Ecosystem: Spark MLlib for Machine Learning

Common ML functionalities

ML Algorithms

common learning algorithms such as classification, regression, clustering, and collaborative filtering

Featurization

feature extraction, transformation, dimensionality reduction, and selection

Pipelines

tools for constructing, evaluating, and tuning ML

Pipelines

Persistence

saving and load algorithms, models, and Pipelines

Utilities

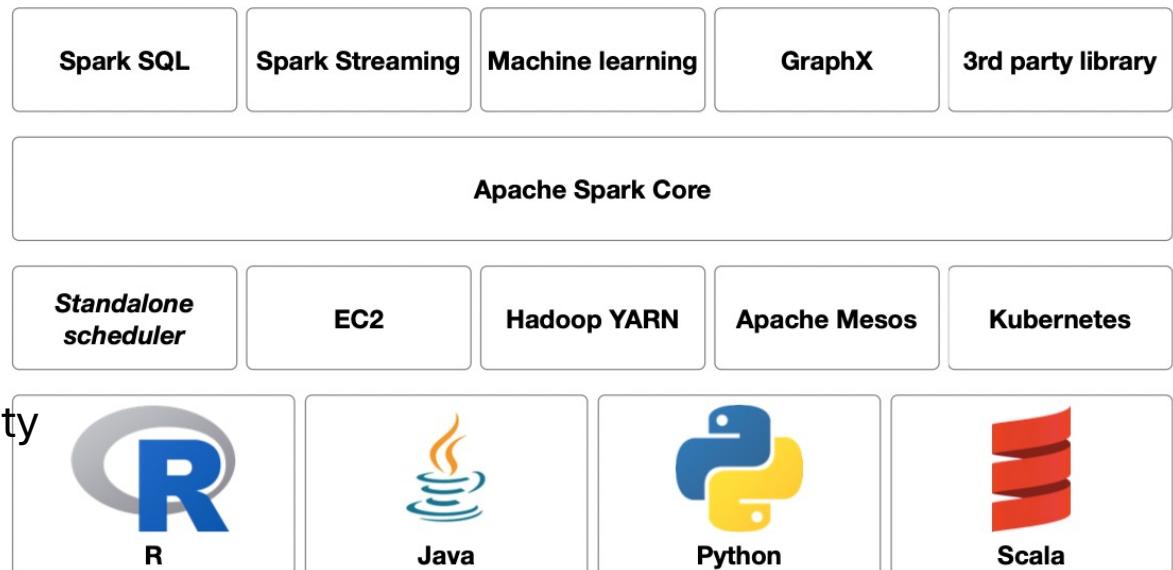
linear algebra, statistics, data handling, etc.

Two APIs

RDD-based API (`spark.mllib package`)

Spark 2.0+, DataFrame-based API (`spark.ml package`)

Methods scale out across the cluster by default



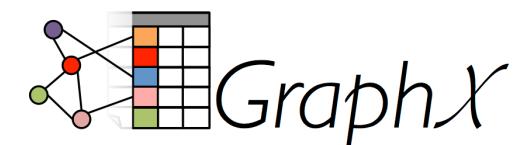
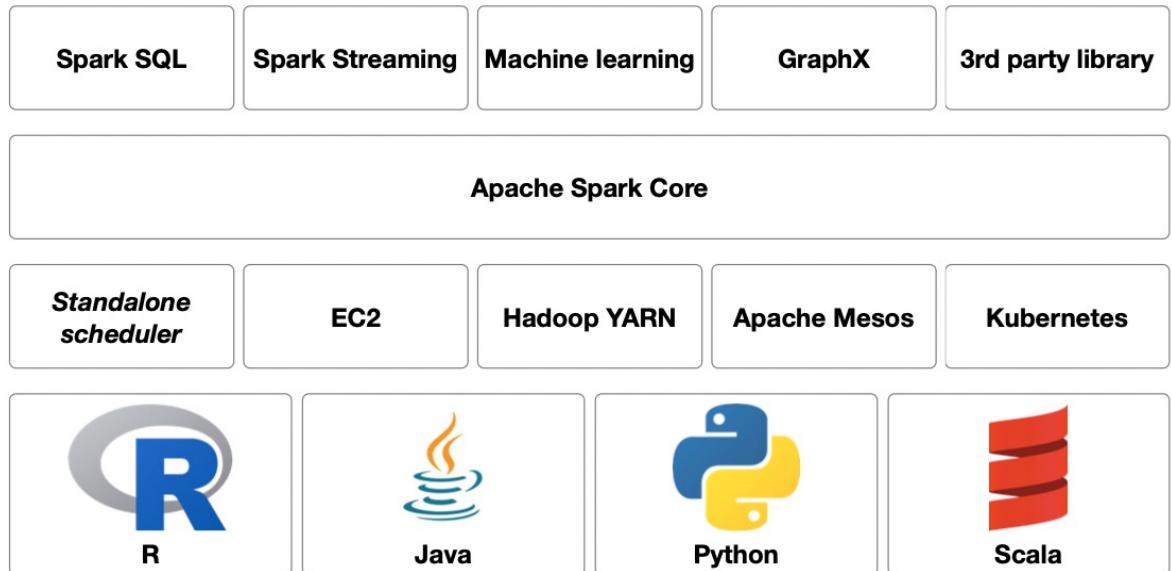
# Ecosystem: GraphX

Support for graphs and graph-parallel computation

- Extension of RDDs (Graph)
- Direct multigraph with properties on vertices and edges

Graph computation operators

- subgraph, joinVertices, and aggregateMessages, etc.
- Pregel API support



# Spark Modes

Spark operates in 4 different modes:

- **Standalone Mode:** Here all processes run within the same JVM process.
- **Standalone Cluster Mode:** In this mode, it uses the Job-Scheduling framework in-built in Spark.
- **Apache Mesos:** In this mode, the work nodes run on various machines, but the driver runs only in the master node.
- **Hadoop YARN:** In this mode, the drivers run inside the application's master node and is handled by YARN on the Cluster.

# Spark Context

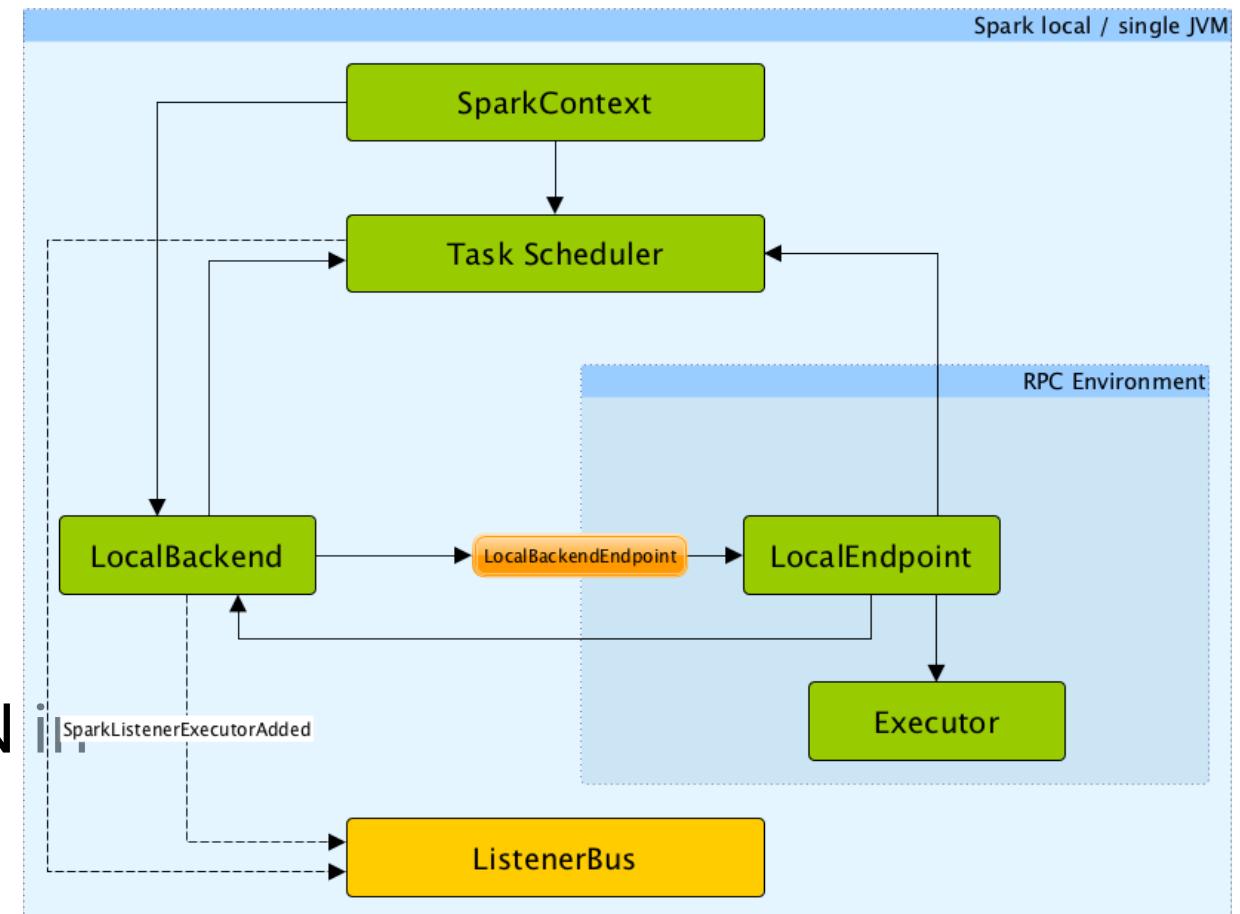
- driver Spark application which communicates the user commands to the Spark Workers
- `SparkContext` is an object, which coordinates with the cluster manager about the resources required for execution and the actual tasks that need to be executed
- Submitting spark applications: [Submitting Applications - Spark 3.2.1 Documentation \(apache.org\)](#)

# Execution modes

- Local mode
  - „Pseudo-cluster“ ad-hoc setup using script
- Cluster mode
  - Running via cluster manager
- Interactive mode
  - Direct manipulation in a shell (*pyspark*, *spark-shell*)

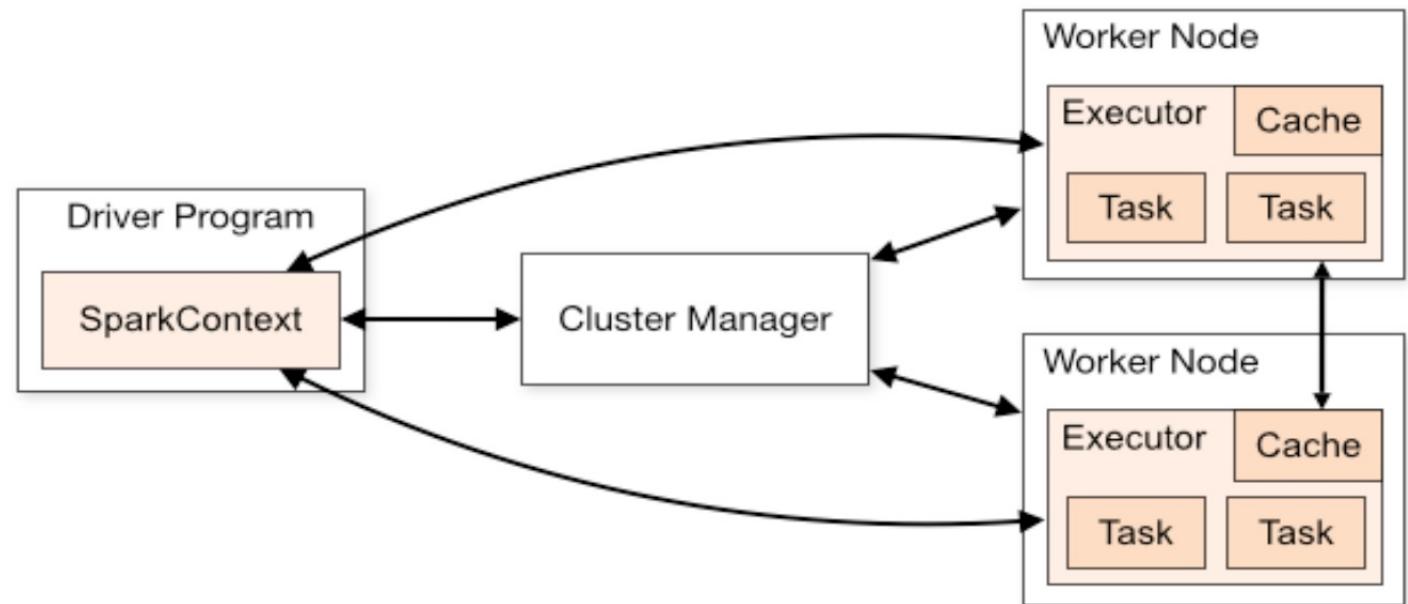
# Spark execution modes Local mode

- Non-distributed single-JVM deployment mode
- Spark library spawns (in a JVM)
  - driver
  - scheduler
  - master
  - executor
- Parallelism is the number of threads defined by a parameter N in a spark master URL
  - *local[N]*



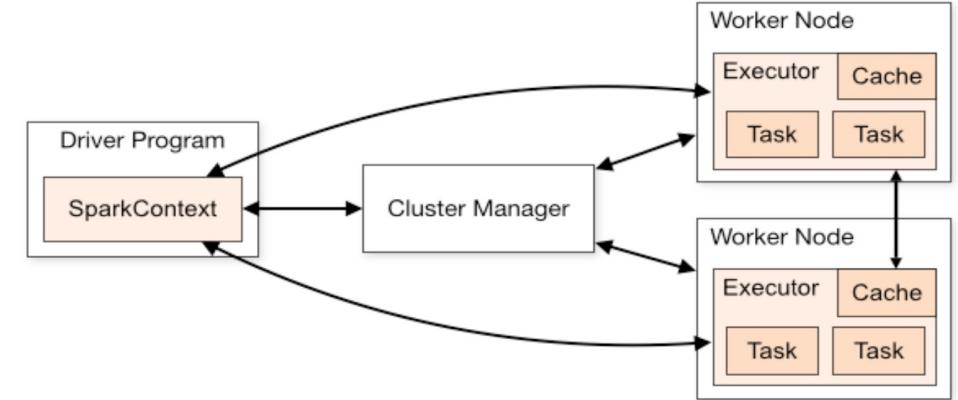
# Spark execution modes Cluster mode

- Deployment on a private cluster
  - Apache Mesos
  - Hadoop YARN
  - Kubernetes
  - Standalone mode, ...



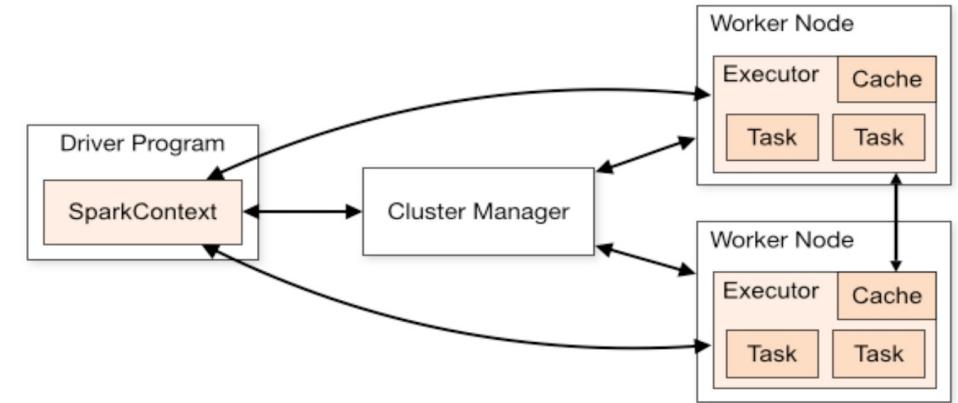
# Spark execution modes Cluster mode

- Components
  - Worker
    - Node in a cluster, managed by an executor
    - Executor manages computation, storage and caching
  - Cluster manager
    - Allocates resources via SparkContext with Driver program
  - Driver program
    - A program holding SparkContext and main code to execute in Spark
    - Sends application code to executors to execute
    - Listens to incoming connections from executors



# Spark execution modes **Cluster mode**

- Deploy modes (*standalone clusters*)
  - Client mode (default)
    - Driver runs in the same process as client that submits the app
  - Cluster mode
    - Driver launched from a worker process
    - Client process exits immediately after application submission



# Spark execution process

## 1. Data preparation/import

- RDDs creation – i.e. parallel dataset with partitions

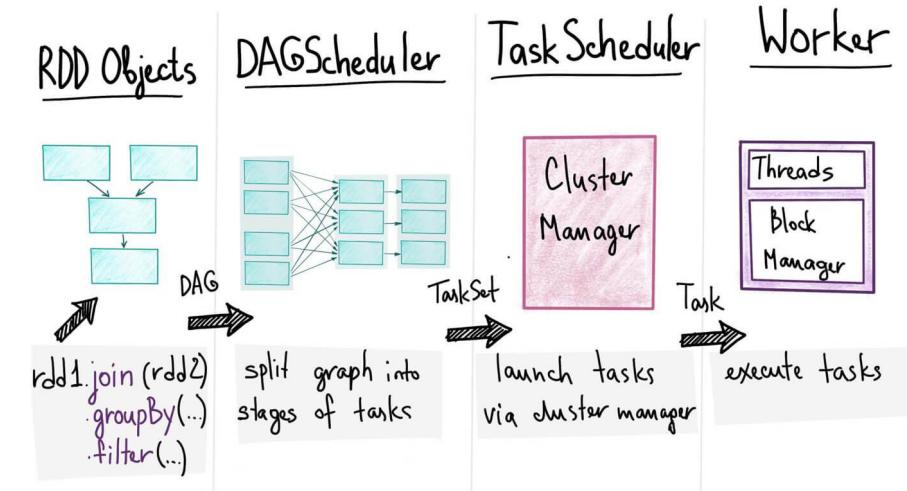
## 2. Transformations/actions definition\*

- Creation of tasks (units of work) sent to one executor
- Job is a set of tasks executed by an action\*

## 3. Creation of a directed acyclic graph (DAG)

- Contains a graph of RDD operations
- Definition of stages – set of tasks to be executed in parallel (i.e. at a partition level)

## 4. Execution of a program (application)



@luminousmen.com

# Shell vs.Cluster mode

- Shell Is interactive
- Cluster is for connecting into private network with several machines
- In cluster mode, it's called submitting applications

# Submitting applications

Pseudo code

```
./bin/spark-submit \  
--class <main-class> \  
--master <master-url> \  
--deploy-mode <deploy-mode> \  
--conf <key>=<value> \  
... # other options <application-jar> \  
[application-arguments]
```

# Applications: Locally, in Standalone cluster

```
# Run application locally on 8 cores
./bin/spark-submit \
--class org.apache.spark.examples.SparkPi \
--master local[8] \
/path/to/examples.jar \
100
```

```
# Run a Python application on a Spark standalone cluster
./bin/spark-submit \
--master spark://207.184.161.138:7077 \
examples/src/main/python/pi.py \
1000
```

# Applications: Mesos and Kubernetes

```
# Run on a Mesos cluster in cluster deploy mode with supervise
./bin/spark-submit \
--class org.apache.spark.examples.SparkPi \
--master mesos://207.184.161.138:7077 \
--deploy-mode cluster \
--supervise \
--executor-memory 20G \
--total-executor-cores 100 \
http://path/to/examples.jar \
1000
```

```
# Run on a Kubernetes cluster in cluster deploy mode
./bin/spark-submit \
--class org.apache.spark.examples.SparkPi \
--master k8s://xx.yy.zz.ww:443 \
--deploy-mode cluster \
--executor-memory 20G \
--num-executors 50 \
http://path/to/examples.jar \
1000
```

## Module 2

Covering data storage and compute offerings.

For Microsoft Fabric we will cover the outstanding OneLake data store, learn how it works, what it is capable of and how to get navigate.

In Azure Machine Learning we will look into different Azure data store options like ADSL Gen2, and learn about creating datastores and datasets.

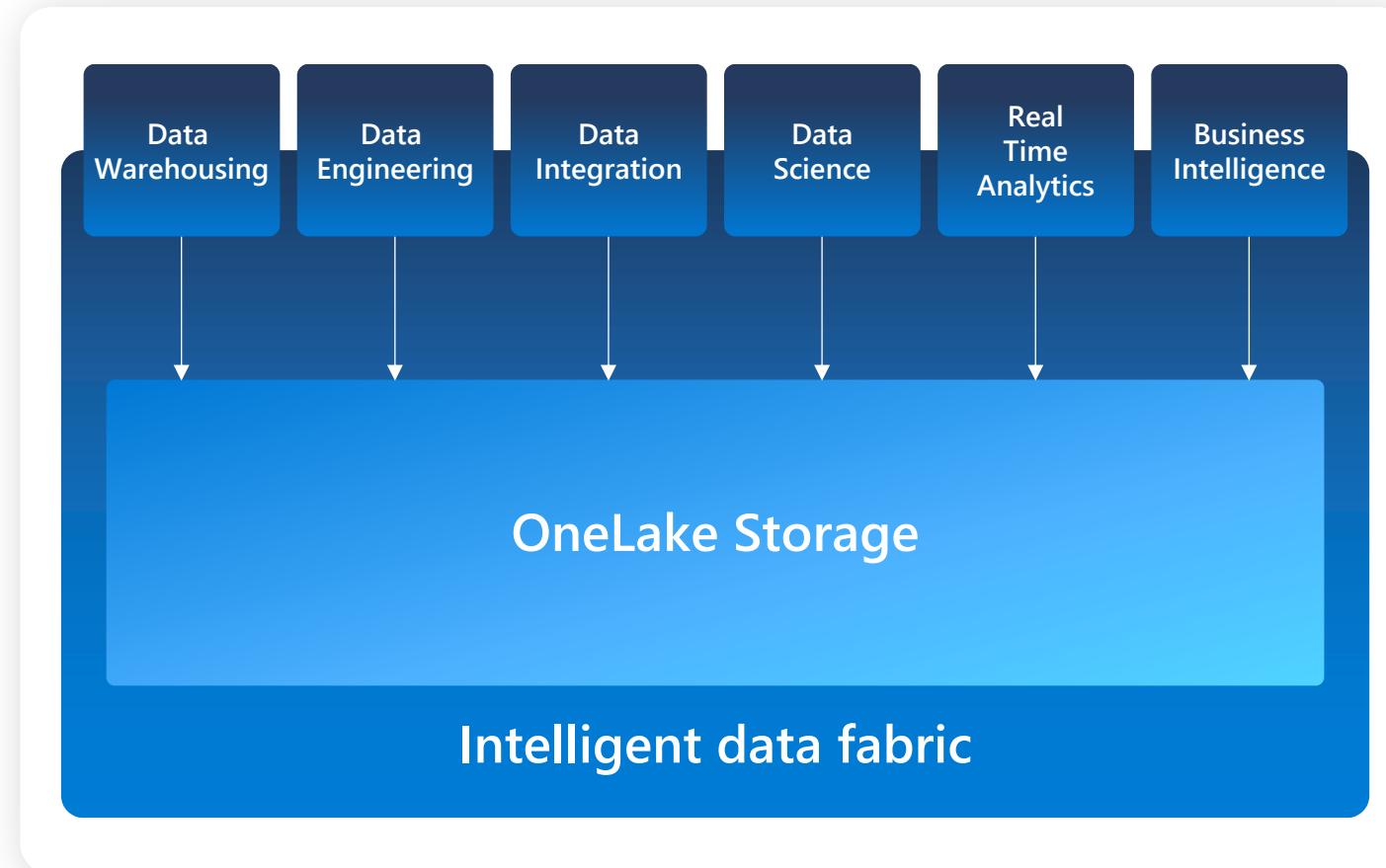
# Concepts

- Distributed Computation
- One Lake
- Compute



# OneLake for all Data

"The OneDrive for Data"



A single SaaS lake for the whole organization

Provisioned automatically with the tenant

All workloads automatically store their data in the OneLake workspace folders

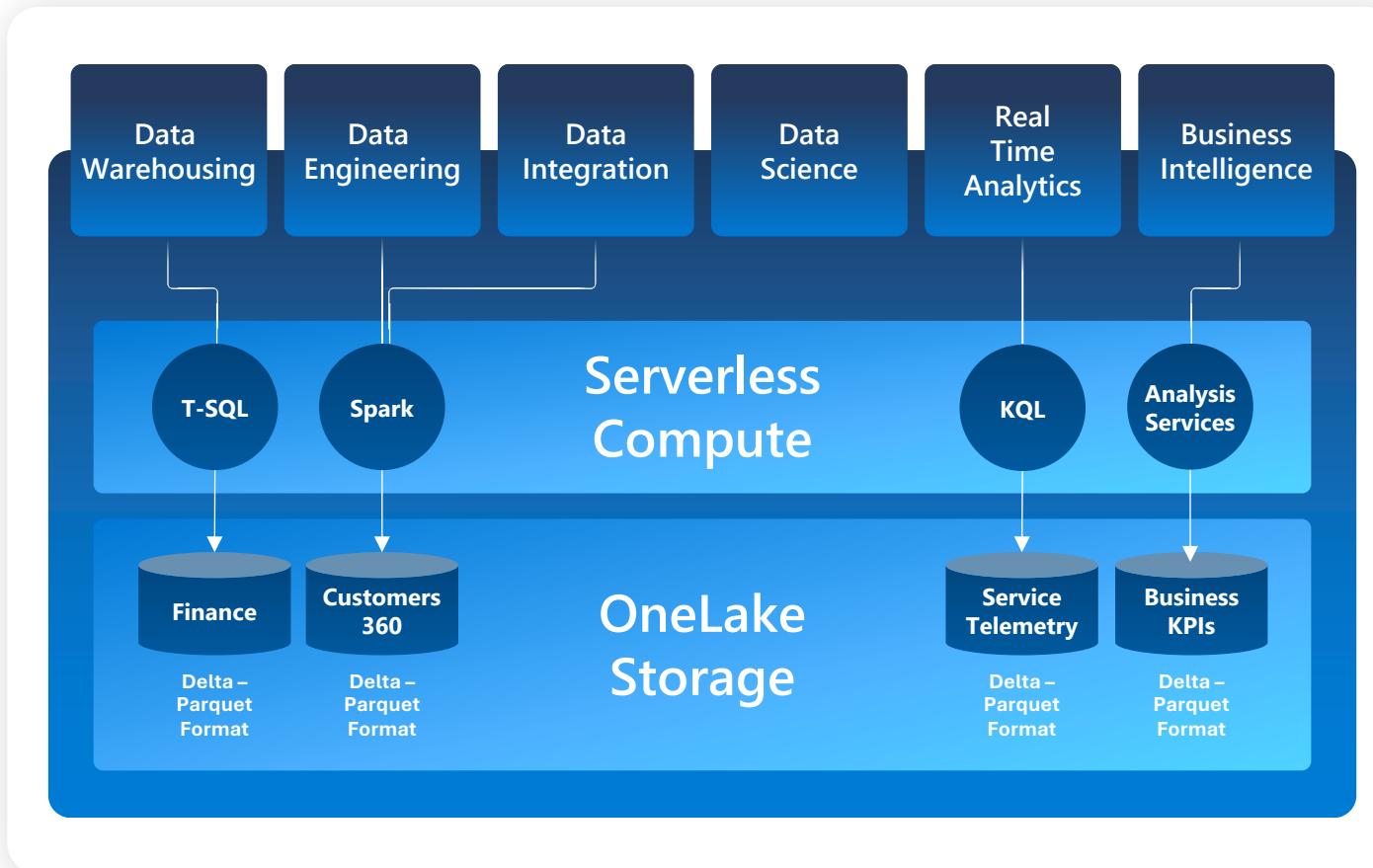
All the data is organized in an intuitive hierarchical namespace

The data in OneLake is automatically indexed for discovery, MIP labels, lineage, PII scans, sharing, governance and compliance



# One Copy for all computers

Real separation of compute and storage



All the compute engines store their data automatically in OneLake

The data is stored in a single common format

**Delta – Parquet**, an open standards format, is the storage format for all tabular data in Fabric

Once data is stored in the lake, it is directly accessible by all the engines without needing any import/export

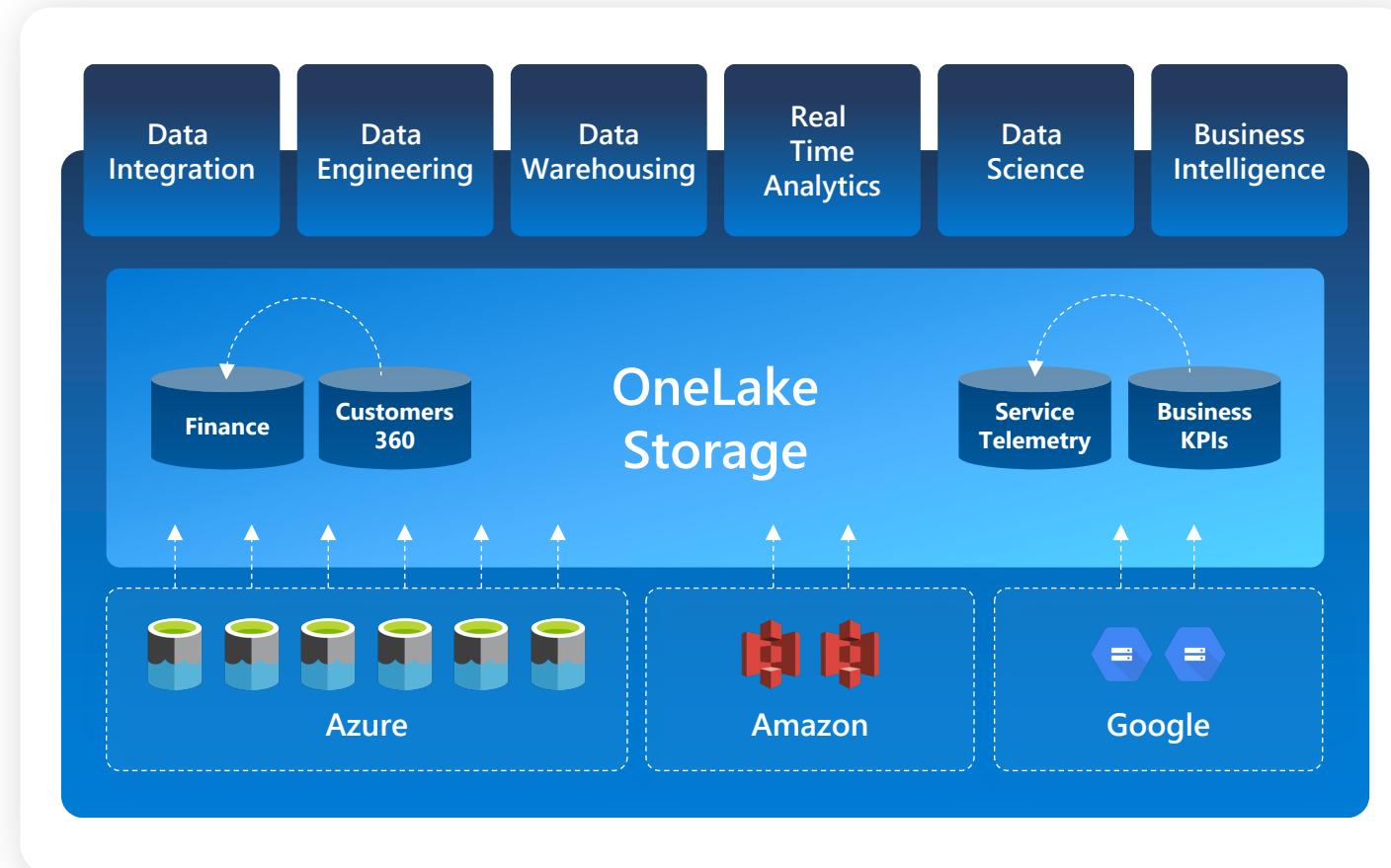
All the compute engines have been fully optimized to work with Delta Parquet as their native format

Shared universal security model is enforced across all the engines



# Taking One Copy to the next level

## Shortcuts



Sharing data in OneLake is as easy as sharing files in OneDrive, removing the needs for data duplication

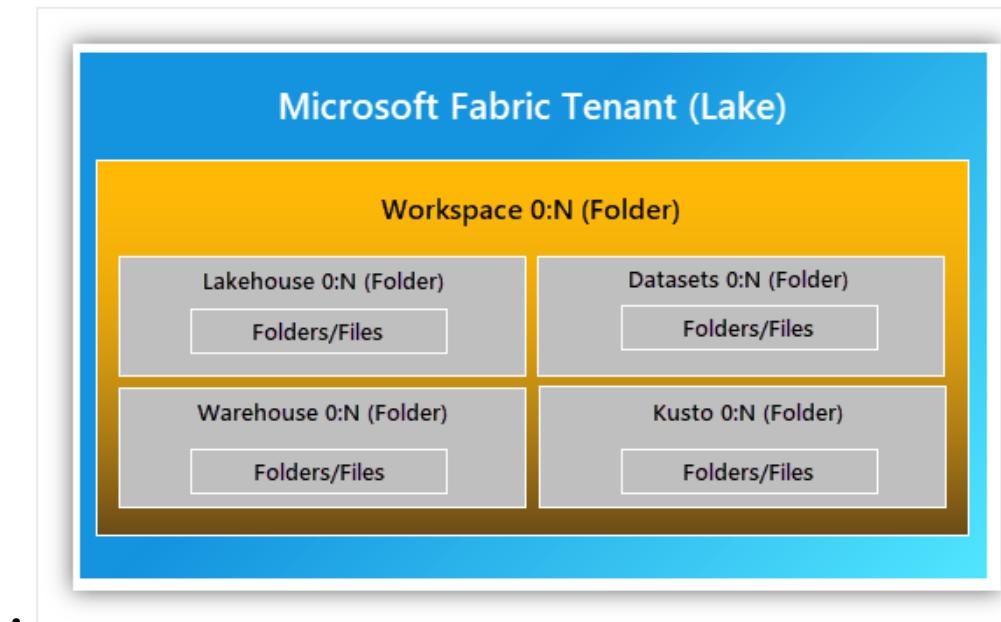
With **shortcuts**, data throughout OneLake can be composed together without any data movement

Shortcuts also allow instant linking of data already existing in Azure and in other clouds, without any data duplication and movement, making **OneLake the first multi-cloud data lake**

With support for industry standard APIs, OneLake data can be directly accessed by any application or service

# OneLake organisation

- is hierarchical and there is no need to up-front provisioning
  - one per tenant and single file system namespace across users and regions
  - divided into manageable containers
  - user can create n-number of workspaces within tenant (WS can be understood as folder)
  - every developer / business unit can create WS and ingest, process and analyse data, automatically the compute will be prewired to this WS
- Use Shortcut feature to access data in Azure Data Lake storage without migration



# Data Engineering - Data storage



Working with both files and tables

A screenshot of the Microsoft Fabric interface comparing two views: "Lake view" on the left and "Table view" on the right. Both views include a search bar and filter dropdown. The "Lake view" shows a hierarchical list under "Tables" and "Files". The "Table view" shows a flat list of tables and files.

Lake view	Table view
Search	Search
Table	Tables
> Deltatable_1	Deltatable_1
> Deltatable_2	Deltatable_2
> Account.parquet	Account
> Customer.csv	Customer
Files	
> TestFolder1	
> RandomFiles	
> CustomReview.csv	
> Username.parquet	
> Deltafolder	



# Data Engineering - Data storage

Browse your OneLake from Windows Explorer

The screenshot shows a Windows Explorer window displaying the contents of a OneLake storage location. The left pane shows a tree view of folders and files, while the right pane shows a detailed list view.

**Left Pane (File Explorer Tree View):**

- OneLake - Microsoft (Preview)
- > \_delta\_log
- > 311fc2bfba2647d9aeee9664b8ffe6ffe.parquet
- > c952fa477ac94582a10091d9ebc48767.parquet
- >
- >
- >
- >
- >
- > JobsGenie
- > Lakehouse Happy Path chmaneu
- > wwlakehouse.Lakehouse
  - > Files
  - > Tables
    - > \_mashup\_temporary
    - > Customers
      - > \_delta\_log
      - > silver\_dimension\_city

# Data Engineering - Data storage



Copy data from a wide range of services...

All   Azure   Database   File   Generic protocol   Services and apps    Search

Amazon RDS for SQL Server Database	Amazon Redshift Database	Amazon S3 File	Amazon S3 Compatible File
Apache Impala Database	Azure Blob Storage Azure	Azure Cosmos DB for NoSQL Azure	Azure Data Explorer (Kusto) Azure
Azure Data Lake Storage Gen1 Azure	Azure Data Lake Storage Gen2 Azure	Azure Database for PostgreSQL Azure	Azure SQL Database Azure
Azure SQL Database Managed Instance Azure	Azure Synapse Analytics Azure	Azure Table Storage Azure	Dataverse Services and apps
Dynamics CRM Services and apps	Google Cloud Storage File	HTTP Generic protocol	Hive Database
Microsoft 365 Services and apps	OData Generic protocol	PostgreSQL Database	REST Generic protocol, Services and apps
SQL server Database	SharePoint Online List Services and apps	Snowflake Services and apps	Spark Database

## **Exercise module 2 - Fabric**

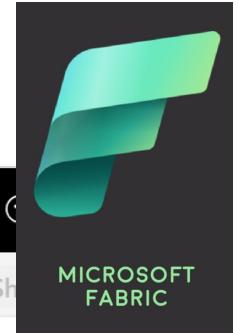
- Exercise 1: Creating a Lakehouse and Writing Data to a Delta Table
- Exercise 2: Creating a Spark Cluster for Distributed Computation
- Exercise 3: Partitioning Large Datasets in Storage for Optimized Compute Performance

## Module 3

- Data Movement, data pipelines and data transformation
- Explore the usage of notebooks with Python or PySpark
- Microsoft Fabric: the UI-based and code-based data engineering tasks and explore the use of shortcuts and data lakes
- Azure Machine Learning: designer and learn how to build pipelines using the Python SDK

# Data Engineering - Data delivery

Access lakehouse data from Notebooks and Spark Jobs



The screenshot shows the Microsoft Fabric Notebook interface. The top navigation bar includes 'Customer r...', 'Confidential\Micr...', 'Saved', a search bar, and various icons for notifications, settings, and help. The left sidebar has a 'Lakehouse explorer' section with a tree view of a 'wwlakehouse' workspace, including 'bronze', 'training-datasource', 'csv', 'full', and several dimension and fact tables. The main workspace shows a PySpark (Python) code cell with the following content:

```
1 import logging;
2
3 logging.warning("hello");
[1] ... hello
```

The status bar at the bottom indicates the notebook is 'Ready' and contains '1 of 1 cell'.

# Data Engineering - Data delivery

Expose Datasets ready for Power BI



The screenshot shows the Microsoft Fabric interface for a dataset named 'NubesGen - Monthly Report'. The top navigation bar includes 'NubesGen - M...', a shield icon, a search bar, and a user profile. The left sidebar has links for Home, Create, Browse, Data hub, Workspaces, My workspace, NubesGen - Monthly..., and More... A Power BI icon is at the bottom. The main content area displays dataset details: 'Details for NubesGen - Monthly Report', 'Add description', 'Location: My Workspace', 'Refreshed: 4/20/23, 6:35:59 AM', and 'Sensitivity: General'. Below this are two cards: 'Visualize this data' (with a pie chart icon) and 'Share this data' (with a people icon). A 'Create from scratch' button is in the 'Visualize' card, and a 'Share dataset' button is in the 'Share' card. At the bottom, there's a section for 'See what already exists' with a 'Filter by keyword' and 'Filter' button.

NubesGen - M... |

File Refresh Share Create a report Analyze in Excel Lineage Open data model ...

Home Create Browse Data hub Workspaces My workspace NubesGen - Monthly... More... Power BI

**Details for NubesGen - Monthly Report**

+ Add description

Location Refreshed Sensitivity  
My Workspace 4/20/23, 6:35:59 AM General

**Visualize this data**

Create an interactive report, or a table, to discover and share business insights. [Learn more](#)

+ Create from scratch

**Share this data**

Give people access to the dataset and set their permissions to work with it. [Learn more](#)

Share dataset

See what already exists

These items use the same data source as NubesGen - Monthly Report

# Data Engineering - Data delivery

Use Lakehouse data from Excel



wwlakehouse

Search

Home

New SQL query

New visual query

Home

Create

Workspaces

Lakehouse Happy Pat...

wwlakehou se

wwlakehou se

SimpleSpark Job

More...

Explorer

+ Warehouses

StoredProcedures

Tables

Customers

dimension\_city

dimension\_cu...

dimension\_d...

dimension\_e...

dimension\_st...

fact\_sale

silver\_dimens...

testdelta

Views

Customers

Run Save as view

```
1 SELECT TOP (100) [AccountCreationDate]
2 , [Email]
3 , [FirstName]
4 , [Id]
5 , [LastName]
6 , [Phone]
7 FROM [wwlakehouse].[dbo].[Customers]
```

Get the full current results in an Excel worksheet.

Messages Results Save as table Download Excel file Visualize results Search

	AccountCreationDate	Email	FirstName	Id	LastName	Phone
1	2023-01-08T21:11:57.8870000	Yasmine_Wunsch99@yahoo.com	Yasmine	4156	Wunsch	3228813783
2	2022-06-02T02:03:12.4370000	Yasmine55@gmail.com	Yasmine	6527	Roberts	3279177223
3	2022-03-31T04:30:51.0200000	Yasmine_Lynch76@gmail.com	Yasmine	7351	Lynch	8121975420
4	2022-02-02T07:51:31.9400000	Yasmine_Jast@hotmail.com	Yasmine	8454	Jast	5680932006
5	2022-11-23T13:48:46.7830000	Yasmine_Crooks78@gmail.com	Yasmine	8867	Crooks	4530626066

# Data Engineering - Data delivery

Access your Lakehouse data from any SQL Client



Search

SQL connection string

Copy this string and use it to connect externally to the item from Power BI desktop or client tools.

About

x6eps4xrq2xudenlfv6naeo3i4-



Sensitivity label

Endorsement

SQL endpoint

The screenshot shows a SQL query editor interface. On the left, there's a sidebar with navigation links: 'Lakehouse', 'Tables', 'dbo.Customers', 'Columns', 'Keys', 'Constraints', 'dbo.dimension\_city', 'dbo.dimension\_customer', 'dbo.dimension\_date', 'dbo.dimension\_employee', 'dbo.dimension\_stock\_item', 'dbo.fact\_sale', 'dbo.silver\_dimension\_city', and 'BIG DATA CLUSTERS'. The main area has tabs for 'Run', 'Disconnect', 'Change Connection' (set to 'wwilakehouse'), 'Estimated Plan', and 'Actual Plan' (disabled). A 'Copy' button is visible above the results table. The results table displays a list of customer records from the 'dbo.Customers' table, with columns: Id, FirstName, LastName, Email, Phone, and AccountCreationDate. The data includes rows for Yasmine Wunsch, Yasmine Roberts, Yasmine Lynch, Yasmine Jast, Yasmine Crooks, Yasmine Daniel, Yasmine Doyle, Yasmine Conn, Yasmine O'Hara, Yasmine Klein, and Yasmine Koelpin.

	Id	FirstName	LastName	Email	Phone	AccountCreationDate
1	4156	Yasmine	Wunsch	Yasmine_Wunsch99@yahoo.com	3228813783	2023-01
2	6527	Yasmine	Roberts	Yasmine55@gmail.com	3279177223	2022-06
3	7351	Yasmine	Lynch	Yasmine_Lynch76@gmail.com	8121975420	2022-03
4	8454	Yasmine	Jast	Yasmine_Jast@hotmail.com	5680932006	2022-02
5	8867	Yasmine	Crooks	Yasmine_Crooks78@gmail.com	4530626066	2022-11
6	9687	Yasmine	Daniel	Yasmine_Daniel15@hotmail.com	3505196520	2022-03
7	11182	Yasmine	Doyle	Yasmine.Doyle@yahoo.com	0265165530	2022-07
8	12003	Yasmine	Conn	Yasmine23@yahoo.com	3667236525	2022-06
9	12344	Yasmine	O'Hara	Yasmine78@gmail.com	7778866216	2022-02
10	15347	Yasmine	Klein	Yasmine59@yahoo.com	7183792198	2022-05
11	17023	Yasmine	Koelpin	Yasmine_Koelpin@hotmail.com	4855619223	2022-03

# Data Engineering - Data delivery

Organize your data through workspaces & shortcuts



Home      Lakehouse

Get data | New Power BI dataset | Open notebook

A SQL endpoint for SQL querying and a default dataset for reporting were created and will be updated with any tables added to the lakehouse. You can access the SQL endpoint using the dropdown.

Lakehouse explorer

- NubesgenLH
  - Tables
    - bronze\_logs
    - Customers
  - Files
    - bronze
    - stblobnubesgen001

Load data in your lakehouse

- New Dataflow Gen2
- New data pipeline
- Open notebook



# Data Factory

- enables fast data copy and movement to your DataLake and Warehouse
- helps you get the data, prepare the data and do the orchestration
- Scalable, governed via Purview, low-code, AI enabled ☺
- Two high-level features in Data factory:
  - data flow
  - data pipelines

# Data Factory



## Creating a Dataflow for data ingestion

Screenshot of the Power Query Editor interface showing the creation of a Dataflow for data ingestion.

The interface includes the following elements:

- Power Query | Dataflow 11** - Title bar.
- Home**, **Transform**, **Add column**, **View**, **Help** - Main menu tabs.
- Get data**, **Enter data**, **Options**, **Manage parameters**, **Refresh**, **Advanced editor**, **Add data destination**, **Properties**, **Query** - Data management tools.
- Choose columns**, **Remove columns**, **Keep rows**, **Remove rows**, **Filter rows**, **Sort**, **Split column by**, **Group**, **ABC** (Data type: Text), **Use first row as headers**, **Append queries**, **Merge queries**, **Replace values**, **Combine files**, **Map to entity**, **CDM**, **Export template** - Transformation and query management buttons.
- Queries [2]** - A list of loaded queries: **Customers** and **Orders**.
- CustomerID**, **CompanyName**, **ContactName**, **ContactTitle**, **Address**, **City**, **Region**, **PostalCode**, **Country**, **Phone**, **Fax**, **Orders**, **CustomerID**, **OrderID**, **ProductID**, **Quantity**, **UnitPrice**, **ShippedDate** - Columns listed in the main table area.
- Navigation 1** - Applied step in the Query settings pane.
- Properties**, **Name** (set to **Customers**), **Entity type** (set to **Custom**), **Applied steps** (list of steps), **Data destination** (set to **No data destination**) - Query settings pane.
- Completed (9.51 s)**, **Columns: 13**, **Rows: 91** - Status bar at the bottom.

CustomerID	CompanyName	ContactName	ContactTitle	Address	City	Region	PostalCode	Country	Phone	Fax	Orders	OrderID	ProductID	Quantity	UnitPrice	ShippedDate
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	null	1209	Germany	030-0074321	030-0076545	[Table]					
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.	null	05021	Mexico	(5) 555-4729	(5) 555-3745	[Table]					
ANTON	Antonio Moreno Taqueria	Antonio Moreno	Owner	Mataderos 2312	México D.F.	null	05023	Mexico	(5) 555-3932	null	[Table]					
AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	null	WA1 1DP	UK	(171) 555-7788	(171) 555-0750	[Table]					
BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Berguvsvägen 8	Luleå	null	95882	Sweden	0921-12 34 65	0921-12 34 67	[Table]					
BLAUS	Blauer See Delicatessen	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim	null	68306	Germany	0621-08460	0621-08924	[Table]					
BLONP	Blondesdal pâté et fils	Frédérique Citeaux	Marketing Manager	24, place Kléber	Strasbourg	null	67000	France	88.60.15.31	88.60.15.32	[Table]					
BOUD	Bólido Comidas preparadas	Martin Sommer	Owner	C/ Araquil, 67	Madrid	null	20023	Spain	(91) 555 22 82	(91) 555 91 99	[Table]					
BONAP	Bon app'	Laurence Lebihan	Owner	12, rue des Bouchers	Marseille	null	13008	France	91.24.45.40	91.24.45.41	[Table]					
BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager	23 Tsawassen Blvd.	Tsawassen	BC	T2B 8M4	Canada	(604) 555-4729	(604) 555-3745	[Table]					
BSBEV	B's Beverages	Victoria Ashworth	Sales Representative	Fauntleroy Circus	London	null	EC2 5NT	UK	(171) 555-1212	null	[Table]					
CACTU	Cactus Comidas para llevar	Patricia Simpson	Sales Agent	Cerrito 333	Buenos Aires	null	1010	Argentina	(1) 135-5555	(1) 135-8982	[Table]					
CENTC	Centro comercial Móctezuma	Francisco Chang	Marketing Manager	Sierras de Granada 9993	México D.F.	null	05022	Mexico	(5) 555-3392	(5) 555-7293	[Table]					
CHOPS	Chop-suey Chinese	Yang Wang	Owner	Hauptstr. 29	Bern	null	3012	Switzerland	0452-076545	null	[Table]					
COMM1	Comércio Mineiro	Pedro Alfonso	Sales Associate	Av. dos Lusíadas, 23	Sao Paulo	SP	05432-043	Brazil	(11) 555-7647	null	[Table]					
CONSH	Consolidated Holdings	Elizabeth Brown	Sales Representative	Berkeley Garden 12	Brewery	London	null	WX1 6LT	(171) 555-2282	(171) 555-9199	[Table]					
DRAOD	Drachenblut Delikatessen	Sven Ottieb	Order Administrator	Walsenweg 21	Aachen	null	52066	Germany	0241-039123	0241-059428	[Table]					
DUMON	Du monde entier	Janine Labrune	Owner	87, rue des Cinquante Otages	Nantes	null	44000	France	40.67.88.88	40.67.89.89	[Table]					
EASTC	Eastern Connection	Ann Devon	Sales Agent	35 King George	London	null	WX3 6FW	UK	(171) 555-0297	(171) 555-3373	[Table]					
ERNSH	Ernst Handel	Roland Mendel	Sales Manager	Kirchgasse 6	Graz	null	8010	Austria	7675-3425	7675-3426	[Table]					
FAMIA	Familia Arquibaldo	Aria Cruz	Marketing Assistant	Rua Orós, 92	Sao Paulo	SP	05442-030	Brazil	(11) 555-9857	null	[Table]					
FISSA	FISSA Fabrica Inter. Salchichas S.A.	Diego Roel	Accounting Manager	C/ Moratalaz, 86	Madrid	null	28034	Spain	(91) 555 94 44	(91) 555 55 93	[Table]					
FOLIG	Folies gourmandes	Martine Rancé	Assistant Sales Agent	184, chaussée de Tournai	Lille	null	59000	France	20.16.10.16	20.16.10.17	[Table]					
FOLKU	Folk och fä HB	Maria Larsson	Owner	Älvängatan 24	Bräcke	null	5-844 67	Sweden	0995-34 67 21	null	[Table]					
FRANK	Frankenversand	Peter franken	Marketing Manager	Berliner Platz 43	München	null	80085	Germany	089-0877310	089-0877451	[Table]					
FRANR	France restauration	Carine Schmitt	Marketing Manager	54, rue Royale	Nantes	null	44000	France	40.32.21.21	40.32.21.20	[Table]					
FRANS	Franchi S.p.A.	Paolo Accorti	Sales Representative	Via Monte Bianco 34	Torino	null	10100	Italy	011-4988260	011-4988261	[Table]					
FURIB	Furia Bacalhau & Frutos do Mar	Lino Rodriguez	Sales Manager	Jardim das rosas n. 32	lisboa	null	1675	Portugal	(1) 354-2334	(1) 354-2335	[Table]					
GALED	Galería del gastrónomo	Eduardo Saavedra	Marketing Manager	Rambla de Cataluña, 23	Barcelona	null	08022	Spain	(93) 203 4560	(93) 203 4561	[Table]					
GODOS	Godos Cocina Típica	José Pedro Freyre	Sales Manager	C/ Romero, 33	Sevilla	null	41101	Spain	(95) 555 82 82	null	[Table]					
GOURL	Gourmet Lanchonetes	André Fonseca	Sales Associate	Av. Brasil, 442	Campinas	SP	04876-786	Brazil	(11) 555-9482	null	[Table]					
GREAL	Great Lakes Food Market	Howard Snyder	Marketing Manager	2732 Baker Blvd.	Eugene	OR	97403	USA	(503) 555-7555	null	[Table]					
GROSIR	GROSELLA-Restaurante	Manuel Pereira	Owner	54 Av. Los Palos Grandes	Caracas	DF	1001	Venezuela	(2) 283-2951	(2) 283-3397	[Table]					
HANAR	Hanari Carnes	Mario Pontes	Accounting Manager	Rua do Pago, 67	Rio de Janeiro	RJ	05454-076	Brazil	(21) 555-0091	(21) 555-8765	[Table]					
HILAA	HILARION-Abastos	Carlos Hernández	Sales Representative	Carrera 22 con Avenida Carlos Soublette #8-35	San Cristóbal	Táchira	5022	Venezuela	(5) 555-1340	(5) 555-1948	[Table]					

# Data Factory

## Transform data with Power Query



Microsoft Data Factory - Power Query

Queries [3]

Customers (2 steps) → Ranked Customers (3 steps)

Orders (3 steps) → Ranked Customers (3 steps)

Ranked Customers (3 steps)

Source: Removed other c... Expanded Orders... Added rank col...

Table.AddRankColumn("Expanded Orders (2)", "Rank", {"Count", order.Descending})

	CustomerID	CompanyName	Count	Rank
1	SAVEA	Save-a-lot Markets	31	1
2	ERNSH	Ernst Handel	30	2
3	QUICK	QUICK-Stop	28	3
4	HUNGO	Hungry Owl All-Night Grocers	19	4
5	FOLKO	Folk och f� H�	19	4
6	RATTC	Rattlesnake Canyon Grocery	18	6
7	HILAA	HILARION-Abastos	18	6
8	BERGS	Berglunds snabblop	18	6
9	BONAP	Bon app'	17	9
10	FRANK	Frankenversand	15	10

Completed (3.03 s) Columns: 4 Rows: 91 Column profiling based on top 1,000 rows

Step Publish

# Data Factory

## Aggregation and Merge Operations with Power Query



Screenshot of the Microsoft Power Query Editor interface, showing a dataflow named "Dataflow 11".

The interface includes a ribbon bar with tabs: Home, Transform, Add column, View, and Help. The "Add column" tab is currently selected.

The main workspace displays a dataflow diagram with three queries:

- Customers**: A query with 2 steps.
- Orders**: A query with 3 steps.
- Merge**: A merge operation combining the two queries. It shows the "Source" query, the "Removed other c..." step, the "Expanded Orders..." step, and the "Added rank colu..." step.

The bottom pane shows the results of the "Merge" step, specifically the "Table.AddRankColumn" step. The resulting table has four columns: CustomerID, CompanyName, Count, and Rank.

	CustomerID	CompanyName	Count	Rank
1	SAVEA	Save-a-lot Markets	31	1
2	ERNSH	Ernst Handel	30	2
3	QUICK	QUICK-Stop	28	3
4	HUNGO	Hungry Owl All-Night Grocers	19	4
5	FOLK0	Folk och fä H&B	19	4
6	RATTG	Rattlesnake Canyon Grocery	18	6
7	HILAA	HILARION-Abastos	18	6
8	BERGS	Berglunds snabbköp	18	6
9	BONAP	Bon app'	17	9
10	FRANK	Frankenversand	15	10

The right side of the screen contains the "Query settings" pane, which includes sections for Properties (Name: Merge), Entity type (Custom), and Applied steps (listing the Source, Removed other c..., Expanded ..., and Added rank... steps).

# Data Factory

Building workflows with pipelines



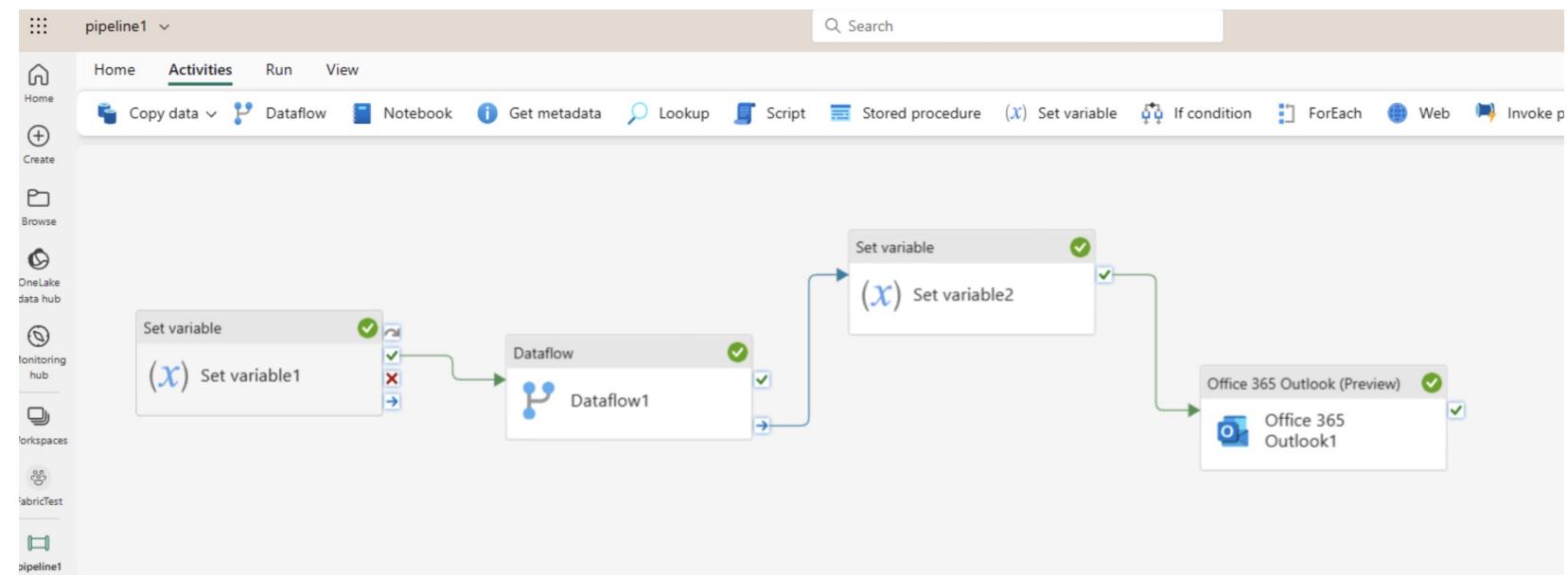
Datasets can be from any source

Activity produces dataset

Dataset consumes activity

Pipelines are a sequence of activities

Activities, datasets, and dataflows can be controlled within the pipeline



## Exercise module 3 - Fabric

- Exercise 1: Data Movement Using Shortcuts from One Lakehouse to Another
- Exercise 2: Creating a Data Pipeline for Extract-Transform-Load (ETL) in Microsoft Fabric
- Exercise 3: Data Transformation and Aggregation in a Data Lake Using PySpark

## Module 4

- Focusing on data analysis using notebooks
- Data analysis and exploration with Fabric and Azure Machine Learning
- build prepare and build a machine learning model and
- run experiments

# EDA

- Exploring data with Fabric and AML
- Data Analysis
- Model creation
- Model training

## **Exercise module 4 - Fabric**

- Exercise 1: Data Analysis and Exploration Using PySpark
- Exercise 2: Data Preparation for Machine Learning Model
- Exercise 3: Build and Train a Machine Learning Model and Run Experiments

## Module 5

- Test the predictions and perform inference.
- Fabric: job scheduling and run jobs using MLFlow to model retrain
- Azure Machine Learning: job scheduling and run jobs using MLFlow to model retrain
- expose endpoints for consumption.

# MLFlow

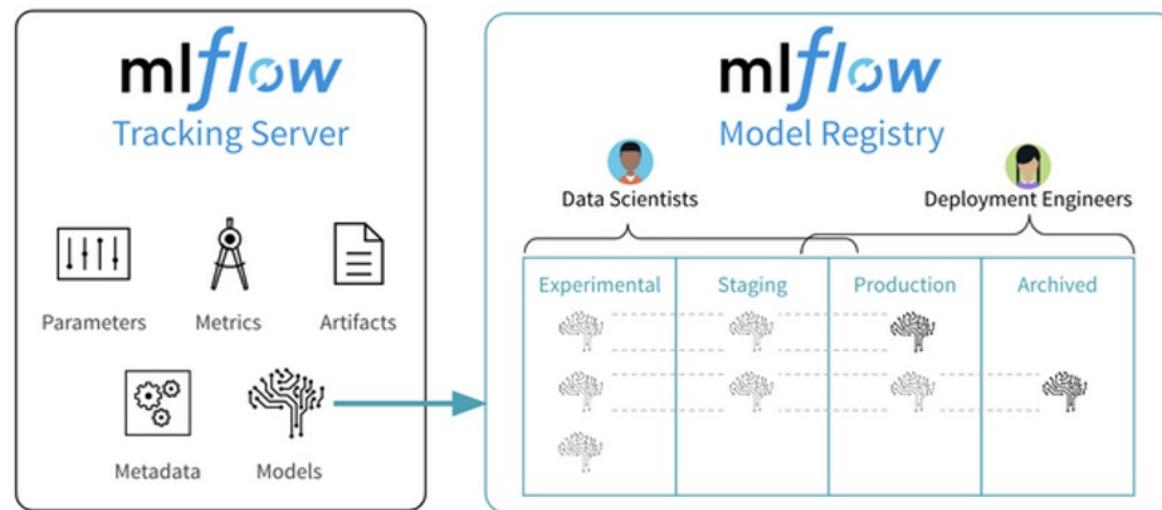
Design, integrate and reproduce your Machine learning models, experiments, artifacts and solutions. Keep your code sane, reproducible and accessible to data scientist, machine learning engineers, data analysts and other departments.

Will help you with:

- Keep track of your experiments -> **ML Flow Tracking**
- Standardize your way for storing models, packages -> **ML Registry**

# Tracking

- Tracking code development in GIT is easy.
- Tracking intensive data science experiment that includes model is almost impossible using GIT only.
- We need to go back and ask what model/settings achieved the best results.



# Experiments

**Each ML experiment contains:**

- **Source:** Name of the notebook.
- **Version:** Notebook revision.
- **Start & end time:** Start and end time of the run.
- **Parameters:** Key-value model parameters.
- **Tags:** Key-value run metadata.
- **Metrics:** Key-value model evaluation metrics.
- **Artifacts:** Output files in any format.

Name	Latest Version	Staging	Production	Last Modified
Item_Recommender	Version 5	Version 5	Version 4	2019-10-11 15:30:02
Airline_Delay_Scikit	Version 3	—	Version 1	2019-10-11 12:41:43
Airline_Delay_SparkML	Version 5	Version 5	Version 3	2019-10-11 12:45:15
Transaction_Fraud_Classifier	Version 1	—	—	2019-10-11 15:18:05
Icon_GAN	Version 1	—	—	2019-10-12 08:20:12
Power_Forecasting_Model	Version 1	—	Version 1	2019-10-07 15:38:27
Product_Image_Classifier	Version 6	—	Version 5	2019-10-12 00:38:56
Comment_Summarizer	Version 3	Version 2	Version 3	2019-10-12 00:39:40
Movie_Recommender	Version 5	Version 5	Version 3	2019-10-10 14:07:07
Translation_Alpha	—	—	—	2019-10-11 16:45:01

## Exercise module 5 - Fabric

- Exercise 1: Test Model Predictions and Perform Inference Using PySpark
- Exercise 2: Job Scheduling and Running Jobs for Model Inference
- Exercise 3: Model Retraining Using MLflow and Job Scheduling

## Module 6

- Creating an end-to-end integrated solution.
- Various options of visualizing the results of our analysis and model predictions.

# E2E in steps

Step 1: Data Preparation and Reading Data from the Lakehouse and Data Lake

Step 2: Exploratory Data Analysis (EDA)

Step 3: Feature Engineering

Step 4: Build and Train the Model

Step 5: Model Evaluation

Step 6: Model Inference and API Implementation

## **Exercise module 6 - Fabric**

- Exercise 1: Go through E2E solution