ONEWAY

SOLUTION

One Way Solution

Batch-ETL

Data Engineering – [Day 2]

LUAN MORENO

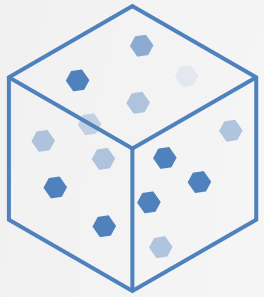
CEO & CDO

Data Engineer & Data Platform MVP


Everyone has talent. What is rare is the courage to follow the talent to the dark place where it leads.
Erica Jong
quotefancy

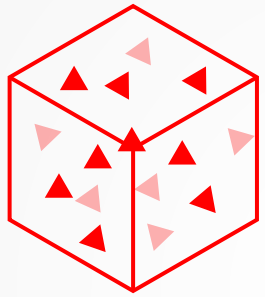
RDBMS

SQL Server | Oracle | PostgreSQL | MySQL



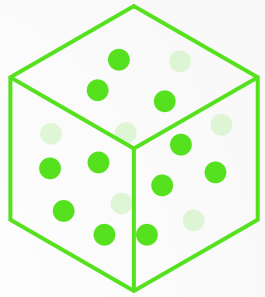
NoSQL

MongoDB | Cassandra | Redis



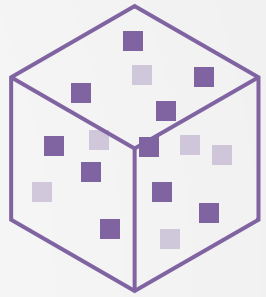
Web

Web App | Mobile App

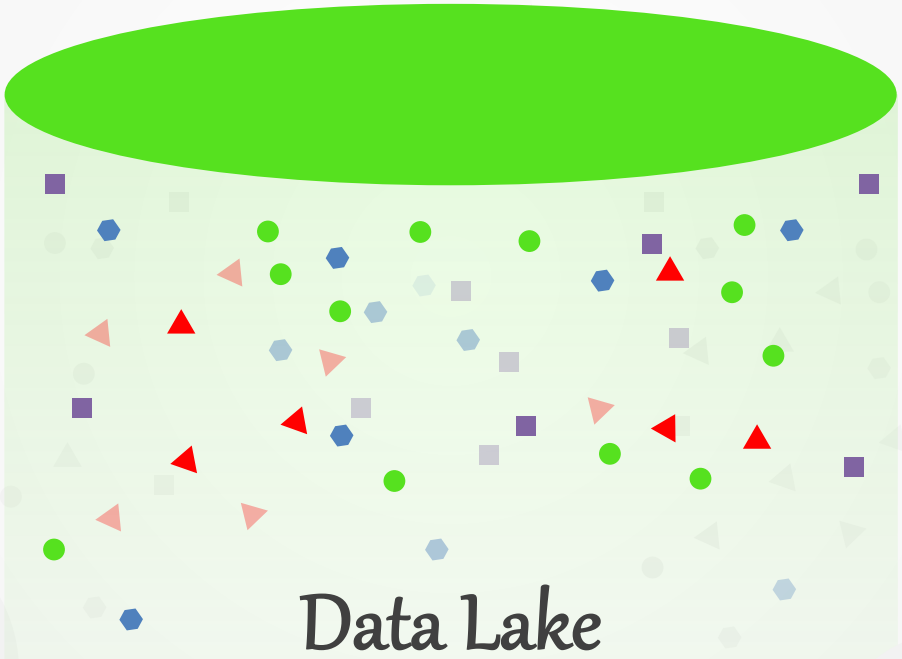


Files

CSV | XML | JSON

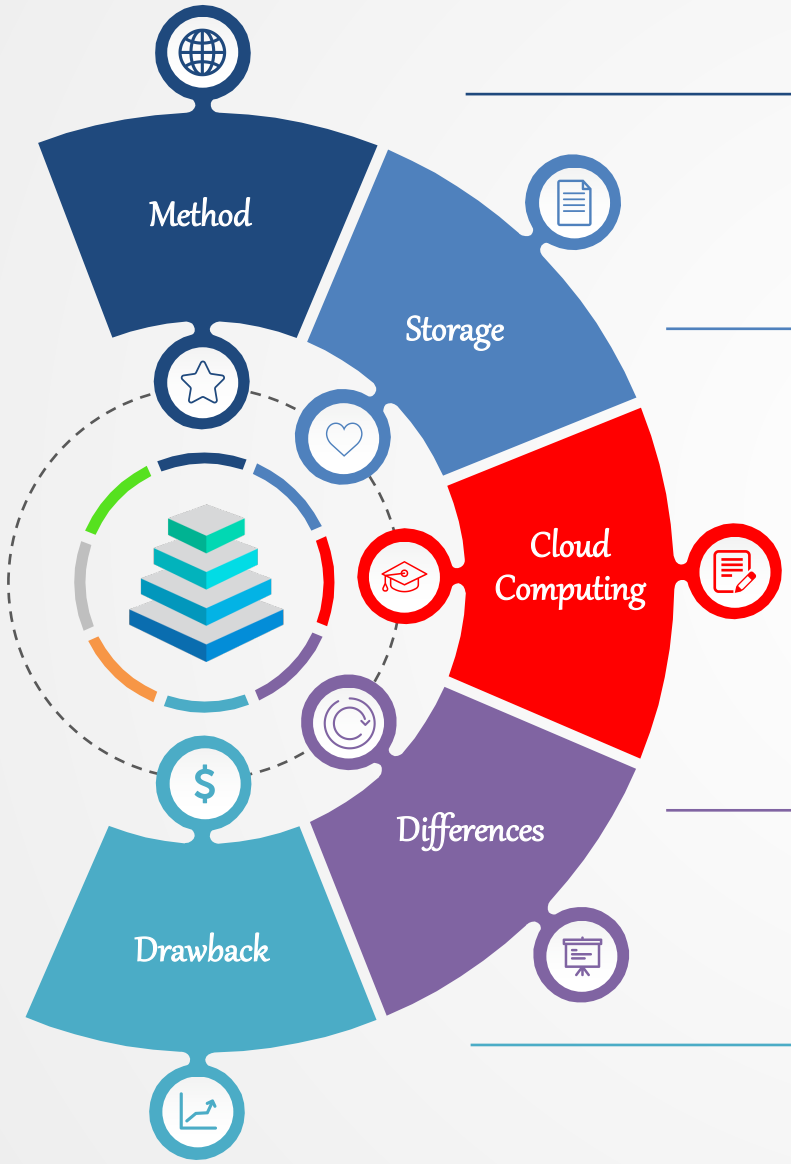


Process (ETL & ELT)



repository of raw data

Data Lake [Concepts]



Repository of Raw Data [Data Democratization]
Data Lake – James Dixon in 2010
Democratization of Data – [Unsiloed Data]

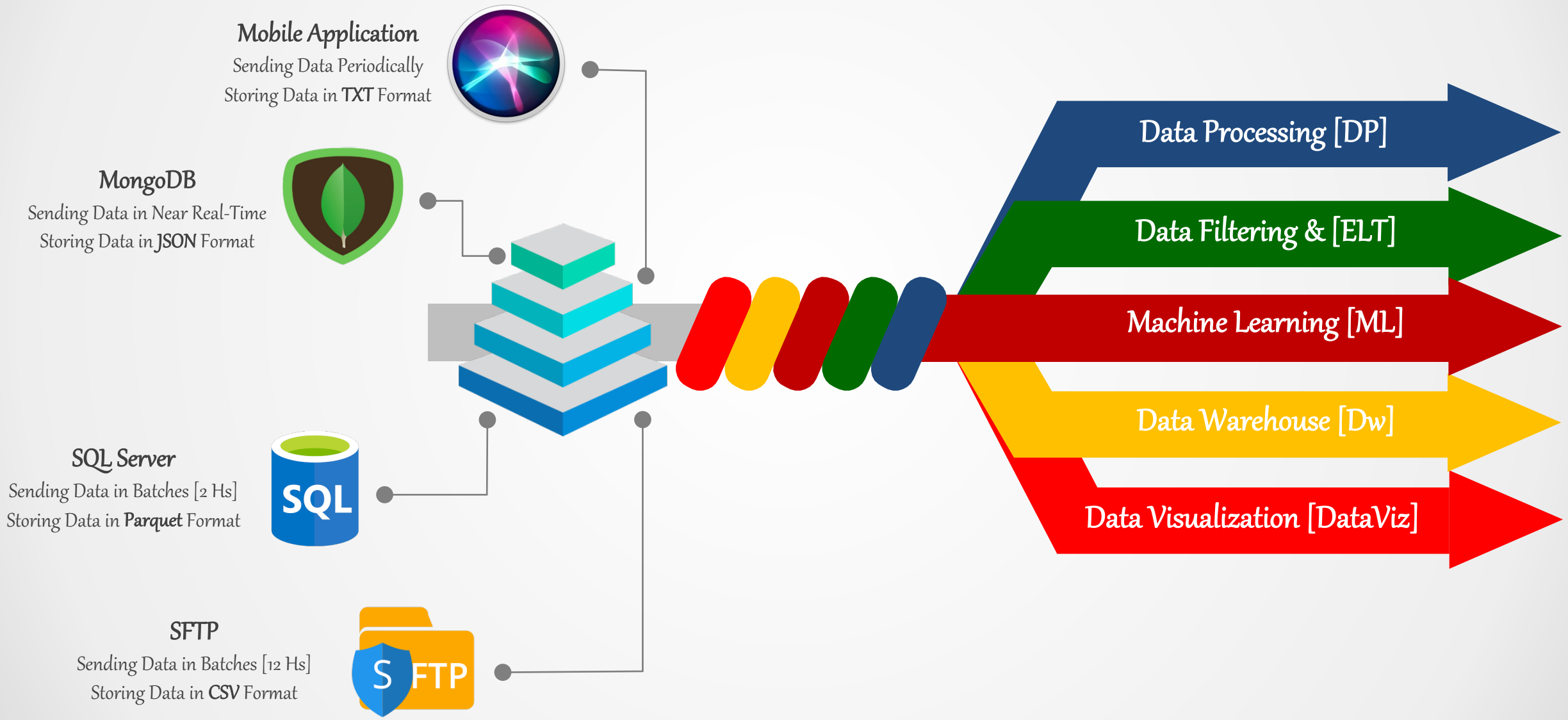
Structured – Relational Databases
Semi-Unstructured – CSV, Logs, XML & JSON
Unstructured – Emails, Documents, Binaries, Audio & Video

Azure – Azure Blob Storage & Azure Data Lake Storage Gen2
AWS – Amazon S3 & AWS Lake Formation
GCP – Google Cloud Storage

Data Mart – Subset of Decision Support for Departments
Data Warehouse – Decision Making Support for Enterprise-Level
Data Lake – Raw Data [Source of Truth]

Data Swamp – Highly Disorganized Data
Data Governance – Management & Control of Data
Data Quality – Accuracy and Data Veracity
Data Security – Protecting Digital Data

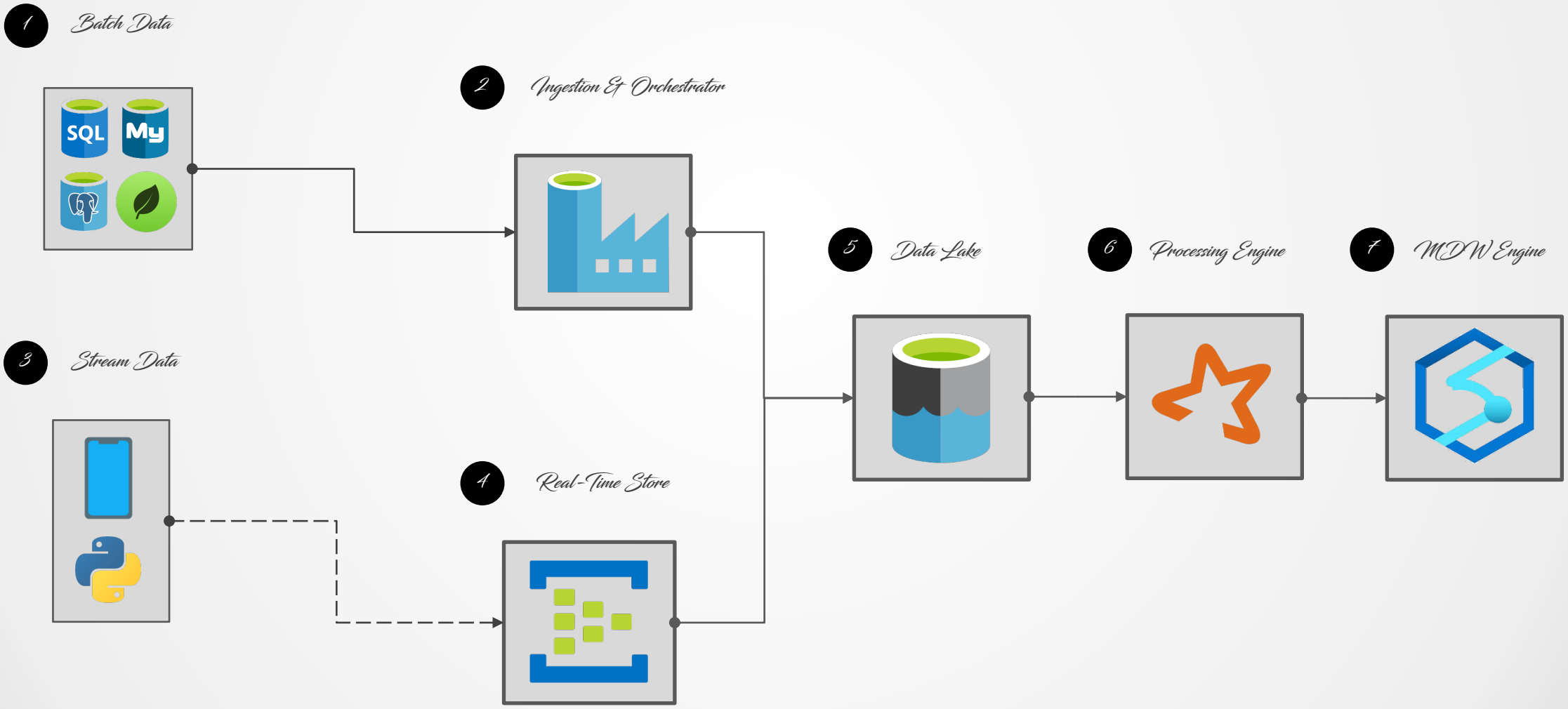
Data Lake [Use-Cases]



Data Acquisition & Store

acquisition and proper store of the data coming from different data sources. process configured for batch processing bringing full and incremental loads and stream processing storing pieces of files into the data lake to perform analytics at scale using synapse analytics to build a modern data warehouse [mdw]

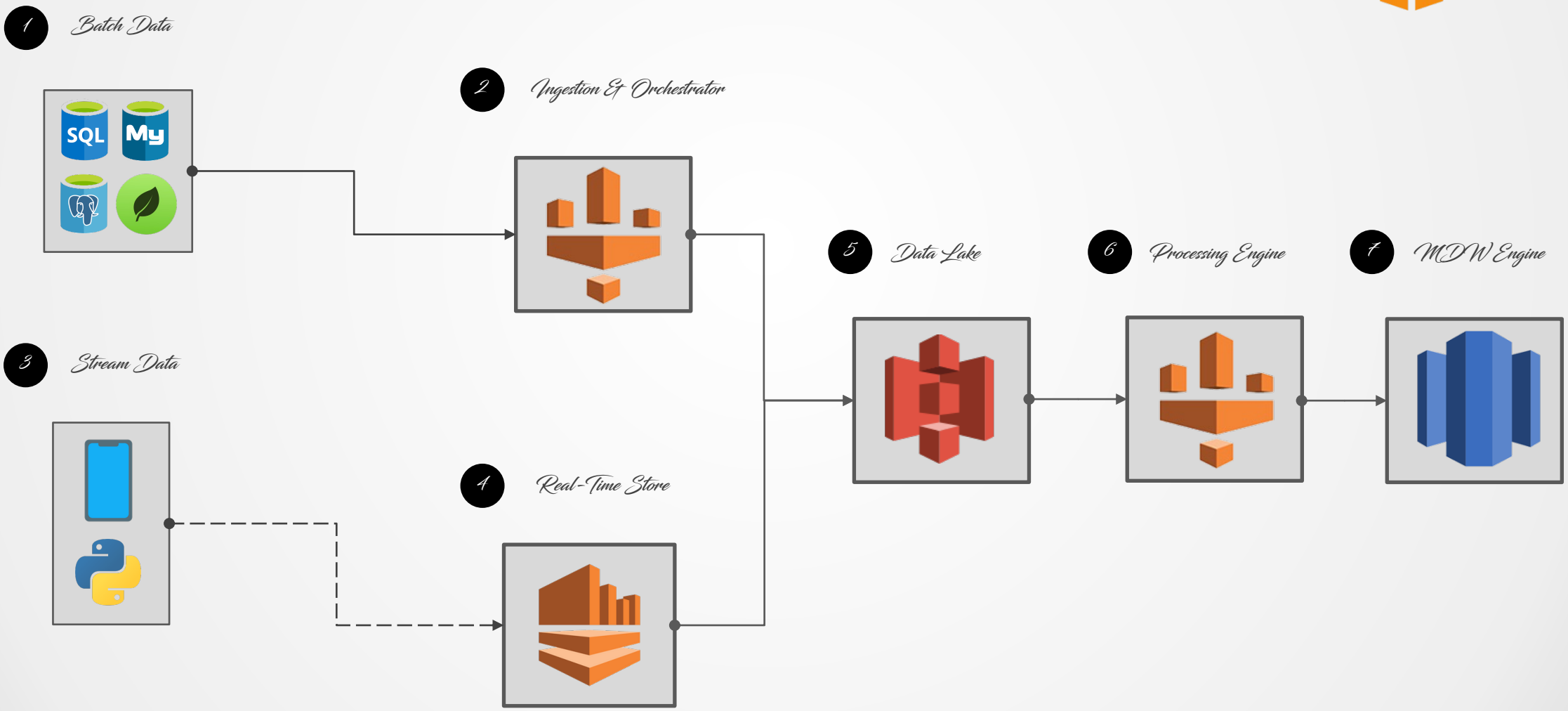
Lambda Architecture



Data Acquisition & Store

acquisition and proper store of the data coming from different data sources. process configured for batch processing bringing full and incremental loads and stream processing storing pieces of files into the data lake to perform analytics at scale using synapse analytics to build a modern data warehouse [mdw]

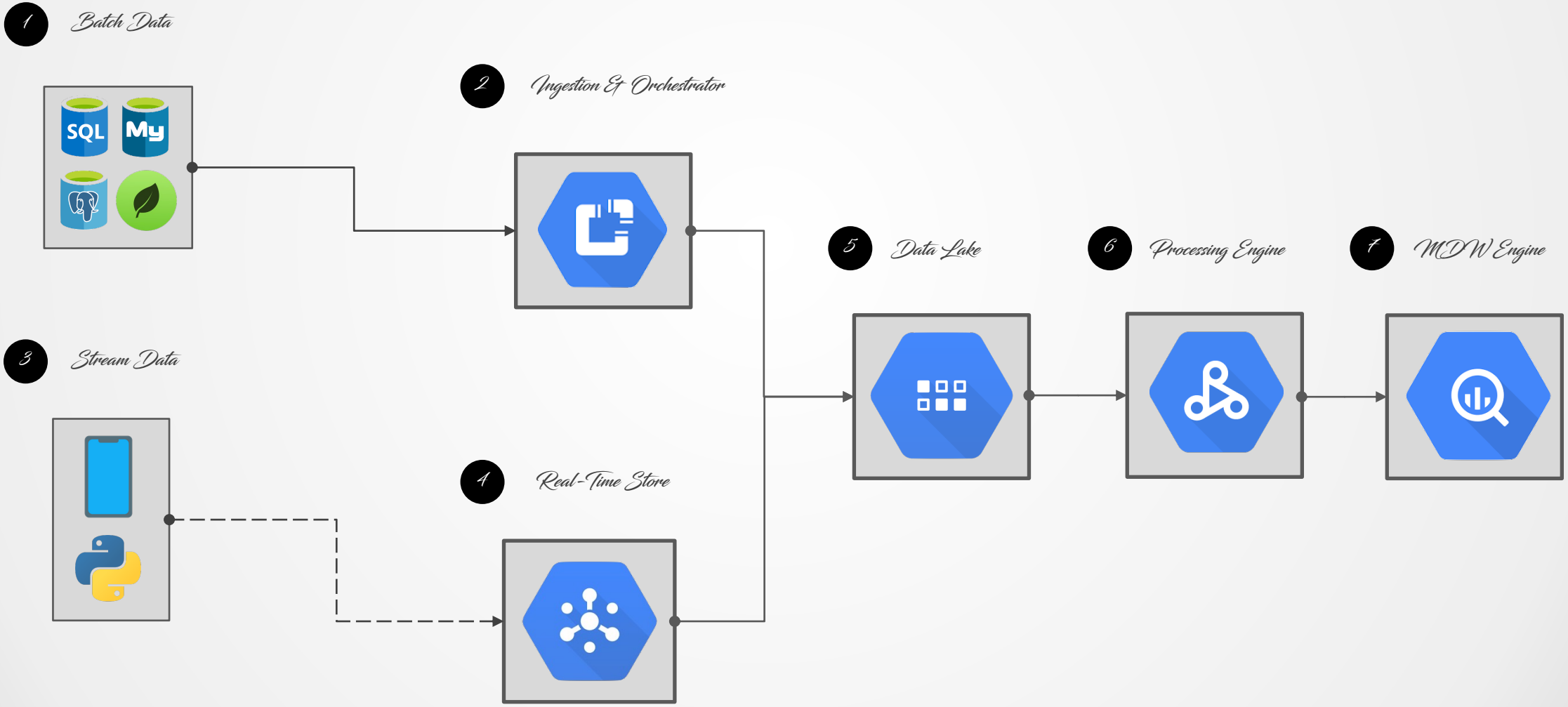
Lambda Architecture



Data Acquisition & Store

acquisition and proper store of the data coming from different data sources. process configured for batch processing bringing full and incremental loads and stream processing storing pieces of files into the data lake to perform analytics at scale using synapse analytics to build a modern data warehouse [mdw]

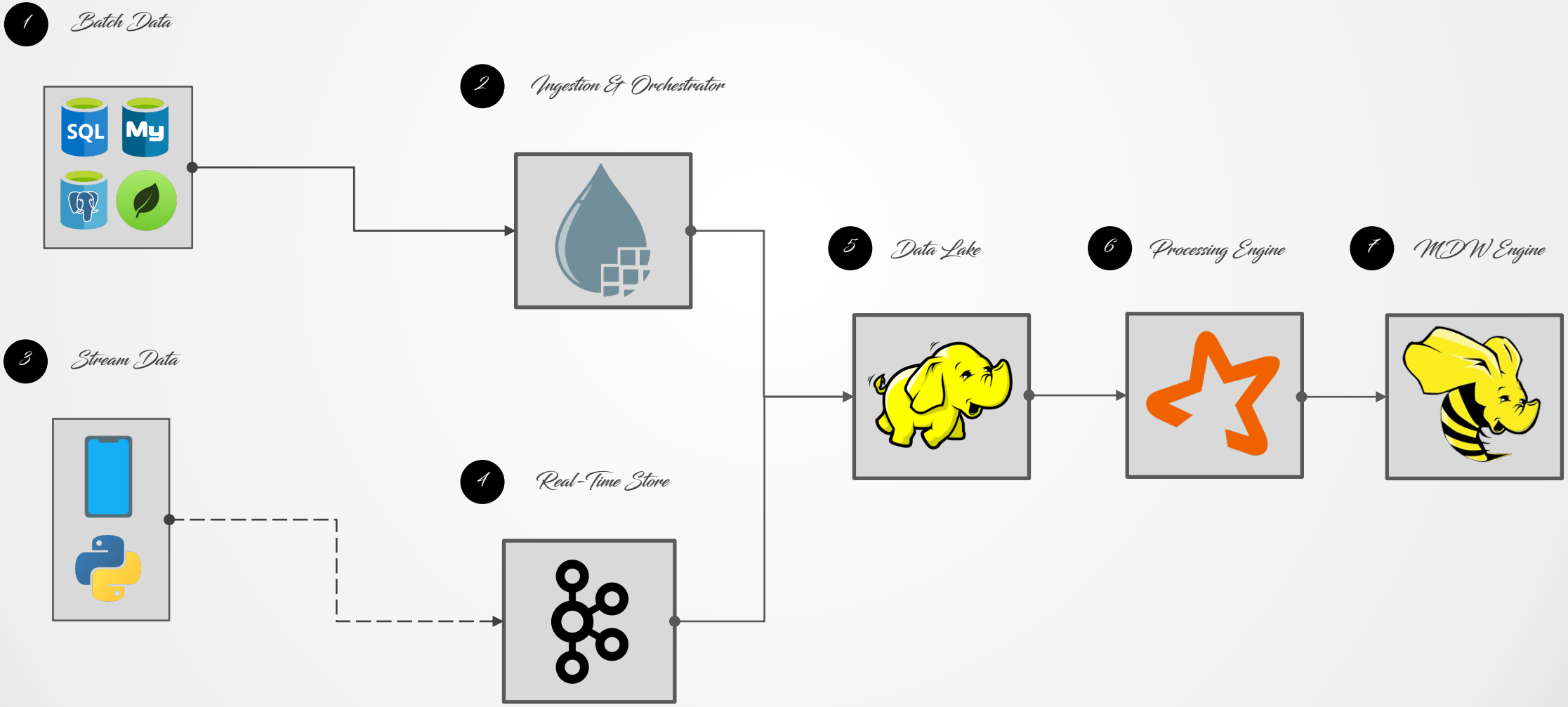
Lambda Architecture



Data Acquisition & Store

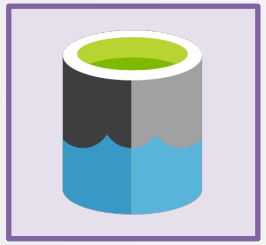
acquisition and proper store of the data coming from different data sources. process configured for batch processing bringing full and incremental loads and stream processing storing pieces of files into the data lake to perform analytics at scale using synapse analytics to build a modern data warehouse [mdw]

Lambda Architecture



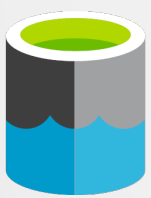
Data Organization Best Practices for Data Lake

patterns, best practices and recommendation in how to properly organize the data lake to avoid data swamp scenarios and to speed up data processing getting the best of the object-store layer



Object Storages

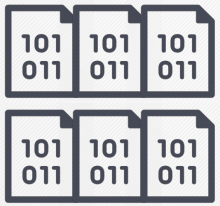
Storages Optimized for Big Data Use-Cases
Scalability & Reliability




Raw Ingestion

Landing Zone

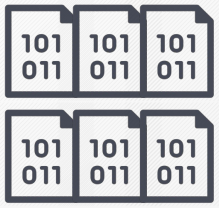
Raw Data without Treatment
Schemaless Environment
Different File Formats



Transformations

Processing Zone

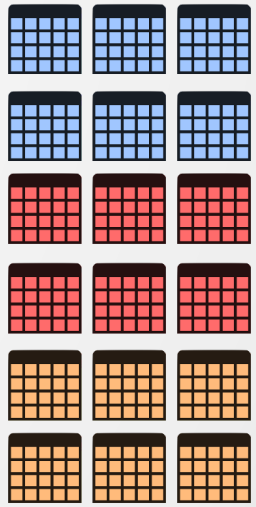
Files in Queue for Processing
Converted & Reconciled
Big Data File Format
Processing Engines Area

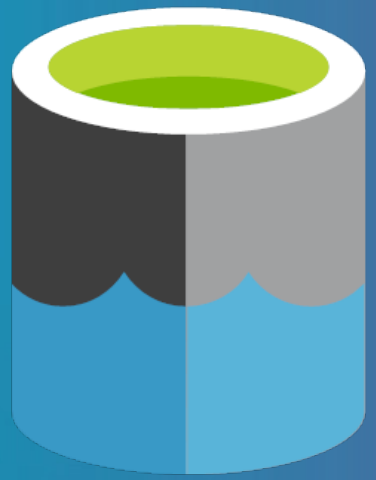
Business-Level

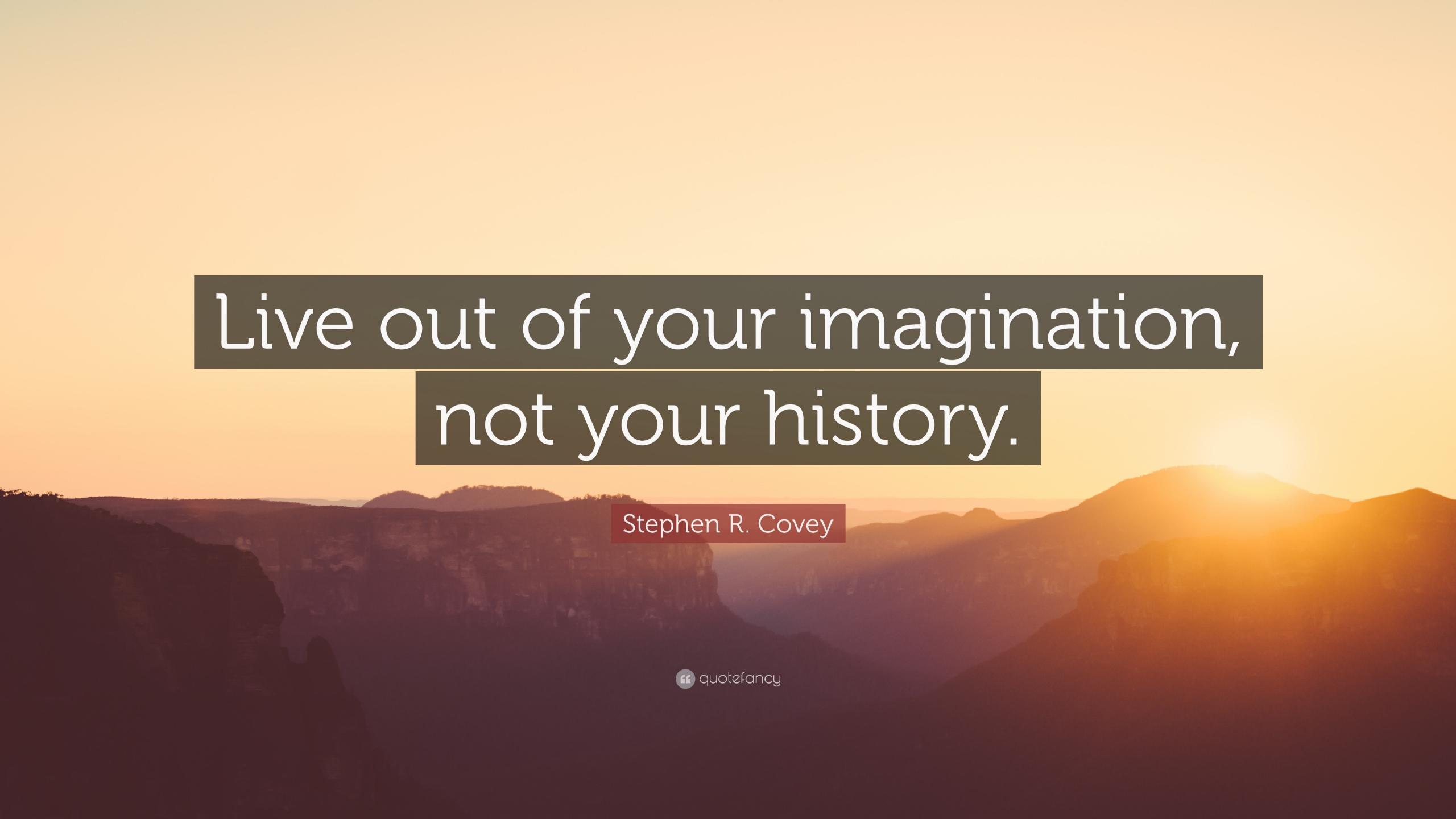
Curated Zone

Sanitized Location
Ready for Analytics
Used for Downstream Resources
Standard DataSets



Best Practices Organizing a Data Lake Repository

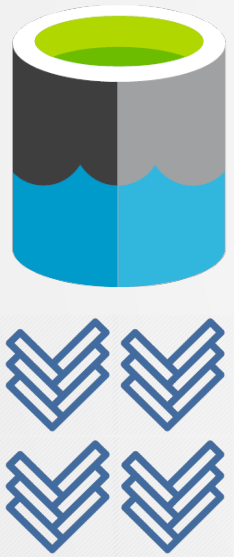



Live out of your imagination, not your history.
Stephen R. Covey
quotefancy

The Spark Lifecycle

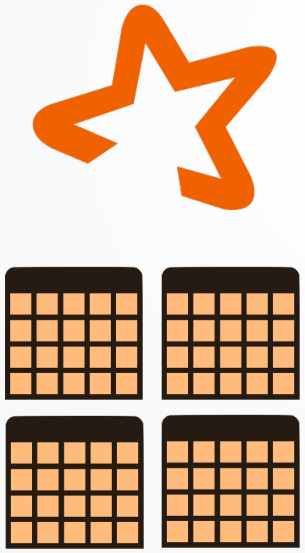
Data Lake

Repository of Raw Data
Without Schema Enforcement



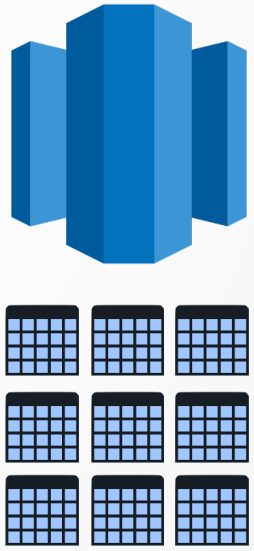
Apache Spark

Distributed Cluster-Computing Framework
Optimized for Memory Computation



Data Warehouse

Analytics Platform for Enterprises
Scalability – Horizontally & Vertically



Raw Ingestion

Transformations

Business-Level

Storage Solutions for Spark

Next Evolution of Storage Solutions

Scalability & Performance

the storage solution should be able to scale to the volume of data and provide the read & write throughput and latency that the workload requires.

Support for Diverse Data Formats

the storage solution should be able to store unstructured data (e.g., text files like raw logs), semi-structured data (e.g., JSON data), and structured data (e.g., tabular data).

Openness

supporting a wide range of workloads often requires the data to be stored in open data formats. standard apis allow the data to be accessed from a variety of tools and engines. this allows the business to use the most optimal tools for each type of workload and make the best business decisions.

Transaction Support

complex workloads are often reading and writing data concurrently, so support for acid transactions is essential to ensure the quality of the end results.

Support for Diverse Workloads

the storage solution should be able to support a diverse range of business

- SQL Workloads ~ Traditional BI Analytics
- Batch Workloads ~ Traditional ETL Jobs Processing Unstructured Data
- Streaming Workloads ~ Real-Time Monitoring & Alerting
- ML & AI Workloads ~ Recommendations and Churn Predictions

Storage Solutions for Spark ~ [Database]

Information

- Online Transaction Processing [OLTP]
- Online Analytical Processing [OLAP]

Limitations

- Growth in Data Sizes
- Growth in Diversity Analytics
- Expensive for Scale-Out
- Poor Support for Non-SQL Based Analytics

Storage Solutions for Spark ~ [Data Lakes]

Information

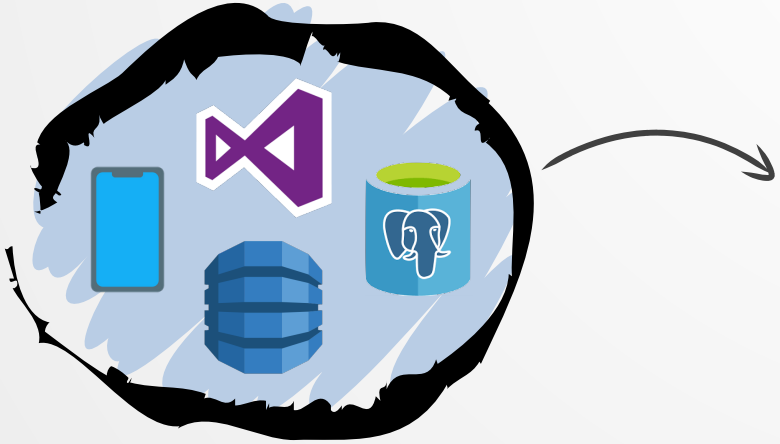
- Support for Diverse Use-Cases
- Support for Diverse File Formats
- Support for Diverse Filesystems

Limitations

- Atomicity & Isolation ~ Corrupted Data
- Consistency – Inconsistency View of Data

External Data

Different Data Sources
Periodically Storing Data

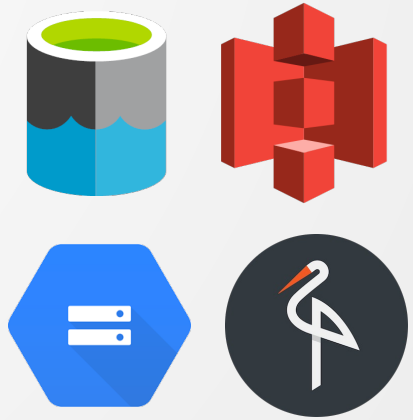


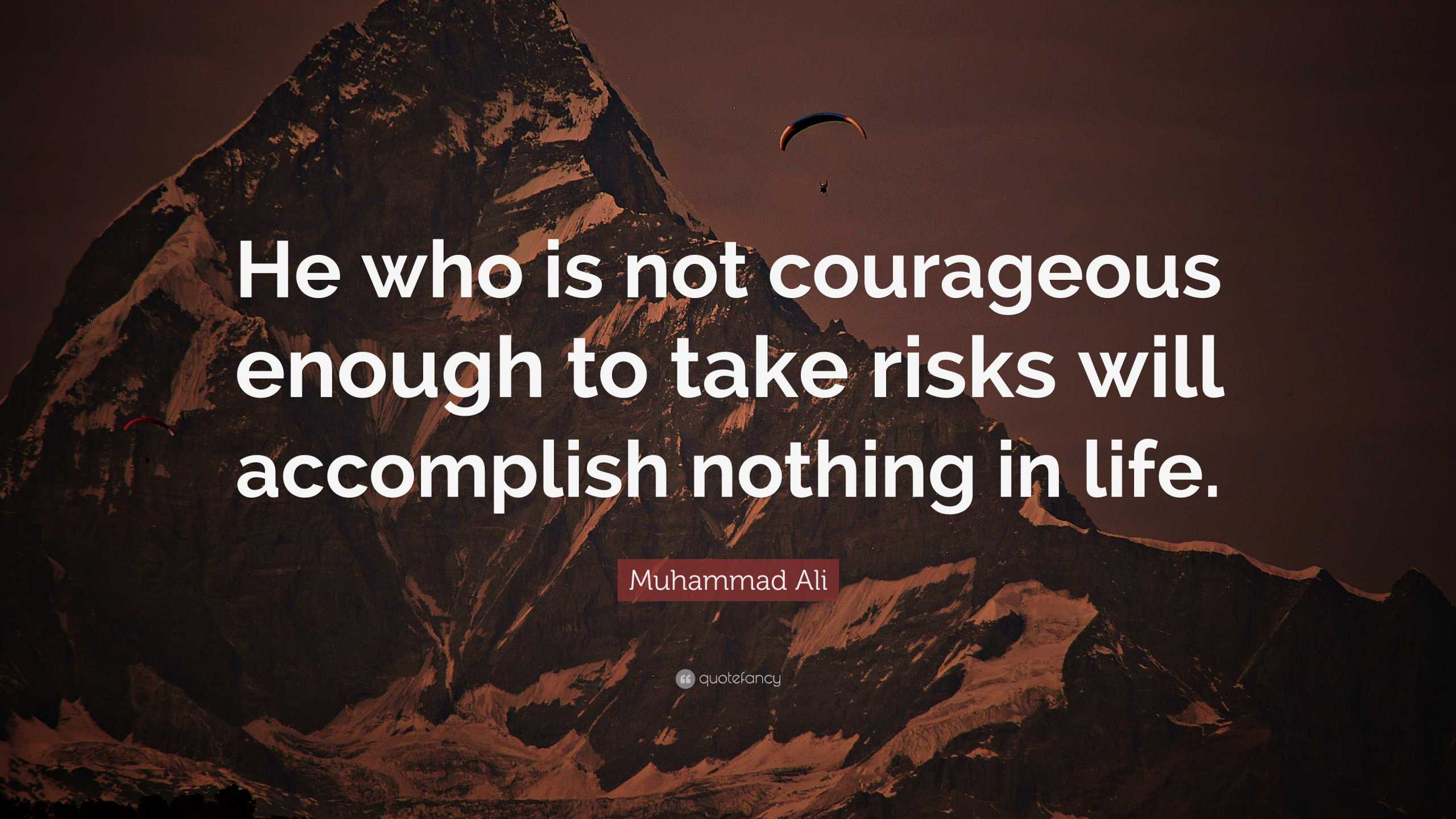
Data Lake

Repository of Raw Data
Without Schema Enforcement

Object Storages

Storages Optimized for Big Data Use-Cases
Scalability & Reliability




He who is not courageous enough to take risks will accomplish nothing in life.
Muhammad Ali
quotefancy

Data Lakehouse

Best of a Data Warehouse and Data Lake

Data Warehouse [Dw] ~ 90s

Decision Support for BI Applications
Since **1980s** evolving & MPP Architecture
Structured Data & No Cost Efficient

OLTP System
RDBMS

ETL
Extract | Transform | Load

Data Mart [DM]
Departmental Business Decision

Data Warehouse [Dw]
Ralph Kimball [Bottom-Up]

Data Lake [DL] ~ 2000s

Large Amount of Data [DV] from Different Sources
Since **2000s** being a Data Raw Repository
Without Transactions, Data Enforcement, Lack of Consistency
Batch & Streaming Jobs not Performing as Expected
AI Advances based on Unstructured Data (Text, Images, Video, Audio)

Data Sources
Applications, RDBMS & NoSQL

ELT
Extract | Load | Transform

Data Lake
Structured, Semi-Structured & Unstructured

Lakehouse [LH] ~ 2020s

Similar Structure of **Dw** with a **Low-Cost Storage** with Features:

- Transaction Support
- Schema Enforcement and Governance
- BI Support
- Decoupled Storage & Computation
- Openness with Structured & Unstructured Data
- End-to-End Streaming

Data Sources
Structured, Semi-Structured & Unstructured Data

ELT
Extract | Load | Transform

Lakehouse
BI, ML, Streaming Analytics & Data Science

Data Lakehouses ~ Hudi, Iceberg & Delta

Apache Hudi

- Ingests & Manages Storage of Large Analytical DataSets over DFS
- Best of Breed for Streaming Processing over Big Data Workloads ~ Primitives over Apache Hadoop
- Update & Delete Records
- Change Streams

https://hudi.apache.org

Apache Iceberg

- Table Format for Huge Analytical DataSets - Trino & Apache Spark Use as a High Performant Format using SQL Interface
- Schema Evolution, Hidden Partitioning, Partition Layout Evolution, Time Travel Feature

https://iceberg.apache.org

Delta Lake

- Building a Data Lakehouse Architecture on Top of Existing Storage System
- S3, ADLS, GCS, HDFS, OCS, IBMCOS
- ACID Transactions, Scalable Metadata Handling, & Unifies Streaming & Batch Data Processing
- Fully Compatible with Apache Spark Engine

https://delta.io

Data Lake

Repository of Raw Data
Without Schema Enforcement

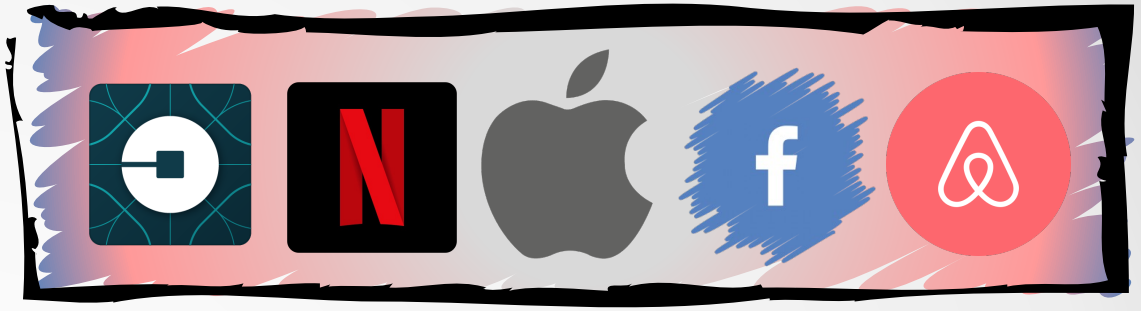
Apache Spark and Data Lakehouses

Distributed Cluster-Computing Framework
Optimized for Memory Computation

Conclusion

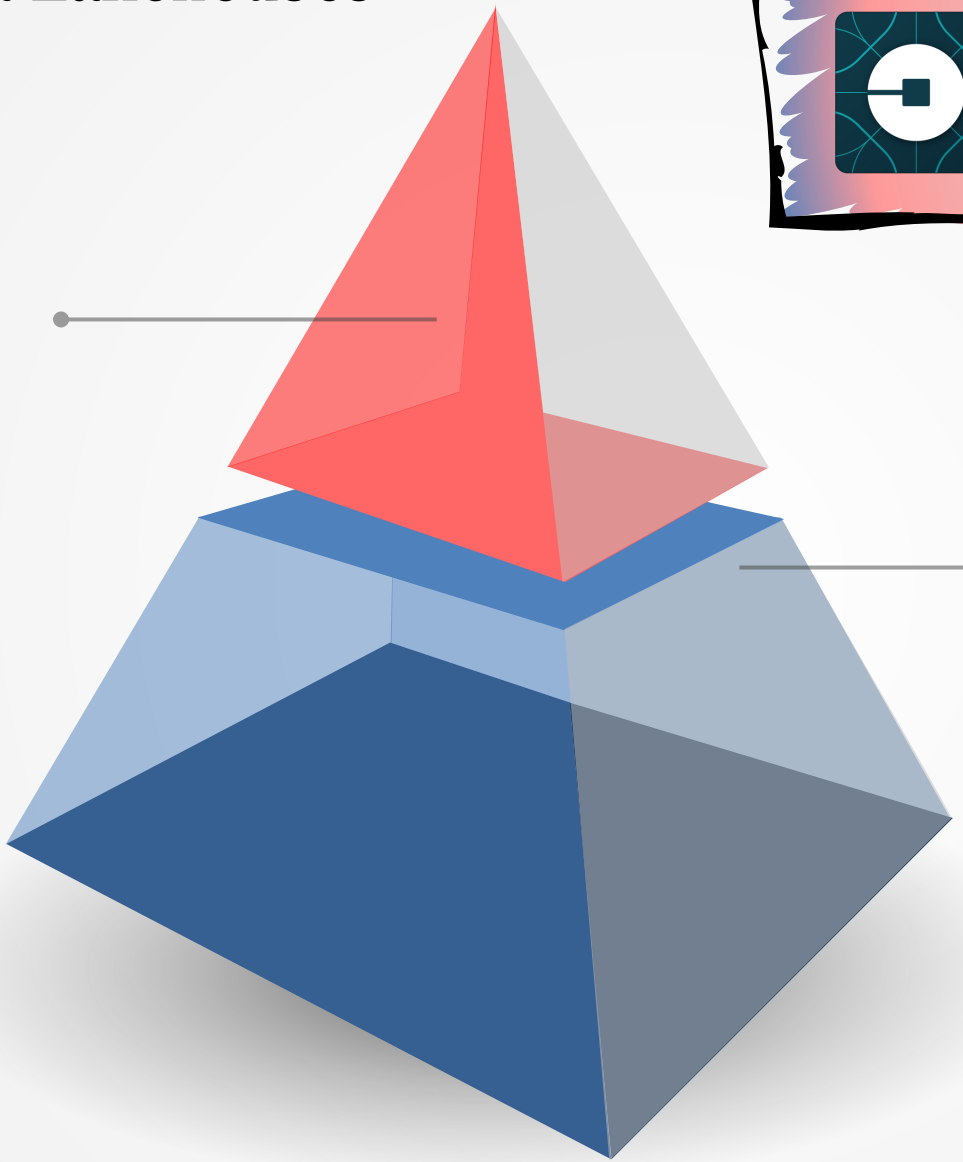
- Apache Hudi ~ Best Conveniences for Streaming Process
- Apache Iceberg ~ Better Design, Abstraction & Integrations
- Delta ~ Best Integration with **Apache Spark** Ecosystem

Data Lake vs. Data Lakehouses



Data Lakehouses

Metadata Layers for Data Lakes
New Query Engine Design
High-Performance SQL Execution
Optimized Access of Data



Data Lake

Repository of **Raw** Data
Unsiloed Data
Without Schema Enforcement
Data Swamp & Data Quality Issues

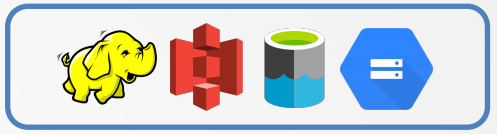
Being deeply loved by someone gives you strength, while loving someone deeply gives you courage.

Lao Tzu

Delta Lake [Features]

Delta Lake

Is an Open-Source Storage Layer ~ ACID Transactions to Apache Spark & Big Data Workloads



ACID Transactions

data lakes typically have multiple data pipelines reading and writing data concurrently, and data engineers must go through a tedious process to ensure data integrity, due to the lack of transactions. it provides serializability, the strongest level of isolation level.

Open Format

all data in delta lake is stored in apache parquet format enabling delta lake to leverage the efficient compression and encoding schemes that are native to parquet.

Audit History

delta lake transaction log records details about every change made to data providing a full audit trail of the changes.

Scalable Metadata Handling

delta lake treats metadata just like data, leveraging spark's distributed processing power to handle all its metadata. as a result, delta lake can handle petabyte-scale tables with billions of partitions and files at ease.

Unified Batch & Streaming

a table in delta lake is both a batch table, as well as a streaming source and sink. streaming data ingest, batch historic backfill, and interactive queries all just work out of the box.

Updates & Deletes

delta lake supports Scala, Java, Python, SQL APIs to merge, update and delete datasets. this allows you to easily comply with GDPR and CCPA and simplifies use cases like change data capture.

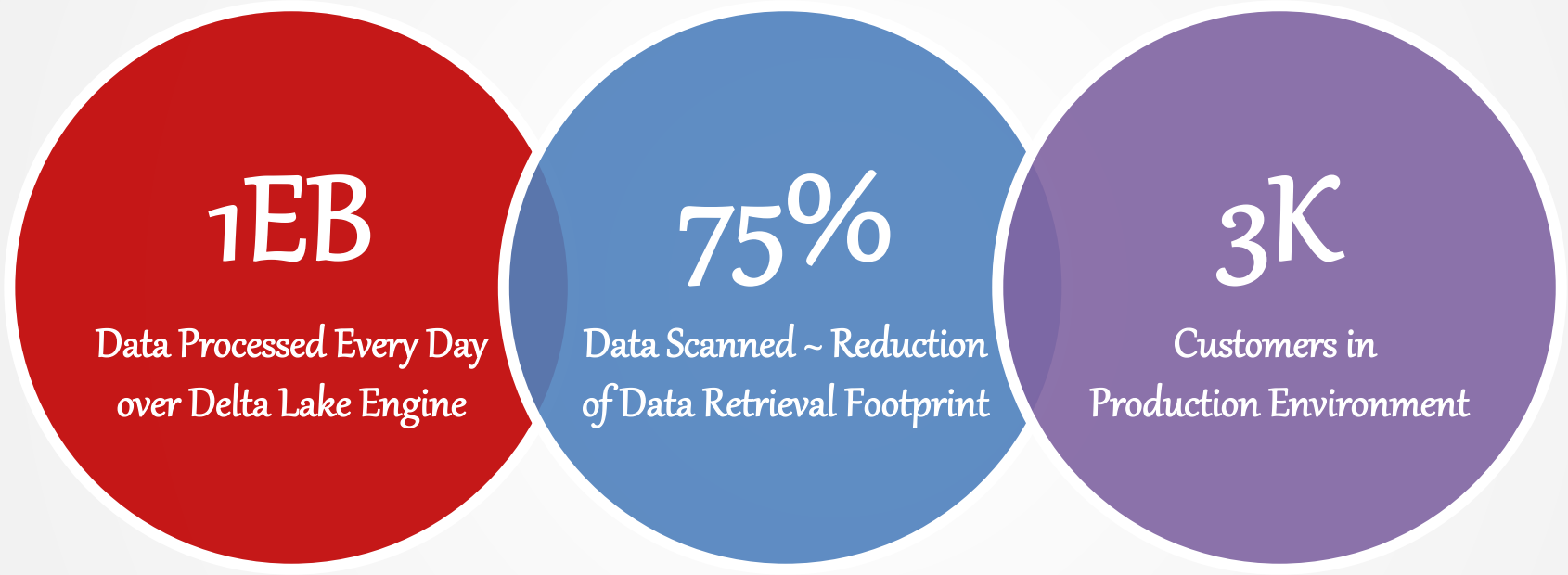
Time Travel [Data Versioning]

delta lake provides snapshots of data enabling developers to access and revert to earlier versions of data for audits, rollbacks or to reproduce experiments.

Schema Enforcement & Evolution

delta lake provides the ability to specify your schema and enforce it. this helps ensure that the data types are correct and required columns are present, preventing bad data from causing data corruption. delta lake enables you to make changes to a table schema that can be applied automatically, without the need for cumbersome dll.

Delta Lake Quick Stats

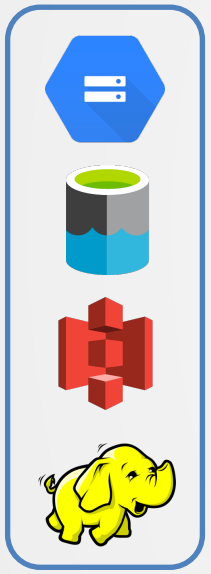


Delta Sharing

Open Standard for Secure Data Sharing

Delta Sharing ~ Industry's First Open Protocol for Secure Data Sharing, Simple to Share Data with Organizations Regardless of Which Computing Platforms of Use

Data Provider

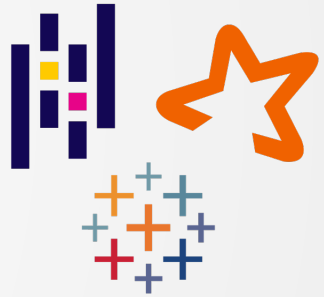


Delta Lake Table

Delta Sharing Server

Delta Sharing Protocol

Data Recipients



Scalability

Share Terabyte-Scale DataSets Reliably & Efficiently by Leveraging Cloud Storage Systems

Share Live Data Directly

Easily Share Existing, Live Data in Your Delta Lake Without Copying to Another System

Security and Governance

Delta Sharing Allows You to Easily Govern, Track & Audit Access to Your Shared Data Sets

Support Diverse Clients

Data Recipients Can Directly Connect to Delta Lake Tables from Pandas, Apache Spark, Rust and Other Systems Without Having to First Deploy a Specific Compute Platform. Reduce Friction to Get Your Data to Your Users

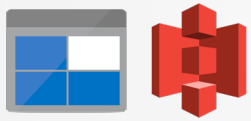
Data Ingestion into [Delta Lake]

Network of Partners [Connectors]

essential ecosystem of connectors to bring data into **Delta Lake**
Azure Data Factory, Fivetran, Qlik, Infoworks, StreamSets, Syncsoft

Cloud Storage [Auto Loader]

loading data continuously from cloud stores with **Exactly-Once** guarantees at low cost, low latency and minimal DevOps work. List is one of the most expensive operations. **Auto Loader** is an optimized file source that uses **Structured Streaming**

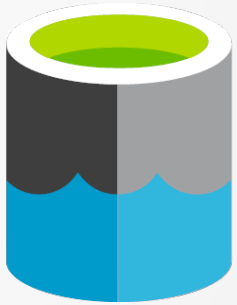


Streaming Load [Structured Streaming]

loading data continuously from **Apache Kafka** with **Exactly-Once** using the Structured Streaming API

Delta Lake

storage system with ACID capabilities that is designated to build a **Data Lakehouse**



Delta Lake [Apple's Use-Case]

Streaming ETL

Telemetry & Logs

Delta

Streaming Table

Phase 1 - Ingestion

Ingestion Tables

Delta

Streaming Refinement

Phase 2 - Refinement

Refined Tables

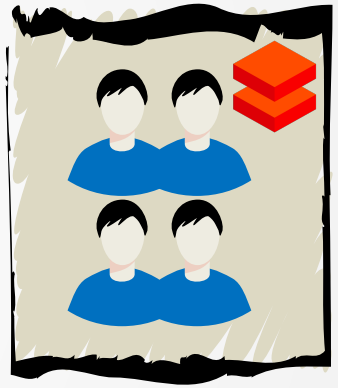
Refined Tables

Refined Tables

Analysts

Analysis & Dashboards

Machine Learning [ML]



Total Size = 504 TB

Scanned Data Size = 36.5 TB

Total Rows = 11,149,387,374,965

Data Skipping = 93.2%

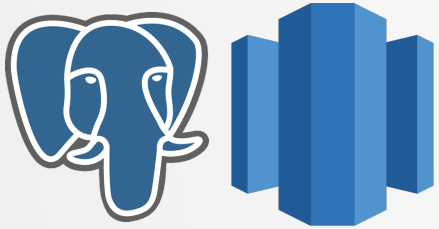
Delta Lake [Atlassian's Use-Case]

Multi Petabyte ~ 24 TB a Day

3.000 Internal Users

200 Data Engineers

First Generation



Data Warehouse

Postgres ~ Most Advanced Open-Source Database, Flexible & Can Serve as a Simple Relational Database, a Time-Series, and Even as an Efficient, Low-Cost, Data Warehousing Solution.

Amazon Redshift ~ Query and Combine Exabytes of Structured and Semi-Structured Data Across Your Data Warehouse, Operational Database, and Data Lake using Standard SQL. Query S3 Data Lake using Open Formats ~ Apache Parquet.

Second Generation

Data Lake

S3 + Apache Hive & Presto using Amazon EMR & EC2 = Data In One Single Place, Infinitely Scalable, Performance Implications, Lack of Dimensional Modeling, Higher Barrier for Self-Service.

Third Generation

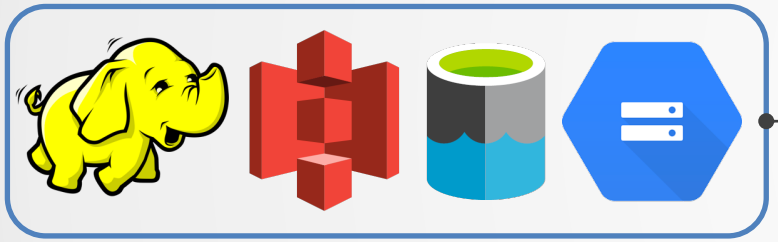
Data Lakehouse

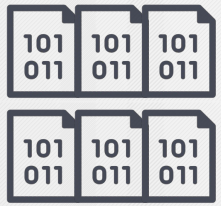
Threating Data Lake as Database, Enabling ACID Transactions and Effective Concurrency, Enabling Batch and Near Real-Time Use-Cases Scenarios using The Delta Architecture

The Medallion Architecture

Data Lake

Repository of Raw Data
Without Schema Enforcement



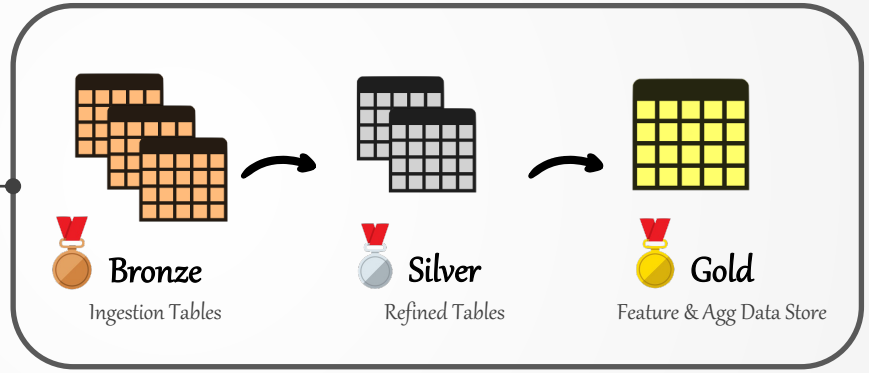


Batch

Stream

Delta Lake

Storage Layer with ACID Transactions
Big Data Workloads



Raw Ingestion

Transformations

Business Level

Building a Data Lakehouse using Batch-ETL


What you focus on grows.
Esther Hicks
quotefancy

ONEWAY

SOLUTION