LAKEHOUSE WITH DELTA LAKE

PART 1: GOOD BYE DATA WAREHOUSE - WELCOME LAKEHOUSE!

What is the Lakehouse?

+

=

Data Warehouse

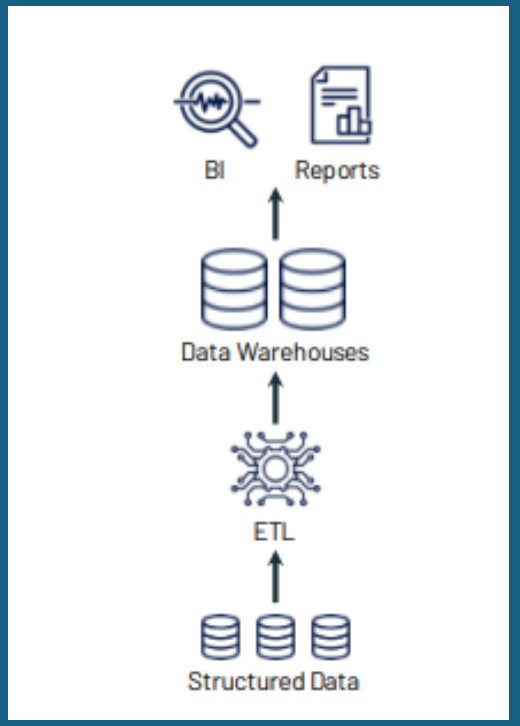
Data Lake

Data Lakehouse

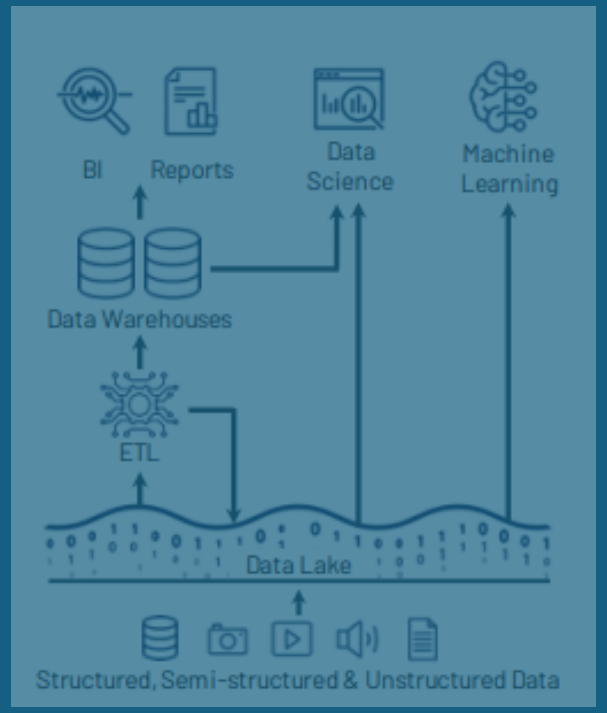
- Open direct access data formats like parquet
- First class support for ML
- State of the art SQL performance

Source: https://delta.io/pdfs/DLTDG_ER4.pdf Based on a research paper published 2021 by Databricks

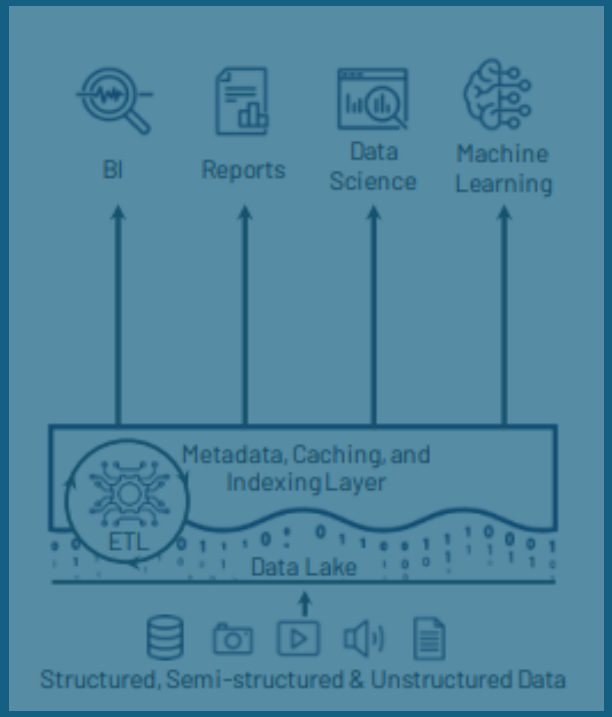
Data Warehouse



Two tier Architecture



Data Lakehouse



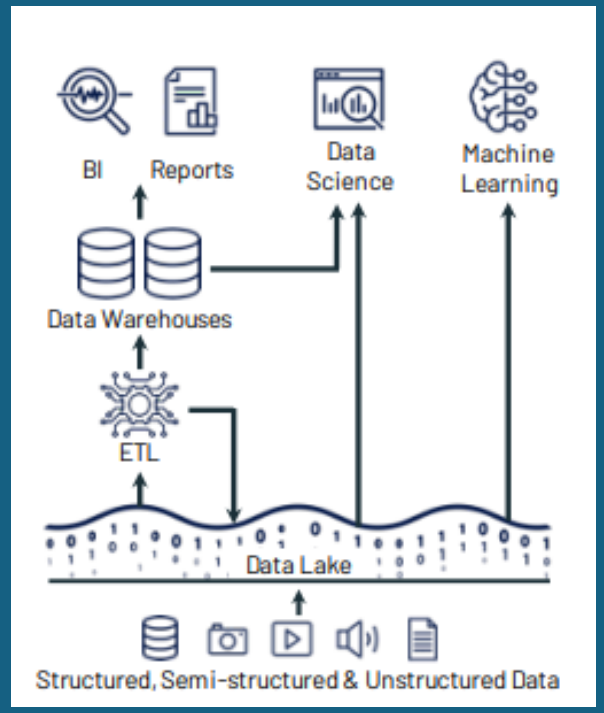
- Collecting data from transactional databases
- Purpose analytics and data insights
- One central place for all data
- Central access control, governance and ACID* for data reliability and integrity

- Coupling of compute and storage
- Set up and thus costs based on peak performance
- Costs scale with the data size and generally high
- Can’t support unstructured and semi-structured data like pictures and JSONs

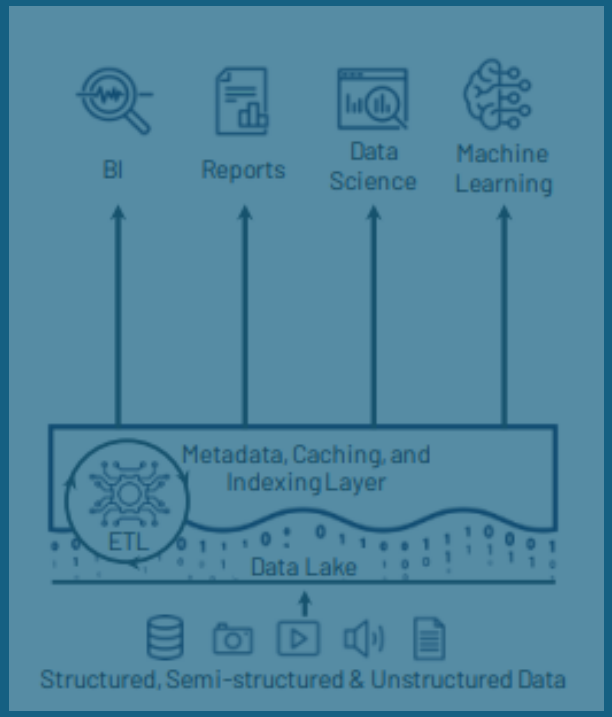
*Atomicity, Consistency, Isolation, and Durability

Data Warehouse

Two tier Architecture



Data Lakehouse

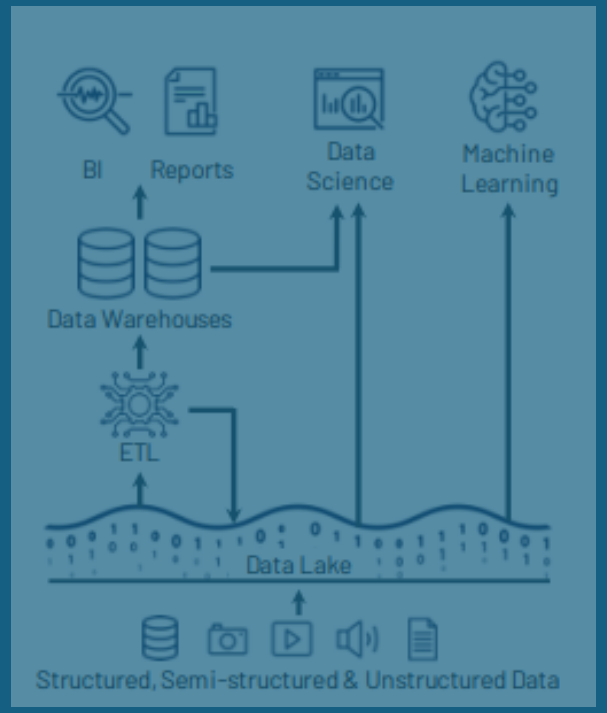


- Save data into low costs storage systems with easy access and flexibility e.g. HDFS
- Later also even cheaper cloud storages like ADL2 with features like geo replication
- Subset of data is moved to Data Warehouse
- Machine Learning possible as DataFrame APIs available for e.g. parquet

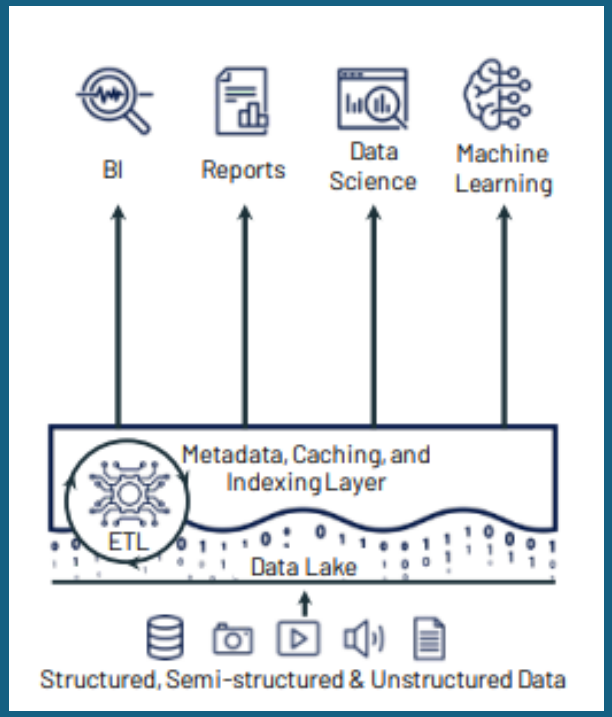
- Duplication of data/costs, low reliability e.g. out of sink, complexity, delays, bugs, data swamps
- Quality, schema enforcement or governance of data hard on ADL2
- ML libraries like XGBoost and Pytorch do not run well on Data Warehouses leading to copy data locally

Data Warehouse

Two tier Architecture



Data Lakehouse



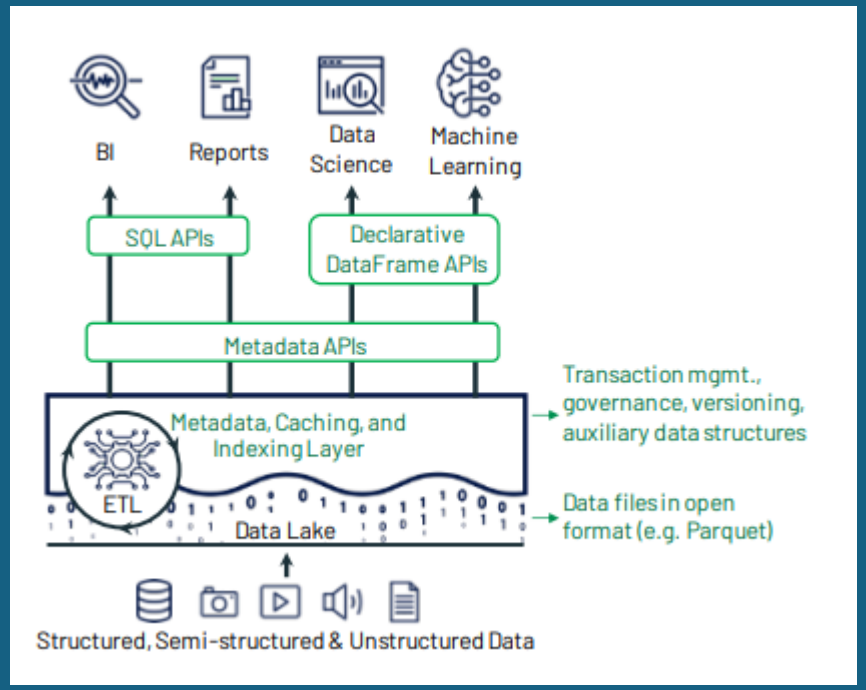
One layer for all kind of workloads:

- Transactional Support for ETL workloads using open formats like parquet for Streaming and Batch processing
- BI Support via SQL and JDBC interfaces with a powerful engine
- Support for Machine Learning
- Use open Standards

Further Features:

- Schema enforcement and Governance incl Audit logs and Data Integrity (ACID)
- Indexing
- Data Versioning
- Separation between Storage and Compute
- Structured, Semi-Structured, Unstructured data support

Key Components of the Lakehouse



Metadata layer:

- File format, transactional meta layer on top of parquet with Data Integrity (ACID), batch and streaming
- Schema enforcement and Governance incl Audit logs and Versioning, Indexing, Transaction history

SQL API:

- Caching in SSDs and RAM, partially decompressed, faster storage
- Maintain and leverage statistics saved in the meta data layer for file skipping
- Indexing/ Cluster parquet files in based on required queries
- Backed by Spark Engine and open source jdbc

Machine Learning:

- Dataframe APIs like Pandas used for ML modules like XGBoost can leverage the saved data

Storage Frameworks serving as Meta Data Layer

Created by Databricks, available 2017

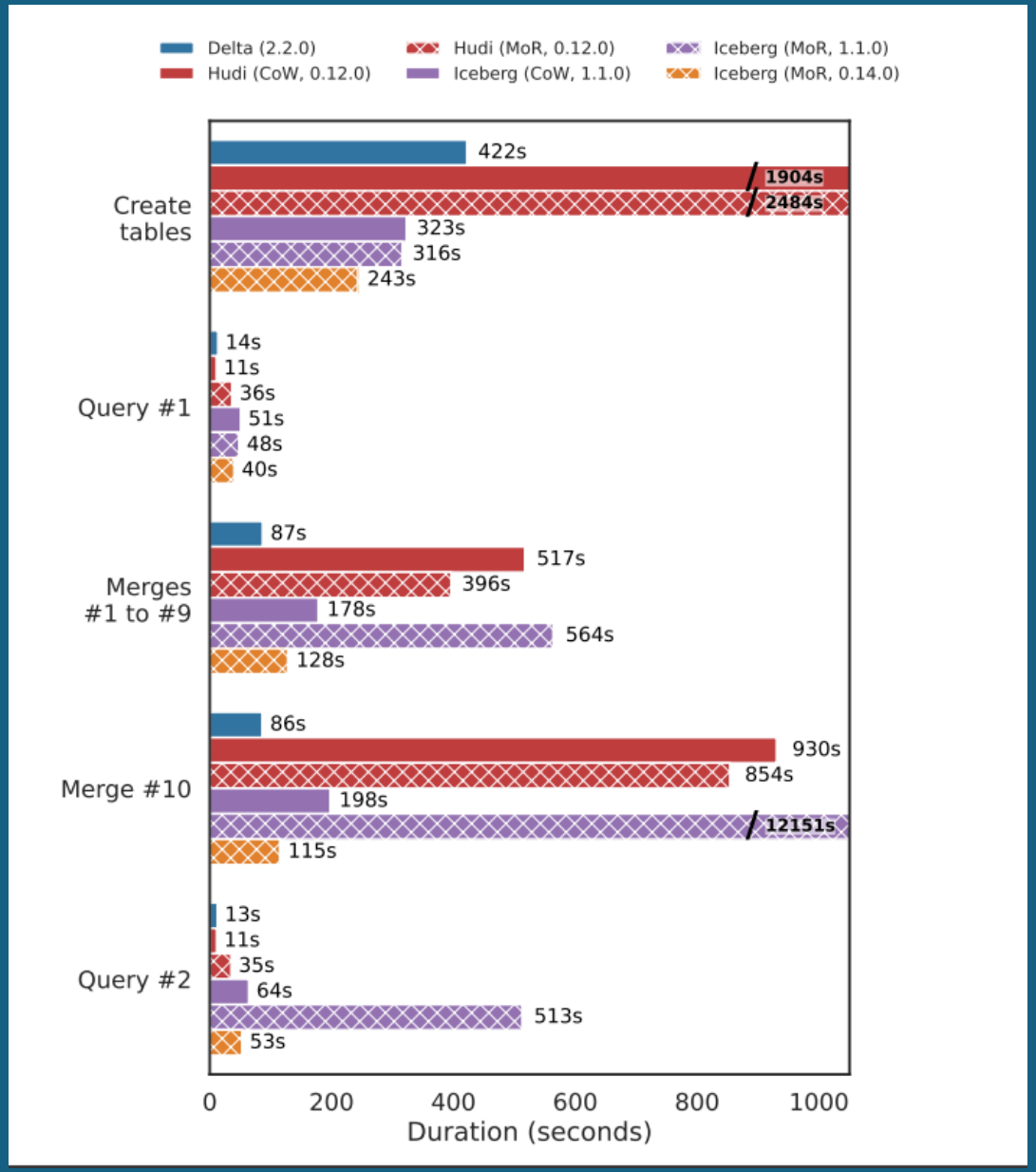
Created by Netflix, available 2017

Created by Uber, available 2016

+

+ **Metadata**

Focus: Delta Lake

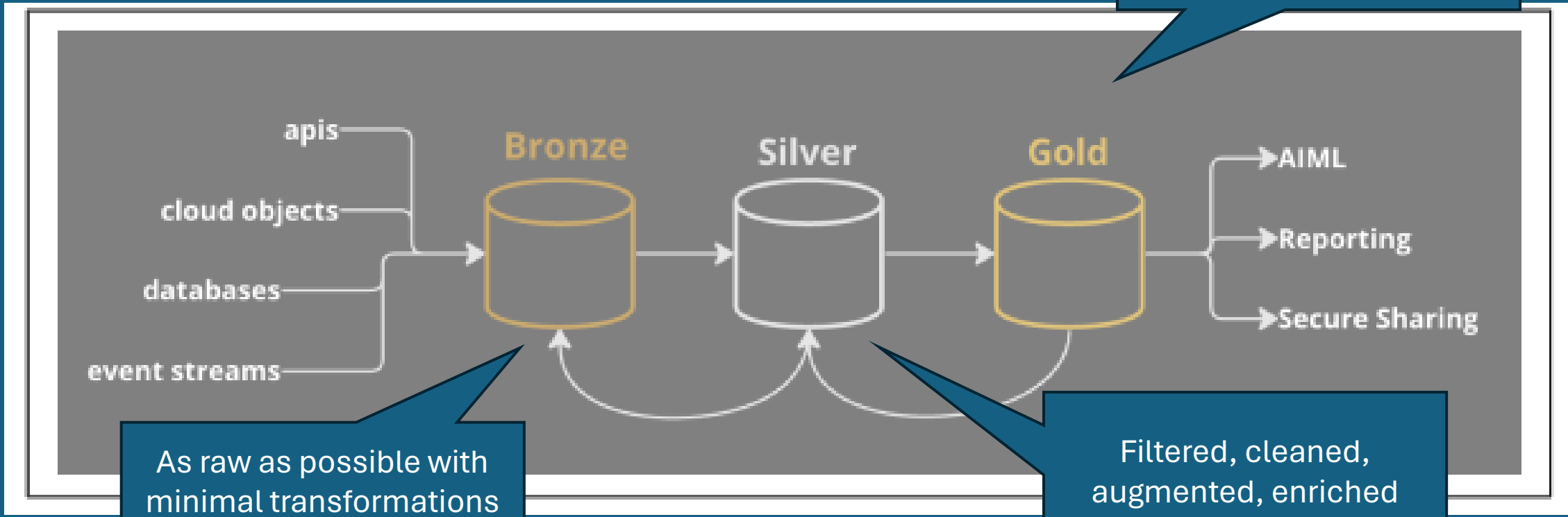


Source: Analyzing and Comparing Lakehouse Storage Systems (cidrdb.org)

The Medalion Architecture

Medallion architecture: Design pattern to logically organize data

Purpose/ use-case build tables incl. views, aggregations, joins, report optimal tables



As raw as possible with minimal transformations for reprocessing and exploration

Filtered, cleaned, augmented, enriched data

Source: https://delta.io/pdfs/DLTDG_ER4.pdf

Summary

- The Lakehouse combines the best of Data Warehouses and Data Lakes from one single place
  - Schema management and Governance
  - ACID transactions
  - Operations like Inserts, Deletes, Updates
  - SQL Support
  - Machine Learning
  - Streaming and Batch processing
  - Structured and unstructured data