Data architectures

Prof. Dr. Jan Kirenz
HdM Stuttgart

Data architecture

Consists of data

- schema (star, snowflake, multidimensional),
- integrations,
- transformations,
- storage,
- workflows,

required to enable the analytical requirements of th information architecture

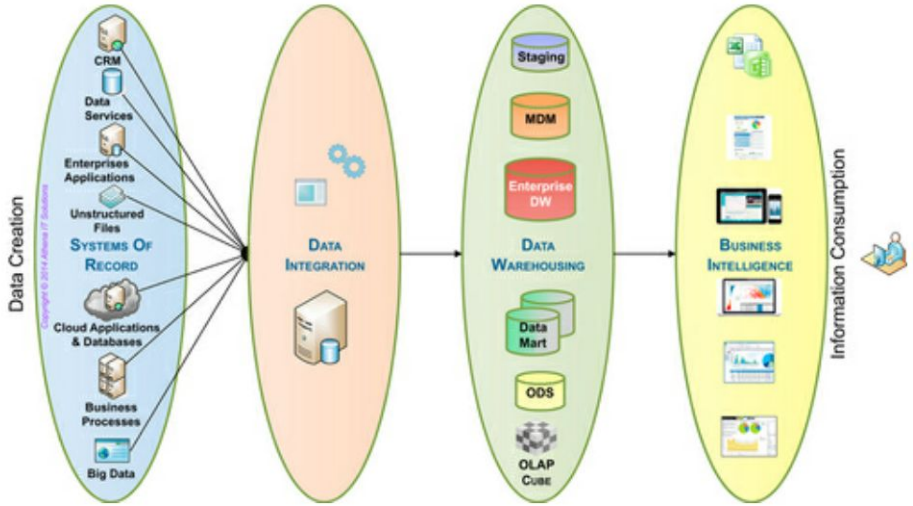


FIGURE 4.3 Data architecture workflow.

Source: Sherman (2014)

Data architecture

The data architecture guides how the data is:

- collected,
- integrated,
- enhanced,
- stored, and
- delivered to

people who use it to do their jobs.

It helps make data:

- available,
- accurate, and
- complete

so it can be used for decision-making.

Source: Sherman (2014)

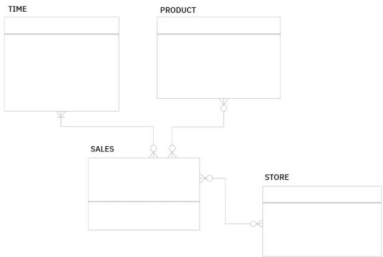
Data architecture vs data modeling

Data architecture

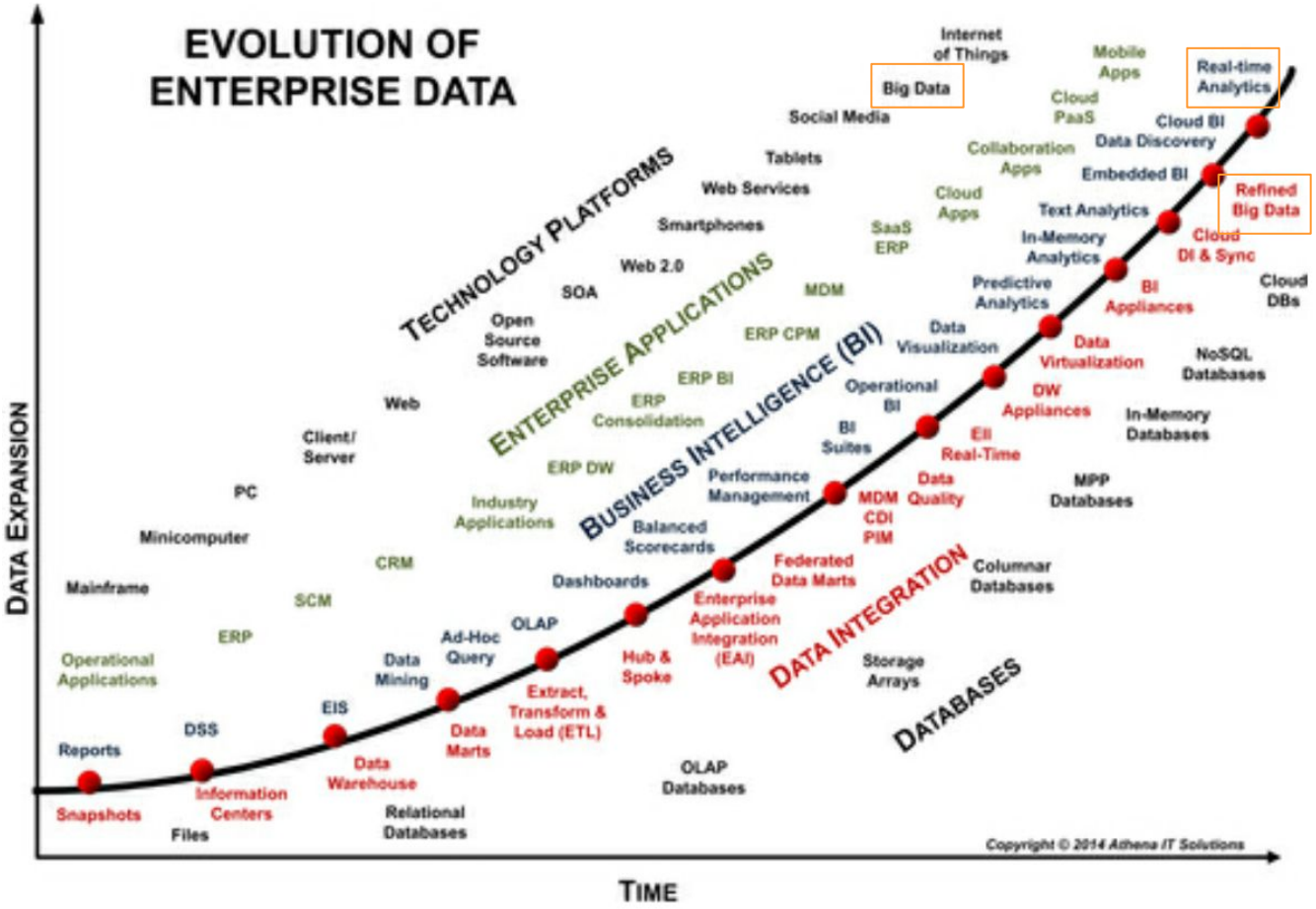
- Applies to the higher-level view of how the enterprise handles its data
- Such as how it is categorized, integrated, and stored.
- “Blueprint for your house”

Data modeling

- Applies to very specific and detailed rules about how pieces of data are arranged in the database.
- “Instructions for installing a faucet”

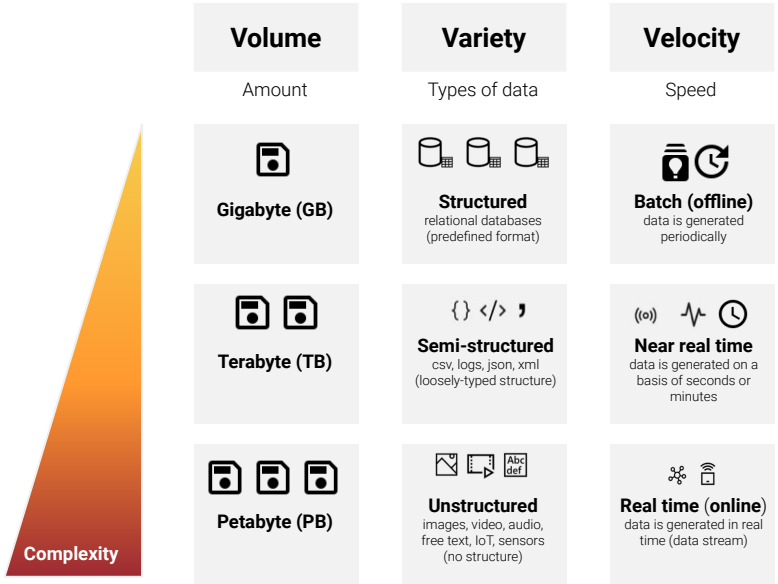


Source: Sherman (2014)



Source: Sherman (2014)

(Big) data characteristics



Databases

Relational database technology

- Used to handle structured data
- Have a certain schema (star, snowflake, …)
- Cannot handle unstructured data
- We can use SQL

NoSQL databases

- Used to handle unstructured data such as
  - Text data, Images, Videos
- “Not Only” SQL: SQL is not required but may be used for some of these databases
- Schema-free design
- Flexibility to start loading data and then changing it later

Source: Sherman (2014)

NoSQL

NoSQL databases fall into several technology architectures categories:

1. Key-Value
2. Column-Family
3. Document
4. Graph

Source: Sherman (2014)

Technical architecture

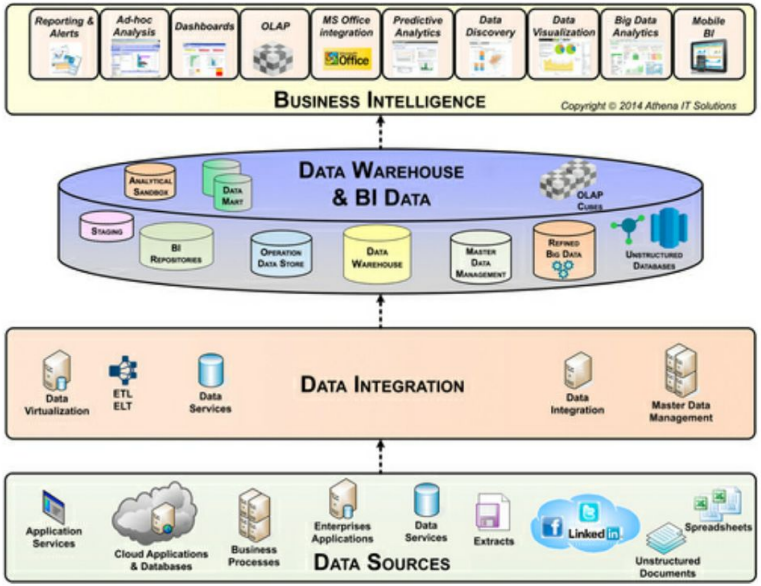


FIGURE 4.6 BI technical architecture categories.

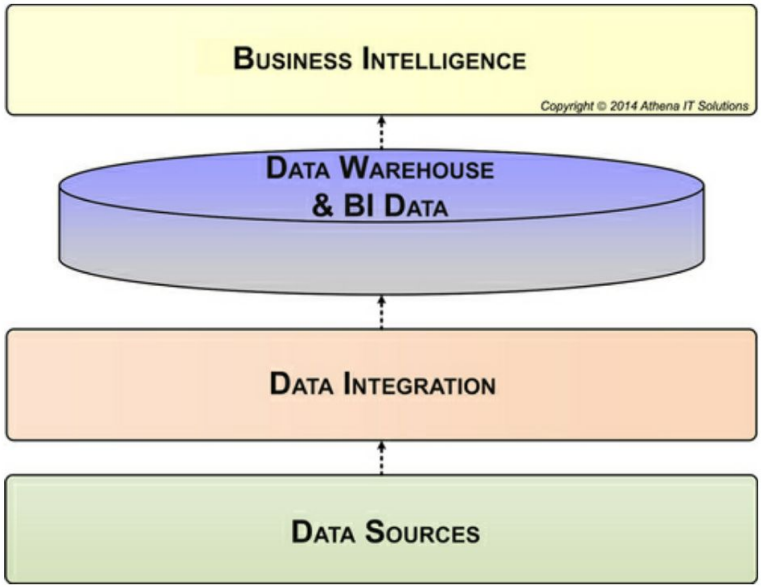


FIGURE 4.5 BI technical architecture.

Source: Sherman (2014)

Data integration: ETL vs ELT

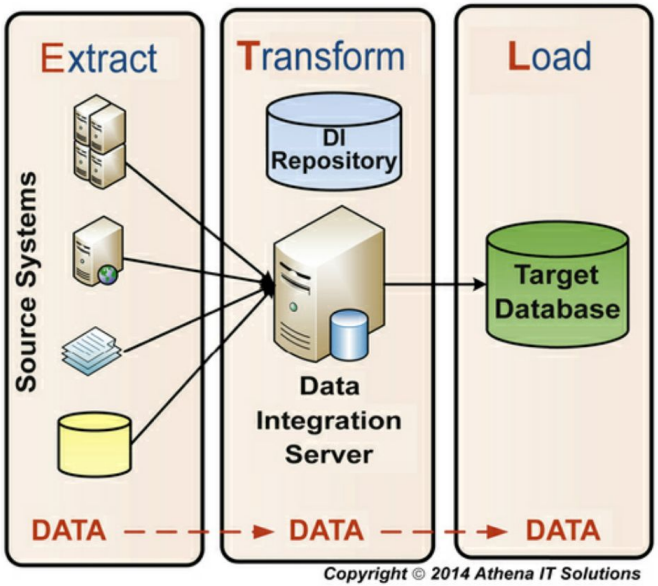


FIGURE 7.5 ETL architecture.

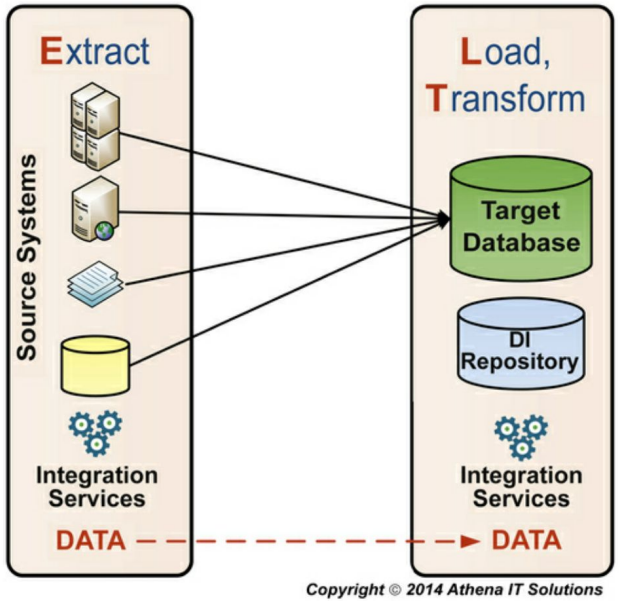


FIGURE 7.6 ELT architecture.

Source: Sherman (2014); E: Extract; T: Transform; L: Load

… data integration is not only ETL/ELT…

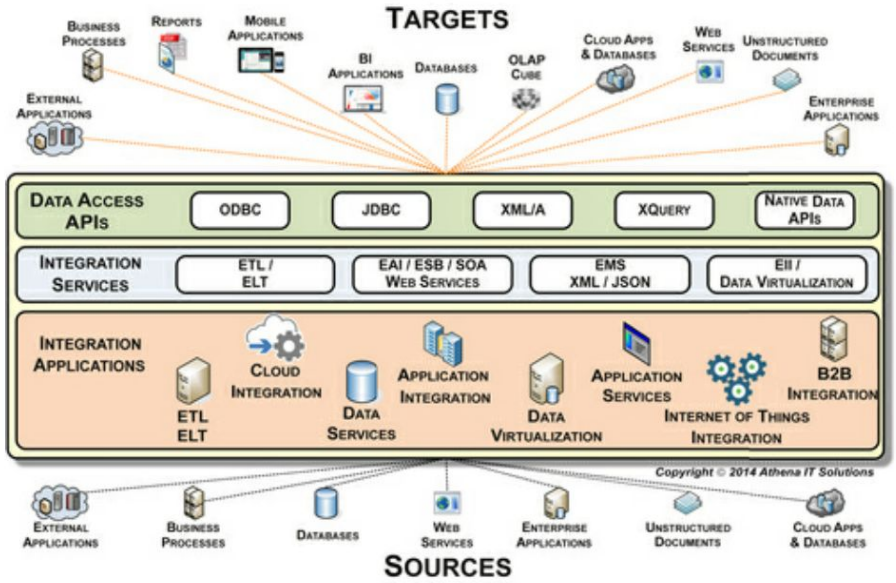
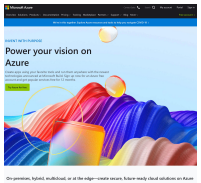


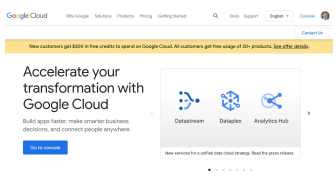
FIGURE 7.4 BI's information access & data integration layer.

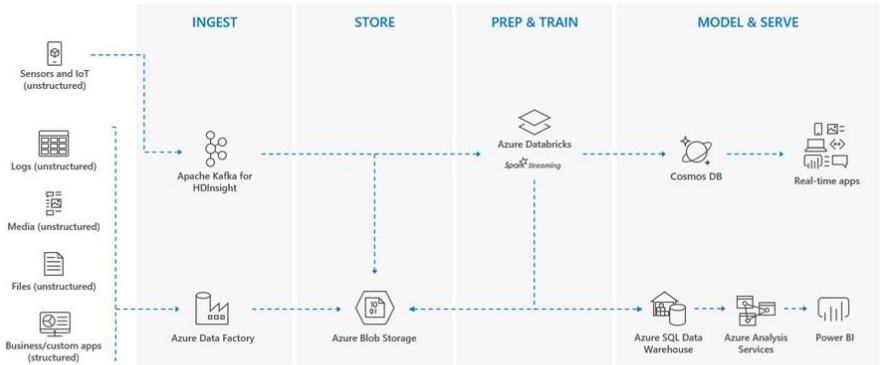
Source: Sherman (2014)

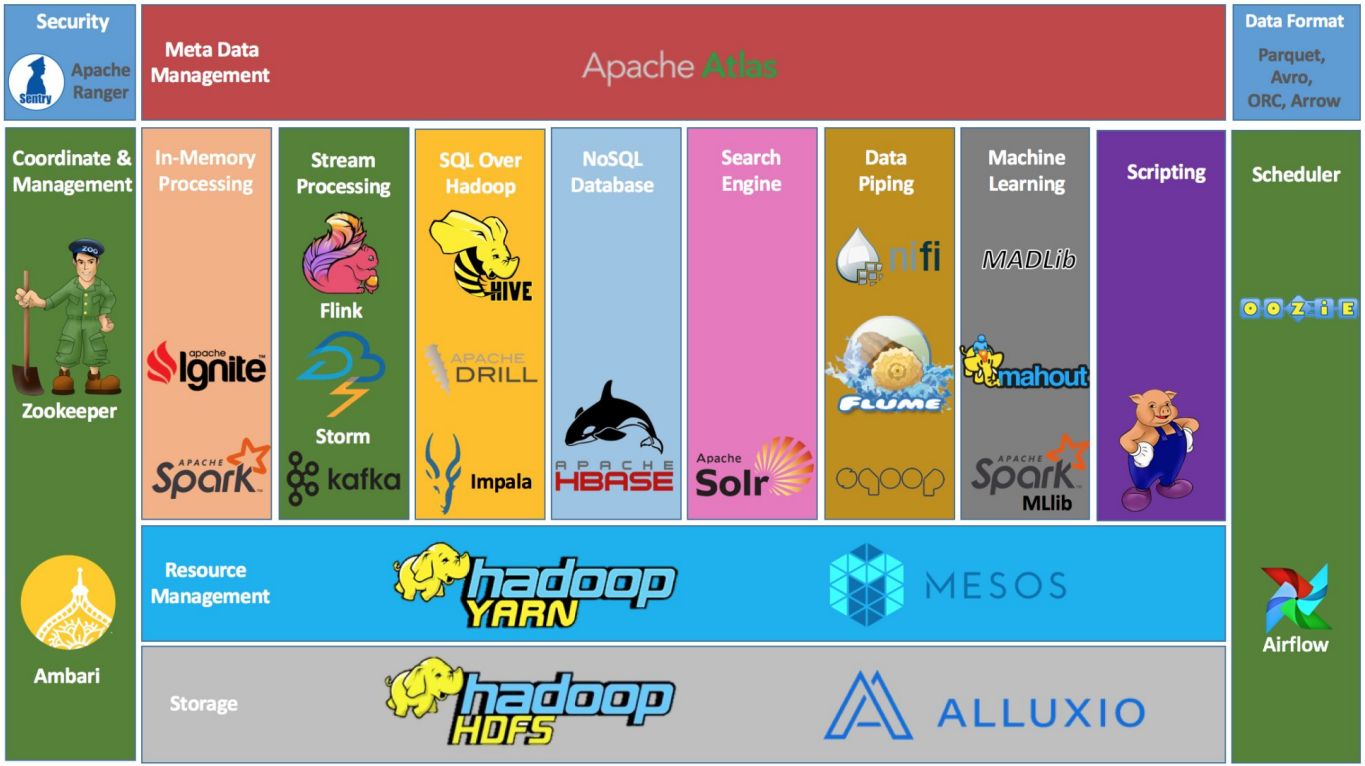
Product architecture

- Products, configurations and how products are interconnected








Security
Apache Ranger
Meta Data Management
Apache Atlas
Data Format
Parquet, Avro, ORC, Arrow
Coordinate & Management
Zookeeper
Ambari
In-Memory Processing
Stream Processing
Flink
Storm
kafka
SQL Over Hadoop
HIVE
APACHE DRILL
Impala
NoSQL Database
APACHE HBASE
Search Engine
Apache Solr
Data Piping
nifi
FLUME
oozie
Machine Learning
MADLib
mahout
Spark MLlib
Scripting
Scheduler
OOZIE
Airflow
Resource Management
hadoop YARN
MESOS
Storage
hadoop HDFS
ALLUXIO

How to select an architecture?

Selecting an architecture

Typical choices are

- Enterprise Data Warehouse (EDW)-only
- Independent data marts
- Hub-and-spoke
- Ralph Kimball's enterprise data bus architecture
- Bill Inmon's Corporate Information Factory (CIF)

Recommended

- Analytical data architecture (ADA), includes unstructured data

Related concepts

- Data lake
- Data Lakehouse

Source: Sherman (2014)

Centralized enterprise data warehouse

- Integrated
  - data gathered and made consistent from one or more source systems
- Subject-oriented
  - organized by data subject rather than by application
- Time-variant
  - historical data is stored
- Non-volatile
  - data did not get modified in the DW, they were read-only

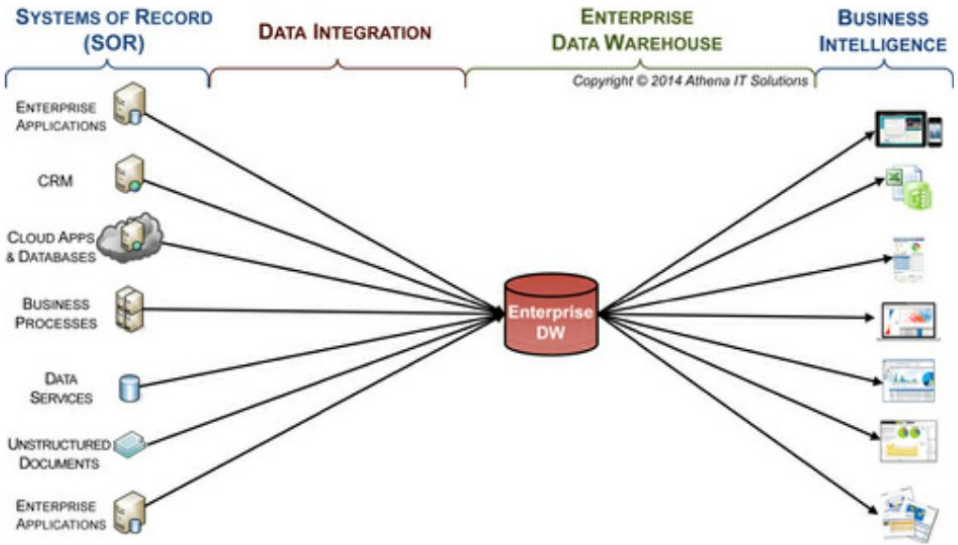


FIGURE 6.1 Enterprise data warehouse (EDW).

Source: Sherman (2014)

Independent data marts

- Independent from the DW
- Extracts data directly from the source systems rather than pulling the data from (and being dependent on) the DW.

Initially, data marts were perceived as a great success—until business groups realized they were debating in meetings which one of their reports (and associated data marts) had the "right" numbers.

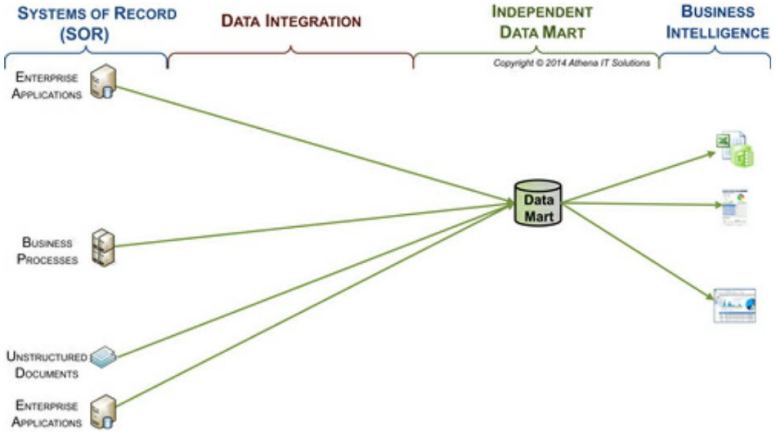


FIGURE 6.2 Data mart.

Source: Sherman (2014)

Multiple independent data marts

- Each data mart pulls data directly from the source systems
- Each report uses one of the independent data marts as its data source.
- Advantages:
  - Can be build quickly
  - Is problem-specific
- Disadvantage:
  - Creation of data silos
  - No data consistency (inconsistent reporting and metrics)

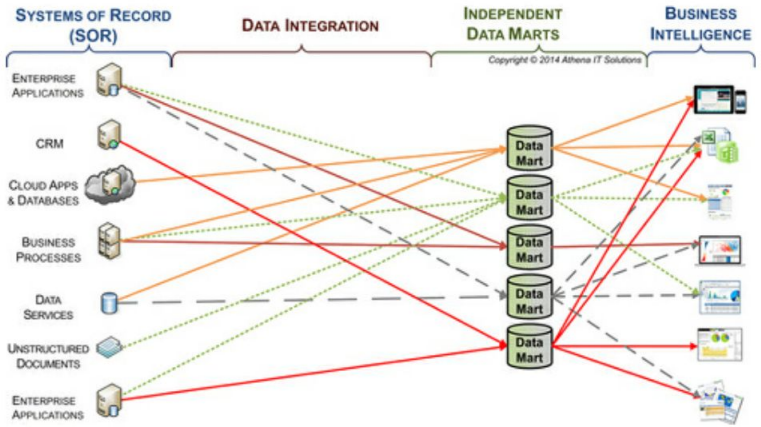


FIGURE 6.3 Multiple independent data marts.

Source: Sherman (2014)

Operational data store (ODS)

- Loading (unstructured) data into one location
- Often used as SOR to the EDW for the data it extracted from the source systems
- Disadvantages:
  - Overlapping data can create inconsistencies (ODS vs DW)
  - inflexible with changes in data and analytical requirements

Goal: bring data together from multiple source systems on as close to a real-time basis as possible to enable specific business processing or operational reporting

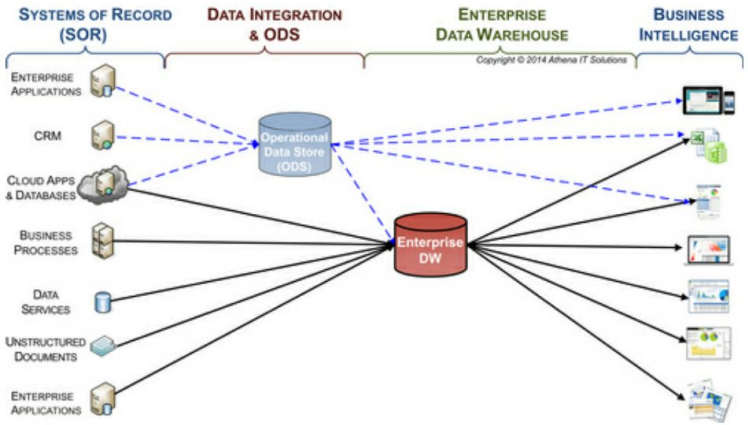


FIGURE 6.4 Operational data store.

Source: Sherman (2014)

Federated DWs

- Split the EDW into multiple physical DWs
  - Geographical regions or countries
  - Business functions such as finance, sales, marketing, or HR
  - Business entities such as divisions or subsidiaries
- Disadvantages:
  - Performance issues,
  - Lack of DI,
  - No reliable source of historical data.

The best practice is to design EDW logically as a single data store, but physically implement it as either one data store or federated based on performance and business needs.

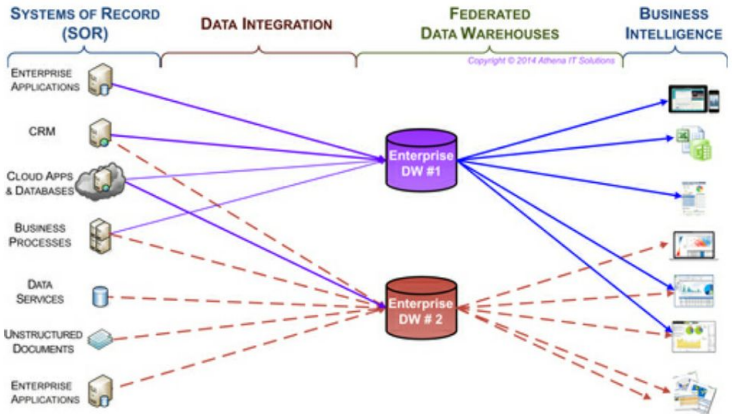


FIGURE 6.5 Federated DWs.

Source: Sherman (2014)

Accidental architecture

- Patchwork of DWs, ODSs, and data marts
- Multiple databases, ETL tools, and BI products.
- Advantages:
  - Fast to address specific needs
- Disadvantages:
  - Data silos (different units have each built their own DWs independently without synchronizing data or transformations).
  - Overlapping and conflicting data

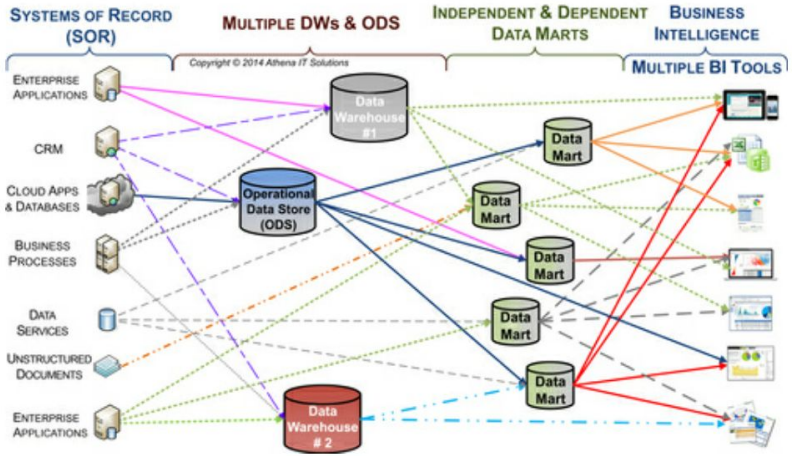


FIGURE 6.6 Multiple built BI silos & multiple BI tools.

Source: Sherman (2014)

Hub-and-spoke

- Enterprise data warehouse is feeding data marts
- Builds a physical DW (data hub) rather than trying to achieve a virtual hub.

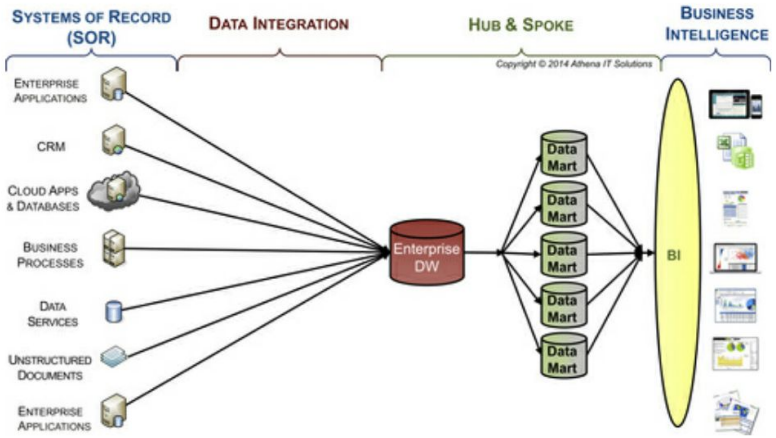


FIGURE 6.7 Hub-and-Spoke with one BI platform.

Source: Sherman (2014)

Hub-and-spoke

- Theoretically the hub-and-spoke enables the 5C's of information:
  - consistent, clean, comprehensive, conformed, and current.
- In reality this typically results in the scenario shown in Figure 6.8.
  - Multiple BI silos not only of different BI tools, but more damaging redundant and inconsistent reporting.

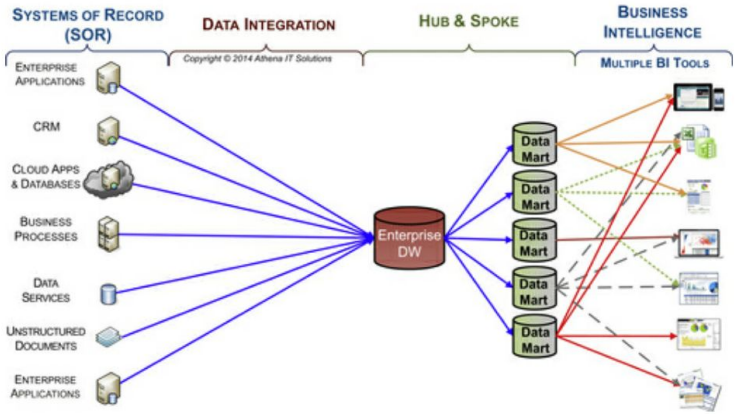


FIGURE 6.8 Hub and spoke with multiple BI tools.

Source: Sherman (2014)

Two divergent BI camps emerged in the 1990s to address the limitations of the EDW-only and independent data mart architectures:

- Bill Inmon’s Corporate Information Factory (CIF)
- Ralph Kimball’s enterprise data bus architecture

Source: Sherman (2014)

Inmon's CIF vs Kimball's enterprise data bus

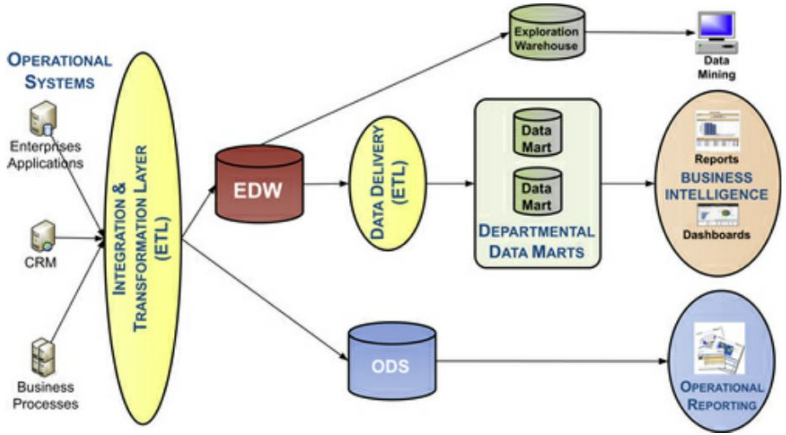


FIGURE 6.9 Inmon's CIF architecture.

CIF is a top-down design creating an enterprise-wide data model from the source systems to design the EDW, which is the core of the architecture

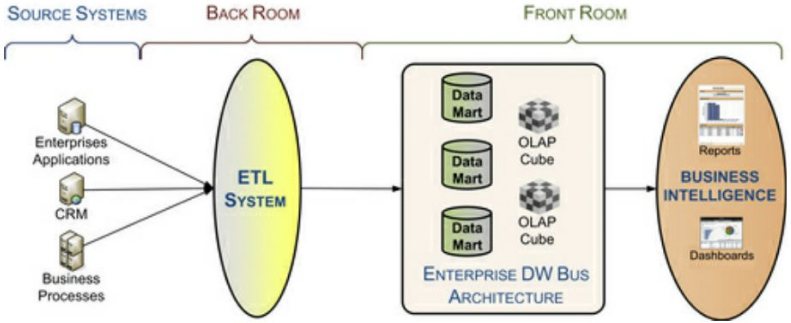


FIGURE 6.10 Kimball's enterprise data bus architecture.

Kimball's architecture assumed the ETL system could perform the 5C's (consistent, clean, comprehensive, conformed, and current) either on-the-fly or using a staging area and thus making the need for an EDW superfluous.

Source: Sherman (2014)

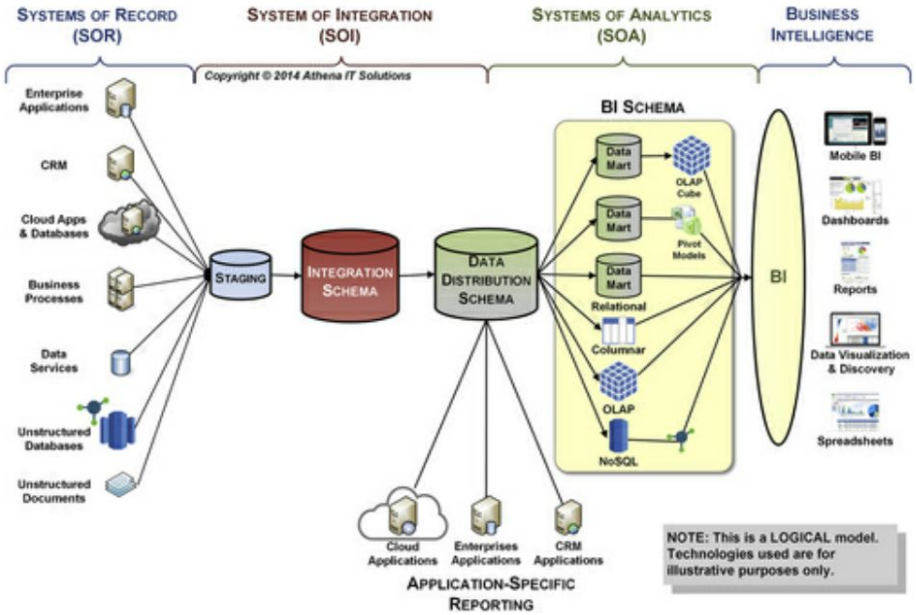
Inmon vs Kimball

Concepts	Inmon's CIF	Kimball's Data Bus
Key building block	EDW	Data marts
Data management process focus	Data integration	Business intelligence
Scope	Enterprise-wide	Business processes
Key data modeling technique	Normalization for EDW only; marts can be any design	Dimensional models
Data warehouse	Yes	Unnecessary
Data marts	Summarized from DW	Yes

Source: Sherman (2014)

Analytical data architecture (ADA)

- Systems of **record** (SOR)
  - Source systems where data is created, updated and deleted.
  - This layer is the source of data for systems of integration (SOI) layer but is managed outside the ADA.
- Systems of **integration** (SOI)
  - Data structures used to integrate data to enable the 5C's of information: consistent, clean, comprehensive, conformed and current.
- Systems of **analytics** (SOA)
  - data structures used by BI application to enable analytics



Source: Sherman (2014)

ADA: Enhanced Hub-and-Spoke Model

- The SOI replaces a lone physical EDW with SOI feeding the SOA.
- The goal of the SOI is to integrate data to create comprehensive, current, clean, and consistent information for the enterprise.
- The goal of the SOA is to enable BI by providing business-oriented information for analysis.
- The SOI logically contains an integration schema and staging area.
- The integration schema will be a physical data store, while the staging area may be persistent or transient based on use cases. The latter is the case when the DI processes do not need persistent data stores for acquiring or integrating data.
- The SOA contains the logical data distribution schema that potentially feeds data marts that, in turn, may feed sub-marts such as OLAP cubes or pivot tables.
- The SOA is a source of BI data either by BI tools accessing the data or by operational applications receiving distributed data from the data distribution schema. The latter would prevent operational applications from redundantly sourcing data from other applications when the EDW has already done so.

Source: Sherman (2014)

ADA: Enterprise data warehouse (EDW)

- The EDW is split into two schemas with different data objectives.
- The two schemas are:
    - EDW integration schema—its purpose is to enable the enterprise DI processes.
    - EDW data distribution schema—its purpose is to distribute the data downstream to the data marts so BI consumers can analyze the data or do application-specific reporting and advanced analytics that support the business.
- Although the EDW is the source of the enterprise's integrated 5 C's data, it is pragmatic to recognize that not all data needed for analytics will be available in the EDW.

Source: Sherman (2014)

ADA: Data marts

- The EDW data distribution model represents the enterprise dimensional model that feeds both data marts and application-specific reporting.
- Data marts are designed to support business processes or business groups.
- Sub-data marts are created to support different types of analysis performed within a business process or a business group.
- These sub-data marts are fed from data marts and derived by applying additional filters, business rules, and transformations that are specific to a particular business line of analysis.
- The dimensional model and data workflow depicted in the ADA are logical rather than physical.
- The logical dimensional model may be implemented in a variety of database technologies and the data workflow may be built in one or more databases (and database technologies).

Source: Sherman (2014)

ADA: Data schemas

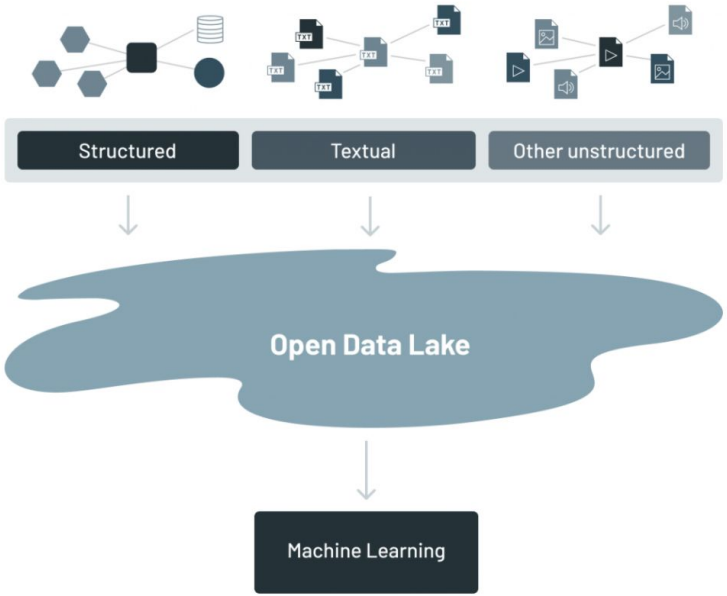
- The EDW staging area, if it is a persistent data store, may use a variety of schemas such as relational, flat file, or XML, depending on its use case.
- There may be a variety of EDW staging area data stores to support acquiring data from a similar variety of different source systems.

Source: Sherman (2014)

Data lake

Data Lake

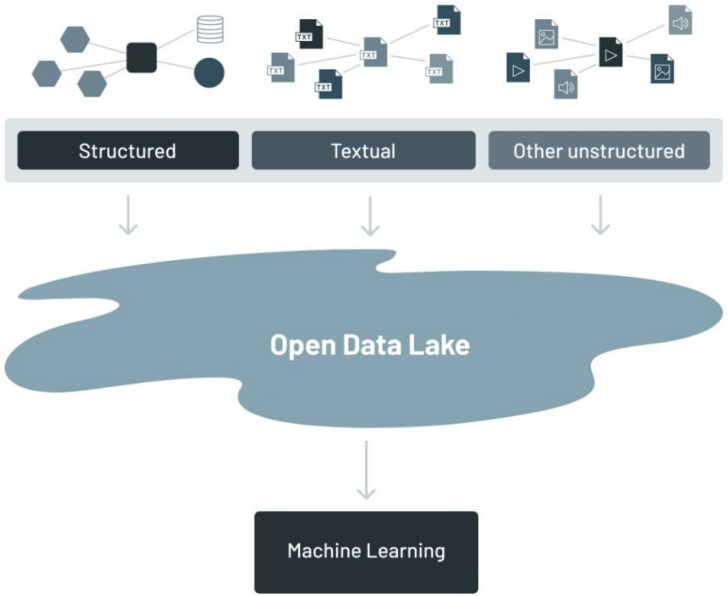
- System or repository of data stored in its natural/raw format and can include:
    - structured data from relational databases (rows and columns),
    - semi-structured data (CSV, logs, XML, JSON),
    - unstructured data (emails, documents, PDFs, images, audio, video)
- Introduced to overcome typical data marts problems, such as information siloing



Source: Inmon (2021)

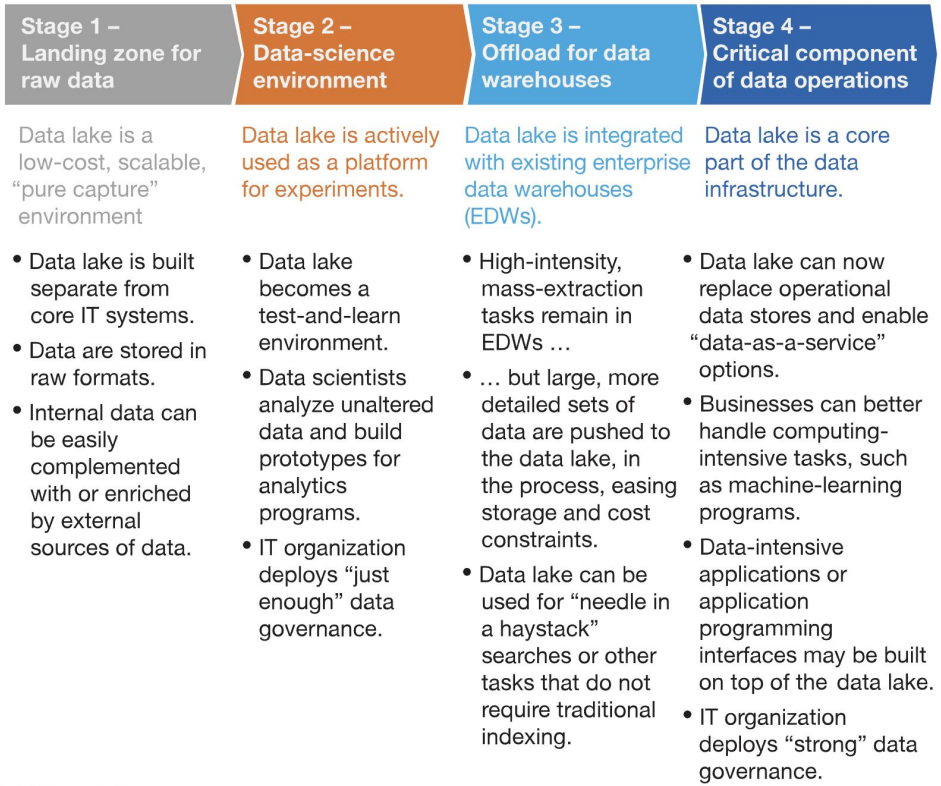
Data Lake

- Typically used in conjunction with traditional enterprise data warehouses (EDWs),
- In general, they cost less to operate than EDWs



Source: McKinsey (2017)

Companies may go through any or all of these four stages of building and integrating data lakes within technology architectures.



McKinsey&Company

Source: McKinsey (2017)

Problems

Many of the promises of the data lakes have not been realized due to the lack of some critical features:

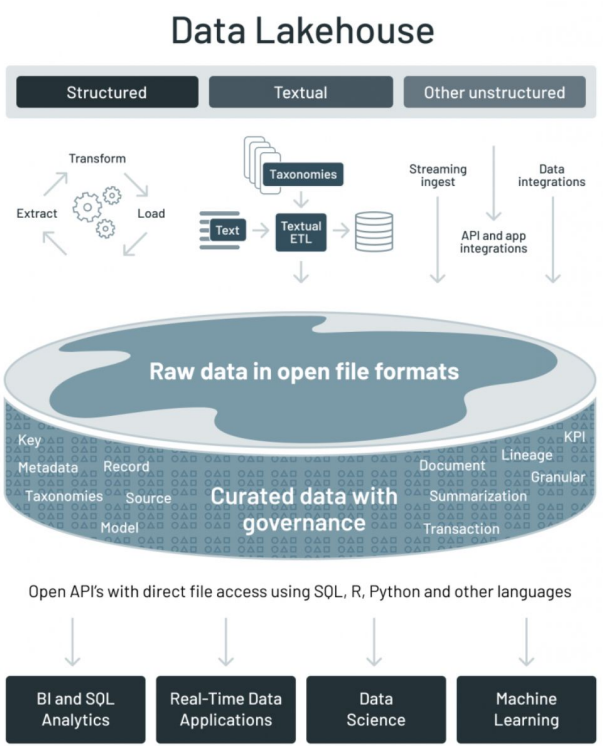
- no support for transactions,
- no enforcement of data quality or governance
- poor performance optimizations.
- As a result, most of the data lakes in the enterprise have become data swamps.

According to McKinsey, a data lake should be viewed as a service model for delivering business value within the enterprise, not a technology outcome.

Source: Inmon (2021)

Data lakehouse

- Implements similar data structures and data management features to those in a data warehouse
- Is based on the kind of low cost storage used for data lakes.
- All types: Structured data, semi-structured data, textual data, unstructured (raw) data
- High quality, reliable data with ACID transactions



Source: Inmon (2021)

	Data warehouse	Data lake	Data lakehouse
Data format	Closed, proprietary format	Open format	Open format
Types of data	Structured data, with limited support for semi-structured data	All types: Structured data, semi-structured data, textual data, unstructured (raw) data	All types: Structured data, semi-structured data, textual data, unstructured (raw) data
Datenzugriff	SQL-only, no direct access to file	Open APIs for direct access to files with SQL, R, Python and other languages	Open APIs for direct access to files with SQL, R, Python and other languages
Reliability	High quality, reliable data with ACID transactions	Low quality, data swamp	High quality, reliable data with ACID transactions
Governance and security	Fine-grained security and governance for row/columnar level for tables	Poor governance as security needs to be applied to files	Fine-grained security and governance for row/columnar level for tables
Performance	High	Low	High
Scalability	Scaling becomes exponentially more expensive	Scales to hold any amount of data at low cost, regardless of type	Scales to hold any amount of data at low cost, regardless of type
Use case support	Limited to BI, SQL applications and decision support	Limited to machine learning	One data architecture for BI, SQL and machine learning

Summary of architecture action plan

Architecture	Deliverables
Data	• Define what data is needed to meet business user needs. • Examine the completeness and correctness of source systems that are needed to obtain data. • Identify the data facts and dimensions. • Define the logical data models. • Establish preliminary aggregation plan.
Information	• Define the framework for the transformation of data into information from the source systems to information used by the business users. • Recommend the data stages necessary for data transform and information access. • Develop source-to-target data mapping for each data stage. • Review data quality procedures and reconciliation techniques. • Define the physical data models.
Technology	• Define technical functionality used to build a data warehousing and business intelligence environment. • Identify available technologies available and review trade-offs associated between any overlapping or competing technologies. • Review the current technical environment and company's strategic technical directions. • Recommend technologies to be used to meet your business requirements and implementation plan.
Product	• List product categories needed to implement the technology architecture. • Review trade-offs between overlapping or competing product categories. • Outline implementation of product architecture in stages. • Identify a short list of products in each of these categories. • Recommend products and implementation schedule.

Source: Sherman (2014)