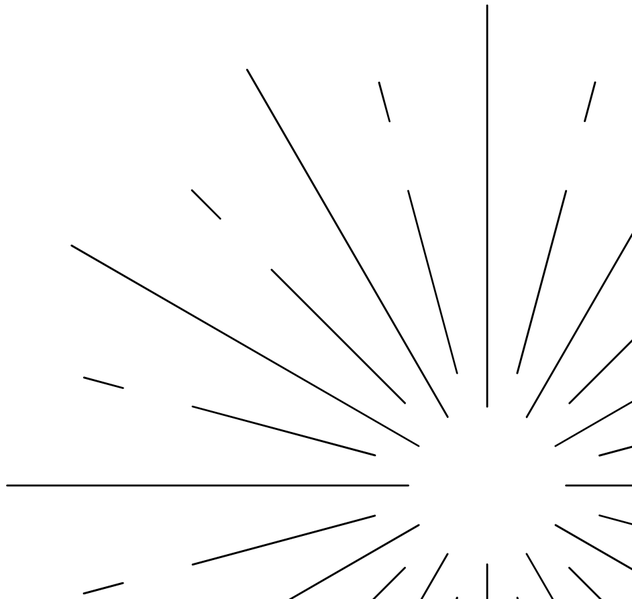
Data Lake Fundamentals

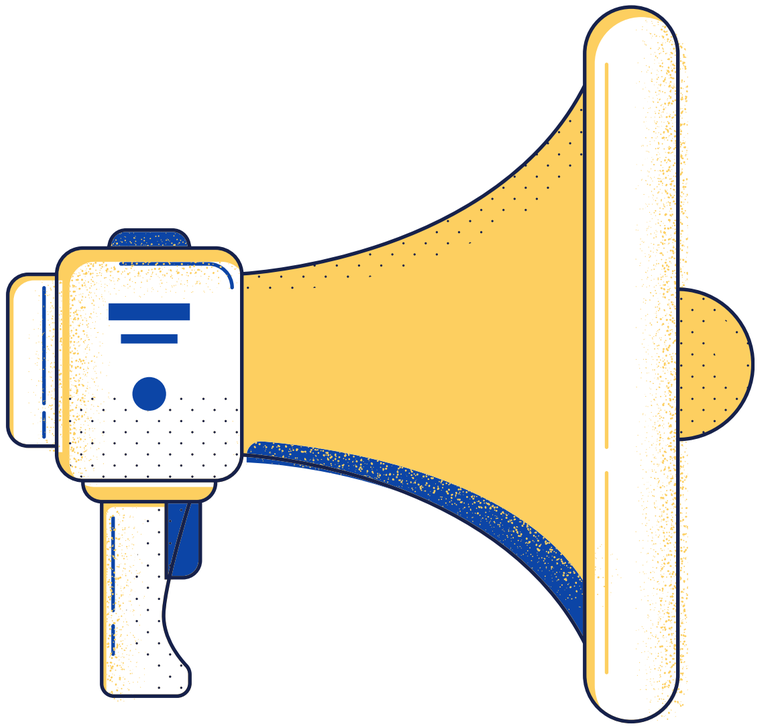
Created for Udemy by George Smarts

What will you learn in this course?

Course Modules

Module 1	Data Lake Basics
Module 2	Data Lake Architecture
Module 3	Evaluating Data Lake fit
Module 4	Implementing a Data Lake
Module 5	Data Lake tools and technology
Module 6	Use Cases and Case Studies
Module 7	The Future
Module 8	Next Steps and Conclusion





Download the course resources

Main Resource - PDF Presentation file with all lessons

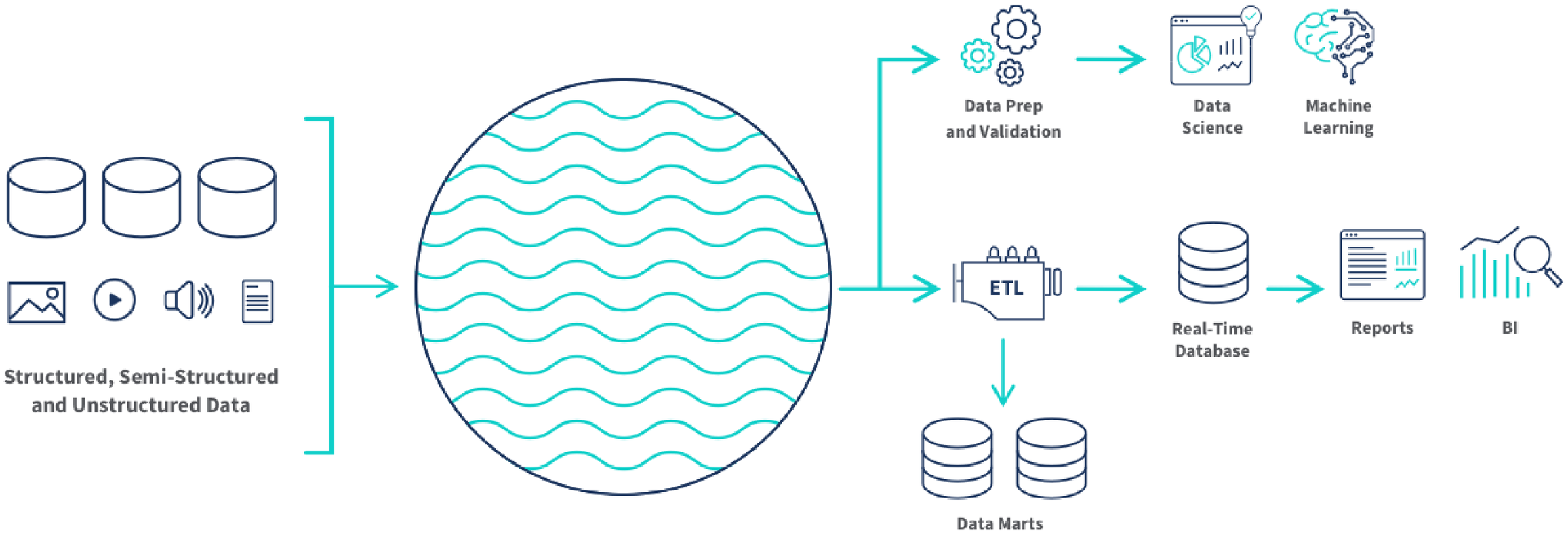
Other resources - additional files/resources provided in various lessons

MODULE 1 - DATA LAKE BASICS

Let's undertand what a Data Lake is!

What is a Data Lake?

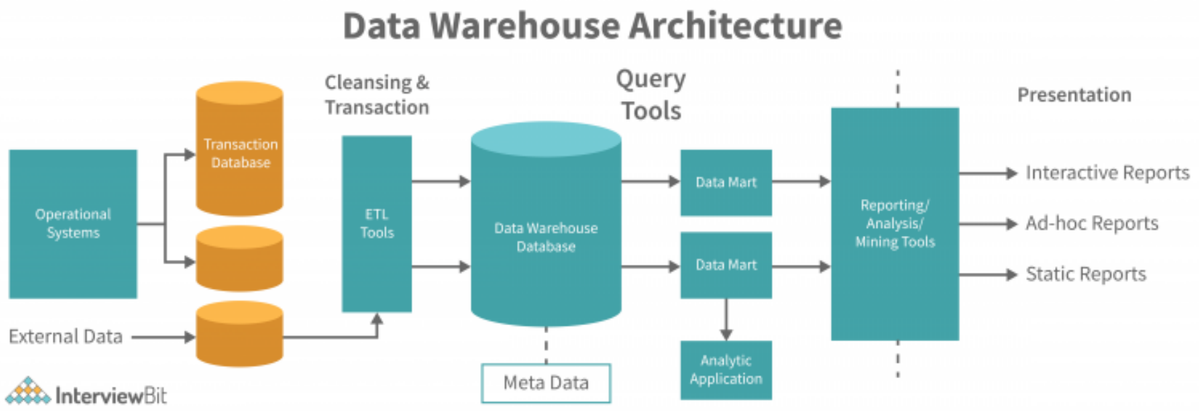
A data lake is a storage repository that holds vast amounts of raw data in its native, unprocessed form until it is needed. It provides a flexible and cost-effective solution for storing diverse types of data, such as structured, semi-structured, and unstructured data, without the need for predefined schemas.

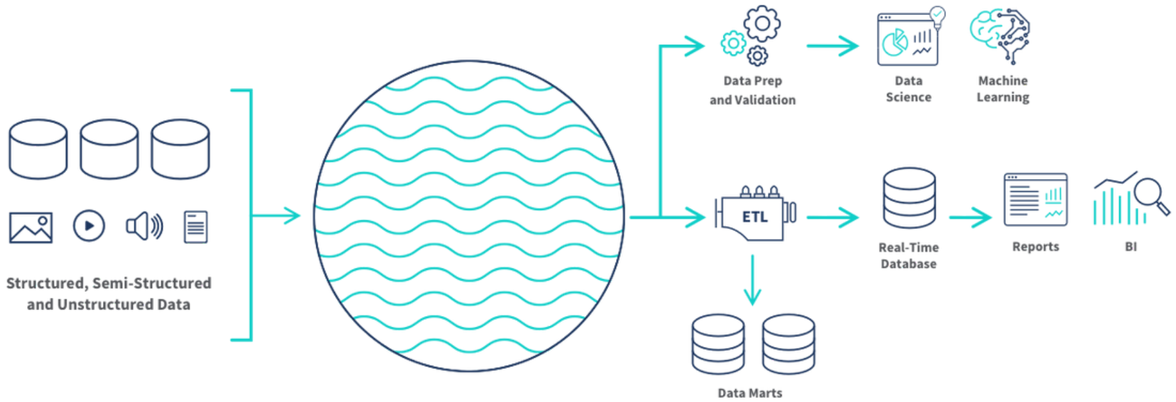


Source of image: https://www.qlik.com/us/data-lake/data-lake-architecture

Full Comparison between Data Warehouse and Data Lake

	Data Warehouse	Data Lake
Definition	Centralized repository for structured data optimized for querying and analysis.	Storage repository that holds raw, unstructured, and structured data at any scale.
Data Types	Primarily structured data.	Accommodates structured, semi-structured, and unstructured data.
Schema	Schema-on-write (data is pre-organized before being ingested).	Schema-on-read (structure is applied at the time of analysis/query).
Processing Speed	High-speed processing for complex queries.	Slower processing due to schema flexibility and handling of diverse data types.
Data Quality	Highly curated and cleansed data.	Raw data ingested, quality control is done at analysis time.
Use Case	Best suited for structured business data and reporting.	Ideal for big data analytics, machine learning, and exploration of diverse data sources.
Storage Cost	More expensive due to pre-processing and indexing.	Typically more cost-effective, especially for large volumes of raw data.

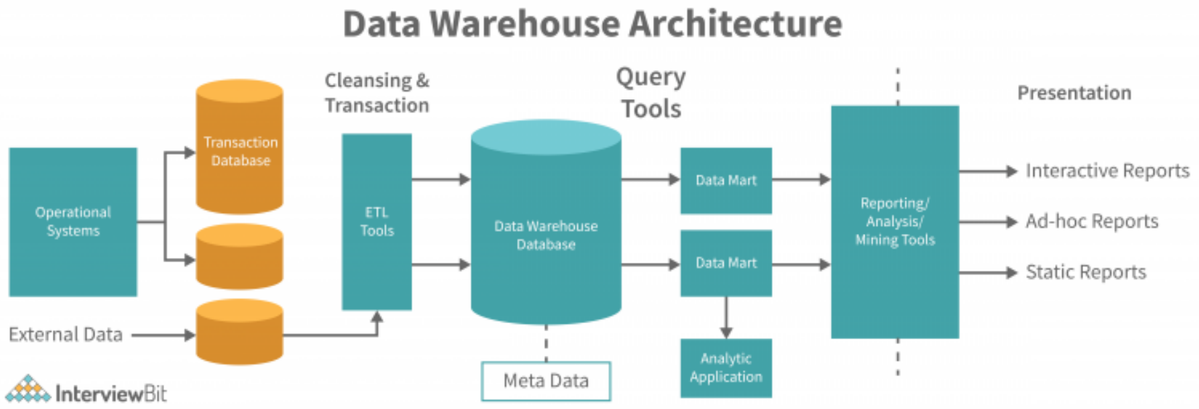


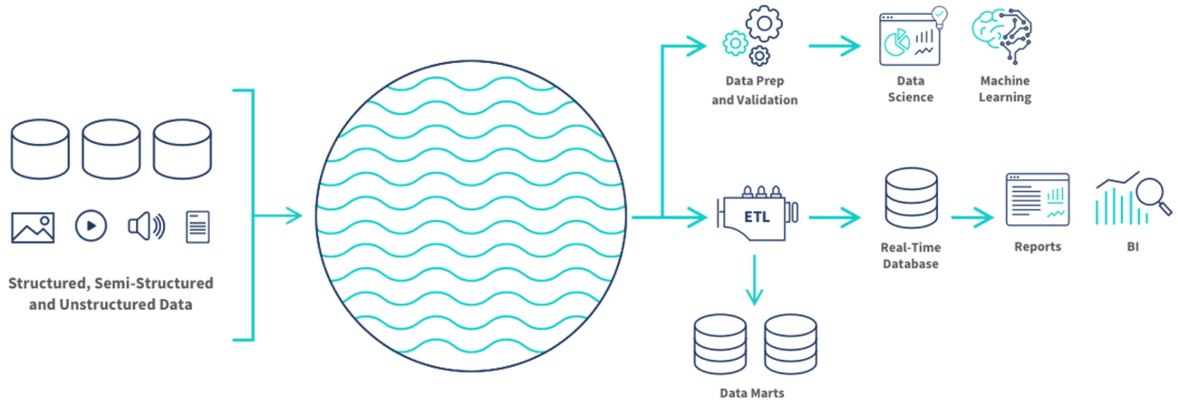


Source of images:
https://www.qlik.com/us/data-lake/data-lake-architecture
https://www.interviewbit.com/blog/data-warehouse-architecture/

Full Comparison between Data Warehouse and Data Lake

	Data Warehouse	Data Lake
Scalability	Scalability can be challenging and may require careful planning.	Highly scalable, able to handle vast amounts of data with ease.
Data Processing Tools	Relational database systems (e.g., SQL-based).	Distributed processing frameworks (e.g., Hadoop, Spark) and various analytics tools.
User Skill Requirements	SQL and data modeling skills are often required.	More technical skills are needed for working with diverse data types and processing frameworks.
Data Governance	Strong data governance and control over data access.	Data governance can be more complex due to the diversity and volume of data.
Use of Metadata	Extensive use of metadata for data organization and management.	Metadata is crucial for discovering, understanding, and managing diverse data sources.
Query Performance	High query performance for structured data.	Query performance may vary based on the schema-on-read approach and the nature of the data.

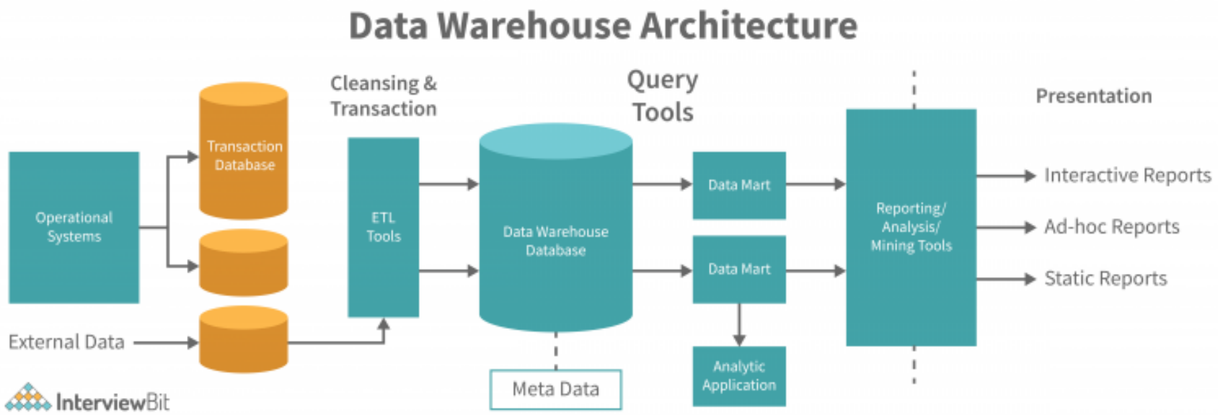




Source of images:
https://www.qlik.com/us/data-lake/data-lake-architecture
https://www.interviewbit.com/blog/data-warehouse-architecture/

Challenges of traditional Data Warehouse approach

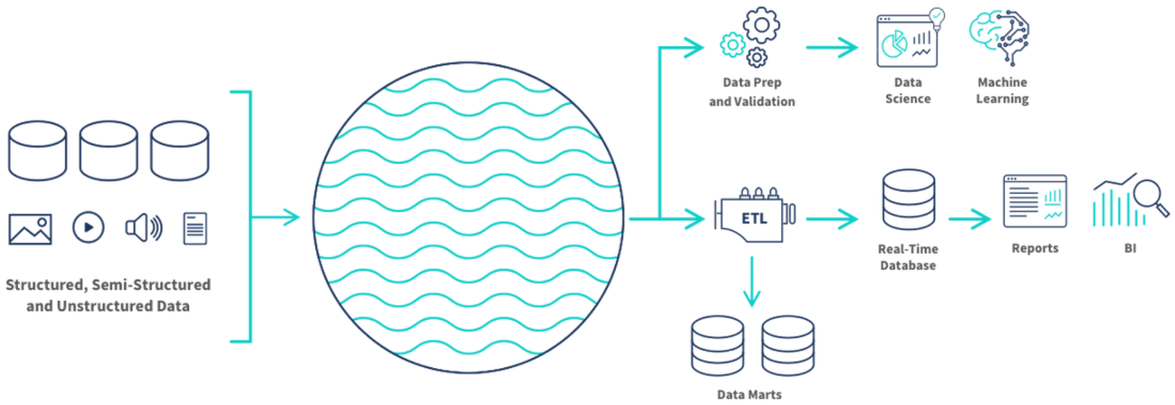
Challenge	What is the Issue?
Schema Flexibility	Traditional data warehouses require a predefined schema before data is loaded. This can be limiting when dealing with diverse and unstructured data.
Data Variety	Data warehouses are typically designed for structured data, making it challenging to handle unstructured or semi-structured data types.
Data Processing Speed	Traditional data warehouses may struggle with real-time processing, especially when dealing with large datasets.
Cost of Storage	Storing large volumes of raw data in a traditional data warehouse can be expensive due to the structured storage requirements.
Data Integration	Integrating data from different sources into a data warehouse can be time-consuming and complex, particularly when dealing with disparate data formats.



Source of Image - https://www.interviewbit.com/blog/data-warehouse-architecture/

How Data Lakes solves these challenges

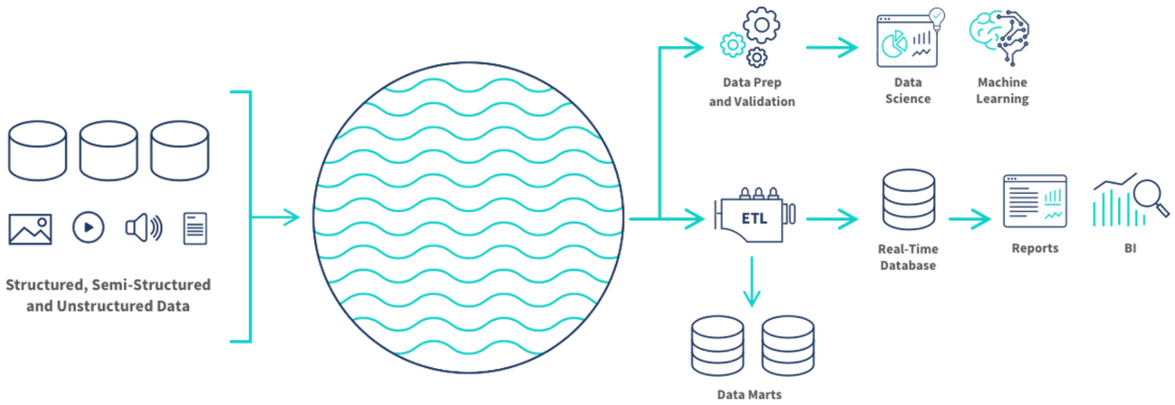
Challenge	Resolution
Schema Flexibility	Data Lakes allow for schema-on-read, meaning data can be ingested without a predefined schema. This flexibility makes it easier to handle diverse and evolving data types.
Data Variety	Data Lakes can store data in its raw, native format, including structured, semi-structured, and unstructured data. This versatility enables organizations to store and process a wide variety of data.
Data Processing Speed	Data Lakes can leverage technologies like Apache Spark for distributed processing, enabling faster and more scalable data processing.
Cost of Storage	Data Lakes use cost-effective storage solutions, like Hadoop Distributed File System (HDFS) or cloud-based object storage, which can be more economical for storing vast amounts of raw data.
Data Integration	Data Lakes provide a unified repository for raw data, simplifying the integration process. Data can be ingested in its raw form and transformed as needed during analysis.



Source of images: https://www.qlik.com/us/data-lake/data-lake-architecture

Limitations of Data Lakes

Challenge	What is the Issue?
Data Consistency and Structure	In Data Lakes, the lack of predefined schemas may lead to inconsistencies in data quality and structure, making it more challenging to maintain a uniform view of the data.
Schema-on-Read Complexity	While the schema-on-read approach in Data Lakes provides flexibility, it can also introduce complexity during the data exploration and analysis phase. Users need to understand and interpret the data at the time of querying, which can be challenging.
Query Performance	Data Warehouses are optimized for query performance, especially for complex analytical queries involving structured data. Data Lakes, on the other hand, may face challenges in providing the same level of performance, particularly when dealing with large volumes of raw and unstructured data.
Data Processing Overhead	The flexibility of storing raw data in Data Lakes comes with the cost of increased processing overhead during query execution. Extracting meaningful insights from raw data often requires additional processing steps compared to pre-processed data in Data Warehouses.
Ease of Use for Business Users	Data Lakes, with their raw and diverse data formats, may be less user-friendly for business users who are not familiar with the underlying technicalities.



Source of images: https://www.qlik.com/us/data-lake/data-lake-architecture

MODULE 2 - DATA LAKE ARCHITECTURE

Let's undertand the Data Lake Architecture

Data Lake Architecture

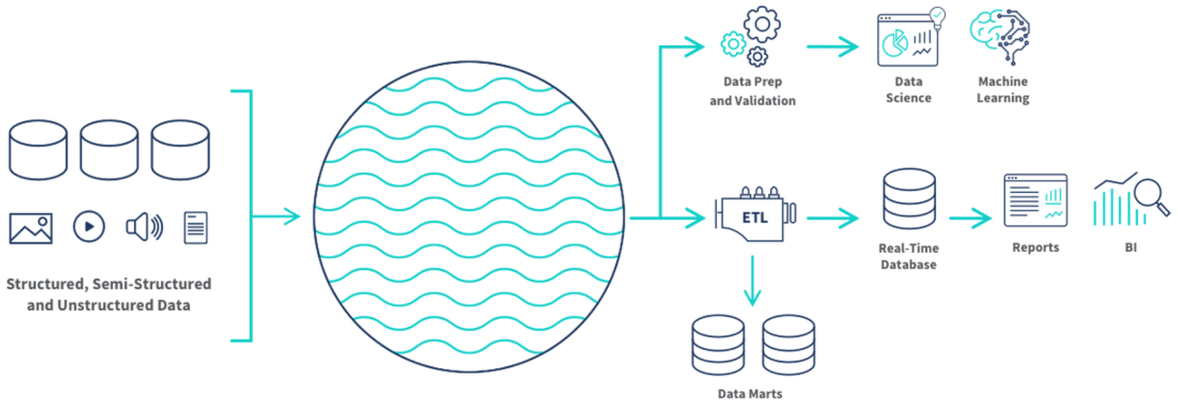
The 8 Key Components

1. Data Sources
2. Data Ingestion Layer
3. Data Storage Layer
4. Metadata Management and Cataloging
5. Data Processing and Analytics Layer
6. Data Governance and Security
7. Data Presentation Layer
8. Monitoring and Management

Data Lake Architecture

1. Data Sources

Description	Data sources are the origin points of data that feed into the Data Lake. They can be internal or external, structured or unstructured, and may include various types of data such as logs, databases, sensor data, and more.
Examples	• Relational databases (MySQL, Oracle) • Log files (Apache logs, server logs) • IoT devices (sensor data) • Cloud applications (Salesforce, Google Analytics)
Best Practices	• Identify and catalog all potential data sources. • Understand the data schema, format, and update frequency for each source. • Establish data acquisition methods (batch, streaming, or hybrid) based on source characteristics.
Use Cases	• Ingesting customer transaction data from a relational database. • Streaming real-time data from IoT devices for analysis.



Source of images: https://www.qlik.com/us/data-lake/data-lake-architecture

Data Lake Architecture

2. Data Ingestion Layer

Description

The Data Ingestion layer is responsible for collecting and transporting data from various sources into the Data Lake. It involves the processes and tools for acquiring, validating, and loading data.

Examples

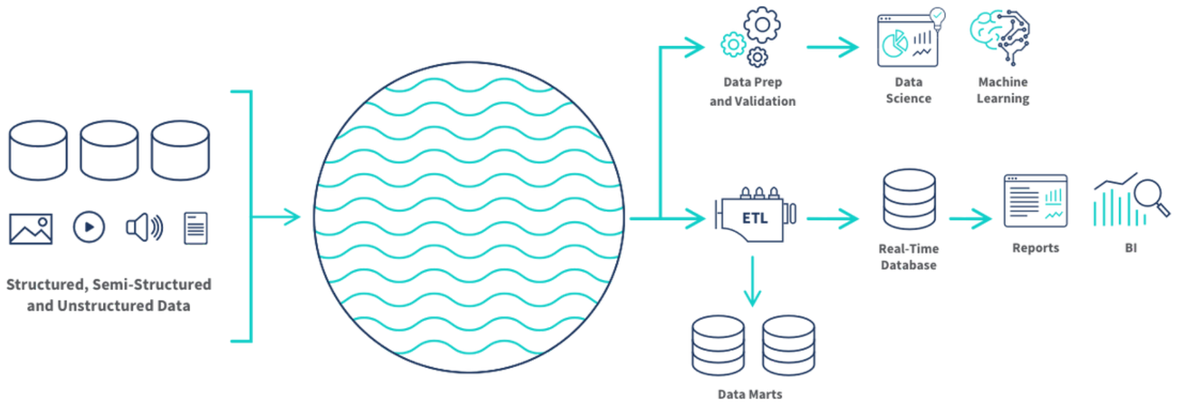
- Apache NiFi
- Apache Kafka
- AWS Glue

Best Practices

- Choose ingestion methods based on the nature of the data source.
- Ensure data integrity during ingestion through validation.
- Implement error handling and retries for robustness.

Use Cases

- Batch ingestion of log files from web servers.
- Real-time ingestion of social media streaming data.



Source of images: https://www.qlik.com/us/data-lake/data-lake-architecture

Data Lake Architecture

3. Data Storage Layer

Description

The storage layer is the foundational component where raw and processed data is stored in a scalable and cost-effective manner.

Examples

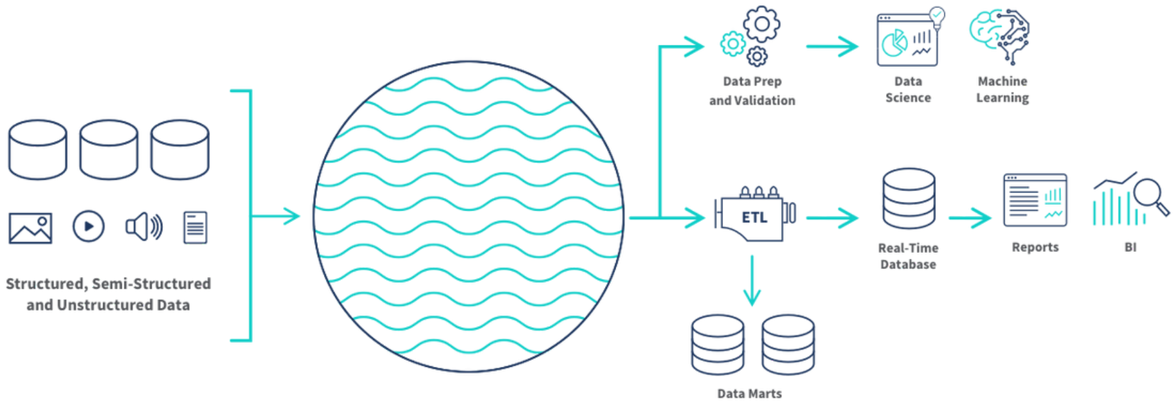
- Amazon S3
- Azure Data Lake Storage
- Google Cloud Storage

Best Practices

- Implement data partitioning for improved query performance.
- Use a tiered storage strategy for cost optimization.

Use Cases

- Storing large volumes of structured and unstructured data.
- Archiving historical data for compliance and analysis.



Source of images: https://www.qlik.com/us/data-lake/data-lake-architecture

Data Lake Architecture

4. Metadata Management and Cataloging

Description

Metadata management and cataloging involve organizing and storing metadata to provide insights into the data stored in the data lake. This includes information about data lineage, schema, quality, and other relevant attributes.

Examples

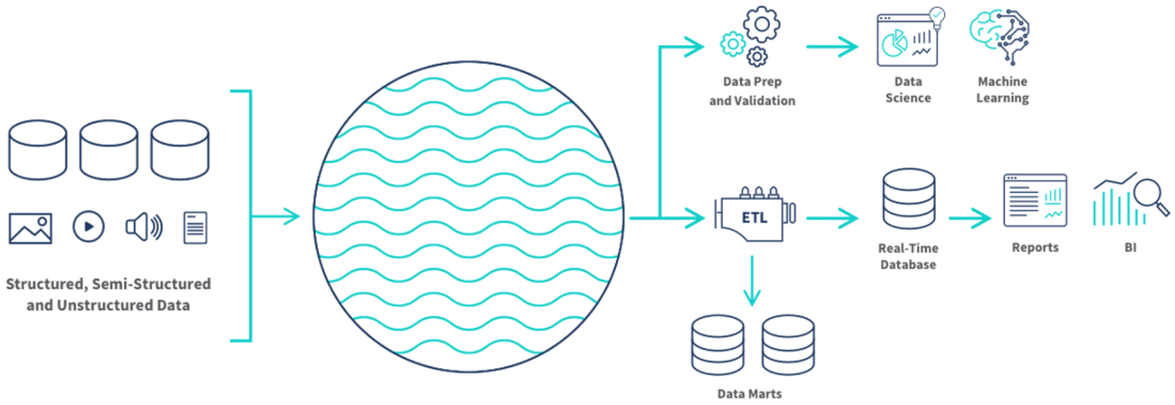
- Apache Atlas
- AWS Glue Data Catalog
- Azure Purview.

Best Practices

- Maintain a comprehensive metadata catalog that includes lineage and quality information.
- Automate metadata capture to reduce manual efforts and ensure consistency.

Use Cases

- Ensuring data lineage for regulatory compliance.
- Facilitating data discovery and understanding.
- Improving collaboration among data users by providing a centralized view of available data assets.

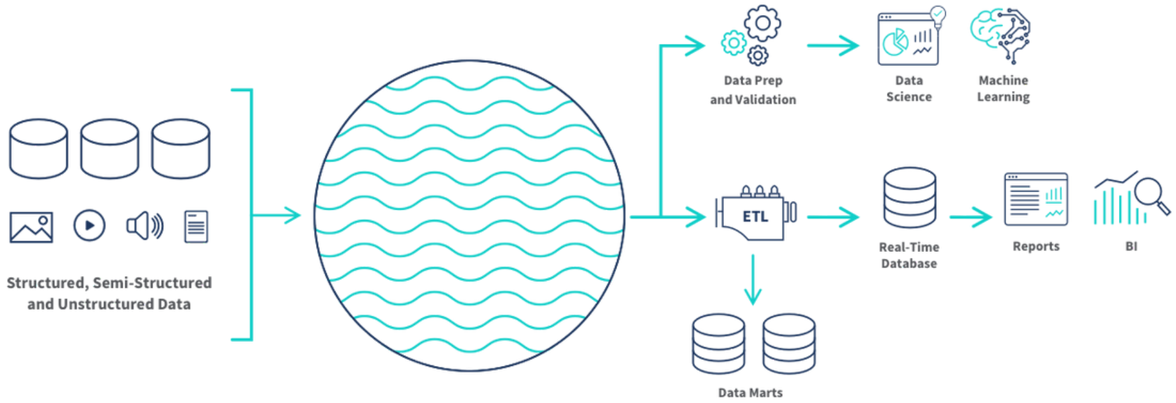


Source of images: https://www.qlik.com/us/data-lake/data-lake-architecture

Data Lake Architecture

5. Data Processing and Analytics Layer

Description	The data processing and analytics layer involves the execution of data processing tasks and analytics on the data stored in the data lake. This includes transforming raw data into usable formats and enabling users to derive insights through various analytics approaches.
Examples	• Data Processing: Apache Spark, Apache Flink. • Analytics: Apache Hive, Presto, Amazon Athena, Azure Synapse Analytics.
Best Practices	• Implement ETL (Extract, Transform, Load) pipelines for data transformation. • Support multiple query engines to provide flexibility for different analytical tasks.
Use Cases	• Transforming and aggregating data for analytics and reporting. • Conducting ad-hoc queries for business intelligence. • Performing advanced analytics, such as machine learning, on the data lake. • Enabling real-time analytics on streaming data.

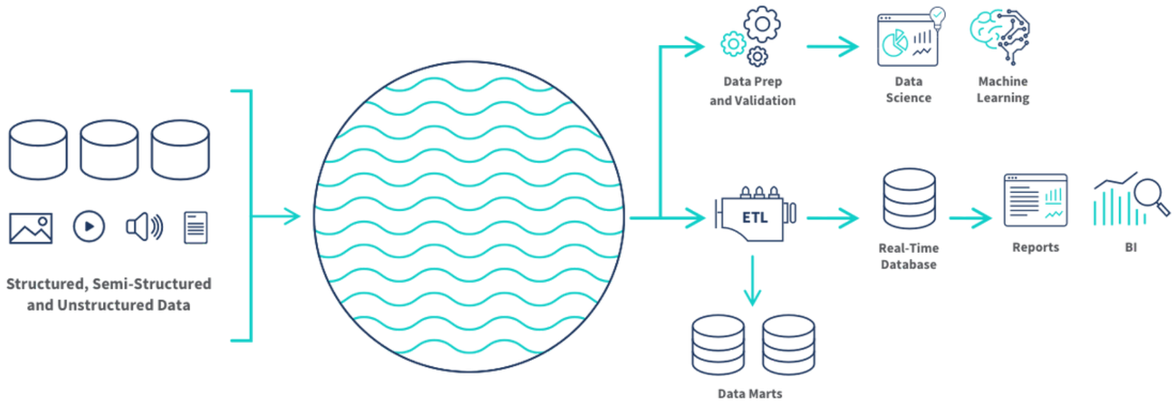


Source of images: https://www.qlik.com/us/data-lake/data-lake-architecture

Data Lake Architecture

6. Data Governance and Security

Description	This layer ensures that the data in the Data Lake is secure, compliant, and properly governed.
Examples	• AWS Glue DataBrew • Apache Ranger • Collibra
Best Practices	• Communicate Data Lake governance policies comprehensively. • Appoint stewards overseeing data quality and adherence to policies. • Periodic assessments of Data Lake for compliance and security.
Use Cases	• Access Control: Restrict and manage access to specific Data Lake datasets. • Data Classification: Label data within the Data Lake based on sensitivity. • Audit and Monitoring: Track user activities, changes, and access patterns in the Data Lake. • Compliance Management: Ensure Data Lake practices align with industry regulations and internal policies.

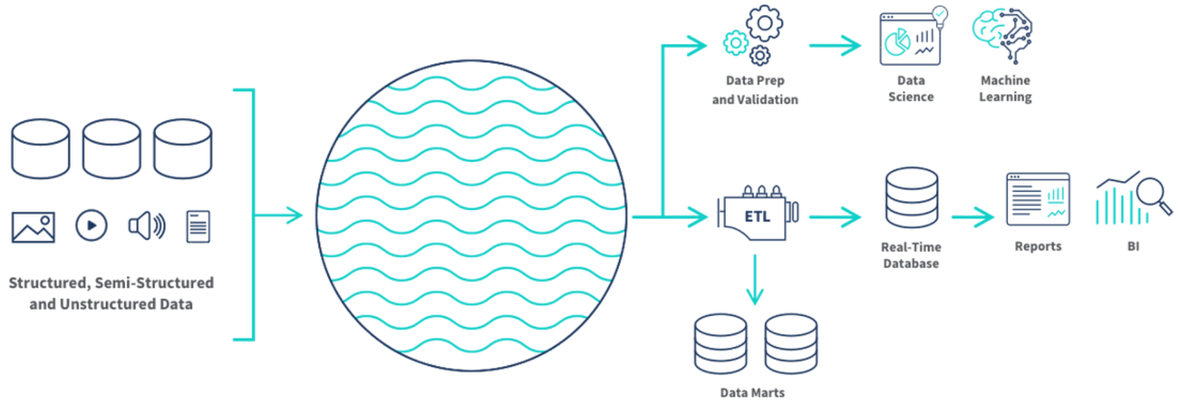


Source of images: https://www.qlik.com/us/data-lake/data-lake-architecture

Data Lake Architecture

7. Data Presentation Layer

Description	The data presentation layer involves providing tools and interfaces for users to interact with and derive insights from the data stored in the Data Lake. This layer focuses on enabling effective data discovery, visualization, and analysis for end-users.
Examples	• Tableau • Power BI • Looker
Best Practices	• Integrate with popular Business Intelligence (BI) tools for data visualization and analysis. • Provide user-friendly interfaces for data exploration and ad-hoc querying. • Ensure responsiveness and performance in delivering query results.
Use Cases	• Ad-hoc analysis: Empower users to perform on-the-fly analysis of data stored in the Data Lake. • Dashboard creation: Enable the development of dashboards for monitoring and reporting. • Self-service analytics: Facilitate business users in deriving insights without extensive technical knowledge.

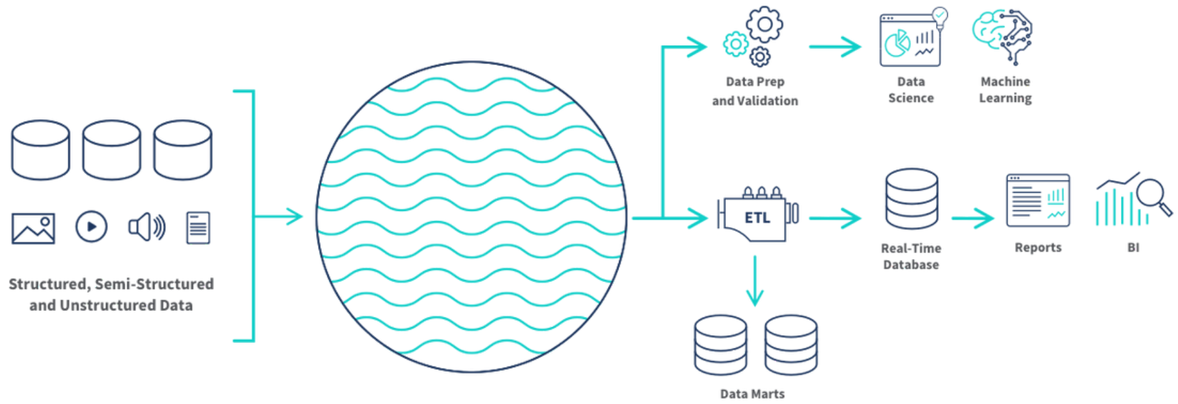


Source of images: https://www.qlik.com/us/data-lake/data-lake-architecture

Data Lake Architecture

8. Monitoring and Management

Description	Monitoring and management involve overseeing the performance, health, and usage of the Data Lake infrastructure, processes, and components. It includes proactive monitoring, troubleshooting, and resource optimization.
Examples	• Monitoring Tools: Prometheus, Grafana, Nagios. • Logging Frameworks: ELK Stack (Elasticsearch, Logstash, Kibana), Splunk. • Cloud-native Services: AWS CloudWatch, Azure Monitor, Google Cloud Operations Suite.
Best Practices	• Implement real-time monitoring to detect and address issues promptly. • Utilize logging and auditing mechanisms for tracking events, changes, and errors. • Set up alerts and notifications for critical events and performance thresholds. • Conduct regular performance tuning and optimization based on monitoring insights.
Use Cases	• Identifying and addressing performance bottlenecks in data processing. • Investigating and resolving issues related to data quality or availability. • Ensuring resource utilization efficiency and optimizing costs. • Monitoring security events and responding to potential threats.



Source of images: https://www.qlik.com/us/data-lake/data-lake-architecture

Module 3 - Evaluating Data Lake fit

How to evaluate if this is for you?

Is Data Lake for your company?

Category	Questions	Example Answers from a Tech Company	Analysis
Business Goals	- What are the key business objectives and goals of the company?	"Our main goal is to enhance customer experience through personalized services and optimize operational efficiency."	A data lake could better sup flexible and scalable platfo
	- How does the company use data to drive decision-making and achieve its objectives?	"We leverage data for predictive analytics to make informed decisions and stay ahead of market trends."	A data lake could enhance diverse and comprehensive
Data Volume and Variety	- What is the volume of data that the company generates and collects?	"We handle terabytes of customer interaction data daily, including structured data from transactions and unstructured data from social media."	A data lake could better acc variety of data, providing a solution.
	- What types of data does the company deal with (structured, semi-structured, unstructured)?	"We work with structured customer records, semi-structured log files, and unstructured data from customer feedback."	A data lake could improve h enable more flexible analys
	- Is the data primarily coming from internal sources, external sources, or a mix of both?	"Our data comes from both internal systems like CRM and external sources such as social media and third-party APIs."	A data lake could facilitate external data sources for a r
Data Integration and	- Does the company currently face challenges with data silos and integration?	Yes, we have challenges with disparate systems and data silos across departments, hindering seamless data access."	A data lake could mitigate repository accessible to all t

CHALLENGES WITH IMPLEMENTING DATA LAKES

01 Data Quality and Governance

02 Security Concerns

03 Skill Set Requirements

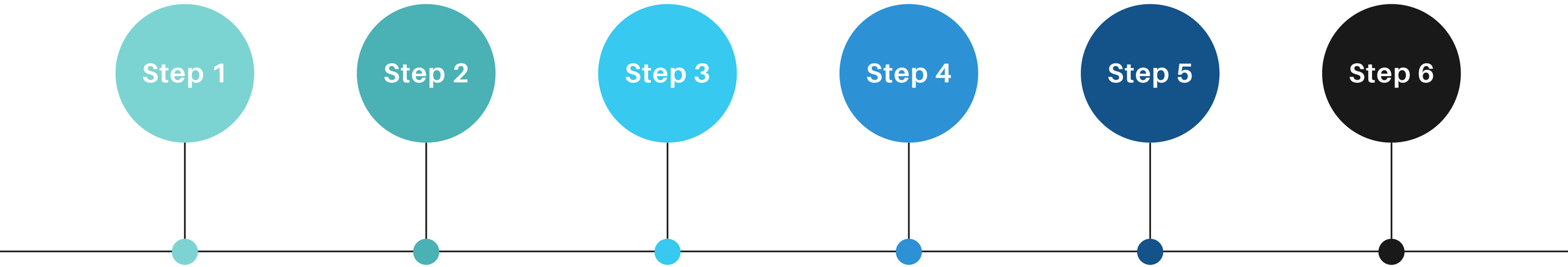
04 Complexity in Tooling and Integration

Module 4 - Implementing a Data Lake

How do we actually implement this? Let's find out!

Implementing Data Lake

Step by Step



Define objectives and use cases

Stakeholder Buy-In

Build a Cross-Functional Team

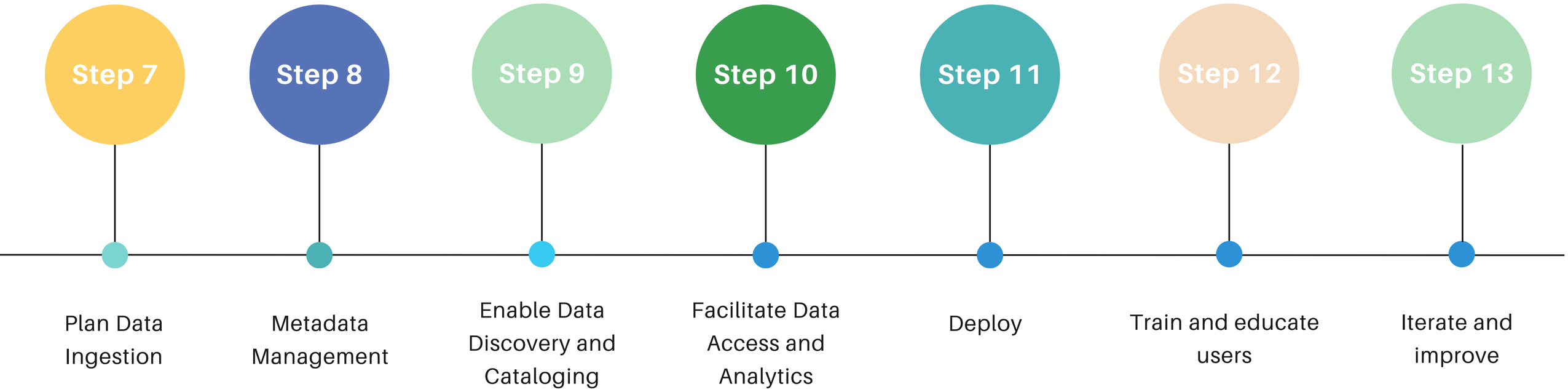
Assess the data sources and specifics

Choose a Data Lake Platform and Architecture

Design Data Governance and Security Policies

Implementing Data Lake

Step by Step

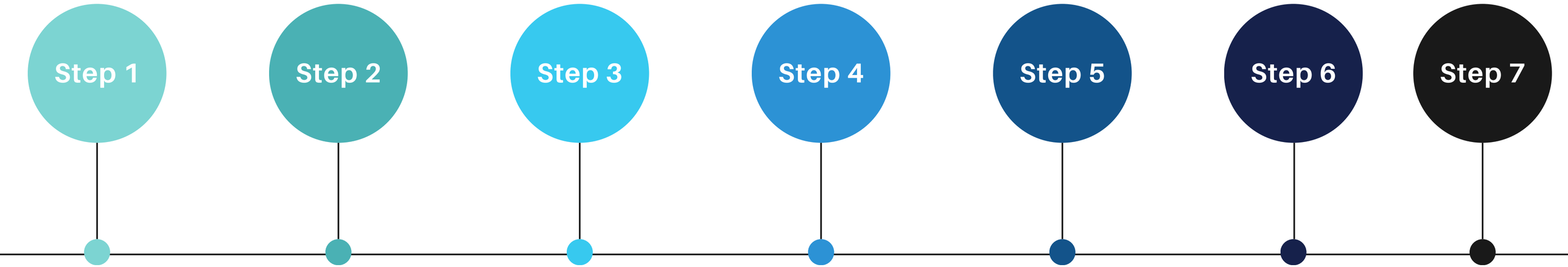


MODULE 5 - Data Lake tools and technologies

What are the best tools out there?
Let's find it!

Choosing the Data Lake storage

Step by Step



Define your requirements

Evaluate features that will fullfil requirements and available solutions

Cost Analysis, Community, Reviews and Reputation

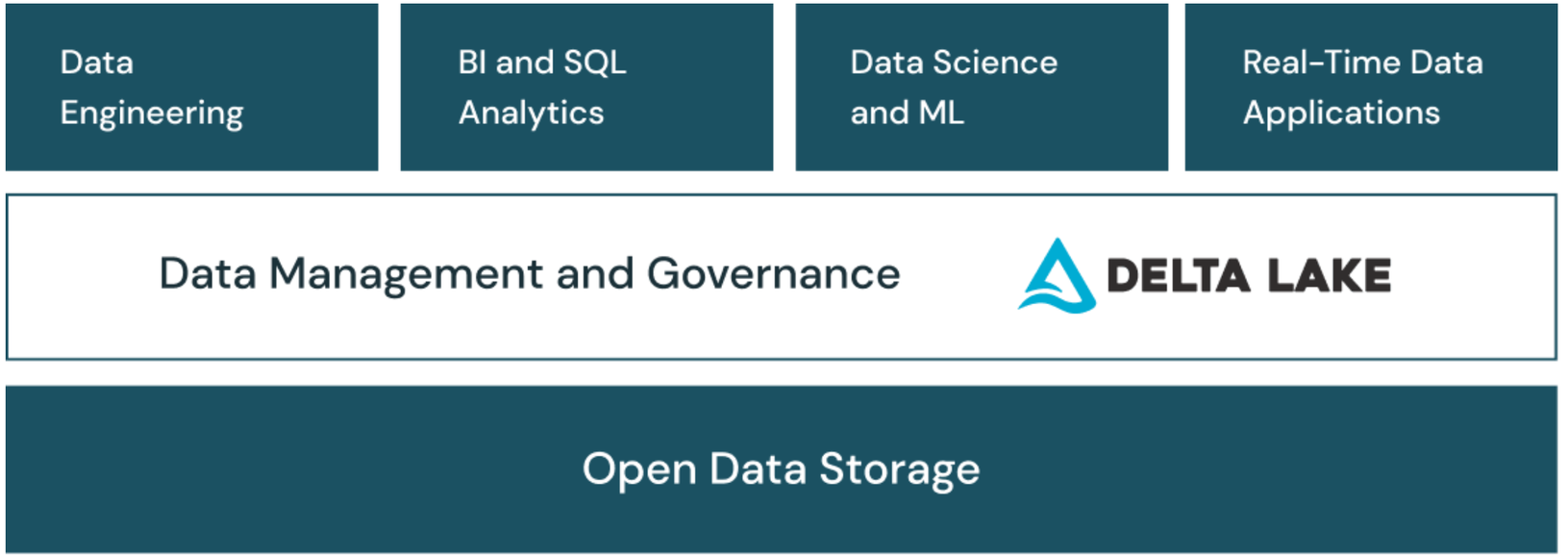
Integration and Compatibility

Check support for metadata, data quality, governance, etc

Trial a few solutions

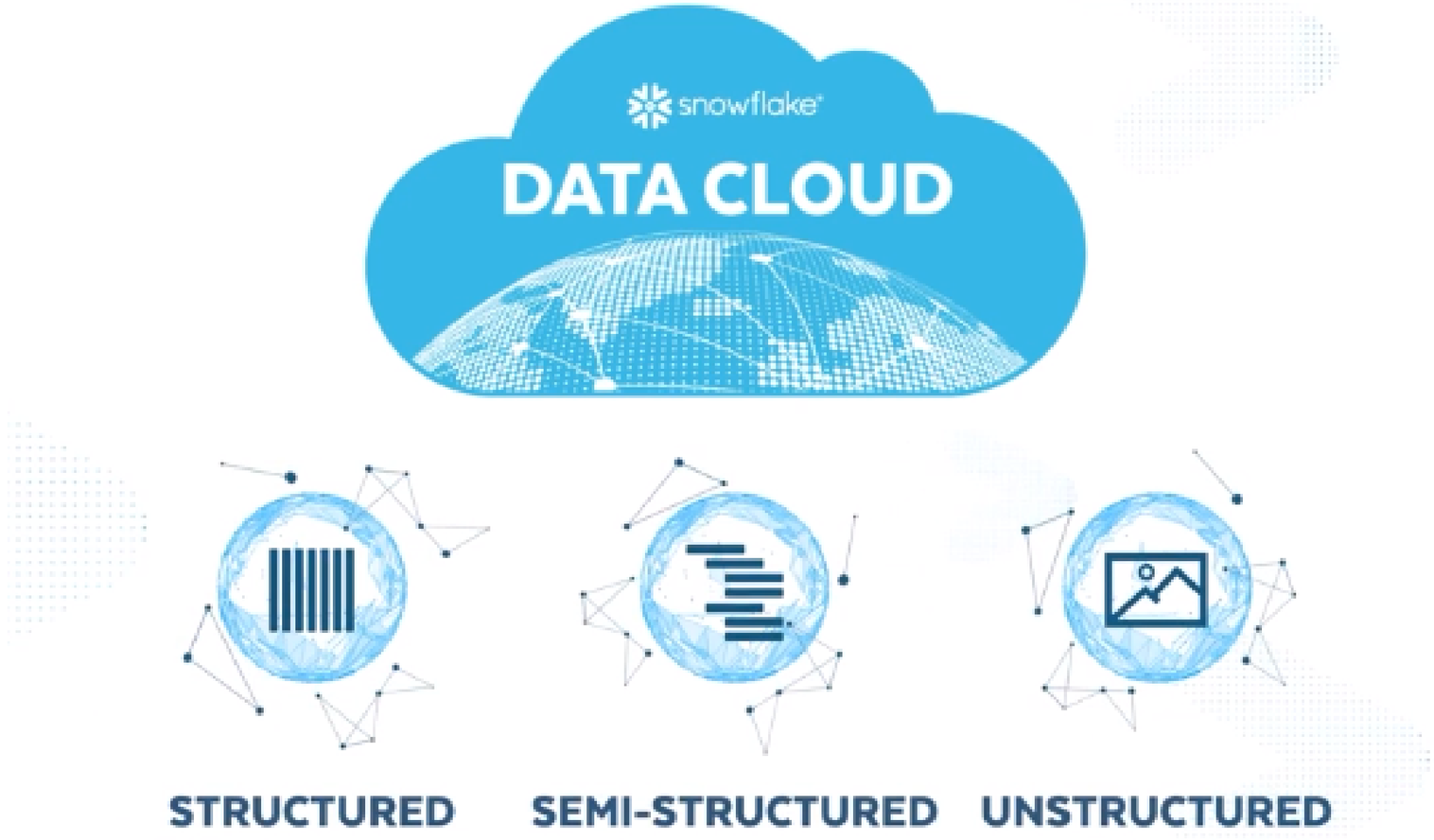
Decide and do a Pilot Implementation

Databricks Delta Lake



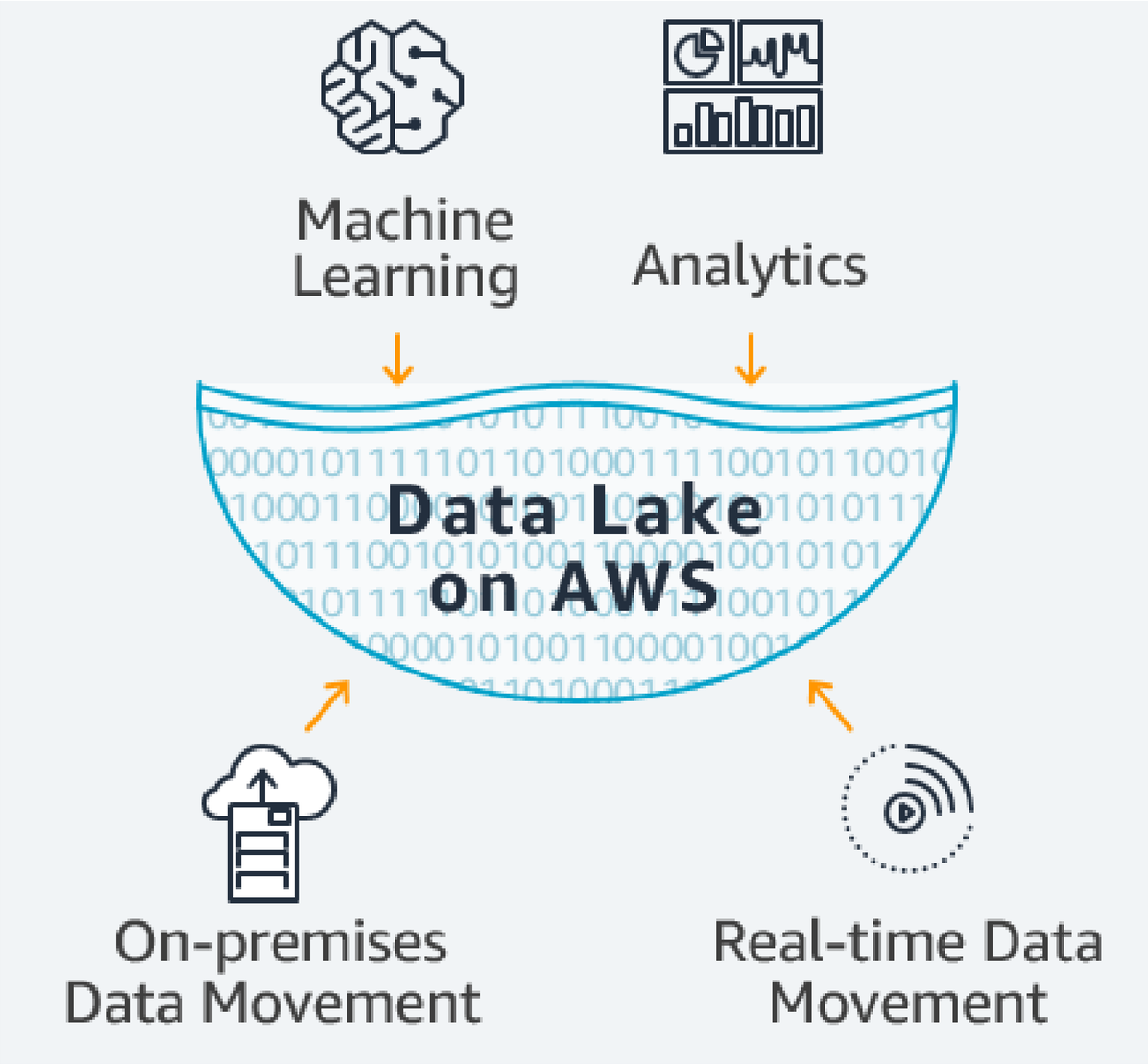
Source:https://www.databricks.com/product/delta-lake-on-databricks

Snowflake Data Cloud



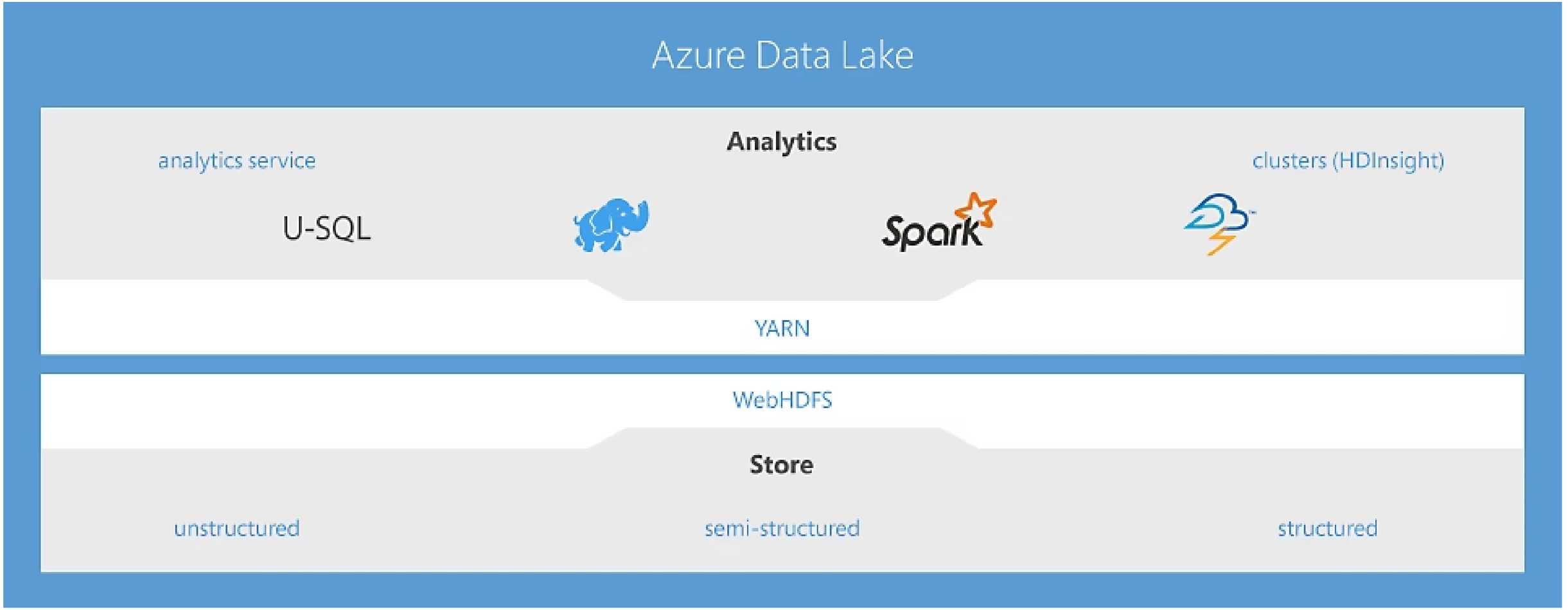
Source: https://www.snowflake.com/en/data-cloud/workloads/data-lake/

Amazon Data Lake



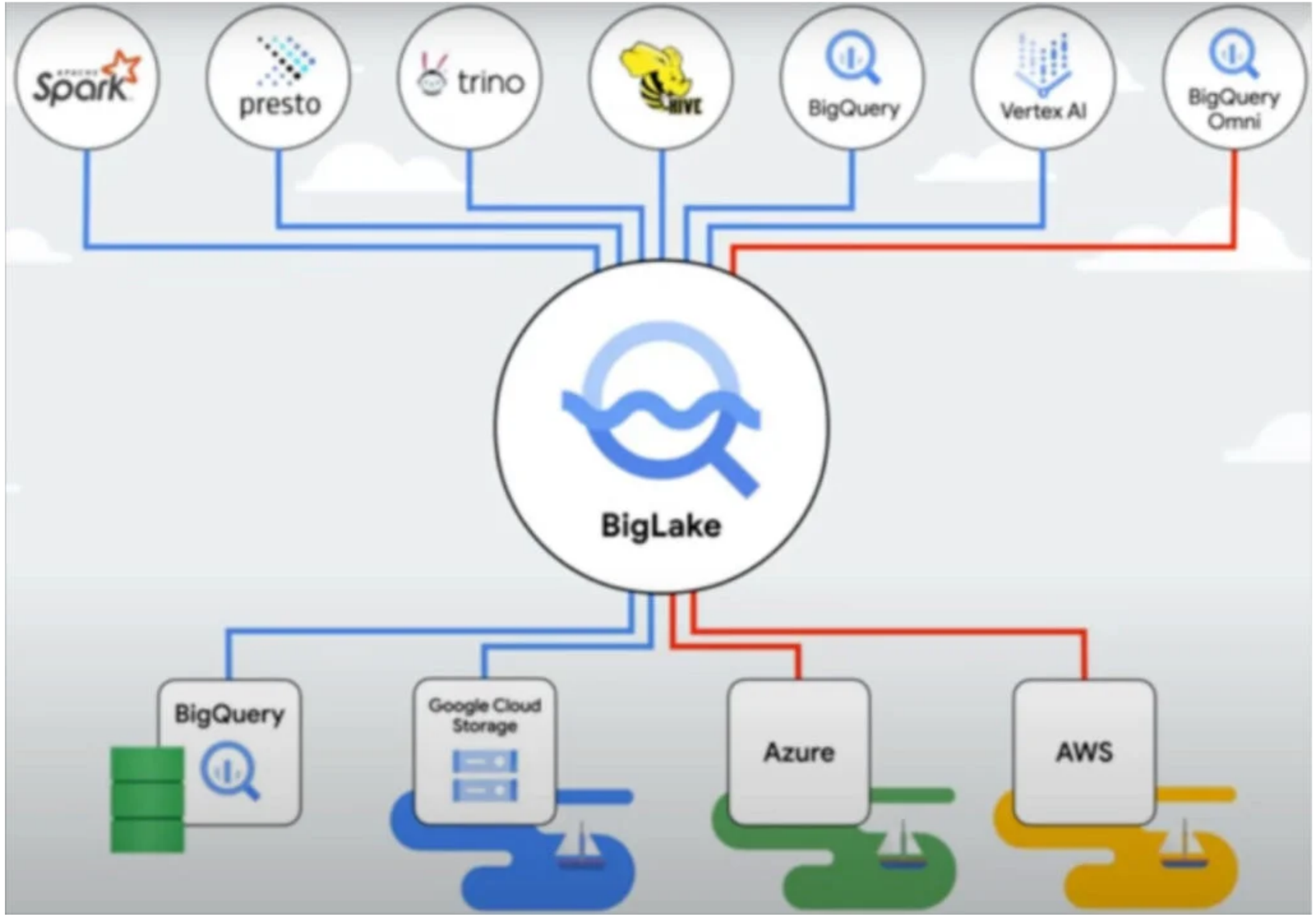
Source: https://aws.amazon.com/lake-formation/

Azure Data Lake



Source: https://aws.amazon.com/lake-formation/

Google’s Data Lake



Source: https://aws.amazon.com/lake-formation/

Module 6 - Use Cases and Case Studies

Who uses Data Lakes and for what?

CASE STUDIES FOR FURTHER READING

01

02

03

Module 7 - The Future

Here are some things to keep in mind

TOP 3 FUTURE TRENDS

01 Shift to Cloud-Native Data Lakes

02 Use of AI/ML for Automation

03 Data Lakehouse

Module 8 - Next Steps and Conclusion

Here are some things to keep in mind

NEXT STEPS

01

Gain knowledge about big data technologies like Hadoop and Apache Spark.

02

Understand how data lakes are implemented on cloud platforms like Amazon S3, Microsoft Azure Data Lake Storage, or Google Cloud Storage

03

Hands-on Projects: Apply your knowledge by working on real-world projects.

04

Engage with the data community through forums, blogs, and social media. Stay informed about the latest developments, case studies, and best practices shared by other data professionals.


THANK YOU