PRESIDIO®

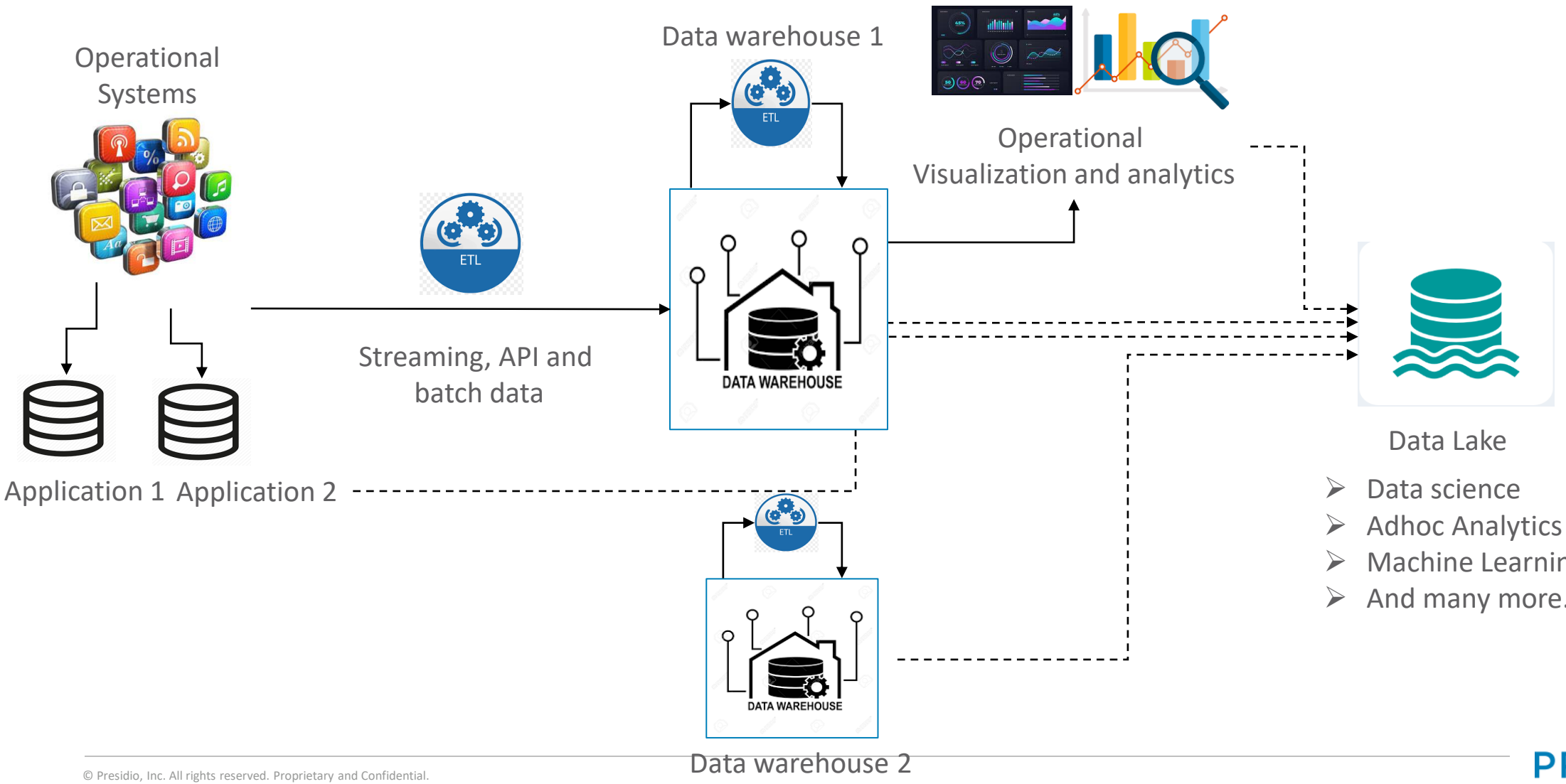
Introduction to Data

Content

- Enterprise Data landscape blueprint
- Database vs Datawarehouse vs DataLake
- DataOps
- Scalable Cloud Data Warehouses
- Azure Synapse Architecture
- Redshift Architecture
- ETL tools – Introduction
- ETL tools
- Orchestration tools
- Sample use cases/Architectures
- Feedback and Questions

PRESIDIO

Enterprise Data landscape Blueprint



PRESIDIO

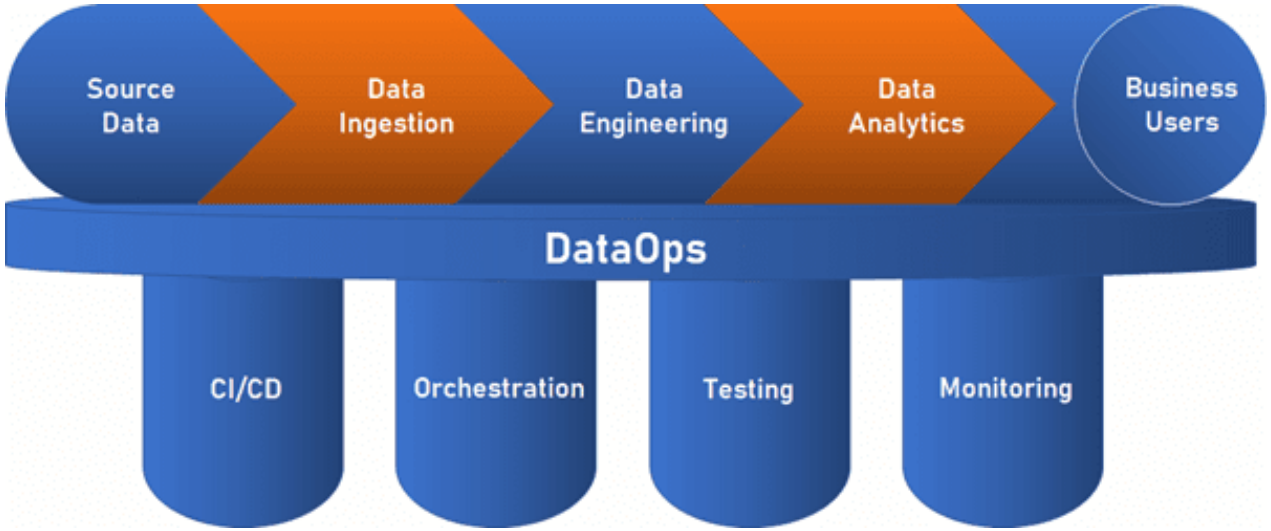
Database vs Data Warehouse vs Data Lake

	Database	Data Warehouse	Data Lake
Purpose	Operational/Transactional systems	Operational Analytical systems	Adhoc Analytics, Machine learning and data science
Data	Structured	Structured	Raw and unstructured
Business Criticality/Operational severity	Very high	Very high to high	Mostly less critical(Varies)
Users	Application users	Business Users/Decision makers	Data scientists and ML Engineers Analytical teams
Security	Closely tied to the application	Rationalized based on the data consumer	Widely available
Schema Flexibility	Rigid	Pre-defined and modelled	Schema on read
Post processing Complexity	Less complex	Prepared data – Less complex	Complex based on the use case
History data	As per application requirement	Historic and current data – Data retention is subjective	Historic and current data – Data retention is subjective

Data Ops

DataOps is an enterprise strategy which is primarily focused on

- Reducing the elapsed time between proposal of new idea and finished analytical pipeline
- Measured by the number of analytical products productionized in a given period of time
- DevOps and Data operations are part of DataOps
- More on DataOps
  https://datakitchen.io/what-is-dataops/



PRESIDIO

Scalable Cloud Data Warehouses

With the power of cloud and limitless infrastructure, data warehouses have become intelligent and scalable based on the workloads

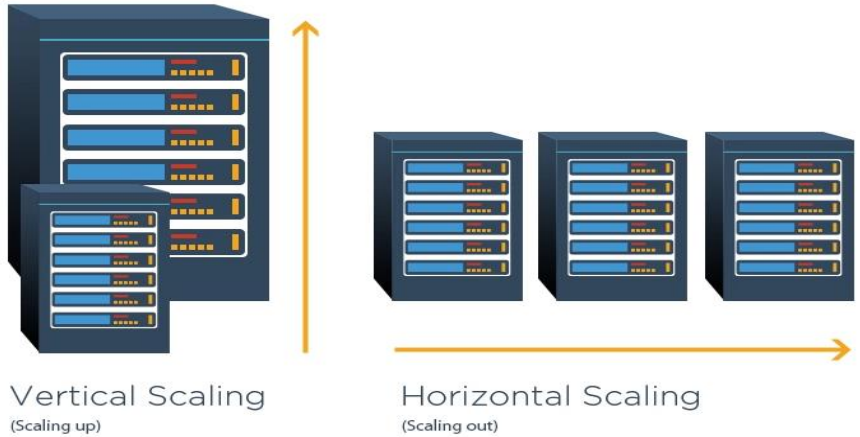
- Scale horizontally based on query workloads
- Support MPP for parallel processing
- Support SQL based query engines
- Variety of connections and compatibilities to Reporting and ETL tools
- Abstract data and underlying infrastructure management to enable quick insights and data driven decision making
- Columnar oriented databases which support OLAP – Read heavy use cases.(Row based architectures are good for OLTP. Transactional systems)
- Fault tolerant with replication and disaster recovery options
- Customizable concurrency and scaling features for effective cost and performance management
- Uses compression for the storage of data, so that the storage costs are optimized

Azure Synapse Analytics

snowflake

amazon REDSHIFT

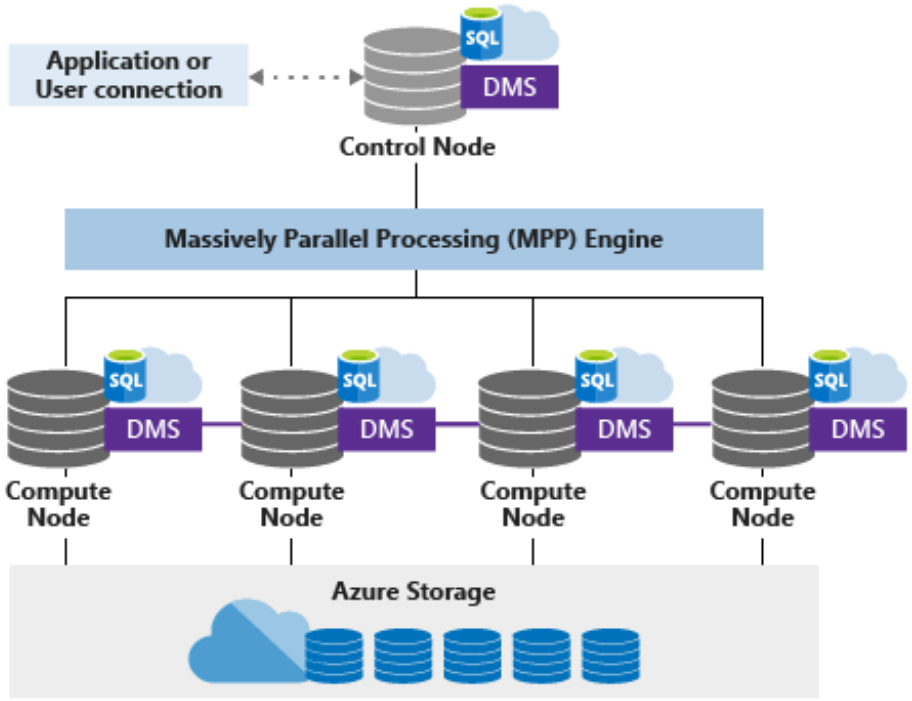
Google BigQuery



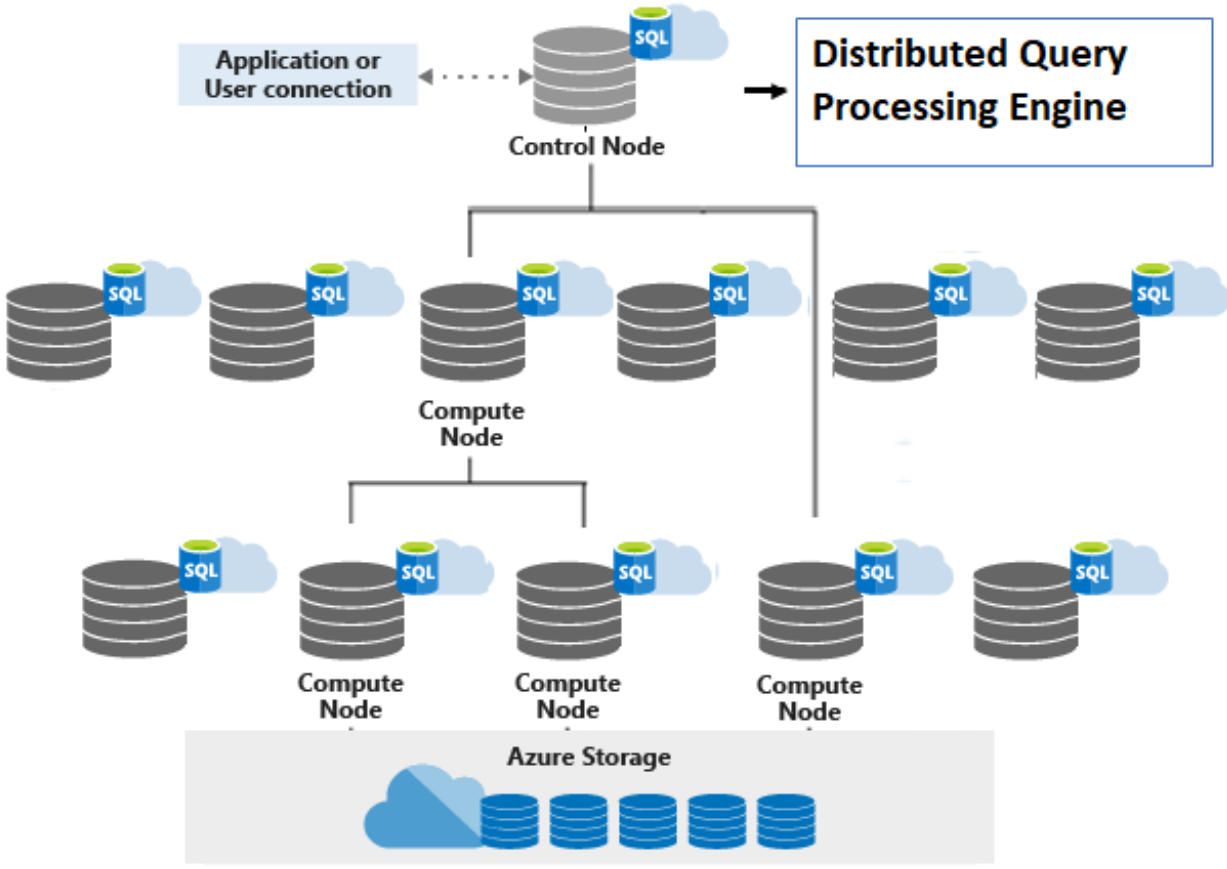
PRESIDIO

Azure Synapse – Architecture

Dedicated SQL pool

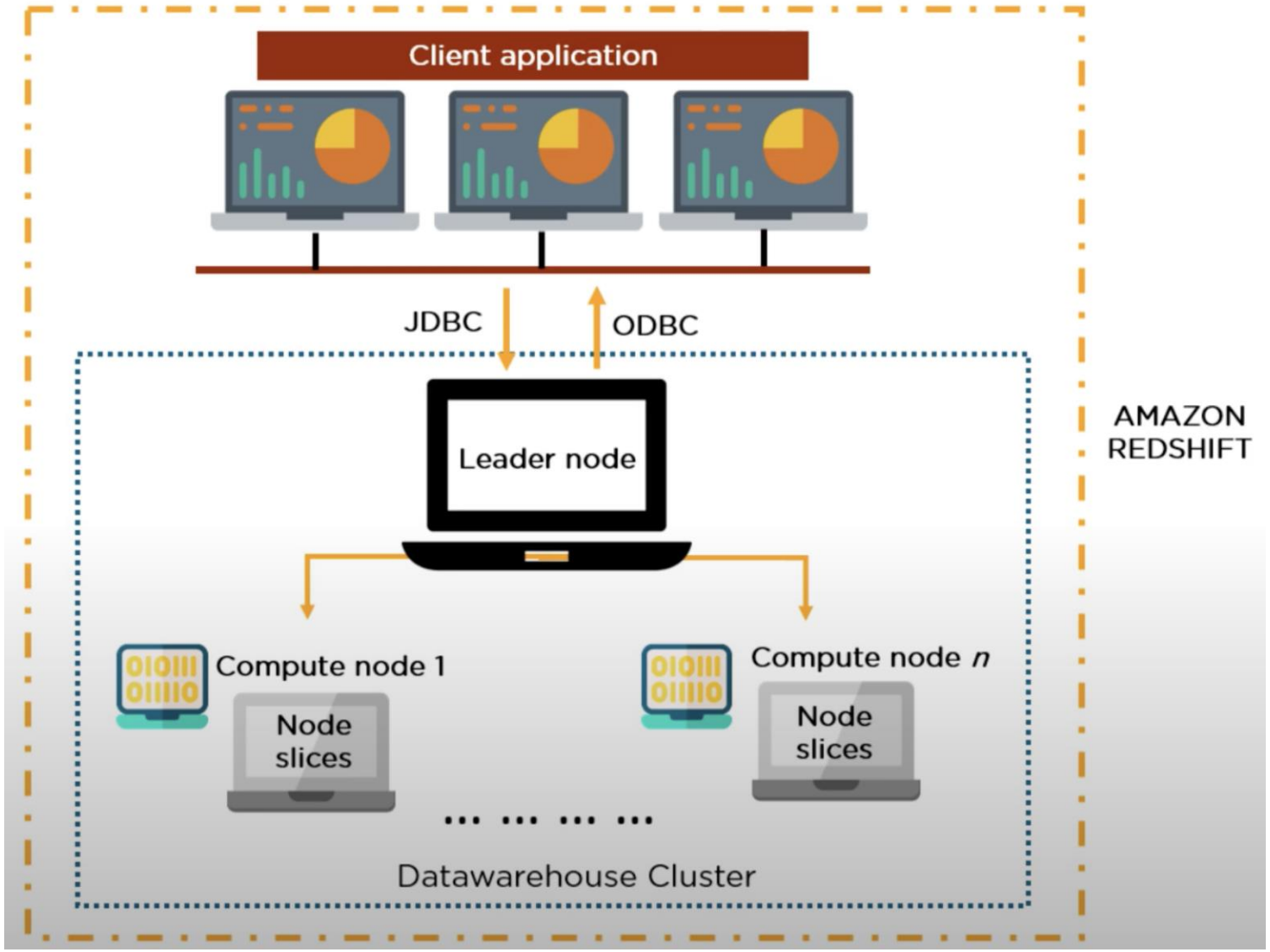


Serverless SQL pool



PRESIDIO

Redshift – Architecture



Sources of batch data:

- Copy from S3
- Copy from EMR
- Copy from remote Host(SSH)
- Copy from DynamoDB

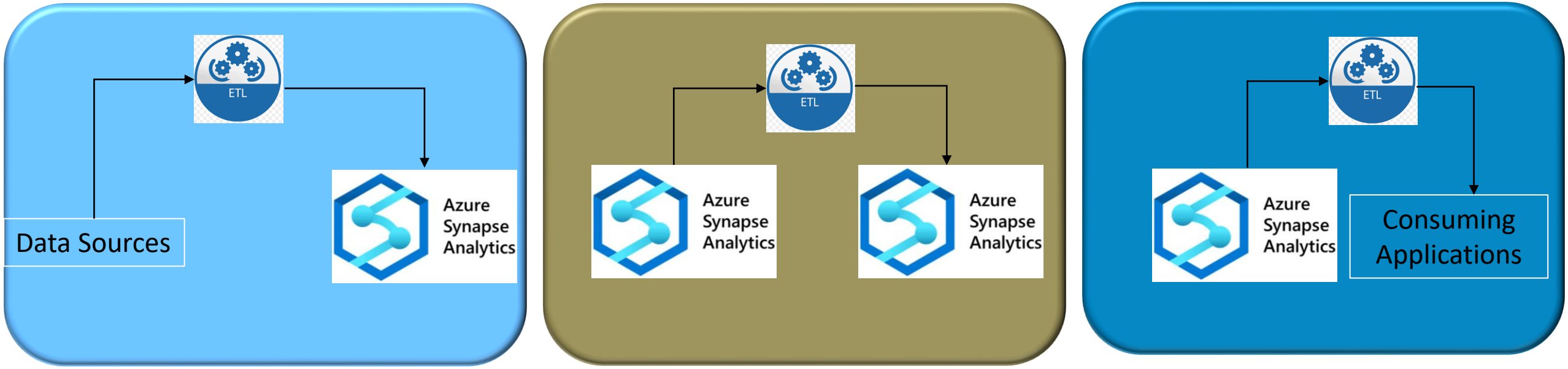
Sources of Streaming data:

- Amazon Kinesis firehose can directly add redshift as a destination

ETL – Tools Introduction

Extract transform and load(ETL) tools are used to move the data from one place to another while transforming them in-flight
Three common usages as below

- Transform and ingest to Datawarehouse
- Transform within Datawarehouse
- Extract and transform from Datawarehouse



Glue and Other tools

AWS

- Glue
- Kinesis
- Data pipeline

Azure

- Azure Data Factory
- Azure Synapse pipelines

GCP

- Cloud Data Fusion
- DataFlow
- Dataproc

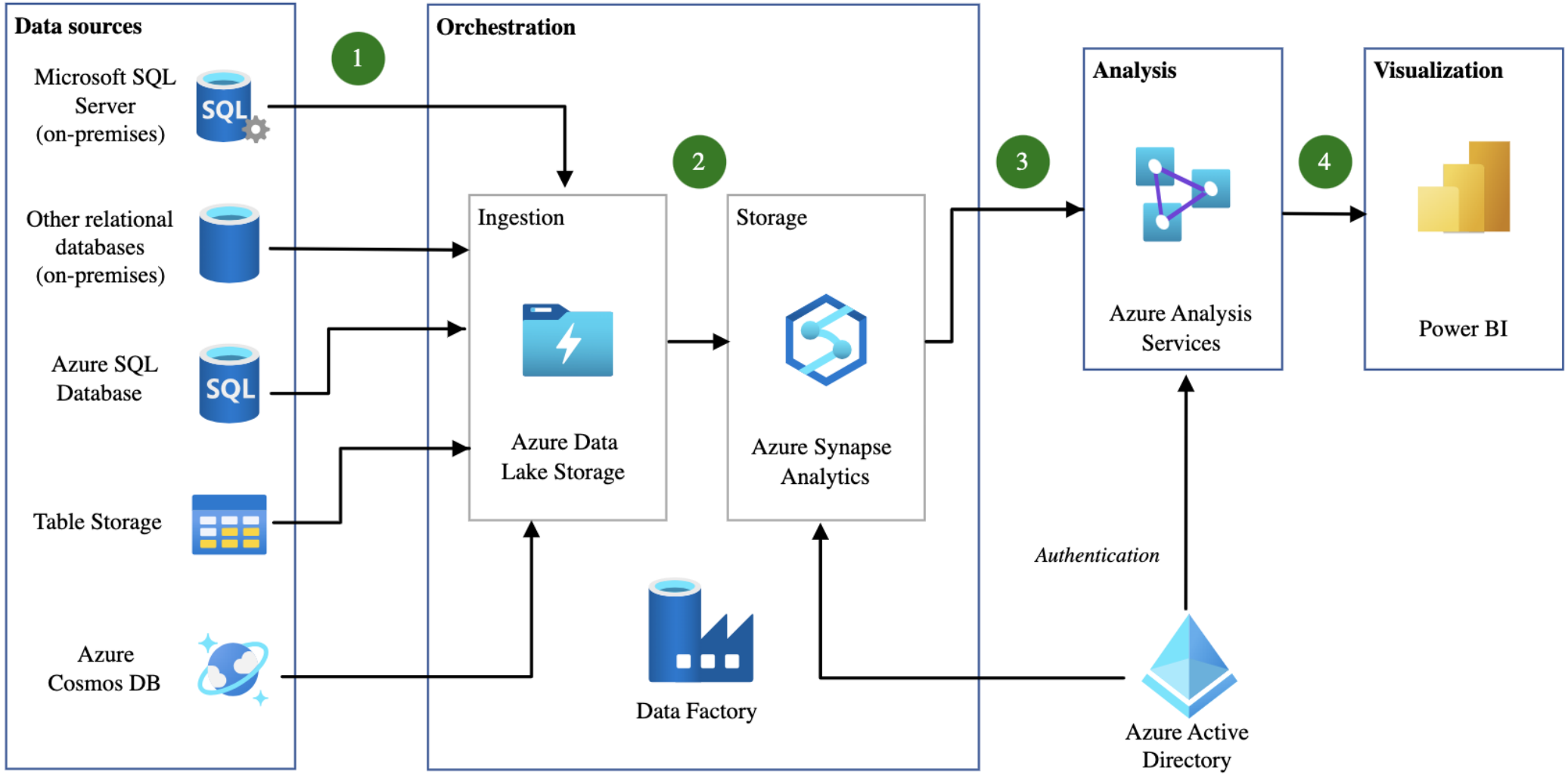
Outside Cloud

- Ab-Initio
- Talend
- Informatica
- Stitch
- And many more…

Orchestration Tools

AWS Step Functions

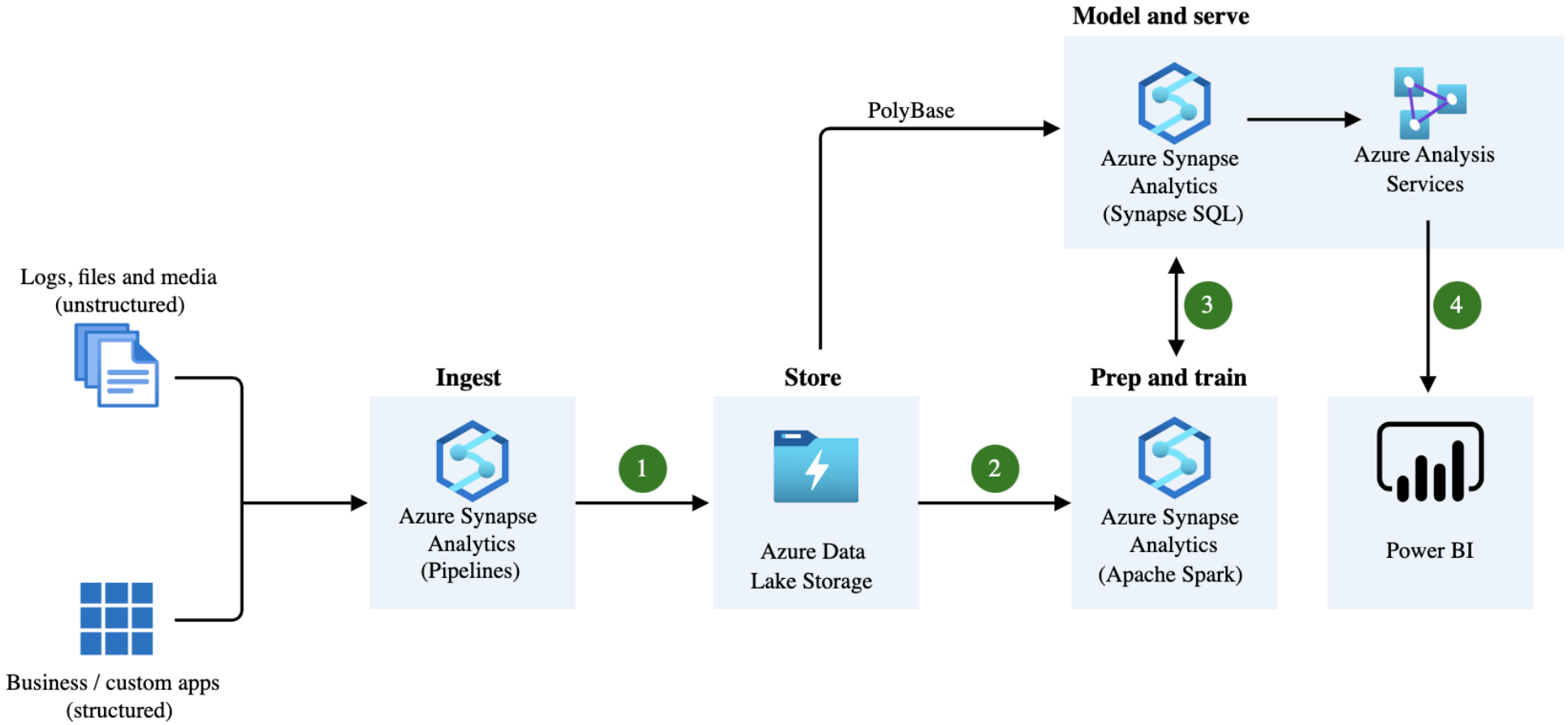
Sample Architecture 1



https://learn.microsoft.com/en-us/azure/architecture/example-scenario/data/data-warehouse

PRESIDIO

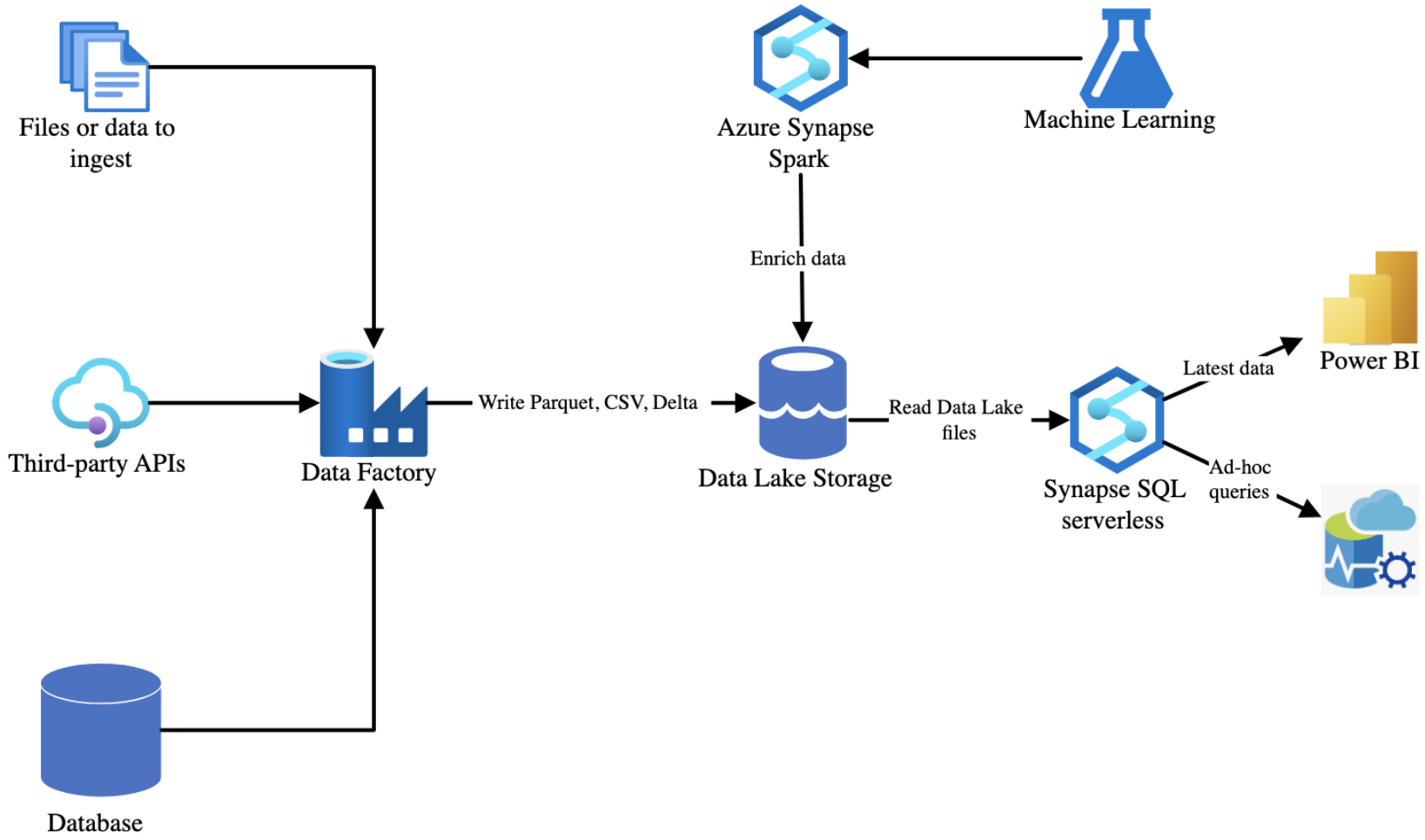
Sample Architecture 2



https://learn.microsoft.com/en-us/azure/architecture/solution-ideas/articles/enterprise-data-warehouse

PRESIDIO

Sample Architecture 3 – Data lake



https://learn.microsoft.com/en-us/azure/architecture/browse/?terms=data%20lake

PRESIDIO

OLTP vs OLAP – Generic differences

	OLTP	OLAP
Nature of workload	OLTP systems are designed to handle transactional workloads, such as recording and processing day-to-day business transactions	OLAP systems are used for analytical processing and complex data analysis.
Design optimization purpose	They are optimized for fast and efficient data input, retrieval, and modification.	They are designed to handle large volumes of data and support complex queries, aggregations, and data mining operations.
Example systems	Examples of OLTP applications include banking systems, e-commerce websites, airline reservation systems, and point-of-sale systems.	Examples of OLAP applications include business intelligence tools, data warehouses, and reporting systems.

OLTP vs OLAP – Data design differences

	OLTP	OLAP
Data model methodology	OLTP databases typically have a normalized data structure, meaning data is organized into separate tables to minimize redundancy and ensure data integrity.	OLAP databases often have a denormalized or star schema data structure, where data is pre-aggregated and organized into a dimensional model for easier analysis.
Indexing	OLTP databases typically use indexes on specific columns to optimize transactional operations, such as searching and updating records. Indexes are essential for efficient data retrieval and maintaining data consistency in OLTP systems.	In OLAP databases, indexing strategies may vary based on the specific analytical requirements. While indexes can still be used in OLAP systems, they are not as crucial as in OLTP systems. OLAP systems often rely on other optimization techniques like pre-aggregation, partitioning, or compression to improve query performance.
Data Volume	OLTP databases generally handle moderate to high volumes of data due to the continuous recording and processing of business transactions. However, the emphasis is on processing individual transactions efficiently rather than storing large amounts of historical data.	In contrast, OLAP databases typically store and process large volumes of historical data for analytical purposes. These systems are optimized to handle complex queries across extensive datasets, including aggregations, slicing, dicing, and multidimensional analysis.
Data Latency	OLTP systems prioritize real-time processing and aim to minimize data latency. Changes made to the database need to be immediately reflected to ensure data consistency and transactional integrity.	OLAP systems focus on analysing and reporting on historical data, and they usually have a higher tolerance for latency. Data in OLAP systems is typically updated periodically, such as on a daily, weekly, or monthly basis, depending on the organization's needs.

Row Vs columnar storage

Row Storage	Columnar Storage
• In row-based storage, data is stored and retrieved by rows. Each row contains a complete record with all its attributes or fields. • Row-based storage is well-suited for OLTP (Online Transaction Processing) systems, where individual transactions and records need to be accessed and modified quickly. • It is efficient for transactional operations that involve inserting, updating, or deleting a small number of records. • Row-based storage is optimized for write-intensive workloads, as it allows for fast record updates and maintains data integrity by minimizing data duplication. • However, it can be less efficient for analytical queries that involve aggregations or accessing a subset of columns.	• In columnar storage, data is stored and retrieved by columns or attributes. Each column contains the values for a specific attribute across all records. • Columnar storage is well-suited for OLAP (Online Analytical Processing) systems, where complex analytical queries and aggregations are performed on large volumes of data. • It is optimized for read-intensive workloads, as columnar storage allows for efficient data compression, reduced I/O operations, and improved query performance. • Columnar storage is particularly useful for data analysis scenarios that involve accessing a subset of columns or performing aggregations on specific attributes. • However, columnar storage can be less efficient for write operations, as updating or inserting new records requires modifying multiple column segments.

https://www.geeksforgeeks.org/what-is-a-columnar-database/