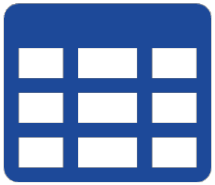
Data Types

Four types of Data

Types of Data



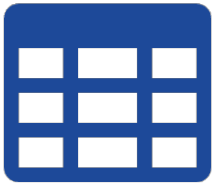
Structure Data

Semi-structured Data

Unstructured Data

Streaming Data

Types of Data



Structure Data

- Data That is organized. It has a strict defined schema which defines field names, data types, and the relationship between tables.
- Example – Database, Data Warehouse, ERP, CRM
- Schema-on-write
  - Highly precise schema that is defined on Write
- Difficult to make changes to the schema to accept new data changes
- Extract Transform Load (ETL)

CAN VIOLATE CONSTRAINTS

TRANSFORM BEFORE LOAD TO CONFORM TO SCHEMA

Types of Data

Semi-structured Data

- Data That is NOT organized and does not conform to a formal structure of tables. But it does have structures such as tags or metadata associated with it. This allows records and fields within the data.
- Example – CSV, XML, JSON
- Easy to make changes in Schema
  - Schema is not strictly enforced
- Schema-on-read

Types of Data

Unstructured Data

- Data that does not have a pre-defined data model, and it is not organized in any particular manner that allows traditional analysis.
- Example – Videos, images, social media post, emails, music
- 90% of all new data is unstructured
- Does not have a schema or attributes within the data
- Highly flexible to accept new changes to the data
- Vast assortment of data types and growing everyday

Types of Data

Streaming Data

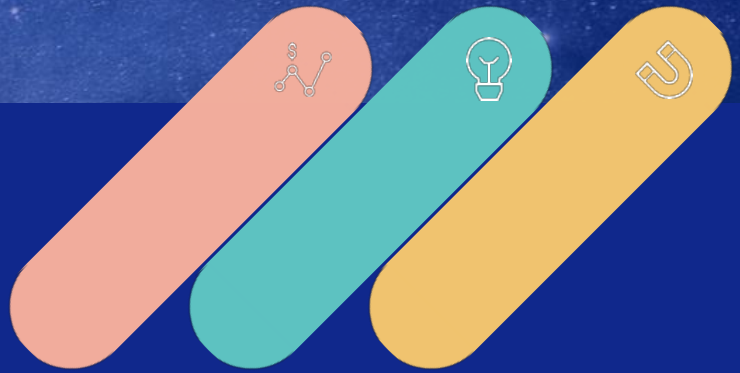
- Data not at rest. Data that is continuous flow from one place to another place.

  This flow of the data provides an opportunity for immediate analysis or consumption.

- Example – Media, Satellite, IOT
- Streaming Data Analysis
  - Batch – After the stream is stored the data is analyzed to look for patterns and relationships
  - Real-time – The data is analyzed during gathering to make an immediate reaction to a trigger

Data store

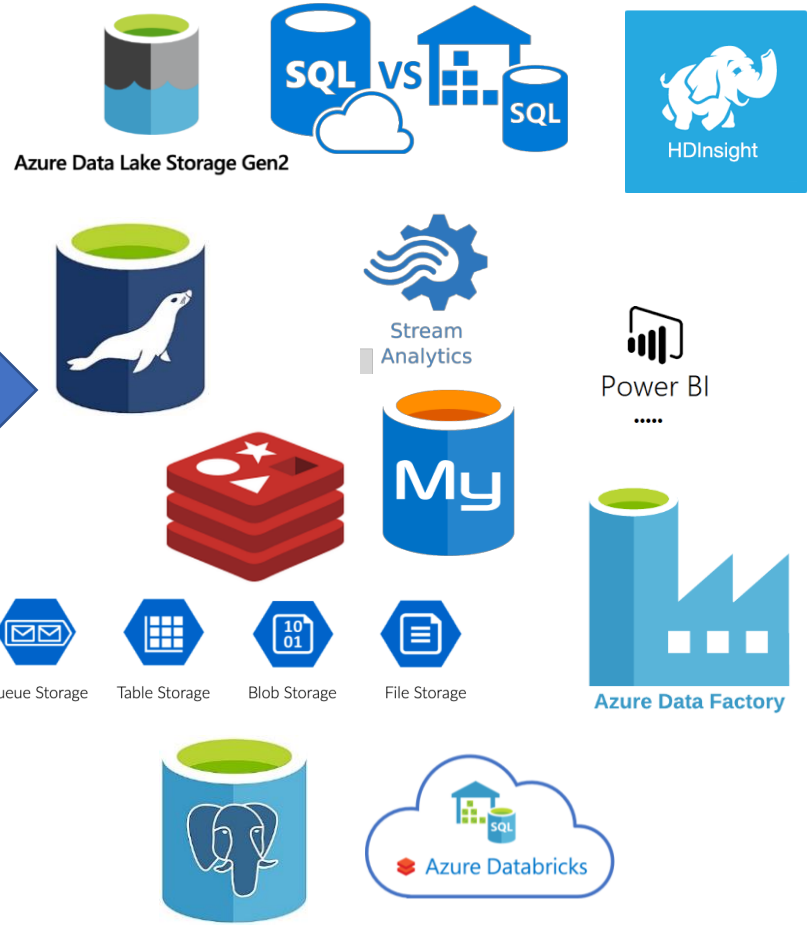
Understand data store models



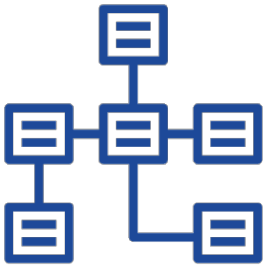
Why we need different data store?

- Store different types of data in different data stores
- Categorized by
  - Structure of data
  - Types of Operations on data.

READ / WRITE
UPDATE
INSERT
DELETE
ETC.



Relational database management systems



Data Structure

- Organize data as a series of two-dimensional tables with rows and columns
- Schema-on-write
- Normalized
- Relationships are enforced using database constraints

Operation

- Structured Query Language (SQL)
- ACID (Atomic, **Consistent**, Isolated, Durable)

Examples

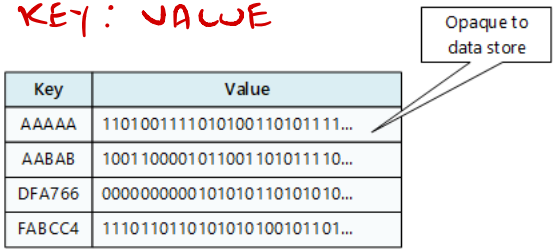
- Inventory management
- Order management
- Reporting database
- Accounting

Azure services

- Azure SQL Database
- Azure Database for MySQL
- Azure Database for PostgreSQL
- Azure Database for MariaDB

Key/value stores

KEY : VALUE



Data Structure

- Each data value associated with a unique key
- Scalable
- No relationships between entities.

NO UPDATES

Operation

- Support simple insert/delete operation
- Highly optimized for applications performing simple lookups

Examples

- Data caching
- Session management
- User preference and profile management
- Product recommendation and ad serving

Azure services

- Azure Cosmos DB
- Azure Cache for Redis

Document databases

KEY: VALUE

Key	Document
1001	{ "CustomerID": 99, "OrderItems": [{ "ProductID": 2010, "Quantity": 2, "Cost": 520 }, { "ProductID": 4365, "Quantity": 1, "Cost": 18 }], "OrderDate": "04/01/2017" }
1002	{ "CustomerID": 220, "OrderItems": [{ "ProductID": 1285, "Quantity": 1, "Cost": 120 }], "OrderDate": "05/08/2017" }

Data Structure

- Stores a collection of documents
- Document contains the data for single entity, such as a customer or an order.
- Documents are retrieves by unique keys
- Document data is semi-structured, meaning that data types of each field are not strictly defined.

Operation

- Individual documents are retrieved and written as a single block.
- Data requires index on multiple fields.

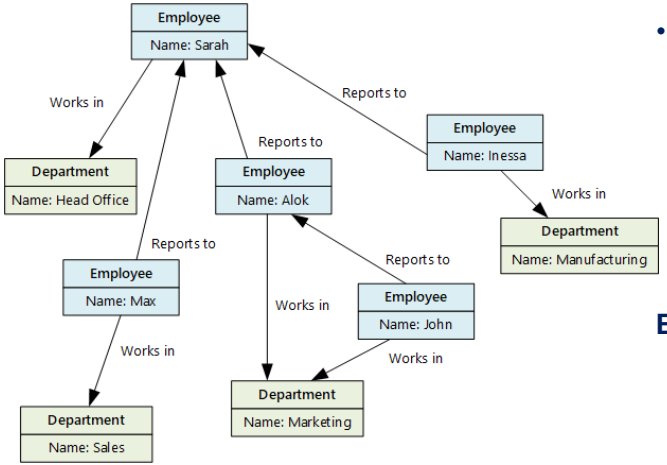
Examples

- Product catalog
- Content management
- Inventory management

Azure services

- Azure Cosmos DB

Graph databases



Data Structure

- Stores two types of information, nodes and edges.
- Nodes are similar to table rows or JSON documents
- Complex relationships between data items

RELATIONSHIPS

Examples

- Organization charts
- Social graphs
- Fraud detection
- Recommendation engines

Operation

- Efficiently perform queries across the network of nodes and edges and analyze the relationships between entities.
- Data requires index on multiple fields.

Azure services

- Azure Cosmos DB Gremlin API
- SQL Server

Column-family databases

CustomerID	Column Family: Identity
001	**First name:** Mu Bae **Last name:** Min
002	**First name:** Francisco **Last name:** Vila Nova **Suffix:** Jr.
003	**First name:** Lena **Last name:** Adamcyz **Title:** Dr.

CustomerID	Column Family: Contact Info
001	**Phone number:** 555-0100 **Email:** someone@example.com
002	**Email:** vilanova@contoso.com
003	**Phone number:** 555-0120

Data Structure

- Organizes data into rows and columns
- Denormalized approach to structuring sparse data
- Each column family holds a set of columns that are logically related together and are typically retrieved or manipulated as a unit.

Examples

- Recommendations
- Personalization
- Sensor data
- Telemetry
- Messaging
- Social media analytics
- Web analytics

Operation

- Read and write operations for a row are usually atomic with a single column-family
- Update and delete operations are rare.

Azure services

- Azure Cosmos DB Cassandra API
- HBase in HDInsight

Data analytics

Data Structure

- Provide massively parallel solutions for ingesting, storing, and analyzing data.
- Data is distributed across multiple servers to maximize scalability.
- Usually denormalized in a "star" or "snowflake" schema
- Consisting of fact and dimension tables.

Operation

- Data analytics
- Enterprise BI

Examples

- Enterprise data warehouse

Azure services

- Azure Synapse Analytics MPP
- Azure Data Lake
- Azure Data Explorer
- Azure Analysis Services
- HDInsight
- Azure Databricks

Object storage

Data Structure

- Optimized for storing and retrieving large binary objects
- Stores can manage extremely large amounts of unstructured data.

Operation

- Identified by key.

Examples

- Images, videos, office documents, PDFs
- Static HTML, JSON, CSS
- Log and audit files
- Database backups

Azure services

- Azure Blob Storage
- Azure Data Lake Storage Gen2

HIERARCHICAL

Shared files

Data Structure

- Using file shares enables files to be accessed across a network.
- Requires SMB interface.
- Cross platform - Mount your Azure Files from Windows, Linux, or macOS.

Operation

- Accessible with standard I/O libraries.

Examples

- Legacy files
- Shared content accessible among a number of VMs or app instances

Azure services

- Azure Files

Time series databases

Data Structure

- Azure Time Series Insights is built to store, visualize, and query large amounts of time series data.

Examples

- Monitoring and event telemetry.
- Sensor or other IoT data.

Operation

- Records are generally appended sequentially in time order.
- Updates are rare.
- Deletes occur in bulk
- Data is read sequentially in either ascending or descending time order

Azure services

- Azure Time Series Insights

Search Engine Databases

Data Structure

- Data indexes from multiple sources and services.

Examples

- Product catalogs
- Site search
- Logging

Operation

- Searching can be exact or fuzzy.

Azure services

- Azure Search

General Consideration

Data Stores

RESOURCES

Relational database

- Azure SQL Database
- Azure Database for MySQL
- Azure Database for PostgreSQL
- Azure Database for MariaDB

Data analytics

- Azure Synapse Analytics
- Azure Data Lake
- Azure Data Explorer
- Azure Analysis Services
- HDInsight
- Azure Databricks

NO SQL

Key/value stores

- Azure Cosmos DB Table API
- Azure Cache for Redis

Document databases

- Azure Cosmos DB SQL API

Column-family databases

- Azure Cosmos DB Cassandra API
- HBase in HDInsight

Graph databases

- Azure Cosmos DB Gremlin API
- SQL Server

Shared files

- Azure Files / SMB PROTOCOL

Object storage

- Azure Blob Storage
- Azure Data Lake Storage Gen2

Search Engine Databases

- Azure Search

Time series databases

- Azure Time Series Insights

NO SQL = NON - RELATIONAL

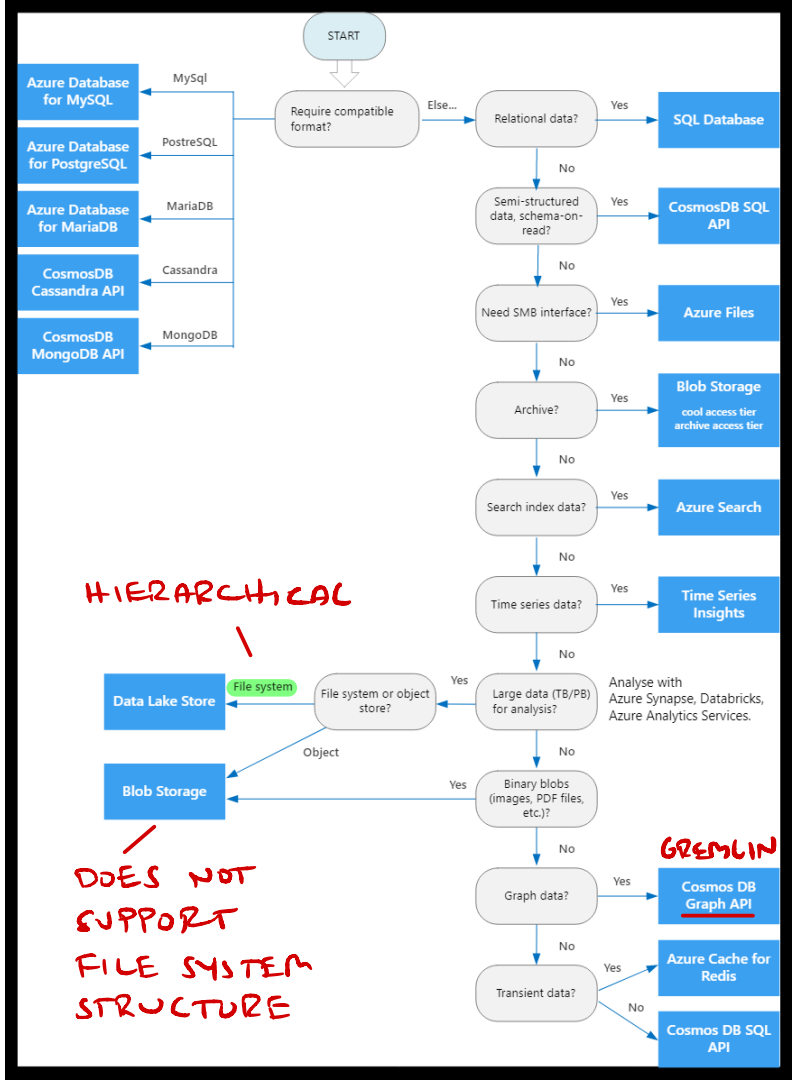
HOW TO CHOOSE DATABASE

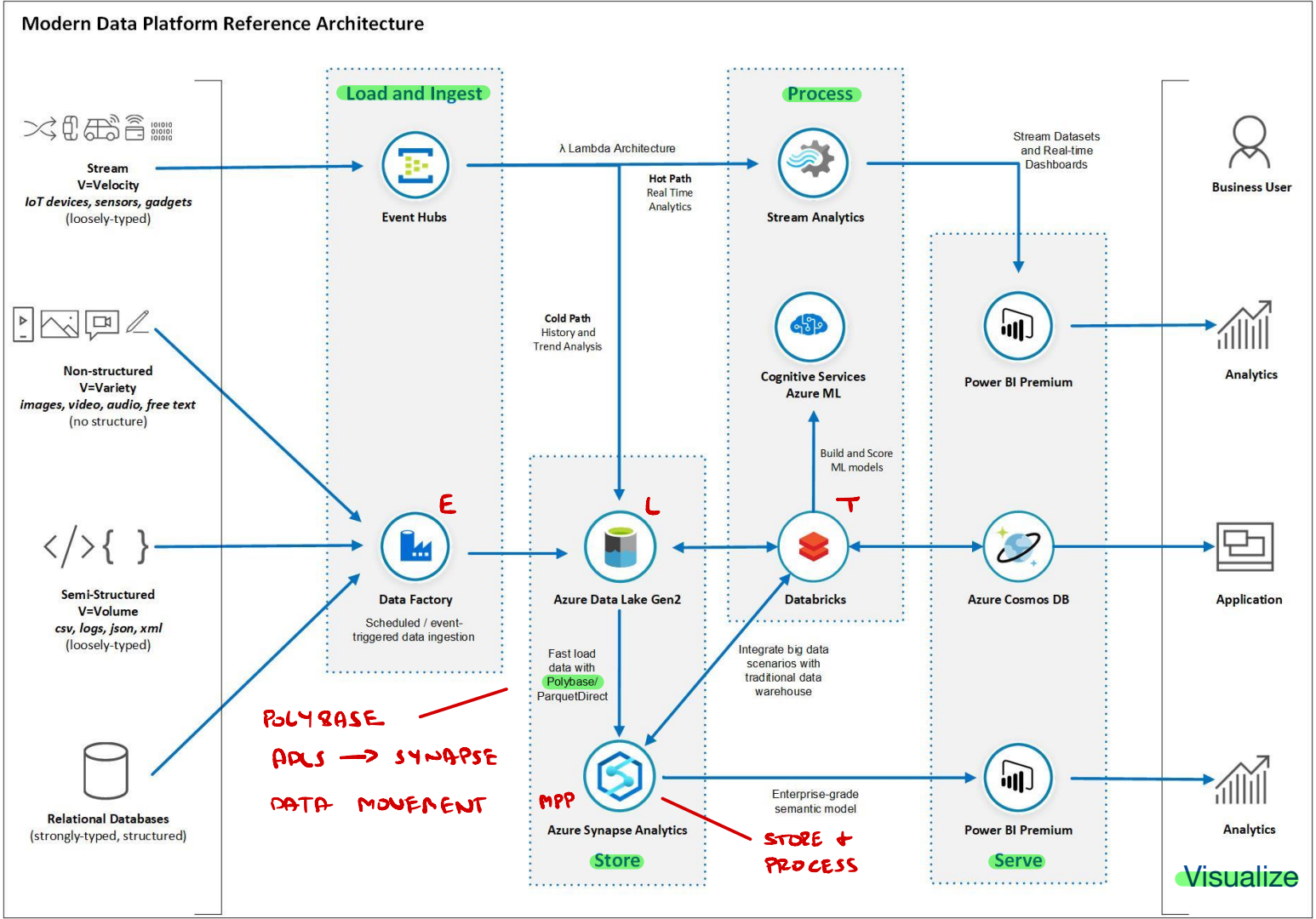
ADLS

CONTAINER
FOLDERS
FILES

BLOB

CONTAINER
FOLDERS




Modern Data Platform Reference Architecture
Load and Ingest
Process
Stream
V=Velocity
IoT devices, sensors, gadgets
(loosely-typed)
Event Hubs
λ Lambda Architecture
Hot Path
Real Time
Analytics
Stream Analytics
Stream Datasets
and Real-time
Dashboards
Business User
Cold Path
History and
Trend Analysis
Cognitive Services
Azure ML
Power BI Premium
Analytics
Non-structured
V=Variety
images, video, audio, free text
(no structure)
Build and Score
ML models
E
L
T
Semi-Structured
V=Volume
csv, logs, json, xml
(loosely-typed)
Data Factory
Scheduled / event-
triggered data ingestion
Azure Data Lake Gen2
Databricks
Azure Cosmos DB
Application
Fast load
data with
Polybase/
ParquetDirect
Integrate big data
scenarios with
traditional data
warehouse
POLYBASE
ADLS → SYNAPSE
DATA MOVEMENT
Relational Databases
(strongly-typed, structured)
MPP
Azure Synapse Analytics
Store
Enterprise-grade
semantic model
Power BI Premium
Serve
Analytics
STORE & PROCESS
Visualize

RTO and RPO

Recovery Time Objective (RTO) and Recovery Point Objective (RPO)

DISASTER RECOVERY

RTO and RPO

Recovery Time Objective (RTO)

The maximum acceptable time that an application can be unavailable after an incident.

If your RTO is 90 minutes, you must be able to restore the application to a running state within 90 minutes from the start of a disaster.

If you have a very low RTO, you might need a warm standby running to protect against a regional outage.

SERVER UP + RUNNING

Recovery Point Objective (RPO)

The maximum duration of data loss that is acceptable during a disaster.

For example; one standalone database with hourly backups provides an RPO of 60 minutes.

If you require a lower RPO you'll need to design accordingly

DESIGNING A SOLUTION

That utilizes Cosmos DB, Data Lake Gen 2 or Blob Storage

Scenario 1

- Company: Fortune 500 car rental
- Goal: design the appropriate cloud architecture
- Details:
  - Millions of customer worldwide
  - 20,000 order per day from locations all over glob
  - Car pricing is dynamic, based on demand NEAR REAL-TIME
  - Allow orders from various web portals
  - Store data from variety of sources NO SQL
  - Provide reporting for multiple business silos

- Options:
  - Cosmos DB — WORLDWIDE ACCESS, NO SQL DATA STORE
  - Data Lake Gen 2 — REAL-TIME, DYNAMIC DATA
  - Blob Storage

Scenario 2

- Company: Fortune 500 financial planning
- Goal: design the appropriate cloud architecture
- Details:
  - Provide business intelligence to finance, human resources, and project management
  - Information coming from servers all over the united states
  - Allow business analysts to access raw data and build reports as needed.
- Options:
  - Cosmos DB
  - Data Lake Gen 2
  - Blob Storage

BI REPORTING (NOT REAL TIME)
COST EFFICIENT
RAW DATA FOR BA

Scenario 3

- Company: Online news agency
- Goal: design the appropriate cloud architecture
- Details:
  - Stores hundreds of terabytes of videos
  - Access videos from locations around the globe
  - Protect costs as much as possible
    - We don't get paid for videos but ads
- Options:
  - Cosmos DB
  - Data Lake Gen 2
  - Blob Storage

BLOB CHEAPEST STORE
ACCESSED WORLDWIDE
NO ANALYTICS REQUIRED

DESIGNING A SOLUTION

That utilizes SQL Server Database and Data Warehouse

Scenario 1

- Company: Craft Company
- Business: Sells to craft shows In different locations
- Need:
  - Collect transactional data at each event.
  - Store this data into 5 different databases based on several factors
    - Only one database is expected to be in use at any given period
  - Client is cost sensitive
  - Build basic executive level reports on data every month

OLTP -> SQL DB

- Options
  - Azure SQL Database – Single
  - Azure SQL Database – Elastic Pool
  - Azure SQL Database – Managed Instance
  - Azure SQL Datawarehouse

MULTIPLE DB'S
SHARED RESOURCES
OLTP
BASIC REPORTS

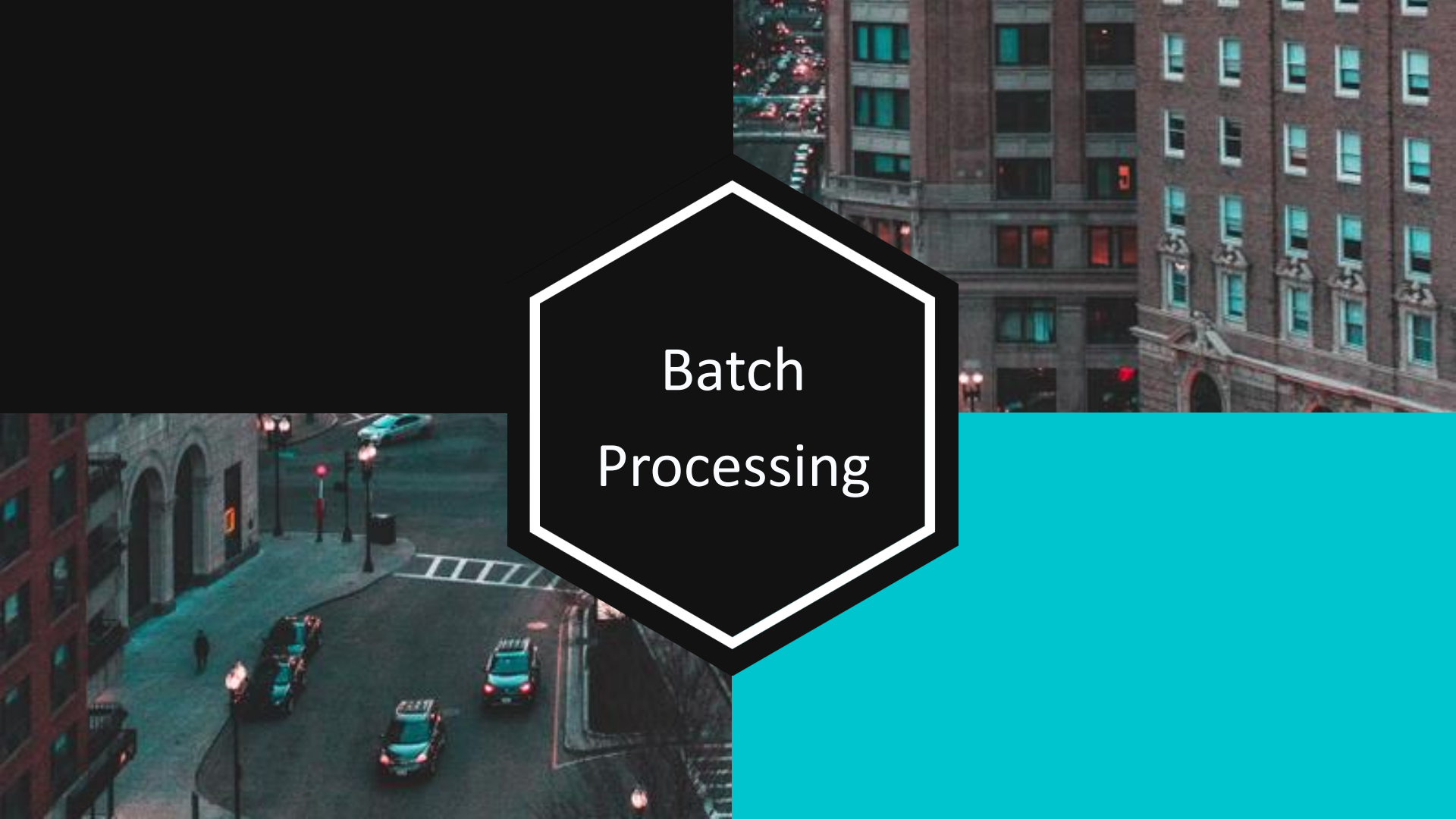
Scenario 2

- Company: Craft Company
- Business: Sells to craft shows In different locations
- Need:
  - Process complex queries from their massive repository of data.
  - Use these complex queries to influence business decisions and determine new opportunities
  - Cost is secondary to answers

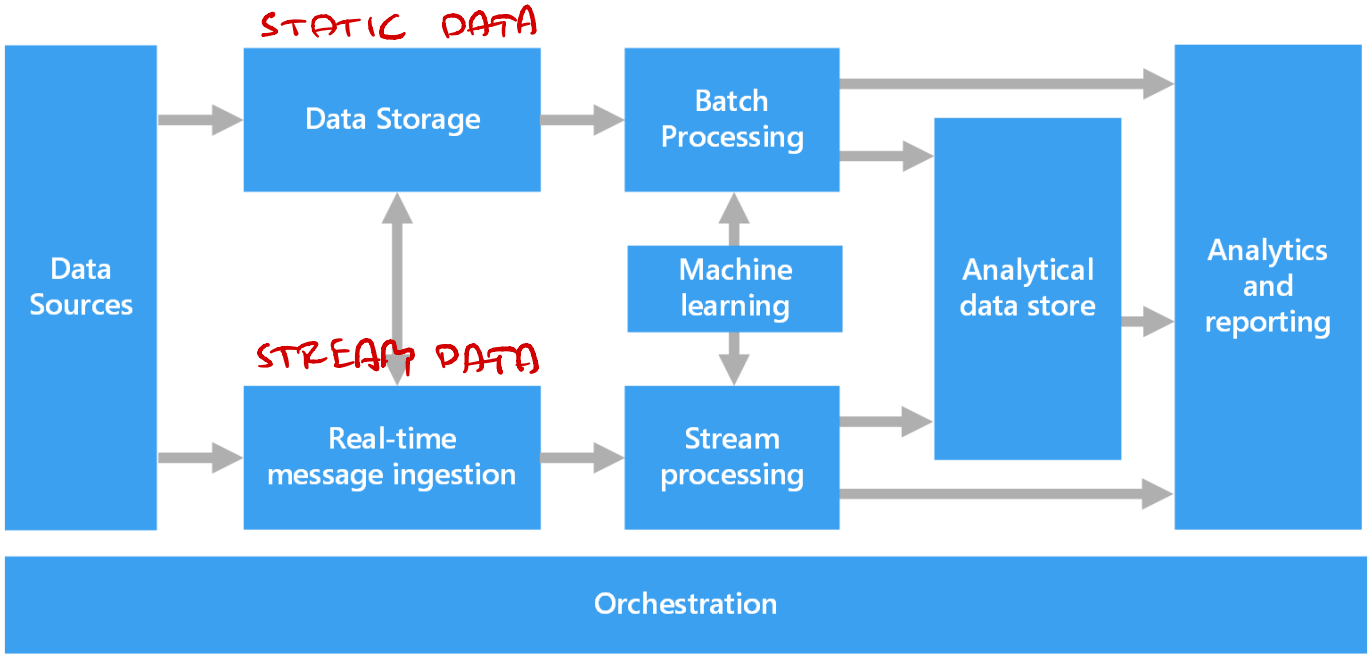
OLAP -> WAREHOUSE

- Options
  - Azure SQL Database – Single
  - Azure SQL Database – Elastic Pool
  - Azure SQL Database – Managed Instance
  - Azure SQL Datawarehouse

ANALYTICS
MPP
OLAP

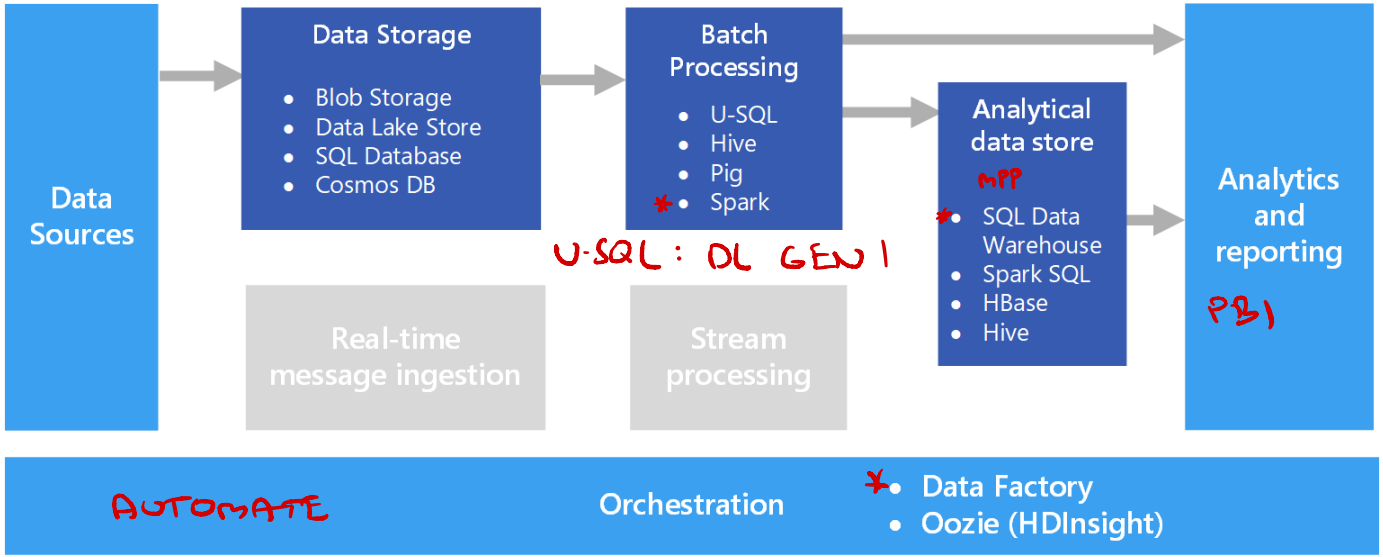

Batch
Processing

Big Data Architecture



Batch Processing

SQL DB: RELATIONAL COSMOS DB: NON-RELATIONAL (NO SQL)



Batch Processing:

- Data at rest STATIC
- Operate on very large dataset
- Computation takes significant time

Use Cases:

- Example – Web Server logs to Report

Challenges:

- Data Format and encoding
- Orchestration time slices

Azure Databricks

Azure Databricks

- Azure Databricks is an Apache Spark-based analytics platform
- think of it as "Spark as a service."

Features:

- Languages: R, Python, Java, Scala, Spark SQL
- Fast cluster start times, auto-termination, autoscaling.
- Manages the Spark cluster for you.
- Built-in integration with Azure Blob Storage, Azure Data Lake Storage (ADLS), Azure Synapse, and other services.
- User authentication with Azure Active Directory.
- Web-based notebooks for collaboration and data exploration.
- Supports GPU-enabled clusters

GRAPHICAL PROCESSING UNIT

Data Pipeline Orchestration

Pipeline Orchestration options:

ADF IS MOST IMPORTANT FOR CERTIFICATION

- Azure Data Factory
- Oozie on HDInsight
- SQL Server Integration Services (SSIS)

Key Selection Criteria:

- Do you need big data capabilities for moving and transforming your data? ADF
- Do you require a managed service that can operate at scale? ADF
- Are some of your data sources located on-premises? ADF / SSIS
- Is your source data stored in Blob storage on an HDFS filesystem? HD-INSIGHT

Data Pipeline Orchestration

General capabilities

Capability	Azure Data Factory	SQL Server Integration Services (SSIS)	Oozie on HDInsight
Managed	Yes	No	Yes
Cloud-based	Yes	No (local)	Yes
Prerequisite	Azure Subscription	SQL Server	Azure Subscription, HDInsight cluster
Management tools	Azure Portal, PowerShell, CLI, .NET SDK	SSMS, PowerShell	Bash shell, Oozie REST API, Oozie web UI
Pricing	Pay per usage	Licensing / pay for features	No additional charge on top of running the HDInsight cluster

Data Pipeline Orchestration

Pipeline capabilities

Capability	Azure Data Factory	SQL Server Integration Services (SSIS)	Oozie on HDInsight
Copy data	Yes	Yes	Yes
Custom transformations	Yes	Yes	Yes (MapReduce, Pig, and Hive jobs)
Azure Machine Learning scoring	Yes	Yes (with scripting)	No
HDInsight On-Demand	Yes	No	No
Azure Batch	Yes	No	No
Pig, Hive, MapReduce	Yes	No	Yes
Spark	Yes	No	No
Execute SSIS Package	Yes	Yes	No
Control flow	Yes	Yes	Yes
Access on-premises data	Yes	Yes	No

Data Pipeline Orchestration

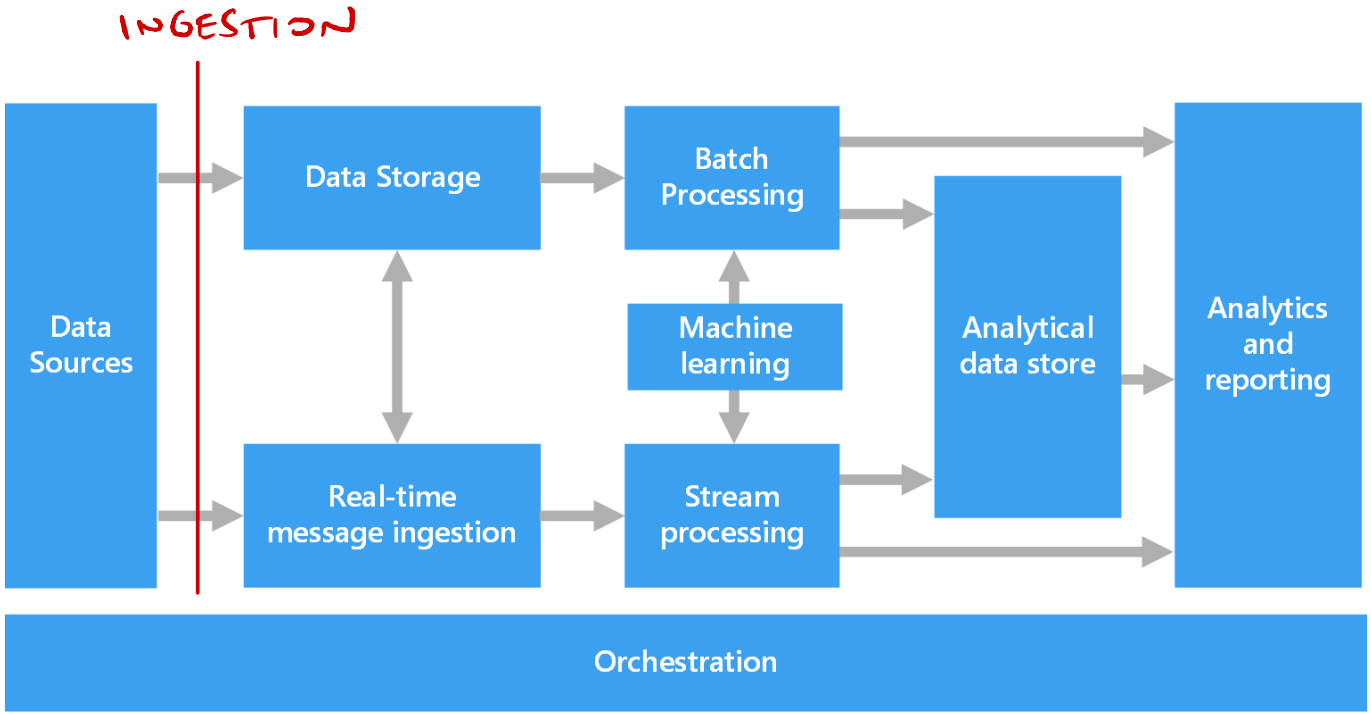
Scalability capabilities

Capability	Azure Data Factory	SQL Server Integration Services (SSIS)	Oozie on HDInsight
Scale up	Yes	No	No
Scale out	Yes	No	Yes (by adding worker nodes to cluster)
Optimized for big data	Yes	No	Yes

Data Ingestion methods

For Batch Processing solution

Big Data Architecture



Data Ingestion tools

Command line tools/APIs

- Azure CLI
- AzCopy
- PowerShell
- AdlCopy DL GEN1
- Distcp
- Sqoop
- PolyBase ✗
- HadoopCommandline

Graphical Interface

- Azure Storage Explorer
- Azure portal

Data pipeline

- Azure Data Factory

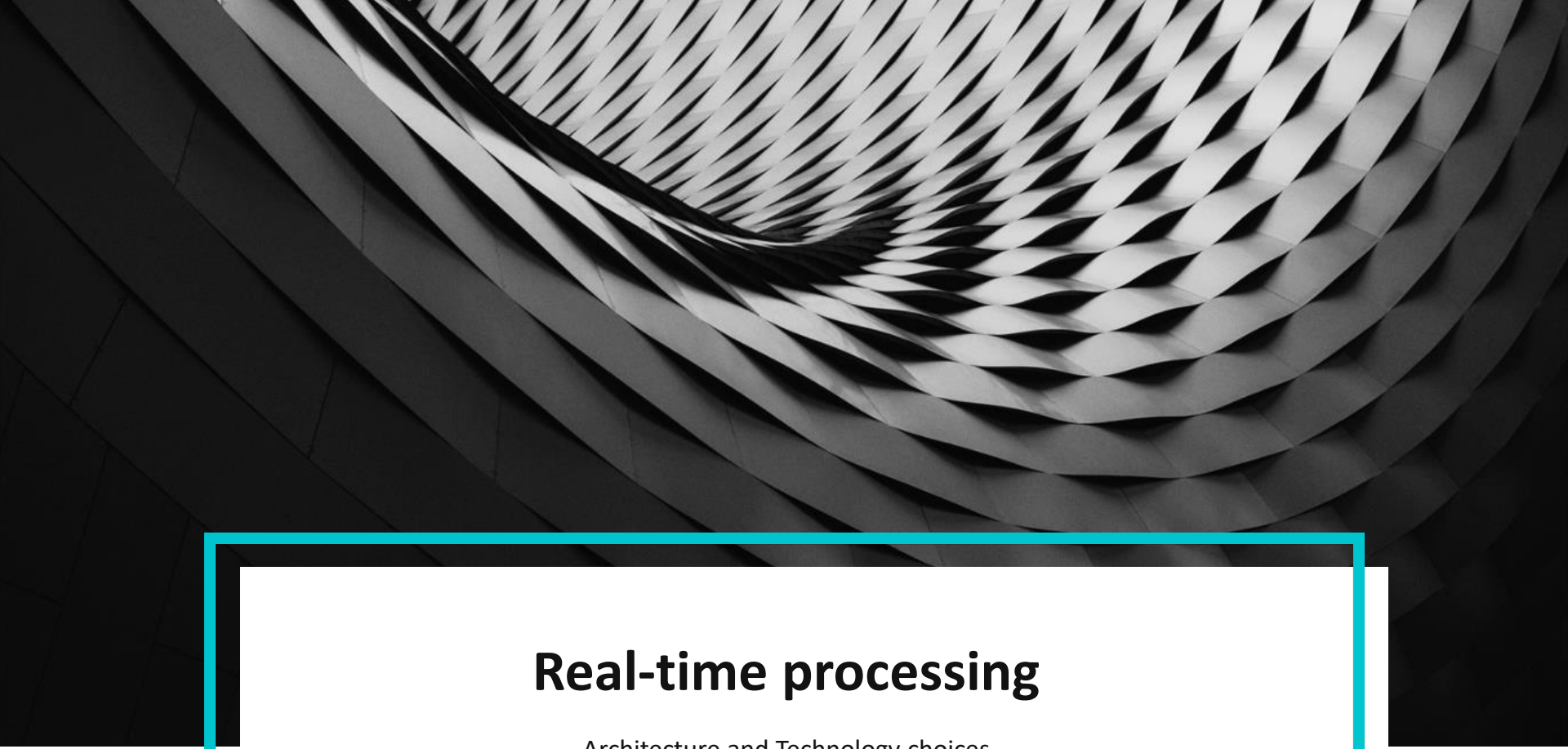
USE T-SQL TO CREATE EXT TABLES & IMPORT USING POLYBASE (SYNAPSE)

Command line tools comparison

Capability	Azure CLI	AzCopy	PowerShell	AdlCopy	PolyBase
Compatible platforms	Linux, OS X, Windows	Linux, Windows	Windows	Linux, OS X, Windows	SQL Server, Azure Synapse
Optimized for big data	No	Yes	No	Yes [1]	Yes [2]
Copy to relational database	No	No	No	No	Yes
Copy from relational database	No	No	No	No	Yes
Copy to Blob storage	Yes	Yes	Yes	No	Yes
Copy from Blob storage	Yes	Yes	Yes	Yes	Yes
Copy to Data Lake Store	No	Yes	Yes	Yes	Yes
Copy from Data Lake Store	No	No	Yes	Yes	Yes

Graphical interface and Azure Data Factory

Capability	Azure Storage Explorer	Azure portal *	Azure Data Factory
Optimized for big data	No	No	Yes
Copy to relational database	No	No	Yes
Copy from relational database	No	No	Yes
Copy to Blob storage	Yes	No	Yes
Copy from Blob storage	Yes	No	Yes
Copy to Data Lake Store	No	No	Yes
Copy from Data Lake Store	No	No	Yes
Upload to Blob storage	Yes	Yes	Yes
Upload to Data Lake Store	Yes	Yes	Yes
Orchestrate data transfers	No	No	Yes
Custom data transformations	No	No	Yes
Pricing model	Free	Free	Pay per usage



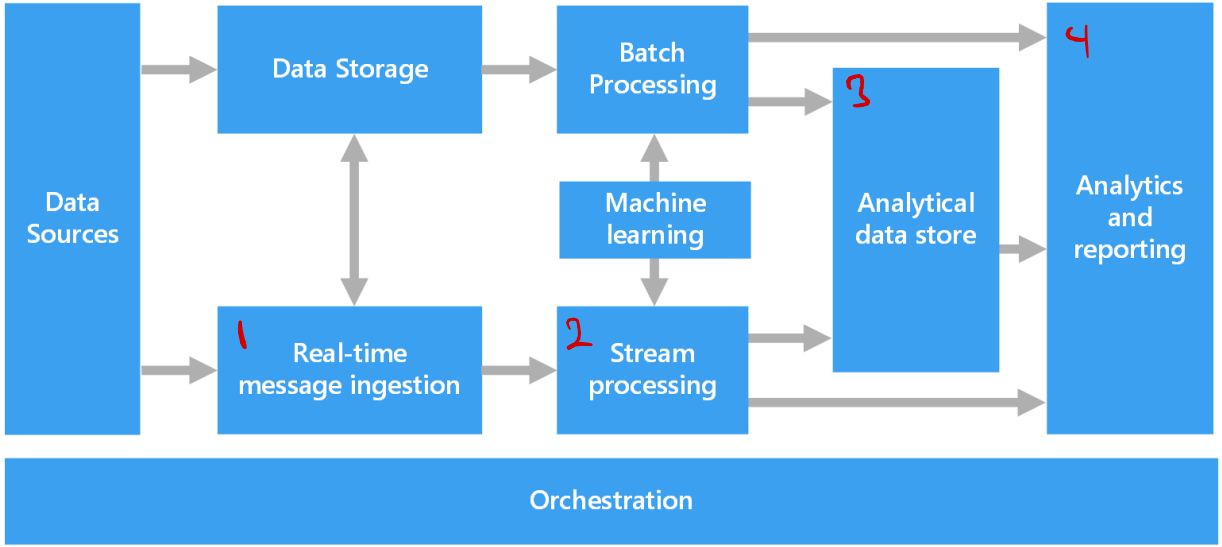
Real-time processing

Architecture and Technology choices

Real Time Processing Architecture

4 COMPONENTS

INGEST
PROCESS
STORE
SERVE



Real time processing:

- Deals with streams of data that are captured in real-time
- Processed with minimal latency
- Incoming data typically arrives in an unstructured or semi-structured format, such as JSON
- Generate real-time (or near-real-time) reports or automated responses.

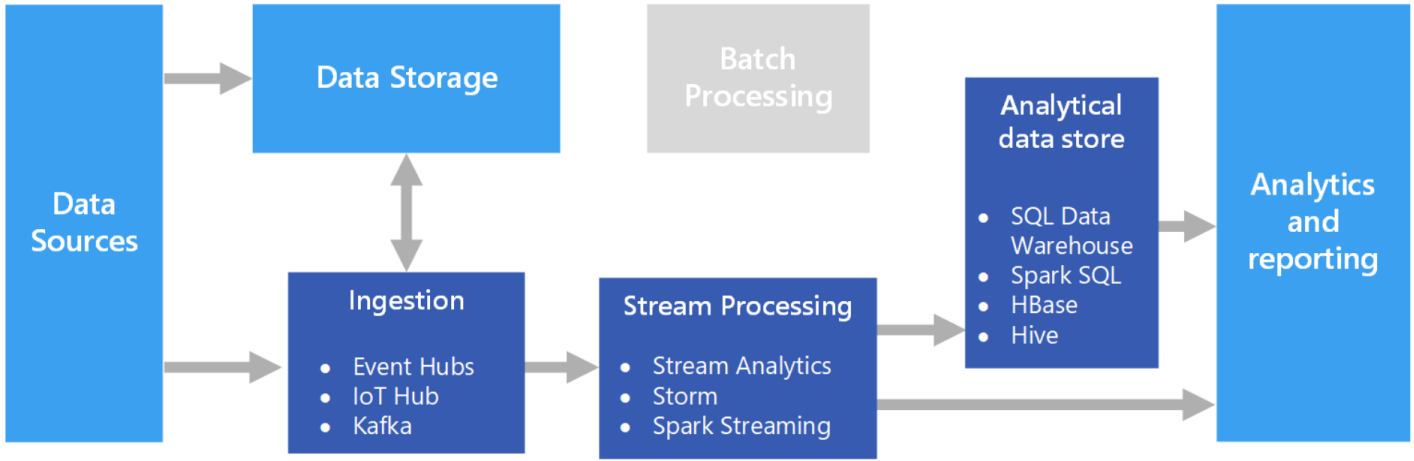
For example:

- Use sensor data to detect high traffic volumes

Challenges:

- Ingest, process, and store messages in real time, especially at high volumes

Real Time Processing Architecture



Real-time message ingestion:

Azure Event Hubs: Messaging solution for ingesting millions of event messages per second.

- Can be processed by multiple consumers in parallel
- natively supports AMQP (Advanced Message Queuing Protocol 1.0)

Azure IoT Hub: Provides bi-directional communication between Internet-connected devices

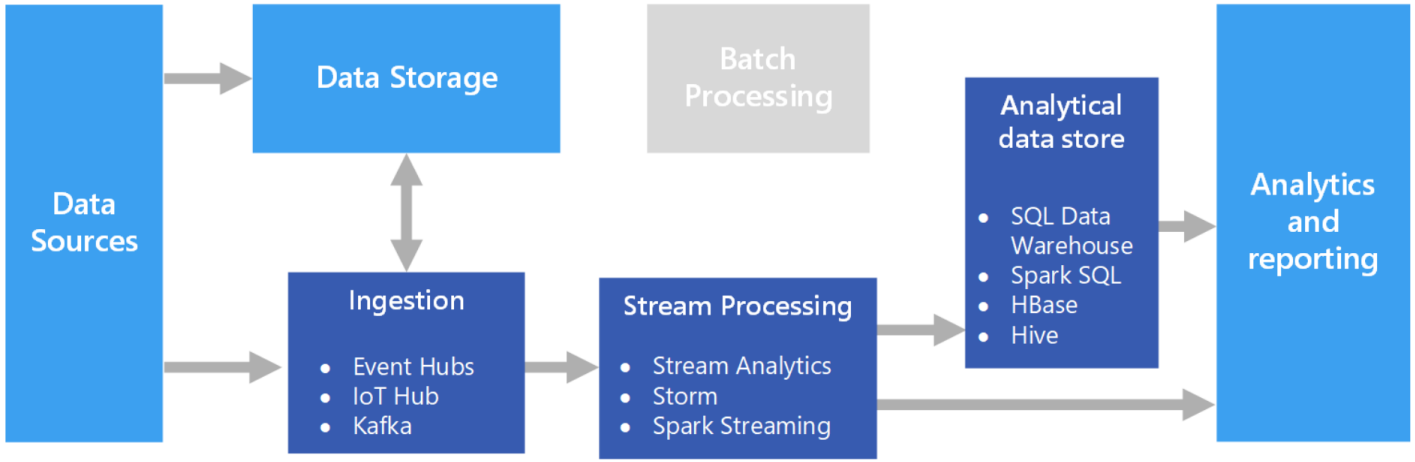
- Scalable message queue that can handle millions of simultaneously connected devices.

Apache Kafka: Open source message queuing and stream processing application

- IOT HUB IS MORE EXPENSIVE THEN EVENT HUB
- DL/BLOB NEW FILE TRIGGERS

Azure Storage Blob Containers or Azure Data Lake Store: can be use for static reference data, or output destination for captured real-time data for archiving

Real Time Processing Architecture



Stream processing

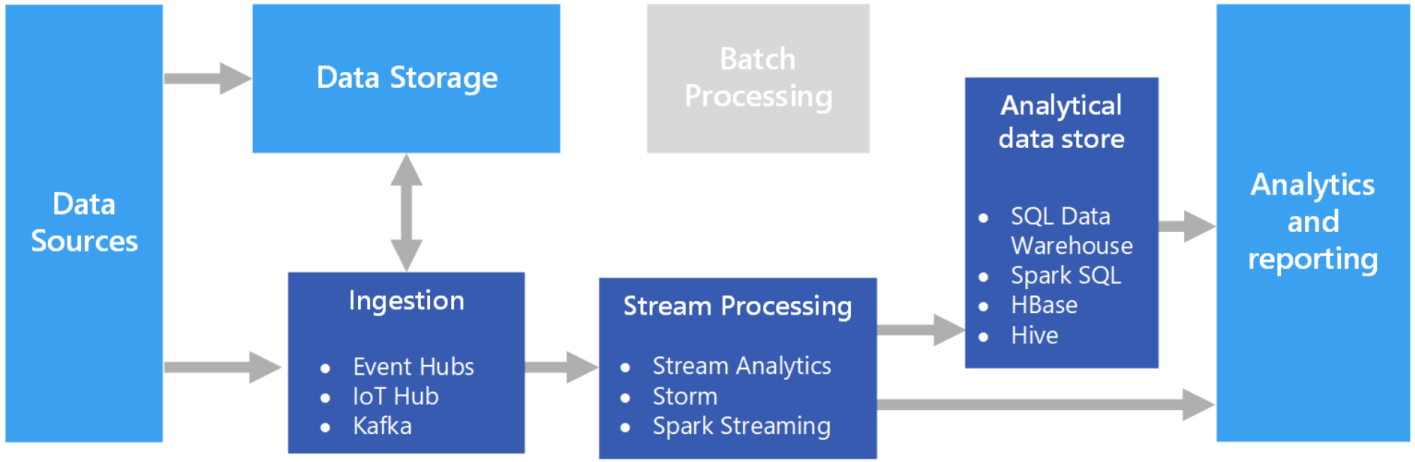
Azure Stream Analytics: can run perpetual queries against an unbounded stream of data

- Consume streams of data from storage or message brokers
- Filter and aggregate the data based on temporal windows
- Write the results to sinks such as storage, databases, or directly to reports in Power BI
- Uses a SQL-based query language

Storm: Open source framework that uses a topology of spouts and bolts to consume, process, and output the results — APACHE

Spark Streaming (Databricks): Open source distributed platform for general data processing, supported Spark language, including Java, Scala, and Python

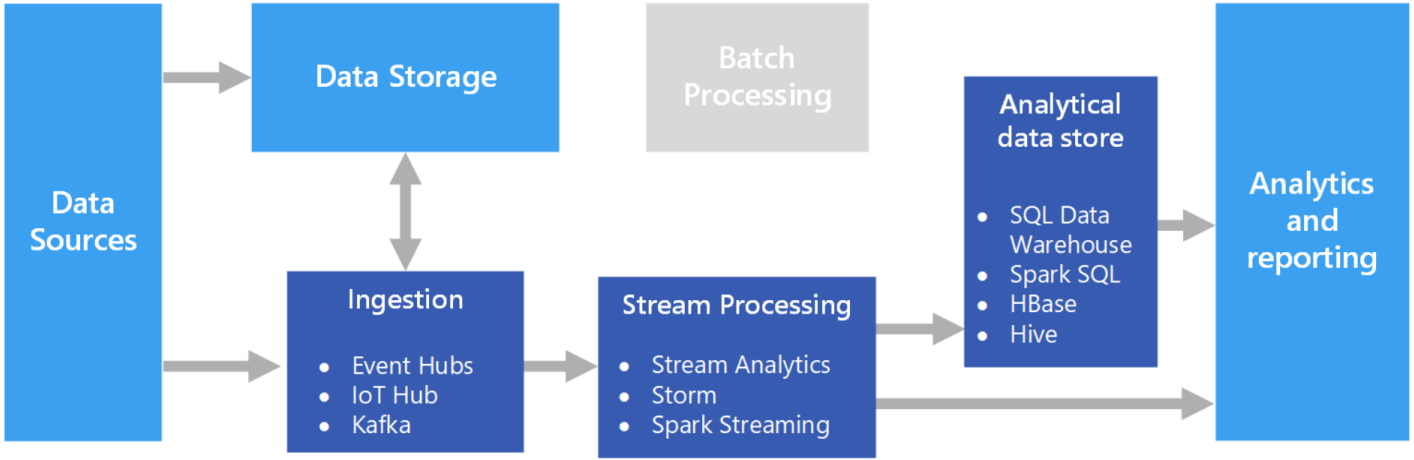
Real Time Processing Architecture



Analytical data store

- **Azure Synapse Analytics:** Relational database
- **Hbase:** NoSQL Store
- **Spark/Hive:** files

Real Time Processing Architecture



Analytics and reporting

- Azure Analysis Services
- **Power BI**
- Microsoft Excel

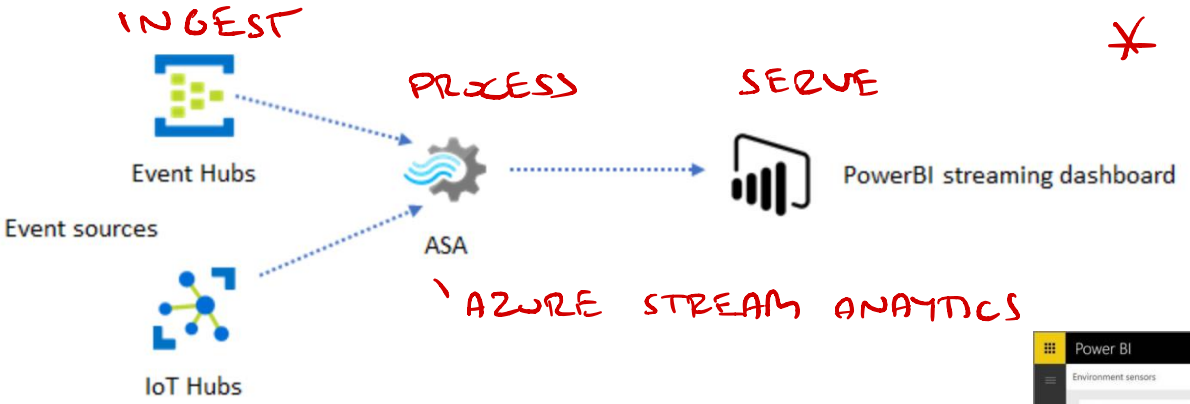
Design and provision compute resources

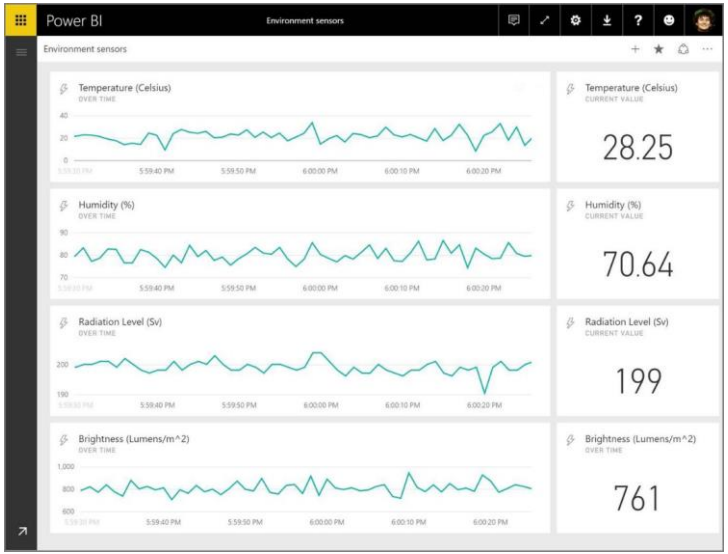
Azure Stream analytics solution and architectural patterns

CONSTANTS

FOR CERT: EVENT / IoT HUBS INGEST, ASA PROCESS

Real time dashboard

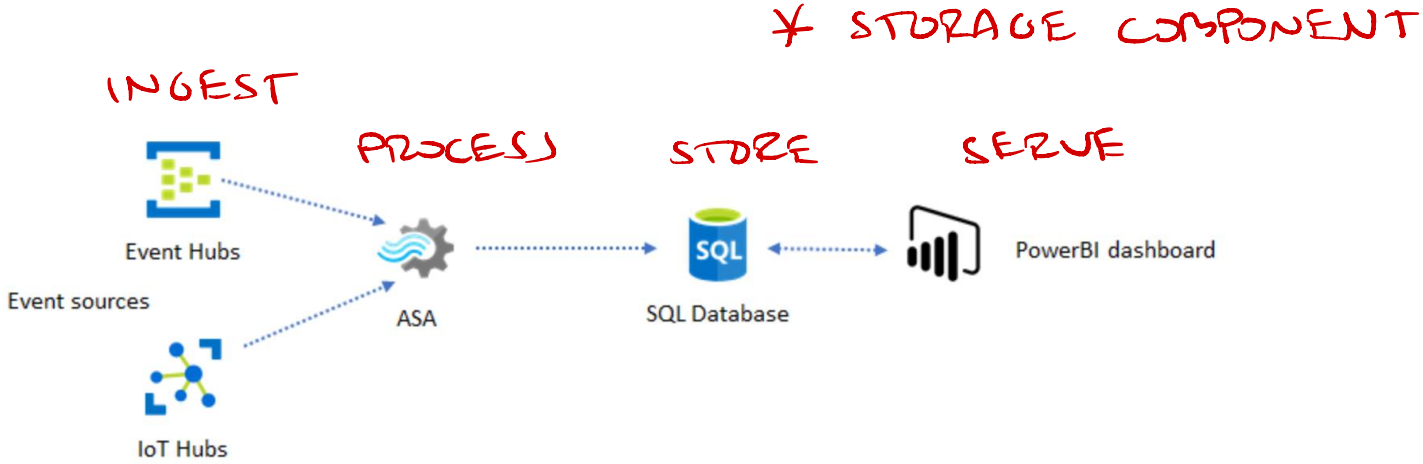




Streaming data can be:

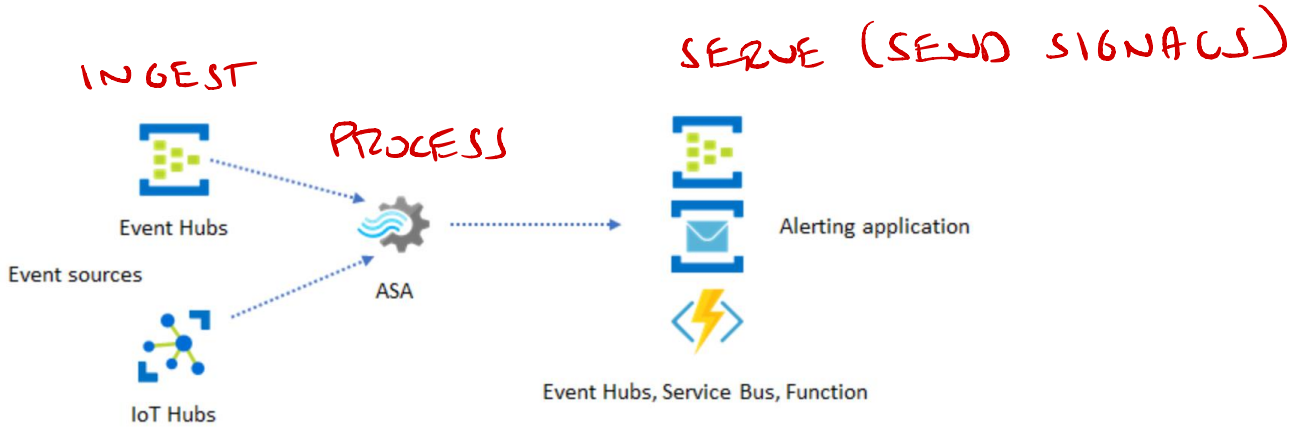
- Factory sensors
- Social media sources
- Service usage metrics
- Or many other time-sensitive data collectors or transmitters

Use SQL for Dashboard



- More flexibility
- Slightly higher latency
- Maximize Power BI capabilities to further slice and dice the data for reports
- Flexibility of using other dashboard solutions, such as Tableau

Real-time insights into your application with event messaging



Why Azure Functions?

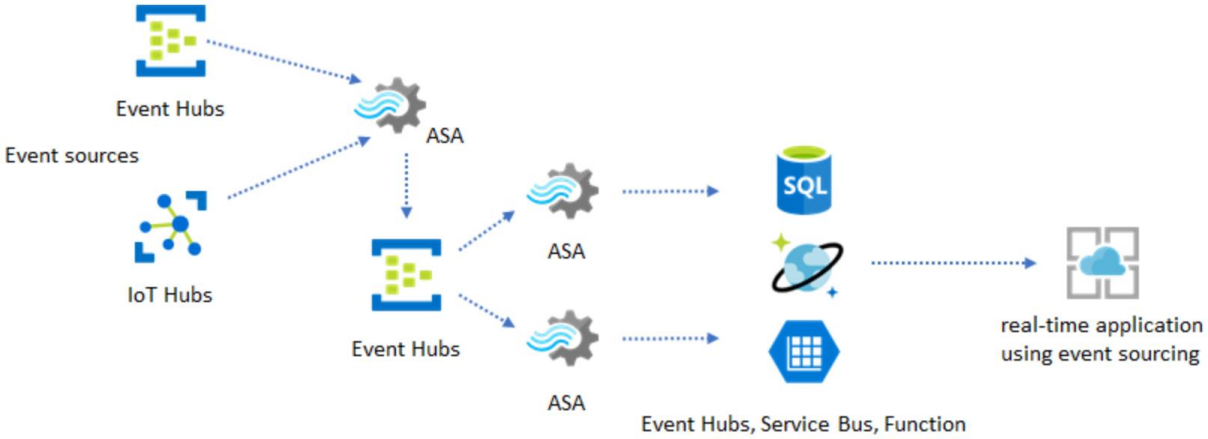
- Custom logic can also be implemented in Azure Functions
- Azure Functions also supports various types of notifications including text and email.

Why Event hubs?

DIRECT CONNECTION

- Most flexible integration point - Azure Data Explorer and Time Series Insights can consume events from Event Hubs
- Services can be connected directly to the Event Hubs sink from Azure Stream Analytics to complete the solution
- Event Hubs is also the highest throughput messaging broker available on Azure for such integration scenarios.

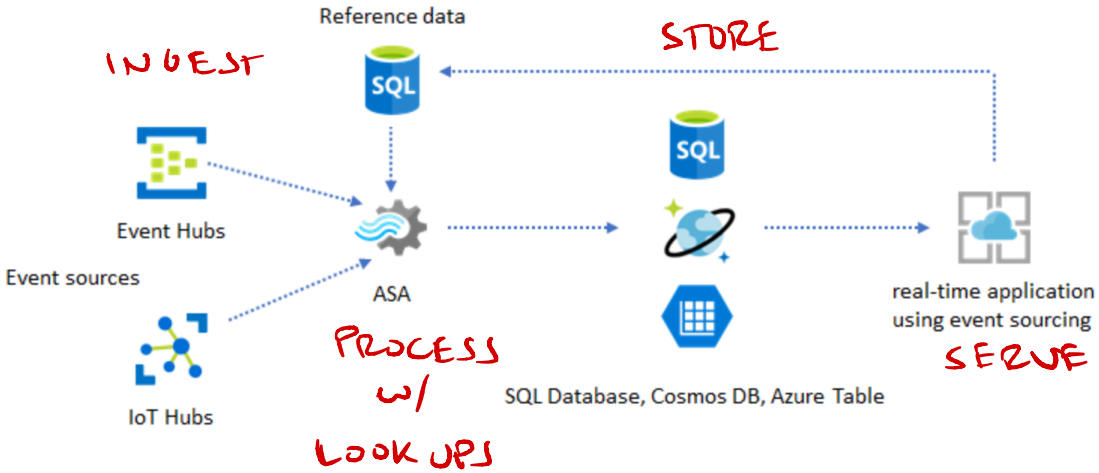
Real-time insights through data stores



Dataflow-based architecture

- Modern high-volume data driven applications often adopt a dataflow-based architecture
- Events are processed and aggregated into data stores by Azure Stream Analytics
- The application layer interacts with data stores using the traditional request/response pattern.

Reference data for application customization

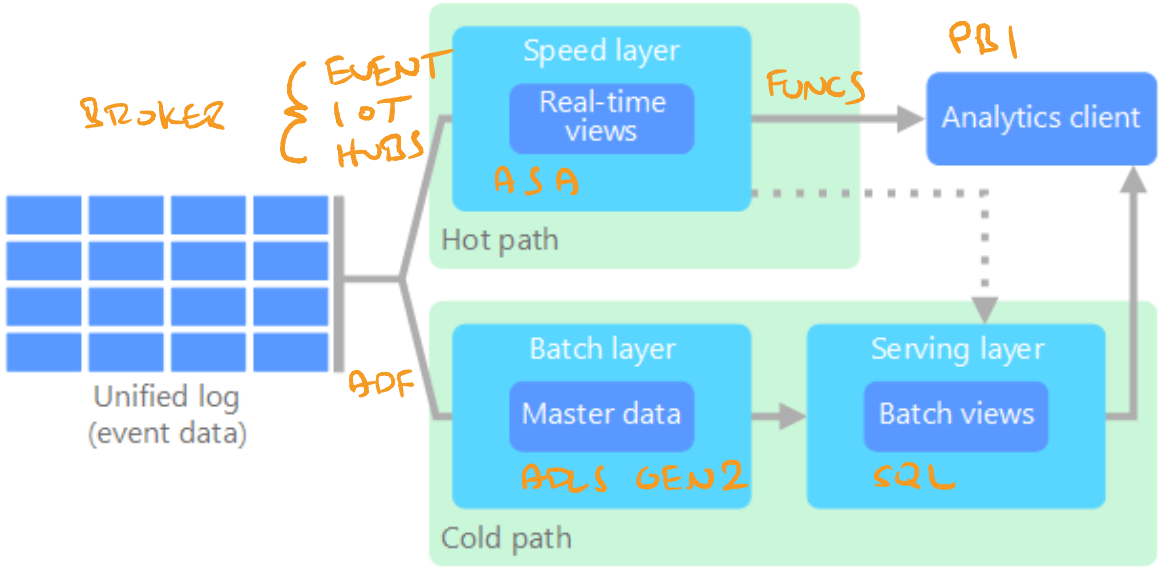


Reference data

- Reference data feature is designed specifically for end-user customization like alerting threshold and processing rules
- Reference data (also known as a lookup table) is a finite data set that is static or slowly changing in nature

Lambda architecture

SINGLE SYSTEM TO PROCESS REAL-TIME (HOT) + ALSO STORE BATCH (COLD)

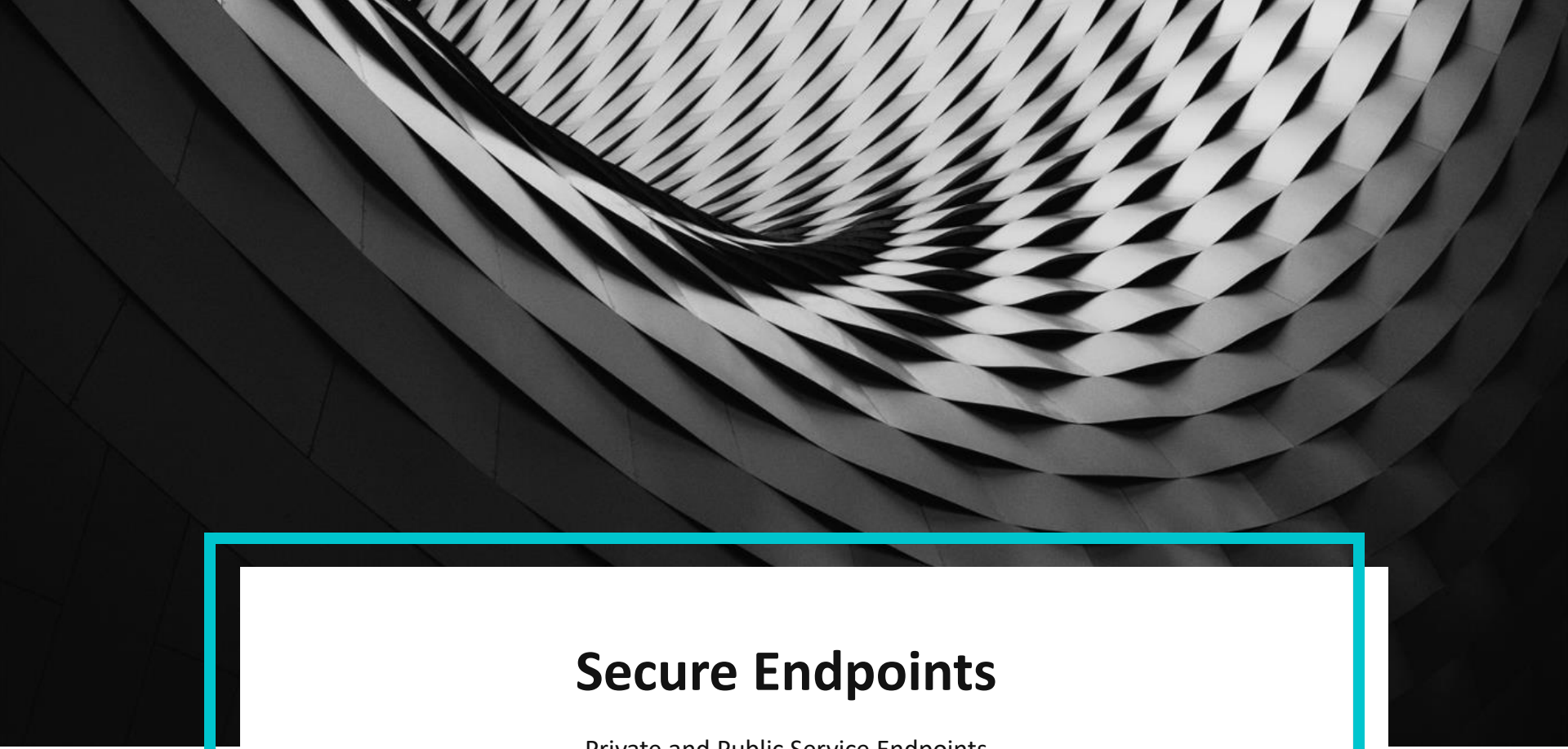


Batch layer (cold path)

- Stores all of the incoming data in its raw form Performs batch processing on the data.
- Result stored as a batch view.

Speed layer (hot path)

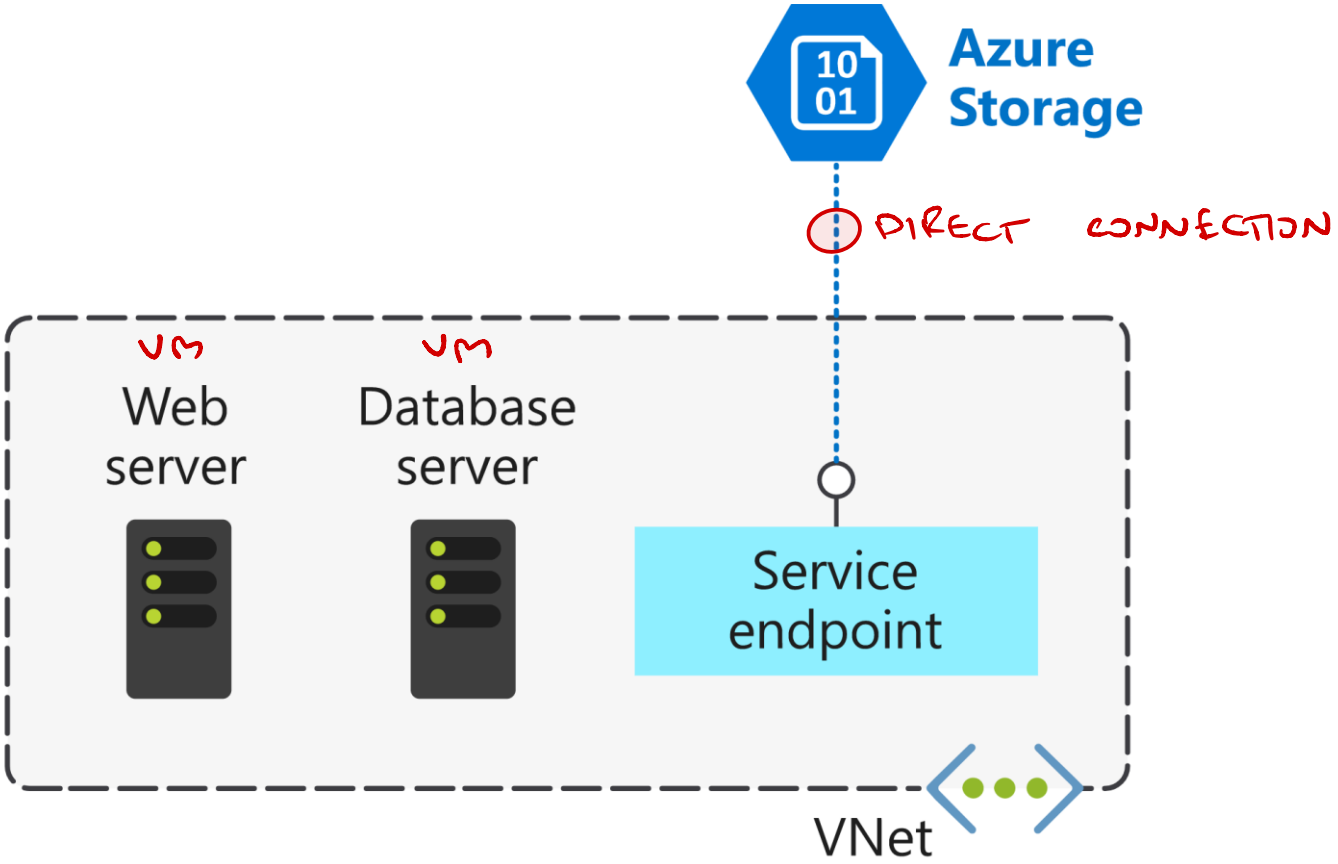
- Analyzes data in real time
- Designed for low latency
- Low accuracy.



Secure Endpoints

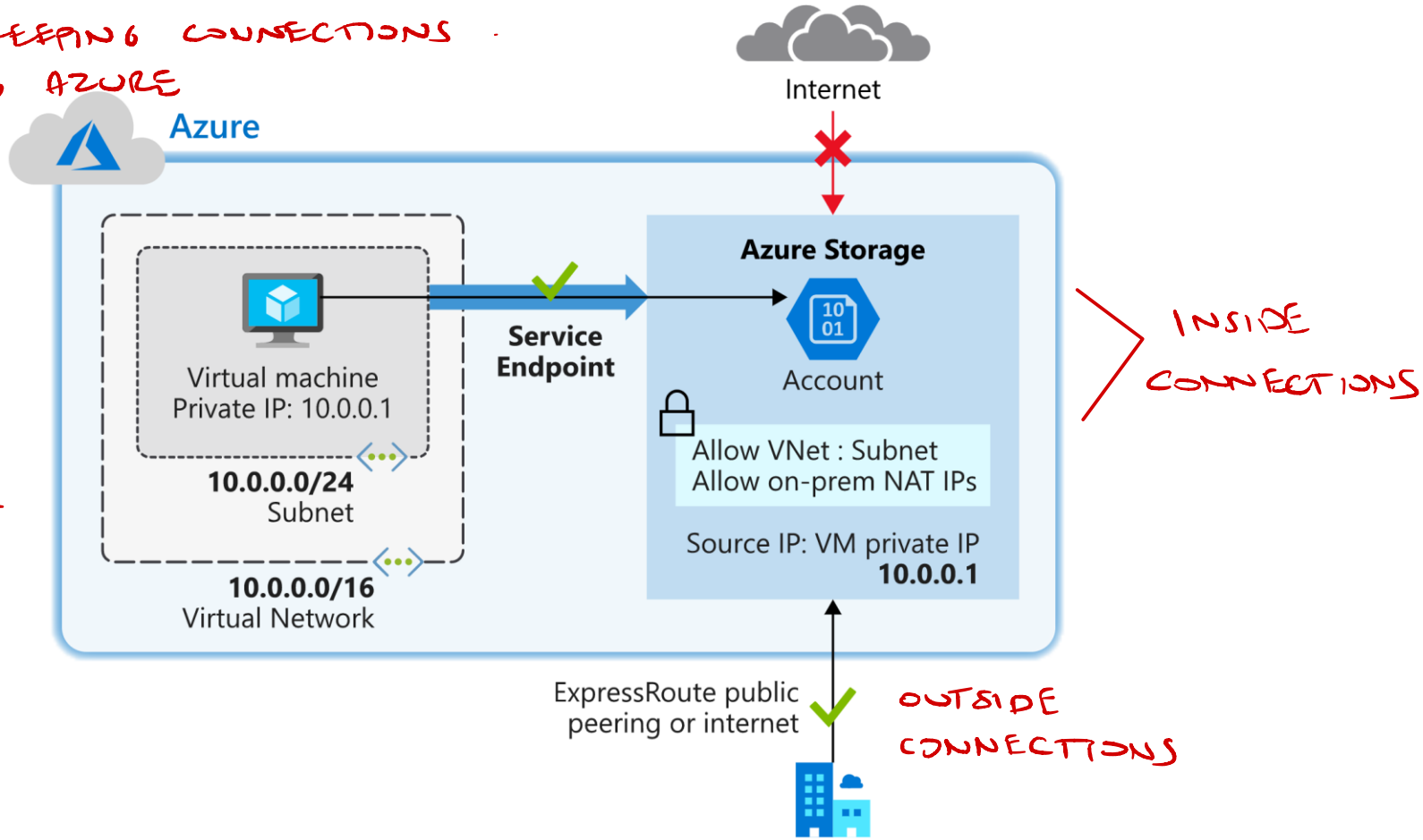
Private and Public Service Endpoints

Virtual network service endpoints



Service endpoints

SECURE BY KEEPING CONNECTIONS INTERNAL TO AZURE



USE PRIVATE LINK CENTER RESOURCE FOR PRIVATE ENDPOINTS

INSIDE CONNECTIONS

OUTSIDE CONNECTIONS