DP-201

Designing an Azure Data Solution

By Vineela

DP-201 skills measured

Design Azure Data Storage Solutions (40-45%)

Azure Data Storage
- Diff data types and storage types

Implement non-relational data stores
- Non-relational data stores (Blob Storage)
- Azure Cosmos DB
- Azure Data Lake

Implement relational data stores
- Azure SQL Server
- Azure Synapse Analytics Service

Design Data Processing Solutions (25-30%)

Develop batch processing solutions
- Azure Data Factory
- Azure Databricks

Develop streaming solutions
- Azure Streaming Service
- Azure Databricks

Design for Data Security and Compliance (25-30%)

Design security for source data access
- Secure endpoints (private/public)
- Authentication mechanism
  - Access keys
  - shared access signatures (SAS)
  - Azure Active Directory (Azure AD)

Design security for data policies and standards
- Data encryption for data at rest and in transit
- Data auditing and data masking
- Data privacy and data classification
- Data retention policy
- Achieving strategy
- Plan to purge data based on business requirements

Intended Audience

- Anyone who wants to clear DP-201 exam
- Anyone who wants to learn Design and Architecture of Azure Data technologies

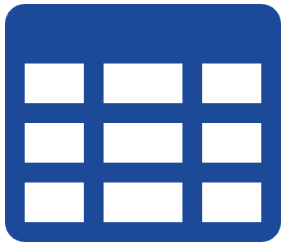
Prerequisite

- Good foundational understanding of Azure Data Service (DP-200 course)
- Good overlap of syllabus, concepts and lessons
- If you are struggling, go back to DP-200 course

Data Types

Four types of Data

Types of Data



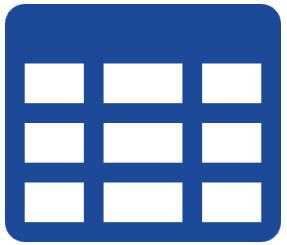
Structure Data

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)

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

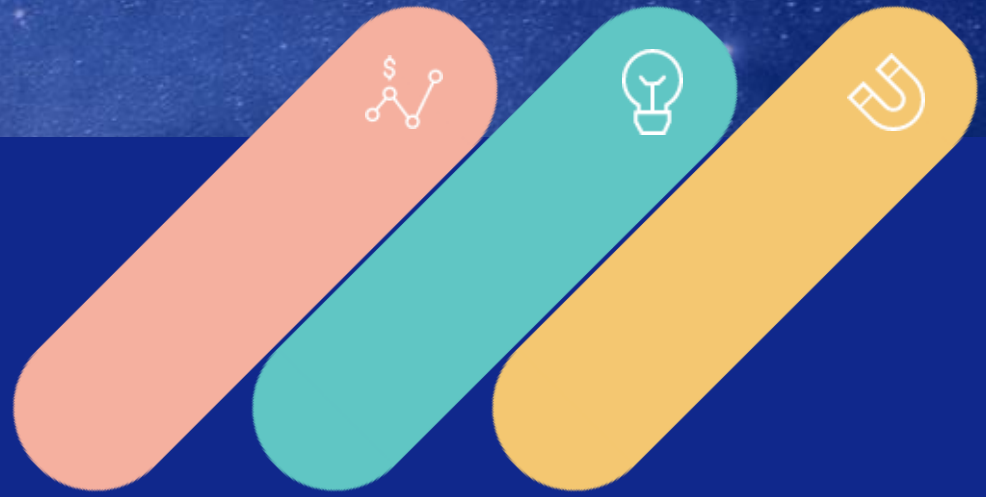
Streaming Data

- 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

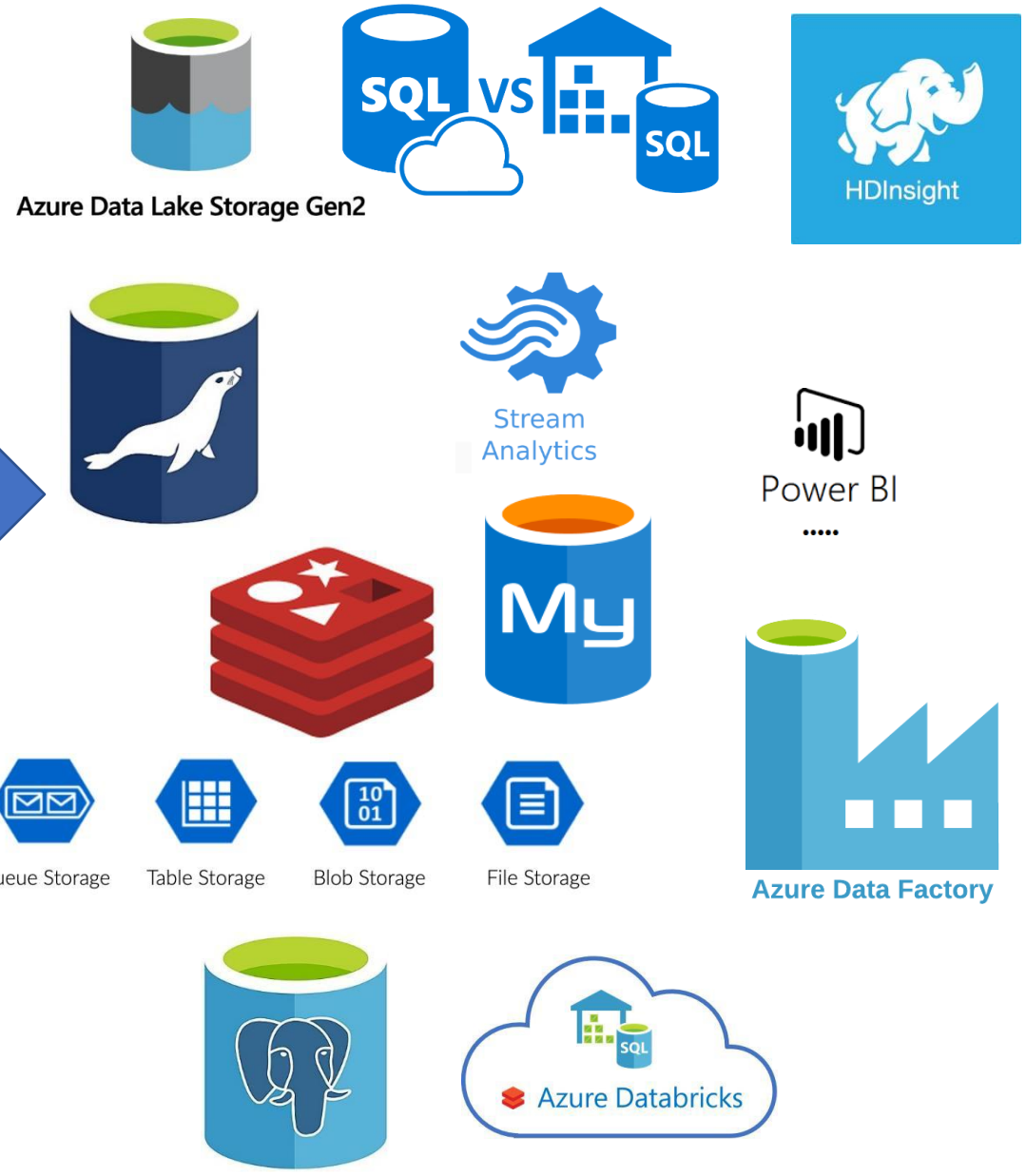
Data store

Understand data store models

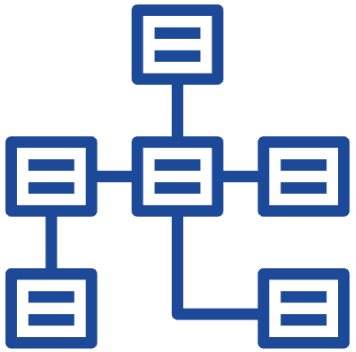


Why we need different data store?

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



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)

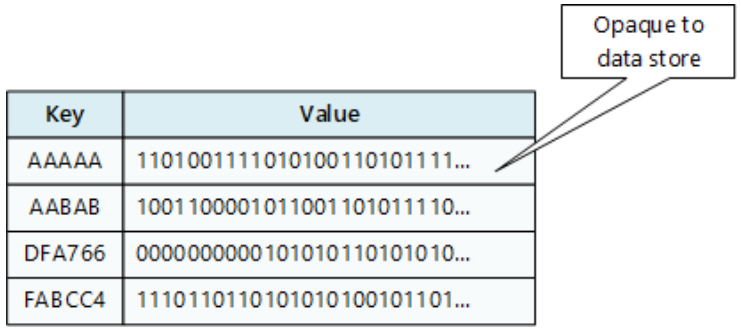
Examples

- 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



Data Structure

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

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	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.

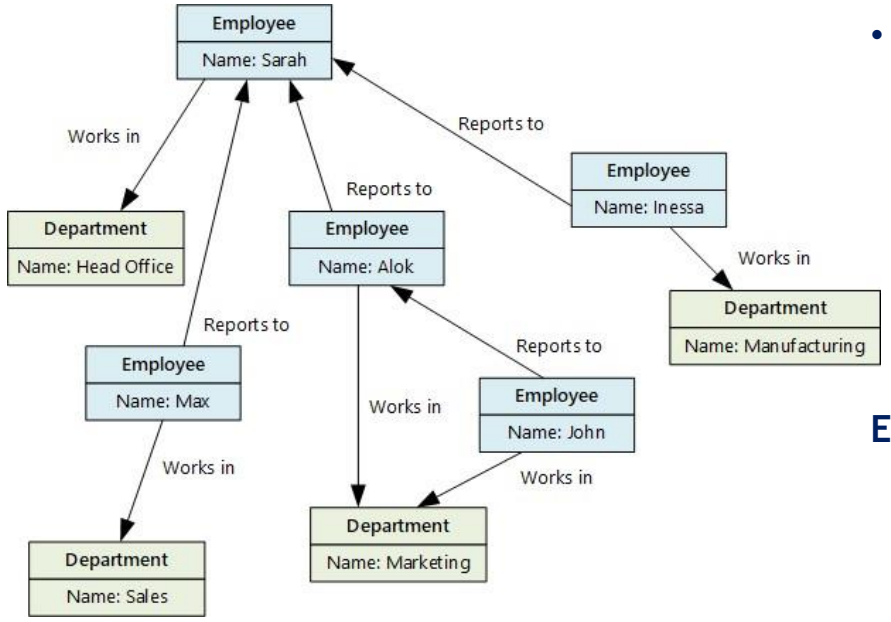
Examples

- 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

Operation

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

Examples

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

Azure services

- Azure Cosmos DB Gremlin API
- SQL Server

Column-family databases

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.

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

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.

Examples

- Enterprise data warehouse

Operation

- Data analytics
- Enterprise BI

Azure services

- Azure Synapse Analytics
- 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

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

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

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

Object storage

- Azure Blob Storage
- Azure Data Lake Storage Gen2

Search Engine Databases

- Azure Search

Time series databases

- Azure Time Series Insights

General Consideration

Data Stores

Functional requirements

- Data format
- Data size
- Scale and structure
- Data relationships
- Consistency model
- Schema flexibility
- Concurrency
- Data movement
- Data lifecycle

Non-functional requirements

- Performance and scalability
- Reliability
- Replication
- Limits

Management and cost

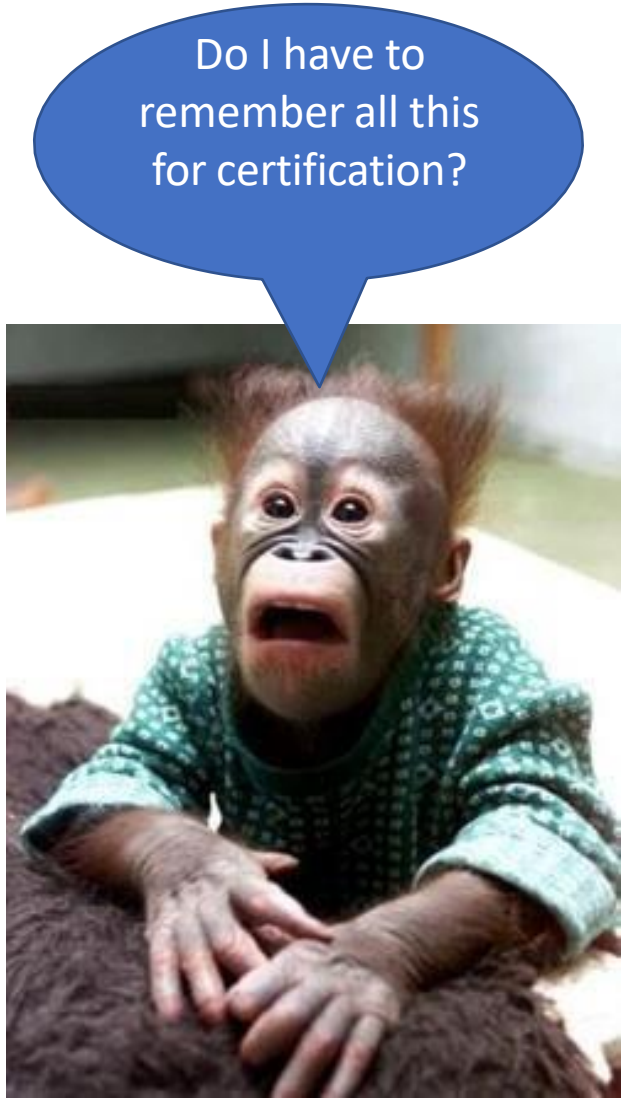
- Managed service
- Region availability
- Portability
- Licensing
- Overall cost
- Cost effectiveness

Security

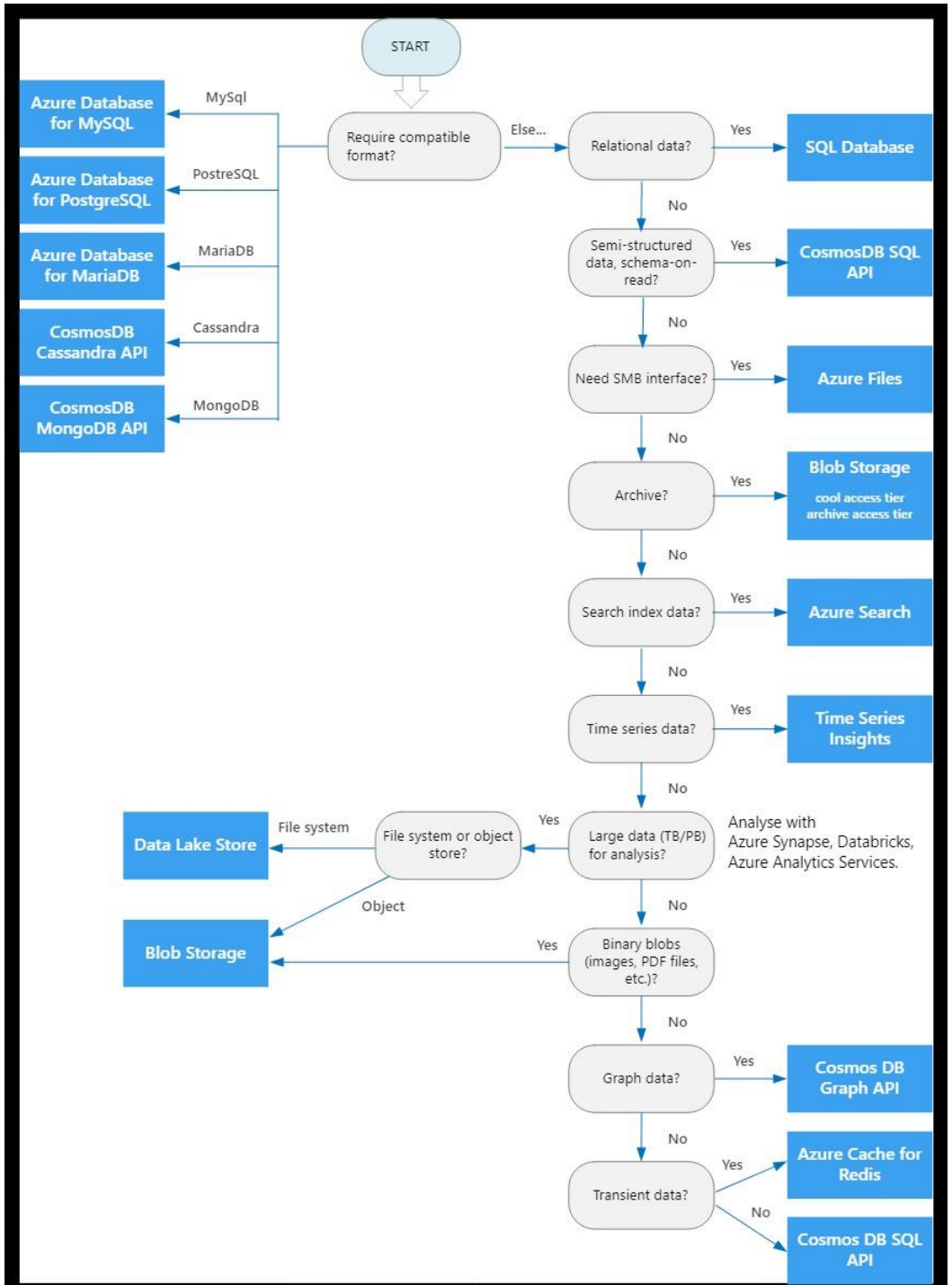
- Security
- Auditing
- Networking requirements

DevOps

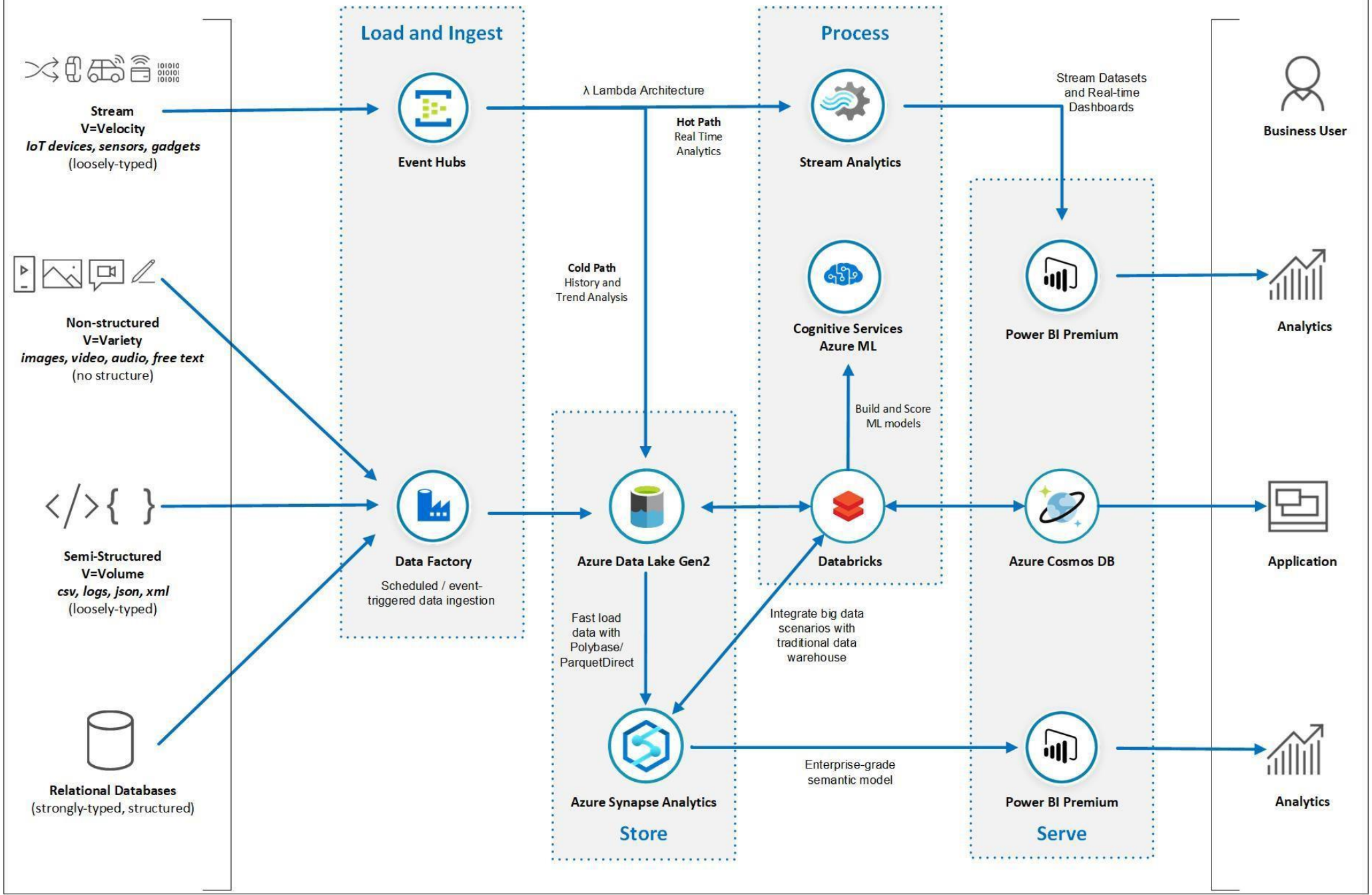
- Skill set
- Clients



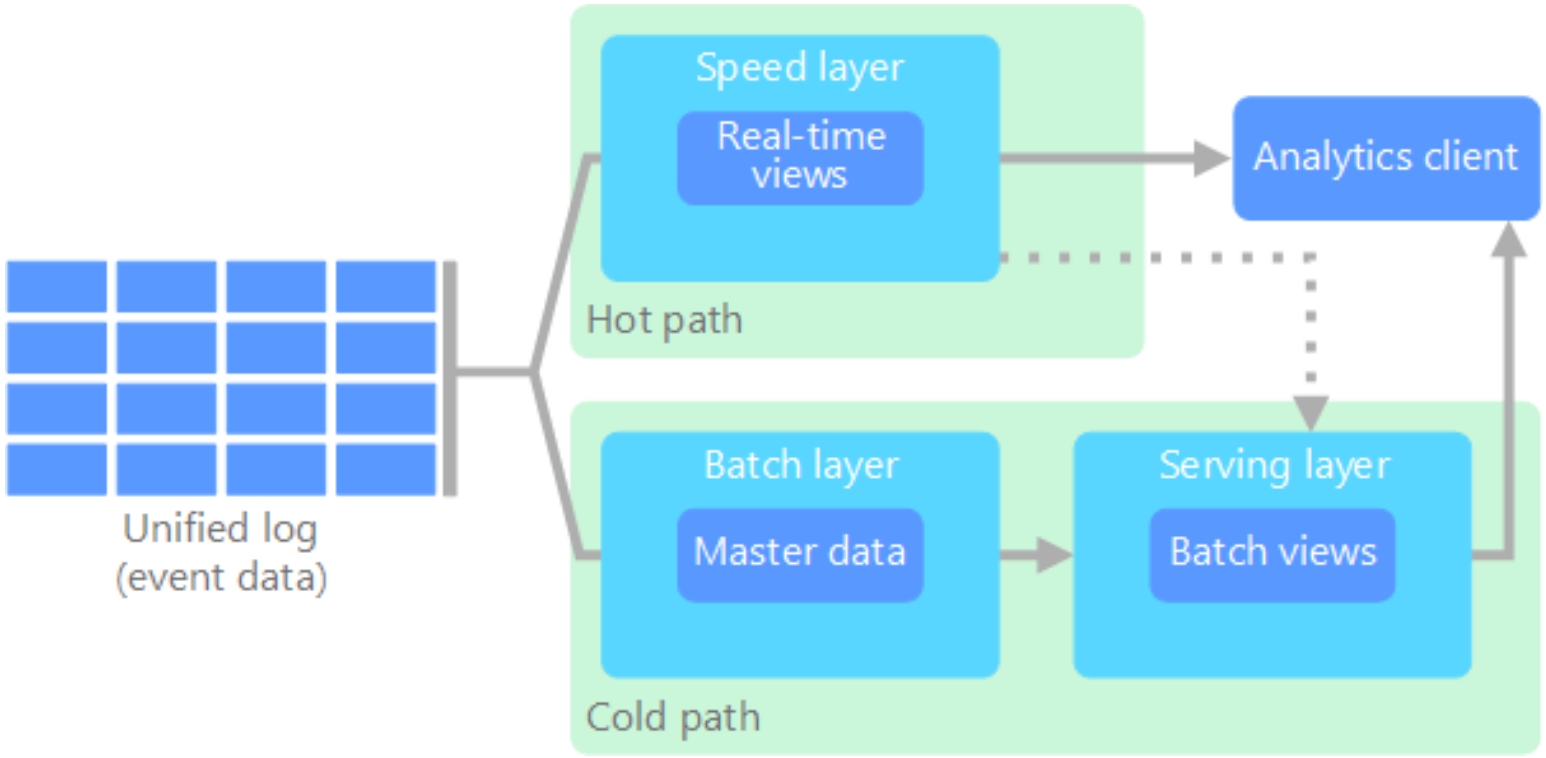
How to choose Database



Modern Data Platform Reference Architecture



Lambda architecture



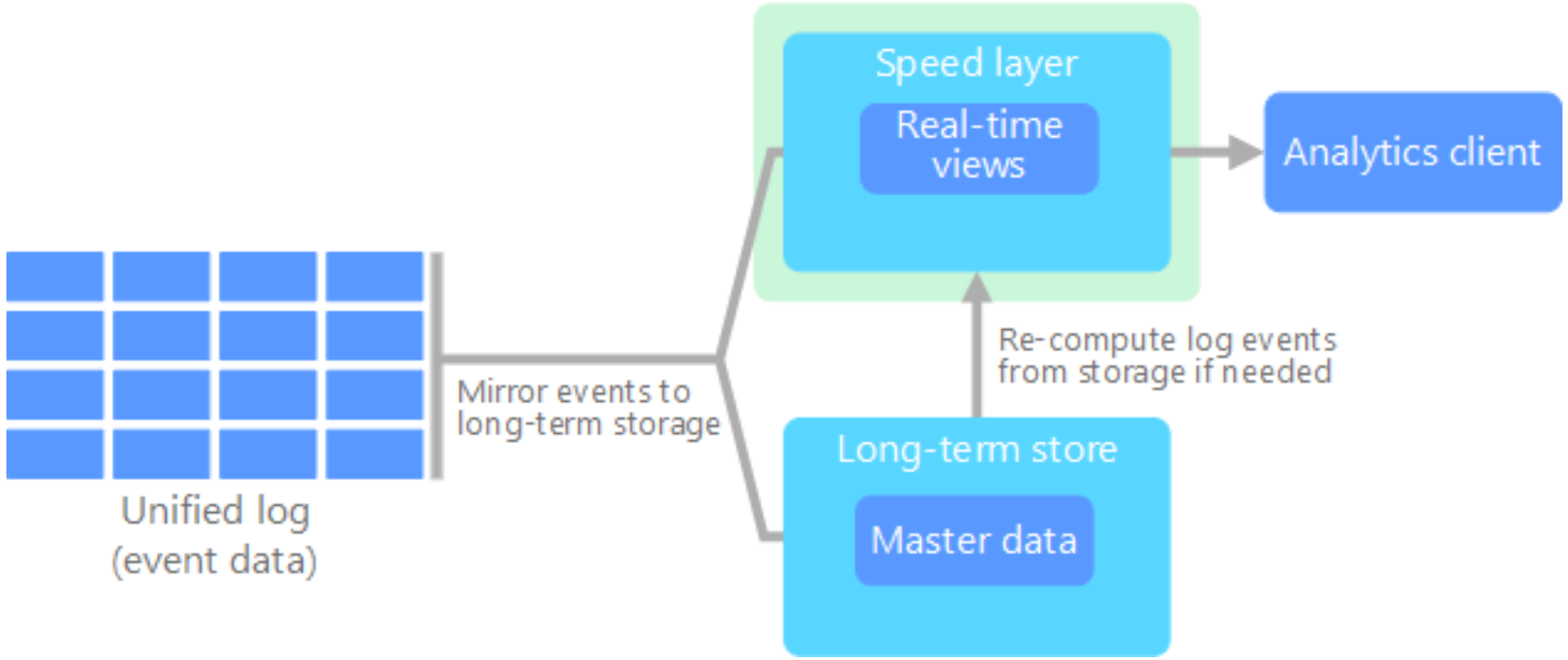
Batch layer (cold path)

- **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.**

Kappa architecture



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.

Data Continuity and Availability

High Availability

- Making a service available within a region
- No expected data loss

Disaster Recovery

- Recovery when whole region goes down
- Typically some data loss

What can go wrong?

Disaster Recovery

Hardware

CPU, memory, controllers, disk, server, rack, AC, power

Software

Application, OS, bugs, logical corruptions

Connectivity

Redundant network connectivity

What can go wrong?

Disaster Recovery

Entire site

Lose power, lose water, bad weather

Entire region

Flooding, earthquake

Human error

People make mistakes

RTO and RPO

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

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.

Recovery Point Objective (RPO)

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

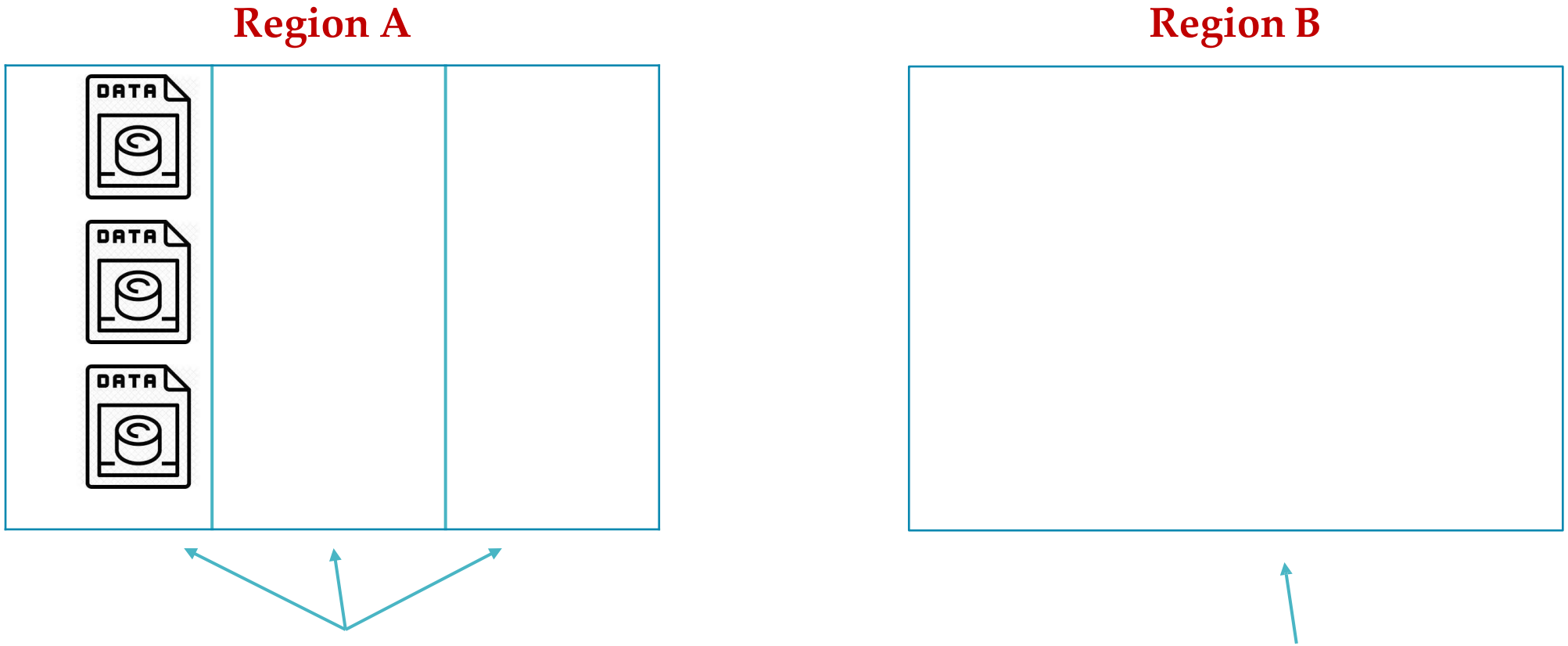
For example; one standalone database with hourly backups provides an RPO of 60 minutes.

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

Azure Storage

High Availability and Disaster recovery options

Locally Redundant Storage (LRS)



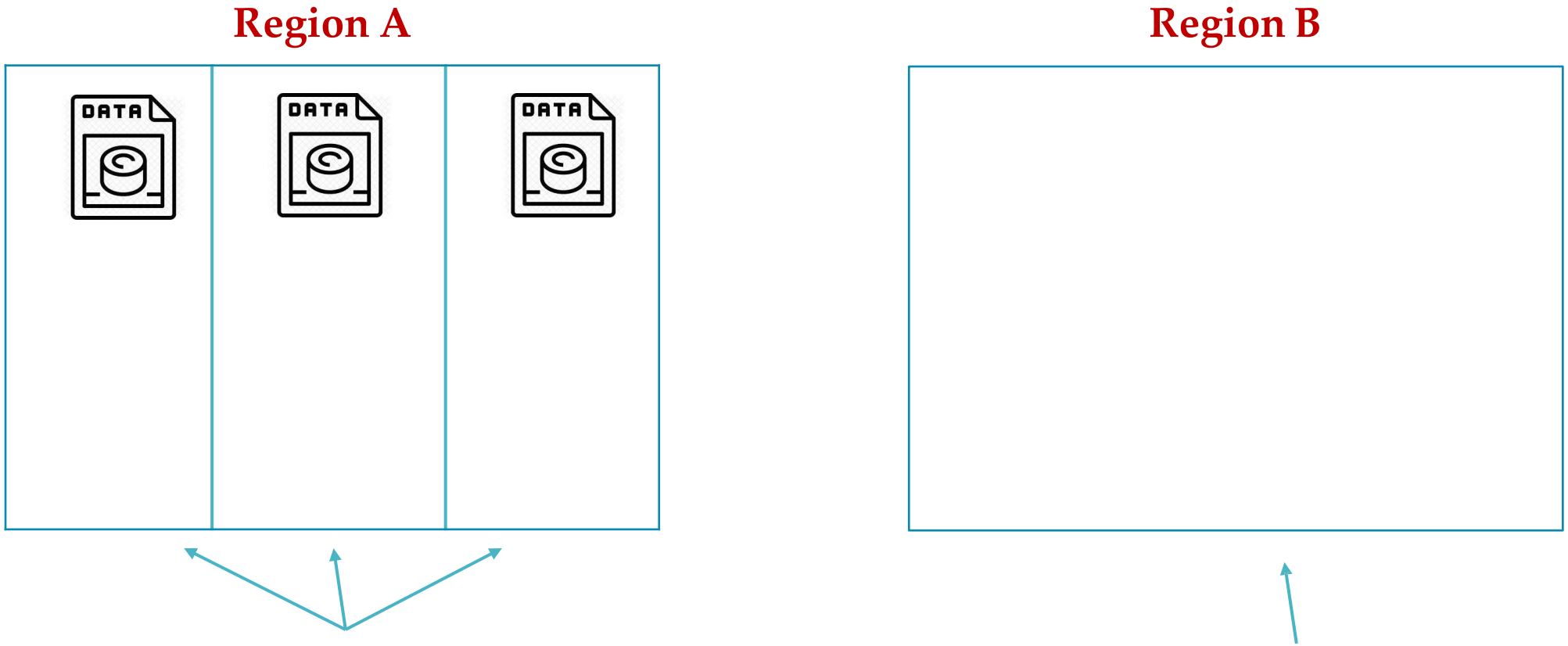
Storage Clusters

Each cluster is physically separate in what's called an availability zone, with its own separate utilities and networking.

Region B

Hundreds of miles away from the primary region to prevent data loss in the event of a natural disaster.

Zone Redundant Storage (ZRS)



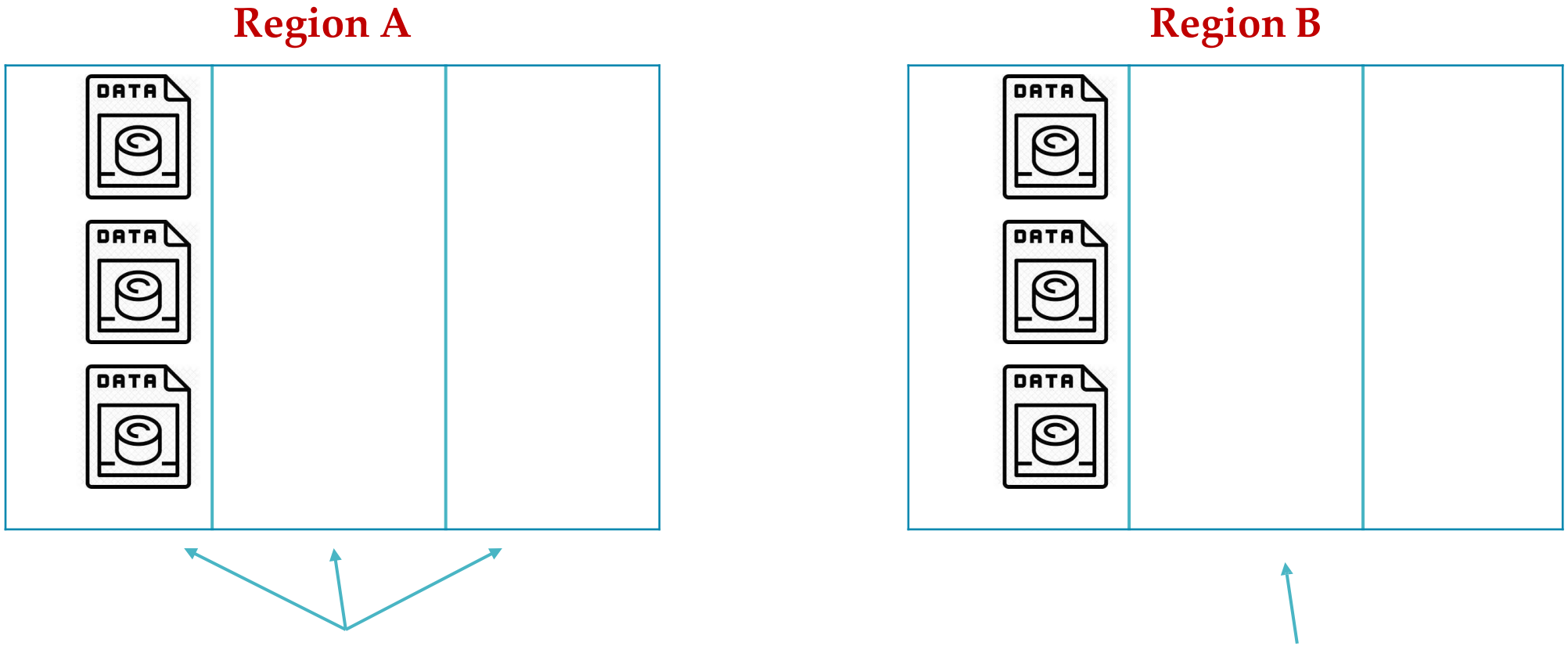
Storage Clusters

Each cluster is physically separate in what's called an availability zone, with its own separate utilities and networking.

Region B

Hundreds of miles away from the primary region to prevent data loss in the event of a natural disaster.

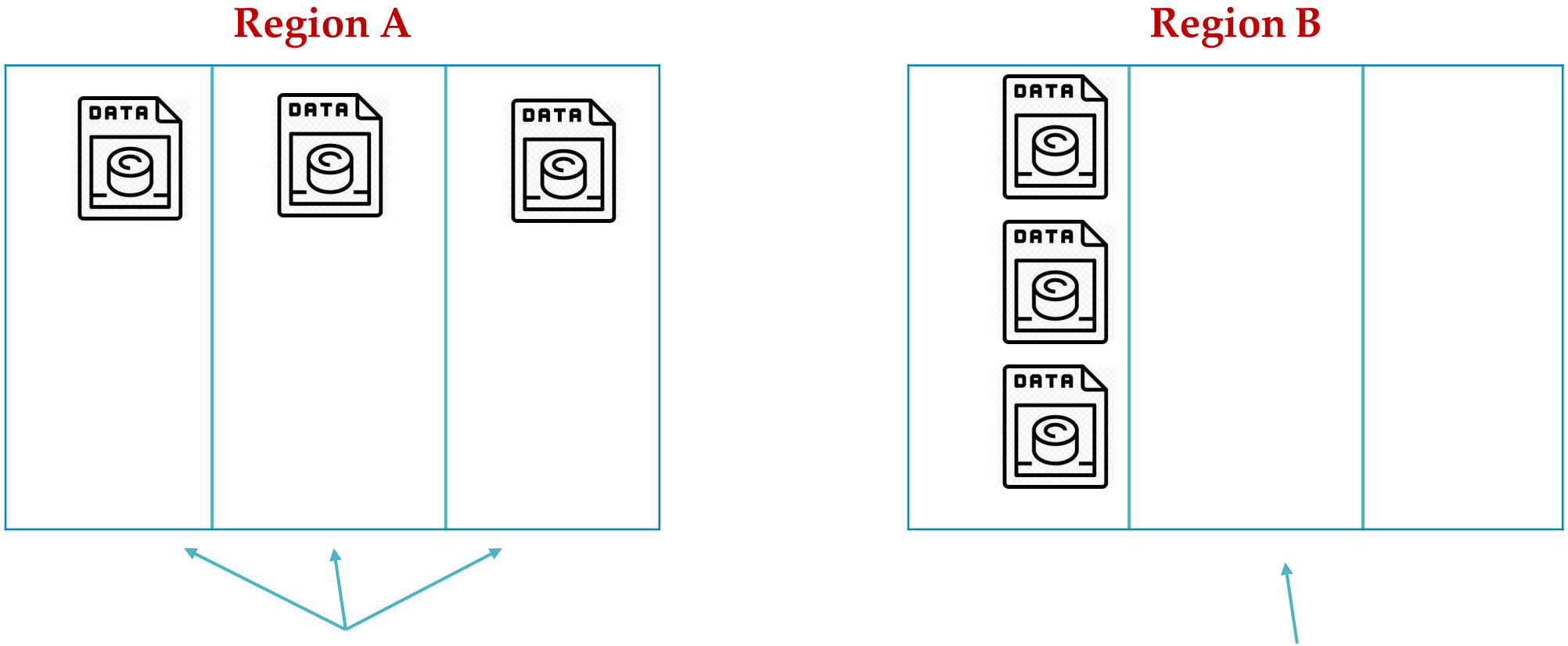
Geo Redundant Storage (GRS)



Storage Clusters
Each cluster is physically separate in what's called an availability zone, with its own separate utilities and networking.

Region B
Hundreds of miles away from the primary region to prevent data loss in the event of a natural disaster.

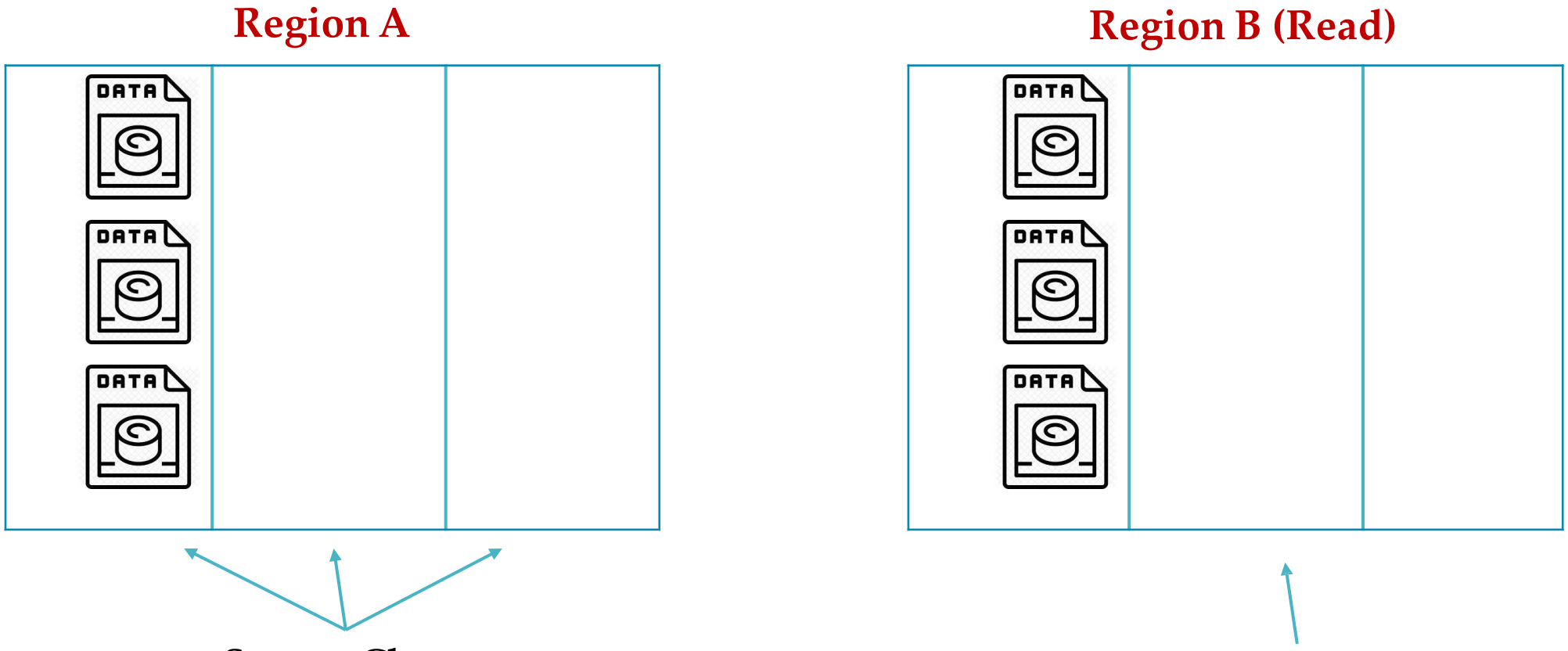
Geo Zone Redundant Storage (GZRS)



Storage Clusters
Each cluster is physically separate in what's called an availability zone, with its own separate utilities and networking.

Region B
Hundreds of miles away from the primary region to prevent data loss in the event of a natural disaster.

Read access geo Redundant Storage (RA-GRS)



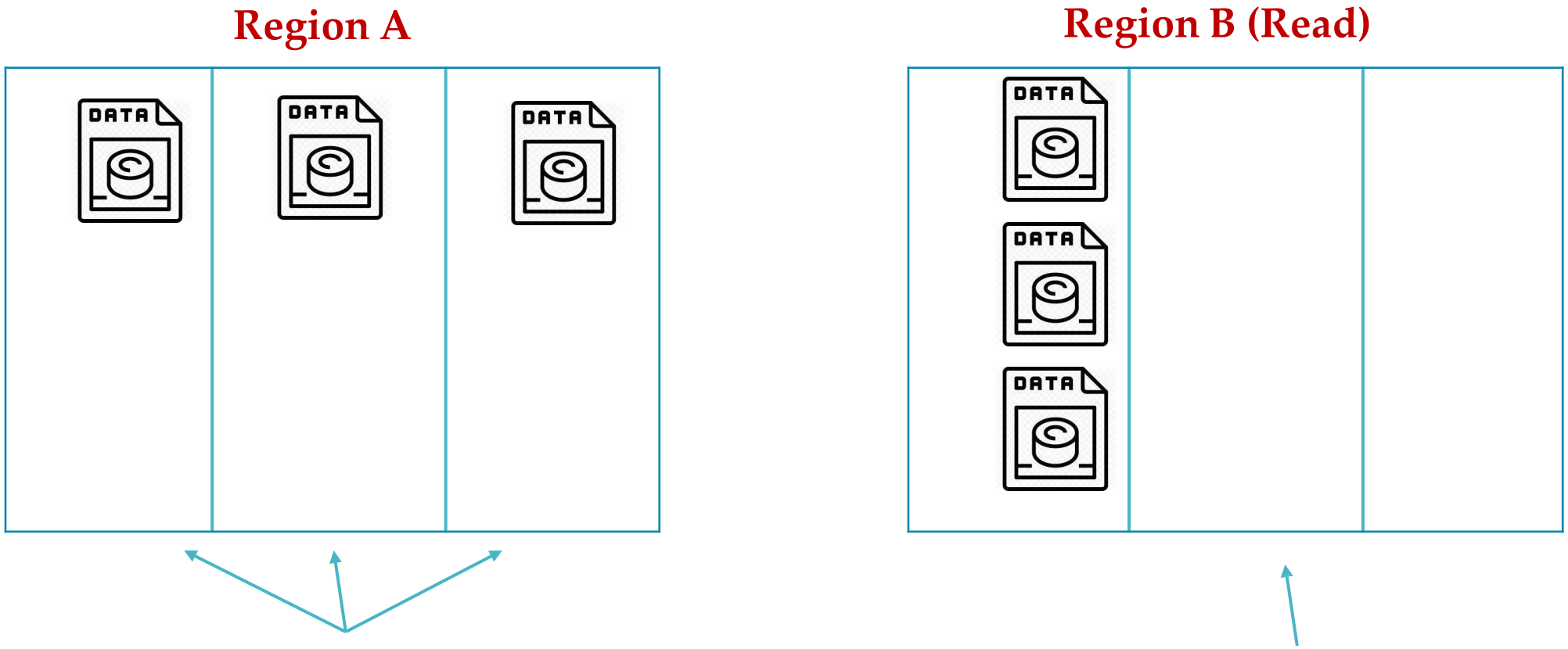
Storage Clusters

Each cluster is physically separate in what's called an availability zone, with its own separate utilities and networking.

Region B

Hundreds of miles away from the primary region to prevent data loss in the event of a natural disaster.

Read access Geo Zone Redundant Storage (RA-GZRS)



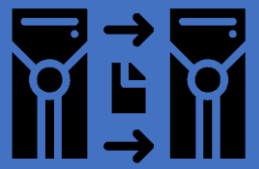
Storage Clusters
Each cluster is physically separate in what's called an availability zone, with its own separate utilities and networking.

Region B
Hundreds of miles away from the primary region to prevent data loss in the event of a natural disaster.

Durability and availability by outage scenario

The following table indicates whether your data is durable and available in a given scenario, depending on which type of redundancy is in effect for your storage account:

Outage scenario	LRS	ZRS	GRS/RA-GRS	GZRS/RA-GZRS
A node within a data center becomes unavailable	Yes	Yes	Yes	Yes
An entire data center (zonal or non-zonal) becomes unavailable	No	Yes	Yes[1]	Yes
A region-wide outage occurs in the primary region	No	No	Yes[1]	Yes[1]
Read access to the secondary region is available if the primary region becomes unavailable	No	No	Yes (with RA-GRS)	Yes (with RA-GZRS)



Azure Storage Outages

Detection: Subscribe to Azure Service health dashboard.

LRS or ZRS

- Wait for recovery

GRS or RA-GRS or GZRS or RA-GZRS

- Manual failover
- Copy data from secondary to some other region
  - Use tools such as AzCopy, Azure PowerShell, and the Azure Data Movement library

Cosmos DB – HA and DR Options

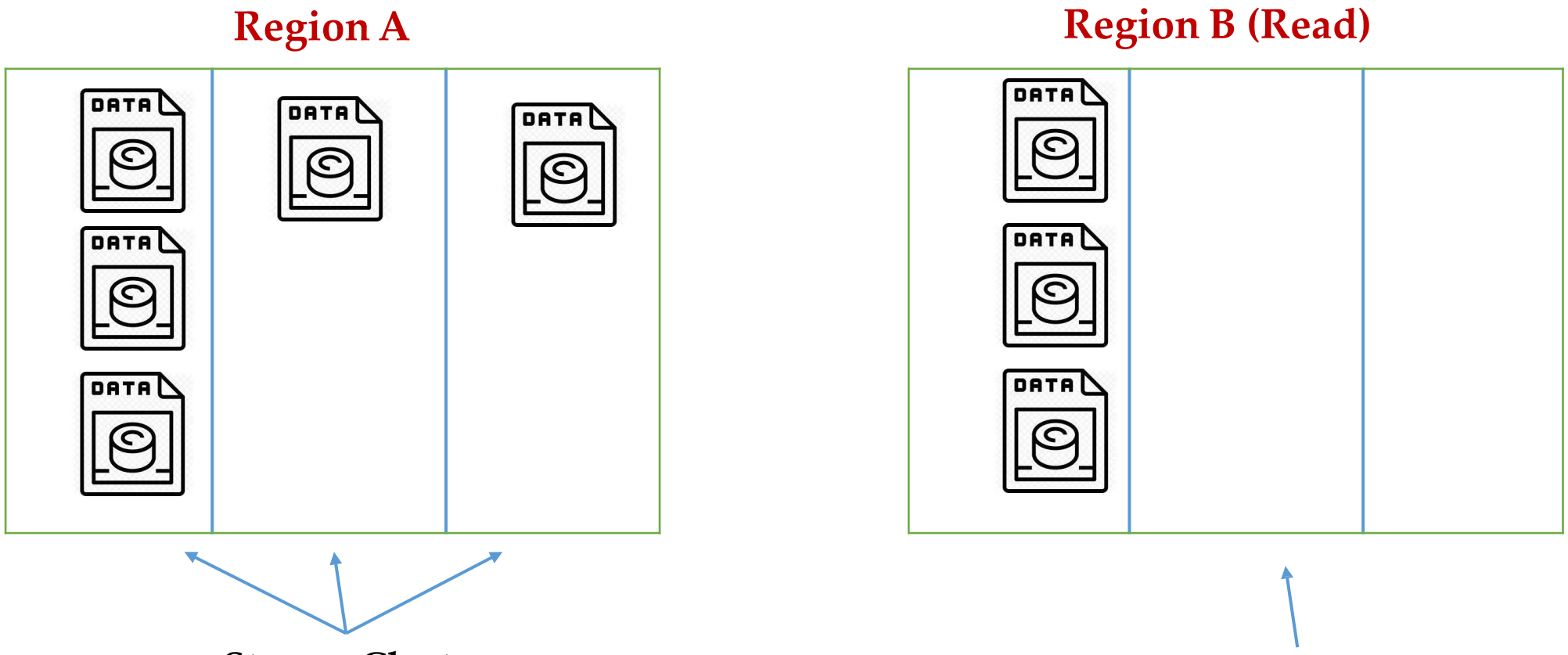
Agenda

- Cosmos DB – High availability and Disaster recovery option
  - Local/Zone/Global – Replication
  - Backup and Restore of Cosmos DB

Prerequisite

- Global distribution
- Multi-master
- Manual vs Automatic Failover

Cosmos DB – HA and DR Options



Storage Clusters

Each cluster is physically separate in what's called an availability zone, with its own separate utilities and networking.

Region B

Hundreds of miles away from the primary region to prevent data loss in the event of a natural disaster.

Cosmos DB – HA and DR Options

Single Region

- Data within a container is durably committed by a majority of replica members within the replica set
- Availability Zone support - replicas are placed across multiple zones within a given region

Multi Region – Multi write

- Regional failovers are instantaneous and don't require any changes from the application.

Multi Region – Single write (read region outage)

- No changes are required in your application code
- The impacted region is automatically disconnected and will be marked offline.
- The Azure Cosmos DB SDKs will redirect read calls to the next available region
- When the impacted read region is back online it will automatically sync with the current write region and will be available again to serve read requests.

Multi Region – Single write (write region outage)

- Manual failover
- Automatic Failover enable – automatically promote a secondary region (no action required)
- When impacted region back only – Lost data can be recovered using conflict feed
- you can switch back to the recovered region as the write region using PowerShell, Azure CLI or Azure portal.

Cosmos DB Backup and Restore

- Accidentally deleted or update data? – use backups
- Backups are completely automated
- No performance impacted, No additional RUs or cost
- Default:
  - Interval: every 4 hours
  - Retention: 8 hours
  - Interval and Retention can be changed but max 2 backups are possible
- Stored separately in a blob storage service
- Backups stored in same region, and also replicated to paired region
- How to Restore? - Raise a ticket to support team

  Custom data backup – solutions can be implemented via Azure Data Factory or via the Cosmos DB change feed
- **Accidently deleted your data? – You have 8 hours**

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**
  - **Allow orders from various web portals**
  - **Store data from variety of sources**
  - **Provide reporting for multiple business silos**
- **Options:**
  - **Cosmos DB**
  - **Data Lake Gen 2**
  - **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**

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

Azure SQL DB

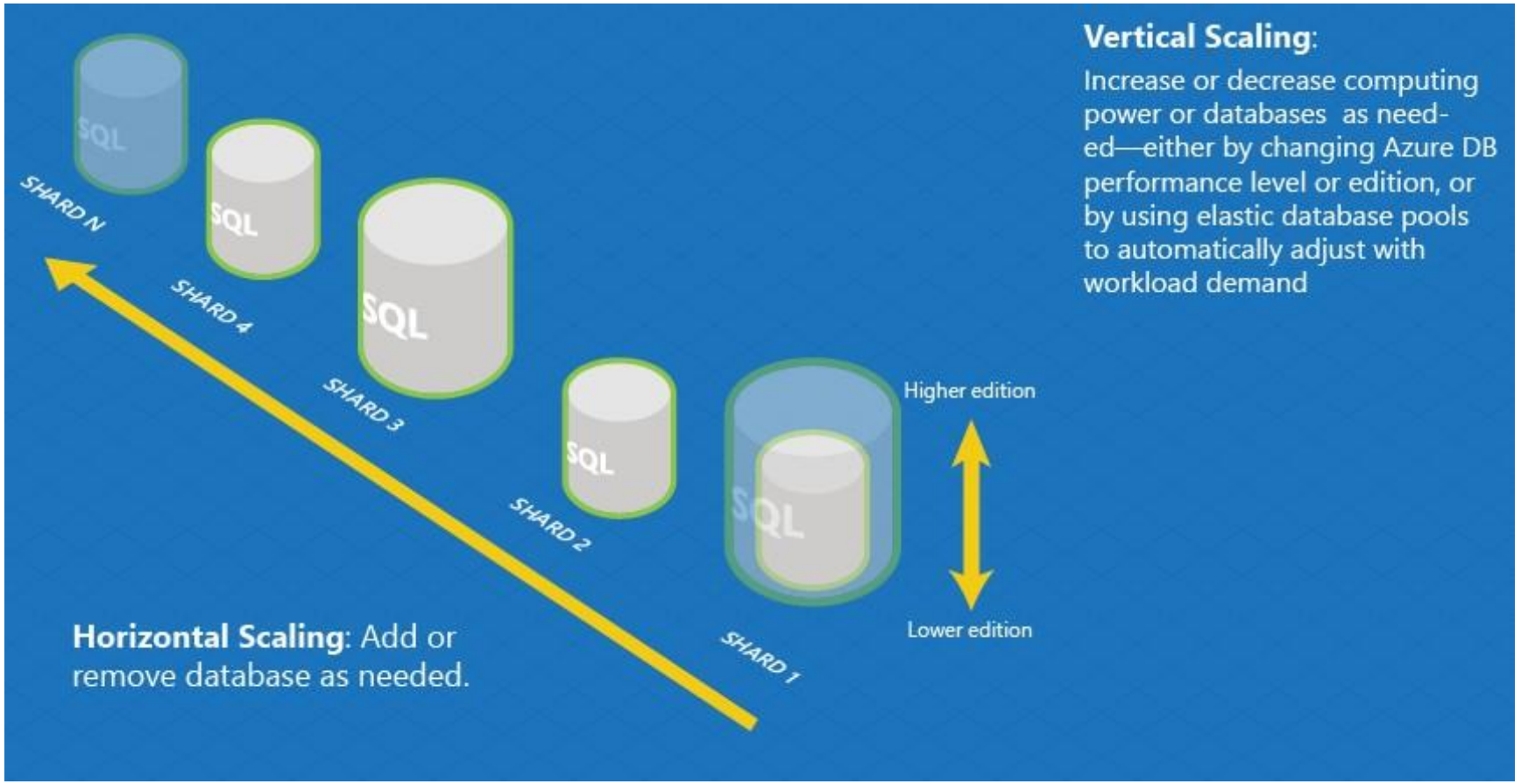
Scaling

Prerequisite: Purchasing model and service tier

Azure SQL Database Scaling

Azure SQL Database supports two types of scaling:

- **Vertical scaling:** Scale up or down the database by adding more compute power.
- **Horizontal scaling:** Scale out or Add more databases and to shard your data into multiple database nodes.



Vertical vs Horizontal Scaling

Azure SQL Database supports two types of scaling:

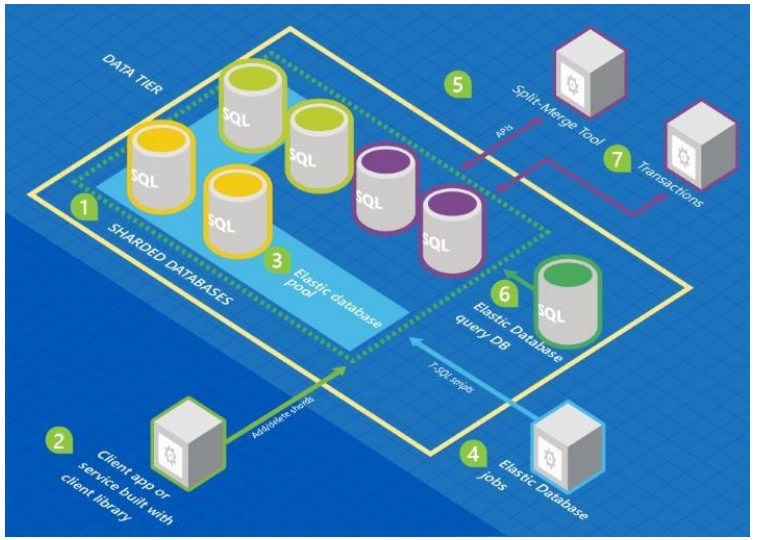
- **Vertical scaling:**
  - Scale up or down the database by adding more compute power.
  - CPU Power, Memory, IO throughput, and storage
  - DTU and vCore models to scale
  - Dynamic Scalability (Note: this is not auto-scale)
  - Any change that you made will be almost instant .

- **Horizontal scaling:** Scale out or Add more databases and to shard your data into multiple database nodes.

Azure SQL Database Scaling

Read Scale Out

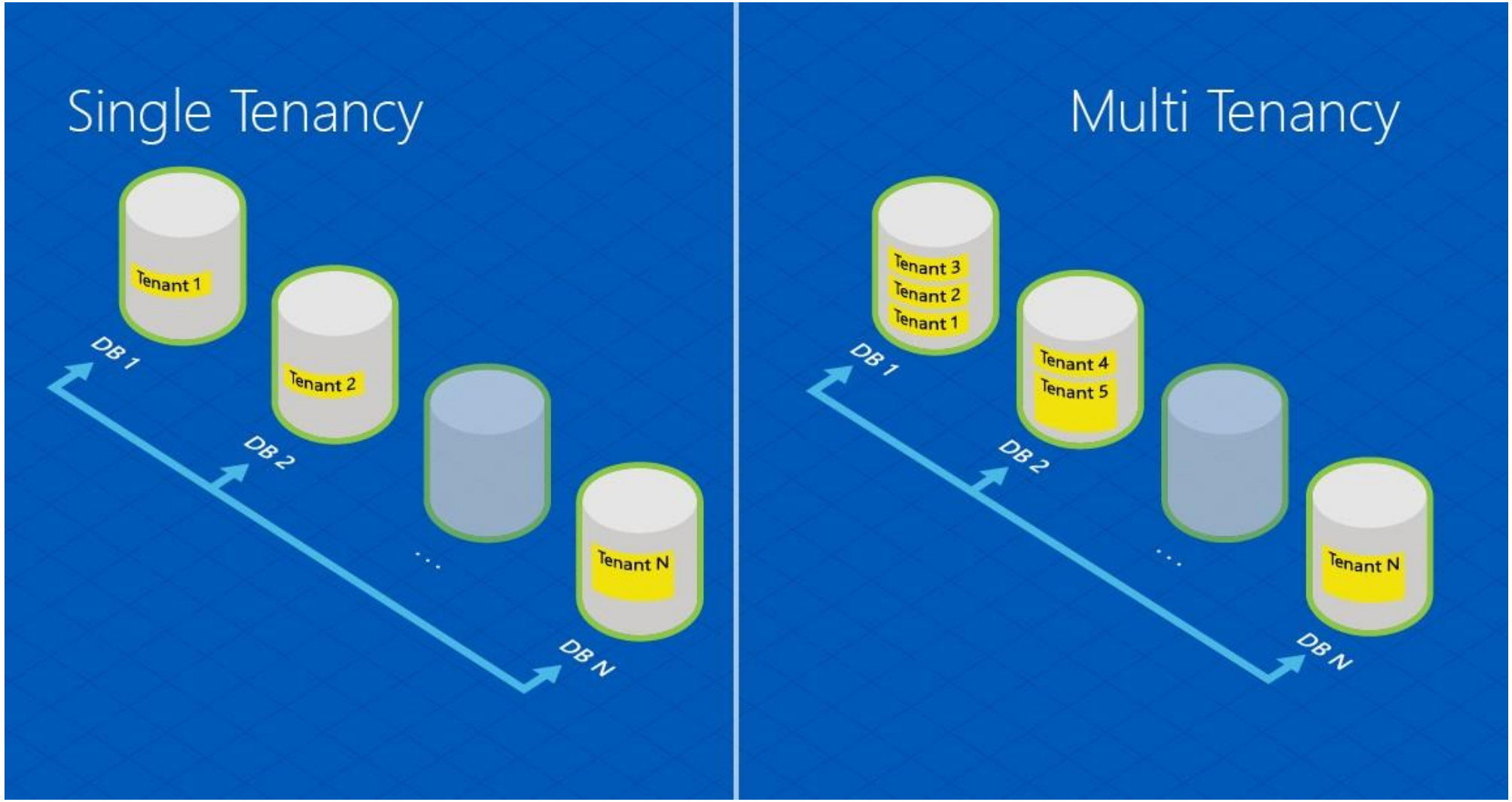
- Allows you to use the capacity of the read-only replicas for read-only queries.
- Feature is intended for the applications that include logically separated read-only workloads, such as analytics
- Benefit: When Secondary nodes handles heavy reports and analytical queries, primary writable node saved resources that might be used to improve the performance
- Secondary nodes is asynchronous
- Premium (DTU-based model) or in the Business Critical (vCore-based model)
- Also available in Hyperscale with secondary replica creation



Global Scale-out/Sharding

- Split your data into multiple database nodes.
- Every database shard is an independent database where you can add or remove resources as needed.
- Application may access only the shard associated to that region without affecting other shards.
- Why?
    - Data or transaction throughput exceed the capabilities of individual database
    - Tenants may require physical isolation
    - Different sections of a database may need to reside in different geographies for compliance, performance, or geopolitical reasons.

Single Tenancy vs Multi Tenancy



Azure SQL Database Scaling

Change Service Tier

- From Standard/General Purpose to Premium/Business Critical.
- In Standard/General Purpose - Data stored on Azure premium disks
- In Premium/Business Critical - Data stored on local SSD

Azure SQL DW

Scaling

Azure SQL Warehouse Scaling

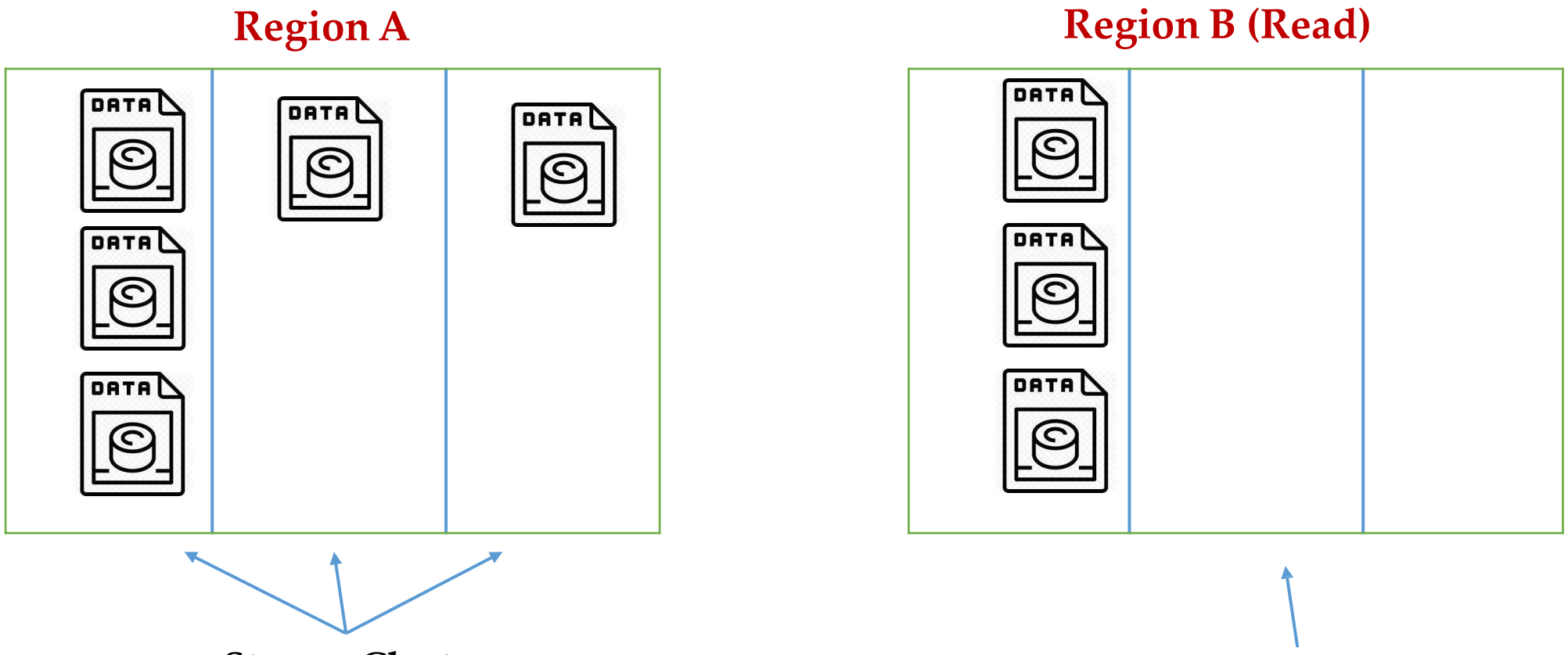
Scaling in SQL Data Warehouse

- SQL DW allows us to scale or pause at will
- Scale up during heavy demand
- Pause to cut cost
- DWUs are CPU, memory, and I/O bundled into units of compute scale
- Increasing DWU's
  - increase query performance
  - Also increase maximum number of concurrent queries and concurrent slc
- Modify with GUI, PowerShell, or TSQL

Azure SQL DB

HA and DR Options

Azure SQL DB – HA and DR Options



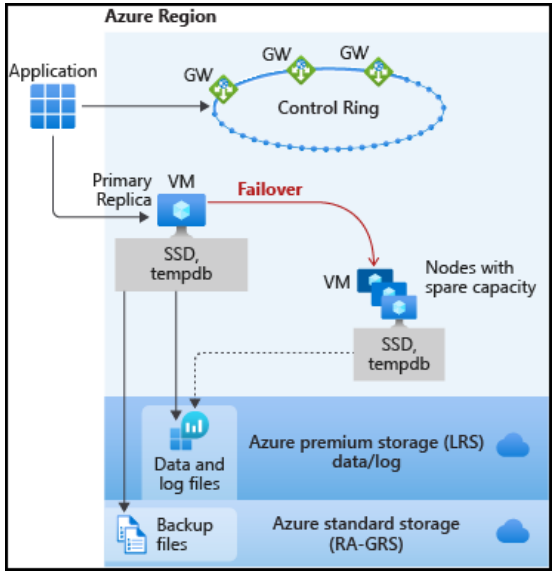
Storage Clusters

Each cluster is physically separate in what's called an availability zone, with its own separate utilities and networking.

Region B

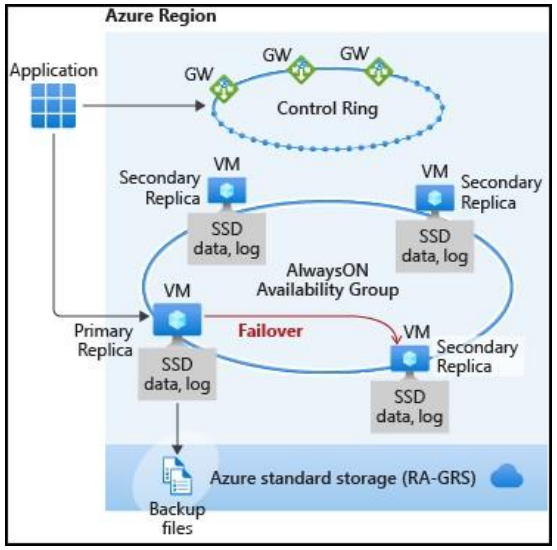
Hundreds of miles away from the primary region to prevent data loss in the event of a natural disaster.

Azure SQL DB – High Availability Architecture



Standard availability model

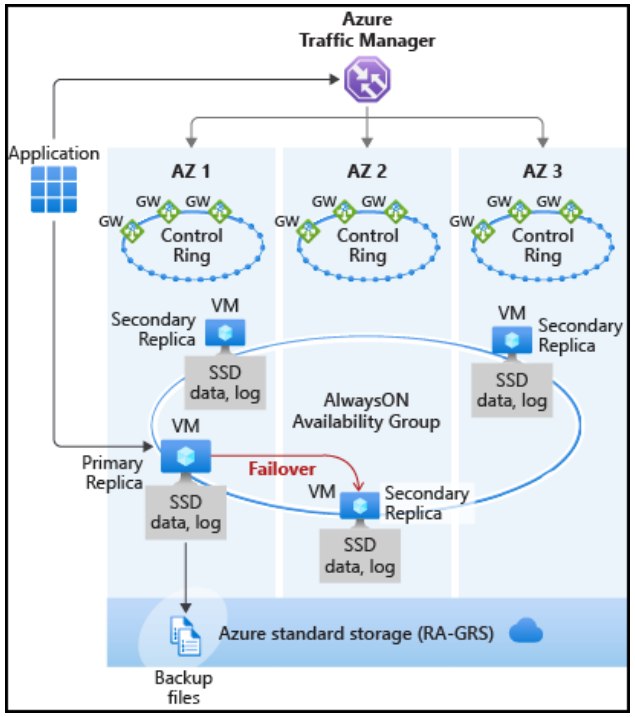
- Separation of compute and storage
- Performance degradation during maintenance activities
- Budget-oriented business applications
- Basic, Standard, and General Purpose service tier availability



Premium availability model

- Integrates compute resources and storage (locally attached SSD) on a single node
- Replicating both compute and storage to additional nodes creating a three to four-node cluster.
- Data is synchronized to at least one secondary replica before committing each transaction.
- Targets mission critical applications with high IO performance and high transaction rate
- Guarantees minimal performance impact during maintenance activities.
- Premium and Business Critical service tier availability
- Read Scale-Out feature

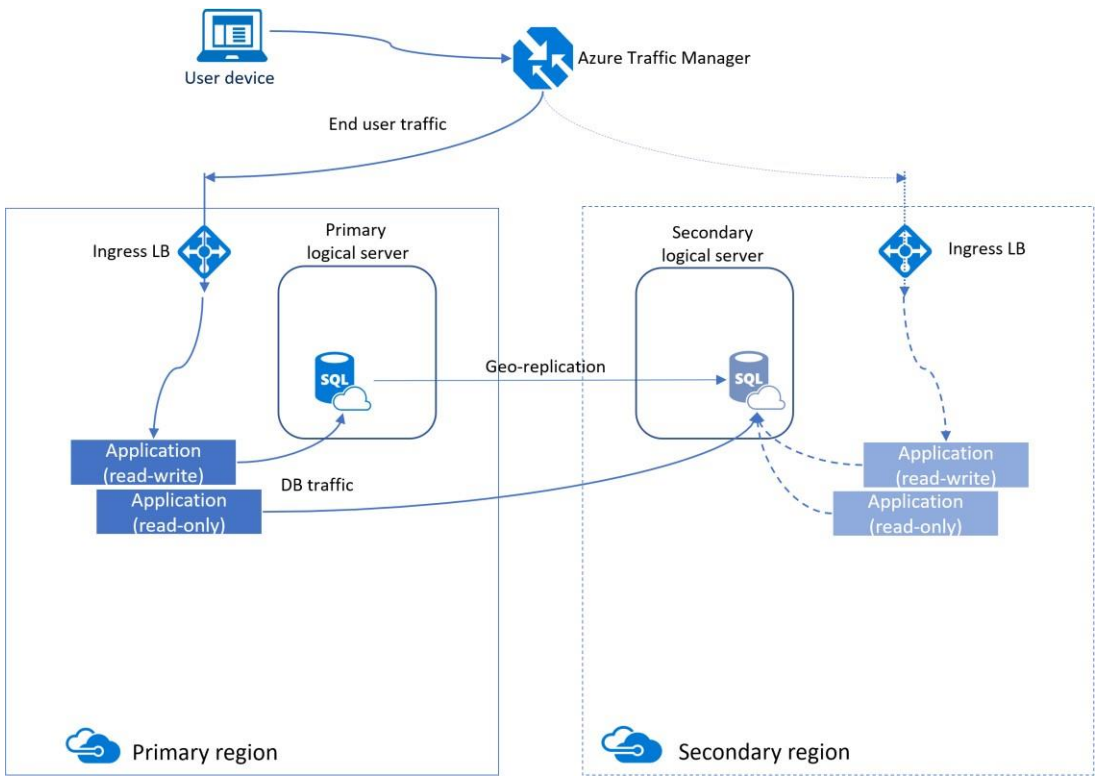
Azure SQL DB –Zone Redundancy



Zone redundant configuration

- Place different replicas of the Business Critical database to different availability zones in the same region
- Does not create additional database redundancy
- Enable it at no extra cost
- Supported in the Premium and Business Critical service tiers
- Not available in SQL Managed Instance.
- Increased network latency may increase the commit time and thus impact the performance of some OLTP workloads.

Azure SQL DB – Geo Replication



- **Create a readable secondary database in the same region or cross-region**
- **Use cases:**
  - Can failover to the secondary database in case of an outage
  - Migrate a database from one server to another server in the same or cross region with minimal downtime.
- We can create up to four secondaries for each primary database.
- **Data Loss:**
  - Uses the Always-on feature to replicate committed transactions to the secondary database asynchronously.
  - May lag the primary database at any point in time.
- **Manual - Forced Failover**
  - This will make your secondary database immediately online and start accepting connections. Forced failover may result in data loss.

- **Issues**:
  - Supports only manual failover
  - End-point connection must be changed in the application after the failover
  - Must have the same firewall rules and the logins to run applications successfully without any discrepancies

Compare geo-replication with failover groups

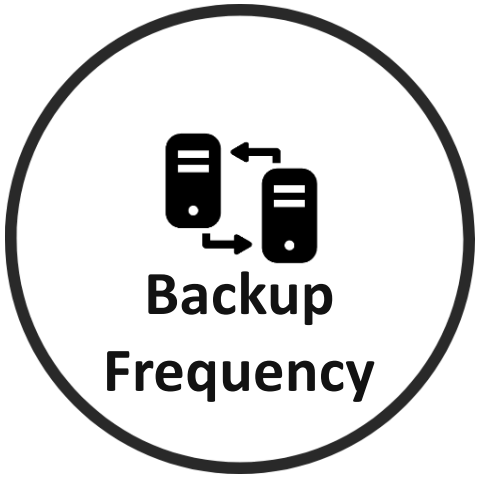
Auto-failover groups simplify the deployment and usage of geo-replication and add the additional capabilities as described in the following table:

	Geo-replication	Failover groups
Automatic failover	No	Yes
Fail over multiple databases simultaneously	No	Yes
User must update connection string after failover	Yes	No
SQL Managed Instance support	No	Yes
Can be in same region as primary	Yes	No
Multiple replicas	Yes	No
Supports read-scale	Yes	Yes

Azure SQL DB

Backup and Restore

Protect your data from corruption or deletion



1

Full Backup

- Backs up the whole database
- Taken every week

2

Differential backup

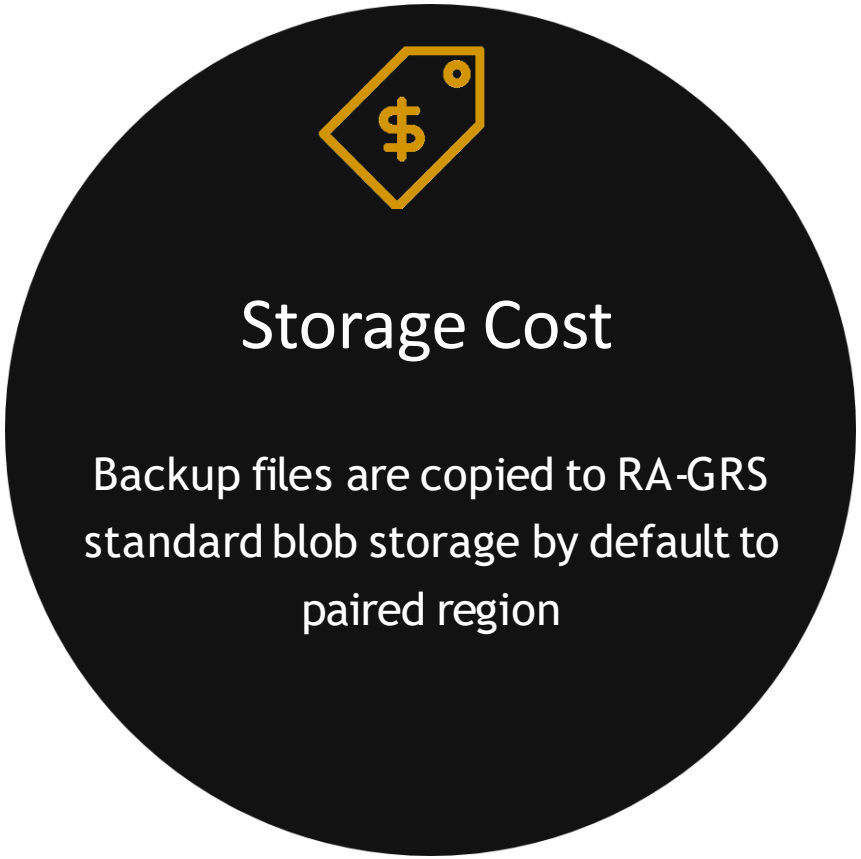
- Captures only the data that has changed since the last full backup
- Taken every 12 hours

3

Transaction log backup

- Records of all the committed and uncommitted transactions
- Taken every 5-10 mins

Storage cost and Security of Backup files

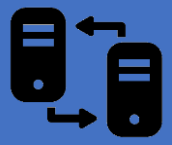


Storage Cost

Backup files are copied to RA-GRS standard blob storage by default to paired region

Security

Backup are automatically encrypted at rest using TDE, blob storage is also protected



Backup Retention Period

Backup storage redundancy

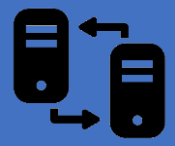
- Point in time restore (PITR) - 7-35 days
- Long-term retention - Up to 10 years

Long term retention (LTR)

- One or more long term retention periods to your database to meet regulatory, compliance or other business purposes
- Full backups can be taken up to 10 years
- Stored in RA-GRS blob storage
- Any change of the LTR policy applies to the future backups

LTR and Managed Instance

- LTR is not yet available for databases in Managed Instances
- You can use SQL Agent jobs to schedule copy only database backups as an alternative to LTR beyond 35 days
- These backups can be kept in the Azure blob storage



Backup Restore

Backup usage

- Point-in-time restore of existing database
- Point-in-time restore of deleted database
- Geo-restore
- Restore from long-term backup
- *If you delete an Azure Logical SQL Server, all elastic pools and databases that belong to that logical server are also deleted and cannot be restored*

Restore Time is impacted By

- Size of database & Compute size of the database
- Number of transaction logs and Amount of activity
- Network bandwidth if the restore is to a different region
- Concurrent restore requests being processed region

Azure SQL DW

Backup and Restore

Protect your data from corruption or deletion

Azure SQL DW Backup and Restore

- Snapshots of your data warehouse are taken throughout the day creating restore points
- These restore points are available for 7 days
  - Retention period of 7 days cannot be changed
- SQL pool supports an 8 hour recovery point objective (RPO)
- Replicated to paired region once a day
- You can also take user-defined snapshots
  - Retention period 7 days cannot be changed
  - 42 max restore point possible
  - Can be created using PowerShell or portal
- When you drop a SQL pool, a final snapshot is created and saved for seven days
  - SQL Pool should not be in paused state

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

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

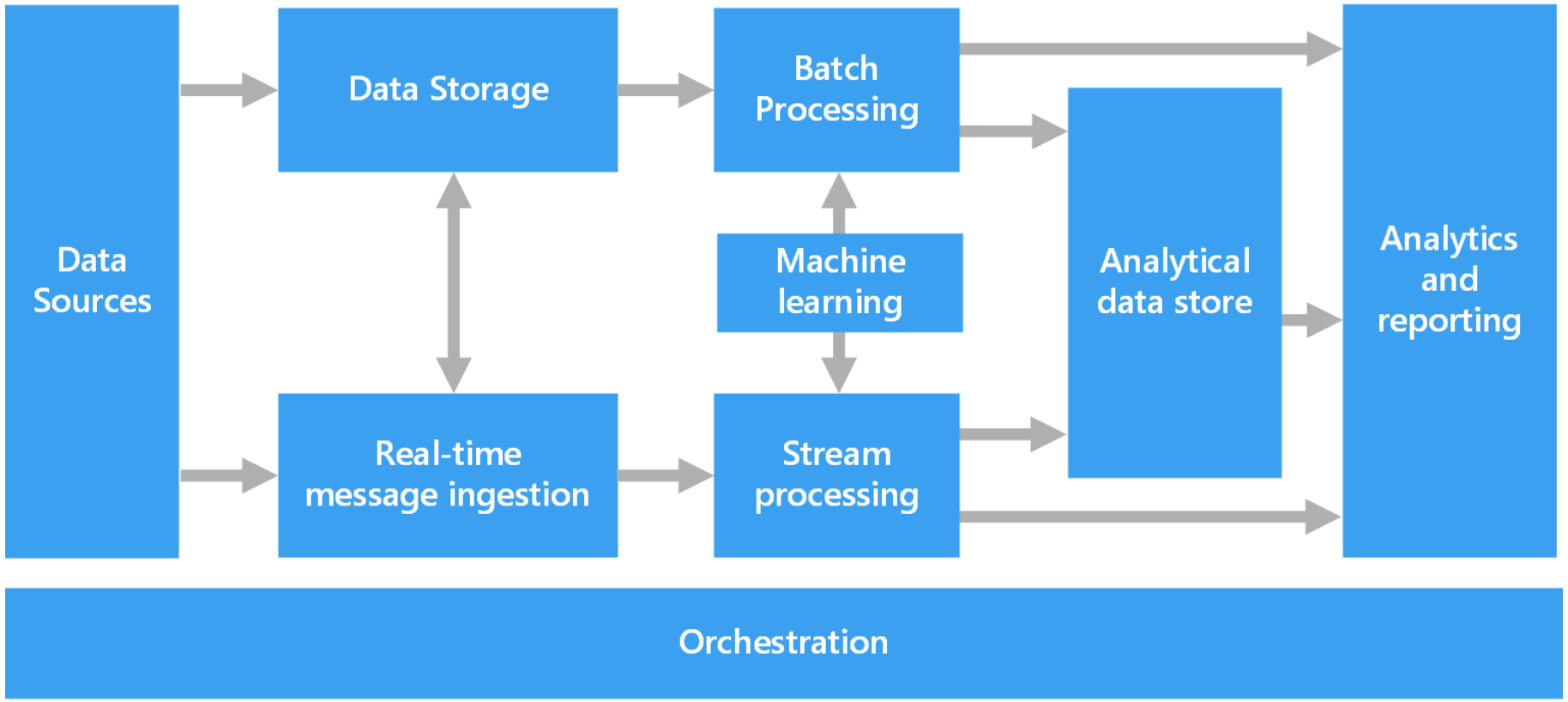
Scenario 2

- 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

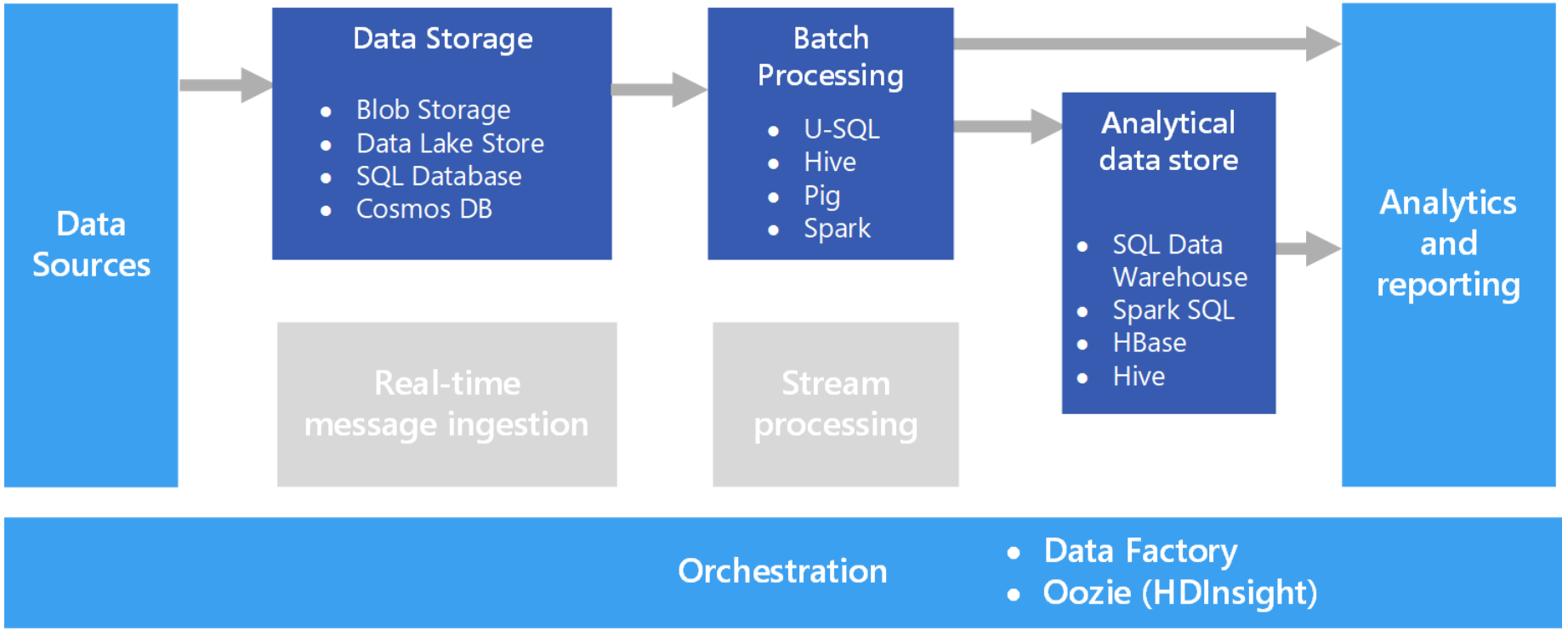
- Options
  - Azure SQL Database – Single
  - Azure SQL Database – Elastic Pool
  - Azure SQL Database – Managed Instance
  - Azure SQL Datawarehouse

Batch Processing

Big Data Architecture



Batch Processing



Batch Processing:

- Data at rest
- 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

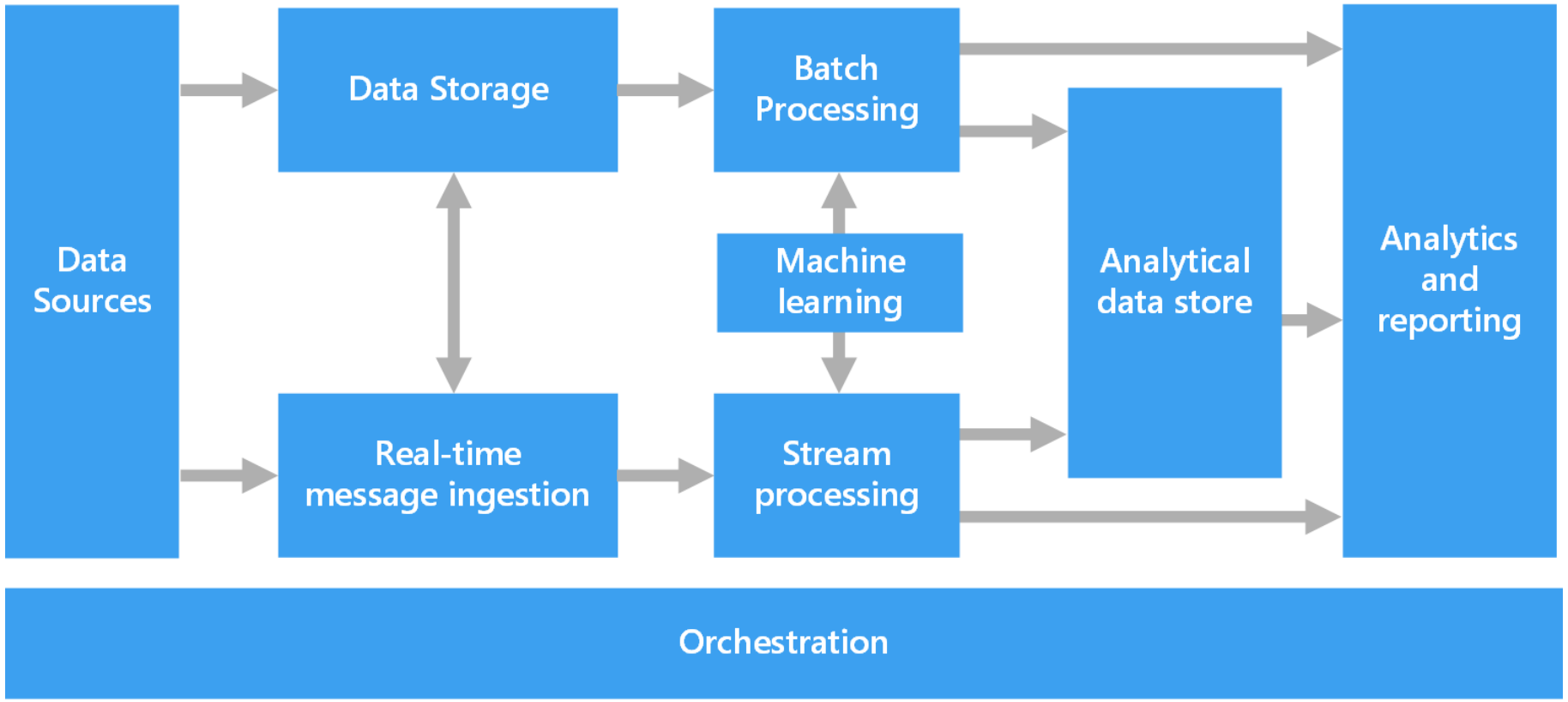
Azure Databricks

- **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**

Big Data Architecture



Data Pipeline Orchestration

Pipeline Orchestration options:

- 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?
- Do you require a managed service that can operate at scale?
- Are some of your data sources located on-premises?
- Is your source data stored in Blob storage on an HDFS filesystem?

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

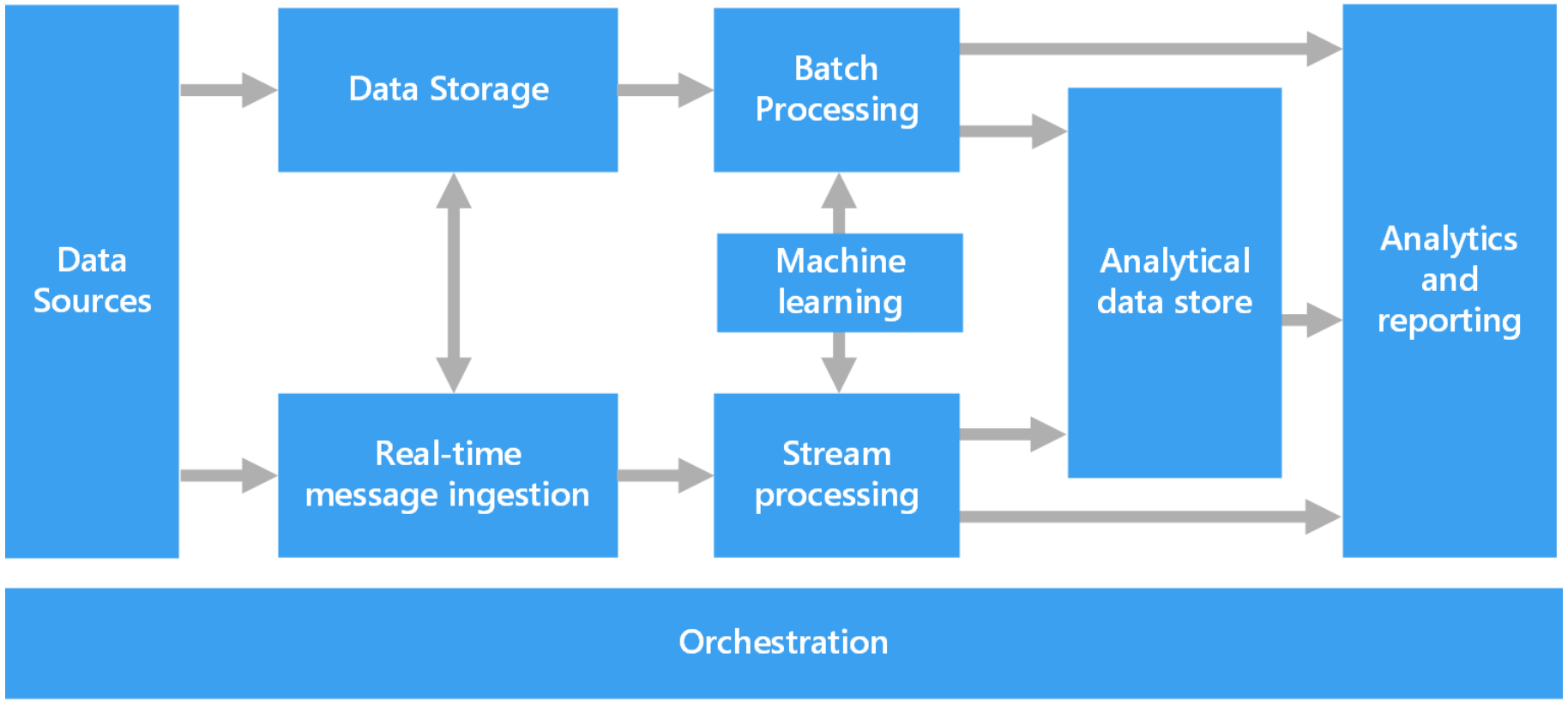
Scalability capabilities

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
- Distcp
- Sqoop
- PolyBase
- HadoopCommandline

Graphical Interface

- Azure Storage Explorer
- Azure portal

Data pipeline

- Azure Data Factory

Command line tools comparison

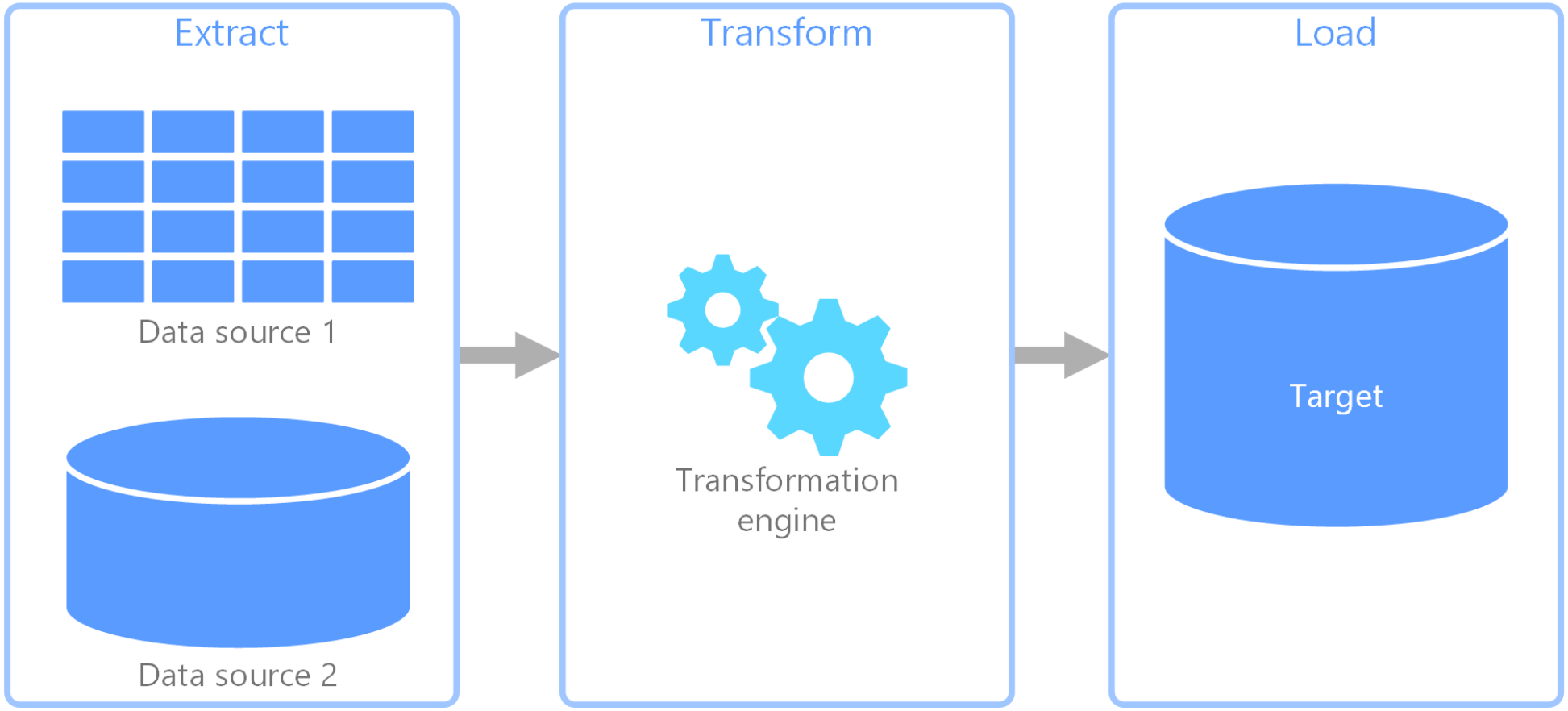
Capability	Azure CLI	AzCopy	PowerShell	AdlCopy	PolyBase
Compatible platforms	Linux, OS X, Windows	Linux, Windows	Windows	Linux, OS X, Windows	SQL Server, Azure Synapse
Optimized for big data	No	Yes	No	Yes [1]	Yes [2]
Copy to relational database	No	No	No	No	Yes
Copy from relational database	No	No	No	No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

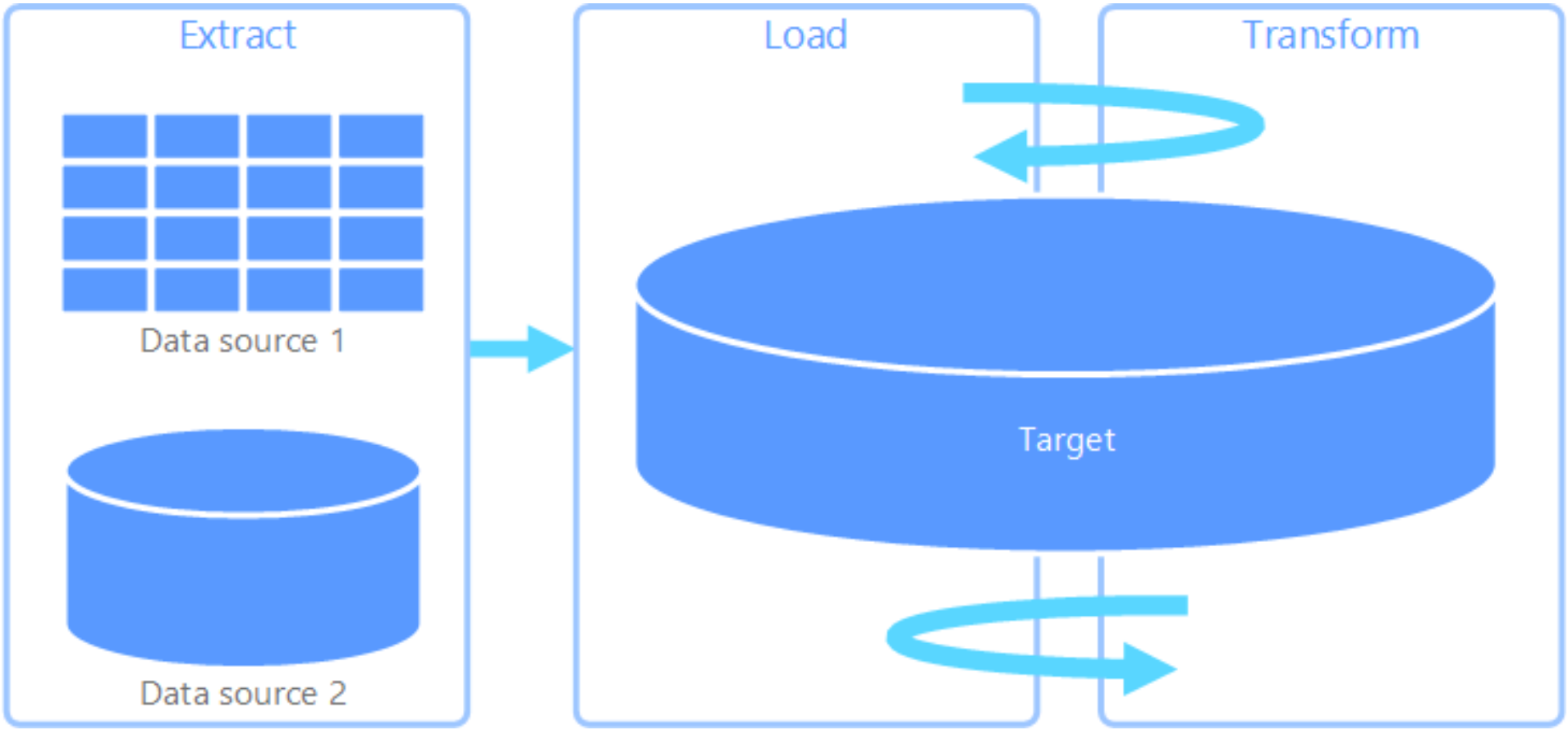
Capability	Azure Storage Explorer	Azure portal *	Azure Data Factory
Optimized for big data	No	No	Yes
Copy to relational database	No	No	Yes
Copy from relational database	No	No	Yes
Copy to Blob storage	Yes	No	Yes
Copy from Blob storage	Yes	No	Yes
Copy to Data Lake Store	No	No	Yes
Copy from Data Lake Store	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

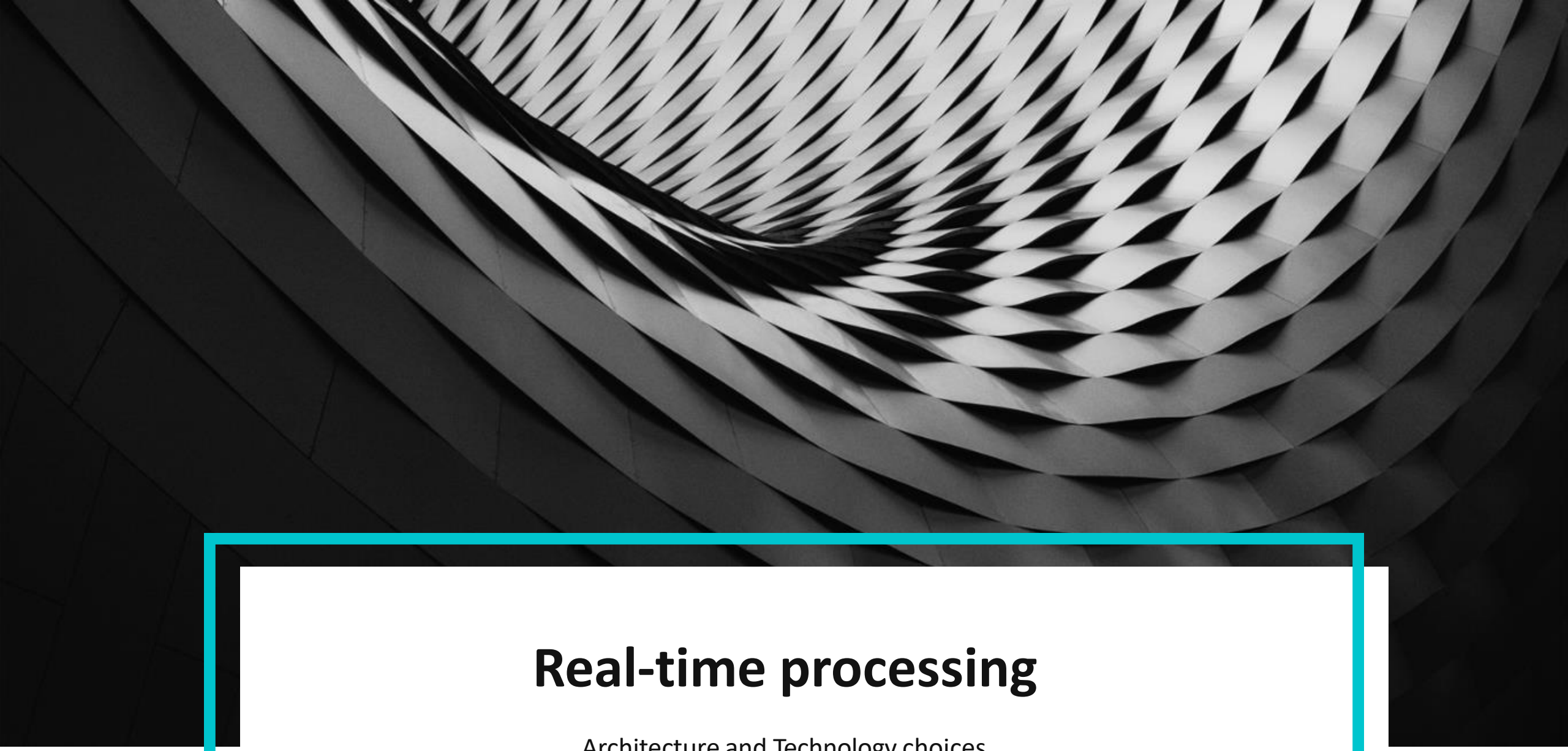
ETL vs ELT

Extract, Transform, and Load (ETL) process



Extract, Load, and Transform (ELT)

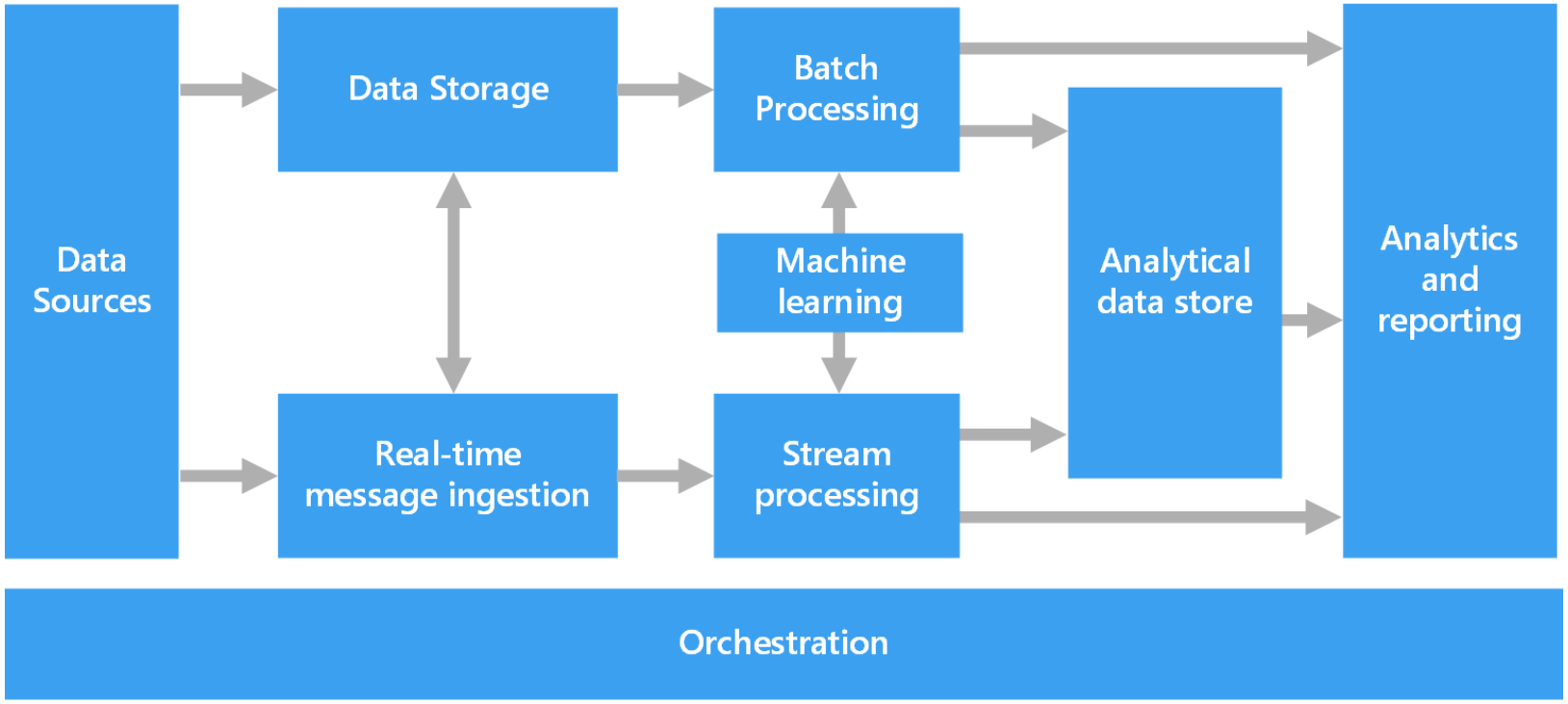




Real-time processing

Architecture and Technology choices

Real Time Processing Architecture



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.

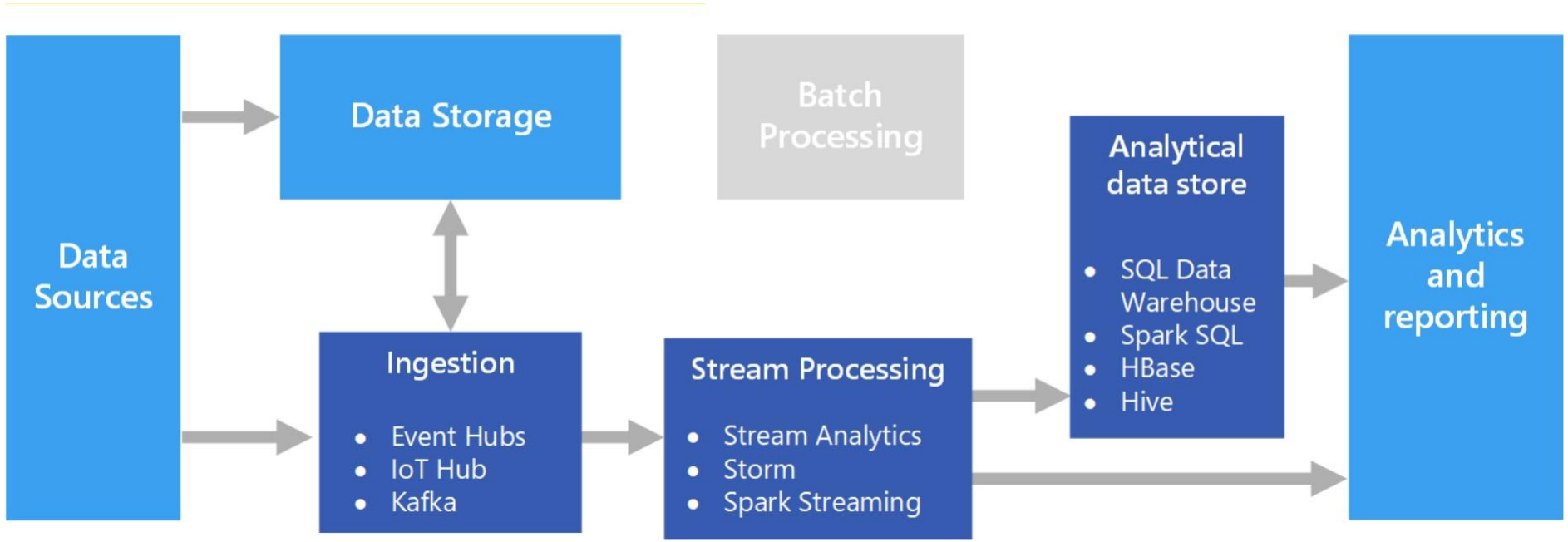
For example:

- 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)

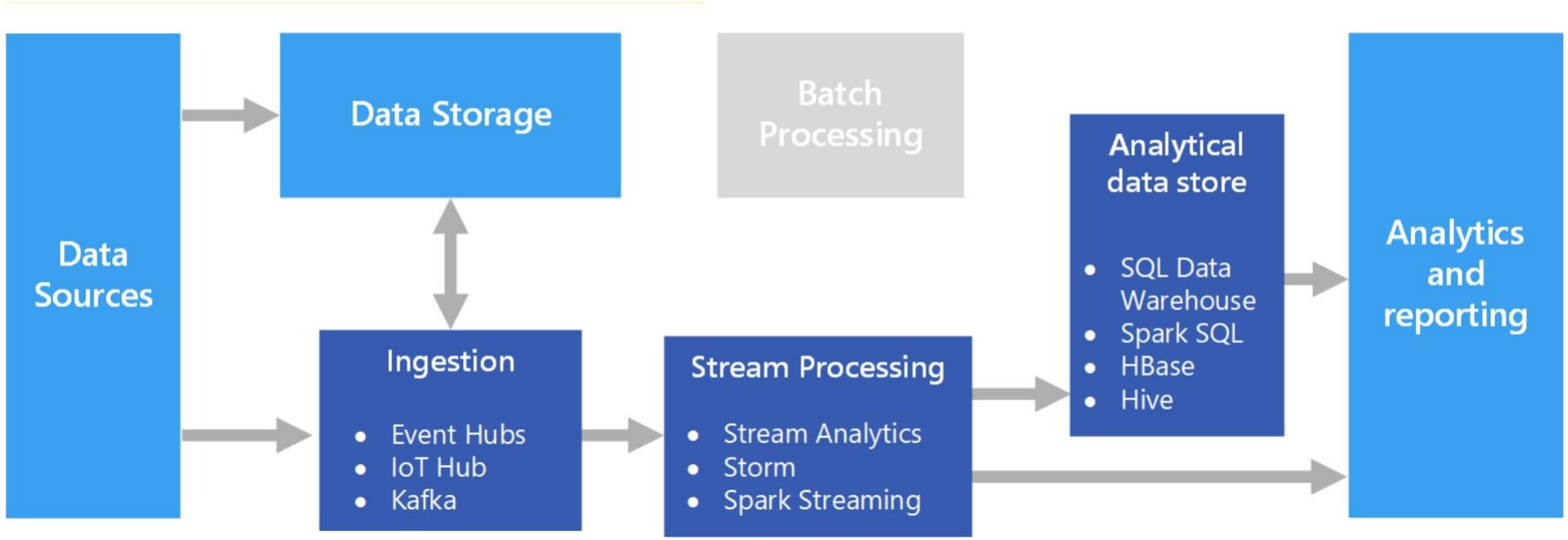
Azure IoT Hub: Provides bi-directional communication between Internet-connected devices

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

Apache Kafka: Open source message queuing and stream processing application

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

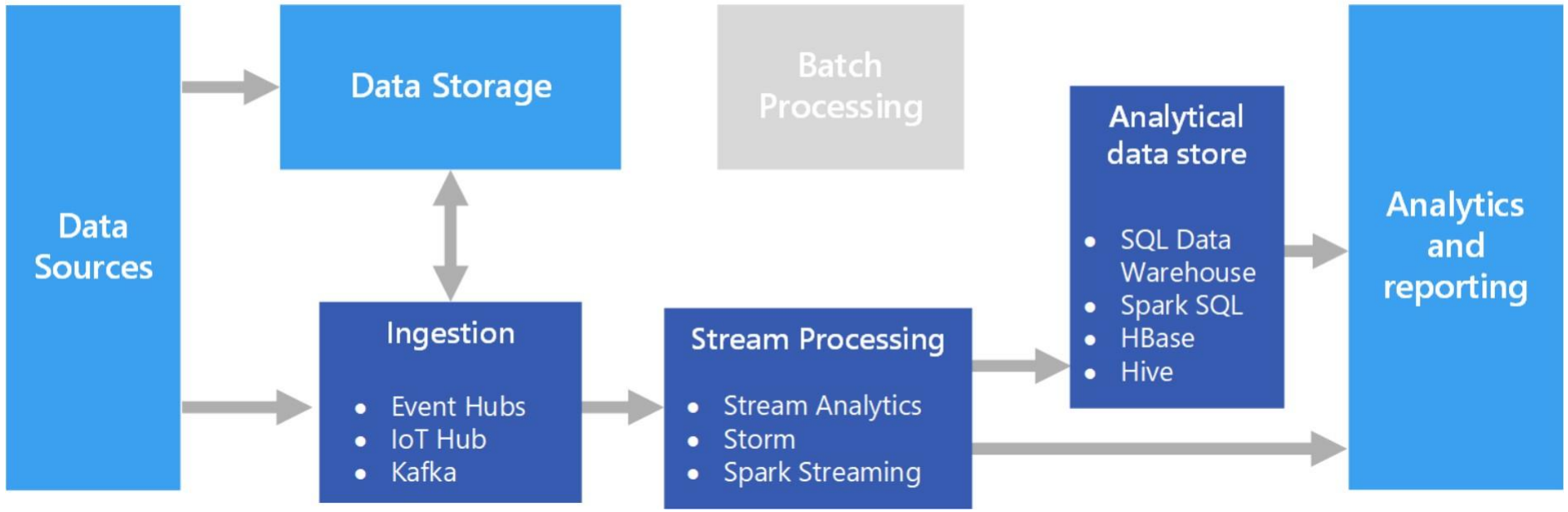
Azure Stream Analytics: can run perpetual queries against an unbounded stream of data

- 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

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

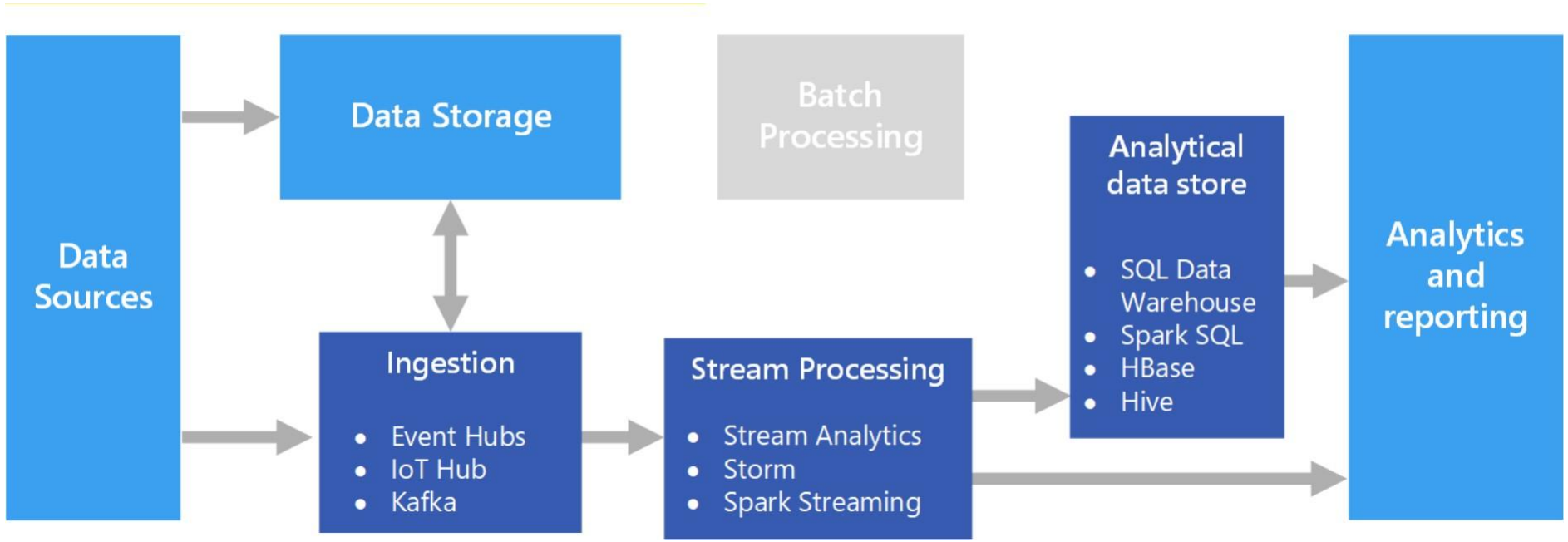
Real Time Processing Architecture



Analytical data store

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

Real Time Processing Architecture



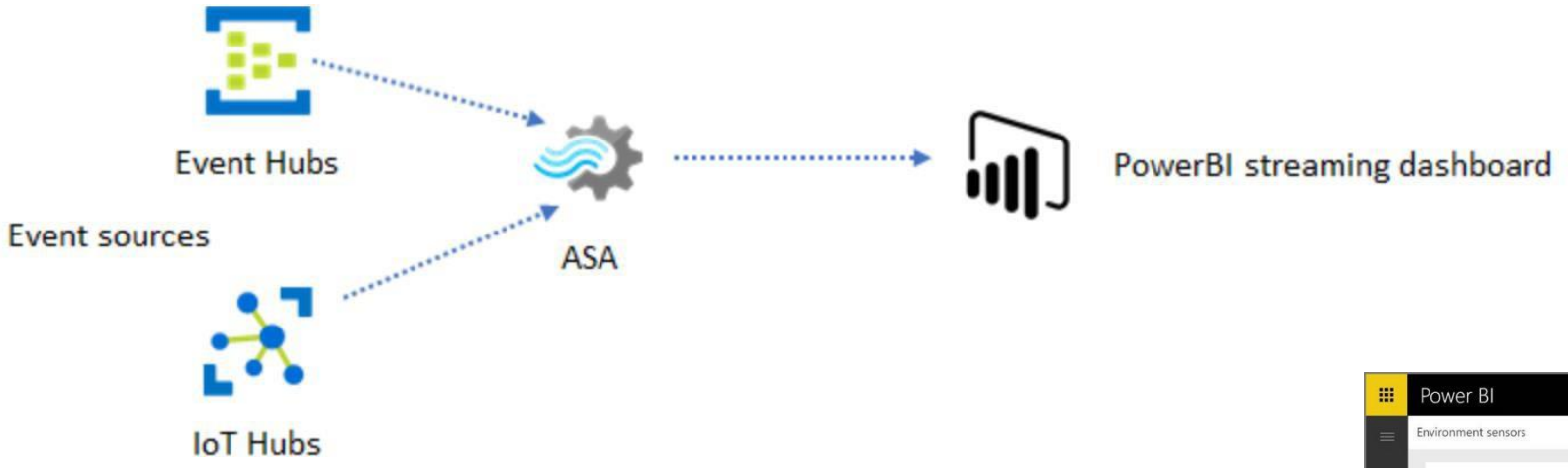
Analytics and reporting

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

Design and provision compute resources

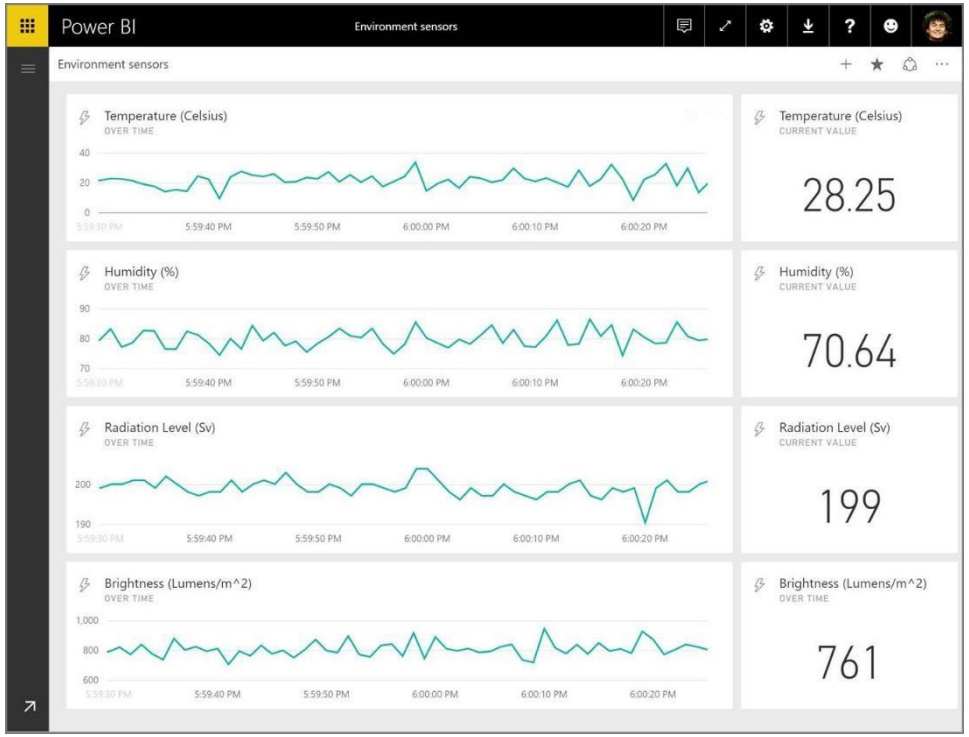
Azure Stream analytics solution and architectural patterns

Real time dashboard



Streaming data can be:

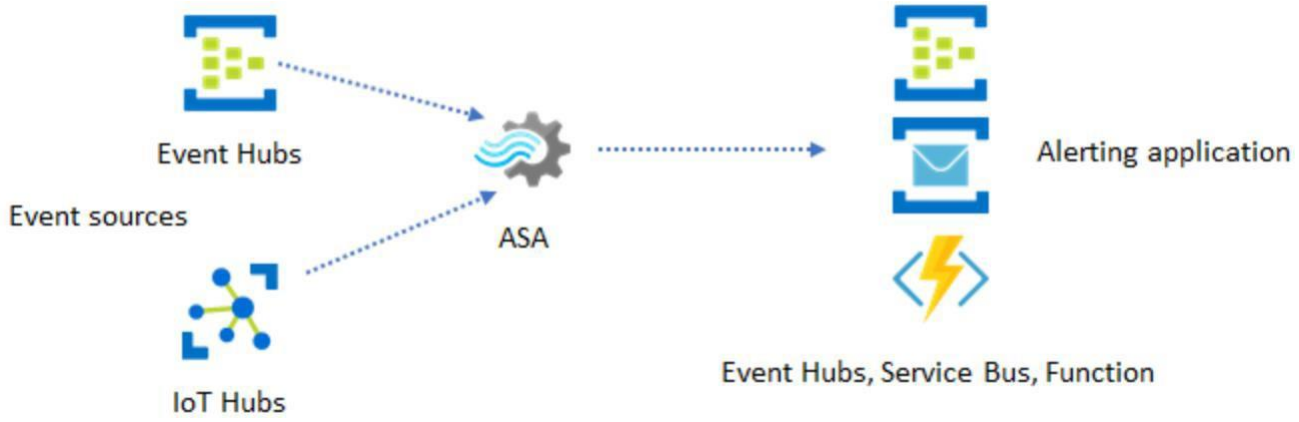
- Factory sensors
- Social media sources
- Service usage metrics
- Or many other time-sensitive data collectors or transmitters



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



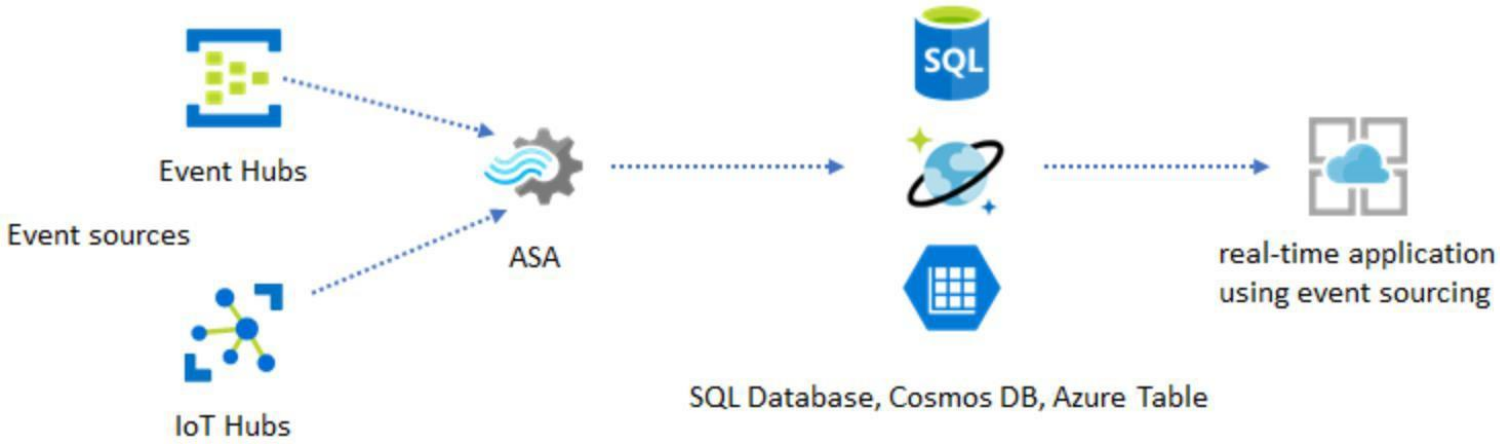
Why Azure Functions?

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

Why Event hubs?

- 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.

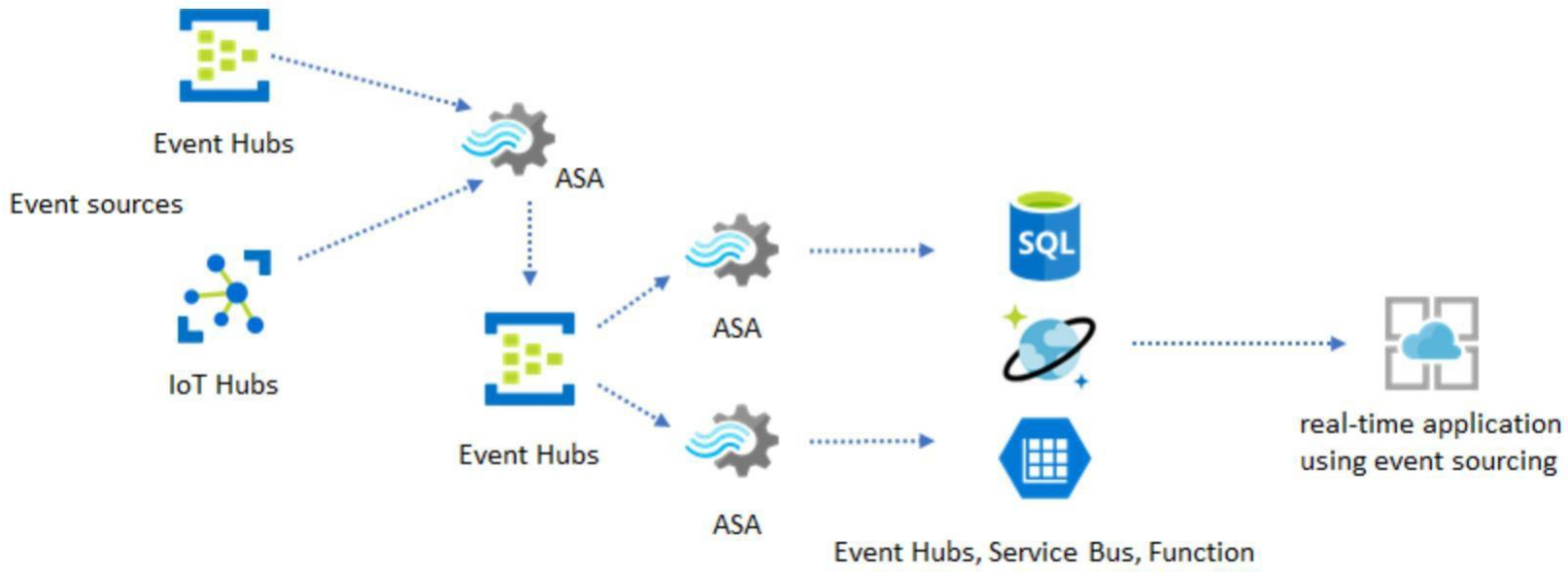
Real-time insights through data stores



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.

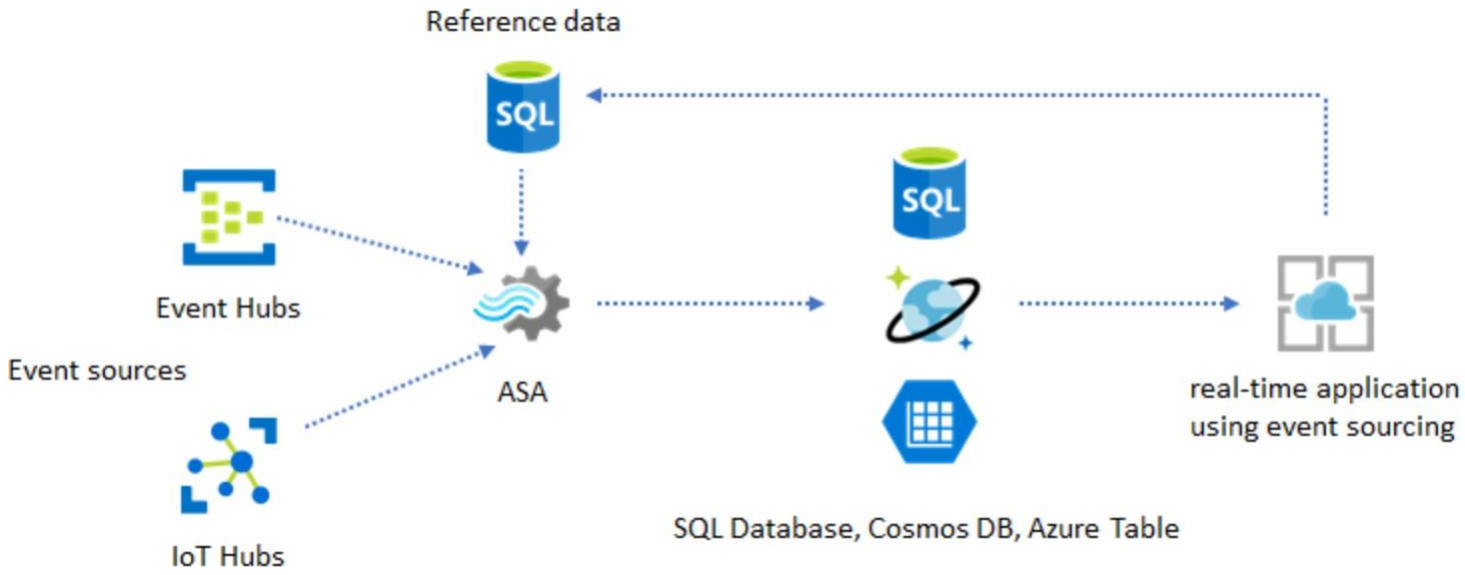
Real-time insights through data stores



Dataflow-based architecture

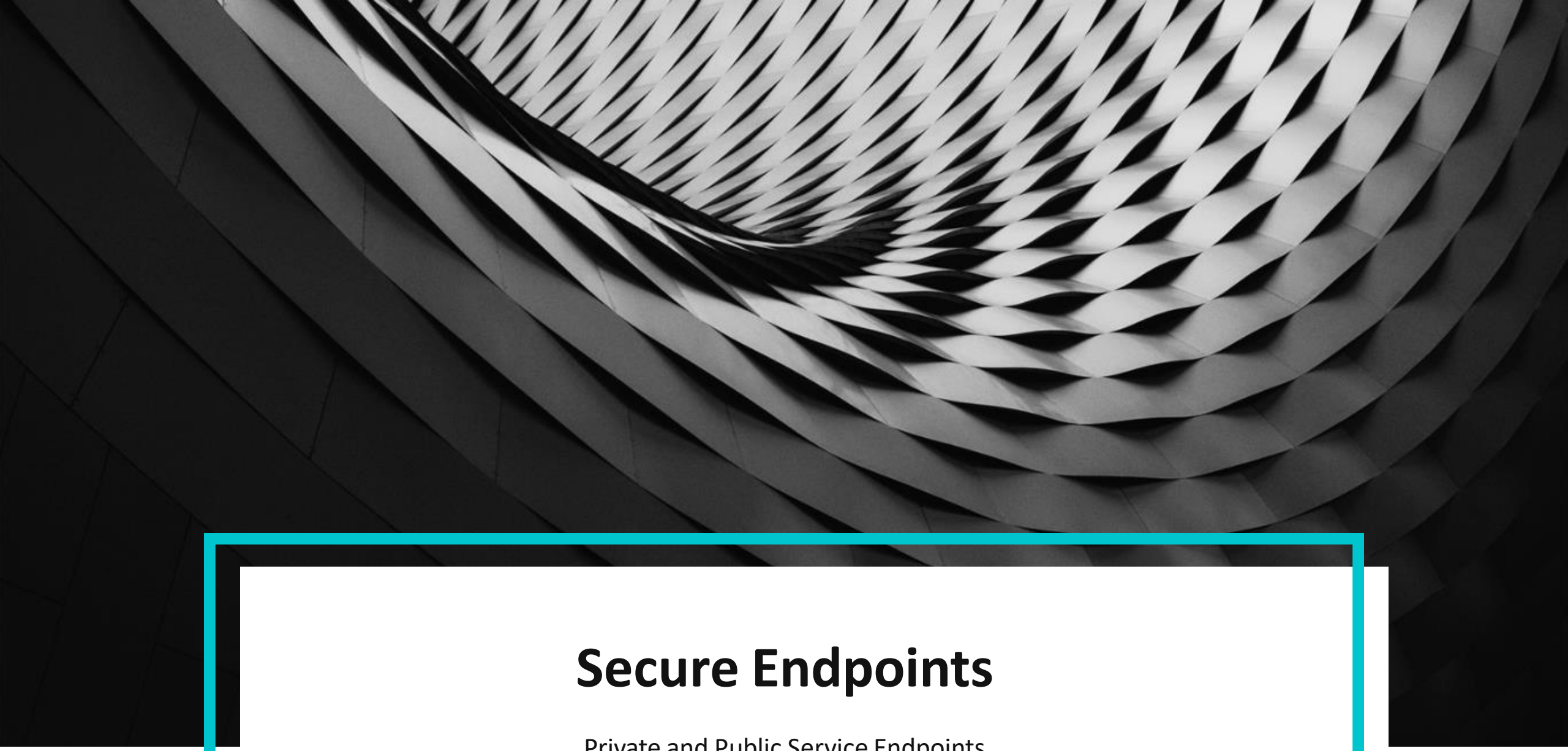
- Modern high-volume data driven applications often adopt a dataflow-based architecture
- Events are processed and aggregated into data stores by Azure Stream Analytics
- The application layer interacts with data stores using the traditional request/response pattern.

Reference data for application customization



Reference data

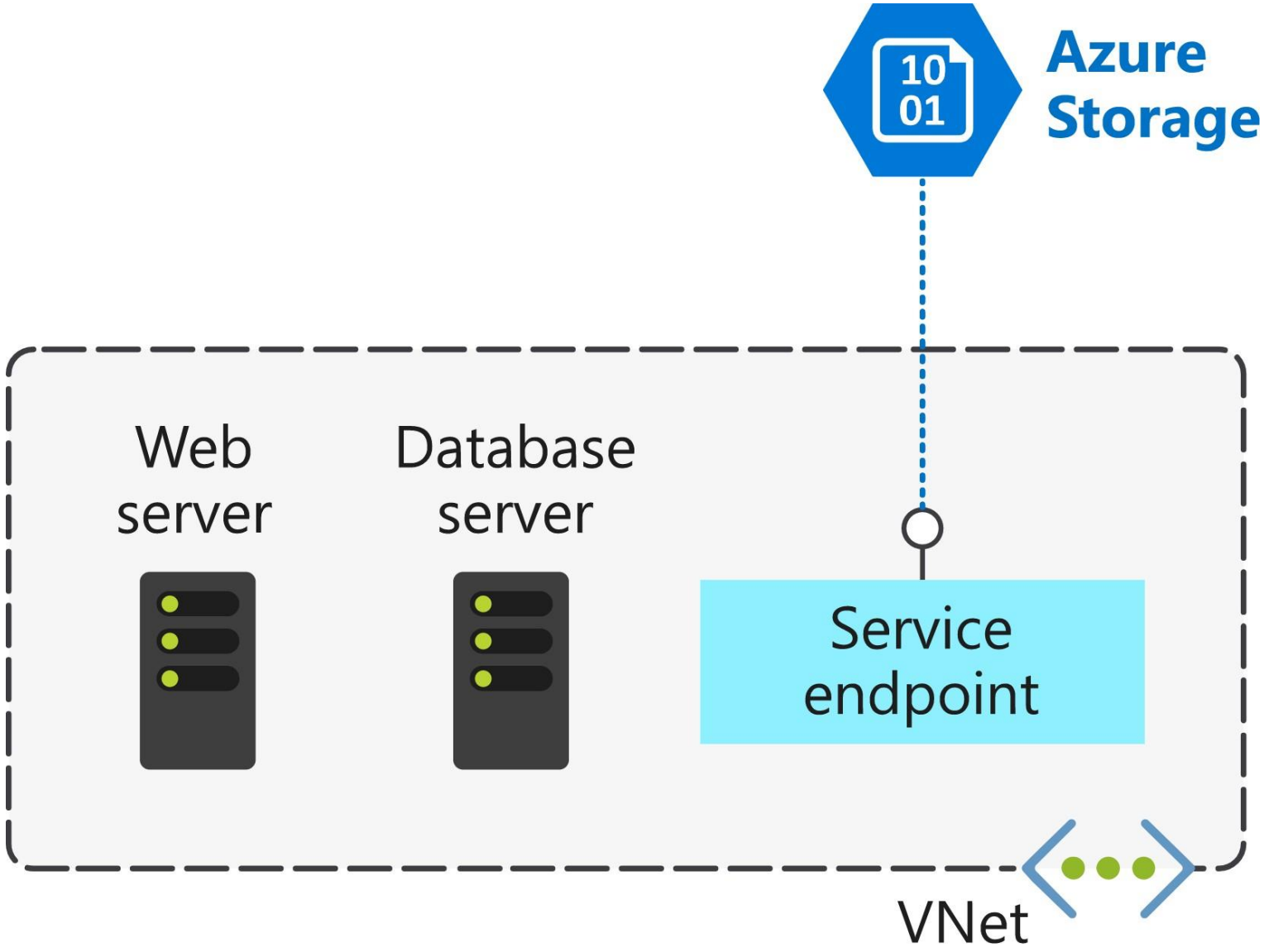
- Reference data feature is designed specifically for end-user customization like alerting threshold and processing rules
- Reference data (also known as a lookup table) is a finite data set that is static or slowly changing in nature



Secure Endpoints

Private and Public Service Endpoints

Virtual network service endpoints



Service endpoints