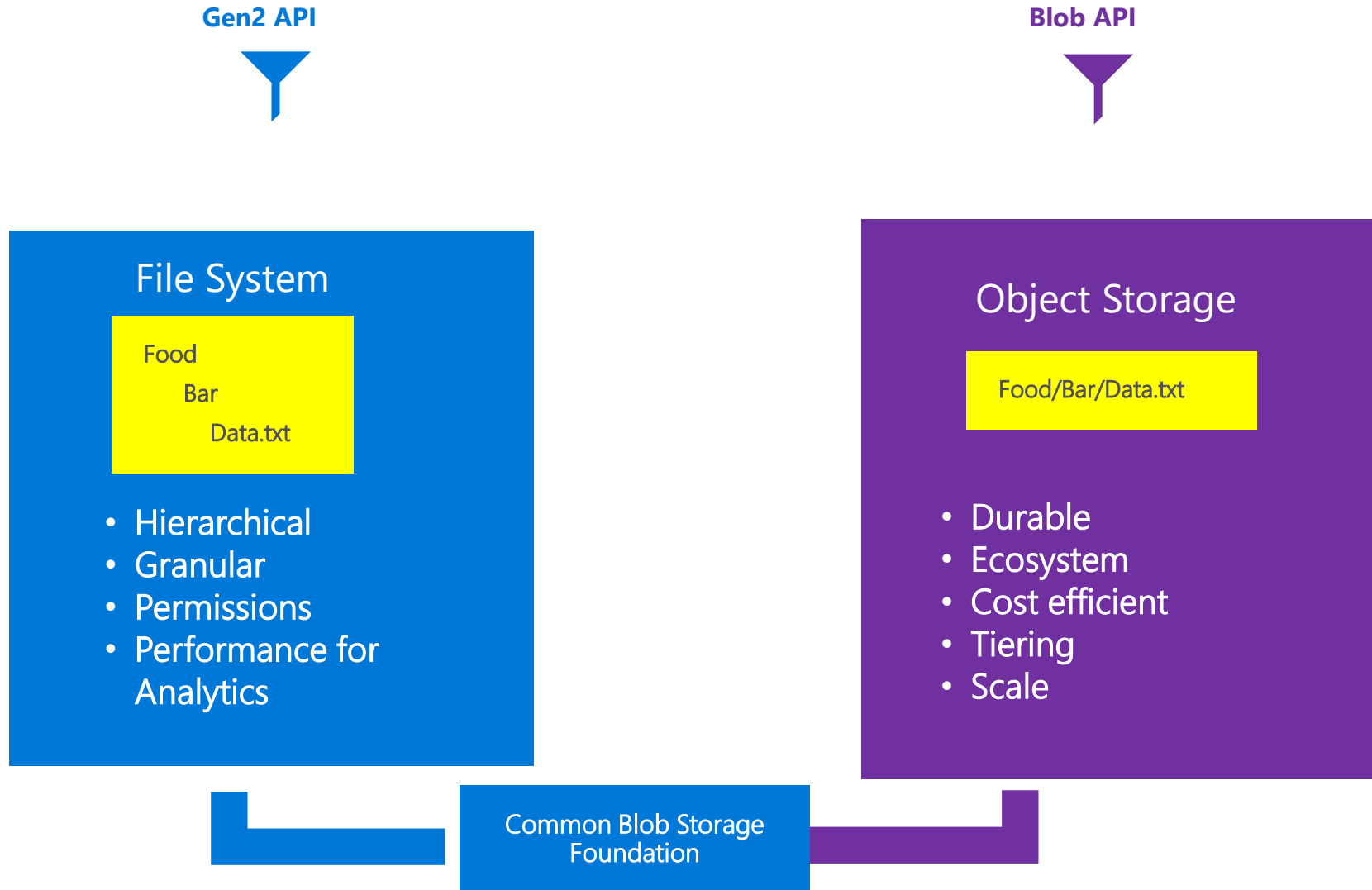


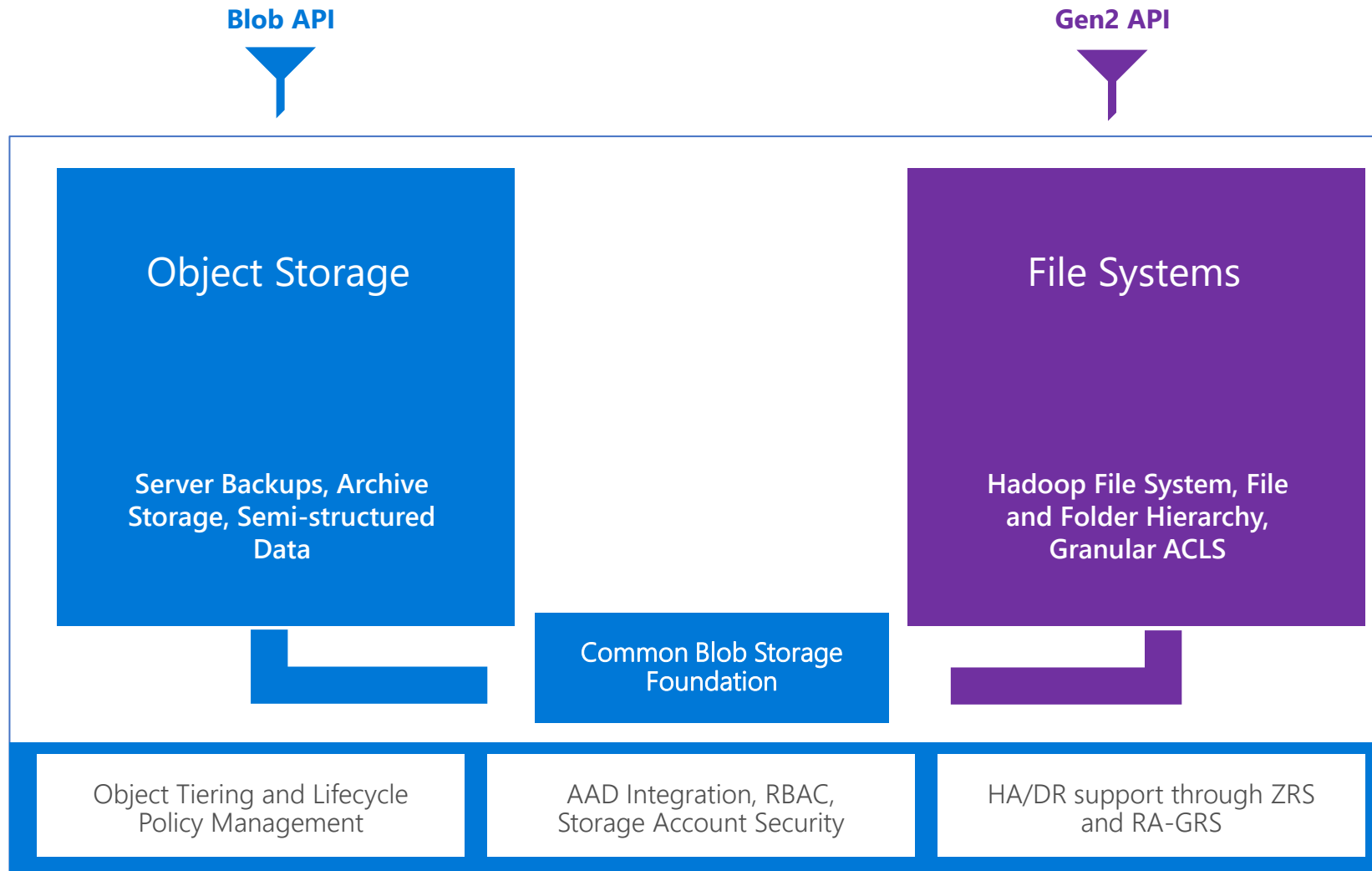
Azure Data Lake Storage Design

Maggie Liu
Cloud Solution Architect

Cloud based Multi-Model Storage Service built on Azure Storage



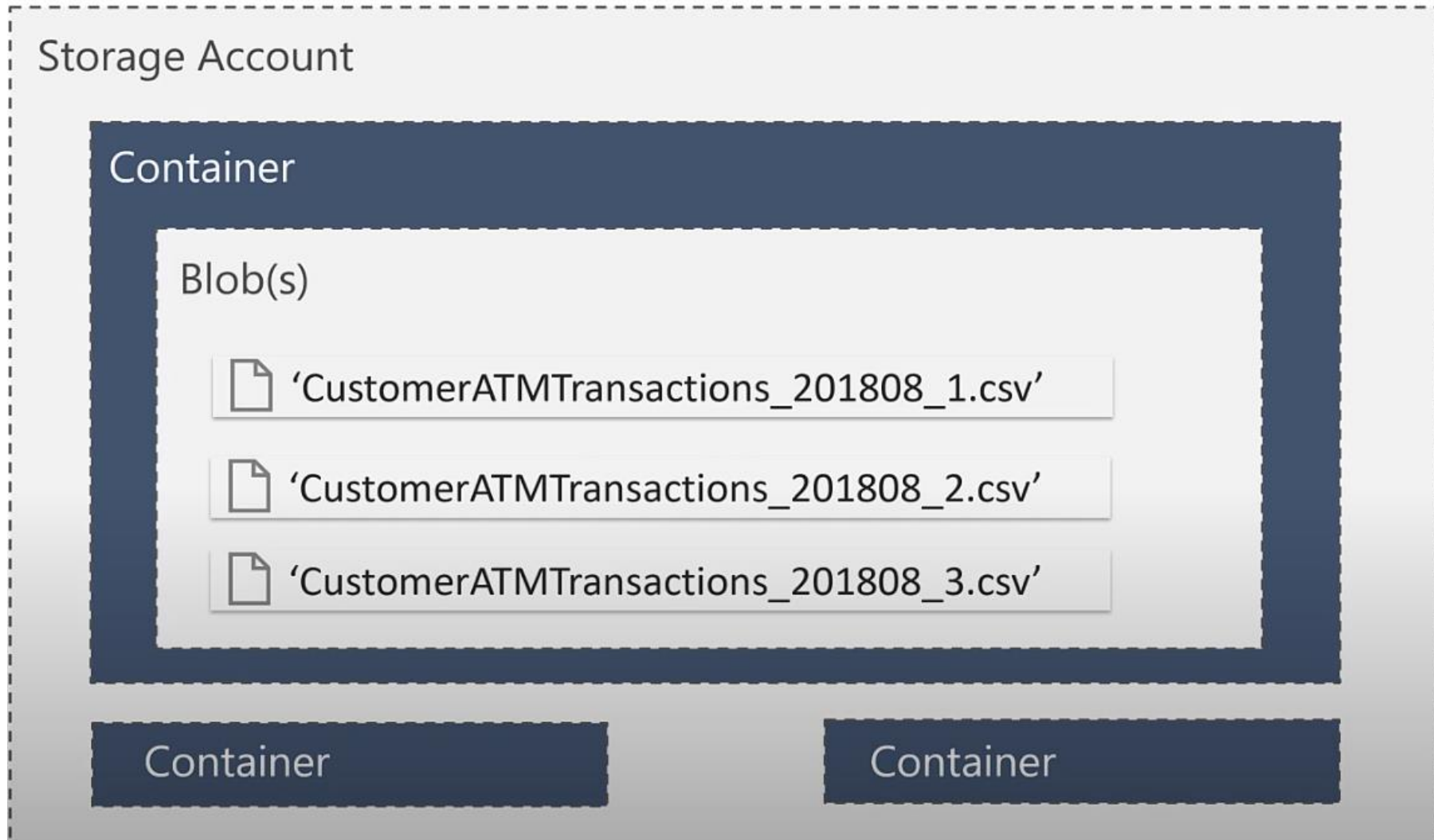
Azure Data Lake Gen2



** Full Blob and Interoperability coming soon*

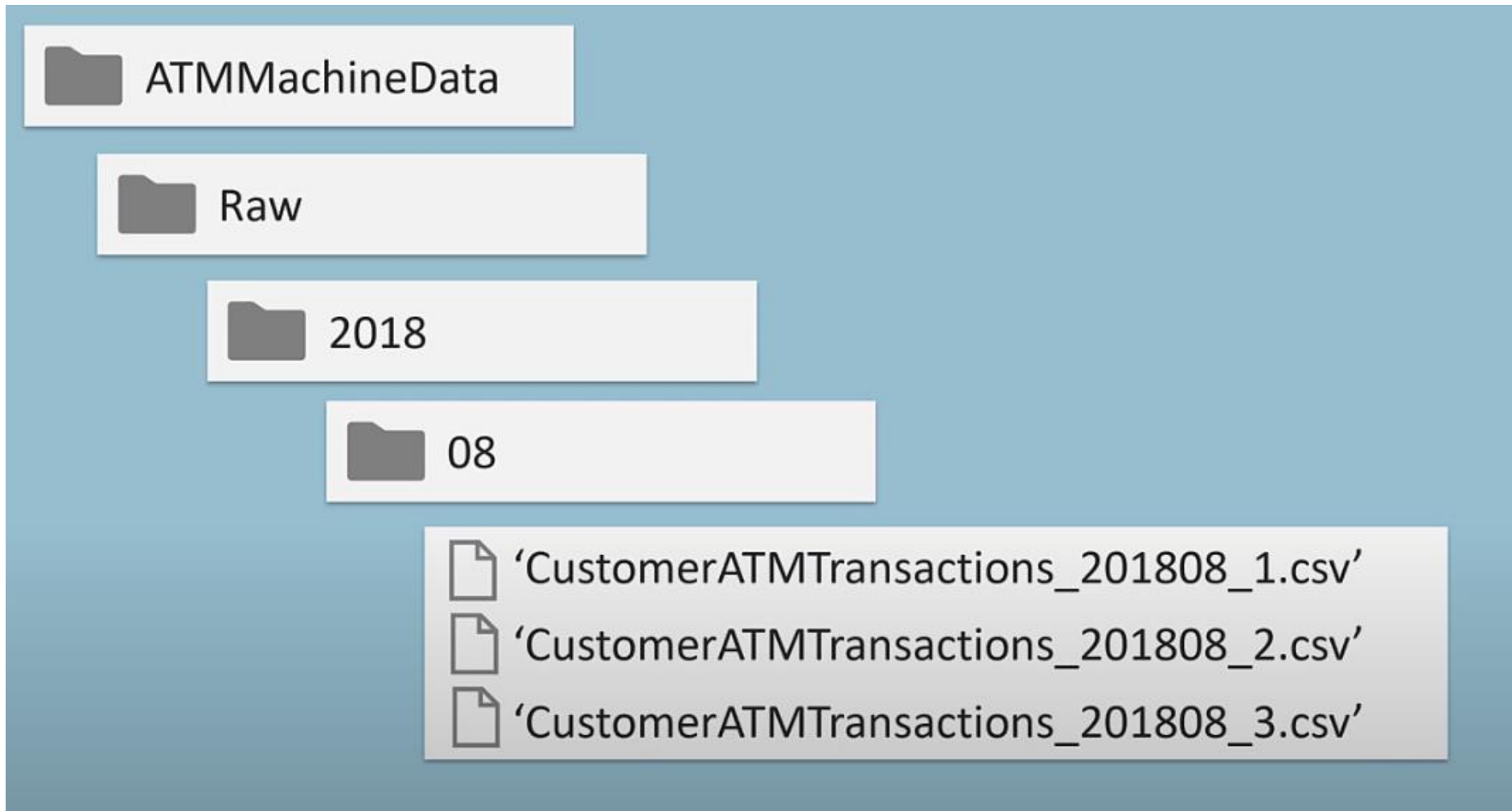
Azure Storage

- Object-based storage manages data as discrete unit
- Folders are merely simulated in traditional object storage



Azure Data Lake Gen2

Hierarchical file-based storage supports nesting of files within folders.



Storage Account

File System

Folders & Files



File System

File System

File System

Hierarchical Namespace

Object store drivers

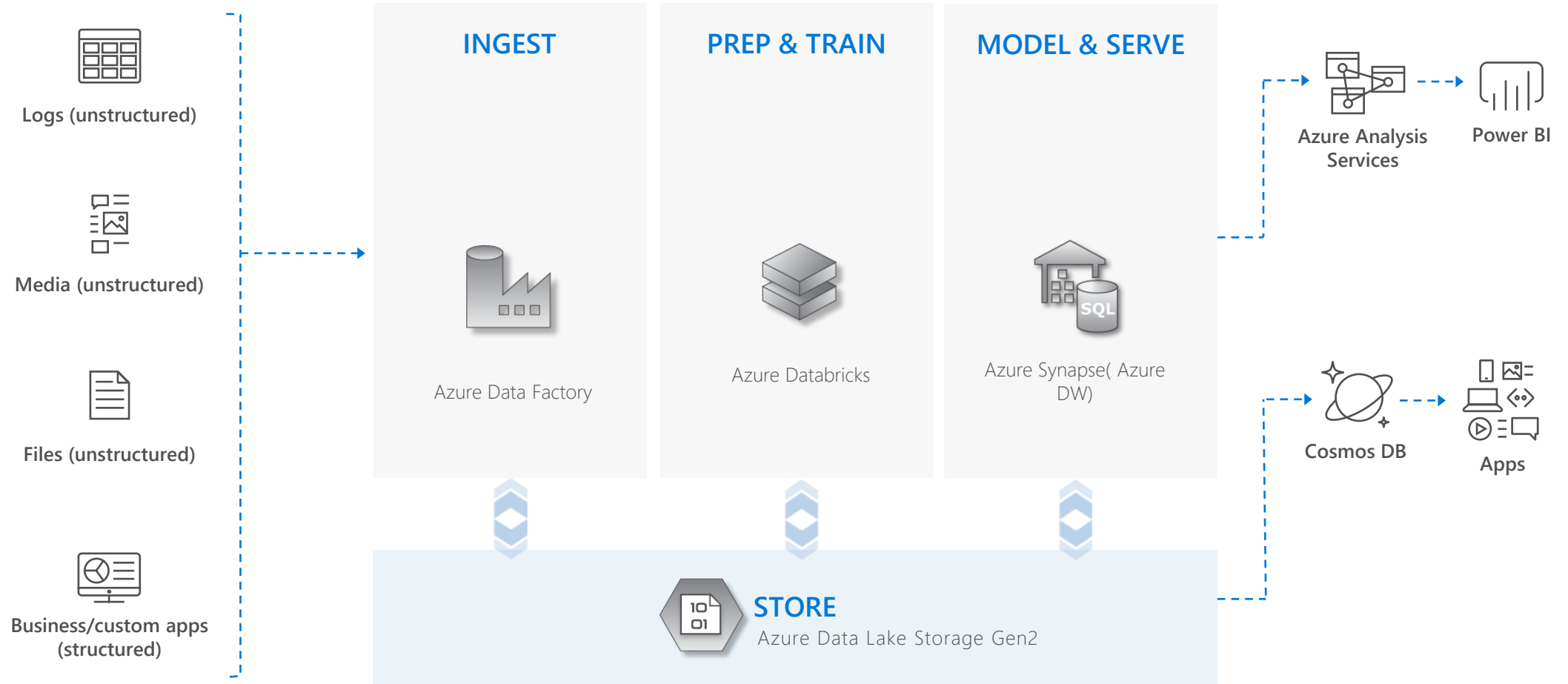
File system drivers

Endpoint: object store access
Blob API using [wasb\[s\]://](#)

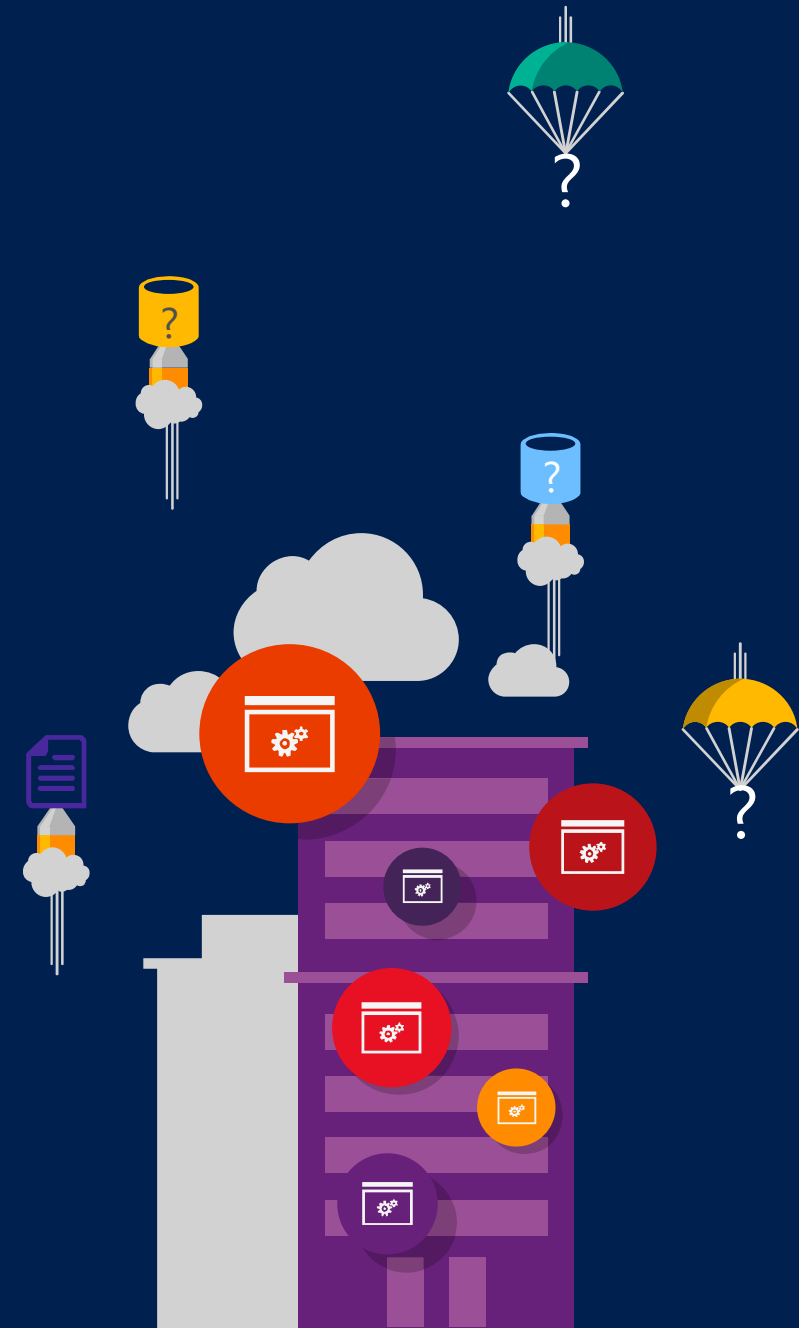
Endpoint: file system access (dfs)
ADLS Gen 2 API using [abfs\[s\]://](#)

End to End Analytics

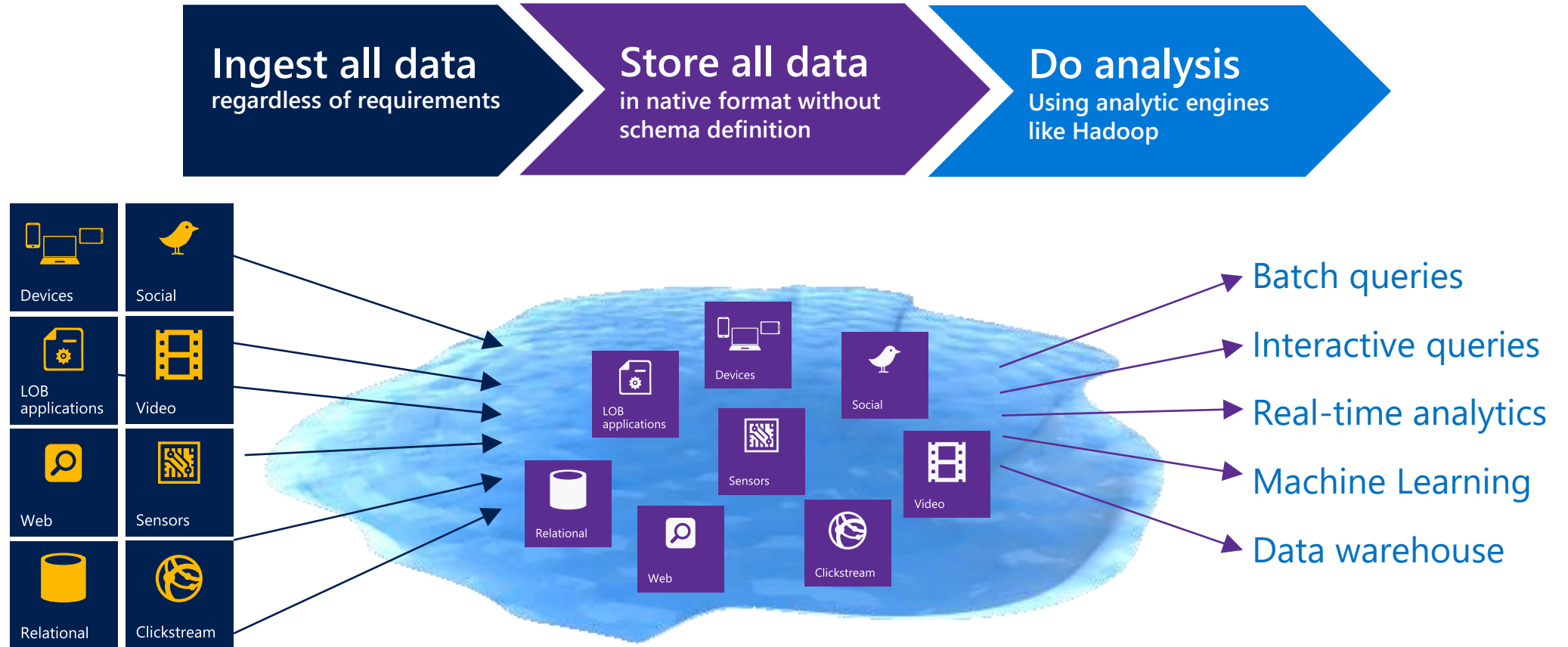
Common Data Lake Implementation



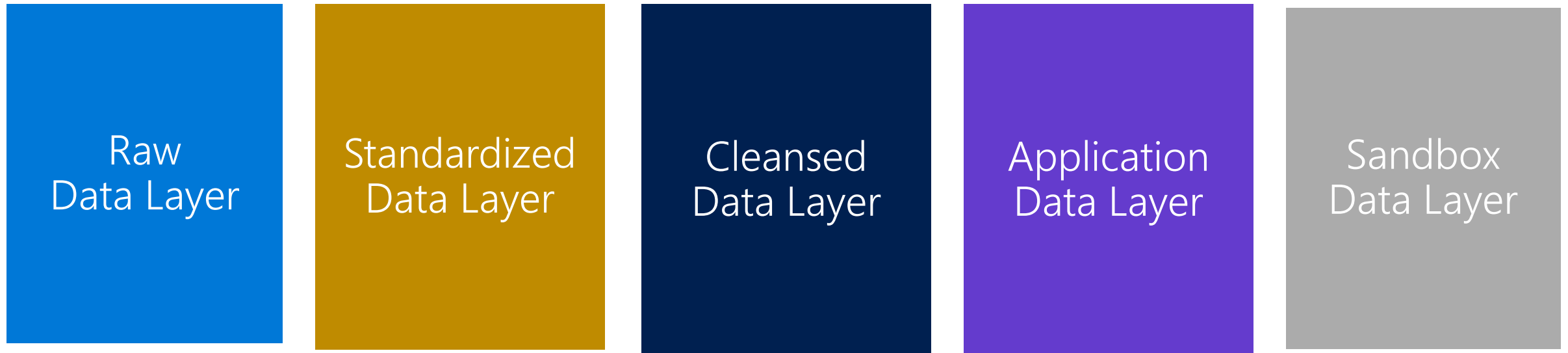
Data Lake: Design Concept



The "data lake" Uses A Bottoms-Up Approach



Data Lake Layers



Needs data governance so your data lake does not turn into a data swamp!

Raw Data Zone

- ✓ Exact copy of source data in native format (aka master dataset in the batch layer)
- ✓ Immutable to change
- ✓ History retained indefinitely
- ✓ Data access is highly limited to few people
- ✓ Everything downstream can be regenerated from raw

Transient/Temp Zone

- ✓ Selectively utilized
- ✓ Separation of "new data" from "raw data" to ensure data consistency
- ✓ Transient low-latency data (aka speed layer)
- ✓ Data quality validations

Master Data Zone

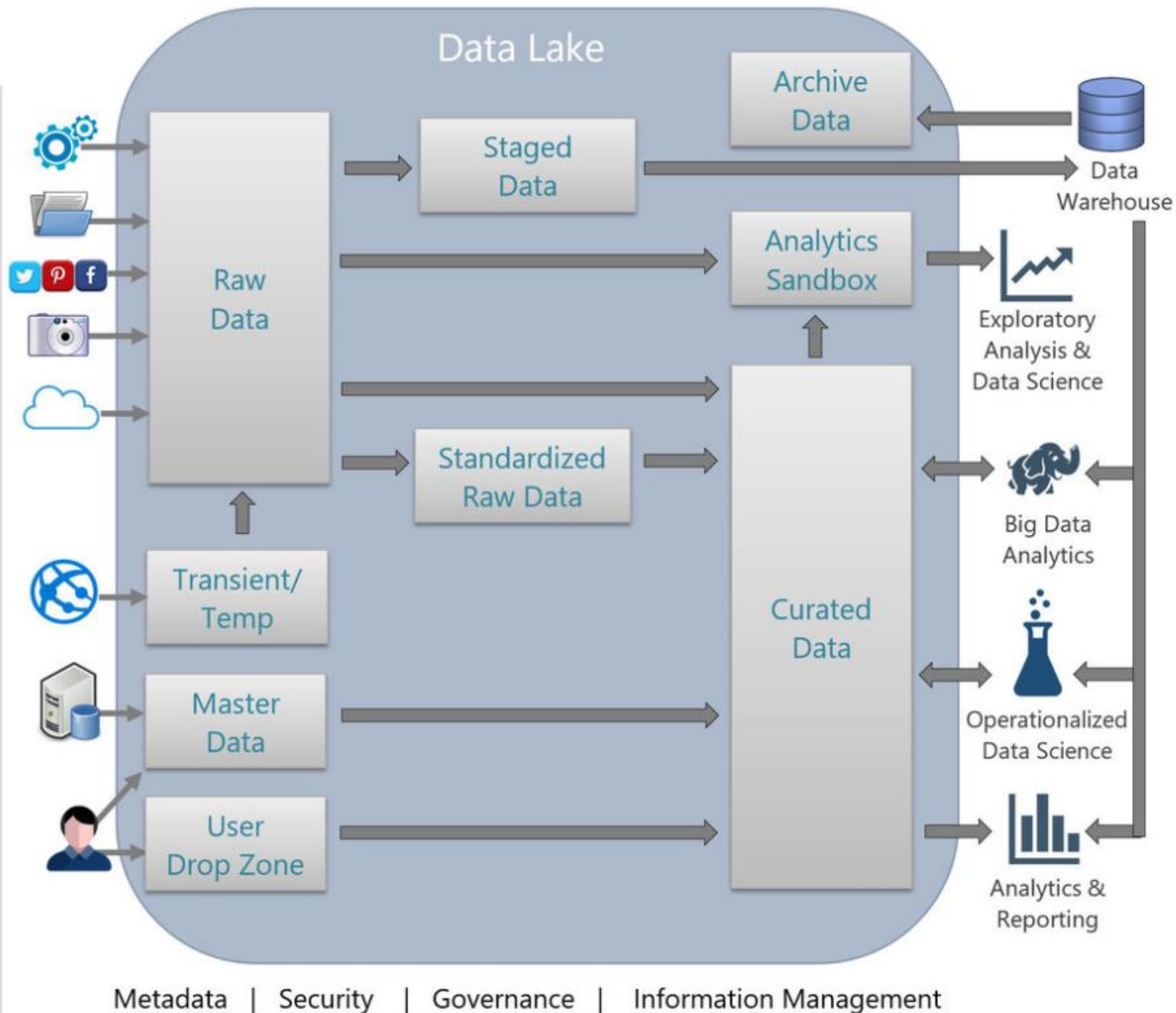
- ✓ Reference data

User Drop Zone

- ✓ Manually generated data

Staged Data Zone

- ✓ Data staged for a specific purpose or application



Standardized Raw Data

- ✓ Raw data which varies in format or schema, such as JSON which is standardized into columns & rows (aka "semantic normalization")
- ✓ File consolidations of data (i.e., to overcome performance issues with many small files)

Archive Data Zone

- ✓ Active archive of aged data, available for querying when needed

Analytics Sandbox

- ✓ Workspace for exploratory data science & analytics
- ✓ Valuable efforts are productionized to the curated data zone

Curated Data Zone

- ✓ Cleansed and transformed data, organized for optimal data delivery (aka serving layer)
- ✓ Supports self-service
- ✓ Standard security, change management, and governance

Organizing a Data Lake – Folder structure

Objectives

- ✓ Plan the structure based on optimal data retrieval
- ✓ Avoid a chaotic, unorganized data swamp

Common ways to organize the data:

Time Partitioning

Year/Month/Day/Hour/Minute

Subject Area

Security Boundaries

Department
Business unit
etc...

Downstream App/Purpose

Data Retention Policy

Temporary data
Permanent data
Applicable period (ex: project lifetime)
etc...

Business Impact / Criticality

High (HBI)
Medium (MBI)
Low (LBI)
etc...

Owner / Steward / SME

Probability of Data Access

Recent/current data
Historical data
etc...

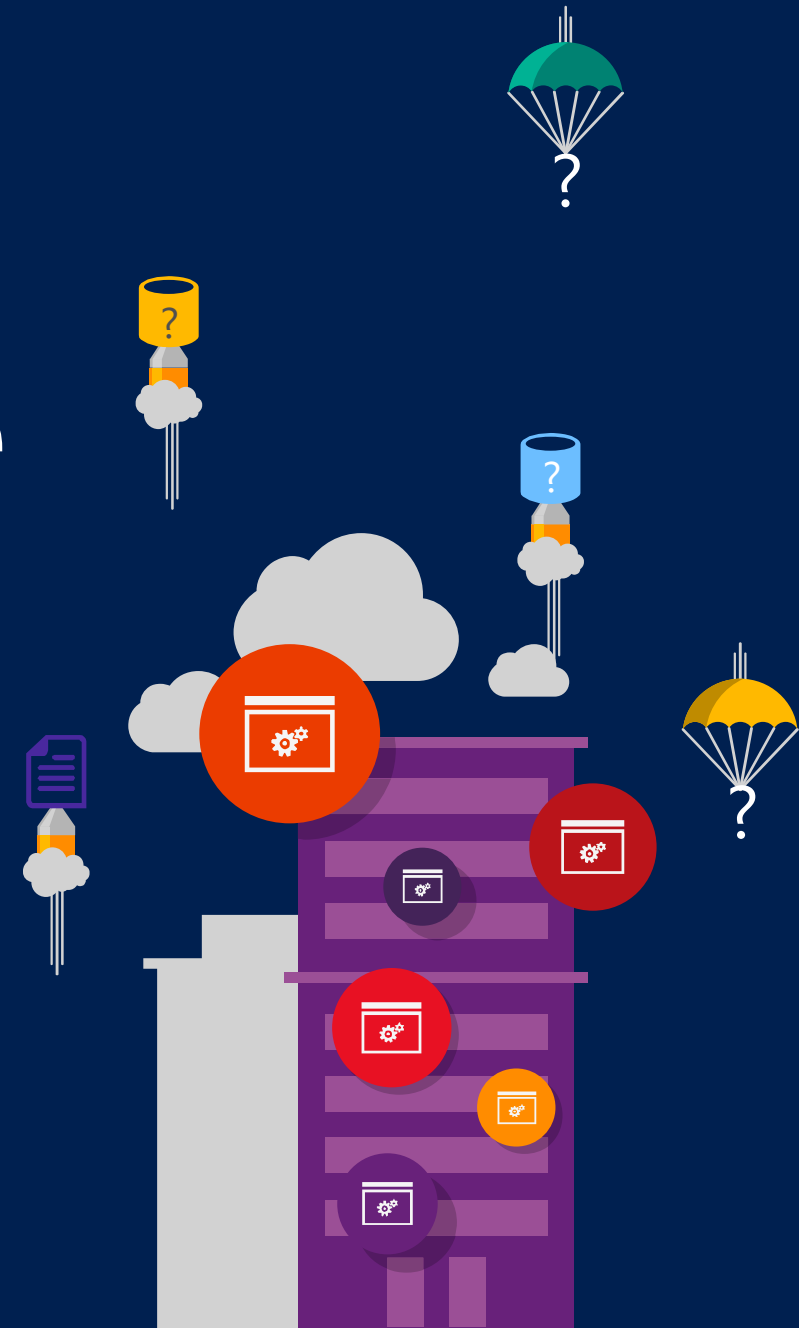
Confidential Classification

Public information
Internal use only
Supplier/partner confidential
Personally identifiable information (PII)
Sensitive – financial
Sensitive – intellectual property
etc...

Planning a Data Lake

- Ingestion needs (push / pull via streaming or batch)
- Enrichment, standardization, cleansing, and curation needs
- Security around data access
- Data retention and archival policies
- Encryption requirements
- Governance
- Data quality
- Master data management
- Metadata management
- Organization of data for optimal data retrieval
- Technology choices comprising the overall data lake architecture (HDFS, Hadoop components, NoSQL DBs, relational DBs, etc.)

Data Lake: Best Practice



Example: Raw Data Layer

Raw Data

Organizational Unit

Subject Area

Original Data Source

Object

Date Loaded

File(s)

This structure could translate into something like:

East Division

Sales

Salesforce

CustomerContacts

2016

2016_12

2016_12_01

CustContact_2016_12_01.txt

Example: Curated Data Layer

Curated Data

Purpose

Type

Snapshot Date (if applicable)

File(s)

This structure could translate into something like:

Sales Trending Analysis

Summarized

2016_12_01

SalesTrend_2016_12_01.txt

Example: Batch Jobs Structure

IOT use case:

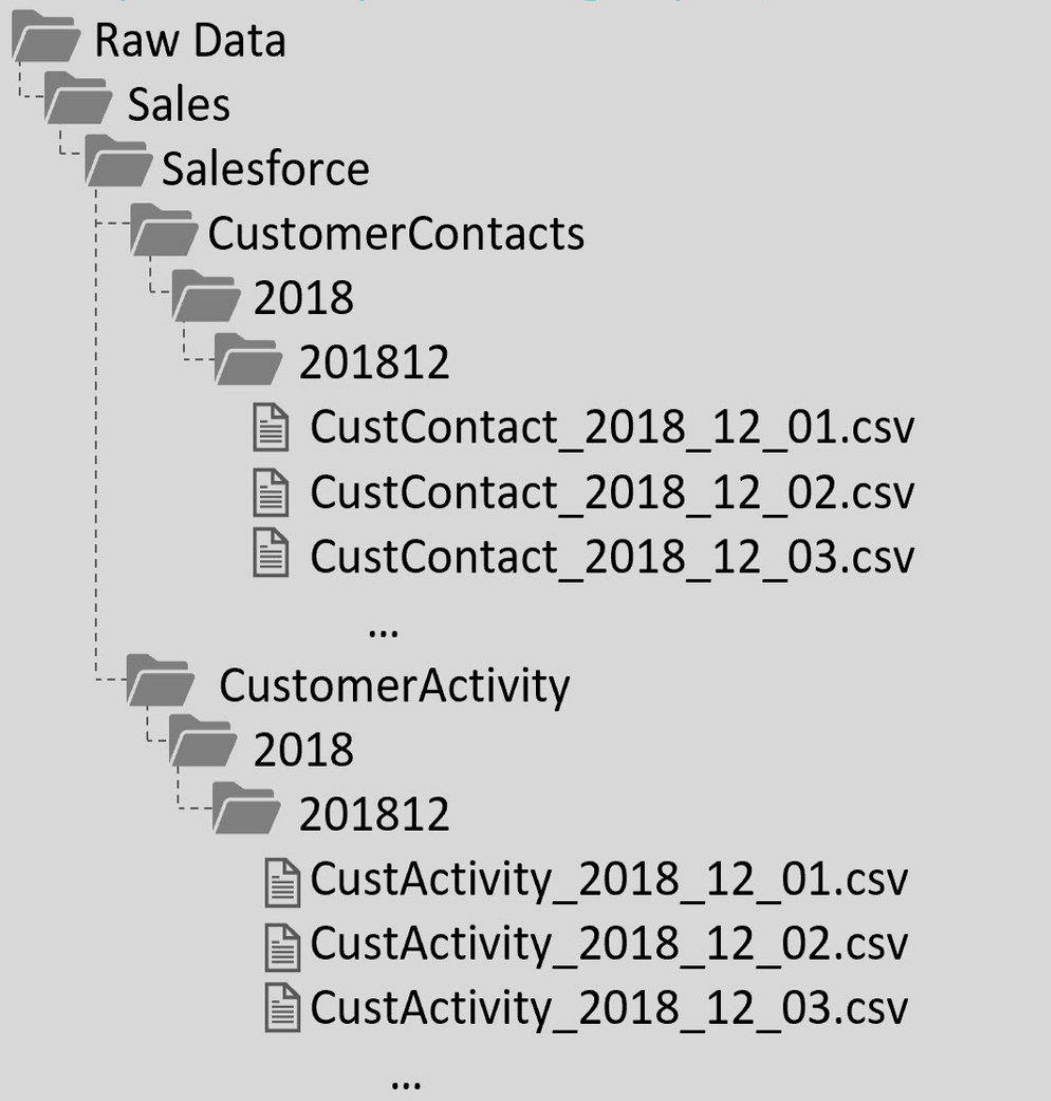
*{Region}/{SubjectMatter(s)}/In/{yyyy}/{mm}/{dd}/{hh}/
{Region}/{SubjectMatter(s)}/Out/{yyyy}/{mm}/{dd}/{hh}/
{Region}/{SubjectMatter(s)}/Bad/{yyyy}/{mm}/{dd}/{hh}/*

Marketing use case:

*NA/Extracts/ACMEPaperCo/In/2017/08/14/updates_08142017.csv
NA/Extracts/ACMEPaperCo/Out/2017/08/14/processed_updates_08142017.csv*

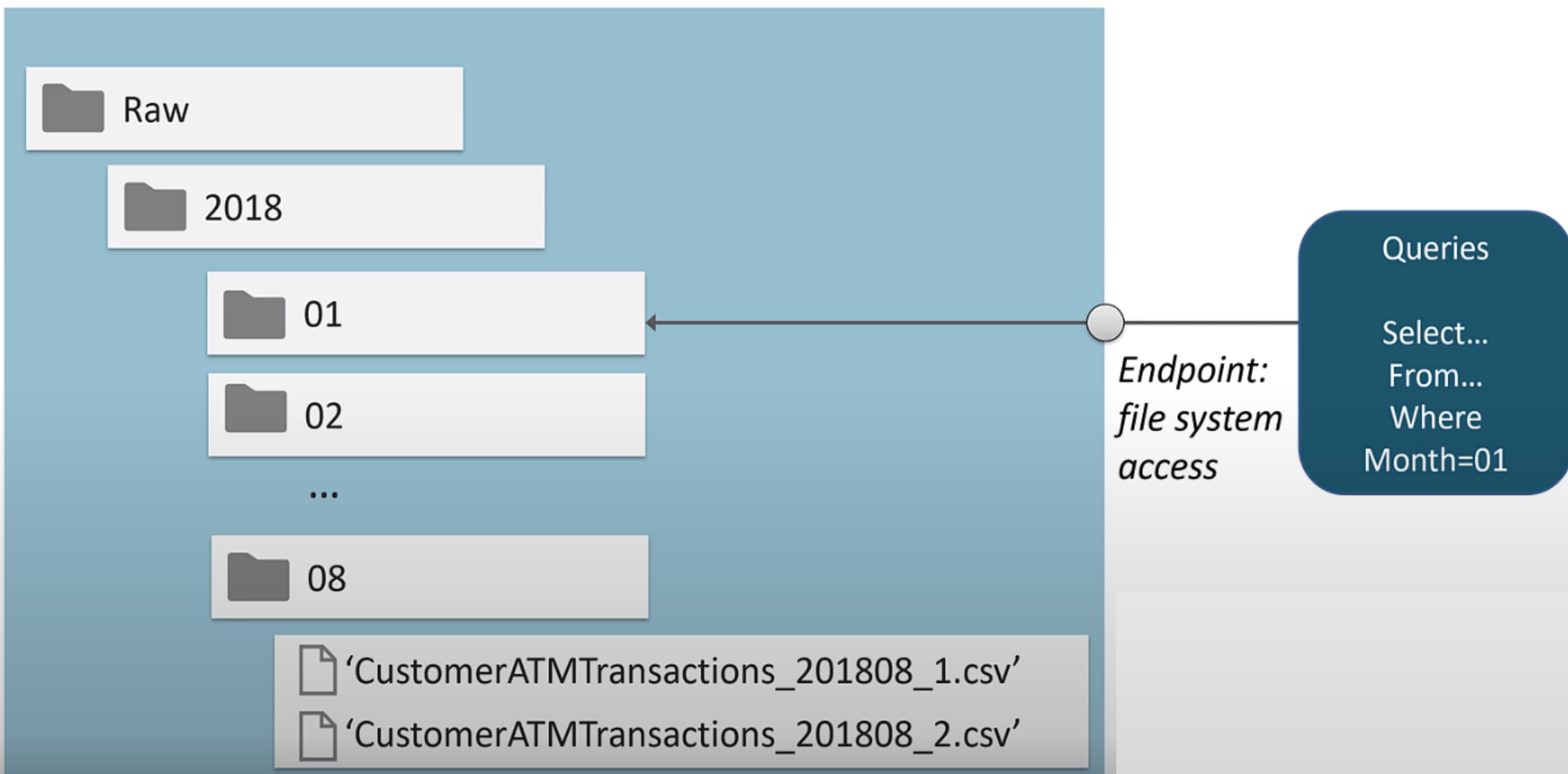
Example: Partitioning by date

Example of date partitioning at year/month level:



Multi-Modal Advantages with ADLS Gen2 – Example

Leverage partition scans & partition pruning to improve query performance:



Is it a good idea If I need a separate dev, test, prod environment, how would this usually be handled?

Separate environments are handled with separate services. For instance, in Azure, that would be **3 separate Azure Data Lake Storage resources** (which might be in the same subscription or different subscriptions).

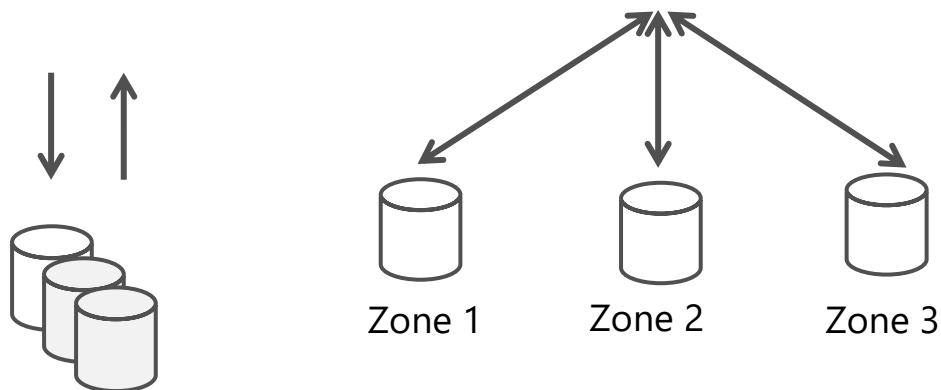
How much do I need to be concerned with the similarity of file contents within a folder?

The general rule is for all files to have the same format underneath a folder node.

Data lakes are supposed to be agile. So I don't need to worry about naming conventions, right?

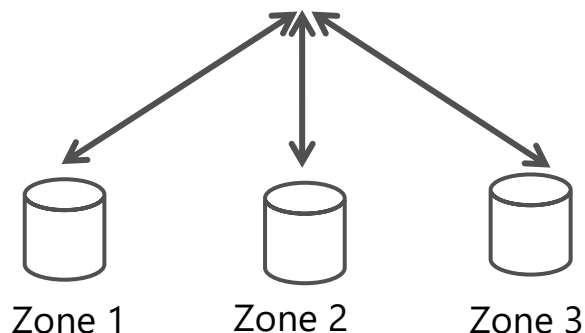
As long as you're consistent!

ADLS Gen2 Replication Options



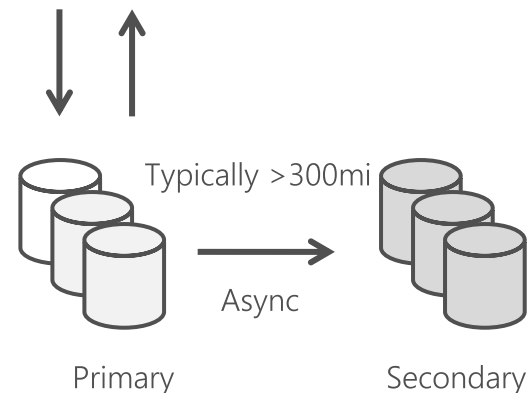
LRS

- Multiple replicas across a datacenter
- Protect against disk, node, rack failures
- Write is ack'd when all replicas are committed
- Superior to dual-parity RAID
- 11 9s of durability
- SLA: 99.9%



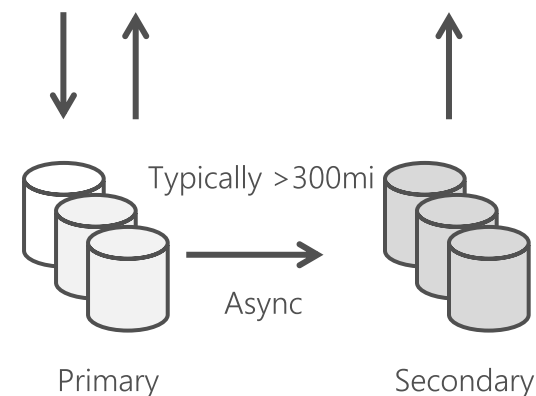
ZRS

- Replicas across 3 Zones
- Protect against disk, node, rack and zone failures
- Synchronous writes to all 3 zones
- 12 9s of durability
- Available in 8 regions
- SLA: 99.9%



GRS

- Multiple replicas across each of 2 regions
- Protects against major regional disasters
- Asynchronous to secondary
- 16 9s of durability
- SLA: 99.9%



RA-GRS

- GRS + Read access to secondary
- Separate secondary endpoint
- RPO delay to secondary can be queried
- SLA: 99.99% (read), 99.9% (write)

New: Customer controlled failover to GRS location
Geo Zone Redundant Storage (GZRS, RA-GZRS)

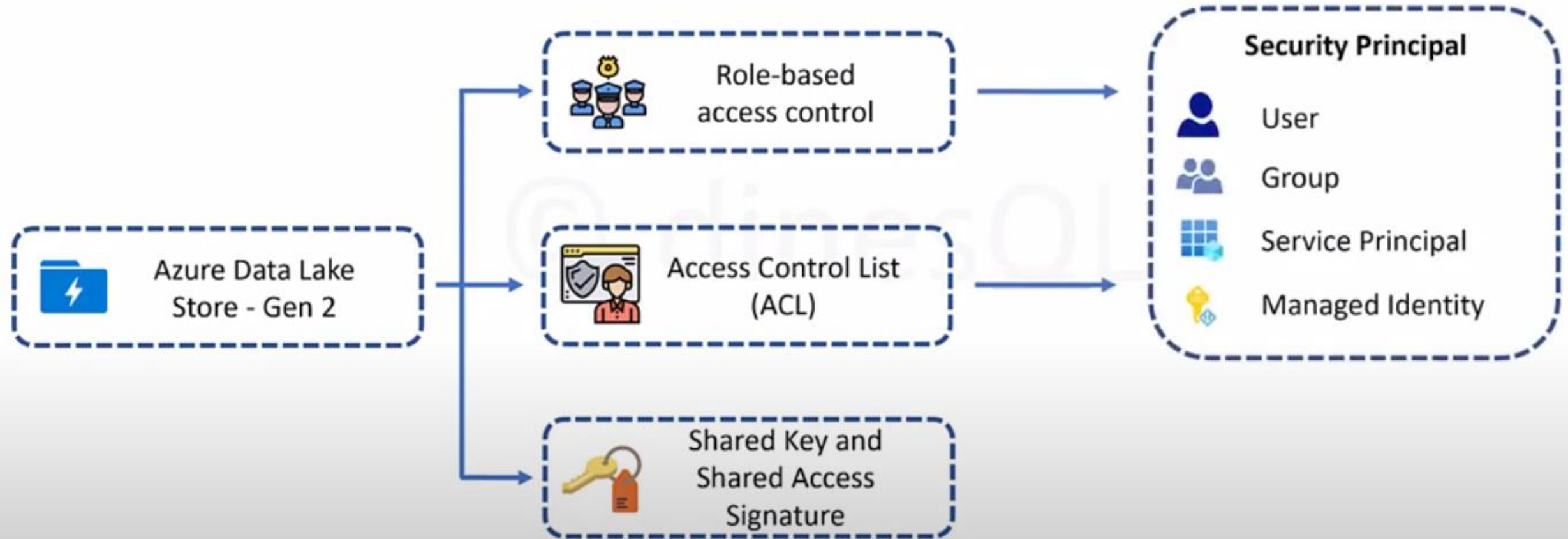
Comparing storage options

	Premium performance	Hot tier	Cool tier	Archive tier
Availability	99.9%	99.9%	99%	Offline
Availability (RA-GRS reads)	N/A	99.99%	99.9%	Offline
Usage charges	Higher storage costs, lower access and transaction cost	Higher storage costs, lower access, and transaction costs	Lower storage costs, higher access, and transaction costs	Lowest storage costs, highest access, and transaction costs
Minimum object size	N/A	N/A	N/A	N/A
Minimum storage duration	N/A	N/A	30 days ¹	180 days
Latency (Time to first byte)	Single-digit milliseconds	milliseconds	milliseconds	hours ²

¹ Objects in the cool tier on GPv2 accounts have a minimum retention duration of 30 days. Blob storage accounts don't have a minimum retention duration for the cool tier.

² Archive Storage currently supports 2 rehydrate priorities, High and Standard, that offers different retrieval latencies. For more information, see [Rehydrate blob data from the archive tier](#).

Ways of Accessing ADLS Gen 2



Lifecycle Management

Add a rule

Action set

Filter set

Review + add

Each rule definition includes an action set and a filter set. The action set applies the tier or delete actions to the filtered set of objects. The filter set limits rule actions to a certain set of objects within a container or objects names.

Rule name *

Status

☐ Disabled

☒ Enabled

Blobs

☐ Move blob to cool storage

Days after last modification

☐ Move blob to archive storage

Days after last modification

☐ Delete blob

Days after last modification

Add a rule

Action set

Filter set

Review + add

Each rule definition includes an action set and a filter set. The action set applies the tier or delete actions to the filtered set of objects. The filter set limits rule actions to a certain set of objects within a container or objects names.

Supported blob types

Block Blob

Prefix match

Apply a rule to a container or a subset of virtual folders with the use of up to 10 prefixes as filters. By default, a rule will apply to the entire storage account.

☐ Browse

☐ Delete

☐ Path

Container/virtual folder

Insights

Gallery Customize Refresh Share Feedback Help

Time Range: Last 4 hours

- Overview
- Failures
- Performance
- Availability
- Capacity

Availability

100

Average | Percentage

Transactions

48

Count

Success E2E Latency

14.71

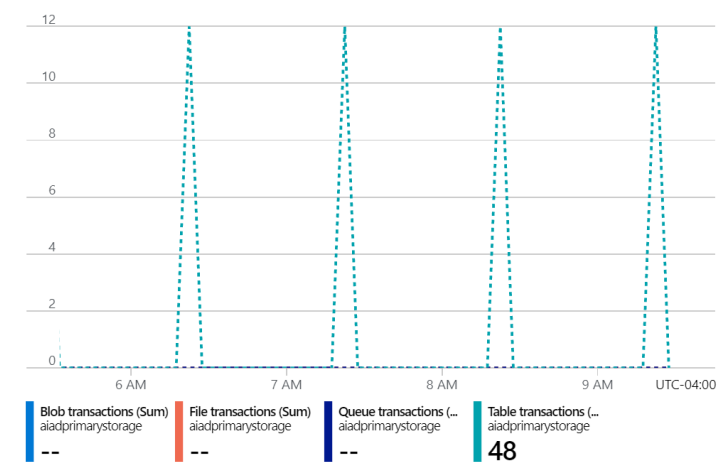
Average | Milliseconds

Success Server Latency

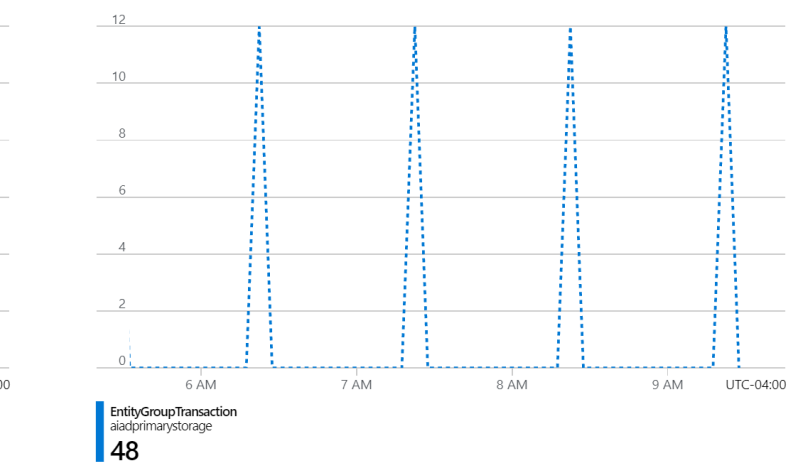
14.63

Average | Milliseconds

Transactions by storage type



Transactions by API name



Availability

Used capacity

Microsoft Azure Storage Explorer

