Data Lake & Big Data

Microsoft

1. Your name

2. Your company

3. Your role

4. Your background

5. What is your definition for Big Data?

Your name: *Liviana Zürcher*

Your company: *Microsoft*

Your role: *Technology Solution Professional – data platform*

Your background: *BI & DW*

5. What is your definition for Big Data? – *All data that I cannot process in a set timeline*

Use-Cases

Every Industry classification benefits from Big Data, Retail and Finance leads the way

Industry Sector	Primary Use-Cases
Retail	Demand prediction
	In-store analytics
	Supply chain optimization
	Customer retention
	Cost/Revenue analytics
	HR analytics
	Inventory control
Finance	Cyberattack Prevention
	Fraud detection
	Customer segmentation
	Market analysis
	Risk analysis
	Blockchain
	Customer retention
Healthcare	Fiscal control analytics
	Disease Prevention prediction and classification
	Clinical Trials optimization
	Patient load analysis
	Episode analytics
Public Sector	Revenue prediction
	Education effectiveness analysis
	Transportation analysis and prediction
	Energy demand and supply prediction and control
	Defense readiness predictions and threat analysis
Manufacturing	Predictive Maintenance (PdM)
	Anomaly Detection
	Pattern analysis
Agriculture	Food Safety analysis
	Crop forecasting
	Market forecasting
	Pipeline Optimization

Azure Data Lake gen 2

Azure SQL DW

Azure Data Factory

Cosmos DB

Azure Data Catalog

Big Data patterns

- Modern Data Warehouse
- Advance Analytics
- Real Time Analytics

Azure Data Lake gen 2

What makes a great Data Lake?

Massive scale	Secure	Optimized for Maximum Performance	Integration Friendly	Cost Effectiveness
PB Scale, data accessible everywhere, growth on demand	Granular security and protection against accidental data loss	Lightning quick job execution	Supports multiple methods of data ingress, processing, egress and visualization	Cloud economic model with the ability to intelligently manage costs

Rich Data Management and Governance

Azure Data Lake Storage Gen2

A "no-compromises" Data Lake: secure, performant, massively-scalable Data Lake storage that brings the cost and scale profile of object storage together with the performance and analytics feature set of data lake storage

SECURE

- ✓ Support for fine-grained ACLs, protecting data at the file and folder level
- ✓ Multi-layered protection via at-rest Storage Service encryption and Azure Active Directory integration

MANAGEABLE

- ✓ Automated Lifecycle Policy Management
- ✓ Object Level tiering

FAST

- ✓ Atomic file operations means jobs complete faster
- ✓ High throughput

SCALABLE

- ✓ No limits on data store size
- ✓ Global footprint (50 regions)

COST EFFECTIVE

- ✓ Object store pricing levels
- ✓ File system operations minimize transactions required for job completion

INTEGRATION READY

- ✓ Optimized for Spark and Hadoop Analytic Engines
- ✓ Tightly integrated with Azure end to end analytics solutions

Convergence of two Storage Services

Azure Blob Storage

General Purpose Object Storage

Global scale – All Azure regions

Full BCDR capabilities

Tiered - Hot/Cool/Archive

Cost Efficient

Large partner ecosystem

Azure Data Lake Store

Optimized for Big Data analytics

Built for Hadoop

Hierarchical namespace

ACLs, AAD and RBAC

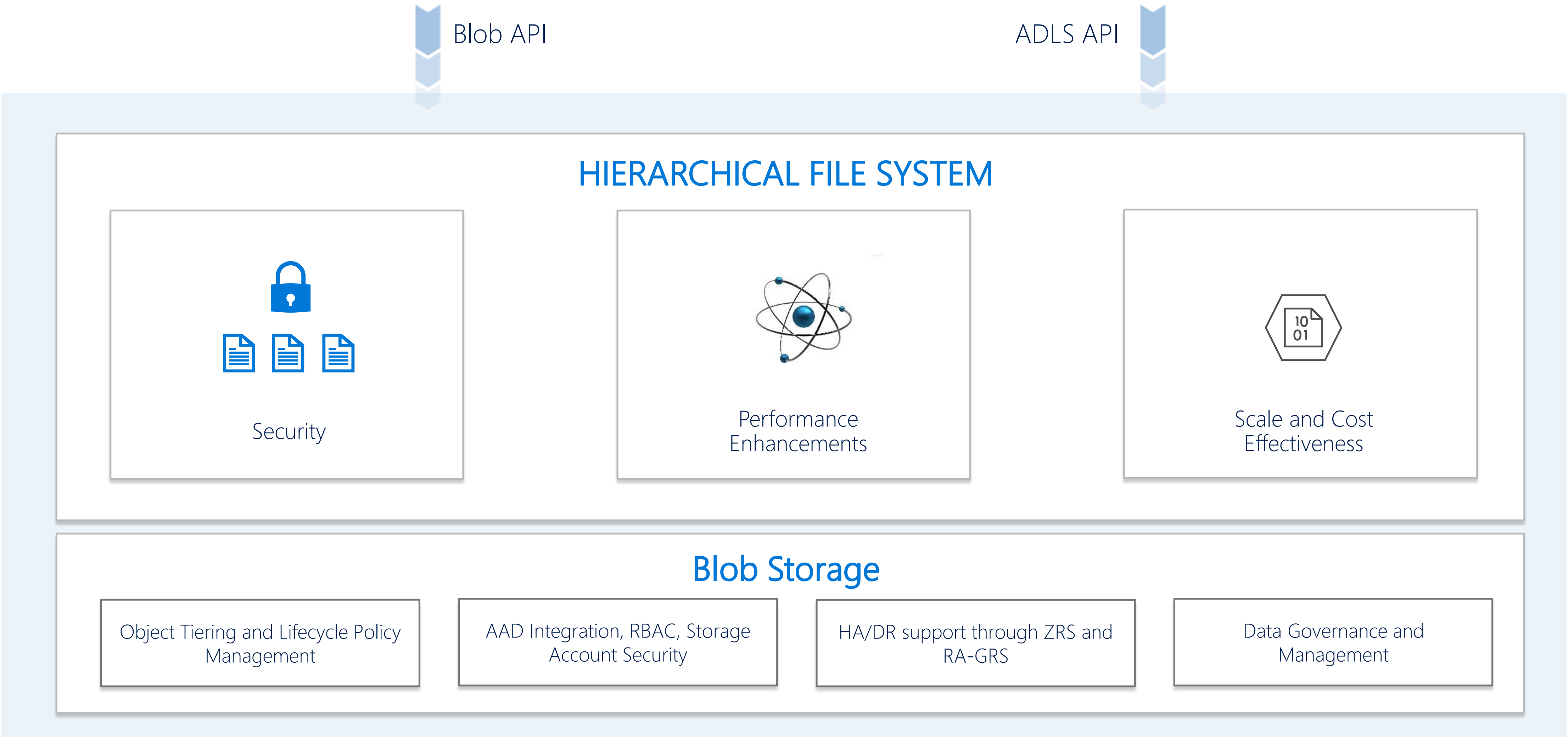
Performance tuned for big data

Very high scale capacity and throughput

Azure Data Lake Storage Gen2

The best of Blobs and ADLS

Azure Data Lake Storage Gen2 architecture



Blob API

ADLS API

HIERARCHICAL FILE SYSTEM

Security

Performance Enhancements

Scale and Cost Effectiveness

Blob Storage

Object Tiering and Lifecycle Policy Management

AAD Integration, RBAC, Storage Account Security

HA/DR support through ZRS and RA-GRS

Data Governance and Management

Data Security in ADLS Gen2

- Azure Active Directory
- Azure RBAC and POSIX-compliant ACLs
  - Integrates with analytics frameworks for end-user authorization

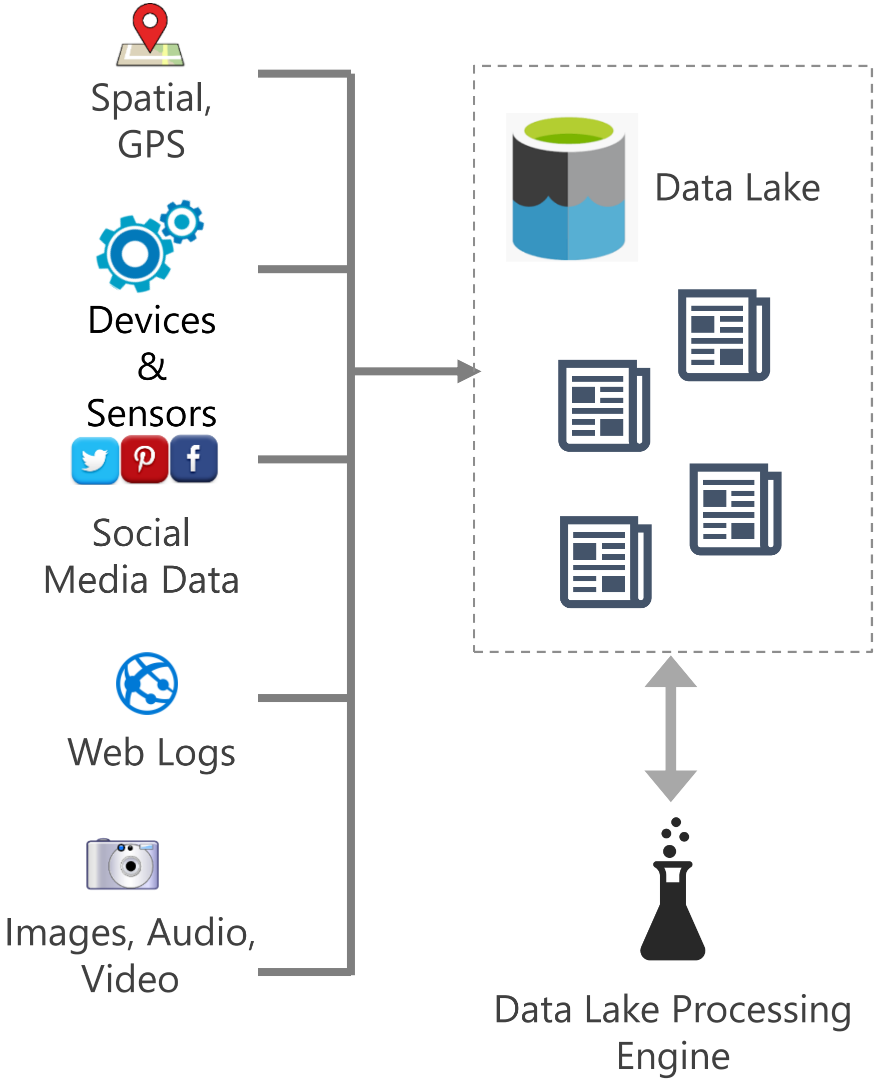
- But security is much more than access control…

- Encryption at rest: Customer or Microsoft managed keys
- Encryption in transit: TLS
- Transport-level protection: VNet service endpoints
- Transport-level protection: VNet service policies (coming soon)

Data Lake Objectives

(1/2)

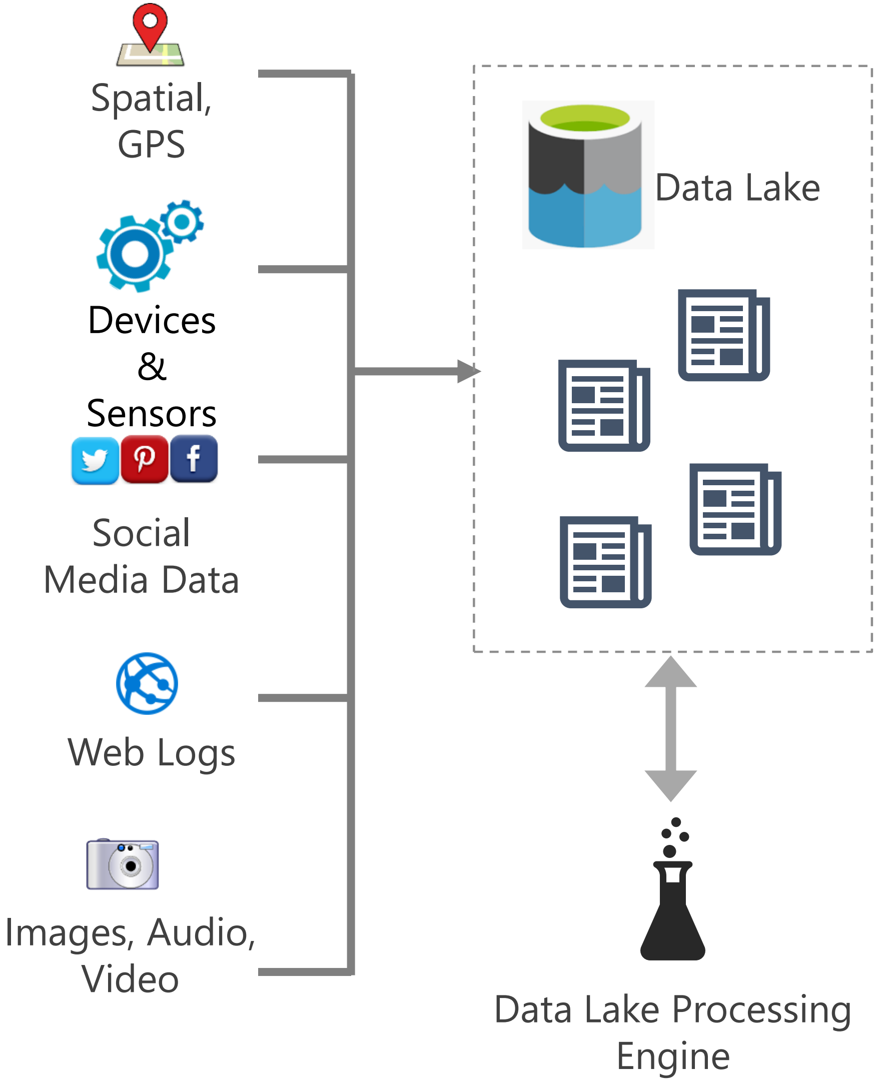
- ✓ Reduce up-front effort by ingesting data in any format, any size, without requiring a schema initially
- ✓ Make acquiring new data easy, so it can be available for data science & analysis quickly
- ✓ Store large volume of multi-structured data in its native format
- ✓ Storage for additional types of data which were historically difficult to obtain or store
- ✓ Reduce the long-term ownership cost of data management & storage



Data Lake Objectives

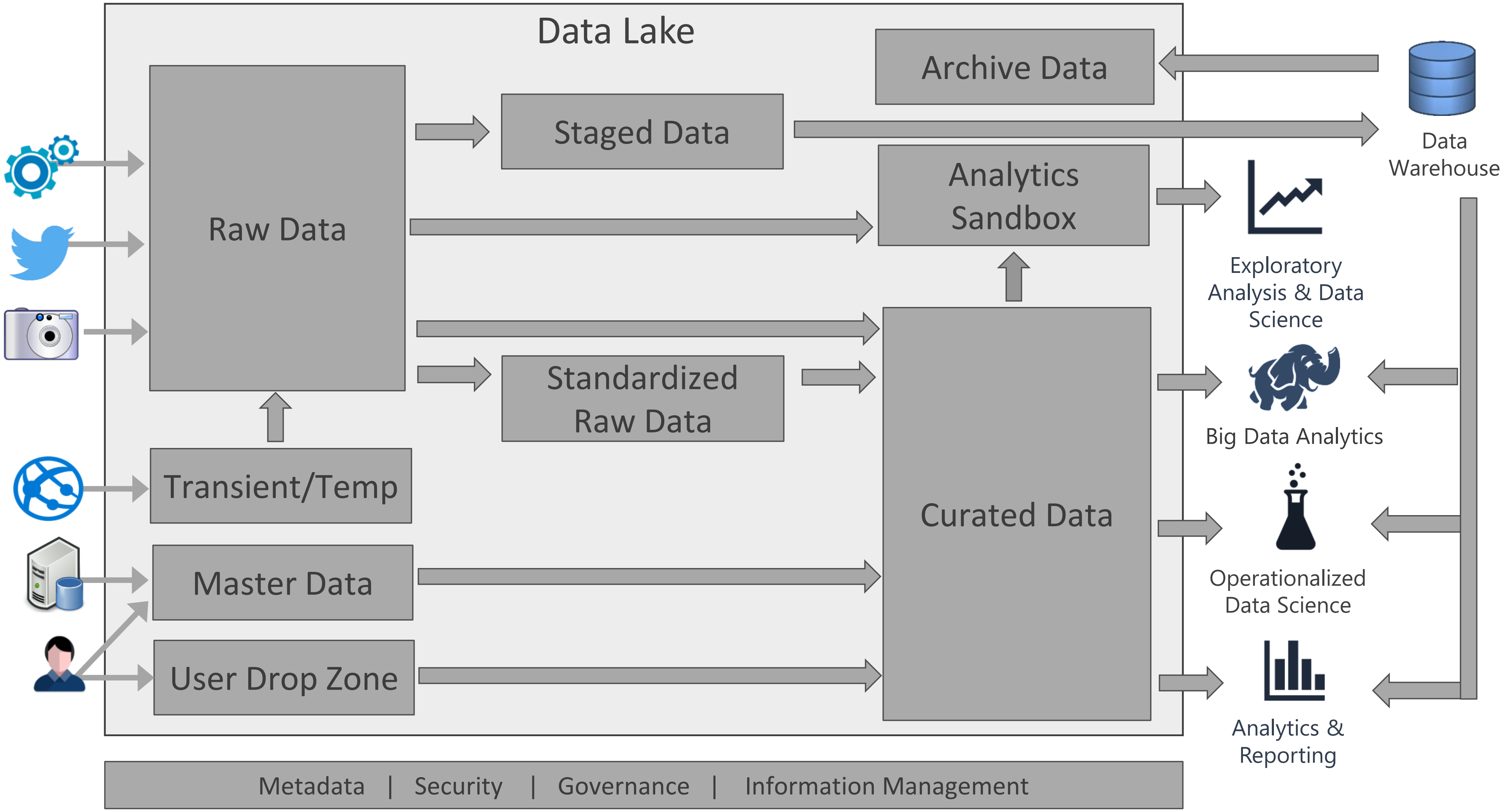
(2/2)

- ✓ Schema-on-read: **Defer work to 'schematize'** after value & requirements are known
- ✓ Achieve **agility** faster than a traditional data warehouse can to speed up **decision-making** ability
- ✓ Access to **low-latency** data
- ✓ Different / new **value proposition** vs. traditional data warehousing
- ✓ Facilitate **advanced analytics** scenarios



Designing the Structure of a Data Lake

Designing the Zones of a Data Lake



Data Lake

Raw Data

Staged Data

Archive Data

Analytics Sandbox

Standardized Raw Data

Curated Data

Transient/Temp

Master Data

User Drop Zone

Data Warehouse

Exploratory Analysis & Data Science

Big Data Analytics

Operationalized Data Science

Analytics & Reporting

Metadata | Security | Governance | Information Management

What are some ways we could potentially organize data in a data lake?

Organizing a Data Lake (1/7)

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…

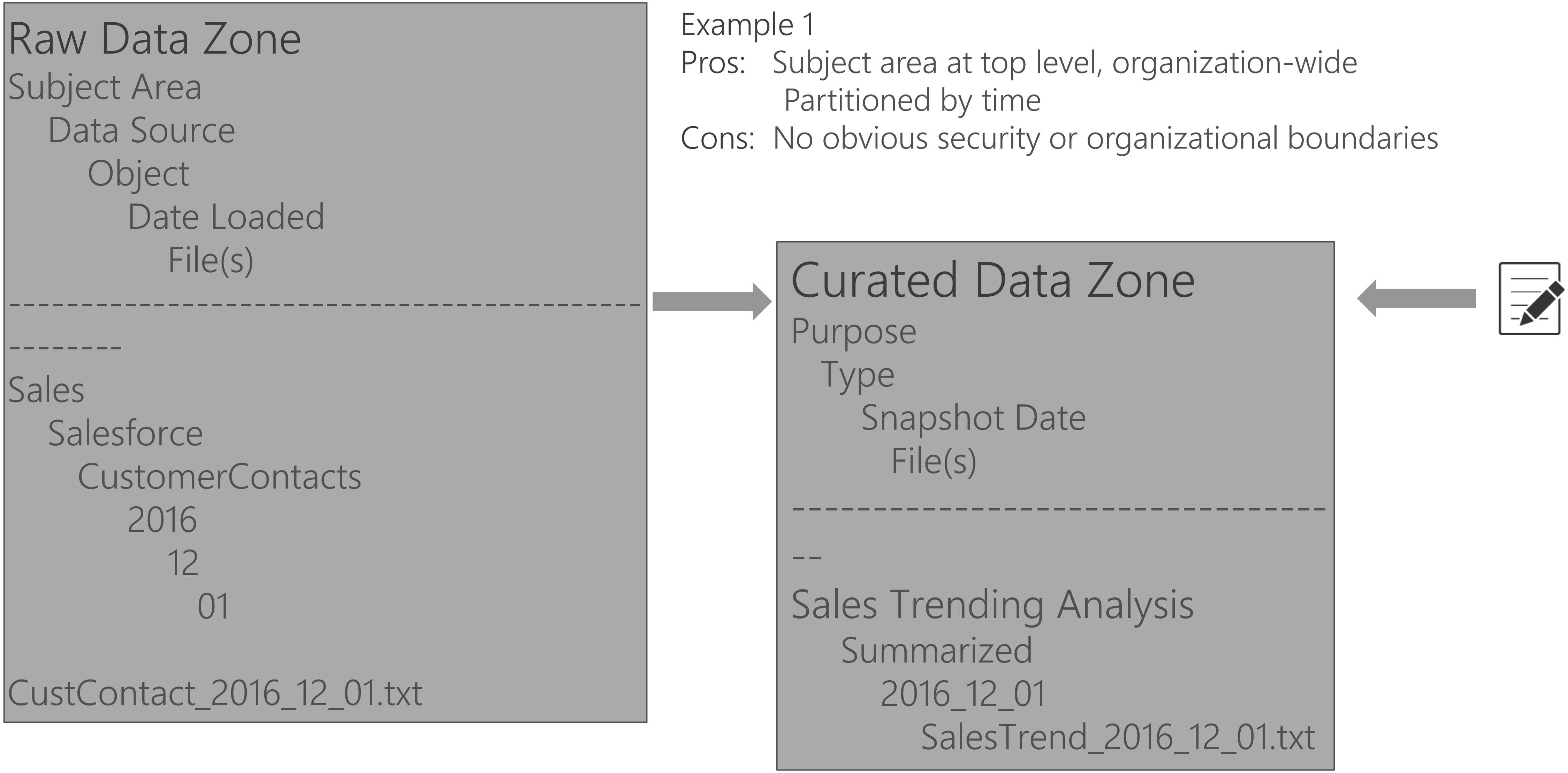
Owner / Steward / SME

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…

Organizing a Data Lake (2/7)

Example 1
Pros: Subject area at top level, organization-wide
Partitioned by time
Cons: No obvious security or organizational boundaries



Raw Data Zone

Subject Area
Data Source
Object
Date Loaded
File(s)

Sales
Salesforce
CustomerContacts
2016
12
01

CustContact_2016_12_01.txt

Curated Data Zone

Purpose
Type
Snapshot Date
File(s)

--

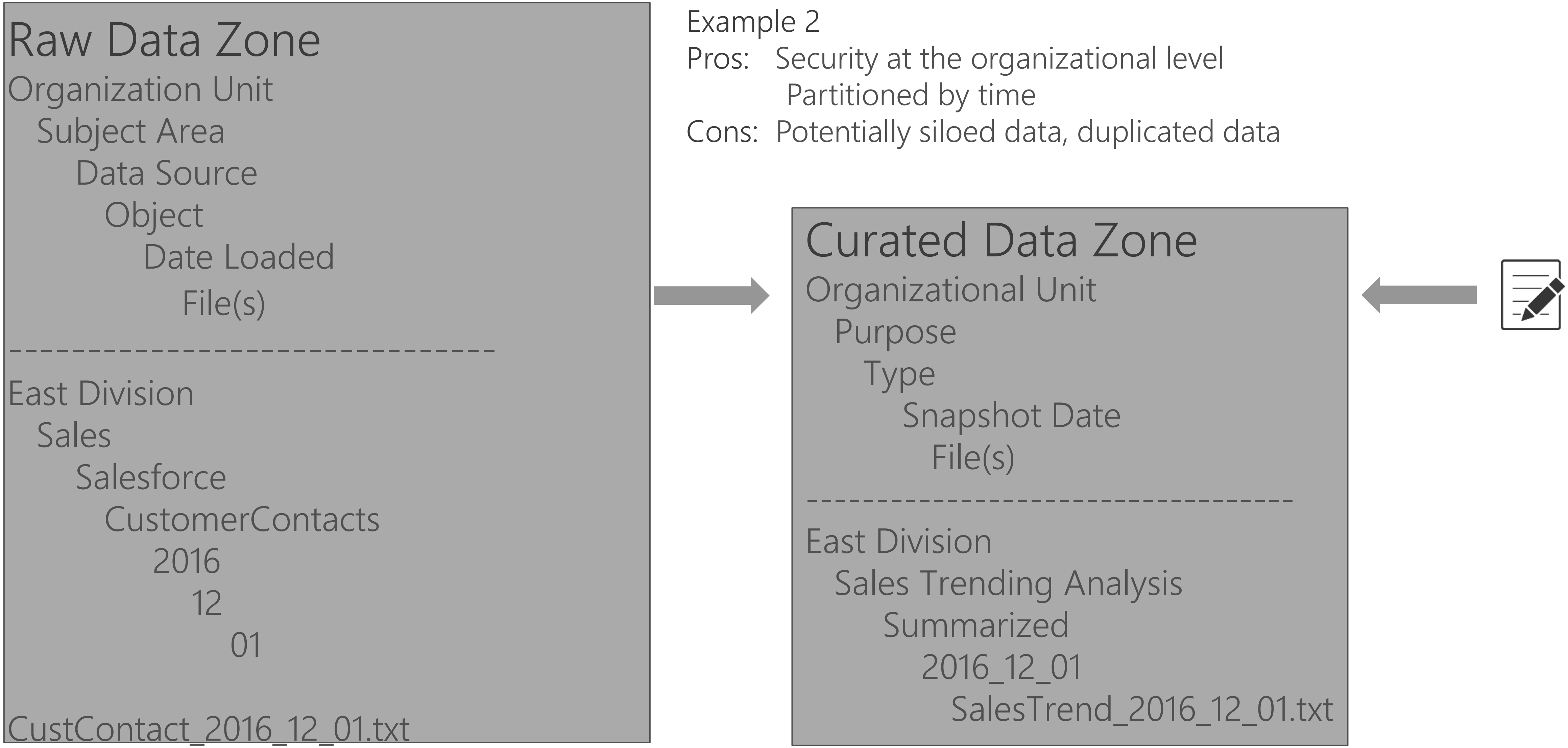
Sales Trending Analysis
Summarized
2016_12_01
SalesTrend_2016_12_01.txt

Organizing a Data Lake (3/7)

Example 2

Pros: Security at the organizational level
Partitioned by time

Cons: Potentially siloed data, duplicated data



Raw Data Zone

- Organization Unit
  - Subject Area
    - Data Source
      - Object
        - Date Loaded
          - File(s)

- East Division
  - Sales
    - Salesforce
      - CustomerContacts
        - 2016
          - 12
            - 01

CustContact_2016_12_01.txt

Curated Data Zone

- Organizational Unit
  - Purpose
    - Type
      - Snapshot Date
        - File(s)

- East Division
  - Sales Trending Analysis
    - Summarized
      - 2016_12_01
        - SalesTrend_2016_12_01.txt

Organizing a Data Lake (4/7)

Example 3

Pros: Segregates records coming in, going out, as well as error records

Time partitioning can go down to the hour, or even minute level, depending on volume (ex: IoT data)

Cons: Not obvious by the names what the purpose of ‘out’ is (which could be ok if numerous downstream applications utilize the same ‘out’ data)

Raw Data Zone

- Organization Unit
  - Subject Area
    - In
      - YYYY
        - MM
          - DD
            - HH
              - File(s)

- Organization Unit
  - Subject Area
    - Out
      - YYYY
        - MM
          - DD
            - HH
              - File(s)

- Organization Unit
  - Subject Area
    - Error
      - YYYY
        - MM
          - DD
            - HH
              - File(s)

Organizing a Data Lake (5/7)

- Subject Area 1
  - RawData
    - YYYY
      - MM
  - CuratedData
  - MasterData
  - StagedData

- Subject Area 2
  - RawData
    - YYYY
      - MM
  - CuratedData
  - MasterData
  - StagedData

Example 4
Zones are a logical need, but they don’t necessarily have to be at the top of the structure
Pros: Security by subject area
Cons: All raw data is not centralized

Organizing a Data Lake (6/7)

Do:

- ✓ Hyper-focus on ease of data discovery & retrieval – will one type of structure make more sense?
- ✓ Focus on security implications early – what data redundancy is allowed in exchange for security
- ✓ Include data lineage & relevant metadata with the data file itself whenever possible (ex: columns indicating source system where the data originated, source date, processed date, etc)
- ✓ Include the time element in ***both*** the folder structure & the file name
- ✓ Be liberal yet disciplined with folder structure (lots of nests are ok)
- ✓ Clearly separate out the zones so governance & policies can be applied separately
- ✓ Register the curated data with a catalog (ex: Azure Data Catalog) to document the metadata–a data catalog is even more important with a data lake
- ✓ Implement change management for migrating from a sandbox zone (discourage production use from the sandbox)
- ✓ Assign a data owner & data archival policies as part of the structure, or part of the metadata

Organizing a Data Lake (7/7)

Don't:

× Do not combine mixed formats in a single folder structure

  ✓ If it's looping through all files in a folder schema-on-read will fail if it finds a different format

  ✓ Files in one folder should all be able to be traversed with the same script

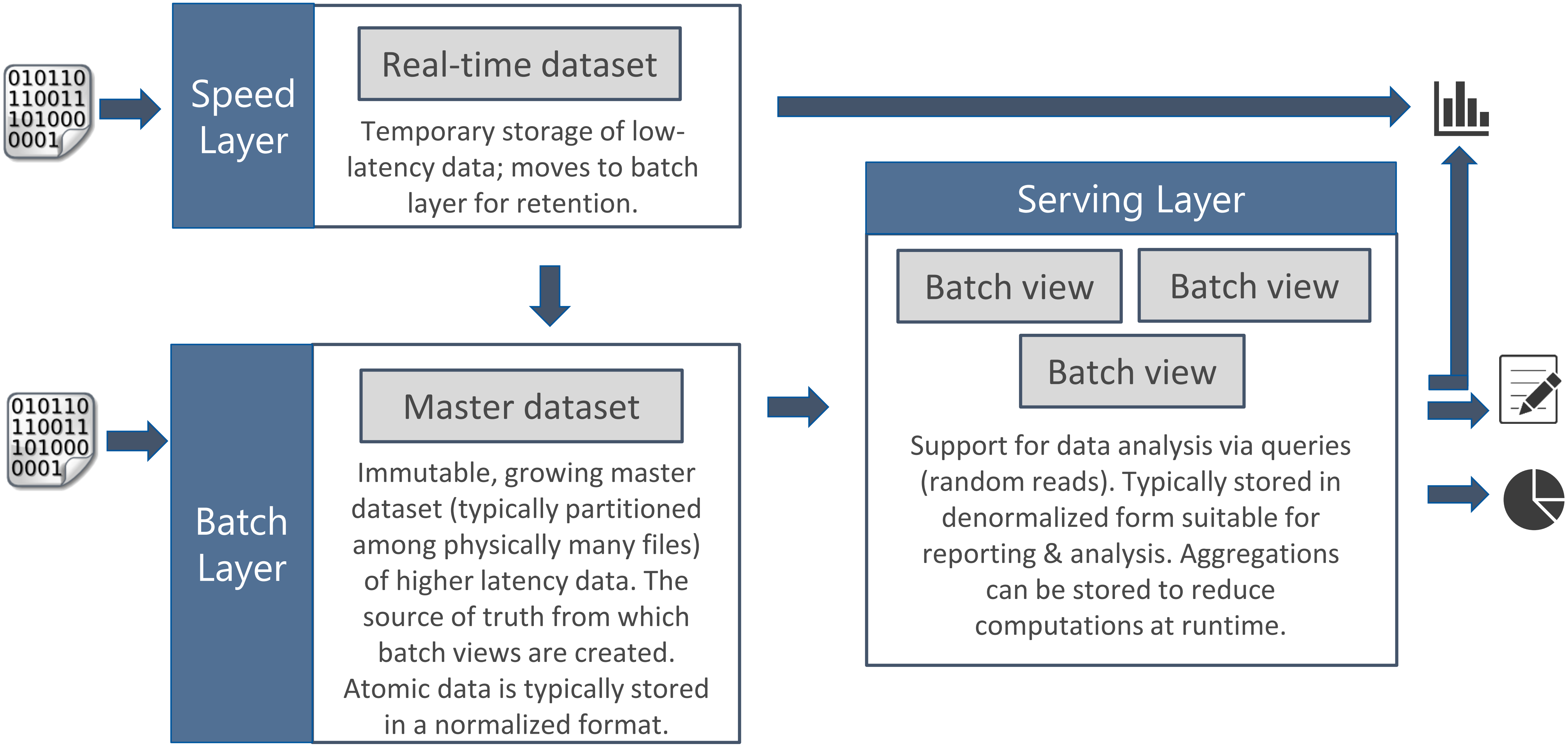
× Do not put your date partitions at the beginning of the file path -- it's much easier to organize & secure by subject area/department/etc if dates are the lowest folder level

Optimal for top level security:	Tedious for enforcing security:
\SubjectArea\YYYY\MM\DD\FileData_YYYY_MM_DD.txt	\YYYY\MM\DD\SubjectArea\FileData_YYYY_MM_DD.txt

× Do not neglect naming conventions. You might use camel case, or you might just go with all lower case – either is ok, as long as you're consistent because some languages are case-sensitive

Following
Big Data Principles
When Designing
A Data Lake

Lambda Architecture



Speed Layer

Real-time dataset

Temporary storage of low-latency data; moves to batch layer for retention.

Batch Layer

Master dataset

Immutable, growing master dataset (typically partitioned among physically many files) of higher latency data. The source of truth from which batch views are created. Atomic data is typically stored in a normalized format.

Serving Layer

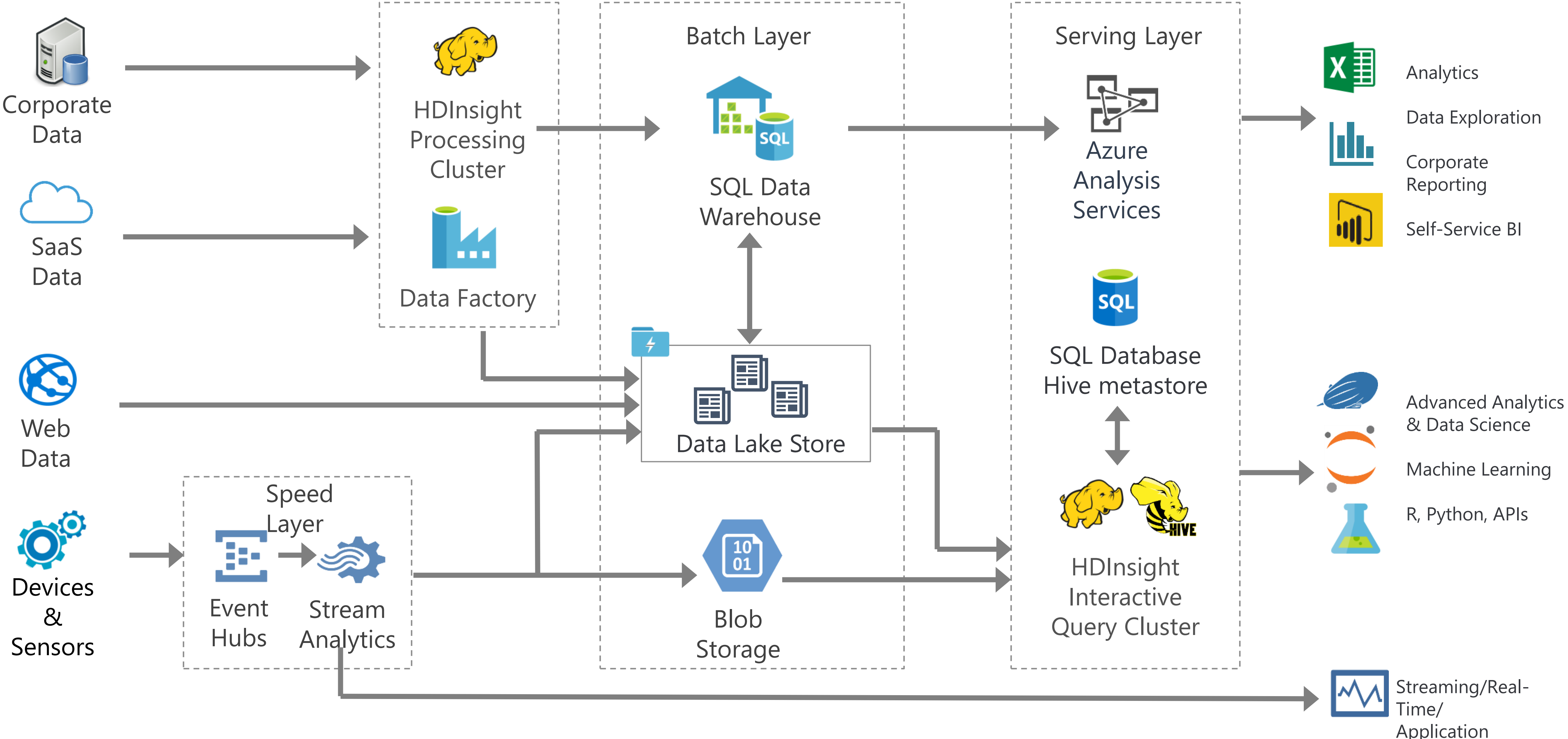
Batch view

Batch view

Batch view

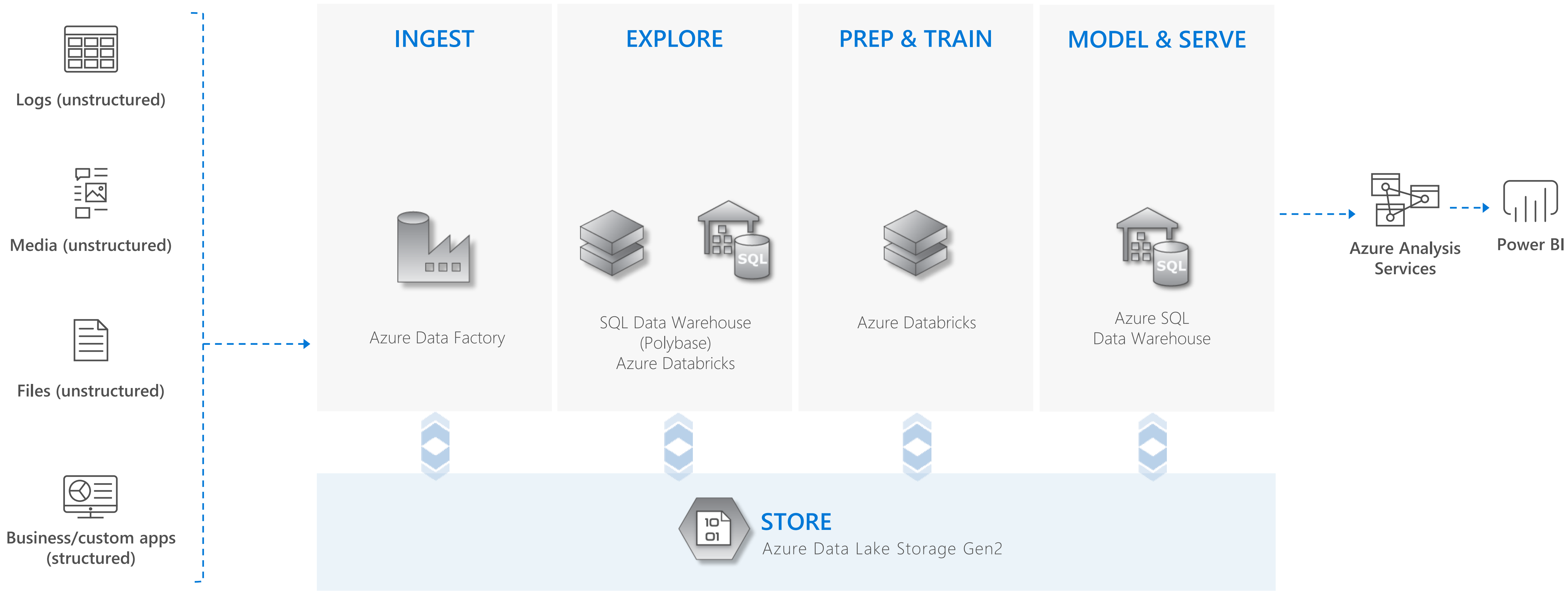
Support for data analysis via queries (random reads). Typically stored in denormalized form suitable for reporting & analysis. Aggregations can be stored to reduce computations at runtime.

Lambda Architecture



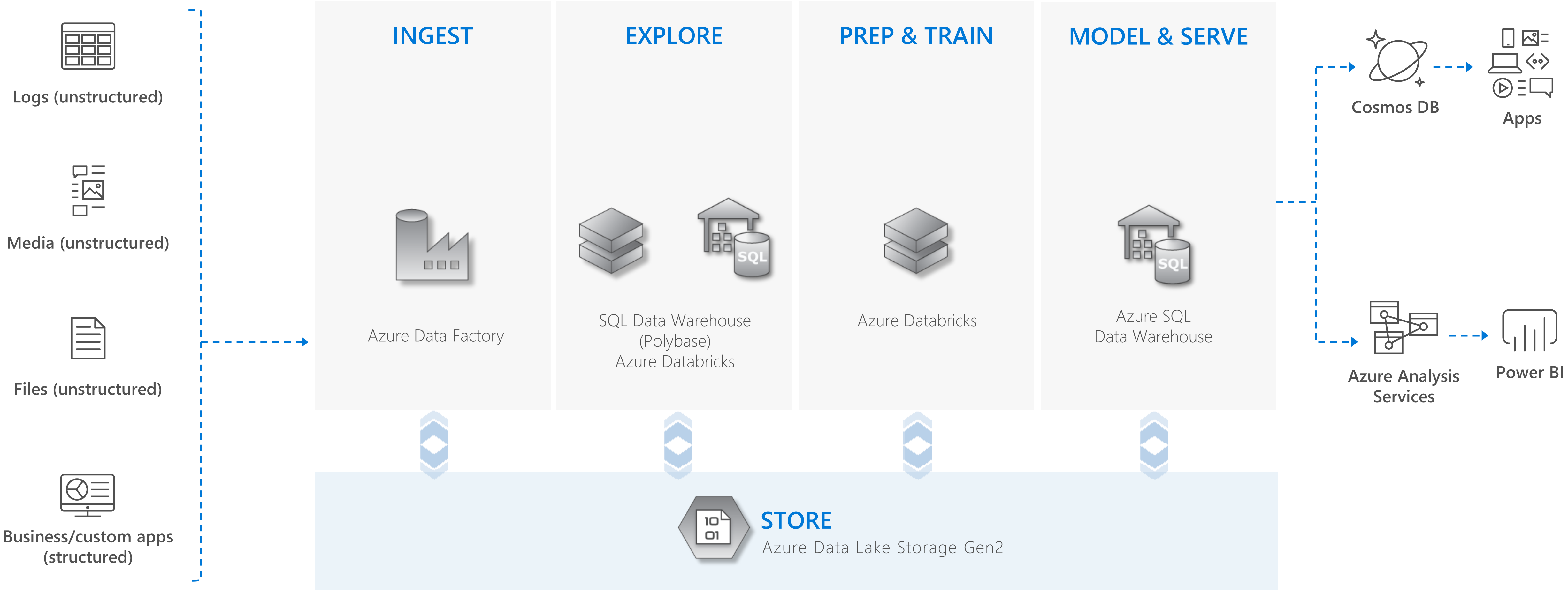
End to End Analytics

Modern Data Warehouse



End to End Analytics

Advanced Analytics



Logs (unstructured)

Media (unstructured)

Files (unstructured)

Business/custom apps (structured)

INGEST

Azure Data Factory

EXPLORE

SQL Data Warehouse (Polybase)
Azure Databricks

PREP & TRAIN

Azure Databricks

MODEL & SERVE

Azure SQL
Data Warehouse

STORE

Azure Data Lake Storage Gen2

Cosmos DB

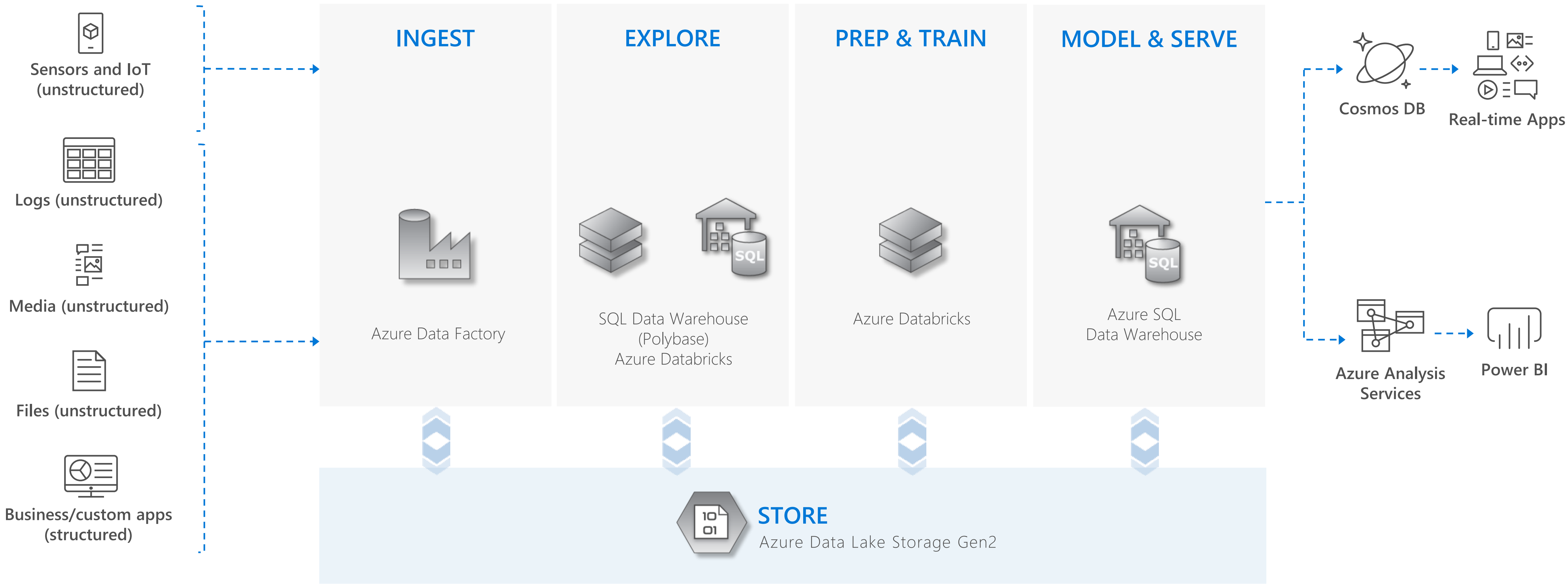
Apps

Azure Analysis Services

Power BI

End to End Analytics

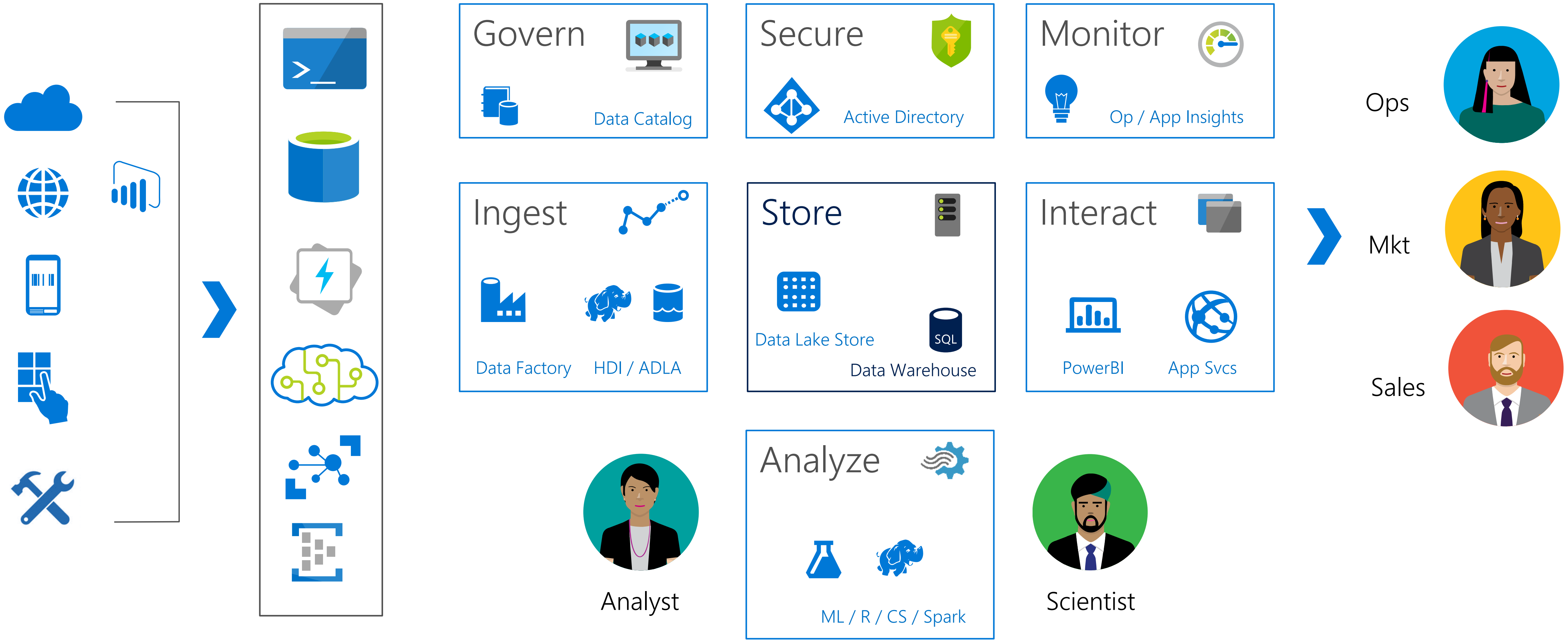
Realtime Analytics



Azure SQL DW

Where does a data warehouse fit? Everywhere!

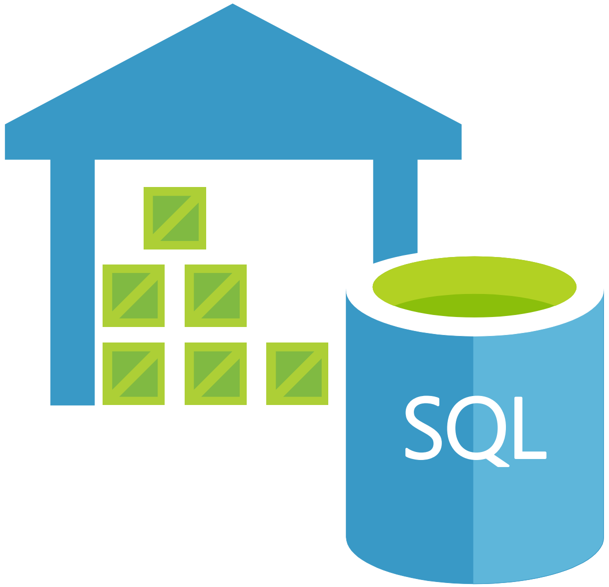
Data & service architecture



Changes in Enterprise Data Warehouse space

Organizations are changing with increasing demand to:

- Integrate with new or unstructured data
- Drive to the cloud
- Reduce or remove hardware renewal
- Reduction in support costs



Introducing Azure SQL Data Warehouse

A relational **platform-as-a-service**, fully managed by Microsoft.
Elastic scale cloud **data warehouse** with **proven** SQL Server capabilities.
Built for businesses of all **shapes, sizes,** and **industry**.

Elastic scale & performance

Scales to petabytes of data

Massively Parallel Processing

Instant-on compute scales in seconds

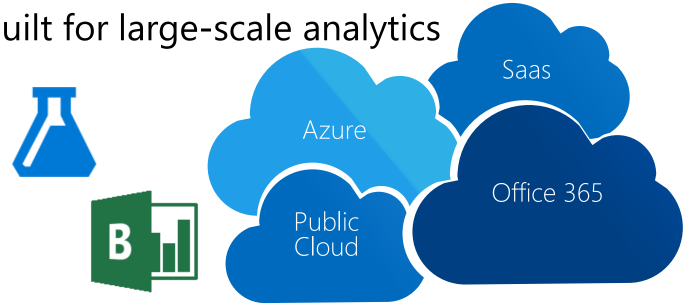
Query Relational and Non-Relational data

Relational batch processing

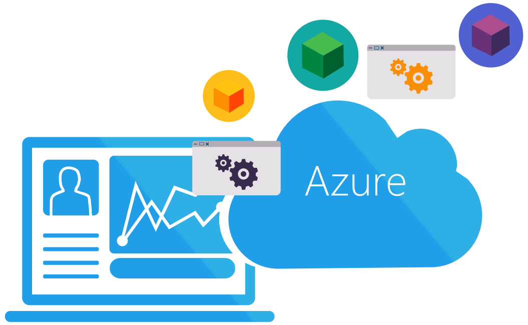
Query large datasets in minutes

Full hub-and-spoke support

Built for large-scale analytics



Market Leading Price & Performance

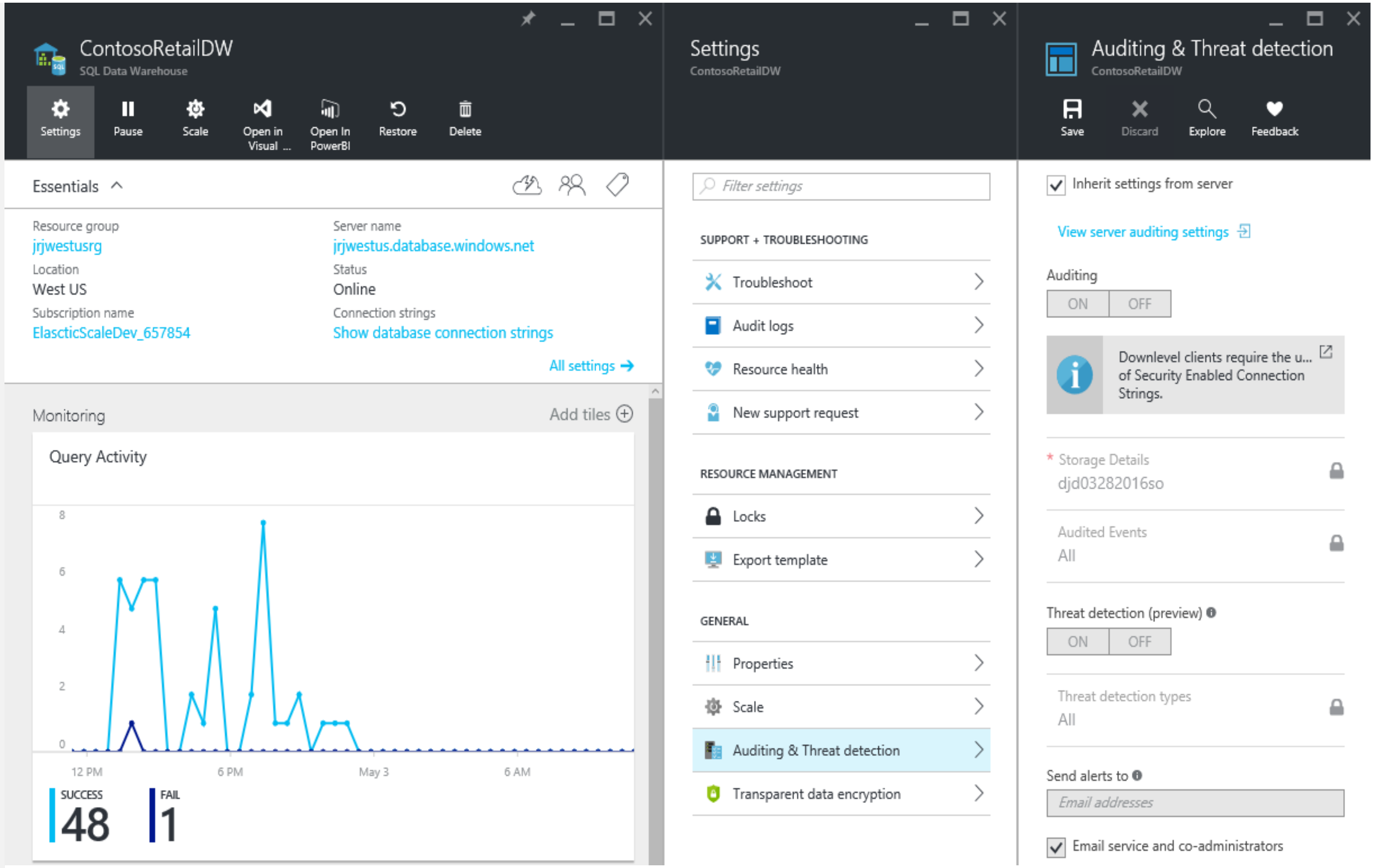


Simple billing compute & storage

Pay for what you need, when you need it with dynamic pause

A fully managed Platform-as-a-Service

- Azure cloud data warehouse service
- Elastic scale
- Separate storage and compute
- Use existing tools and skills
- Deploy and use in minutes!



ContosoRetailDW
SQL Data Warehouse

Resource group
jrjwestusrg

Location
West US

Subscription name
ElascticScaleDev_657854

Server name
jrjwestus.database.windows.net

Status
Online

Connection strings
Show database connection strings

SUCCESS 48

FAIL 1

Settings
ContosoRetailDW

Auditing & Threat detection
ContosoRetailDW

☑ Inherit settings from server

Auditing
ON | OFF

Downlevel clients require the u... of Security Enabled Connection Strings.

* Storage Details
djd03282016so

Audited Events
All

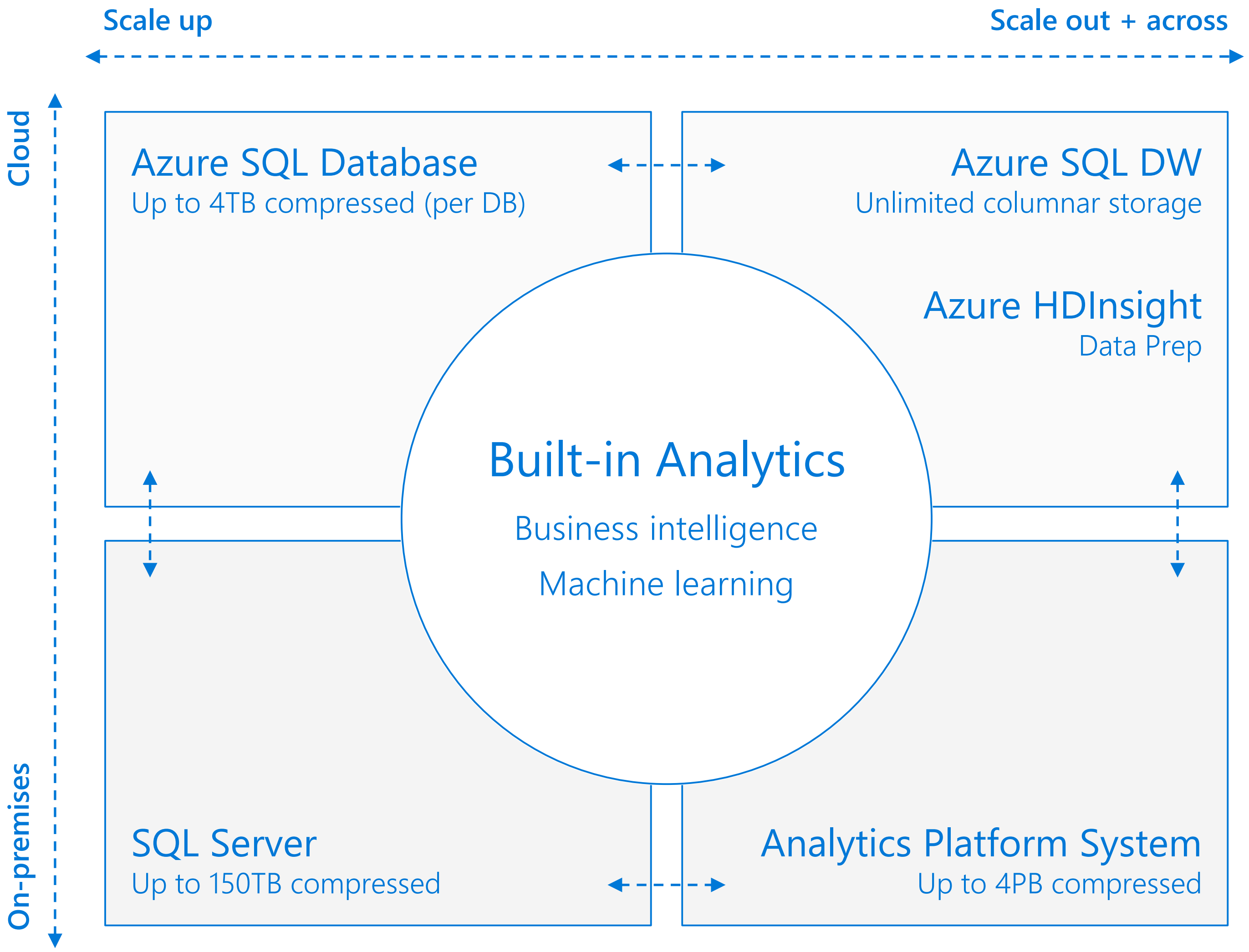
Threat detection (preview)
ON | OFF

Threat detection types
All

Send alerts to
Email addresses

☑ Email service and co-administrators

WHEN TO USE WHAT



Scale up

Scale out + across

Cloud

On-premises

Azure SQL Database
Up to 4TB compressed (per DB)

Azure SQL DW
Unlimited columnar storage

Azure HDInsight
Data Prep

Built-in Analytics

Business intelligence

Machine learning

SQL Server
Up to 150TB compressed

Analytics Platform System
Up to 4PB compressed

Technical capabilities

Industry's **first** enterprise-class cloud data warehouse that can **grow, shrink, and pause** in seconds

Petabyte scale data warehousing leveraging massive parallel processing

Full enterprise-class SQL Server experience

Two performance tiers designed for businesses of all sizes

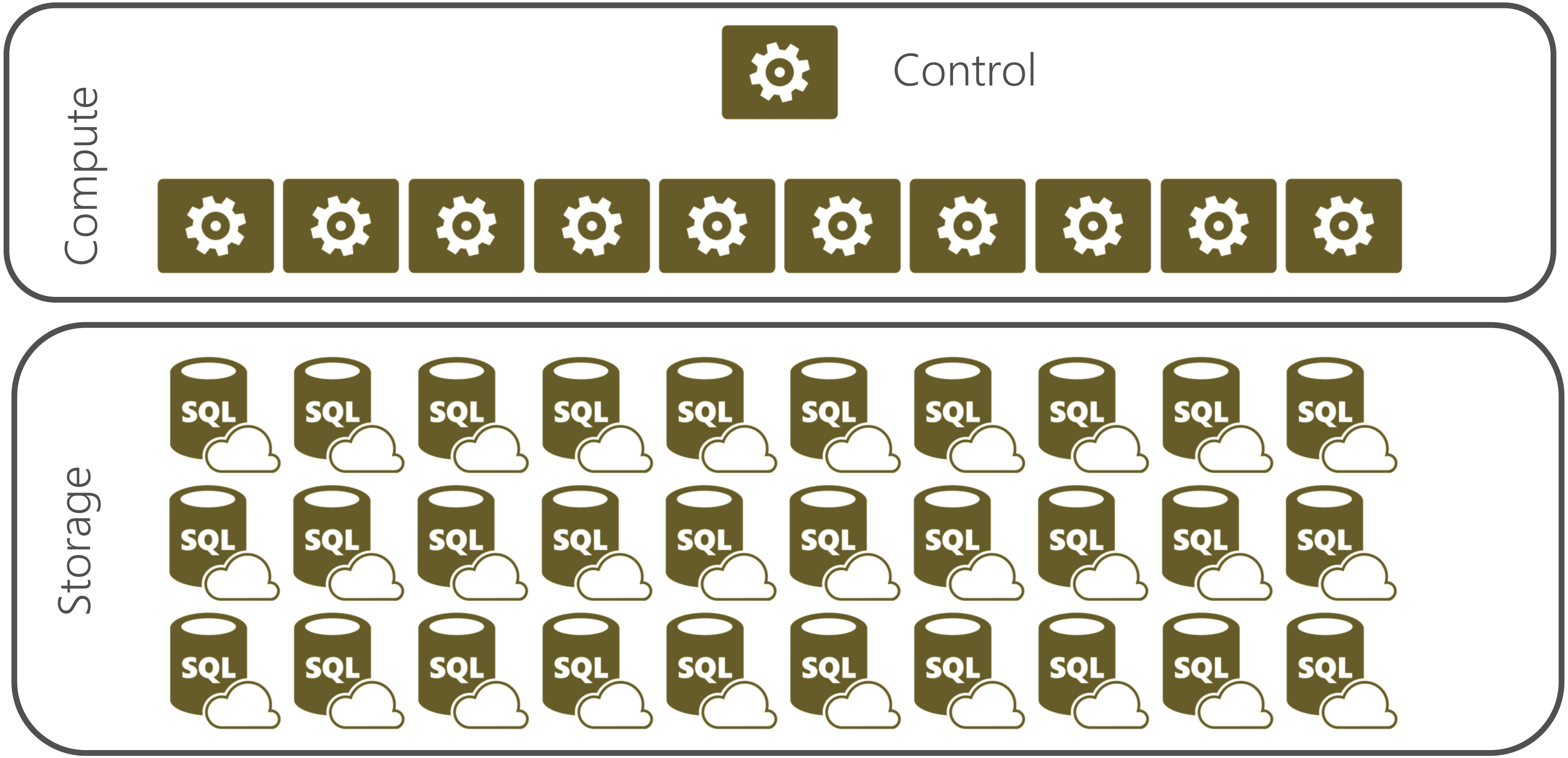
Seamless compatibility with Power BI, Azure Machine Learning, HDInsight, and Azure Data Factory

Query and load big data from Hadoop, HDInsight, Data Lake and Blob Storage using Polybase

SQL DW is good for analytical workloads. Why?

- ✓ Store large volumes of data.
- ✓ Consolidate disparate data into a single location.
- ✓ Shape, model, transform and aggregate data.
- ✓ Perform query analysis across large datasets.
- ✓ Ad-hoc reporting across large data volumes.
- ✓ All using simple SQL constructs.

Logical overview



Azure Data Factory

AZURE DATA FACTORY

A fully-managed data integration service in the cloud

PRODUCTIVE

- ✓ Drag & Drop UI
- ✓ Codeless Data Movement

HYBRID

- ✓ Orchestrate where your data lives
- ✓ Lift SSIS packages to Azure

SCALABLE

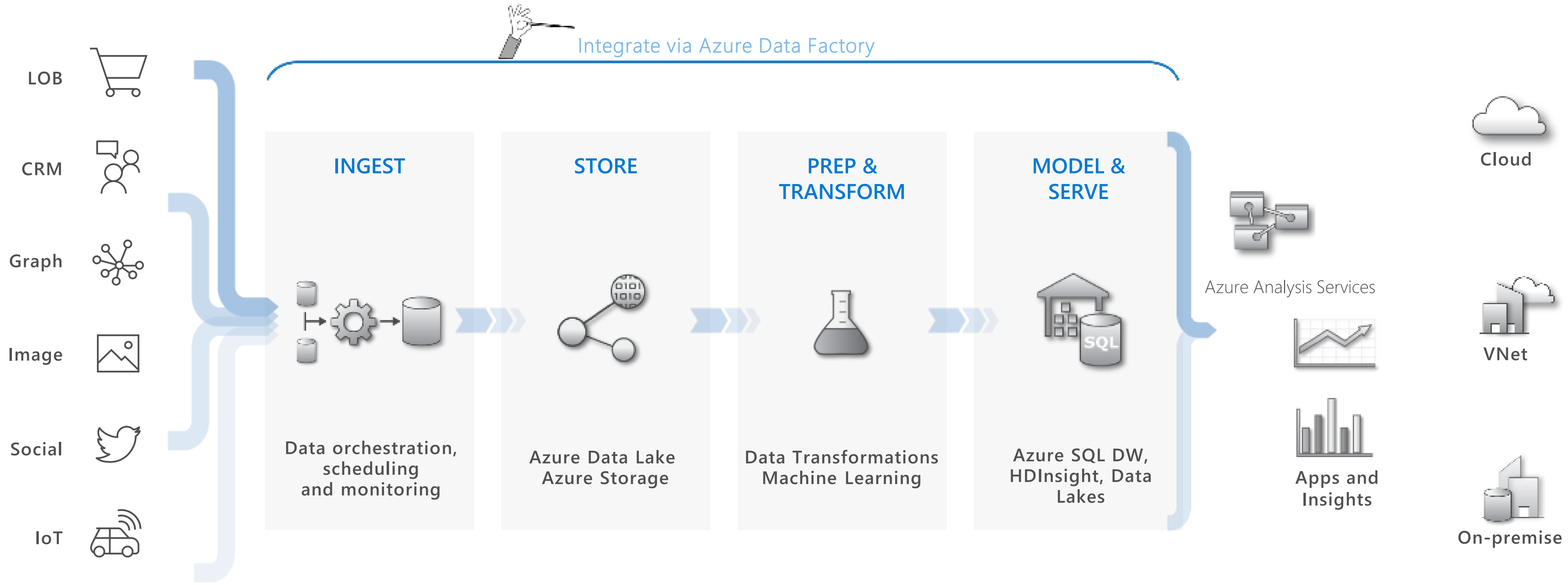
- ✓ Serverless scalability with no infrastructure to manage

TRUSTED

- ✓ Certified compliant Data Movement

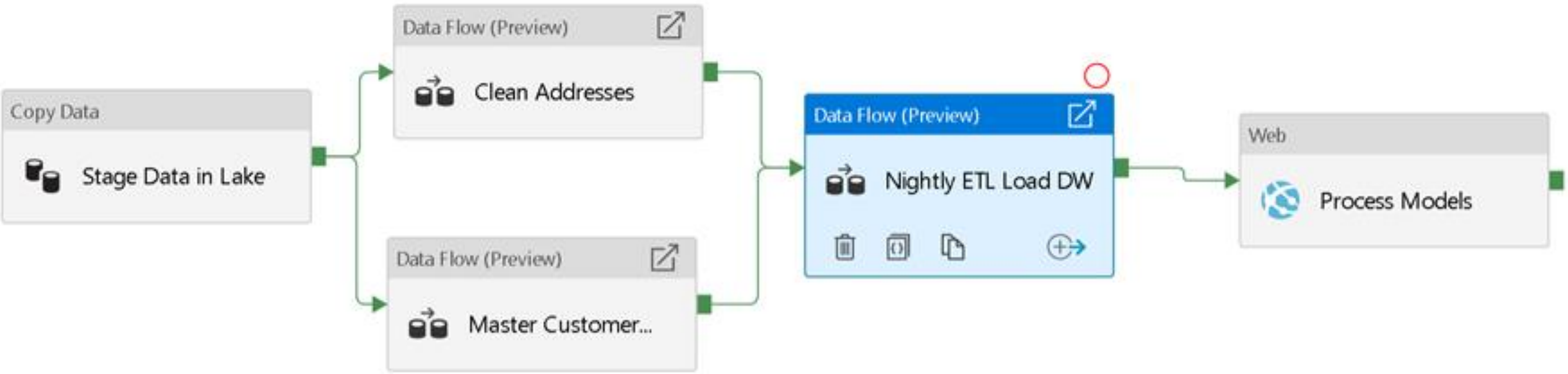
AZURE DATA FACTORY

Modernize your enterprise data warehouse at scale



AZURE DATA FACTORY

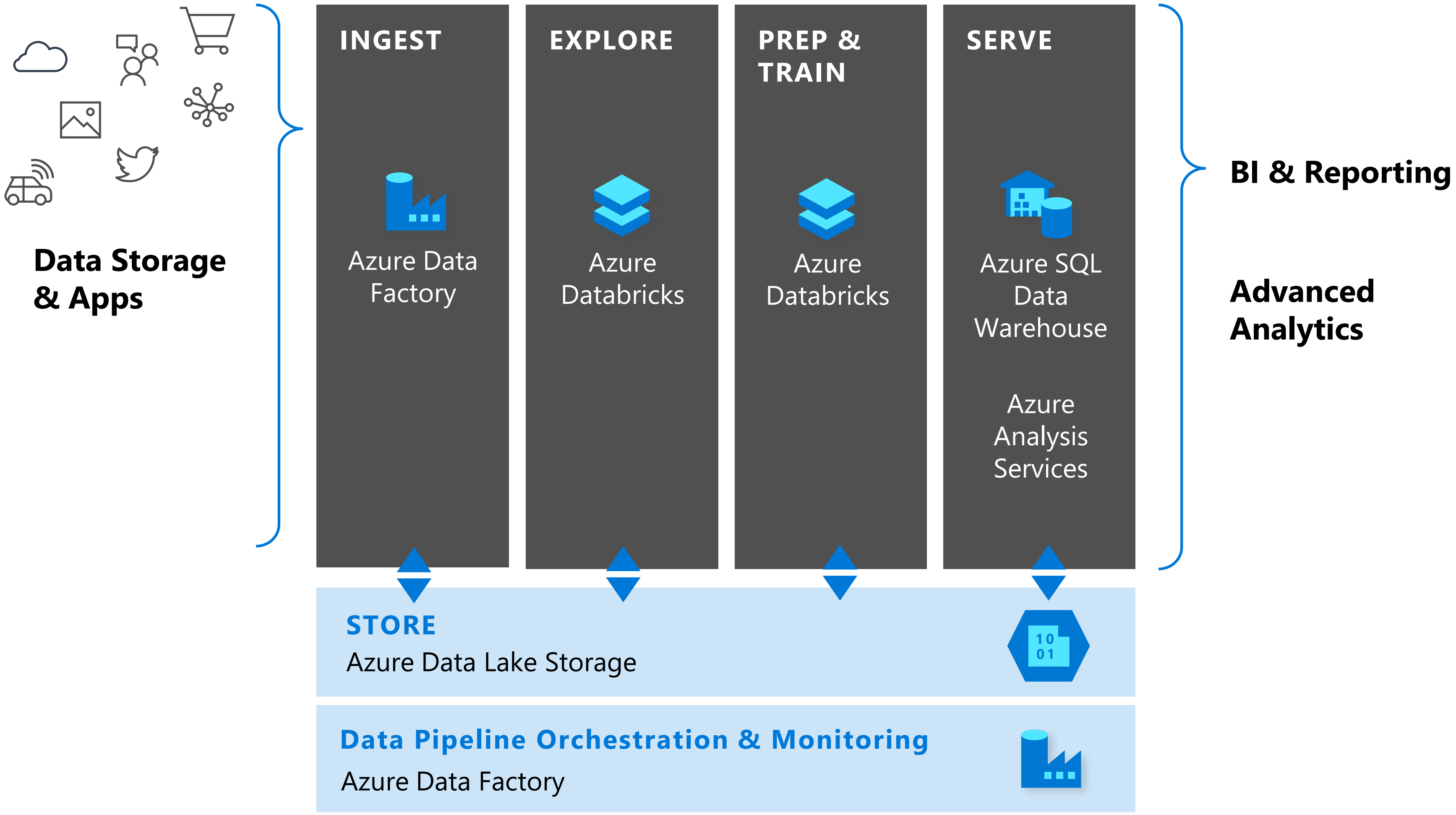
Visual Data Transformation with Mapping Data Flow (in preview)



- ✓ Zero-code experience for data transformation
- ✓ Visually design, build, and manage transformation processes
- ✓ No understanding of Spark or distributed architecture needed
- ✓ Visual drag and drop interface

Sign up for the preview of Mapping Data Flow >>>

Modern Data Warehouse (MDW)



Data Storage & Apps

INGEST

Azure Data Factory

EXPLORE

Azure Databricks

PREP & TRAIN

Azure Databricks

SERVE

Azure SQL Data Warehouse

Azure Analysis Services

BI & Reporting

Advanced Analytics

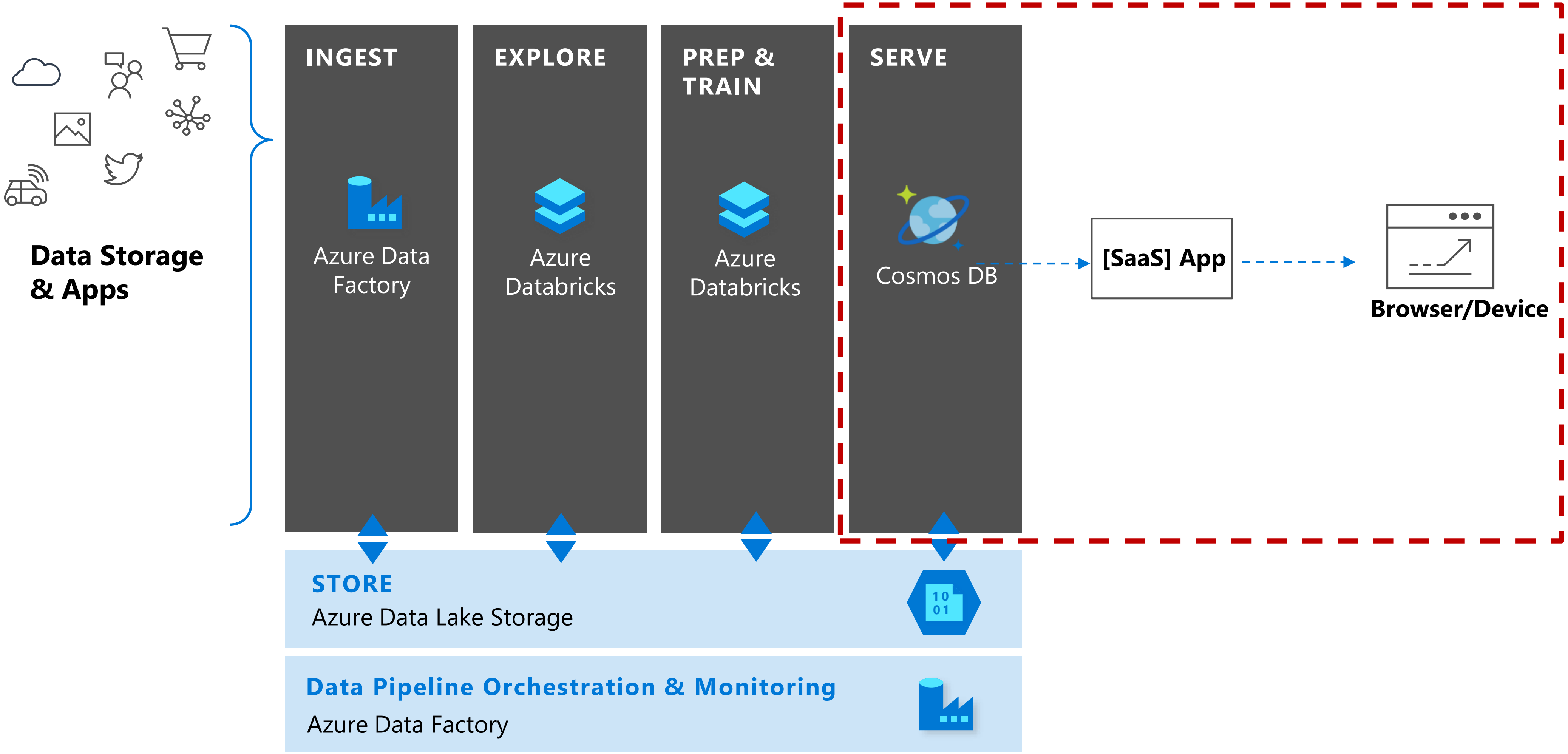
STORE

Azure Data Lake Storage

Data Pipeline Orchestration & Monitoring

Azure Data Factory

Analytics for data-driven apps



Cosmos DB

Azure Cosmos DB

A globally distributed, massively scalable, multi-model database service

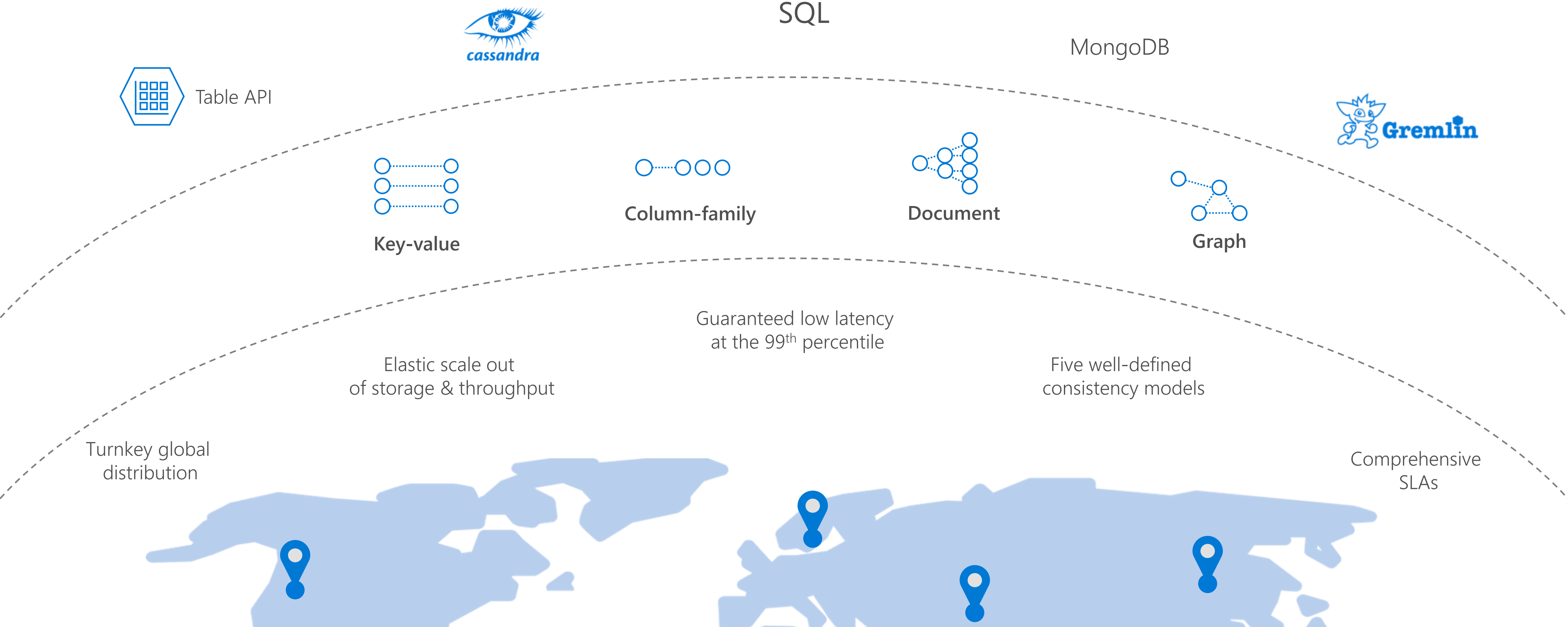


Table API

cassandra

SQL

MongoDB

Gremlin

Key-value

Column-family

Document

Graph

Guaranteed low latency at the 99th percentile

Elastic scale out of storage & throughput

Five well-defined consistency models

Turnkey global distribution

Comprehensive SLAs

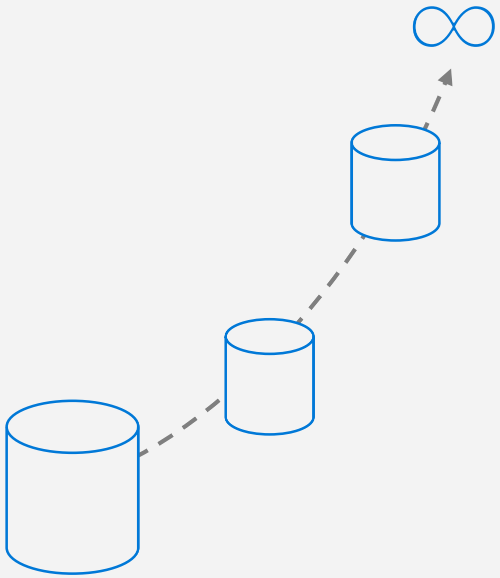
Turnkey Global Distribution

High Availability

- Automatic and Manual Failover
- Multi-homing API removes need for app redeployment

Low Latency (anywhere in the world)

- Packets cannot move fast than the speed of light
- Sending a packet across the world under ideal network conditions takes 100’s of milliseconds
- You can cheat the speed of light – using data locality
    - CDN’s solved this for static content
    - Azure Cosmos DB solves this for dynamic content



FIVE WELL-DEFINED CONSISTENCY MODELS

CHOOSE THE BEST CONSISTENCY MODEL FOR YOUR APP

Five well-defined, consistency models

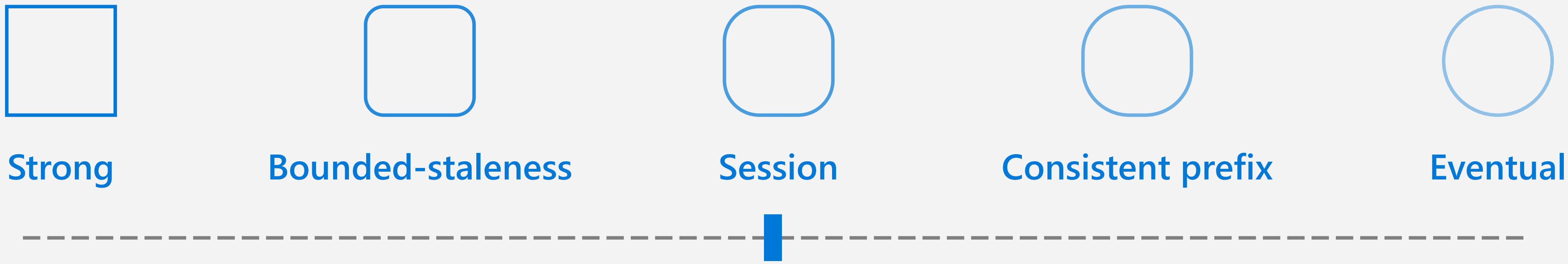
Overridable on a per-request basis

Provides control over performance-consistency tradeoffs, backed by comprehensive SLAs.

An intuitive programming model offering low latency and high availability for your planet-scale app.

CLEAR TRADEOFFS

- Latency
- Availability
- Throughput



Strong

Bounded-staleness

Session

Consistent prefix

Eventual

ELASTIC SCALE OUT OF STORAGE AND THROUGHPUT

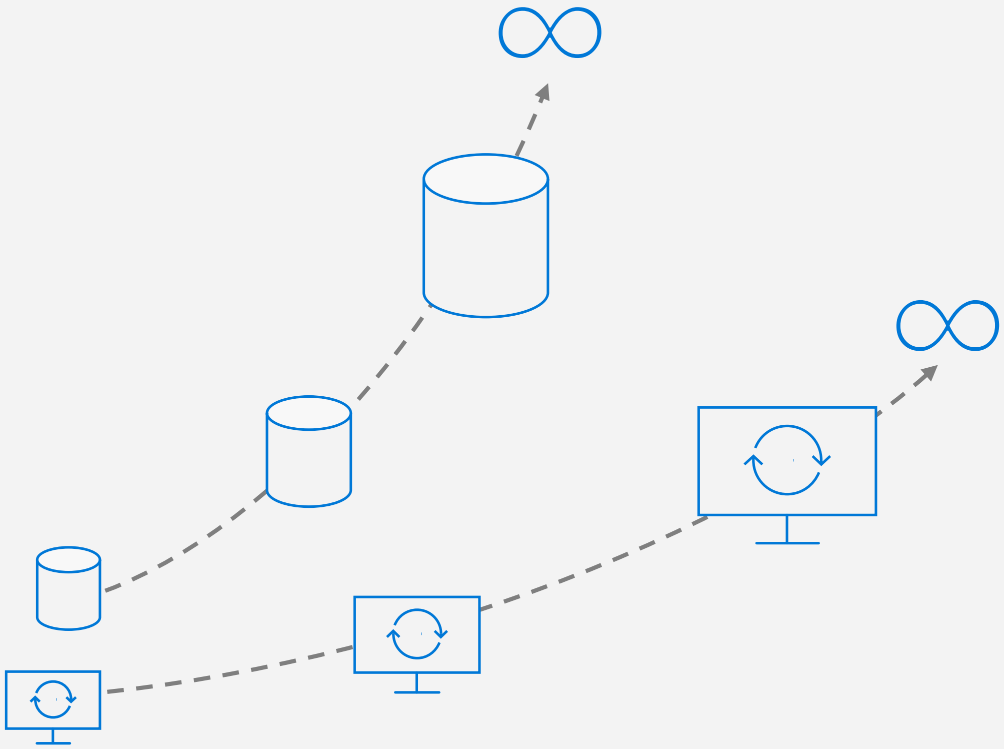
SCALES AS YOUR APPS’ NEEDS CHANGE

Database elastically scales storage and throughput

How? Scale-out!

Collections can span across large clusters of machines

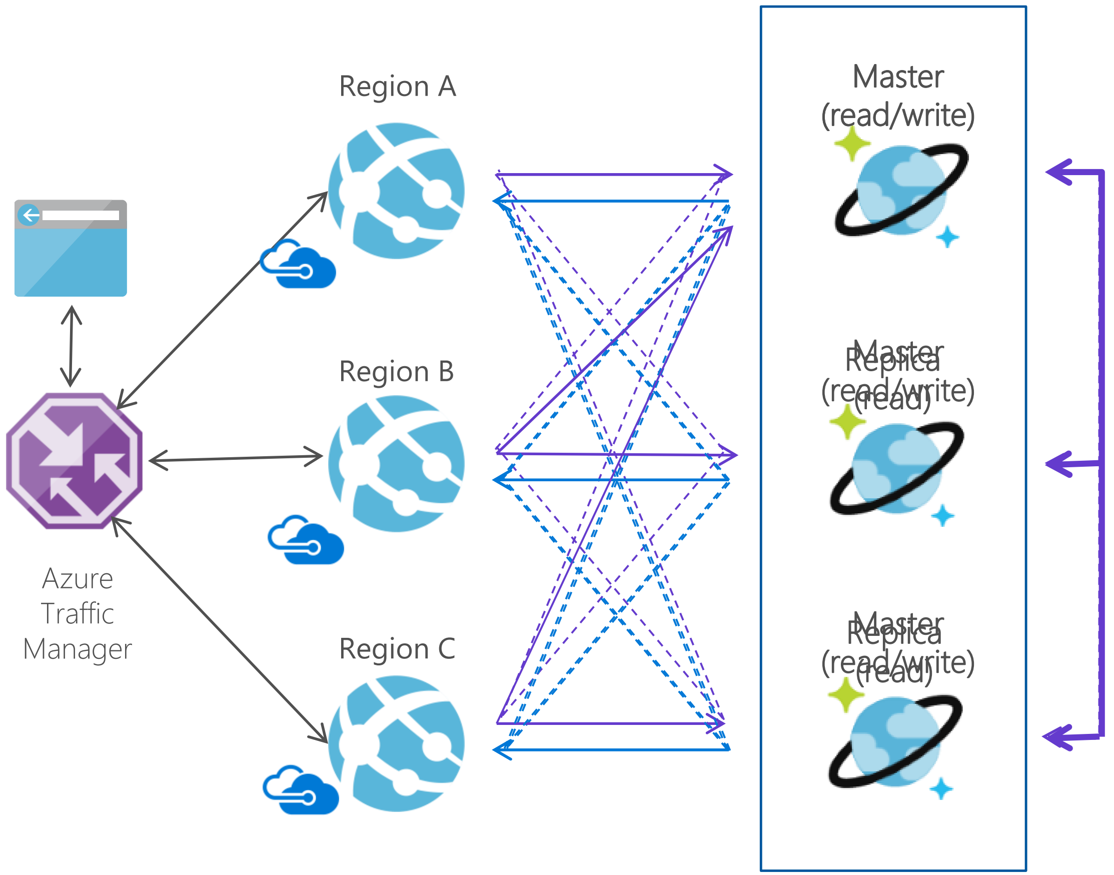
Can start small and seamlessly grow as your app grows



Azure Cosmos DB Multi Master

- Every region now writable
- Single-digit latency
- 99.999% availability
- Tunable consistency levels
- Flexible conflict resolution
- Unlimited endpoint scalability

- All Azure regions
- All data models
- All SDK's



Azure Data Catalog

Azure Data Catalog

Data ***source*** discovery

One stop shop for all enterprise data sources

No data movement, heavy up front investment

Time to value in minutes

Data from ***multiple*** sources

Structured and unstructured

On premises and in the cloud

Microsoft and non-Microsoft

Consumption through ***multiple*** tools

Enabling publishing, discovery and consumption of data sources through various tools

Powered by annotation ***crowdsourcing***

Empowering any user to capture and share their knowledge about registered sources

Responsibility Matrix

IT Admin

Govern

Apply policies and control access

*Analyze

Track and monitor usage

Publisher

Publish

Register Data Sources

Enrich

Categorize and Annotate

Consumer

Discover

Browse and search

Understand

Get context

Enrich

Categorize and Annotate

FEW ADOPTION PATTERNS

Bottom-Up Adoption

Pattern

- Data Catalog provisioned by an individual department
- Department users register, annotate, and consume
- Loosely assigned responsibilities
- Organic growth within the department and to neighboring departments

Advantages

- Immediate business-driven value
- Start small, evaluate, iterate, scale gradually

Disadvantages

- No centralized strategy
- No predictable growth
- Inconsistent patterns of usage

Top-Down Adoption

Pattern

- Data Catalog adopted as part of larger data initiative
- Population and ownership have well-defined responsibilities
- Usage of data catalog incorporated into standard processes

Advantages

- Centralized oversight and ownership
- Standardized processes and points of contact ease adoption and collaboration
- Easier to communicate and understand reach and ROI

Disadvantages

- Value to existing “legacy” processes may be delayed or deprioritized

MSFT PG Feedback:

We also see customers being successful with ADC when they are trying to make datasets discoverable and have a **dedicated set of identified people** who push and refresh data in the catalog (through the REST API or the registration tool).

***This is the opposite of the crowd sourced approach.**

- In this approach there is a dedicated set of people in the customer's company whose job involves pushing data to and keeping ADC up to date.

- In those cases ADC proves useful to a wide variety of folks at the customer because the data is meaningfully curated and up to date.

Big Data patterns

- Modern Data Warehouse
- Advance Analytics
- Real Time Analytics

Big Data

Aquiring data

Data processing

Data Insights

Modern Data Warehousing

The Modern Data Warehouse extends the scope of the data warehouse to serve “big data” that is prepared with techniques beyond relational ETL.

Modern Data Warehousing

“We want to integrate all our data including ‘big data” with our data warehouse”

Advanced Analytics

“We are trying to predict when our customers churn.”

Real-time Analytics

“We are trying to get insights from our devices in real-time, etc.”

Modern Data Warehousing

Modern Data Warehousing

CANONICAL OPERATIONS

Load & Ingest

TRANSFER, STORE

Process

PROCESS, CLEAN

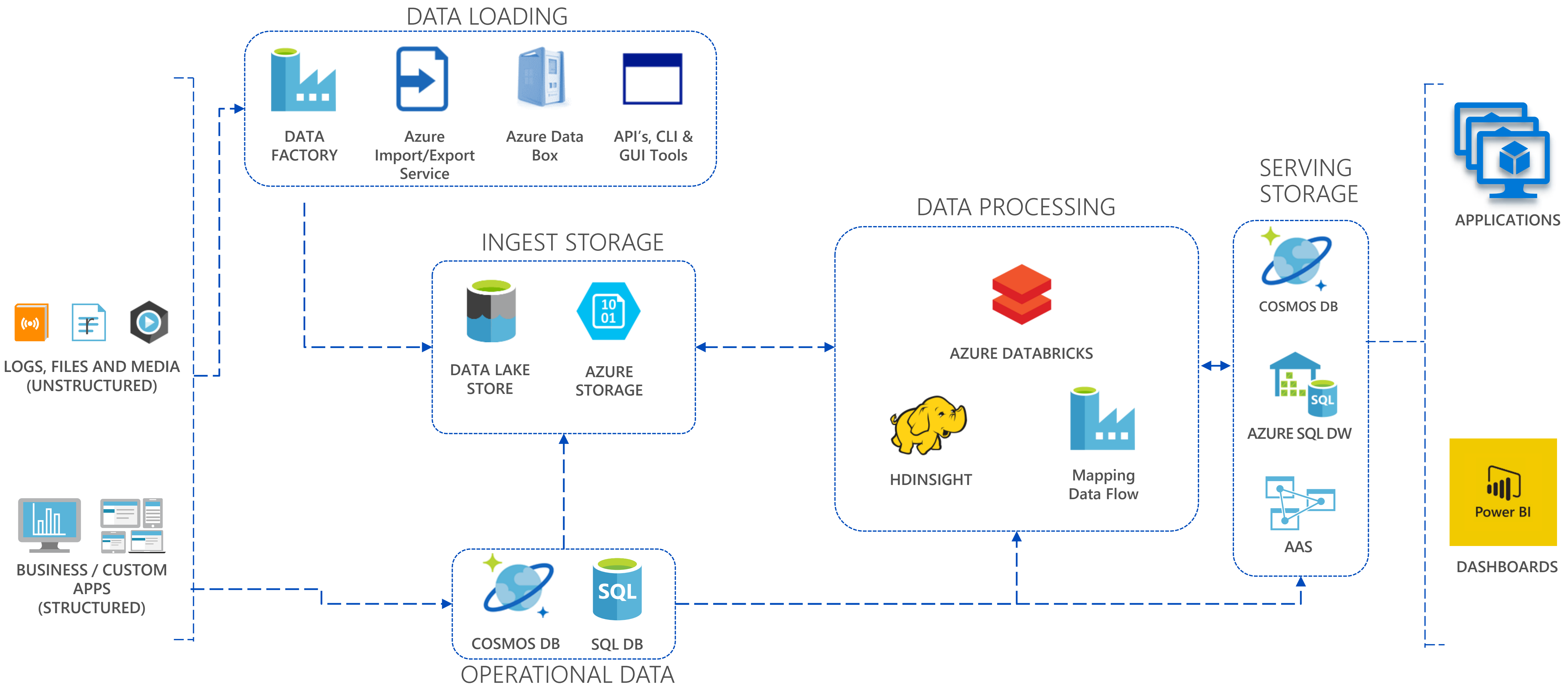
Serve

SERVE, ANALYZE

DATA WAREHOUSING PATTERN IN AZURE

Loading and preparing data for analysis with a data warehouse

Modern Data Warehousing



Advanced Analytics

Advanced Analytics is the process of applying machine learning and/or deep learning techniques to data for the purpose of creating predictive/prescriptive insights.

Modern Data Warehousing

“We want to integrate all our data including ‘big data” with our data warehouse”

Advanced Analytics

“We are trying to predict when our customers churn.”

Real-time Analytics

“We are trying to get insights from our devices in real-time, etc.”

Advanced Analytics

Advanced Analytics

CANONICAL OPERATIONS

Data Acquisition & Understanding

ACQUIRE, UNDERSTAND

Modeling

TRAINING, VALIDATION

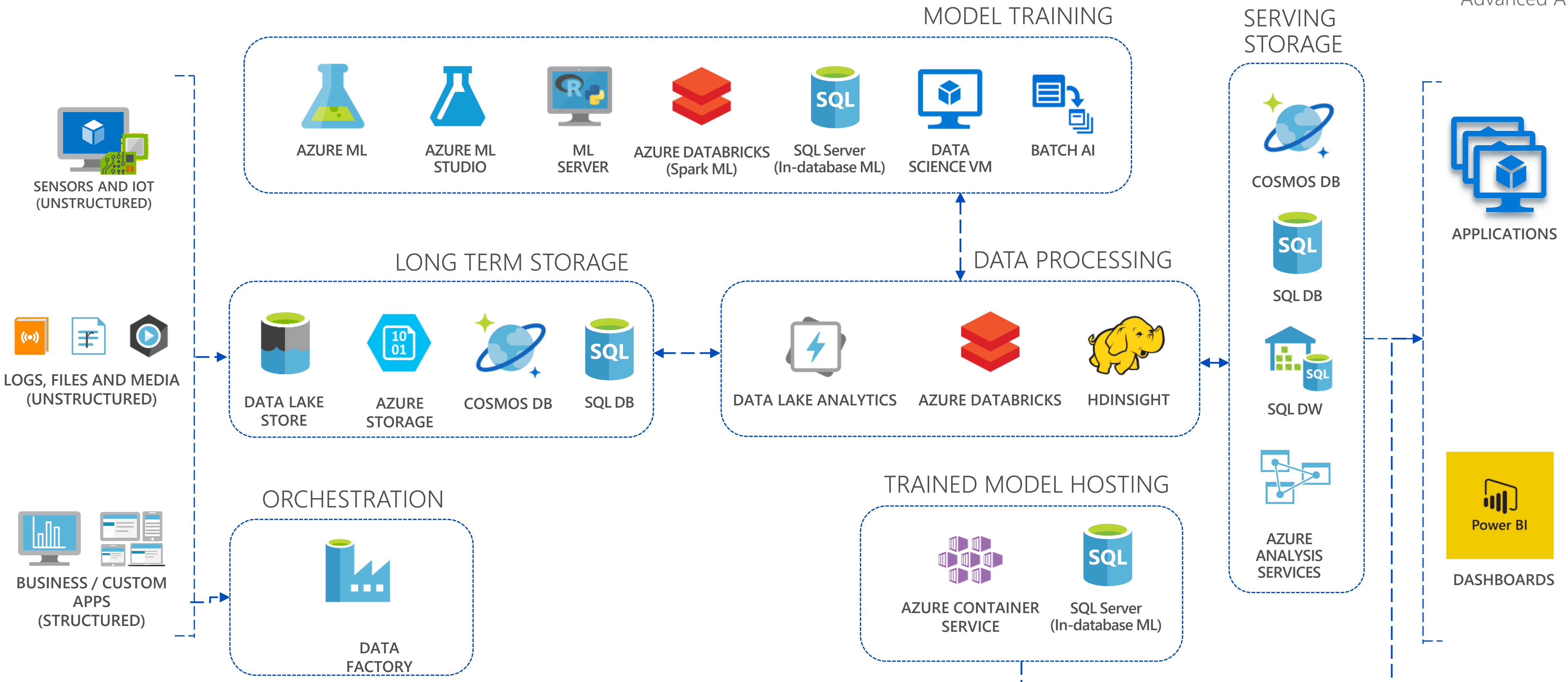
Deployment

DEPLOY, INTEGRATE

ADVANCED ANALYTICS PATTERN IN AZURE

Advanced Analytics

Performing data collection/understanding, modeling and deployment



Real-Time Analytics

Real-time Analytics (aka Stream Analytics) is the phenomenon of processing data as soon as it is generated, to derive very quick analysis/insight for timely action.

Modern Data Warehousing

“We want to integrate all our data including ‘big data” with our data warehouse”

Advanced Analytics

“We are trying to predict when our customers churn.”

Real-time Analytics

“We are trying to get insights from our devices in real-time, etc.”

Real-Time Analytics

Real-time Analytics

CANONICAL OPERATIONS

Ingest

CONNECT, COLLECT, STORE

Analytics

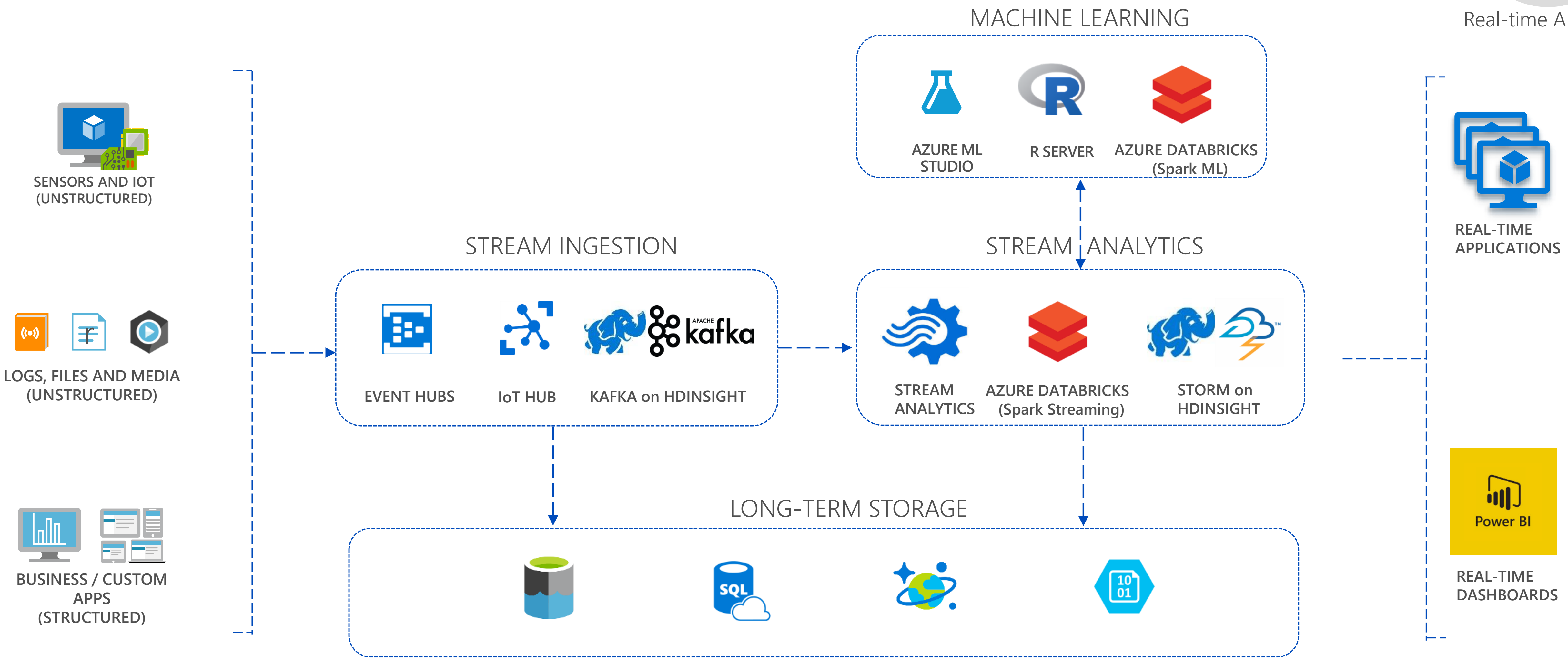
PROCESS, ANALYZE

Actions

REPORT, VISUALIZE, ACT

REAL-TIME ANALYTICS PATTERN WITH AZURE

Real-time Analytics



SENSORS AND IOT (UNSTRUCTURED)

LOGS, FILES AND MEDIA (UNSTRUCTURED)

BUSINESS / CUSTOM APPS (STRUCTURED)

STREAM INGESTION

EVENT HUBS

IoT HUB

KAFKA on HDINSIGHT

MACHINE LEARNING

AZURE ML STUDIO

R SERVER

AZURE DATABRICKS (Spark ML)

STREAM ANALYTICS

STREAM ANALYTICS

AZURE DATABRICKS (Spark Streaming)

STORM on HDINSIGHT

LONG-TERM STORAGE

REAL-TIME APPLICATIONS

REAL-TIME DASHBOARDS

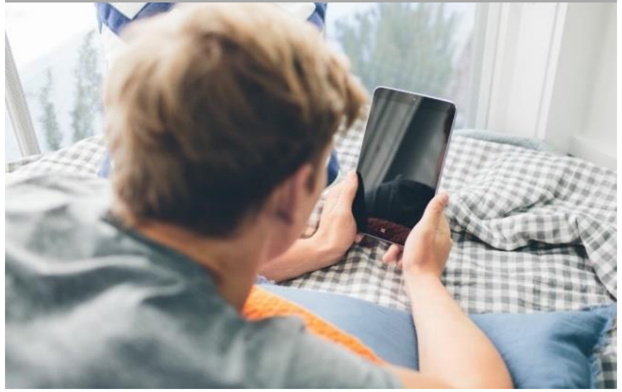
REAL-TIME ANALYTICS SCENARIOS

Real-time fraud detection

Fleet Management and Connected Cars

Click-stream analysis

Real-time Patient Monitoring



Smart grid

Customer Behavior in stores

IT Infrastructure and Networking monitoring

Real-time demand and Inventory Management