Intro to Data Lakes

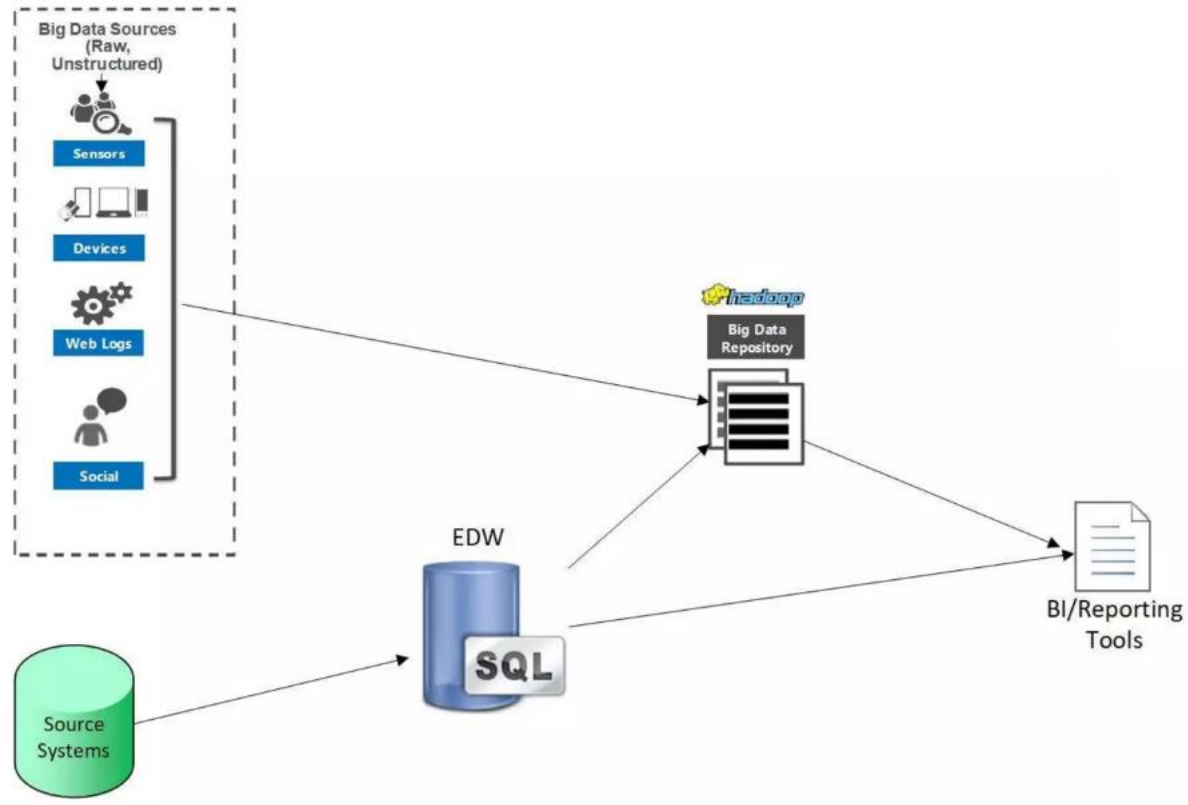
Data Lakehouse

Outline

- Big Data Architectures
- Why data lakes?
- The approach : Top-down vs Bottom-up
- What is Data Lake?
- Data Lake Use Cases

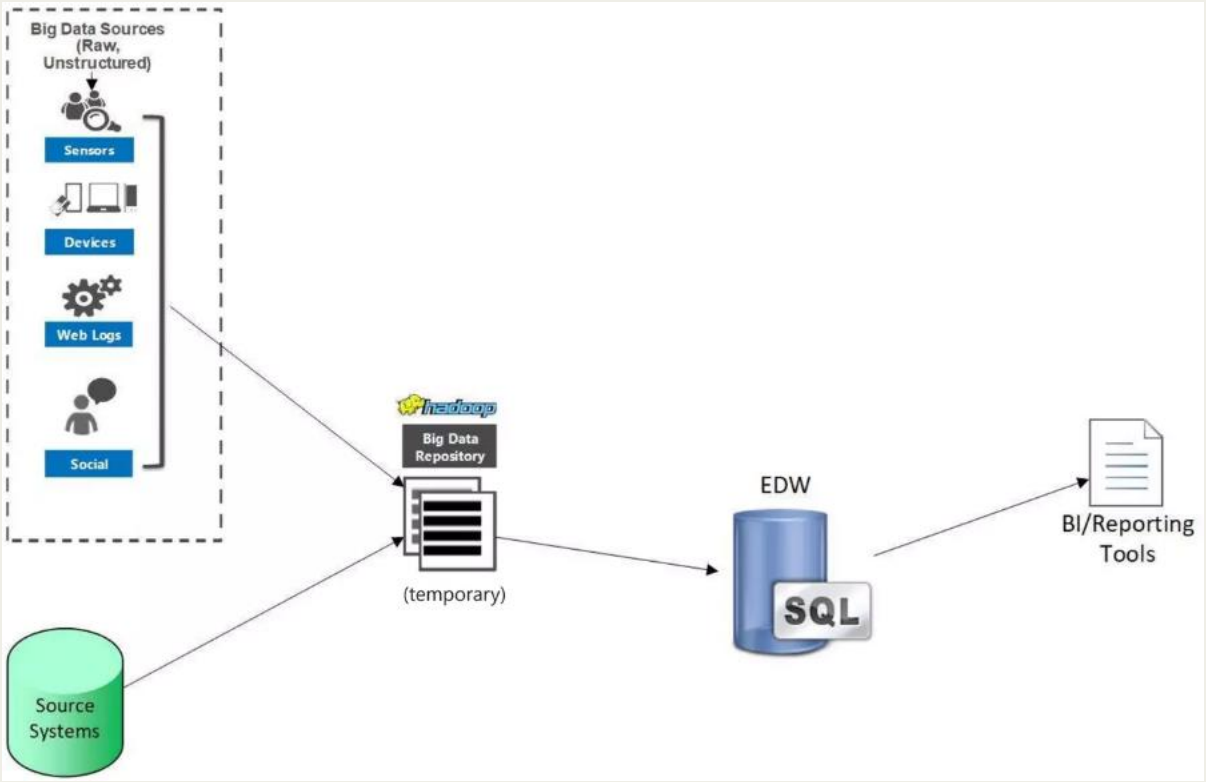
Enterprise data warehouse augmentation

- Seen when EDW has been in existence a while and EDW can handle new data
- Data hub, not data lake
- Cons: not offloading EDW Works, can’t use the existing tools, difficulty joining data in data hub with EDW



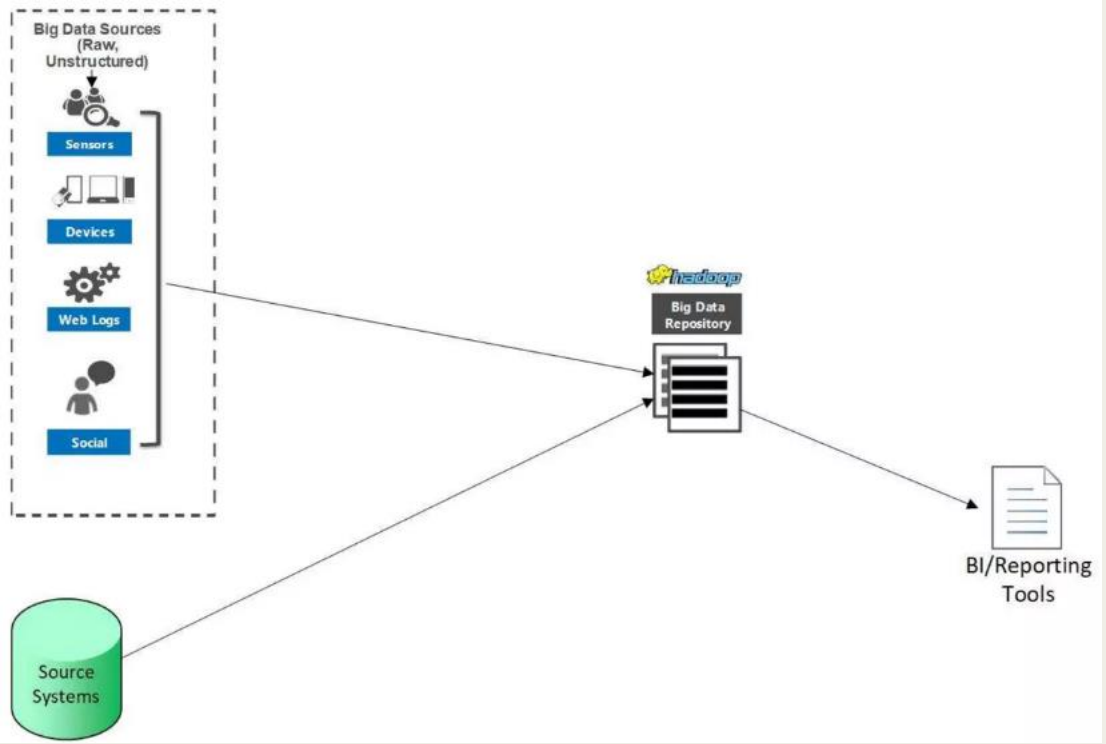
Data Hub plus DW

- Data hub is used as temporary staging and refining, no reporting
- Cons: data hub is temporary, no reporting/analysis done with the data hub



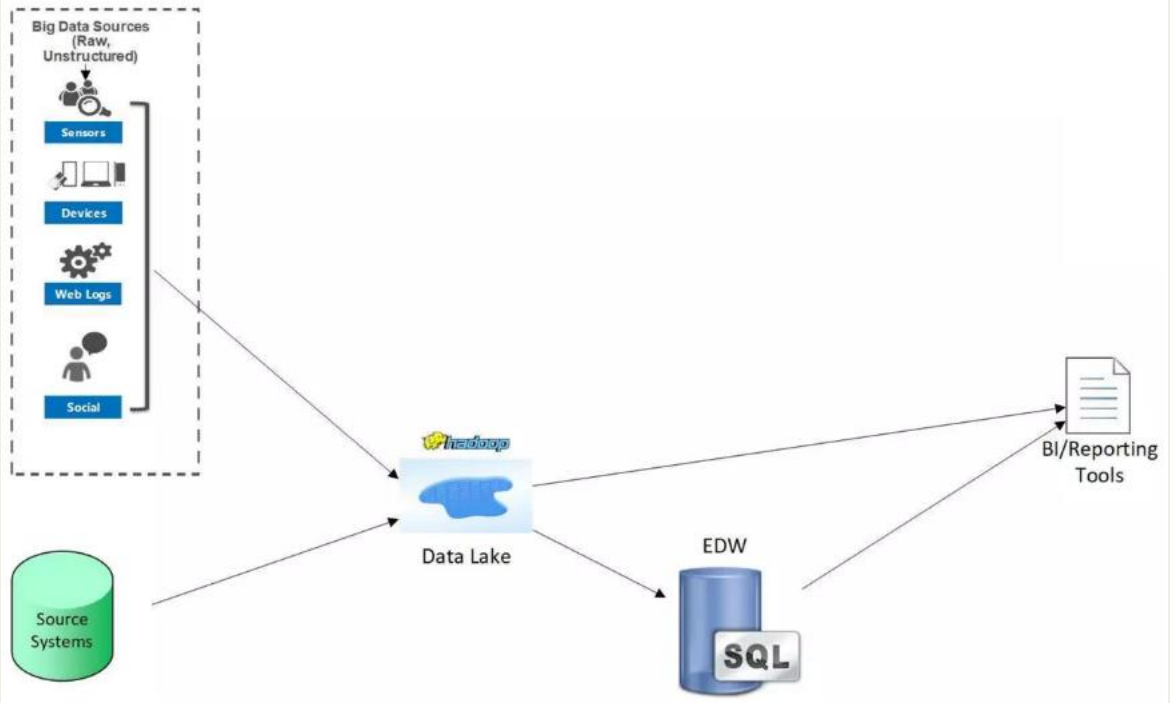
All in one

- Data hub is total solution, no EDW
- Cons: query are slower, new training for reporting tools, difficulty understanding data, security limitation

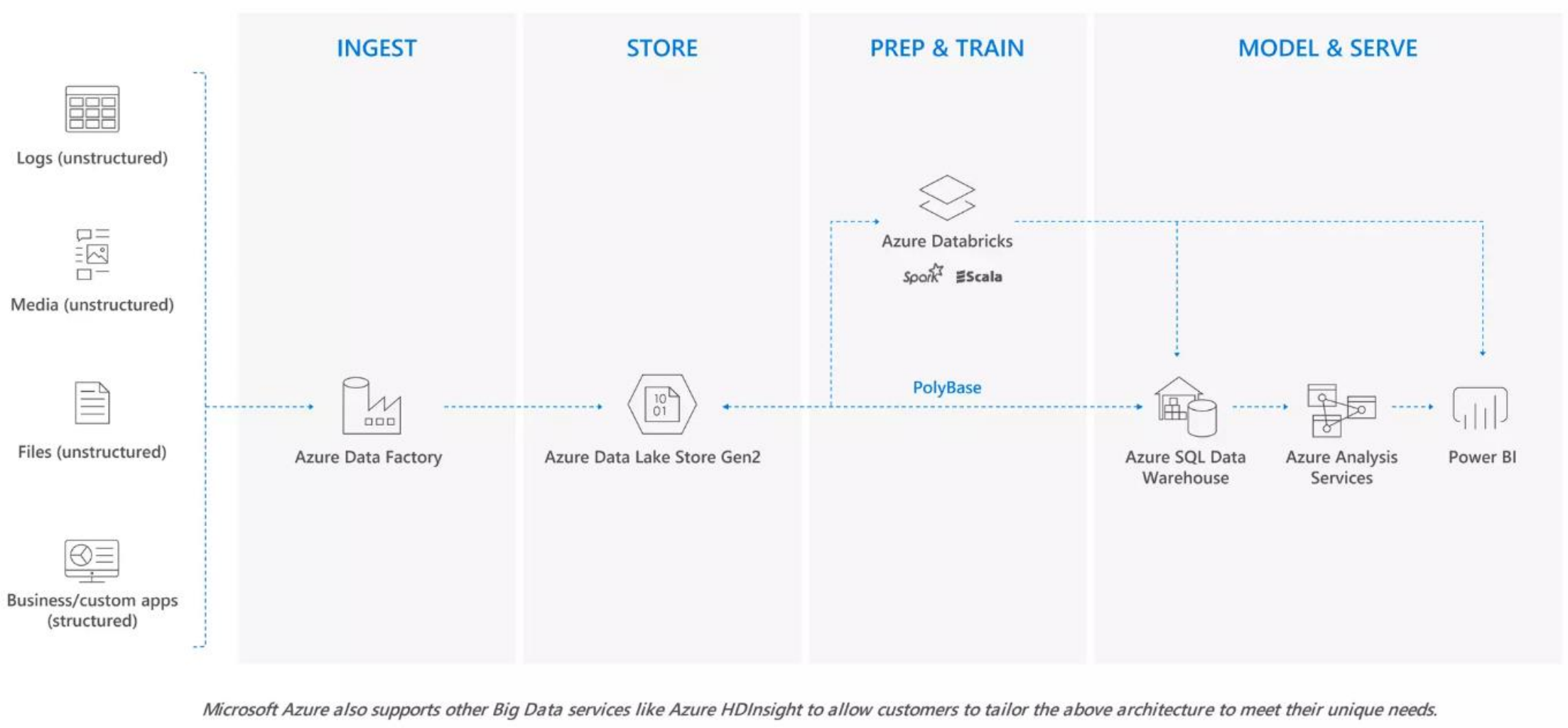


Modern Data Warehouse

- Evolution of three previous scenarios
- Ultimate goals
- Support future data needs
- Data harmonized and analyzed in the data lake or moved to EDW for more quality and performance



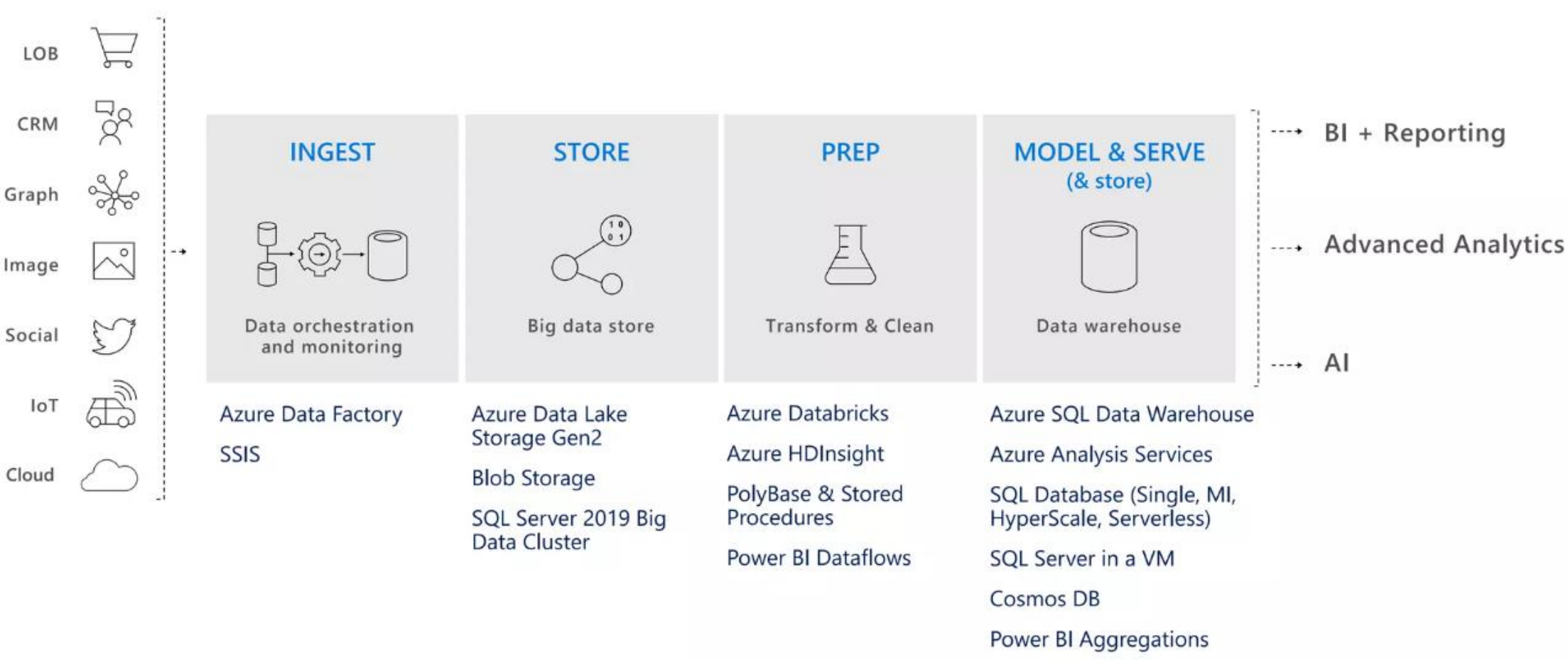
Modern Data Warehouse



Microsoft Azure also supports other Big Data services like Azure HDInsight to allow customers to tailor the above architecture to meet their unique needs.

Modern Data Warehouse

- Possible product by four areas

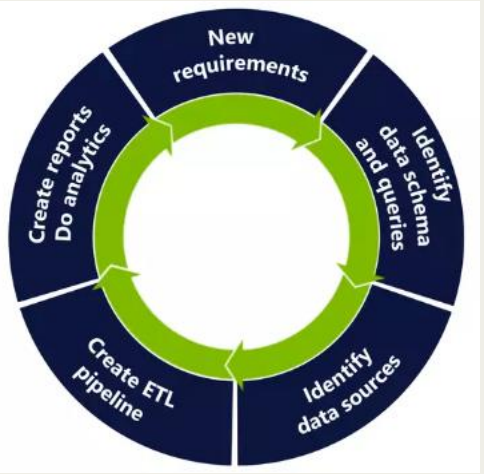


Each product may span its functionality

Why Data Lake?

Traditional Business Analytics Process

- Start with end-user requirements to identify desired reports and analysis
- Defines corresponding database schemas and queries
- Identify the required data sources
- Create ETL Pipeline
- Create reports. Analyze data

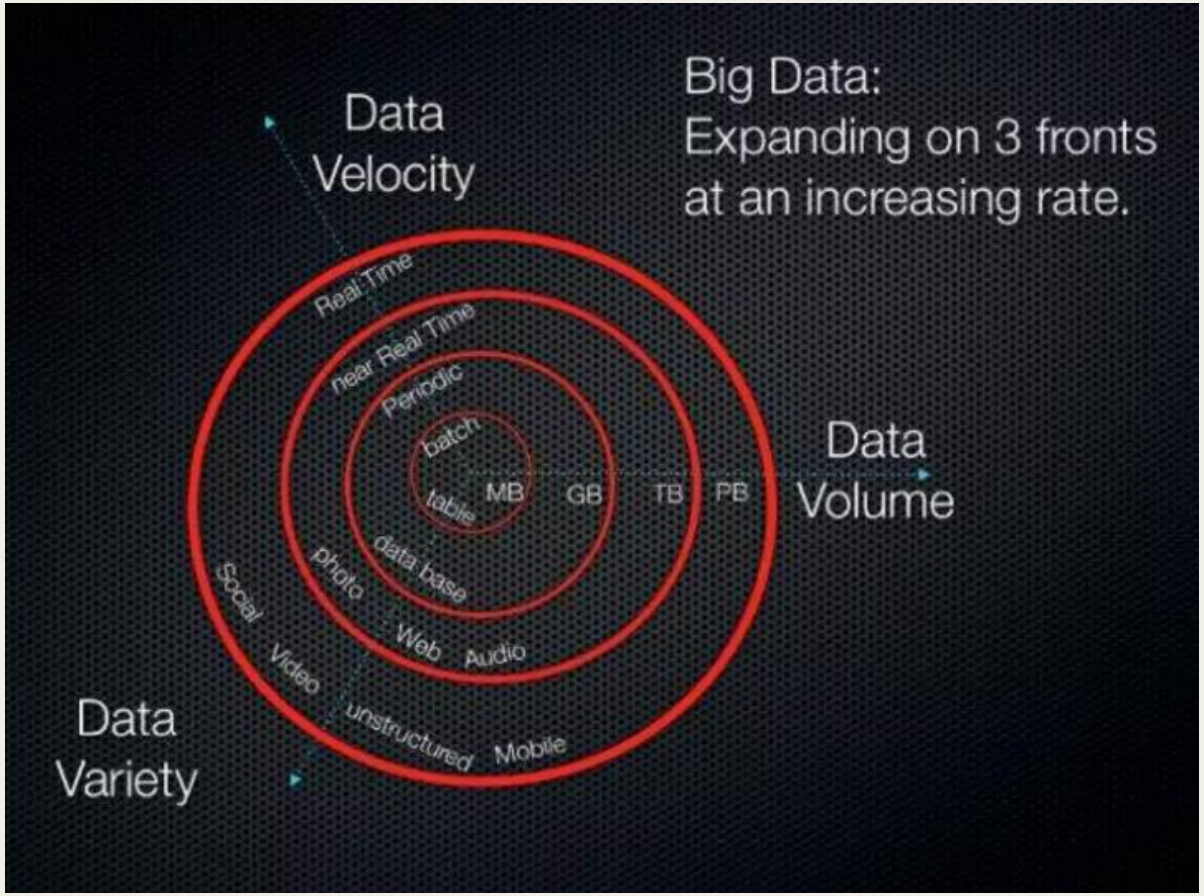


Need to collect any data

- Harness the growing and changing nature of data

- Challenge is combining transactional data stored in relational DB with less structured data
- Big data = All data
- Get the right information to the right people at the right time in the right format

The three V’s

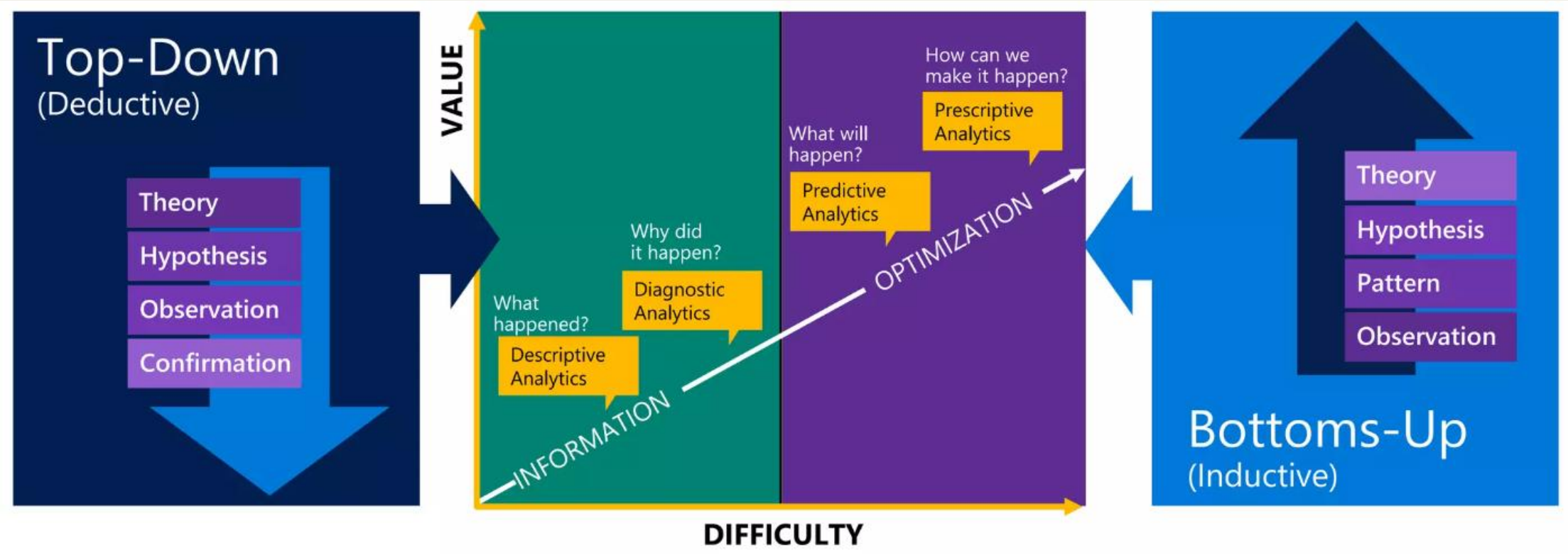

Big Data:
Expanding on 3 fronts
at an increasing rate.
Data
Velocity
Real Time
near Real Time
Periodic
batch
table
MB
GB
TB
PB
Data
Volume
data base
photo
Web
Audio
Social
Video
unstructured
Mobile
Data
Variety

New Big Data Thinking: All data has value

Use a data lake:

- All data has potential value
- Data hoarding
- No defined schema – stored in native format
- Schema is impose and transformations are done at query time (schema on read)
- Apps and users interpret the data as they see fit

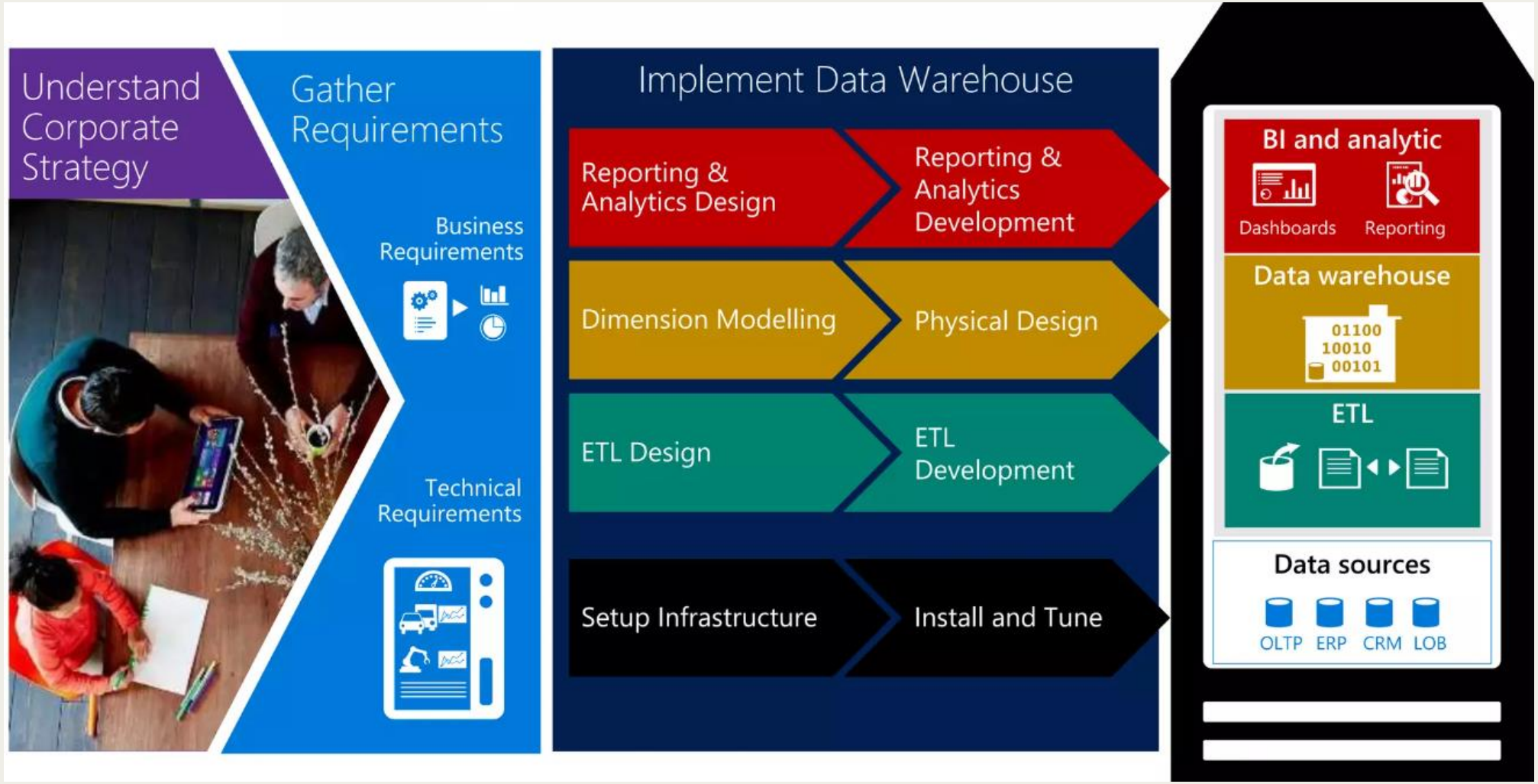
The approach : Top-down vs Bottom-up



- Know the question to ask
- Lots of upfront work to get the data to where you can use it
- Model first

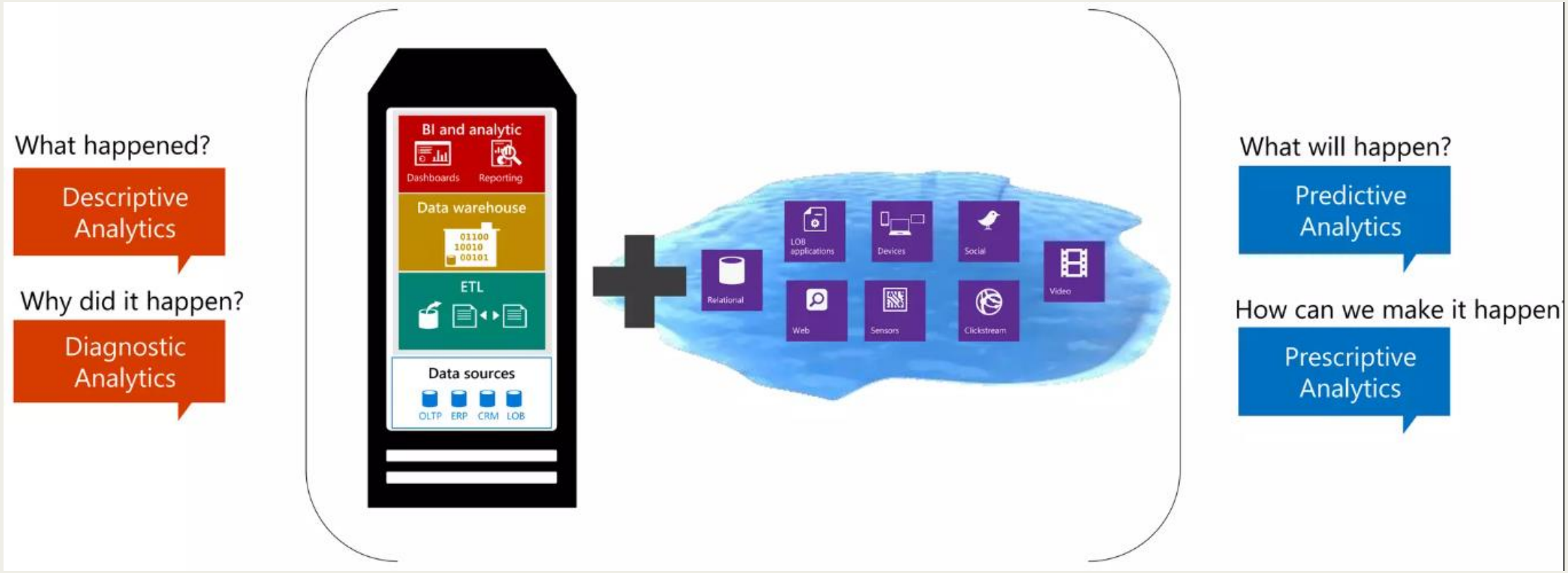
- Don’t know the question to ask
- Little upfront work need to be done to start using data
- Model later

Data Warehousing Uses a Top-Down Approach



Data Lake uses a Bottoms-Up Approach

Data Lake + Data Warehouse Better Together



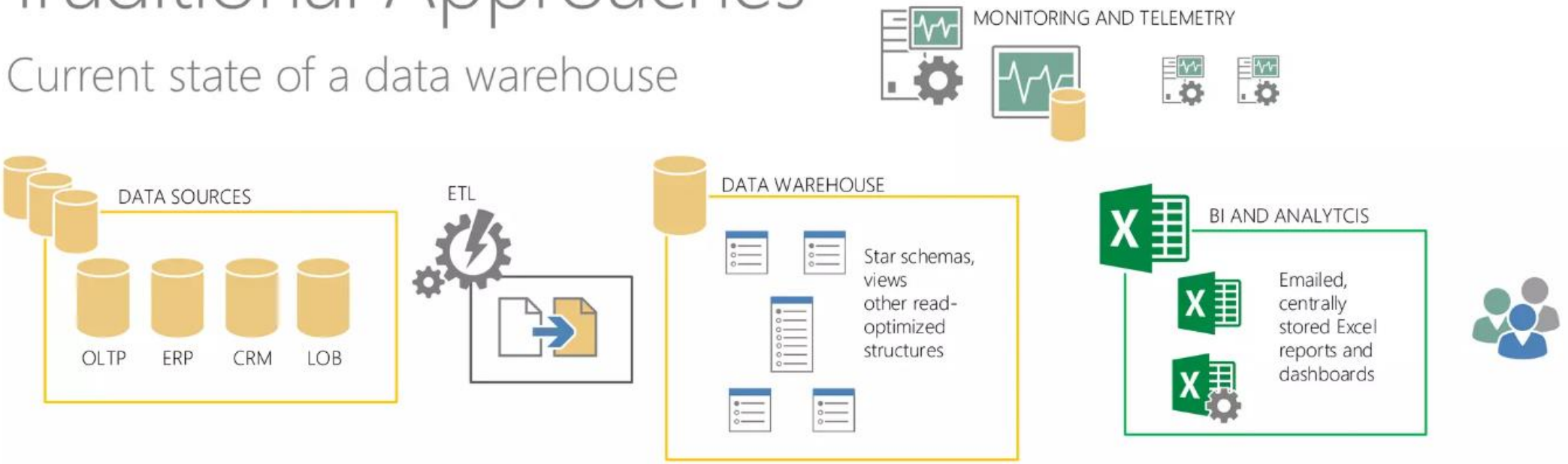
Exactly what is a data lake?

- A storage repository, that holds a vast amount of raw data in its native format until it is needed.

Inexpensively store **unlimited** data – **Centralized** place for **multiple object** (single version of the truth) – **Collect all** data "just in case" (data hoarding) – **Easy integration of differently**-structured data – Store data with **no modeling** (schema on read) – **Complements** enterprise data warehouse (EDW) – **Frees up expensive EDW resources** for queries instead of using EDW resources for transformation (avoid user contention) – **Quick user access** to data for power user/data scientist – Data exploration to **see if data valuable before** writing ETL and schema for relational DB, or use one time report – Place **to land IoT streaming data** – **Online archive or backup** for data warehouse data – **Keep raw data** so don't have to go back to source if need to re-run – Allow for **data to be used many times** for different analytics need an use cases – **Cost savings** and faster transformation **– Extreme performance** for transformations by having multiple compute options – The ability for an end user or product to easily **access the data from any location**

Traditional Approaches

Current state of a data warehouse



Well manicured, often relational sources

Known and expected data volume and formats

Little to no change

Complex, rigid transformations

Required extensive monitoring

Transformed historical into read structures

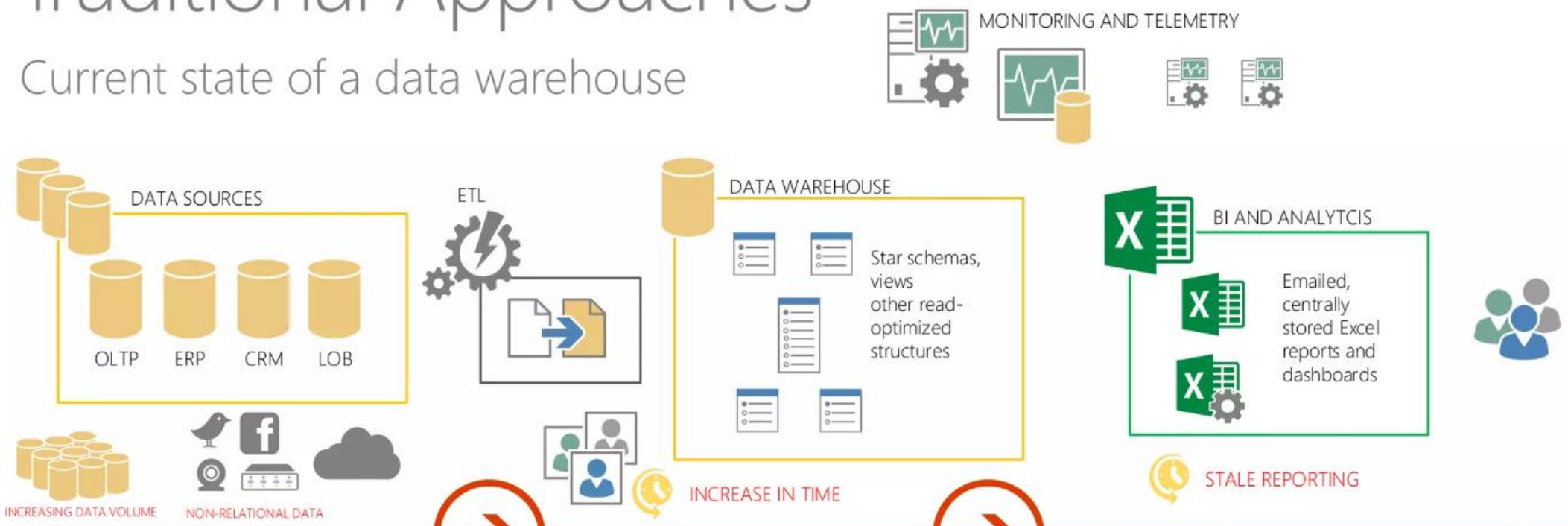
Flat, canned or multi-dimensional access to historical data

Many reports, multiple versions of the truth

24 to 48h delay

Traditional Approaches

Current state of a data warehouse



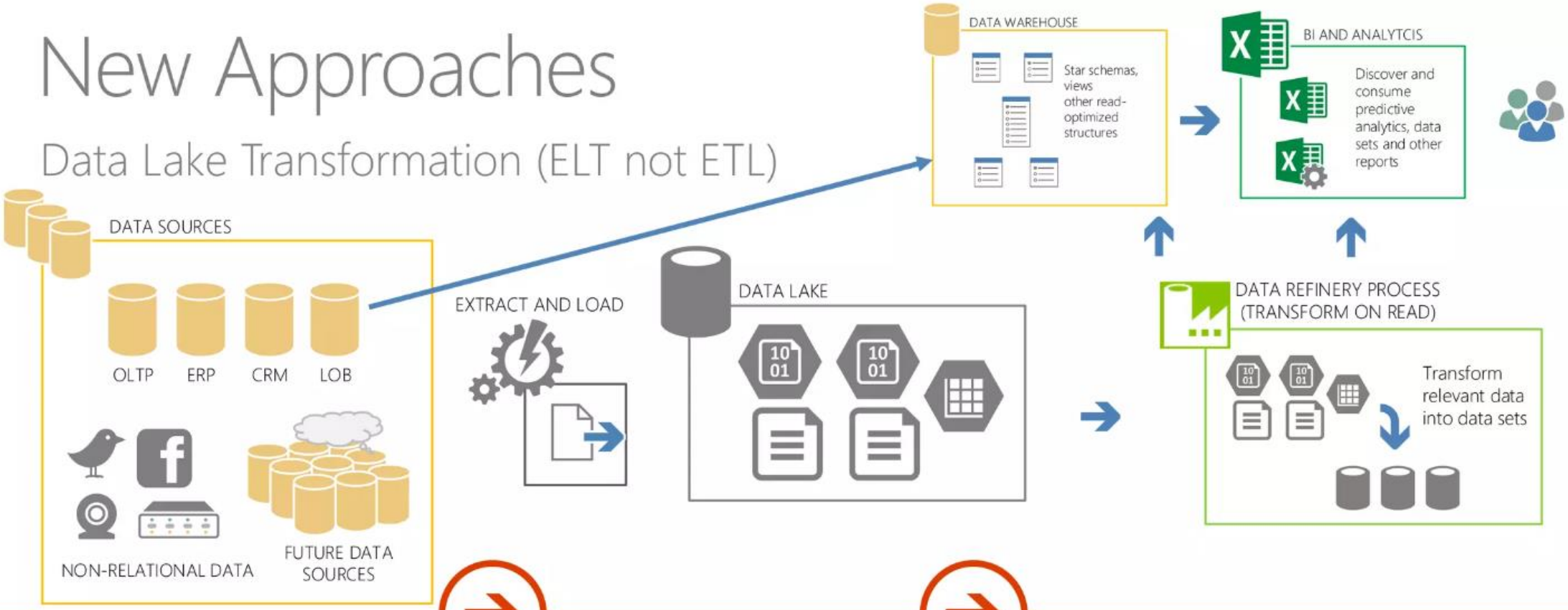
- Increase in variety of data sources
- Increase in data volume
- Increase in types of data
- Pressure on the ingestion engine

- Complex, rigid transformations can't longer keep pace
- Monitoring is abandoned
- Delay in data, inability to transform volumes, or react to new sources
- Repair, adjust and redesign ETL

- Reports become invalid or unusable
- Delay in preserved reports increases
- Users begin to "innovate" to relieve starvation

New Approaches

Data Lake Transformation (ELT not ETL)



All data sources are considered

Leverages the power of on-prem technologies and the cloud for storage and capture

Native formats, streaming data, big data

Extract and load, no/minimal transform

Storage of data in near-native format

Orchestration becomes possible

Streaming data accommodation becomes possible

Refineries transform data on read

Produce curated data sets to integrate with traditional warehouses

Users discover published data sets/services using familiar tools

Data Analysis Paradigm Shift

- OLD WAY : Structure → Ingest → Analyze
- NEW WAY : Ingest → Analyze → Structure

Data Lake Layers

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

Organizing a Data Lake – Folder Structure

Objectives

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

Special thanks to:
Melissa Coates
CoatesDataStrategies.com

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…

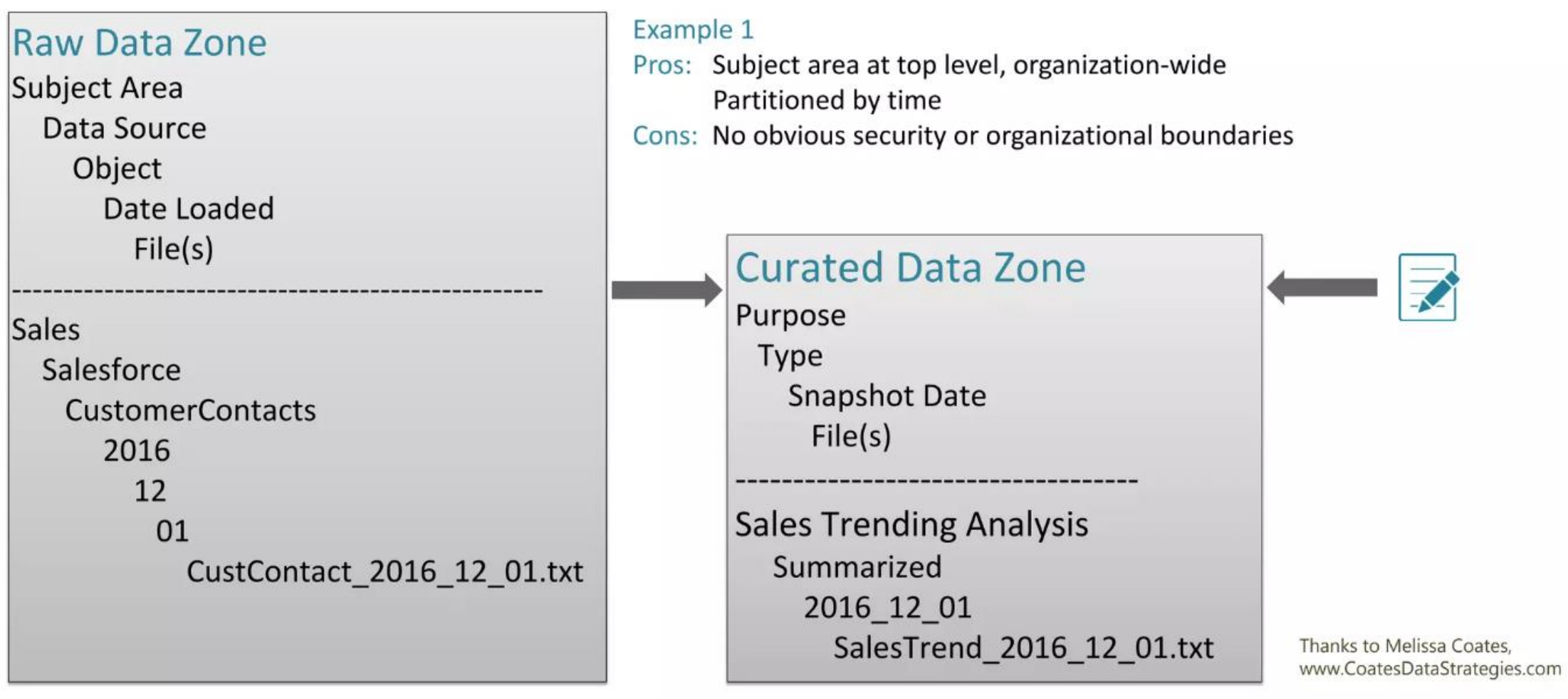
Business Impact / Criticality
High (HBI)
Medium (MBI)
Low (LBI)
etc…

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



Data Lake with DW use cases

Data Lake

Staging & preparation

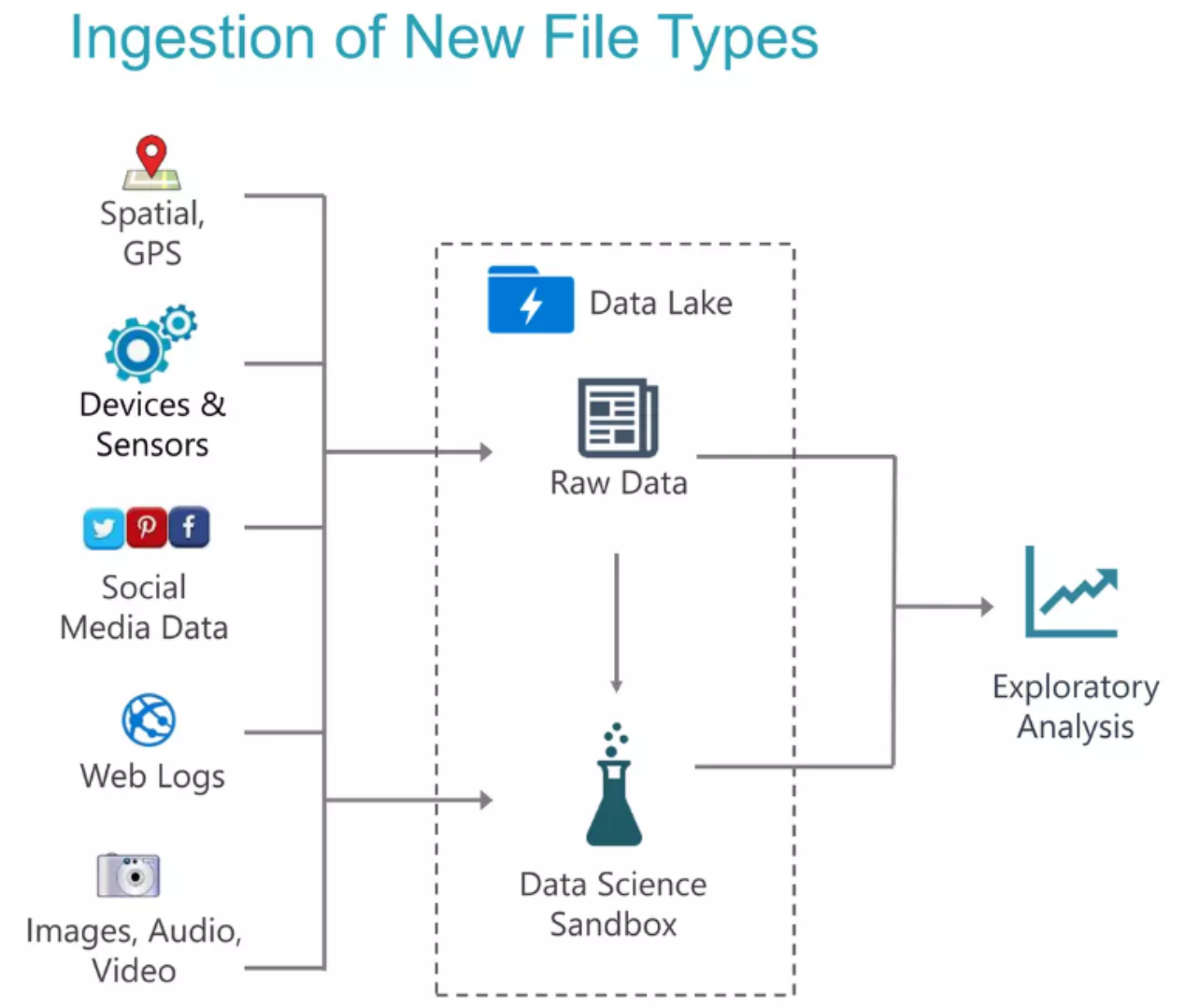
- Data scientists/Power users
- Batch processing
- Data refinement/cleaning
- ETL workloads
- Store older/backup data
- Sandbox for data exploration
- One-time reports
- Quick access to data
- Don't know questions

Data Warehouse

Serving, Security & Compliance

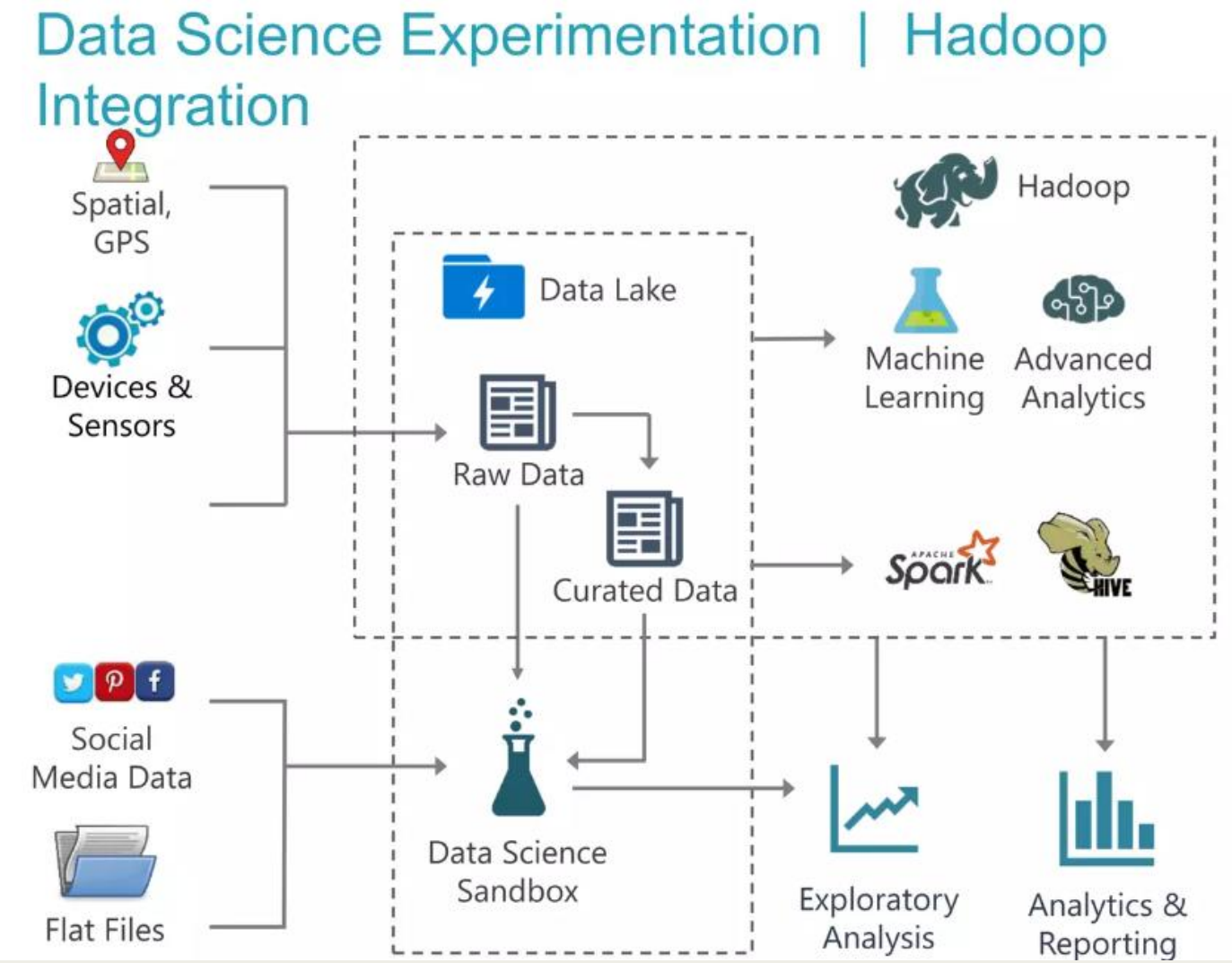
- Business people
- Low latency
- Complex joins
- Interactive ad-hoc query
- High number of users
- Additional security
- Large support for tools
- Dashboards
- Easily create reports (Self-service BI)
- Know questions

Data Lake Use Cases



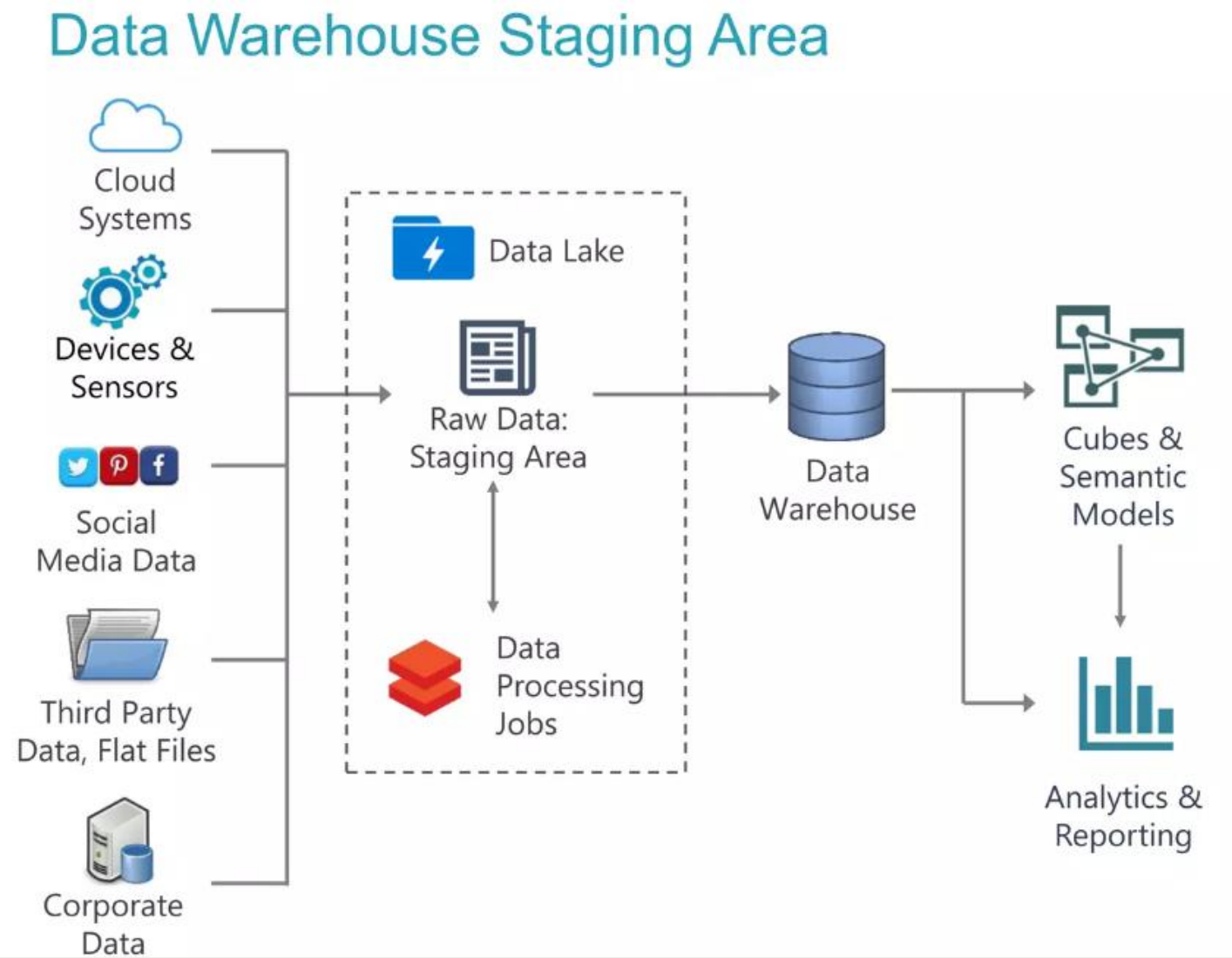
- Preparatory file storage for multi-structured data
- Exploratory analysis + POCs to determine value of new data types & sources
- Affords additional time for longer-term planning while accumulating data or handling an influx of data

Data Lake Use Cases



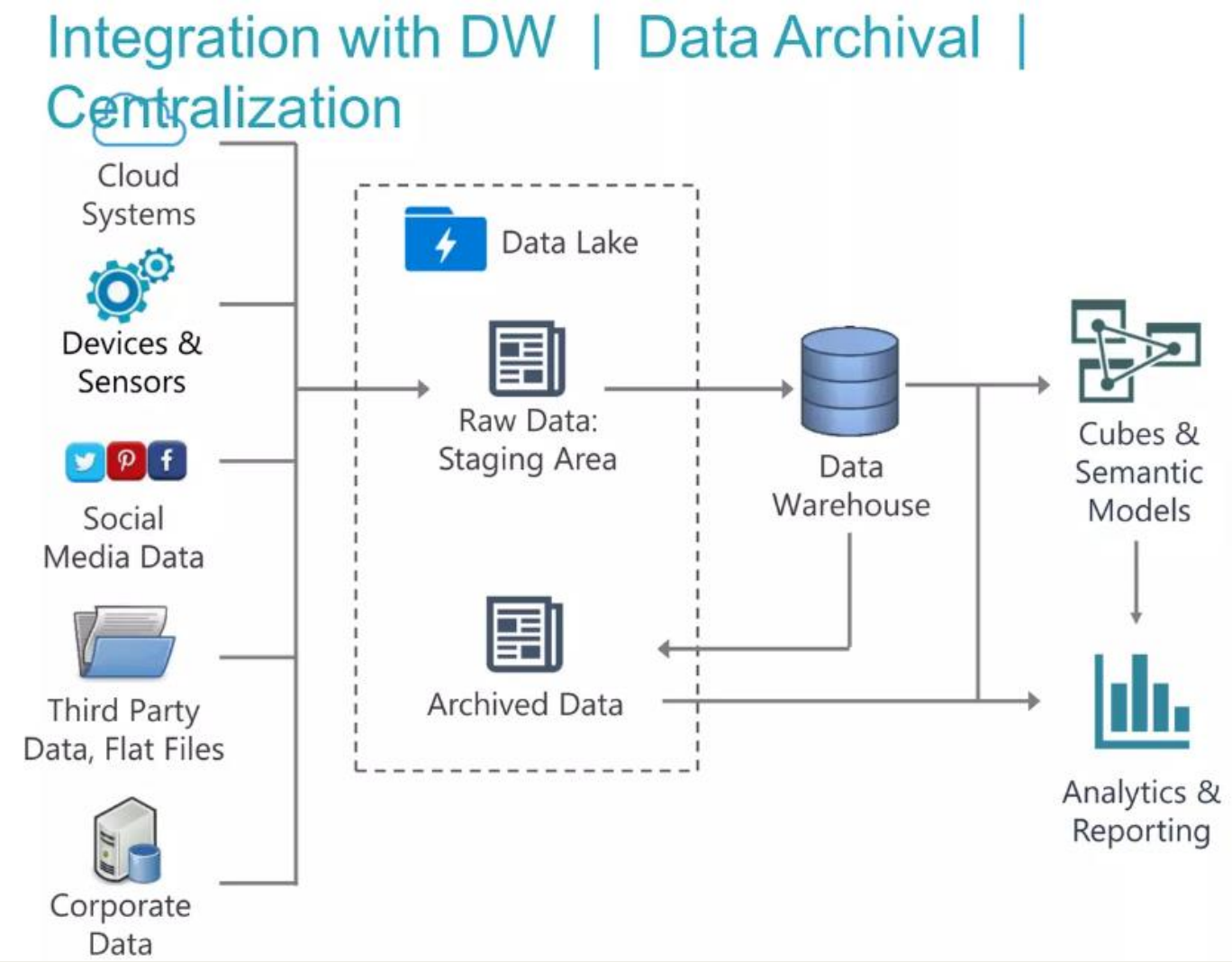
- Sandbox solution for initial data prep, experimentation, and analysis
- Migrate from proof of concept to operationalized solution
- Integrate with open source project such as Hive, Pig, Spark, Storm, etc
- Big data clusters
- SQL-on-Hadoop solution

Data Lake Use Cases



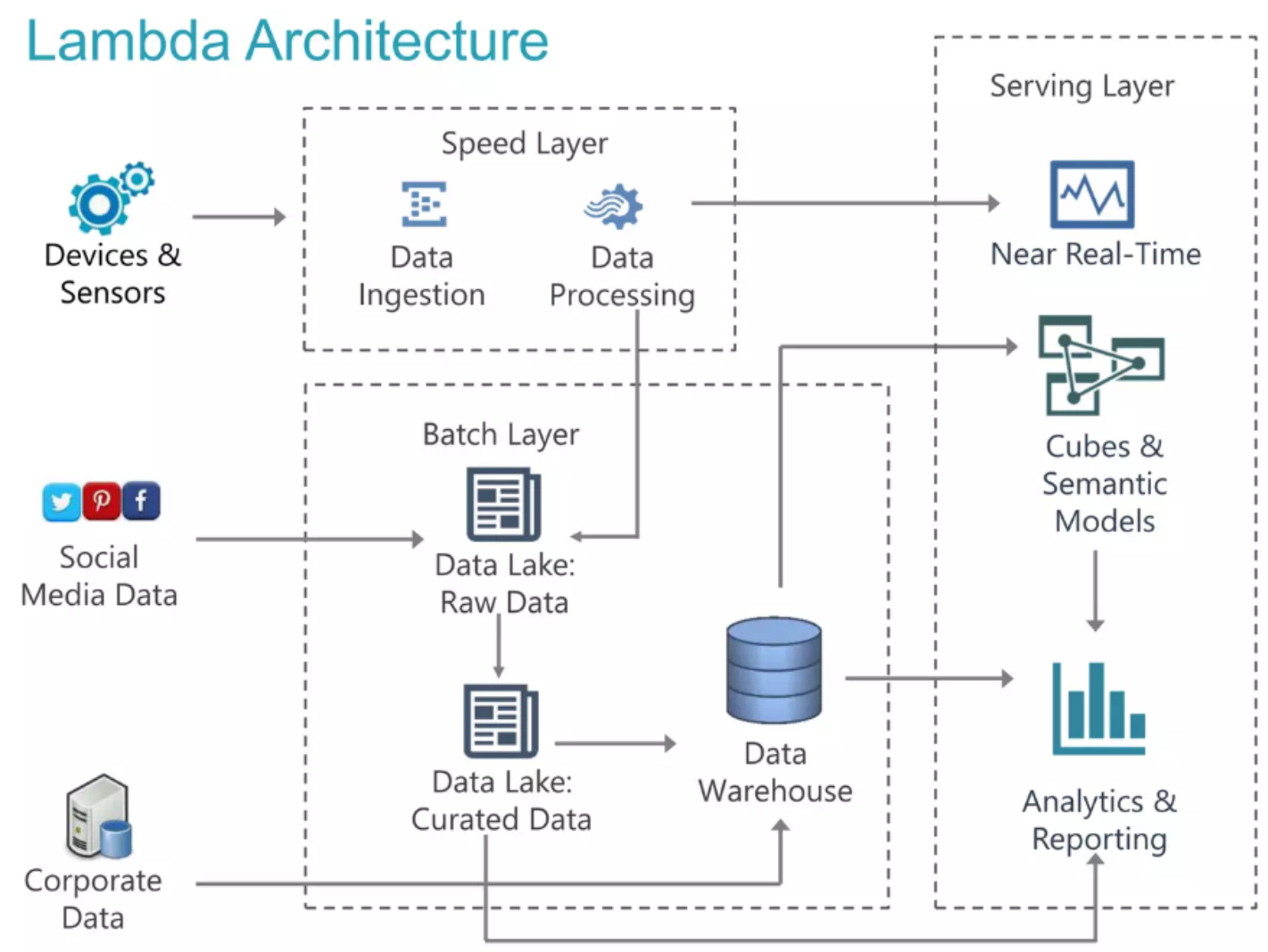
- ETL strategy
- Reduce storage needs in relational platform by using the data lake as landing area
- Practical use from data stored in the data lake
- Potentially also handle transformation in the data lake

Data Lake Use Cases



- Grow around existing DW
- Aged data available for querying when needed
- Complement to the DW via data virtualization
- Federated queries to access current data (relational DB) + archive (data lake)

Data Lake Use Cases



- Support for low-latency, high velocity data in near real time
- Support for batch-oriented operations