Microsoft Azure

Azure Cloud Scale Analytics

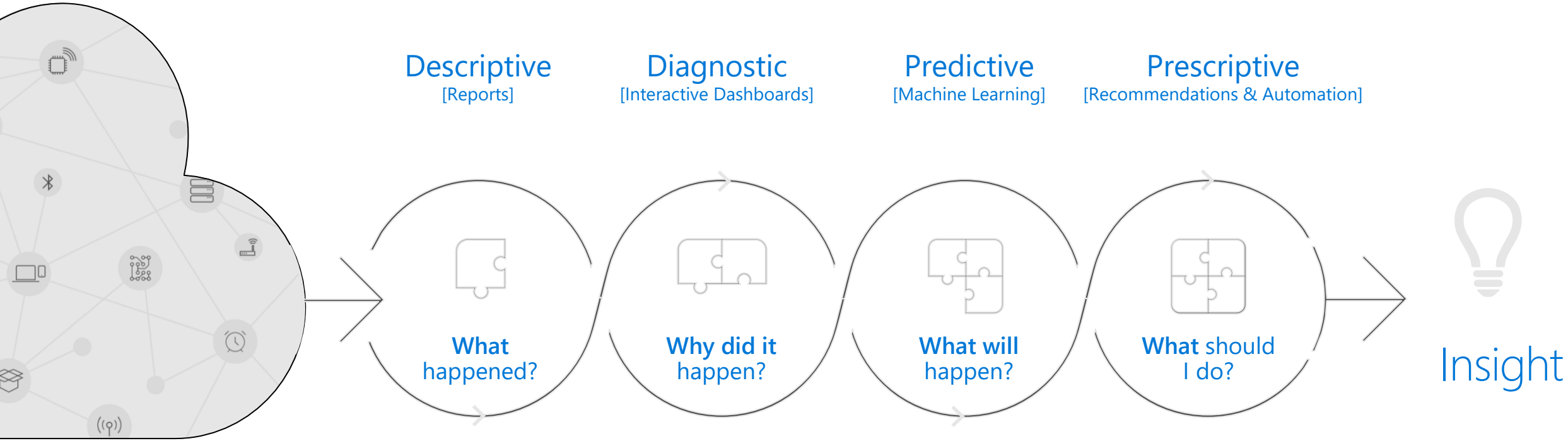
Implement a Modern Data Platform Architecture

Manjunath Suryanarayana
Sr. Cloud Solution Architect
masuryan@microsoft.com

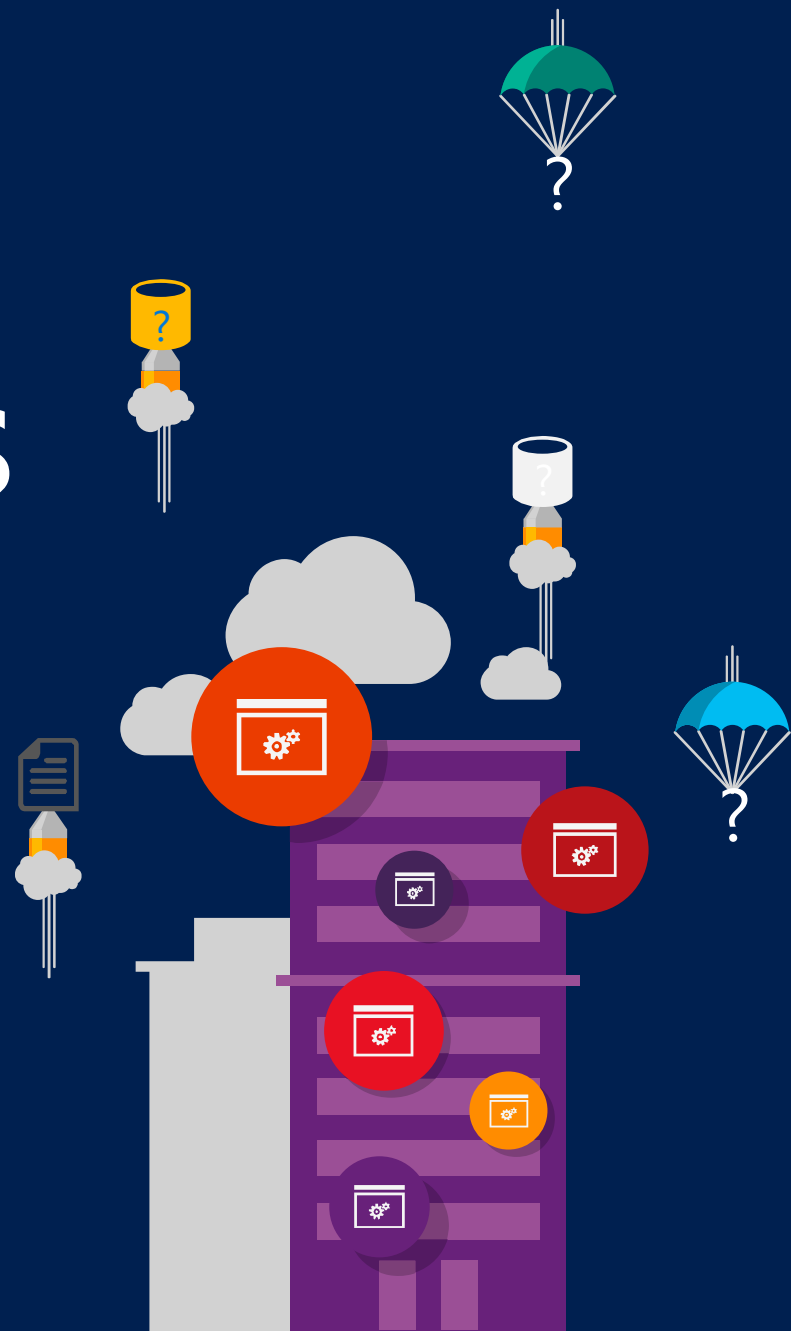
Agenda

- We will understand Cloud and Big Data concepts and technologies used to solve the **most common** advanced analytics problems
- We will understand the role of Microsoft Azure data services in a modern data platform architecture
- We will look at individual Azure Data Services and use them to implement a modern data platform reference architecture

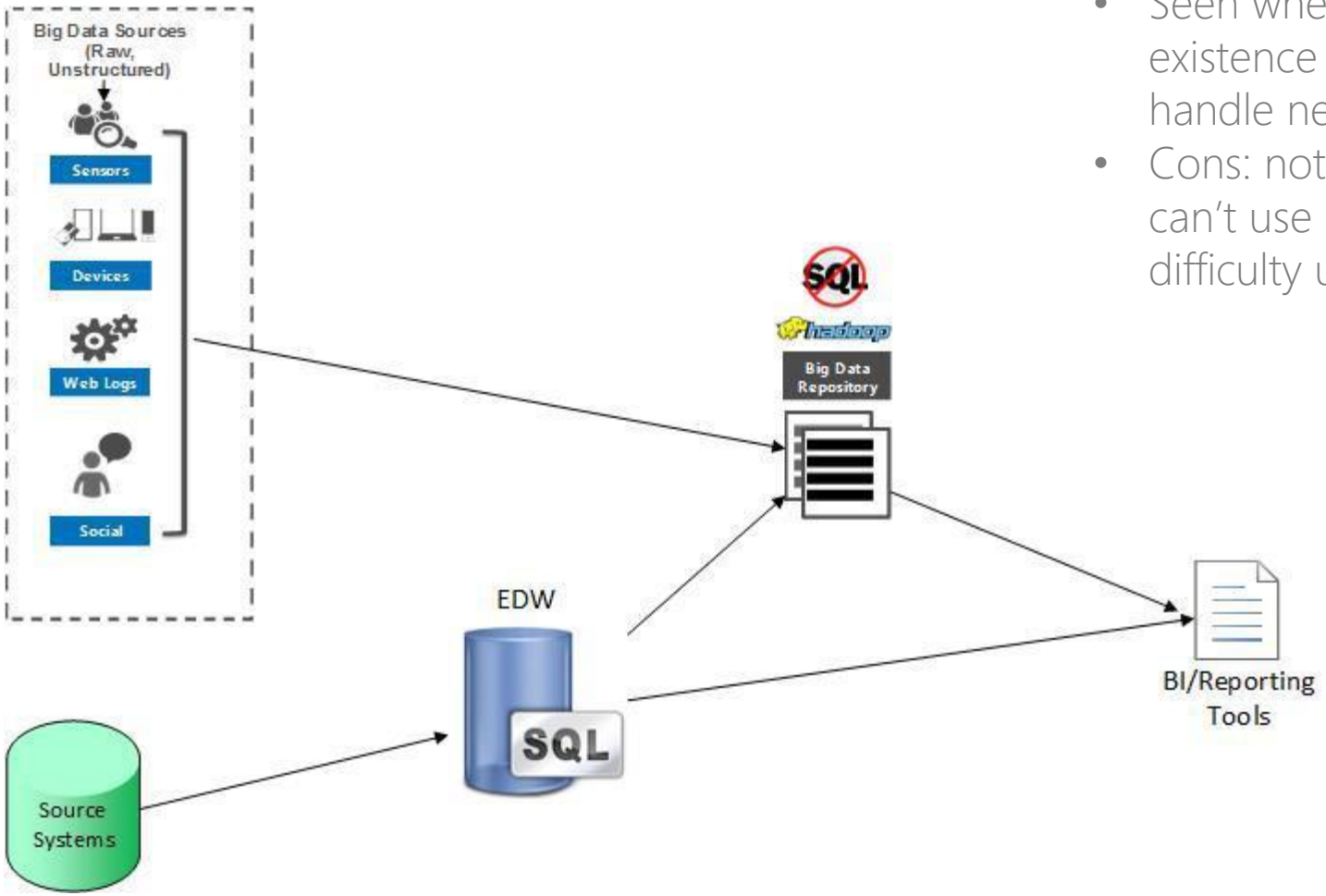
From data to decisions and actions



Big Data Architectures

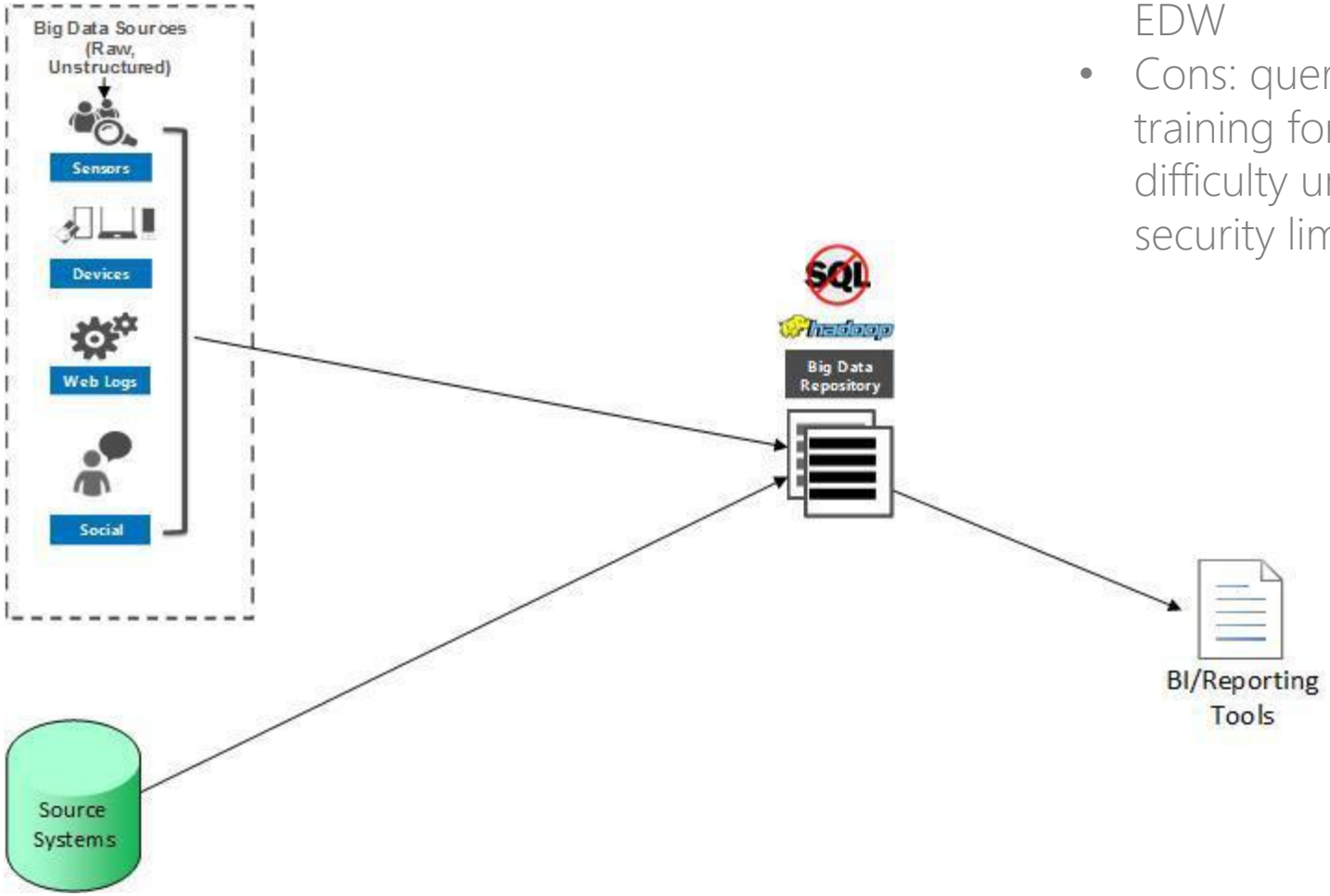


Enterprise data warehouse augmentation



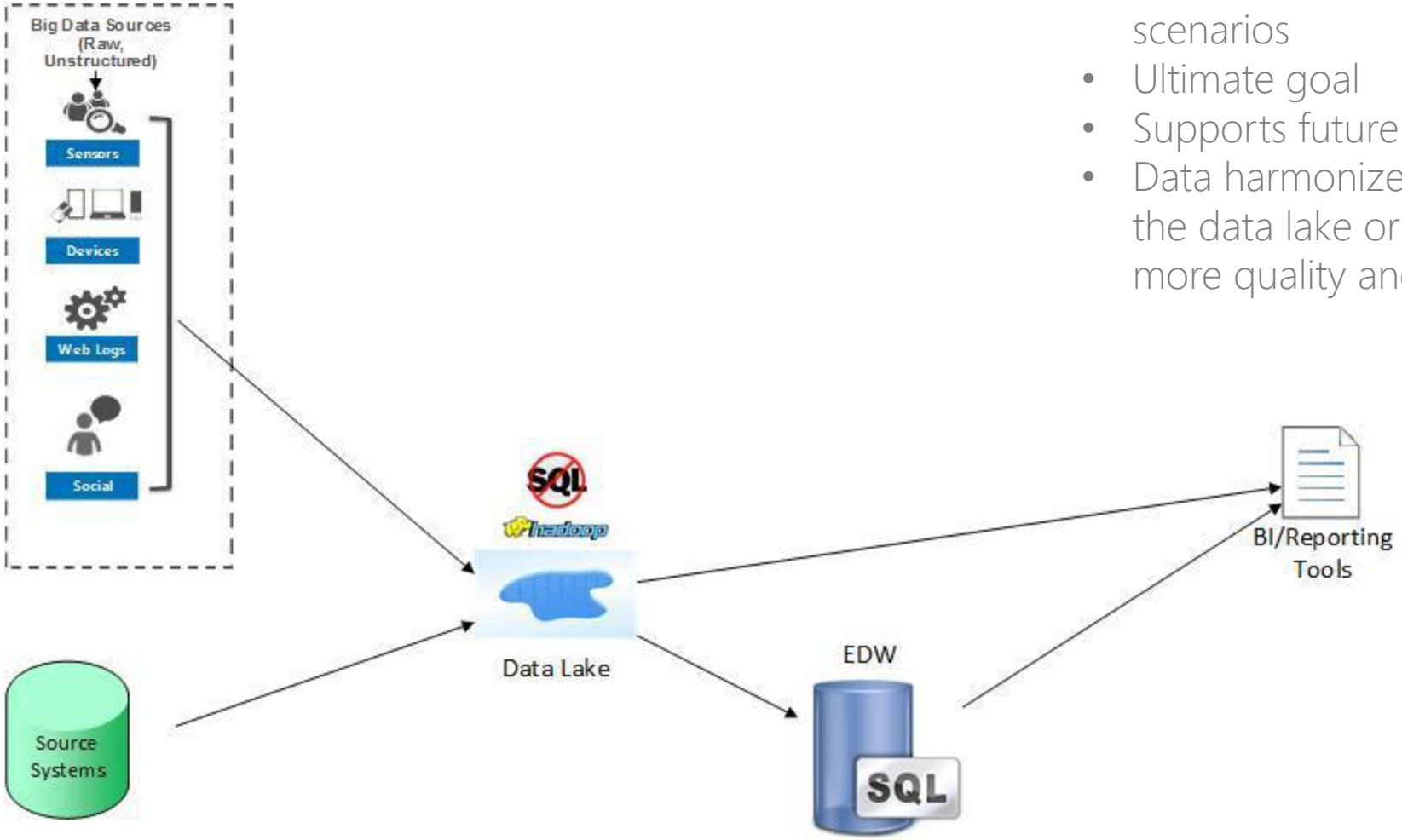
- Seen when EDW has been in existence a while and EDW can't handle new data
- Cons: not offloading EDW work, can't use existing tools, data hub difficulty understanding data

Data Hub



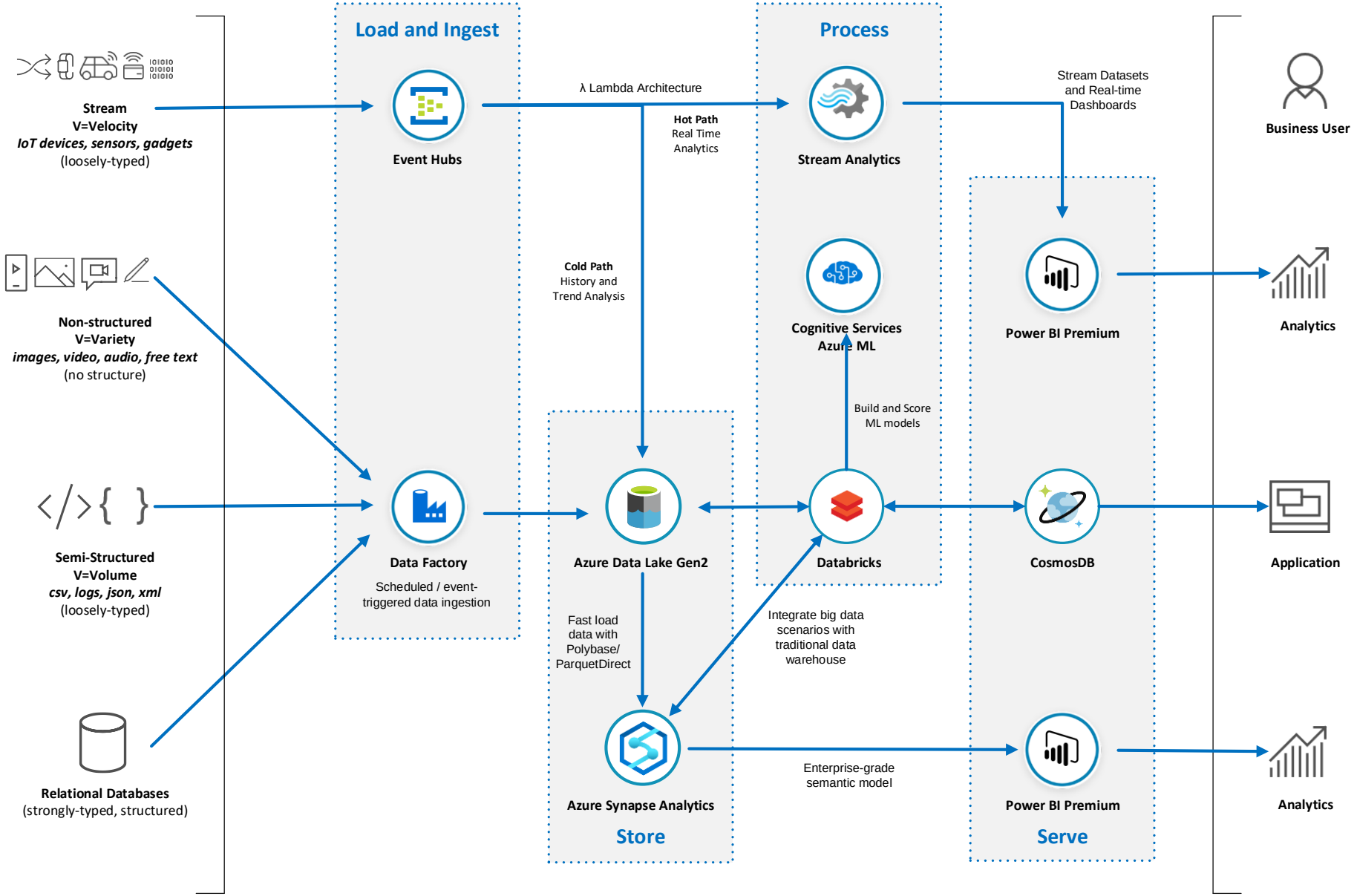
- Data hub is total solution, no EDW
- Cons: queries are slower, new training for reporting tools, difficulty understanding data, security limitations

Modern Data Warehouse



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

Modern Data Platform Reference Architecture

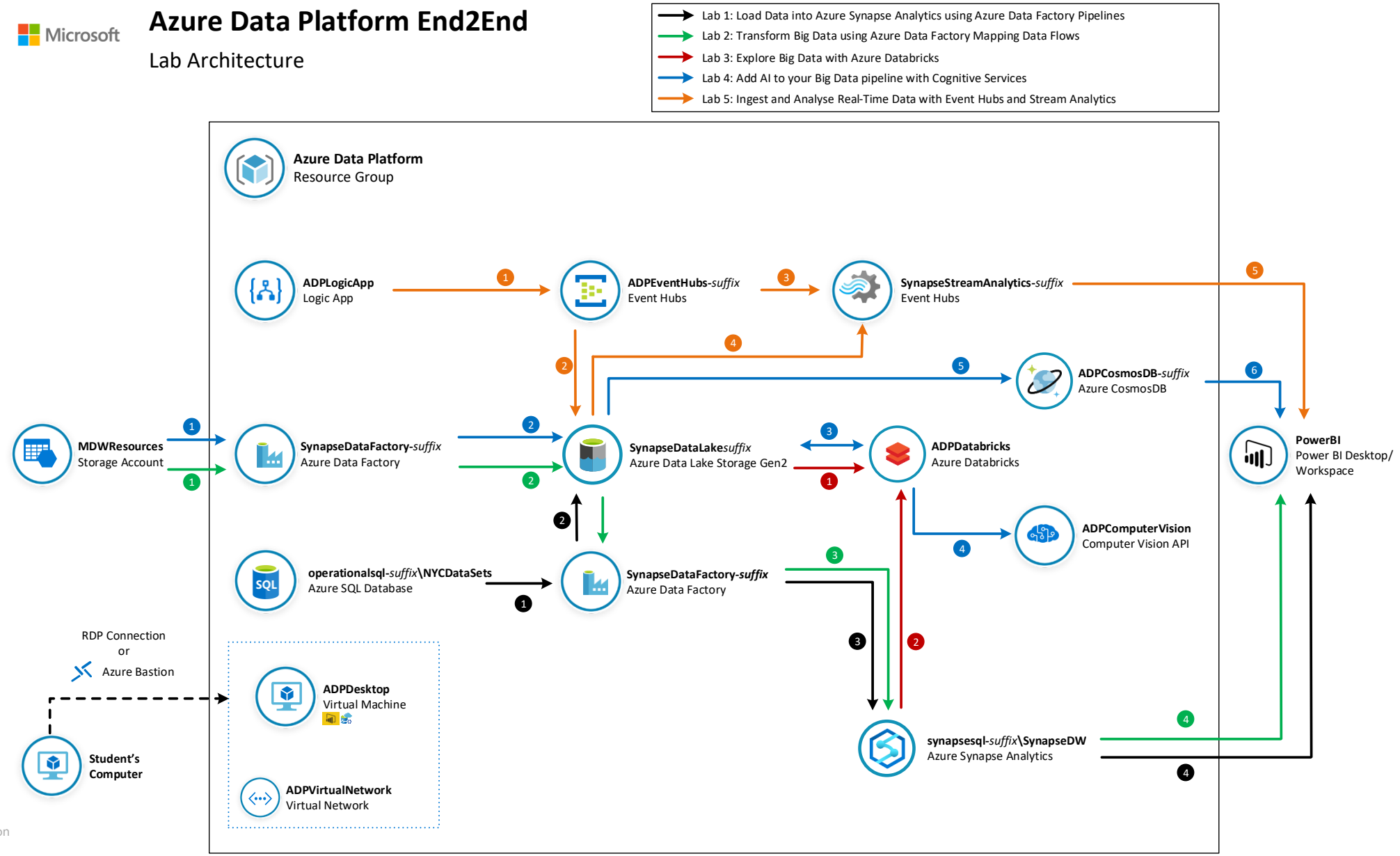


Lab Guide

Microsoft

Azure Data Platform End2End

Lab Architecture



The modern data world out there

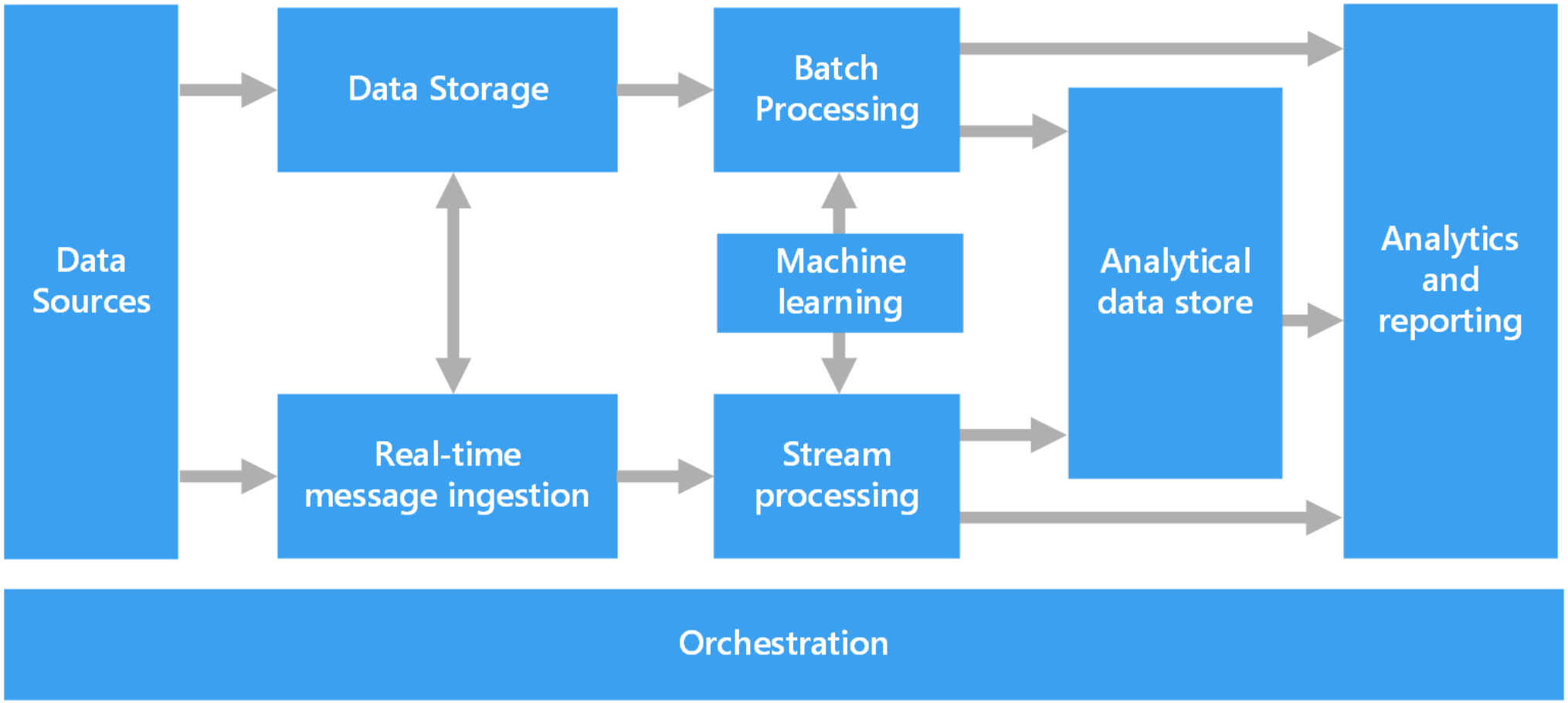
I tried to understand it, but…

No-SQL
Databricks
Storm
Data Catalog
IoT
PaaS vs IaaS
Hadoop
Power BI
Streaming
Deep Learning
Machine Learning
SMP vs MPP
Predictive
Data Mart
ETL vs ELT
Prescriptive
Data Visualisation
Data Warehouse
Data Lake
Master Data
Big Data
Data Factory
Cloud vs On-prem
Data Quality
Velocity, Variety and Volume
Semantic Layer
Spark
AI

Azure Data Architecture Guide

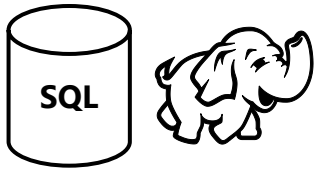
Valuable collection of architecture principles to help you with your technology choices

https://aka.ms/adag



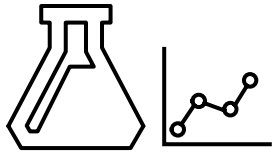
Modern Data Platform Solution Scenarios

Big Data and advanced analytics



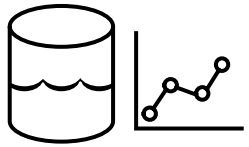
Modern data warehousing

"We want to integrate all our data—including Big Data—with our data warehouse"



Advanced analytics

"We're trying to predict when our customers churn"



Real-time analytics

"We're trying to get insights from our devices in real-time"

What is a Data Warehouse?

A data warehouse is a large collection of business data used to help an organization make decisions. Data in the Data Warehouse has been identified as valuable to specifically defined business cases and is stored in a structured way readily available for reporting and data analysis.

It is not an Operational Database

Different workload types: transactional (DB) versus analytics (DW)

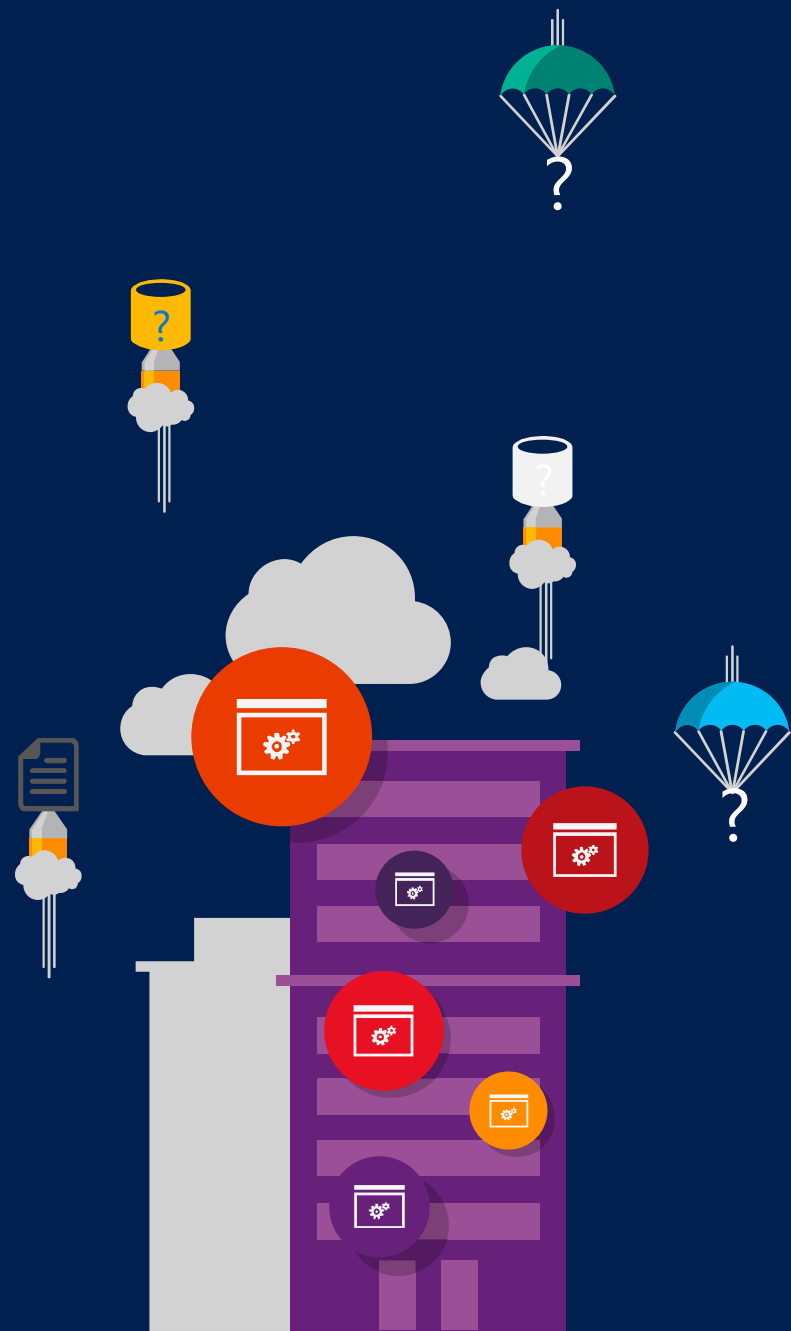
It is not a Data Lake

These are different concepts, they can co-exist and they compliment each other

It is not a Data Mart

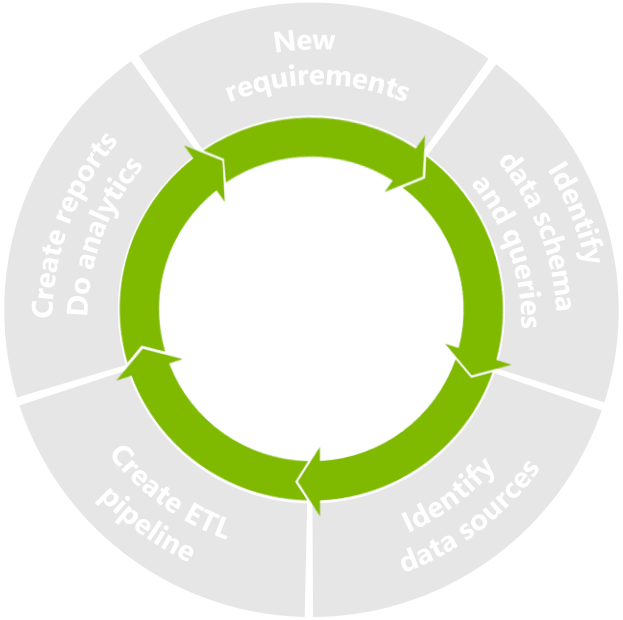
A data mart is a subject-oriented database populated from a subset of the Data Warehouse

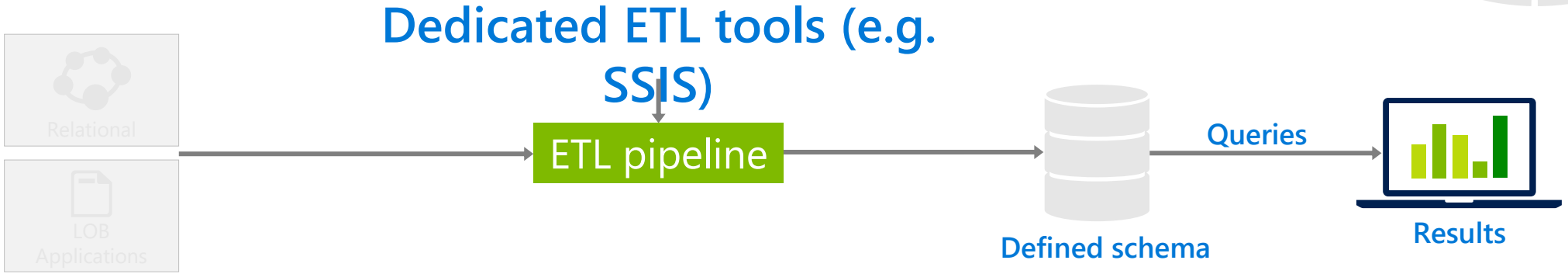
Why data lakes?



Traditional business analytics process

1. Start with end-user requirements to identify desired reports and analysis
2. Define corresponding database schema and queries
3. Identify the required data sources
4. Create a Extract-Transform-Load (ETL) pipeline to extract required data (curation) and transform it to target schema (*'schema-on-write'*)
5. Create reports. Analyze data





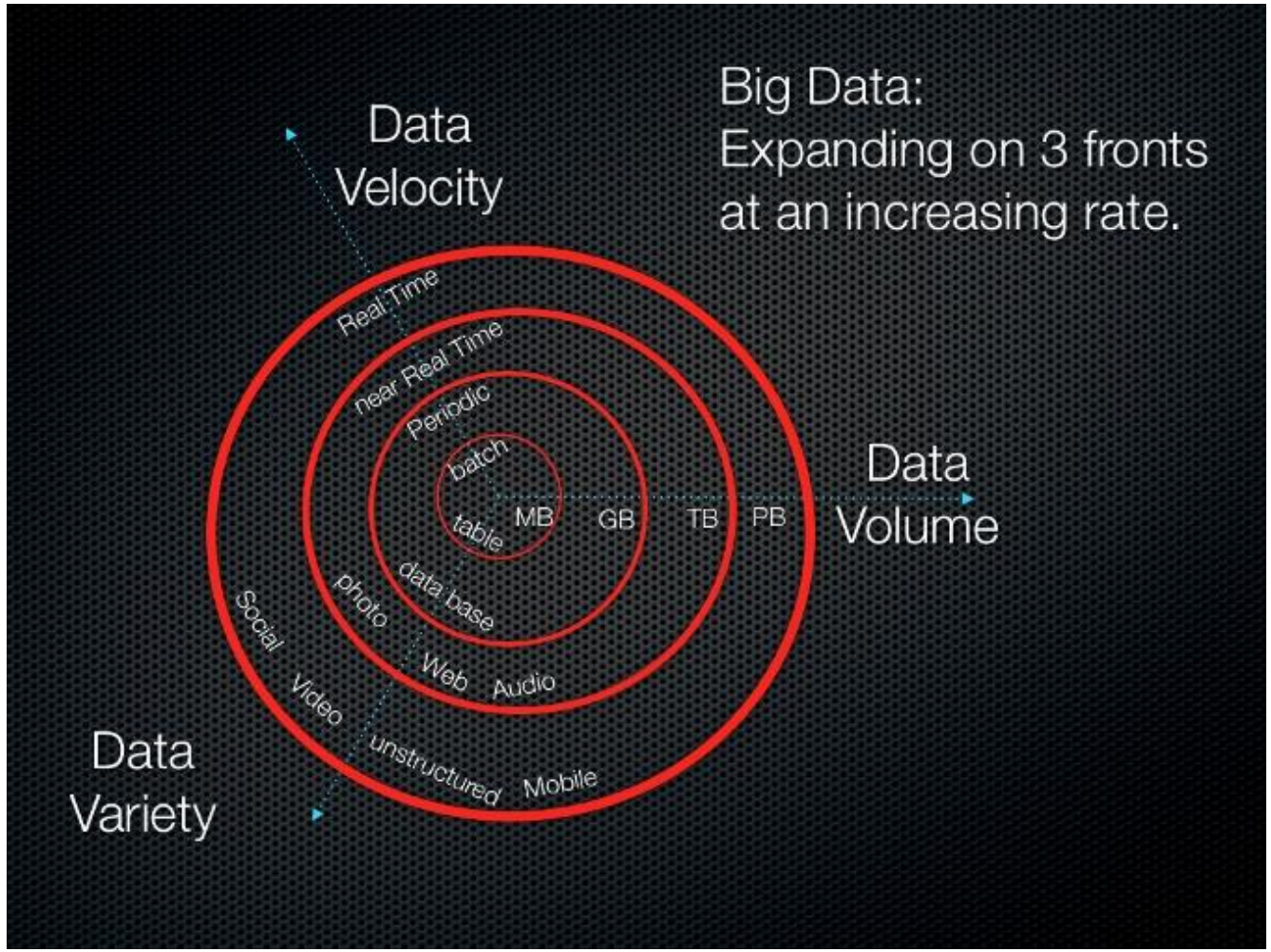
All data not immediately required is discarded or archived

Need to collect any data

Harness the growing and changing nature of data

- Challenge is combining transactional data stored in relational databases 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

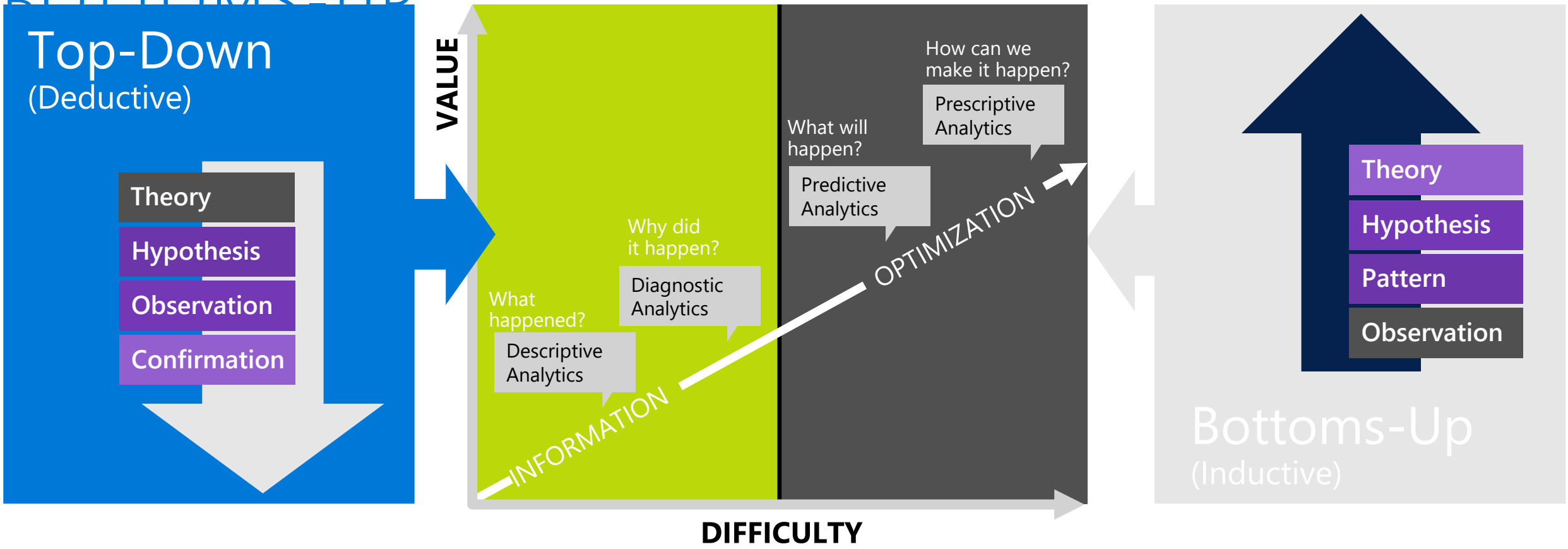

Big Data:
Expanding on 3 fronts
at an increasing rate.
Data
Velocity
Real Time
near Real Time
Periodic
batch
table
MB
GB
TB
PB
Data
Volume
data base
photo
Social
Web
Audio
Video
unstructured
Mobile
Data
Variety

New big data thinking: All data has value

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

Top-down vs Bottom-up

TWO APPROACHES TO INFORMATION MANAGEMENT FOR ANALYTICS: TOP-DOWN + BOTTOMS-UP



DATA WAREHOUSING USES A TOP-DOWN APPROACH

Understand Corporate Strategy

Gather Requirements

Business Requirements

Technical Requirements

Implement Data Warehouse

- Reporting & Analytics Design → Reporting & Analytics Development
- Dimension Modelling → Physical Design
- ETL Design → ETL Development
- Setup Infrastructure → Install and Tune

BI and analytic

Dashboards | Reporting

Data warehouse

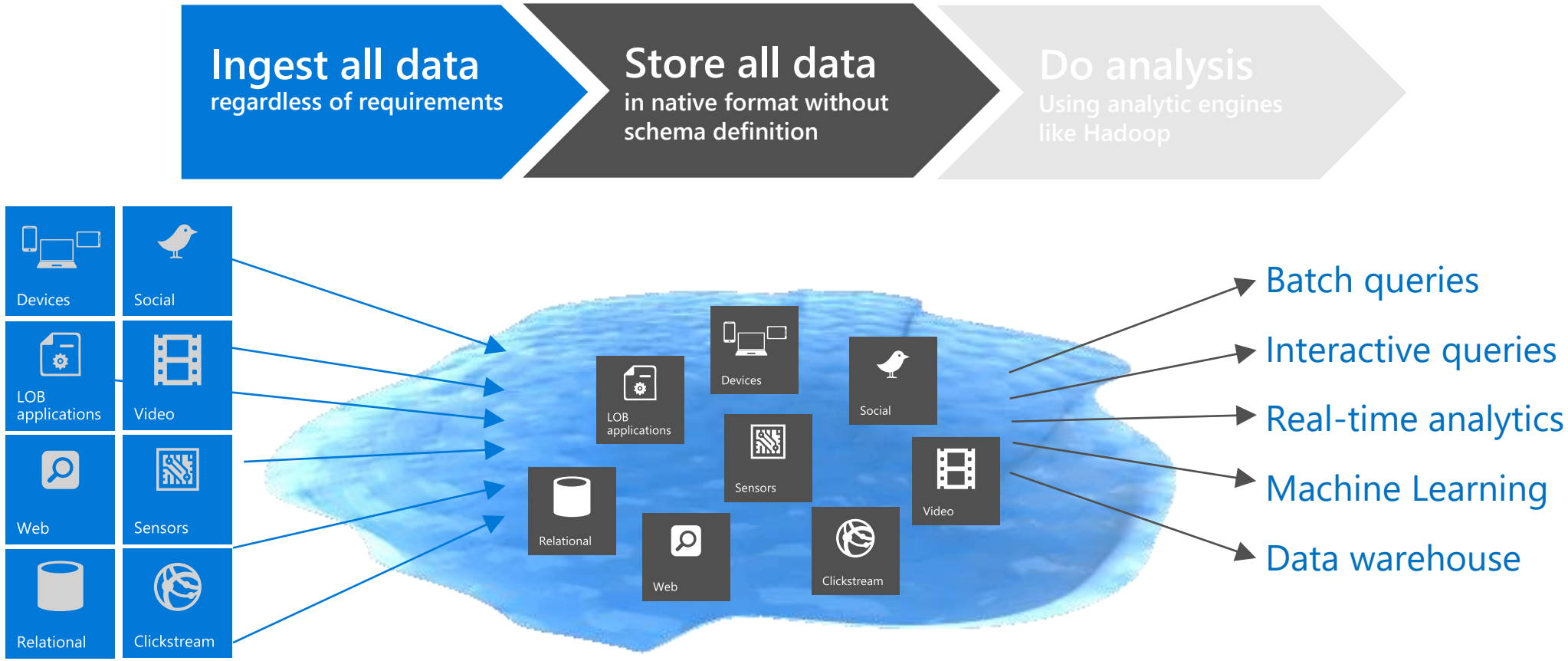
01100
10010
00101

ETL

Data sources

OLTP | ERP | CRM | LOB

THE "DATA LAKE" USES A BOTTOMS-UP APPROACH



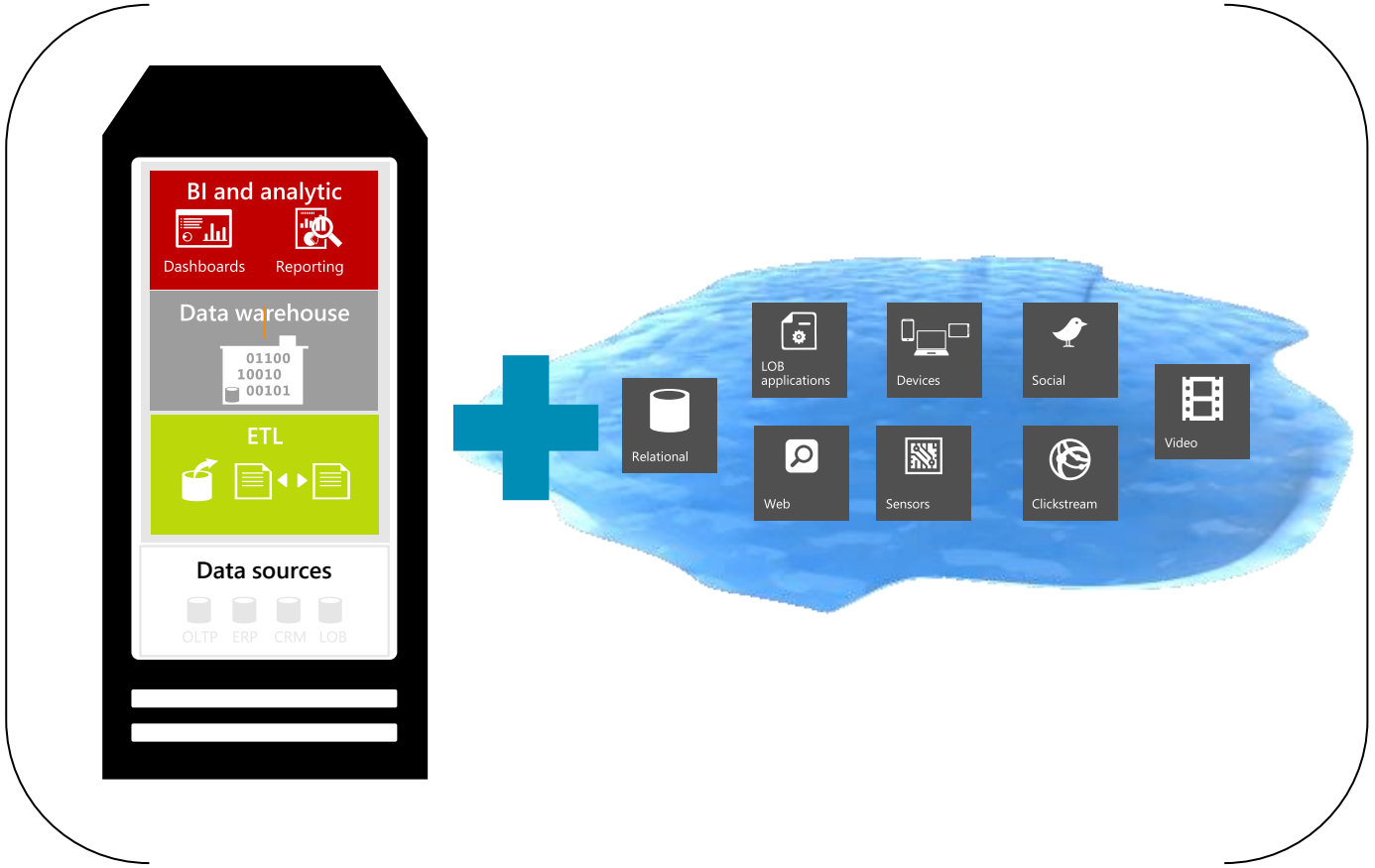
DATA LAKE + DATA WAREHOUSE BETTER TOGETHER

What happened?

Descriptive Analytics

Why did it happen?

Diagnostic Analytics



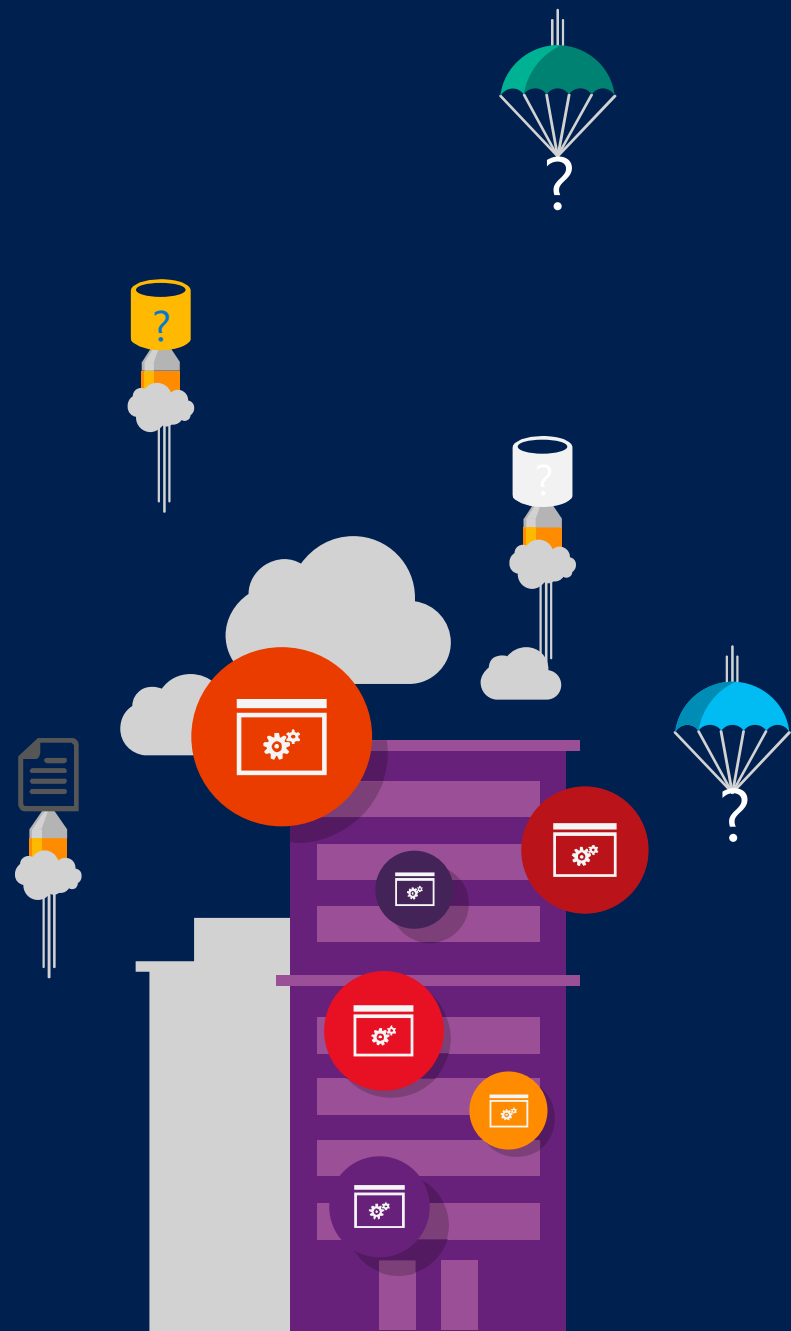
What will happen?

Predictive Analytics

How can we make it happen?

Prescriptive Analytics

Data lake defined



WHAT IS A DATA LAKE?

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

- A place to store unlimited amounts of data in any format **inexpensively**, especially for **archive purposes**
- Allows **collection of data** that you may or may not use later: "just in case"
- A way to describe any large data pool in which the schema and data requirements are not defined until the data is queried: "just in time" or "**schema on read**"
- **Complements EDW** and can be seen as a data source for the EDW – capturing all data but only passing relevant data to the EDW
- **Frees up expensive EDW resources** (storage and processing), especially for data refinement
- Allows for data exploration to be performed without waiting for the EDW team to model and load the data (**quick user access**)
- Some processing in better done with **Hadoop tools** than ETL tools like SSIS
- **Easily scalable**

DATA ANALYSIS PARADIGM SHIFT

OLD WAY: Structure -> Ingest -> Analyze

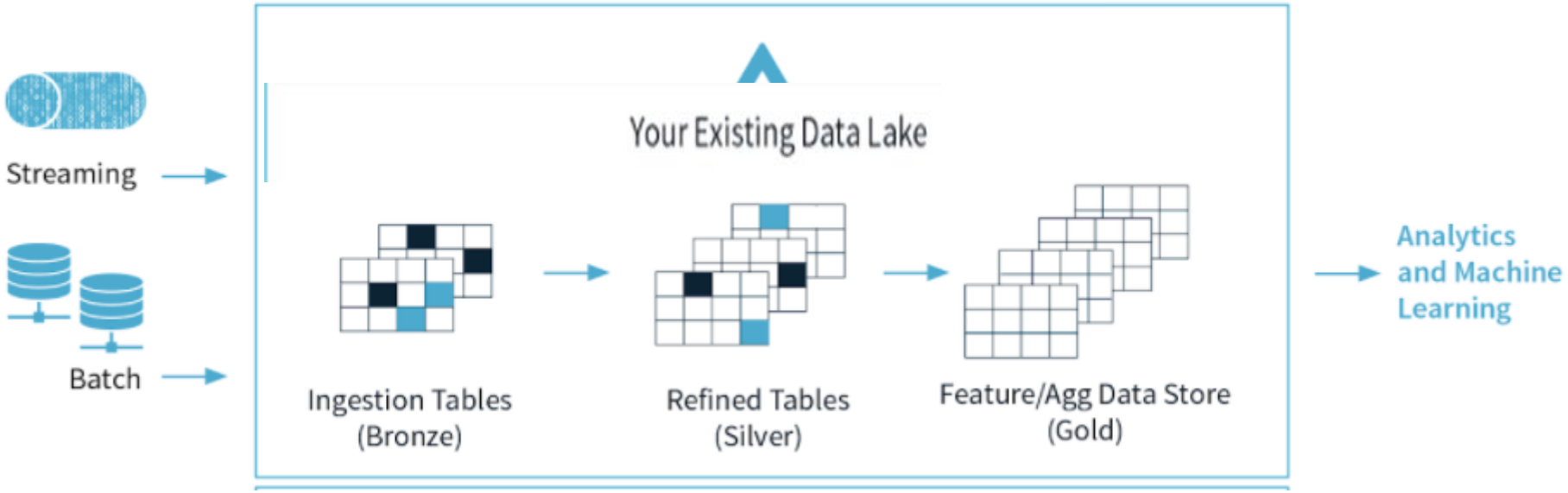
NEW WAY: Ingest -> Analyze -> Structure

DATA LAKE LAYERS

- **Raw data layer**– Raw events are stored for historical reference. Also called staging layer or landing area
- **Cleansed data layer** – Raw events are transformed (cleaned and mastered) into directly consumable data sets. Aim is to uniform the way files are stored in terms of encoding, format, data types and content (i.e. strings). Also called conformed layer
- **Application data layer** – Business logic is applied to the cleansed data to produce data ready to be consumed by applications (i.e. DW application, advanced analysis process, etc). Also called workspace layer or trusted layer

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

DATA LAKE LAYERS – MULTIHOP ARCHITECTURE



Combined, we refer to these tables as a “multi-hop” architecture. It allows data engineers to build a pipeline that begins with raw data as a **“single source of truth”** from which everything flows. Subsequent transformations and aggregations can be recalculated and validated to ensure that business-level aggregate tables still reflective the underlying data, even as downstream users refine the data and introduce context-specific structure.

SHOULD I USE HADOOP OR NOSQL FOR THE DATA LAKE?

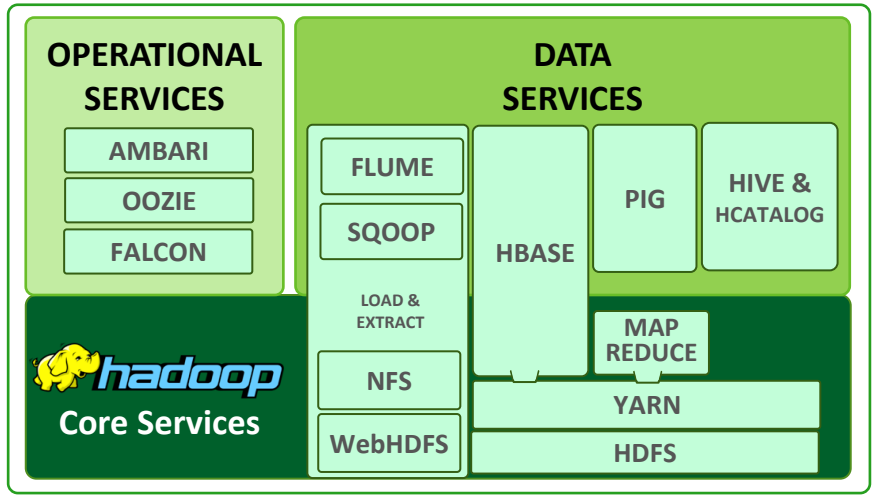
Most implementations use Hadoop as the data lake because of these benefits:

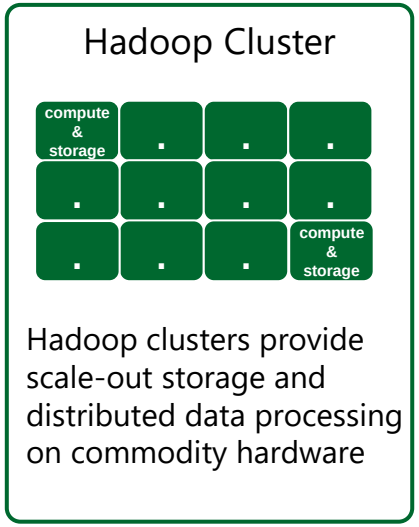
- Open-source software ecosystem that allows for massively parallel computing
- No inherent structure (no conversion to JSON needed)
- Good for batch processing, large files, volume writes, parallel scans, sequential access (NoSQL designed for large-scale OLTP)
- Large ecosystem of products
- Low cost
- Con: performance

Hadoop as the data lake

WHAT IS HADOOP?

- Distributed, scalable system on commodity HW
- Composed of a few parts:
  - HDFS – Distributed file system
  - MapReduce – Programming model
  - Other tools: Hive, Pig, SQOOP, HCatalog, HBase, Flume, Mahout, YARN, Tez, Spark, Stinger, Oozie, ZooKeeper, Flume, Storm
- Main players are Hortonworks, Cloudera, MapR
- WARNING: Hadoop, while ideal for processing huge volumes of data, is inadequate for analyzing that data in real time (companies do batch analytics instead)





HORTONWORKS DATA PLATFORM 2.5

Ongoing Innovation in Apache

		Pig	Hive	Druid	Tez	Solr	Spark	Zeppelin	Slider	HBase	Phoenix	Accumulo	Storm	Falcon	Atlas	Sqoop	Flume	Kafka	Ambari	Zookeeper	Oozie	Knox	Ranger
HDP 2.6* 1H2017	2.7.3	0.16.0	1.2.1+ 2.1***	0.9.2	0.7.0	5.5.1 ****	1.6.3+ 2.1**	0.7.0	0.91.0	1.1.2	4.7.0	1.7.0	1.1.0	0.10.0	0.8.0	1.4.6	1.5.2	0.10.1.0	2.5.0	3.4.6	4.2.0	0.11.0	0.7.0
HDP 2.5 Aug 2016	2.7.3	0.16.0	1.2.1+ 2.1***		0.7.0	5.5.1	1.6.2+ 2.0**	0.6.0	0.91.0	1.1.2	4.7.0	1.7.0	1.0.1	0.10.0	0.7.0	1.4.6	1.5.2	0.10.0	2.4.0	3.4.6	4.2.0	0.9.0	0.6.0
HDP 2.4 Mar 2016	2.7.1	0.15.0	1.2.1		0.7.0	5.2.1	1.6.0		0.80.0	1.1.2	4.4.0	1.7.0	0.10.0	0.6.1	0.5.0	1.4.6	1.5.2	0.9.0	2.2.1	3.4.6	4.2.0	0.6.0	0.5.0
HDP 2.3 Oct 2015	2.7.1	0.15.0	1.2.1		0.7.0	5.2.1	1.4.1		0.80.0	1.1.2	4.4.0	1.7.0	0.10.0	0.6.1	0.5.0	1.4.6	1.5.2	0.8.2	2.1.0	3.4.6	4.2.0	0.6.0	0.5.0
HDP 2.2 Dec 2014	2.6.0	0.14.0	0.14.0		0.5.2	4.10.2	1.2.1		0.60.0	0.98.4	4.2.0	1.6.1	0.9.3	0.6.0		1.4.5	1.5.2	0.8.1	2.0.0	3.4.6	4.1.0	0.5.0	0.4.0
HDP 2.1 April 2014	2.4.0	0.12.1	0.13.0		0.4.0	4.7.2				0.98.0	4.0.0	1.5.1	0.9.1	0.5.0		1.4.4	1.4.0		1.5.1	3.4.5	4.0.0	0.4.0	
HDP 2.0 Oct 2013	2.2.0	0.12.0	0.12.0							0.96.1						1.4.4	1.3.1		1.4.4	3.4.5	3.3.2		

DATA MGMT | DATA ACCESS | GOVERNANCE & INTEGRATION | OPERATIONS | SECURITY

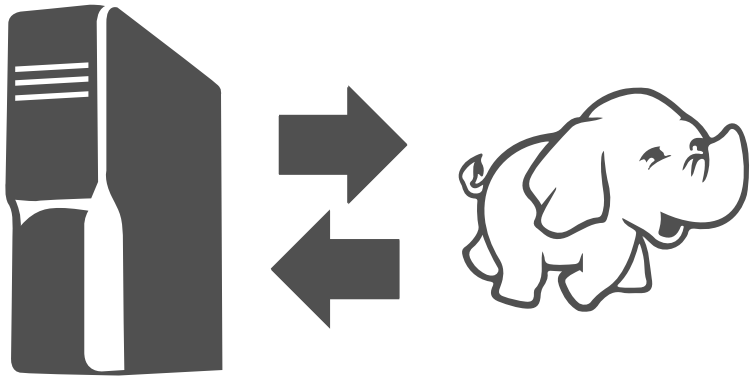
HORTONWORKS DATA PLATFORM

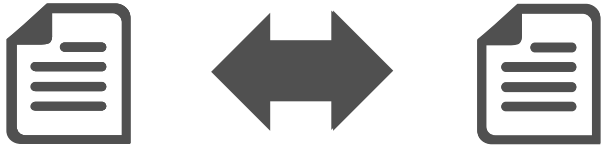
** HDP 2.6 – Shows current Apache branches being used. Final component version subject to change based on Apache release process.*
*** Spark 1.6.3+ Spark 2.1 – HDP 2.6 supports both Spark 1.6.3 and Spark 2.1 as GA.*
**** Hive 2.1 is GA within HDP 2.6.*
***** Apache Solr is available as an add-on product HDP Search.*

Simply put, Hortonworks ties all the open source products together (22)

USE CASES USING HADOOP AND A DW IN COMBINATION

BRINGING ISLANDS OF HADOOP DATA TOGETHER





Archiving data warehouse data to Hadoop (move)
(Hadoop as cold storage)

Exporting relational data to Hadoop (copy)
(Hadoop as backup/DR, analysis, cloud use)

Importing Hadoop data into data warehouse (copy)
(Hadoop as staging area, sandbox, Data Lake)

Hadoop and Spark in Azure

Open Source Apache Projects for Big Data Compute

hadoop	Apache Spark
It was the original open-source framework for distributed processing and analysis of big data sets on clusters.	Effective, fast, general-purpose unified cluster computing framework with high-level APIs in Java, Scala, Python and R.
Read/write from disk.	In-memory processing.
Economical batch mode.	Fast, interactive data processing.
Linear processing of huge datasets.	Streaming and Machine Learning Support

Azure HDInsight is a managed, full-spectrum, open-source analytics service for enterprises

What comes with HDInsight?

Apache Hadoop | Apache Spark | Apache Kafka | Apache HBase | Apache Hive LLAP | Apache Storm | Machine Learning

Azure Databricks

Azure Databricks

A fast, easy and collaborative Apache® Spark™ based analytics platform optimized for Azure

databricks

Best of Databricks

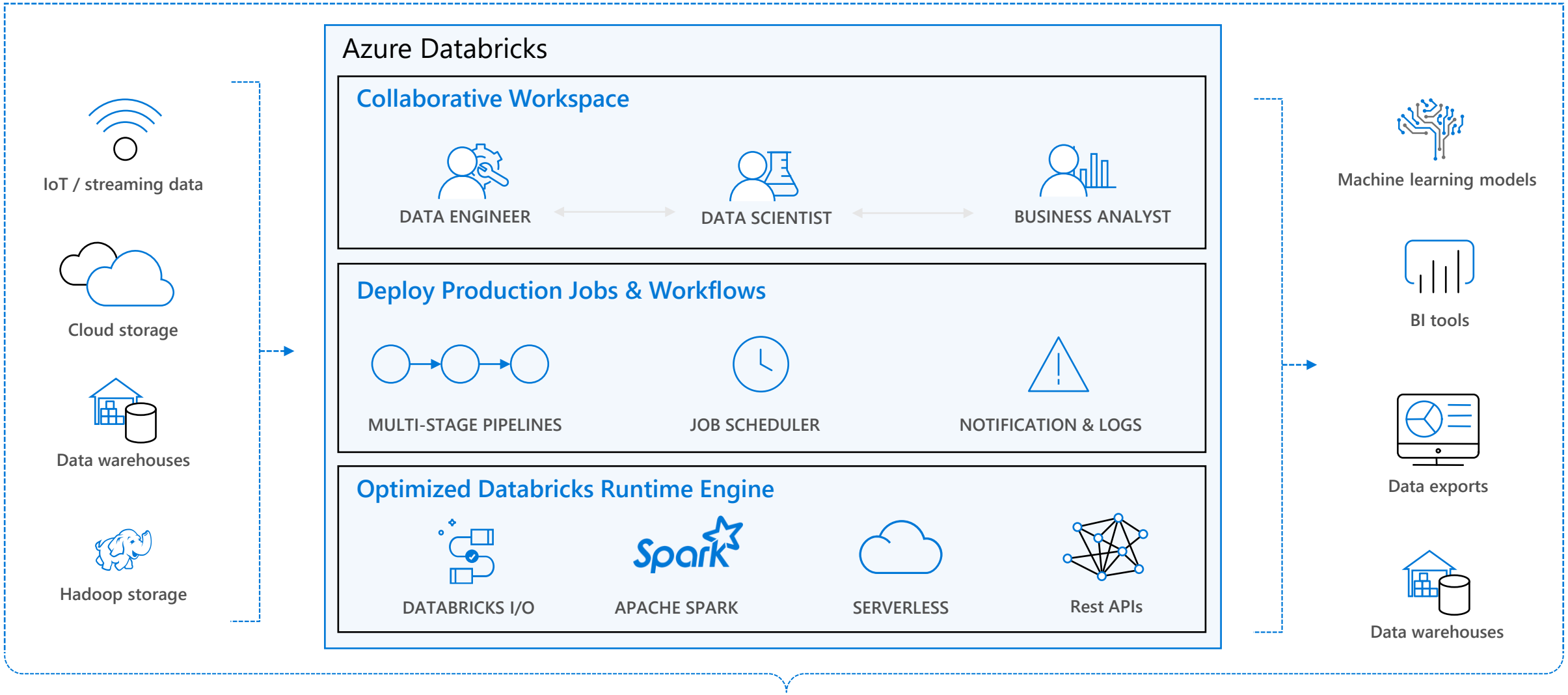
+

Microsoft

Best of Microsoft

- Designed in collaboration with the founders of Apache Spark
- One-click set up; streamlined workflows
- Interactive workspace that enables collaboration between data scientists, data engineers, and business analysts.
- Native integration with Azure services (Power BI, SQL DW, Cosmos DB, ADLS, Azure Storage, Azure Data Factory, Azure AD, Event Hub, IoT Hub, HDInsight Kafka, SQL DB)
- Enterprise-grade Azure security (Active Directory integration, compliance, enterprise-grade SLAs)

Azure Databricks



Enhance Productivity

Build on secure & trusted cloud

Scale without limits

Azure Databricks Notebooks

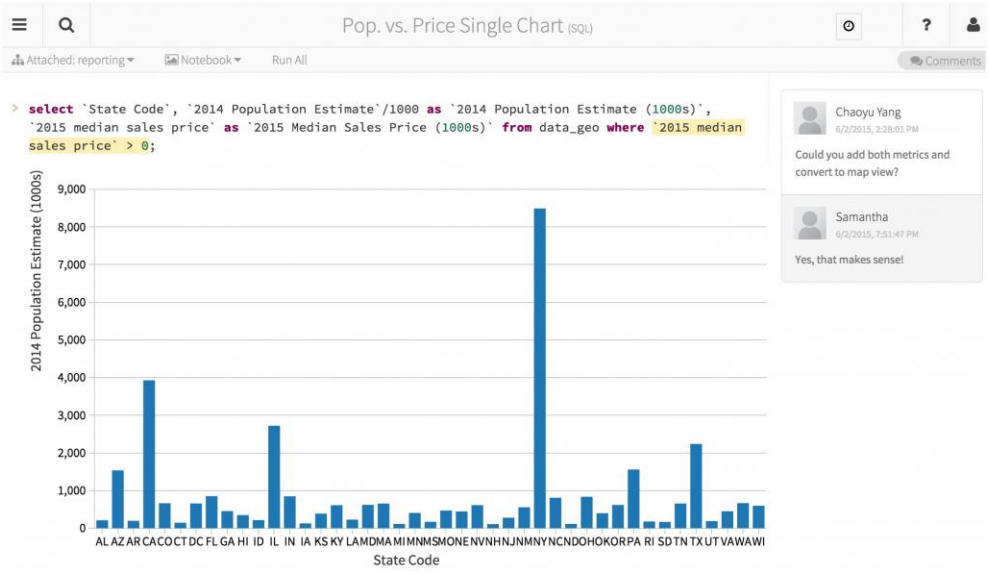
Notebooks are a popular way to develop, and run, Spark Applications

Notebooks are not only for authoring Spark applications but can be *run/executed directly* on clusters

- Shift+Enter
- click the ▶ at the top right of the cell in a notebook
- Submit via Job

Fine grained permissions support so they can be *securely shared* with colleagues for collaboration

Notebooks are well-suited for prototyping, rapid development, exploration, discovery and iterative development



With Azure Databricks notebooks you have a default language but you can mix multiple languages in the same notebook:

`%python`	Allows you to execute python code in a notebook (even if that notebook is not python)
`%sql`	Allows you to execute sql code in a notebook (even if that notebook is not sql).
`%r`	Allows you to execute r code in a notebook (even if that notebook is not r).
`%scala`	Allows you to execute scala code in a notebook (even if that notebook is not scala).
`%sh`	Allows you to execute shell code in your notebook.
`%fs`	Allows you to use Databricks Utilities - dbutils filesystem commands.
`%md`	To include rendered markdown

What's No-SQL?

Term coined in 2009 for a developer meetup – "Not Only SQL" -> "NoSQL".

Databases that allow you to store and retrieve data in various structures, formats, and models other than tabular relational model.

There's a time and a place for everything

Sometimes a relational store is the right choice

Sometimes a NoSQL store is the right choice

Sometimes you need more than one store for an app -> polyglot persistence

Data Structures

Key-Value Databases

Cosmos DB, Redis Cache, Azure Table

Column Family Stores

Cosmos DB, Cassandra, HBase

Graph Databases

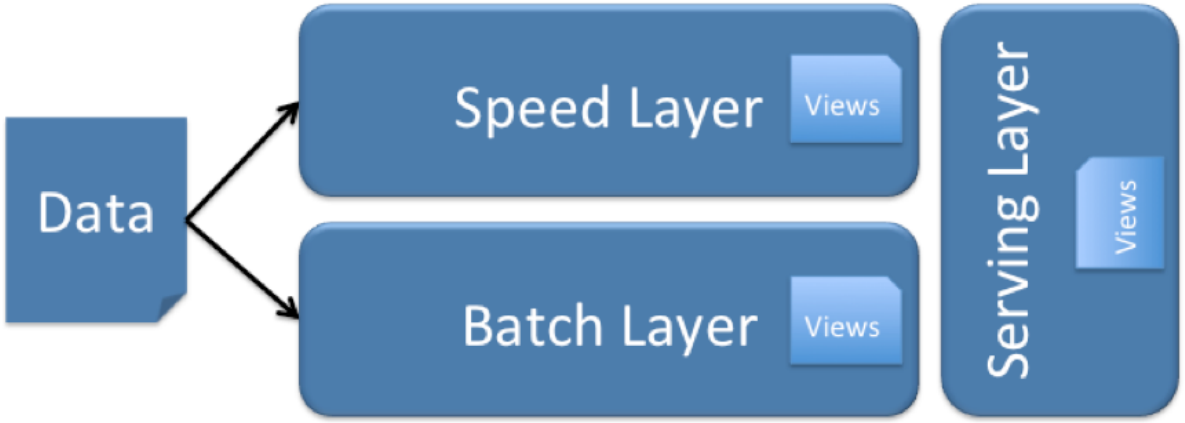
Cosmos DB, Neo4j, Gremlin

Document Databases

Cosmos DB, MongoDB

Lambda (λ) Architecture

Designed to handle Big Data use cases by taking advantage of both batch and stream-processing methods



1. All **data** entering the system is dispatched to both the batch layer and the speed layer for processing.

2. The **batch layer** has two functions:
   I. manage the master dataset (an immutable, append-only set of raw data)
   II. pre-compute the batch views.

3. The **serving layer** indexes the batch views so that they can be queried in low-latency, ad-hoc way.

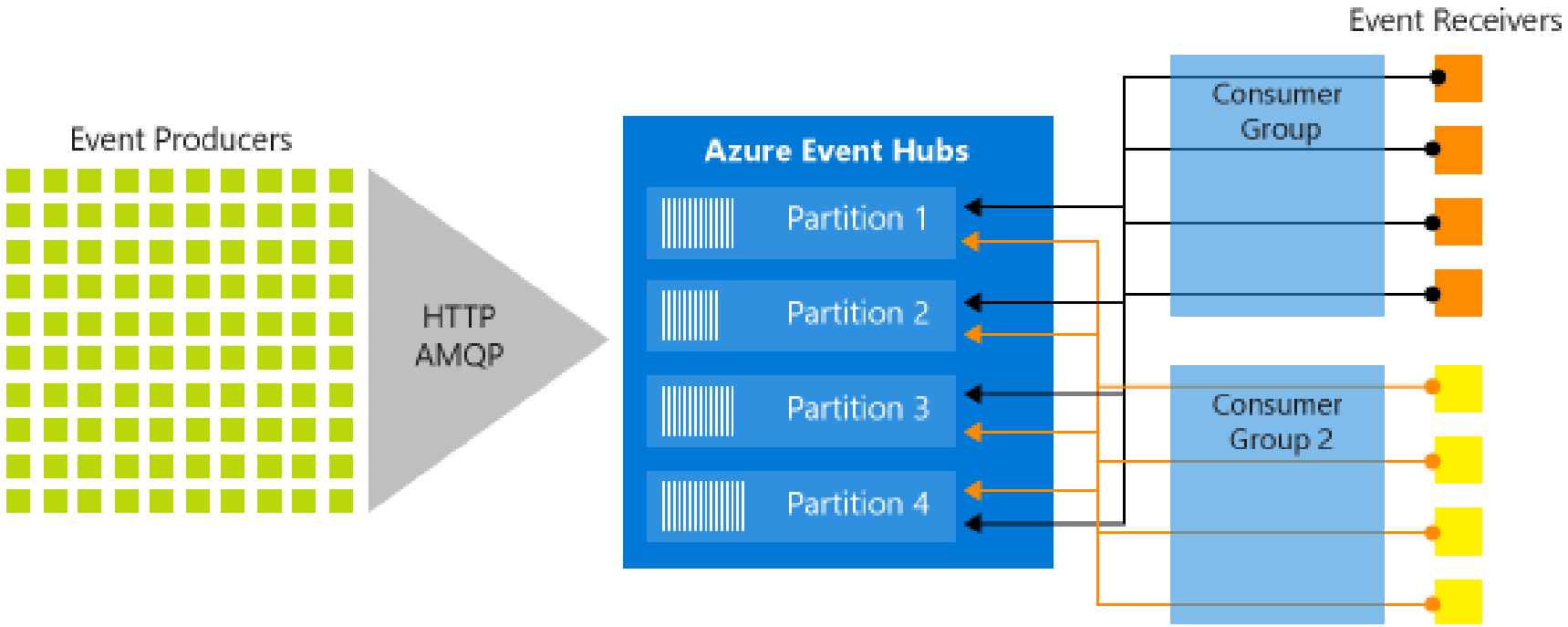
4. The **speed layer** compensates for the high latency of updates to the serving layer and deals with recent data only.

5. Any incoming **query** can be answered by merging results from batch views and real-time

Event Hubs

Event Hubs

Big data streaming platform and event ingestion service capable of receiving and processing millions of events per second.



Event Hubs Capture

Batch on stream

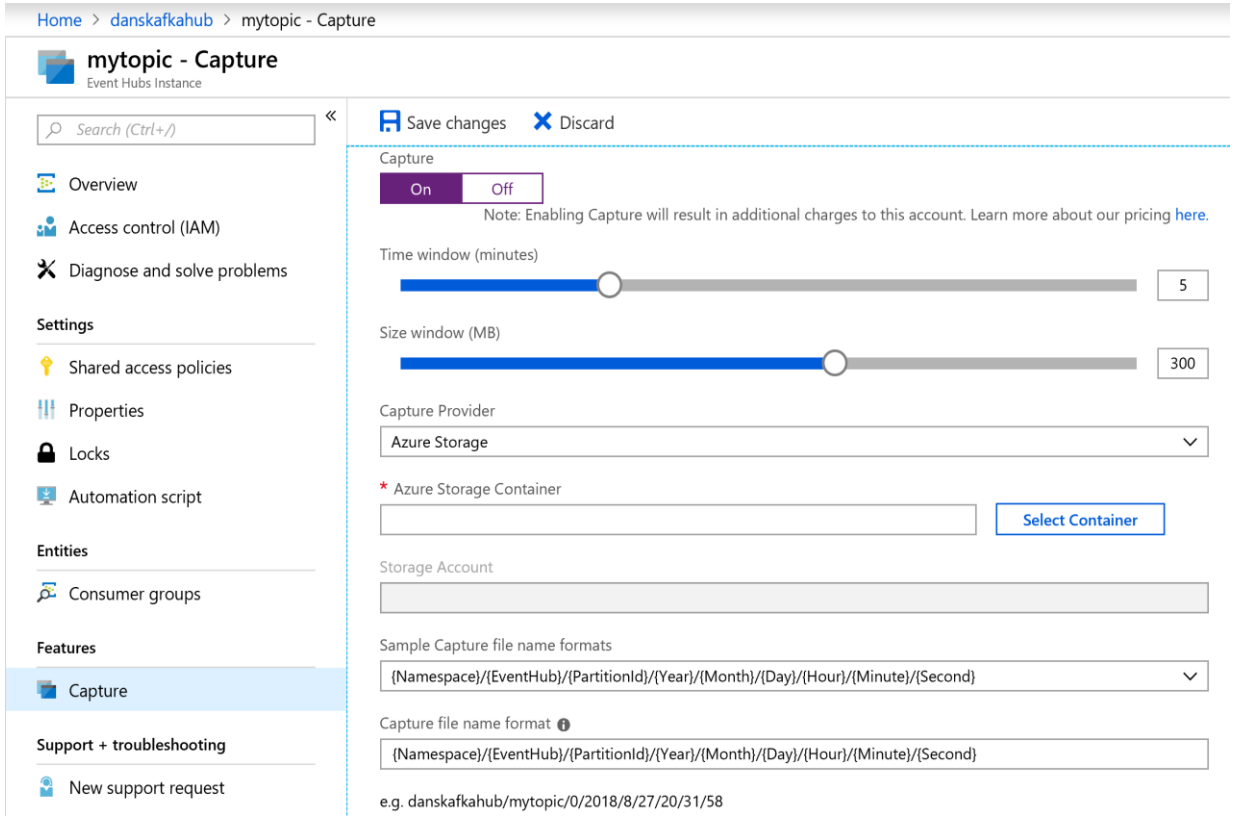
Policy based push to your own storage

Uses Avro format

Raises Event Grid events – connect to Functions, ACI, or whatever you like

Does not impact throughput

Offloads batch processing from your real-time stream



It’s all on

Microsoft Azure

Microsoft