DS-2002: Data Science Systems

A Survey of Data Management Systems

Prof. Jon Tupitza

UNIVERSITY of VIRGINIA

Class Mechanics

A Survey of Data Management Systems

UNIVERSITY of VIRGINIA

Class Mechanics: Course Content

Lectures:

- In-Person
- First Class of the Week
- Demos: live & recorded

Discussion:

- In Class
- Discord

Readings / Videos:

- Articles and Posts
- Supporting tools and videos
- Product Documentation

Labs:

- Hands-On: Dedicated to each concept & technology

Class Mechanics: Course Content

- **Labs**
  - Frequent Hands-On Labs
  - Microsoft Azure Labs (https://labs.azure.com)
  - Completed by Following Week

Class Mechanics: Grading

- **Data Projects**
  - Concrete Examples of Implementation (Two in total)
  - Released long before they're due
  - Detailed Instructions will be in Canvas (GitHub)
  - No Time Limits!

Class Mechanics: Grading

Component	Weight	Frequency
Lectures / Readings / Material		Weekly
Engaged Discussion	5%	Weekly
Quiz	15%	3 or 4
Labs	30%	Weekly
Data Projects	50%	2 projects (25% each)

Class Mechanics: Things You Need to Know

How this course will be conducted: Class Location, Course Materials, Communications

- This class will be taught In-Person
- Thursdays for Hands-on Labs
  - You can do them in groups
  - …or you can do them alone
- The schedule is prone to updates: I'll be sure to keep everyone informed!
- Everything will be updated on GitHub:
  https://github.com/JTupitza-UVA/DS-2002
- Quizzes, Projects and Grades will be released on Canvas
- You can Email me with Questions or to Set-up Office Hours

Learning Objectives

Develop Robust Facility for Handing Data

Acquire a Strong Understanding of SQL and NoSQL Databases

Understand Systems Options in the Public Cloud

Gain Fluency in Tools, Processes, and Services …and How to Choose Among Them

Data Retrieval

Data Shipping

Data Schemas

Data Ingestion

Data Processing

Data Normalization

Data Analysis

Overview *of the* Semester Topics

A Survey of Data Management Systems

UNIVERSITY *of* VIRGINIA

In the Beginning… There Was Mainframe

- Data stored in ISAM files (indexed sequential access method)
  - Flat files having fixed-length fields
- Centralized computing and storage resources
  - "Dumb" terminal clients
- Only affordable to large, wealthy enterprises:
  - Governments
  - Large Banks

Then Came… Client/Server Networks

- Data stored in Relational Database Management Systems (RDBMS):
  - Oracle
  - SQL Server
- Enterprise Data Centers:
  - Company owned & managed
  - Based on commodity hardware
- Some computation and storage resources shared by clients

Which Gave Us… SMP Servers

Relational Database Management running on **Symmetric Multi-Processing Servers**

- Dedicated Database Servers:
  - RDBMS (Relational Database Management Server)
  - Oracle, SQL Server, DB2, etc.
- OLTP (Online Transaction Processing)
  - Characterized by a large volume of transactions each of which affect a small number of rows
  - Online Sales, Bank Deposits & Transfers
  - Highly Normalized Database Schema
- OLAP (Online Analytical Processing)
  - Characterized by a small volume of read transactions each of which affect a large number of rows
  - Periodic Post-hoc Analysis *(What Happened?)*
  - De-Normalized Multi-Dimensional Data Warehouse Schema

But Then… An Explosion of Data – “BIG Data”

A Rapid, Exponential Proliferation of New Devices: The Internet of Things (IoT)

Volume
(Size)

- Explosion in social media, mobile apps, digital sensors, RFID, GPS, and more have caused exponential data growth.

Velocity
(Speed)

- Sources like social networking and sensor signals create data at a tremendous rate; making it a challenge to capture, store, and analyze that data in a timely or economical manner.

Variety
(Structure)

- Traditionally BI has sourced structured data, but now insight must be extracted from unstructured or poly-schematic data like large text blobs, digital media, sensor data, etc.

Veracity
(Quality)

- The anonymity of the WWW, incredible sources like social networking and duplicate systems bring into question the authenticity of the information being generated and collected.

But… How Can We Manage All That Data: Scalability

The Advent of Big Data Drove a Need to Increase Scale to the Petabyte Level: $$$$$

SMP: Symmetric Multi-Processing

Scale Up (Limited)

This

Became

MPP: Massively Parallel Processing

Scale Out (Practically Unlimited)

And Today… Cloud-hosted Databases & Services

Re-centralization of Servers ▪ Based on Commodity HW ▪ Resources Shared by Client

Essential Characteristics

- On-Demand Self-Service
- Broad Network Access
- Resource Pooling
- Rapid Elasticity
- Measured Service

Service Models

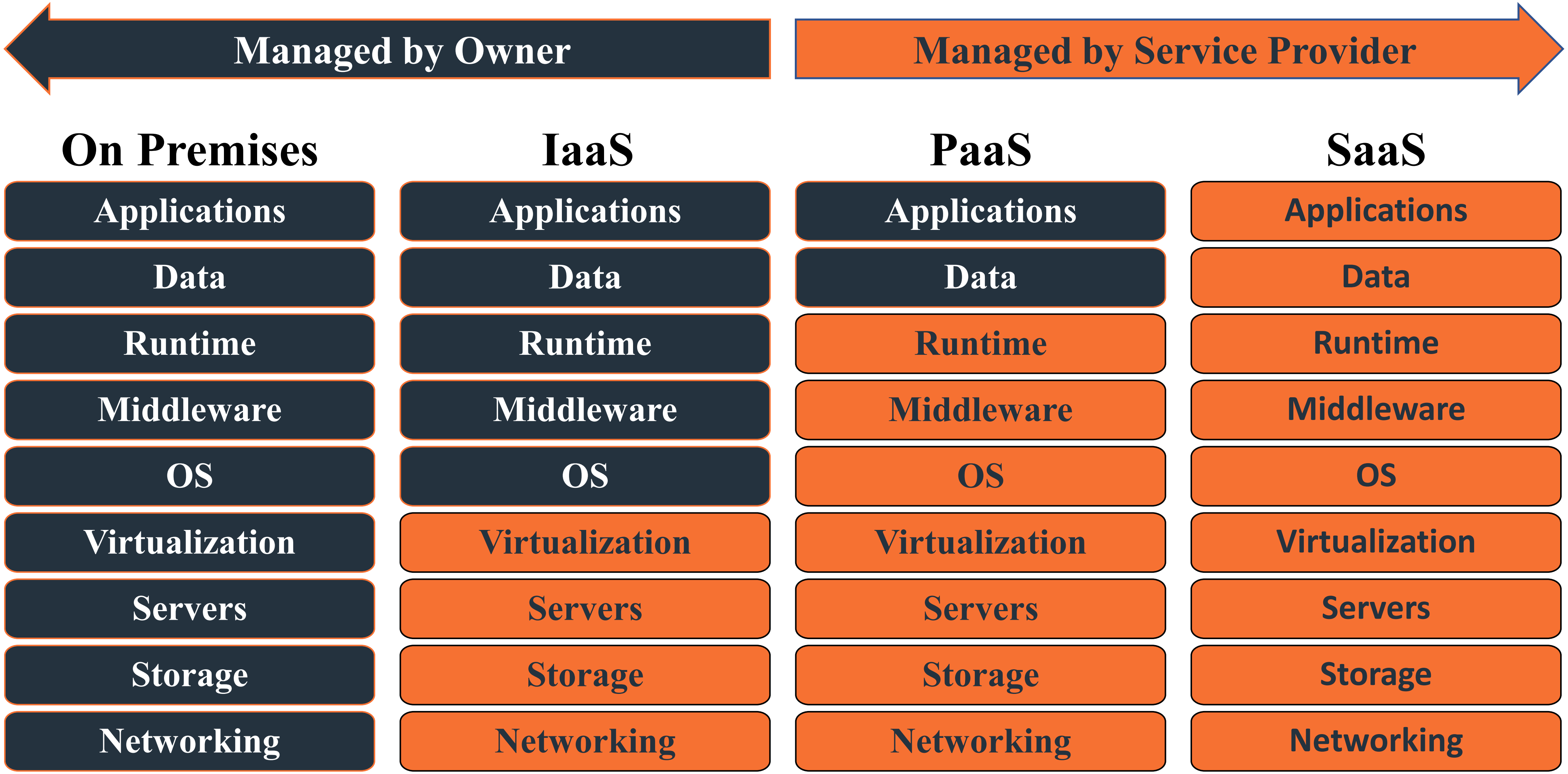
- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

Deployment Models

- Public Cloud
- Private Cloud
- Community Cloud
- Hybrid Cloud

As defined by NIST (National Institute of Standards and Technology)

Service Models: On-Premises vs Cloud-Hosted



Managed by Owner

Managed by Service Provider

On Premises	IaaS	PaaS	SaaS
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
OS	OS	OS	OS
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

Data Management… In the Cloud

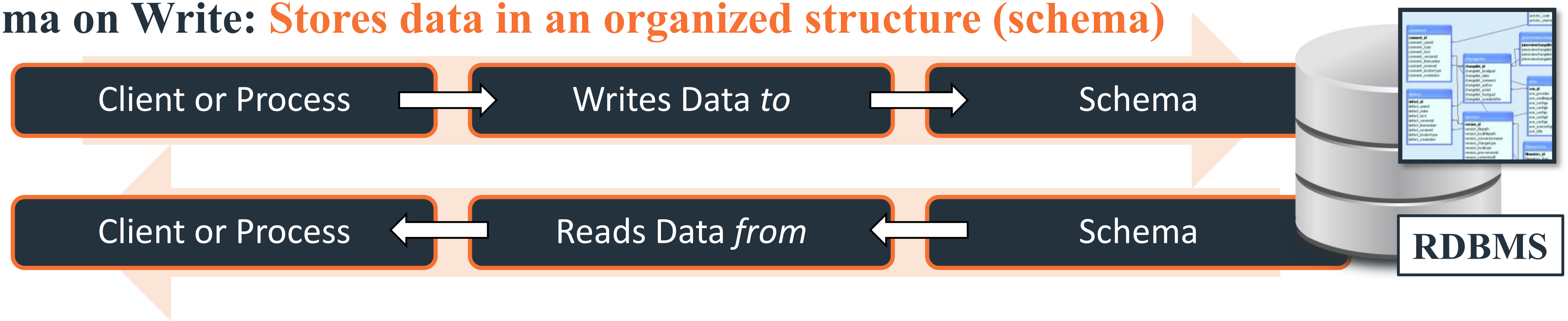
Economies of Scale Afforded by the Cloud Have Enabled Massive Storage & Compute

- File-based Storage: Blobs, Queues, Tables, Data Lakes
- Relational Database Management Systems:
  - Online Transaction Processing (OLTP) systems:
    - **IaaS**: Cloud-hosted VMs Running RDBMS Software & Databases
    - **PaaS:** (Database as a Service) Single Databases, Multi-Database Pools, Managed Instance
  - Online Analytical Processing (OLAP) systems:
    - **PaaS:** Massively Parallel Processing (MPP) Data Warehouse
- NoSQL Database Management Systems:
  - Semi-Structured or Poly-Schematic data
  - Massively Parallel Processing
    - Data partitioned and replicated across many server machines (nodes)
    - Data typically stored in file-based format (e.g., JavaScript Object Notation (JSON)
- Data Lakehouse: Data Warehouse Schema over File-based Storage (Data Lake)

Paradigms: Data Storage and Retrieval

Schema on Write versus Schema on Read

Schema on Write: Stores data in an organized structure (schema)

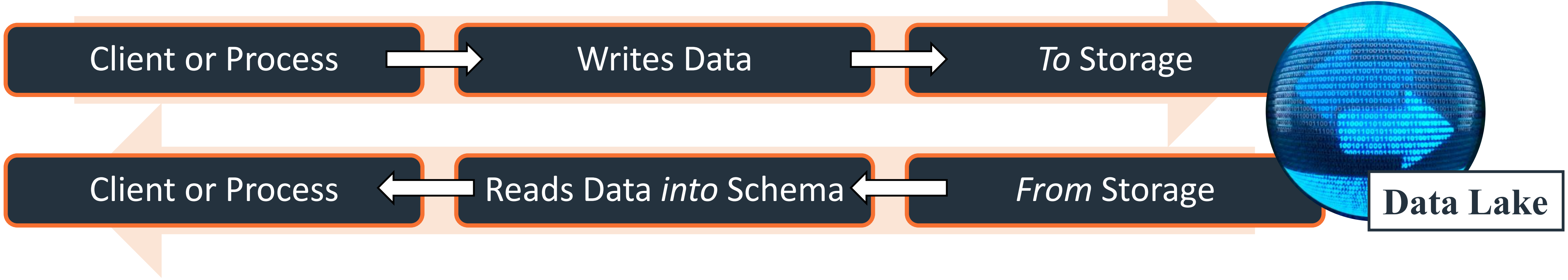


Client or Process → Writes Data *to* → Schema

Client or Process ← Reads Data *from* ← Schema

RDBMS

Schema on Read: Applies schema only when read, data stored in its original format



Client or Process → Writes Data → *To* Storage

Client or Process ← Reads Data *into* Schema ← *From* Storage

Data Lake

NoSQL: Not Only SQL

Key-value stores

- The simplest NoSQL database; based on “dictionaries” or “maps”.
- Items are stored in associative arrays; pairing a name (or "key"), with a value.
- **Riak, FoundationDB, and Redis**

Column stores

- Combines a key, value and timestamp for each item.
- Optimized for large datasets by storing columns of data together, rather than in rows.
- **Cassandra, BigTable and HBase**

Document databases

- Pairs keys with complex data structures (documents) using XML, JSON or BSON.
- Documents may contain key-value pairs, key-array pairs, and nested documents.
- **MongoDB, MarkLogic, and Apache CouchDB**

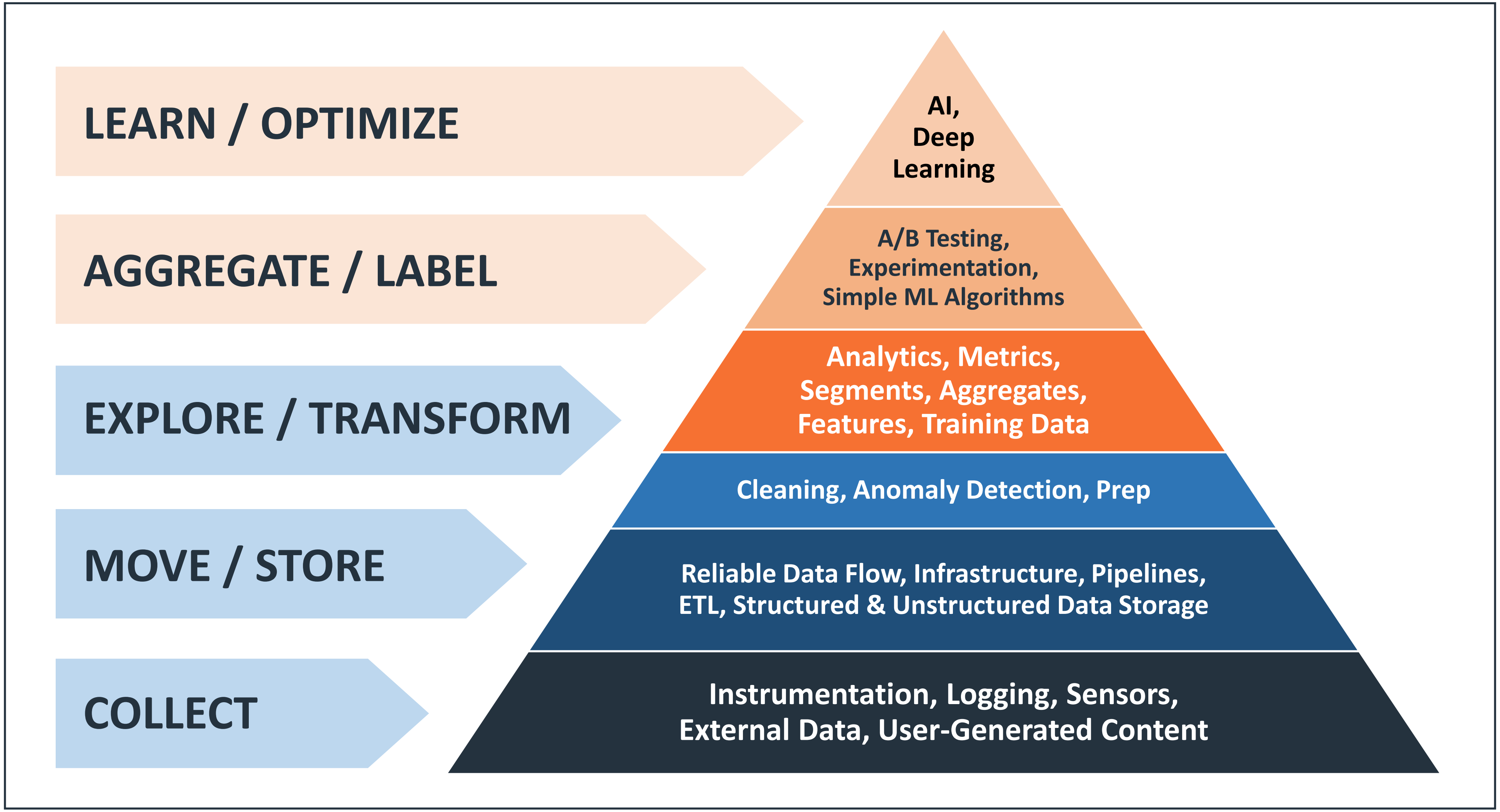
Graph stores

- Stores interrelated networks of data such as social connections, or network topologies.
- Optimized for interconnected data elements with an undetermined number of relations.
- **AllegroGraph, Neo4J and HyperGraphDB**.

Onward to Data Science

A Brief Overview of How Data Systems are Involved

The Data Science: Hierarchy of Needs



LEARN / OPTIMIZE

AGGREGATE / LABEL

EXPLORE / TRANSFORM

MOVE / STORE

COLLECT

AI,
Deep
Learning

A/B Testing,
Experimentation,
Simple ML Algorithms

Analytics, Metrics,
Segments, Aggregates,
Features, Training Data

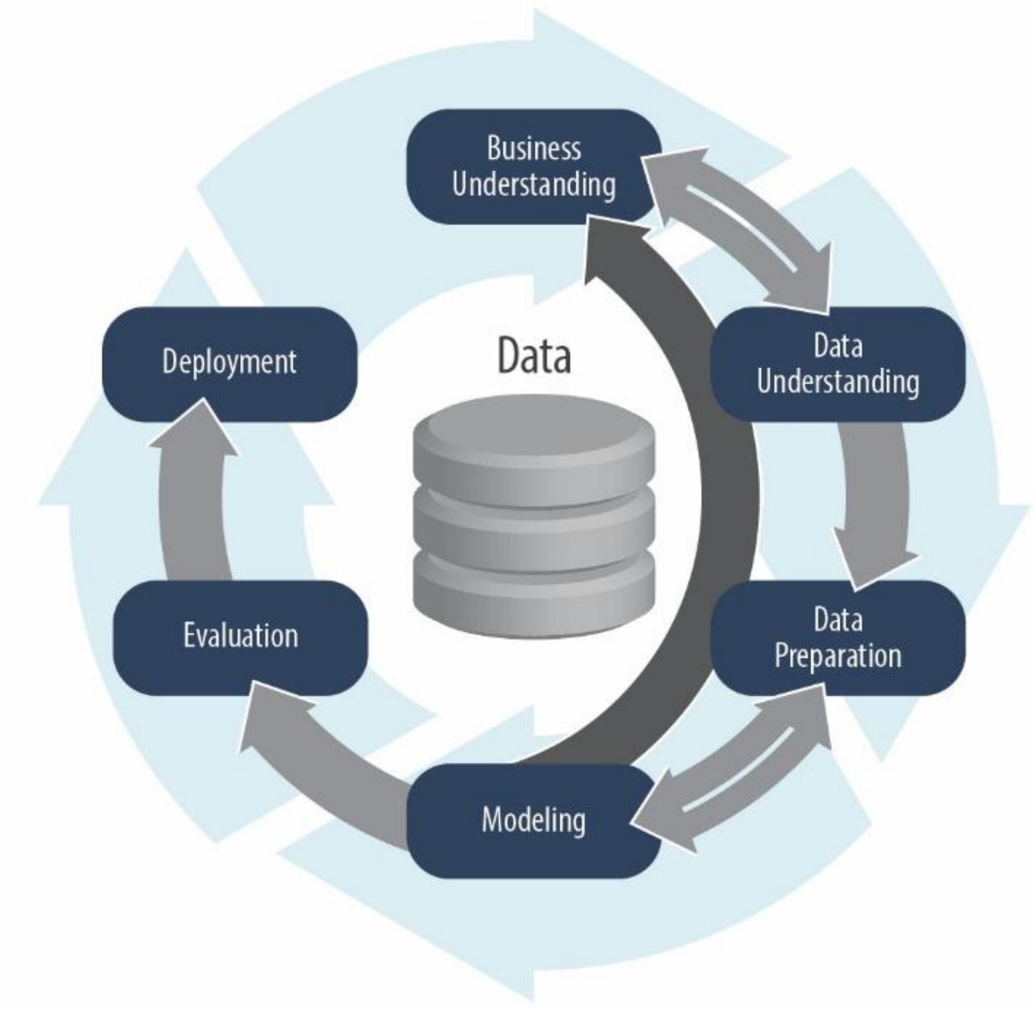
Cleaning, Anomaly Detection, Prep

Reliable Data Flow, Infrastructure, Pipelines,
ETL, Structured & Unstructured Data Storage

Instrumentation, Logging, Sensors,
External Data, User-Generated Content

CRISP-DM: Cross-Industry Standard Process-Data Mining

First Introduced in 1996!



Business Understanding	• Identify the Problem Domain • Identify the Solution Scenario
Data Understanding	• Load and Explore Data • Identify Influential Features
Data Preparation	• Remove Duplicates & Nulls • Impute Missing Values • Select & Engineer Features
Modeling	• Train Models Using a Variety of Algorithms • Tune Hyper-parameters
Evaluation	• Test Models' Performance & Predictive Power • Cross-Validate to Appraise Goodness-of-Fit • Select Most Effective Model for Deployment
Deployment	• Publish Models On-premises or in the Cloud • Consume Models Visually & Programmatically

Data Engineering… for Data Science

Frequently, Data Must Be Moved from Sources to a Database and/or Data Lake

Extract

- This is the step where sensors wait for upstream data sources to land. Once available, we transport the data from their source locations to further transformations.

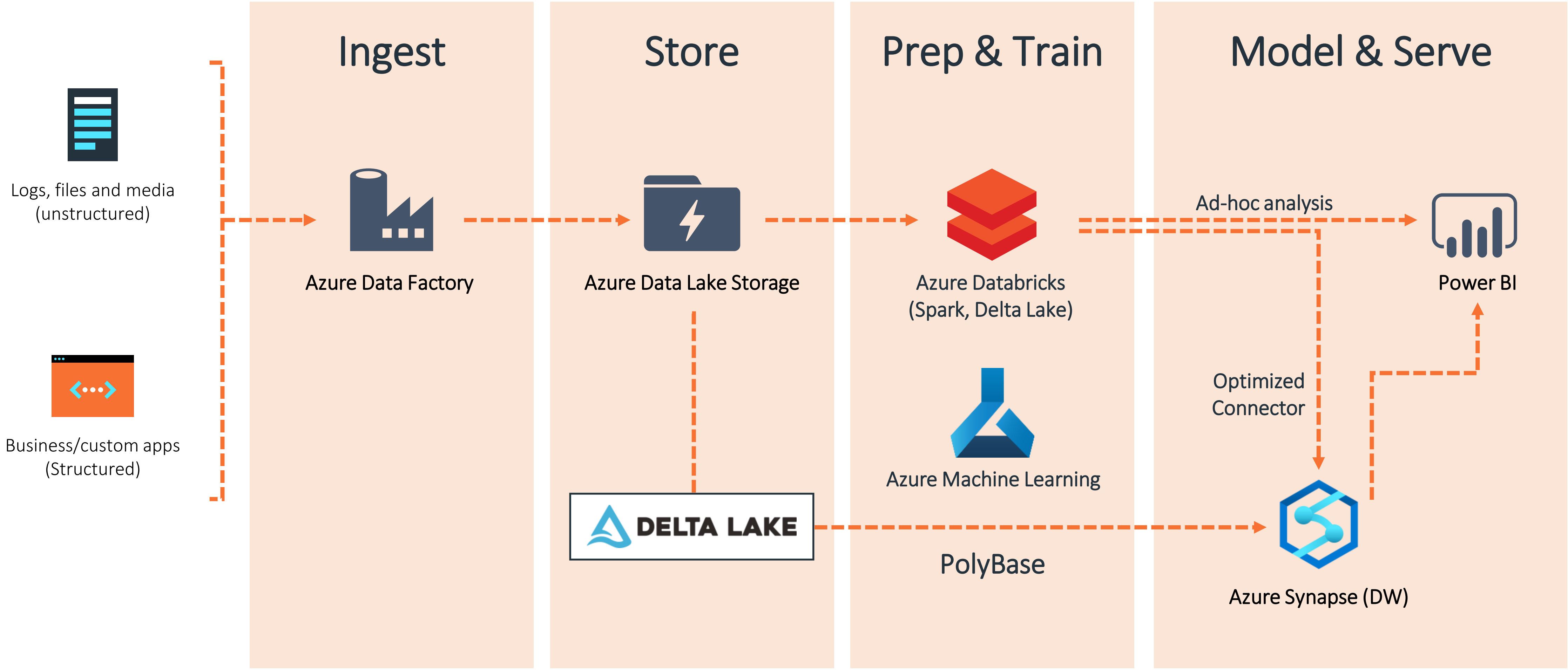
Transform

- The heart of any ETL job: apply business logic, perform actions such as filtering, grouping, and aggregation to translate raw data into analysis-ready datasets.

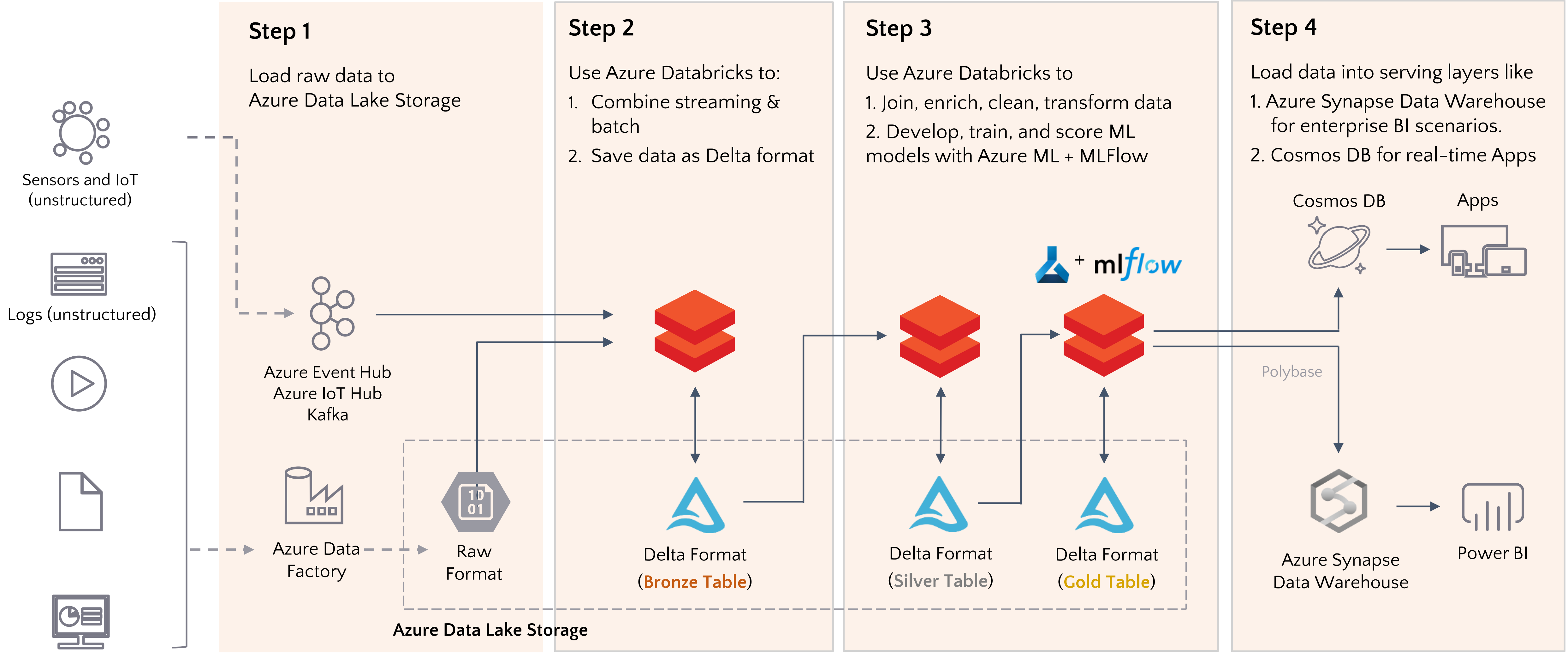
Load

- Load the processed data and transport to a final destination. Can now be consumed directly by end-users or treated as yet another upstream dependency.

Data Engineering… for Data Science



Design Pattern: Modern Data Warehousing



Step 1

Load raw data to Azure Data Lake Storage

Step 2

Use Azure Databricks to:

1. Combine streaming & batch
2. Save data as Delta format

Step 3

Use Azure Databricks to

1. Join, enrich, clean, transform data
2. Develop, train, and score ML models with Azure ML + MLFlow

Step 4

Load data into serving layers like

1. Azure Synapse Data Warehouse for enterprise BI scenarios.
2. Cosmos DB for real-time Apps

Q & A

A Survey of Data Management Systems