



DS-2002: Data Science Systems

A Survey of Data Management Systems

Prof. Jon Tupitza

Class Mechanics

A Survey of Data Management Systems



Who is... Teaching this Course?

- **Jon Tupitza @ Microsoft:** Sr. Cloud Solution Architect for Data, AI & Analytics
 - Lead Enterprise Data Science Program (EDSP) for Retail & Consumer Goods OU
 - Mentor & Courseware developer for EDSP Mentoring Program – Certification Proctor
- MS Data Analytics – 2018
- More Industry Certifications than I can remember – see LinkedIn (MCP since 1999)
- Microsoft Guy: 25+ yrs. Focusing on Microsoft technologies
 - Software Engineering & Architecture
 - Data Platform Architect – Data Engineer – Business Intelligence – Data Science
- PubSec/National Security SME: 22 yrs. Supporting DoD (Army, DARPA), IC (NGA, FBI)
- Past @ Buchanan-Edwards: CTO/Solution Architect – Practice Lead: Predictive Analytics
- Past @ Microsoft Consulting: Sr. Consultant specializing in SQL Server & Business Intelligence
- Past @ Avanade: Principal Solution Developer (.NET)
- Past @ Microsoft Partners: Data Scientist, Data Warehouse Architect, Software Architect



Who am I ?

- **Love to Teach:** Born to be a Professor... *no kidding – it's been my life's goal*
 - Wanted to be interesting (and financially secure) so I've done lots of other professional stuff first
- Philadelphia area (Valley Forge) native
- Recent Transplant from DC Metro area – Alexandria, VA (*Mt. Vernon*)
- 2 Kids and 1 Wife... *of 32 yrs.*
 - Wife: CFP, MS Financial Planning – VP of Wealth Mgmt. & Trust
 - Daughter: UVA Alum – PhD Candidate @ LSU (Oceanography & Coastal Sciences)
 - Son: VA Tech Alum – Works in the Clean Energy sector
- Transitioned from an Earlier Career in Advertising... *Art Director & Agency Computer Wiz (Mac)*
- BFA Degree (Fine Art)... *from a long line of artists and craftsmen*
- 41+ yrs. studying & 28 yrs. teaching Okinawan & Japanese Martial Arts
 - 9th Dan – Hanshi – Shoshinkan-ha Shorinken Karate/Kobudo & Itto-Ryu Kenjutsu
 - 6th Dan – Kyoshi – Classical Judo

Class Mechanics: Course Content



Lectures:

- In-Person
- First Class of the Week (Monday)



Discussion:

- Tuesdays as a part of Lecture Day



Readings / Videos:

- Articles and Posts
- Supporting tools and videos
- ~Hours per Week Prep for Tuesdays Class



Labs:

- Hands-On, Virtual & Thursday.
- I may Provide a Zoom invite.

Class Mechanics: Course Content



■ Labs

- Frequent Hands-On Labs
- Microsoft Azure Labs (<https://labs.azure.com>)
- Begin on Thursdays and Complete 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
Student Survey		First Week
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 we may be on Zoom for Hands-on Labs
 - Sometimes we'll do them in groups
 - ...And sometimes you'll 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-2002-002>
- Quizzes, Projects and Grades will be released on Canvas
- You can DM me on Discord with Questions or to Set-up Office Hours

Learning Objectives

Develop Robust Facility for Handling 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

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

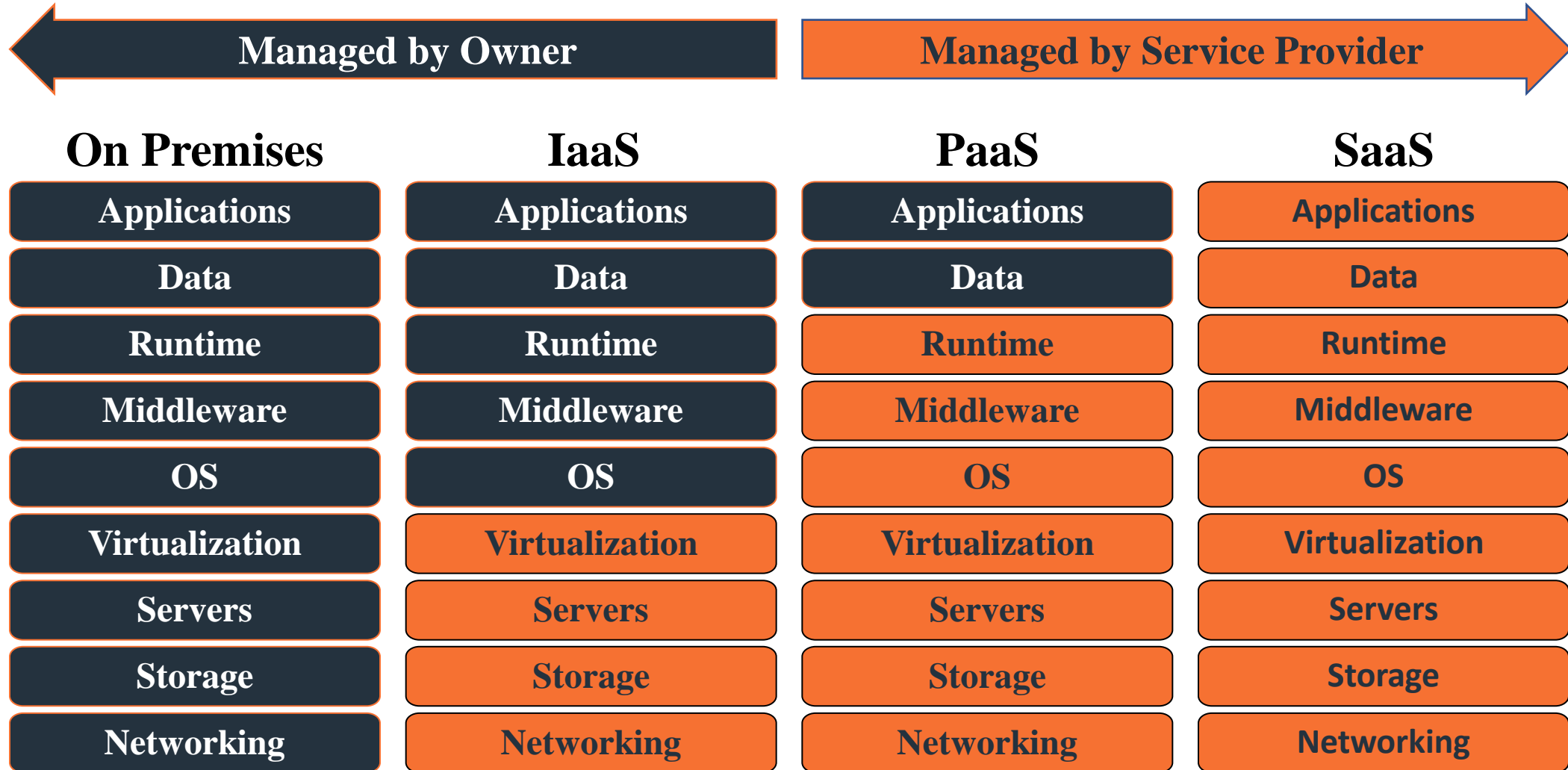
Private Cloud

Community Cloud

Hybrid Cloud



Service Models: On-Premises vs Cloud-Hosted





Data Management... In the Cloud

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

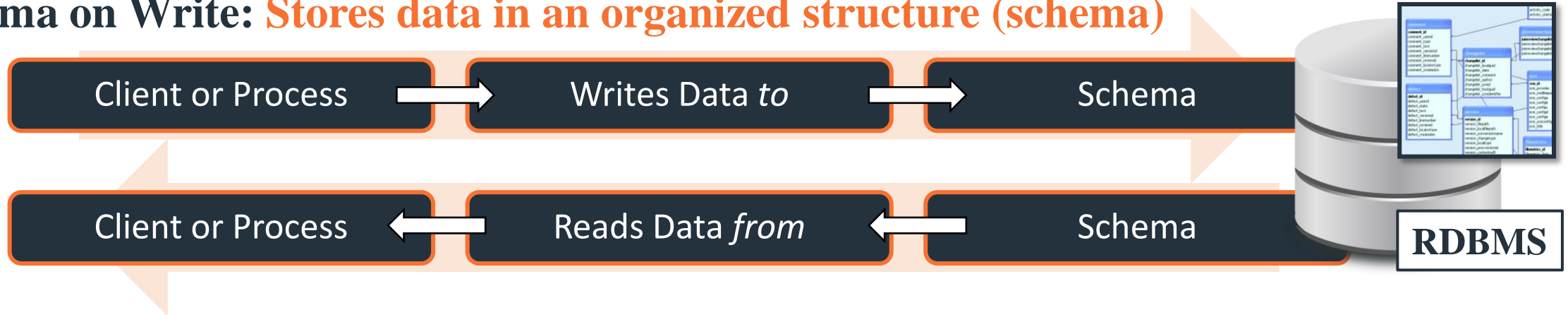
- File-based Storage: **Blobs, Queues, Tables, Data Lakes**
- Relational Database Management Systems:
 - Online Transaction Processing (OLTP) systems:
 - **IaaS**: Cloud-hosted VMs Running RDBMS Software & Databases
 - **PaaS**: Managed RDBMS Instance
 - **SaaS**: (Database as a Service) – Single Databases or Multi-Database Pools
 - Online Analytical Processing (OLAP) systems:
 - 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))



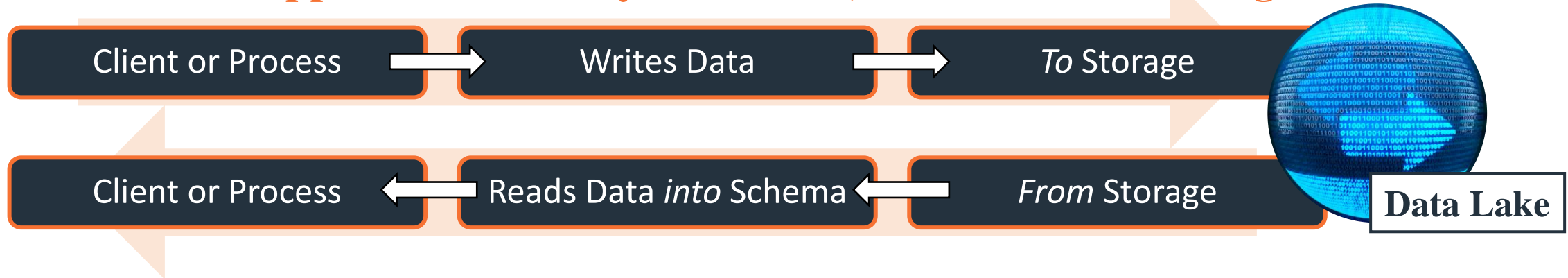
Paradigms: Data Storage and Retrieval

Schema on Write versus Schema on Read

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



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





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.**



SQL versus NoSQL Databases

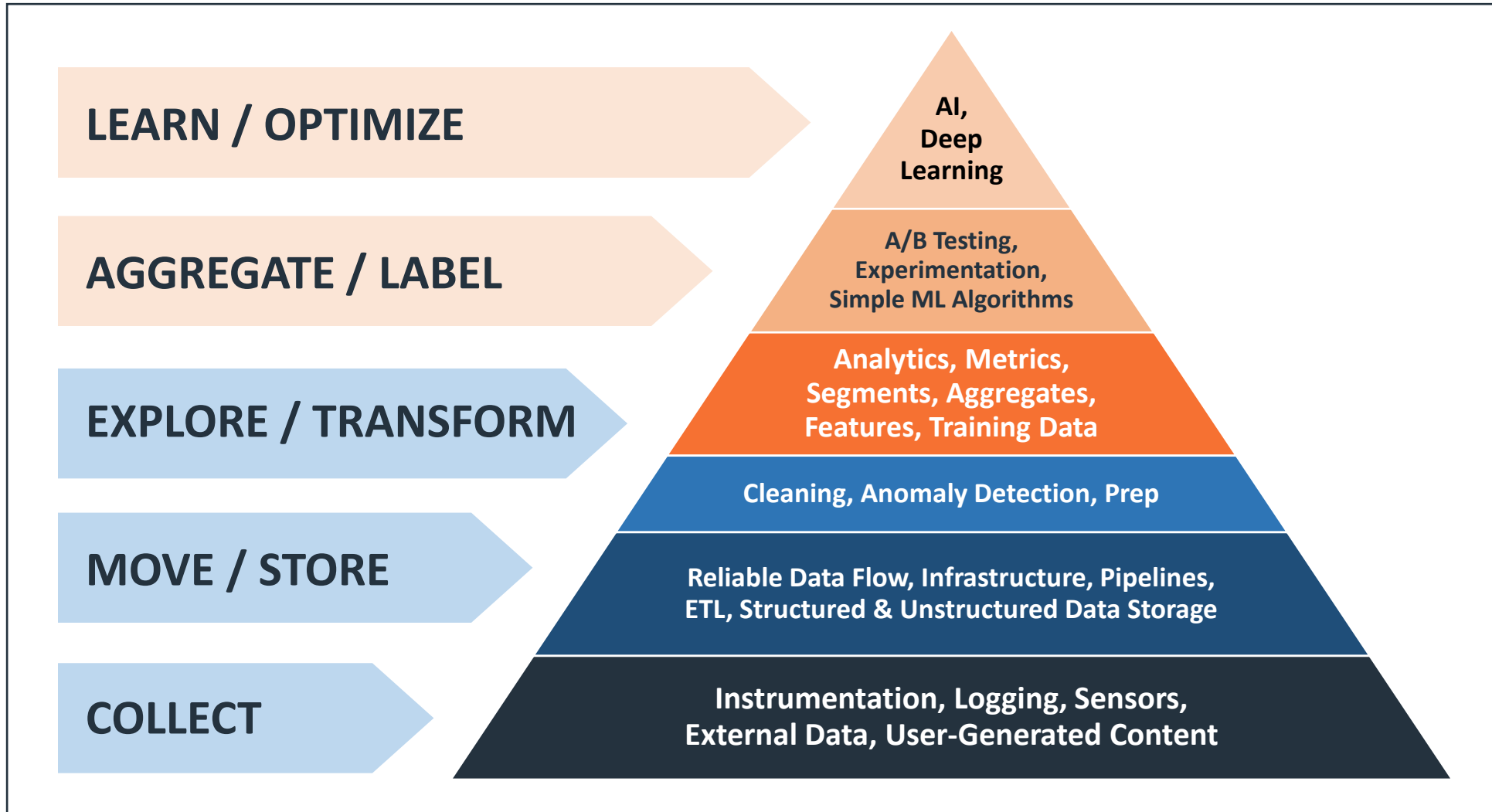
	SQL Databases	NoSQL Databases
Types	<u>One</u> : Relational Database	<u>Many</u> : Key-value, document, column, and graph databases
Data Storage	Records are stored as rows in tables that represent entities. Entity relationships are modeled by joining tables.	Records may be stored as “documents” using XML or JSON. Entity relationships are modeled by nesting them hierarchically.
Schemas	<u>Static</u> : Structure & data types are fixed at design time. Adding new elements requires schema design changes.	<u>Dynamic</u> : New elements can be added at runtime. <u>Poly-schematic</u> : Dissimilar data can be stored together as necessary.
Scaling	<u>Vertically</u> : A single server must be made increasingly powerful to cope with increasing demand.	<u>Horizontally</u> : New commodity servers and storage are added to an array. Data is automatically distributed across all servers.
Transactions	Full transactional consistency (ACID)	Single element (document) only.
Manipulation	Using platform-specific languages (SQL)	Using OSS API's, low-level lang., JavaScript
Consistency	Supports strong consistency	Per-product: Most are eventually consistent.

Onward to Data Science

A Brief Overview of How Data Systems are Involved

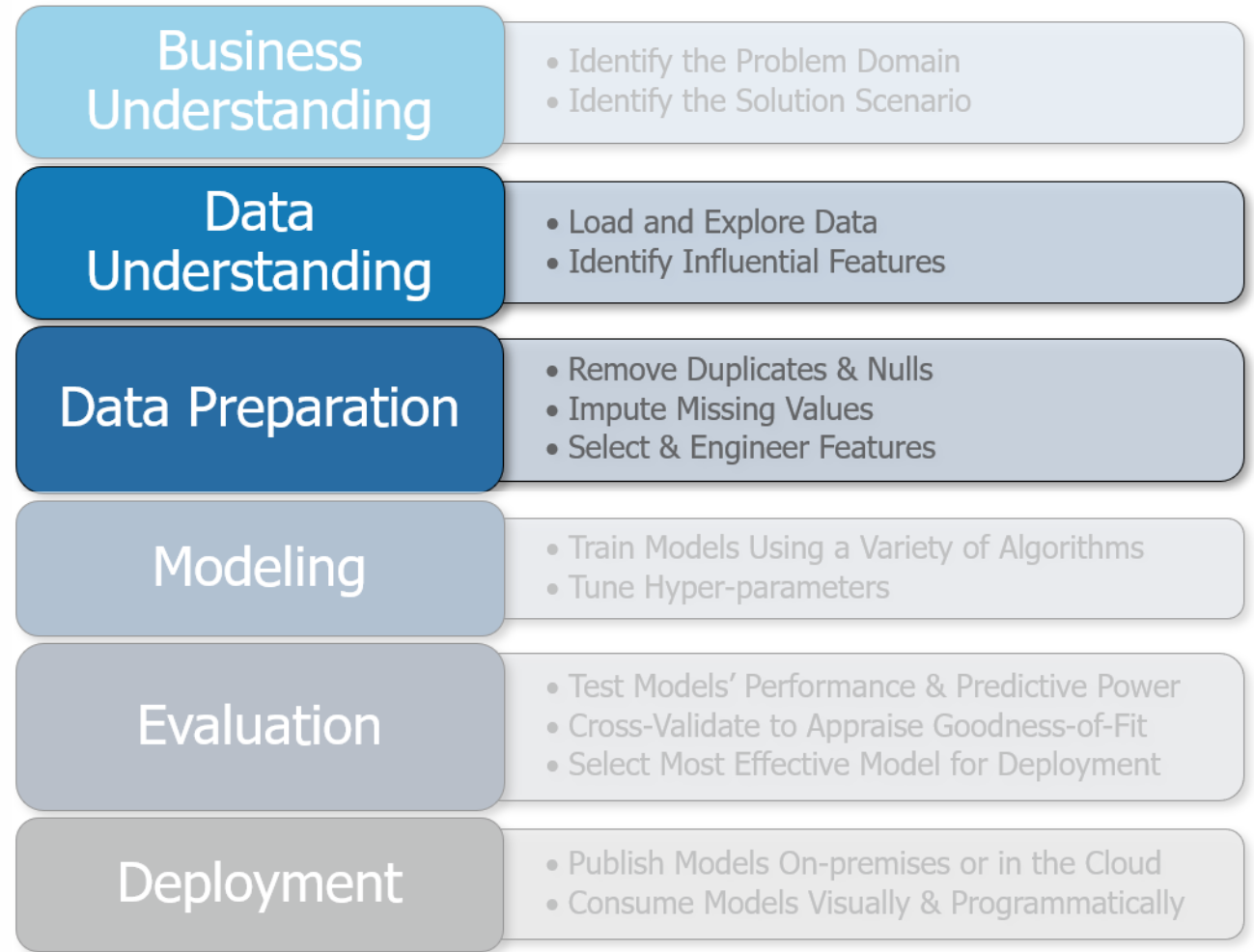
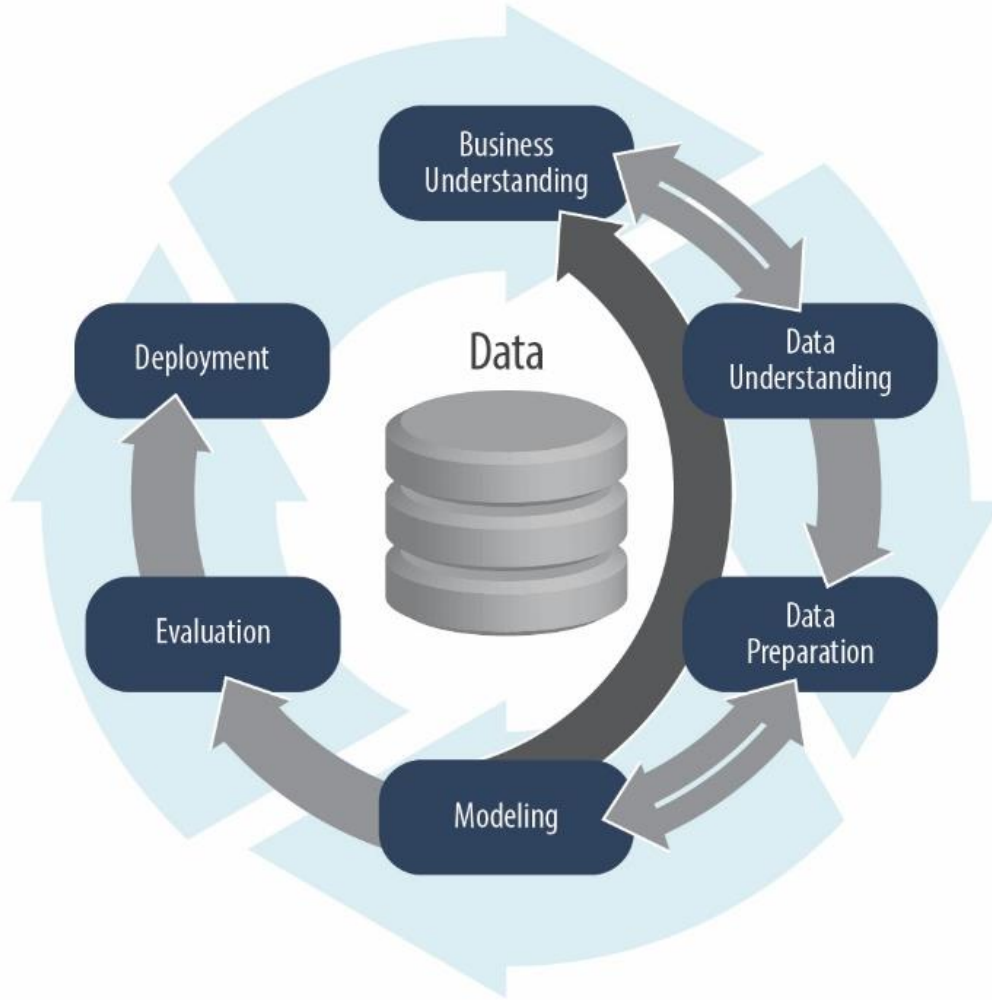


The Data Science: **Hierarchy of Needs**



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

First Introduced in 1996!





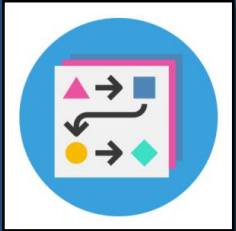
Data Engineering... for Data Science

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

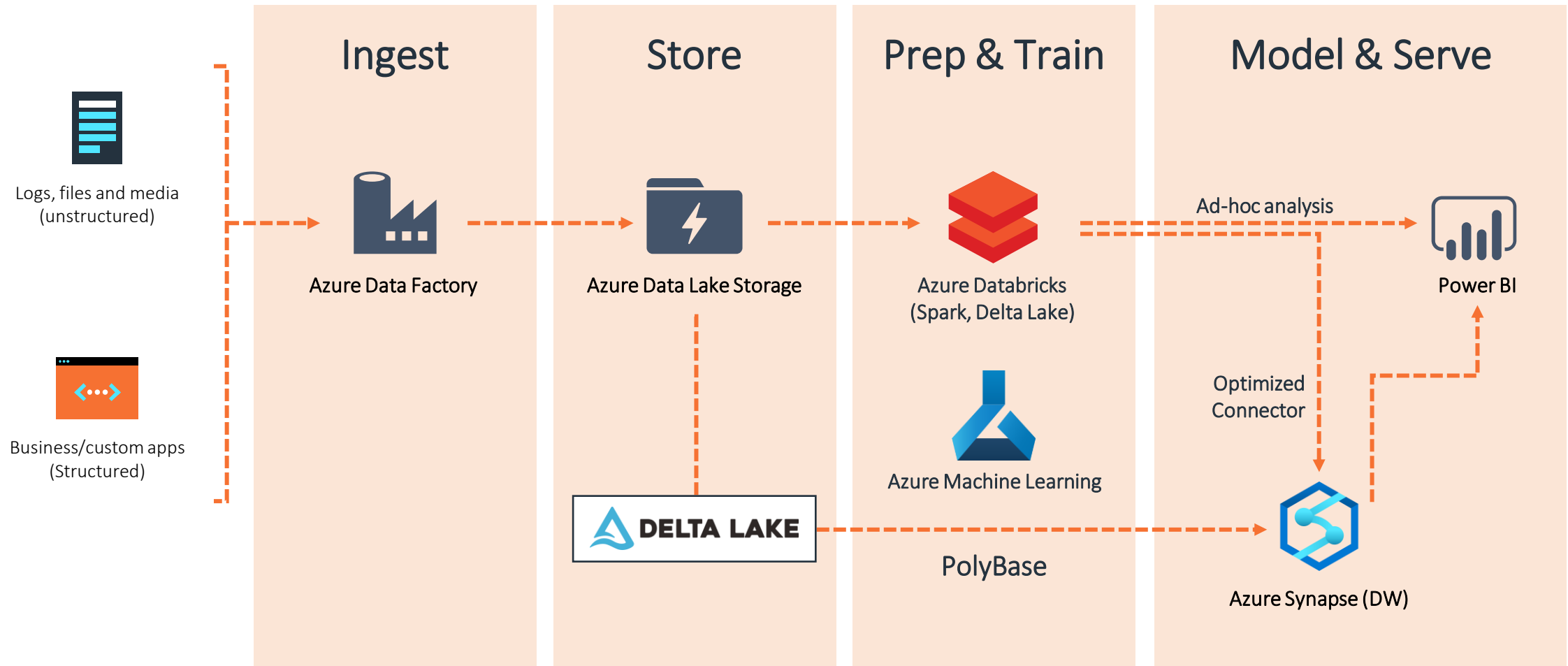
- The heart of any ETL job: apply business logic, perform actions such as filtering, grouping, and aggregation to translate raw data into analysis-ready datasets.



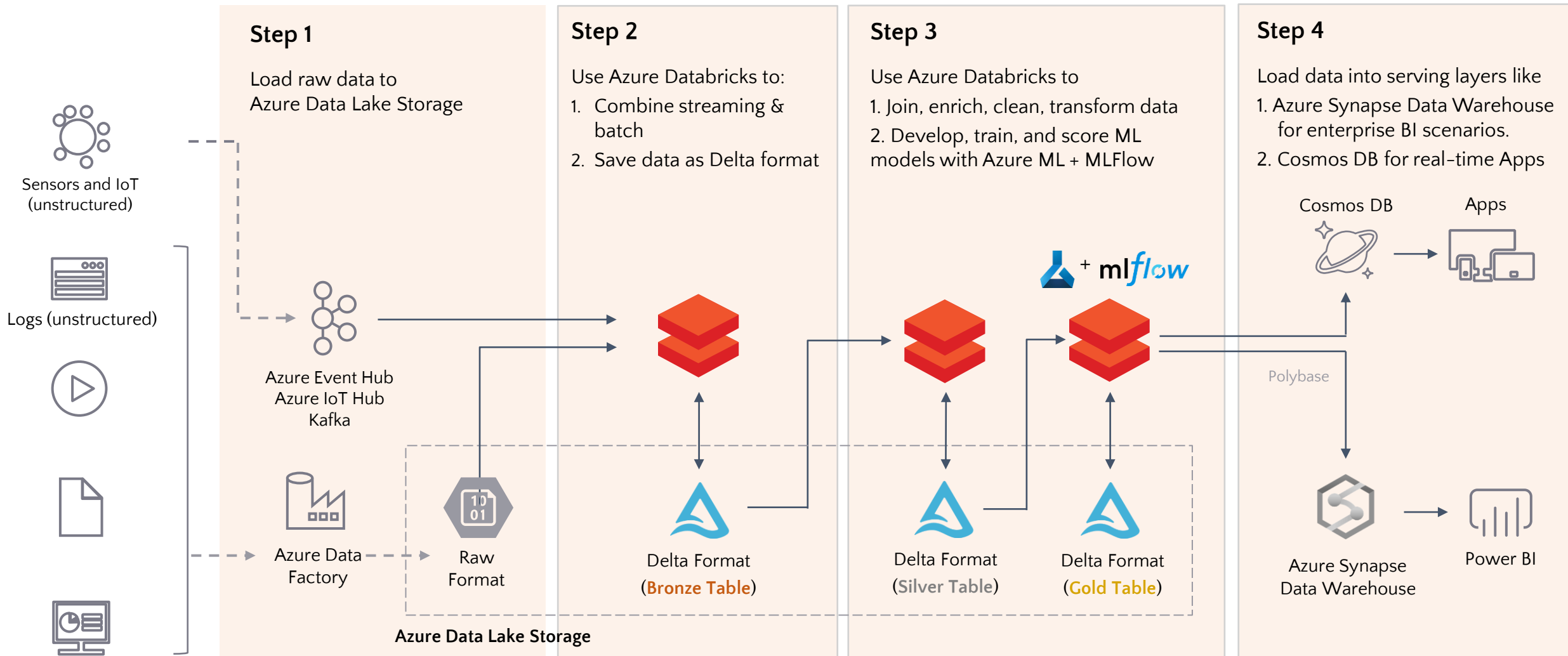
Load

- 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



Q & A

A Survey of Data Management Systems

