Microsoft | Microsoft Fabric


#FABCONSQLCON2026
FABCON
Microsoft Fabric
COMMUNITY CONFERENCE
SQLCON
Microsoft SQL
COMMUNITY CONFERENCE
ATLANTA MARCH 16 - 20, 2026

Building Modern Data Architectures with Microsoft Fabric

James Serra

Data & AI Solution Engineer

Microsoft, Global Technology Sales Japan

jamesserra3@gmail.com

Blog: JamesSerra.com

About Me

- Microsoft, Data & AI Solution Architect in Microsoft Federal Civilian
- At Microsoft for most of the last eleven years as a Data & AI Architect, with a brief stop at EY
- In IT for 40 years, worked on many BI and DW projects
- Worked as desktop/web/database developer, DBA, BI and DW architect and developer, MDM architect, PDW/APS developer
- Been perm employee, contractor, consultant, business owner
- Presenter at PASS Summit, SQLBits, Data Summit, SQLDay, Enterprise Data World conference, Big Data Conference Europe, SQL Saturdays, Informatica World
- Blog at JamesSerra.com
- Former SQL Server MVP
- Author of the book “Deciphering Data Architectures: Choosing Between a Modern Data Warehouse, Data Fabric, Data Lakehouse, and Data Mesh”

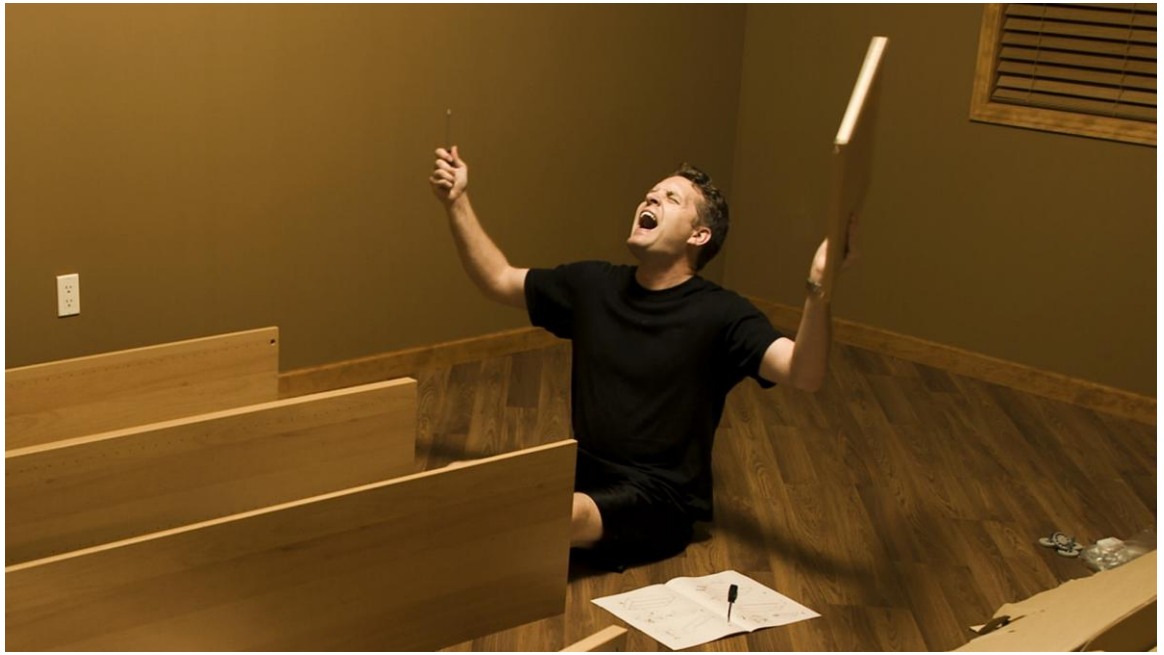
Agenda

- Relational Data Warehouse
- Data Lake
- Four architectures:
  - Modern Data Warehouse
  - Data Fabric
  - Data Lakehouse
  - Data Mesh
- Data architectures on Microsoft Fabric

Note: These are James Serra's opinions and not that of Microsoft!

Architecture looked easy in the diagram…

Then I opened the box and found 147 parts and zero clarity



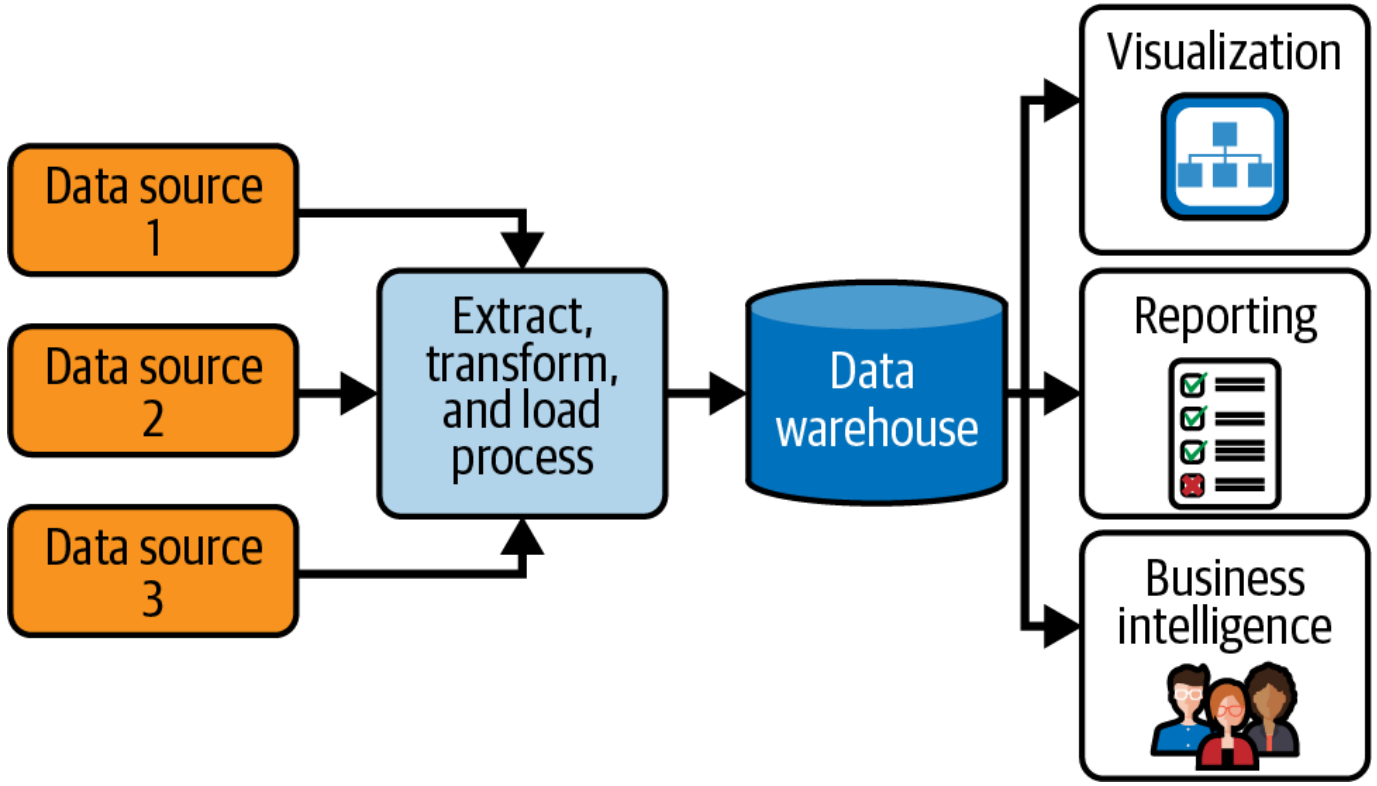
Let’s prevent that from happening…

Relational Data Warehouse (RDW)

What is a Relational Data Warehouse?

(or, why do we need a copy of the source data?)

A relational data warehouse is where you store data from multiple data sources into relational storage to be used for historical and trend analysis reporting **to make better business decisions** by getting greater insights into your company. It acts as a central repository for many subject areas and contains the "single version of truth". It is NOT to be used for OLTP (Online transaction processing) applications.



Why use a Relational Data Warehouse?

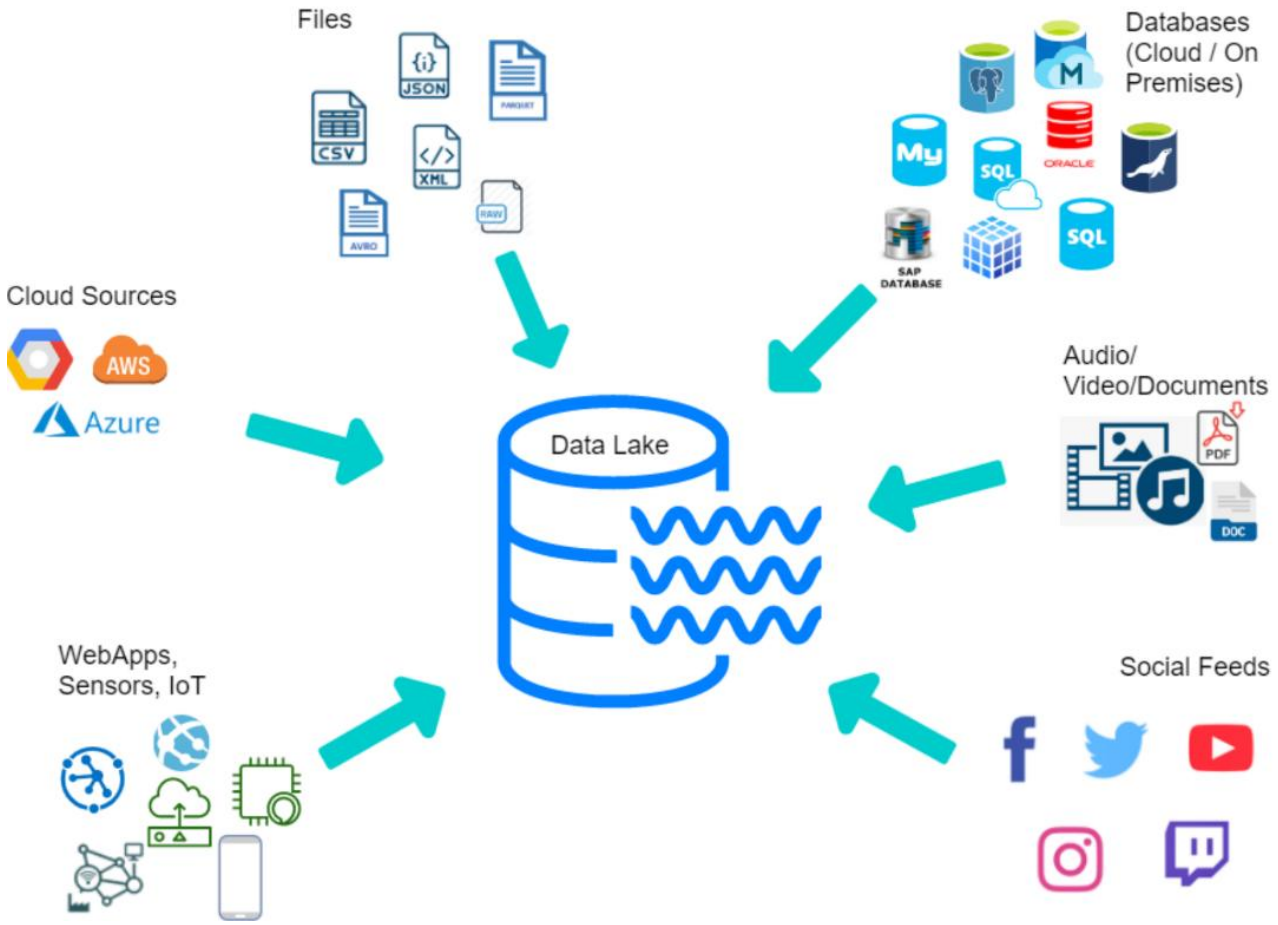
- **Single version of the truth**
- **Reduce stress on the production system**
- Optimized for read access
- **Restructure and rename tables and fields**
- Protection against application upgrades
- Improve data quality by plugging holes in source systems
- **No IT involvement needed to create reports (self-service BI)**

Azure SQL Database

Data Lake

What is a Data Lake?

A schema-on-read storage repository that holds a vast amount of raw data in its native format until it is needed. It's a glorified file folder – just storage, where a RDW is both storage and compute.



OneLake

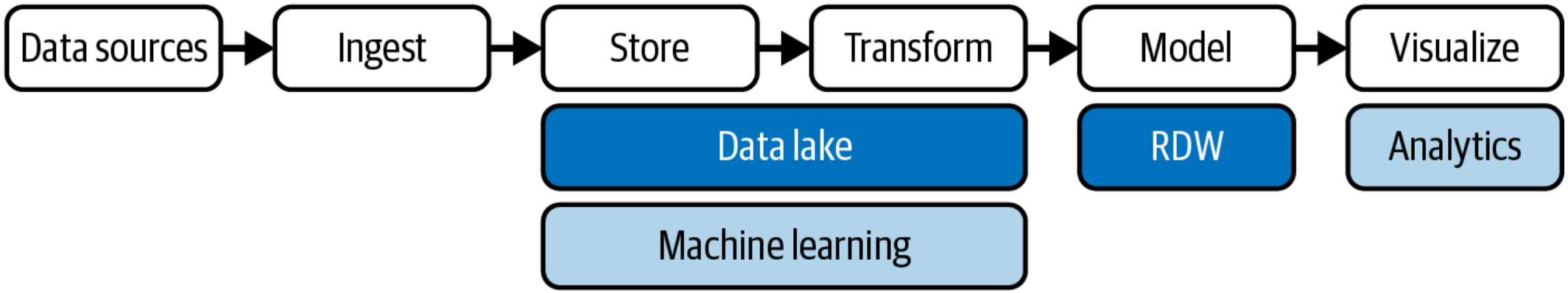
Azure Data Lake Storage Gen2

Why use a data lake?

- Store data with no modeling: Schema-on-read (RDW is schema-on-write)
  - Allows for quick user access to data for power users/data scientists (allowing for faster ROI)
  - Provides for data exploration to see if data valuable before writing ETL and schema for relational database, or use for one-time report/query
- **Frees up expensive enterprise data warehouse (EDW) resources for queries instead of using EDW resources for transformations. Removes need for EDW maintenance window**
- Extreme performance for transformations by having multiple compute options each accessing different folders containing data
- **Stockpiling data cheaply**

Modern Data Warehouse (MDW)

MDW high-level architecture



Data Lake with Data warehouse use cases

Data Lake

Staging & preparation

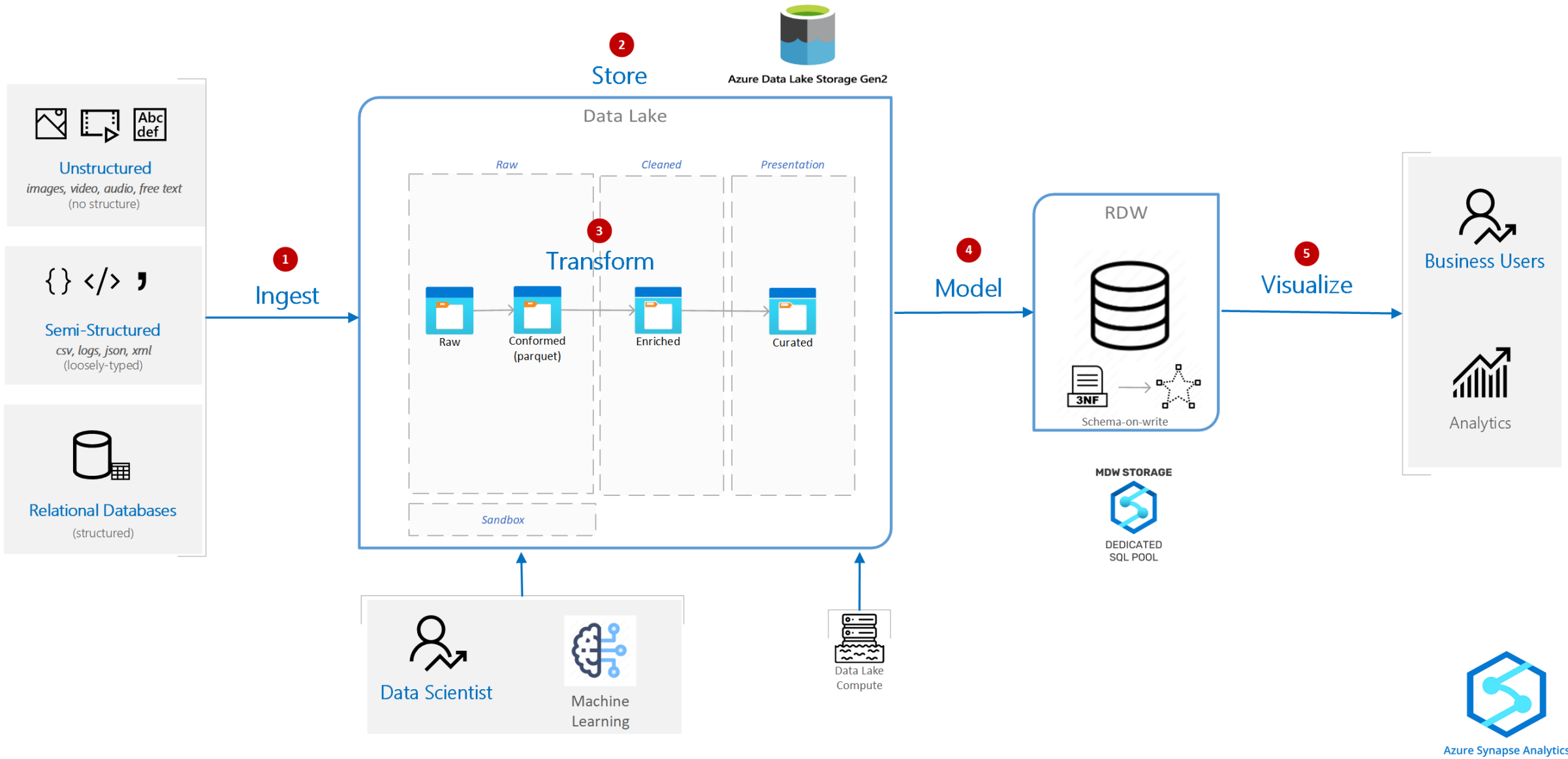
- Data scientists/Power users
- Batch processing
- Data refinement/cleaning
- ETL workloads
- Store older/backup data
- Sandbox for data exploration
- One-time reports
- Quick access to data
- Don’t know questions

Relational 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

Modern Data Warehouse architecture



Data Fabric

What is Data Fabric?

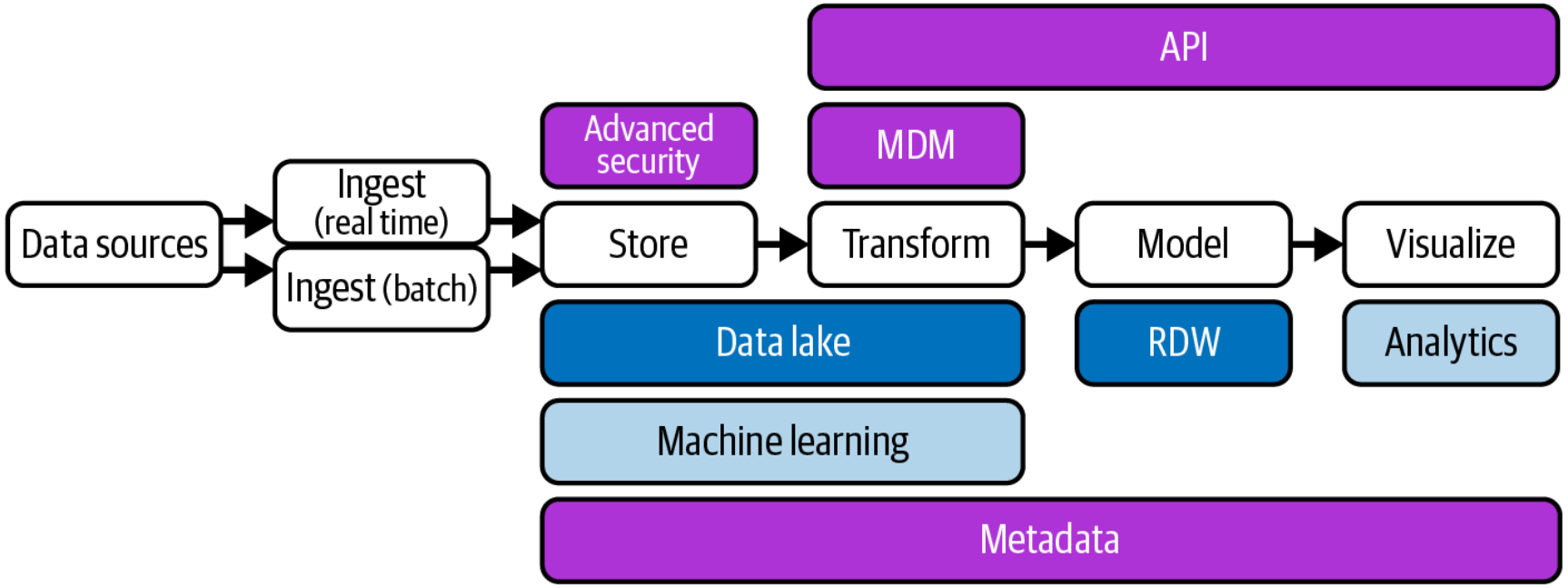
A data fabric is a term used to describe the architecture of taking disparate systems and weaving them together, like fabric, to create a consistent layer on top of an organization's data.

Data Fabric adds to a modern data warehouse:

- Data access policies
- Metadata catalog
- Master Data Management (MDM)
- Data virtualization
- Real-time processing
- APIs
- Building blocks/Services
- Products

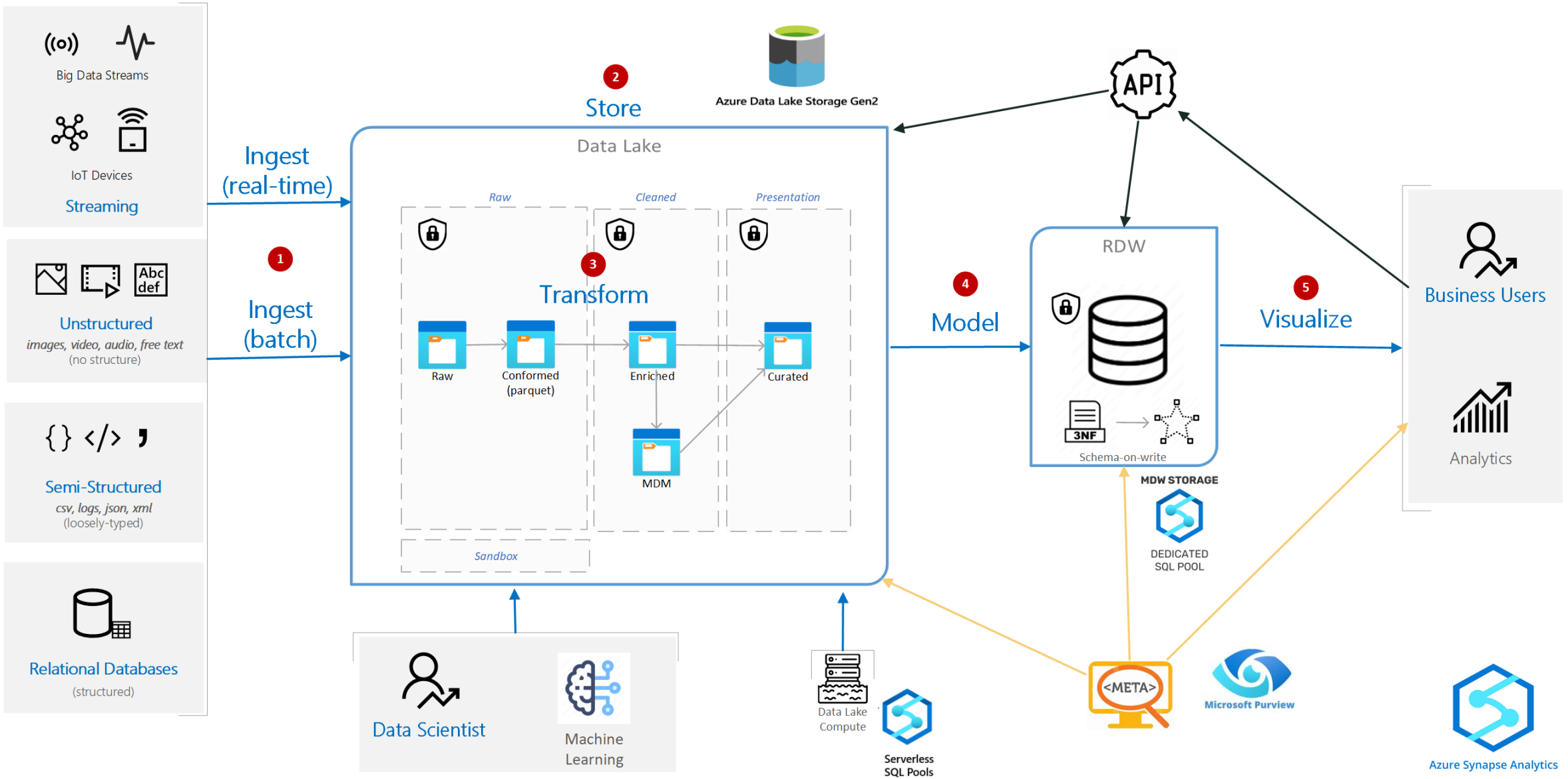
Bottom line: Data fabric provides additional technology to source more data, secure it, and make it available. *Think of it as an evolution of the MDW*

Data Fabric high-level architecture



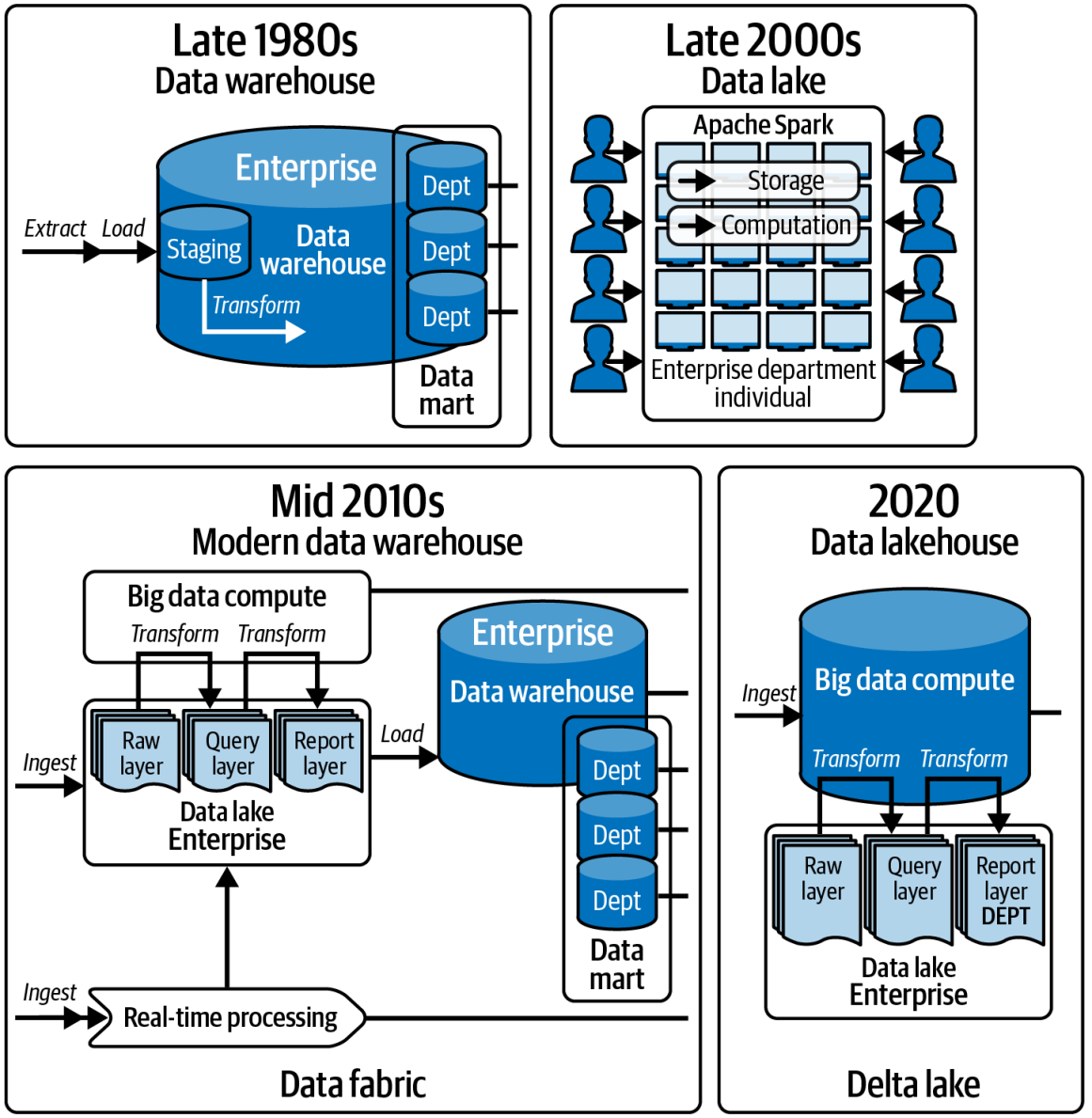
In purple are the data fabric features

Data Fabric architecture



Data Lakehouse

Data Lakehouse historical timeline



Years:

- RDW: 1984
- Data Lake: 2010
- MDW: 2011
- Data Fabric: 2016
- Data Lakehouse: 2020
- Data Mesh: 2019

Delta Lake

A transactional storage software layer that runs on top of an existing data lake, adding RDW-like features.

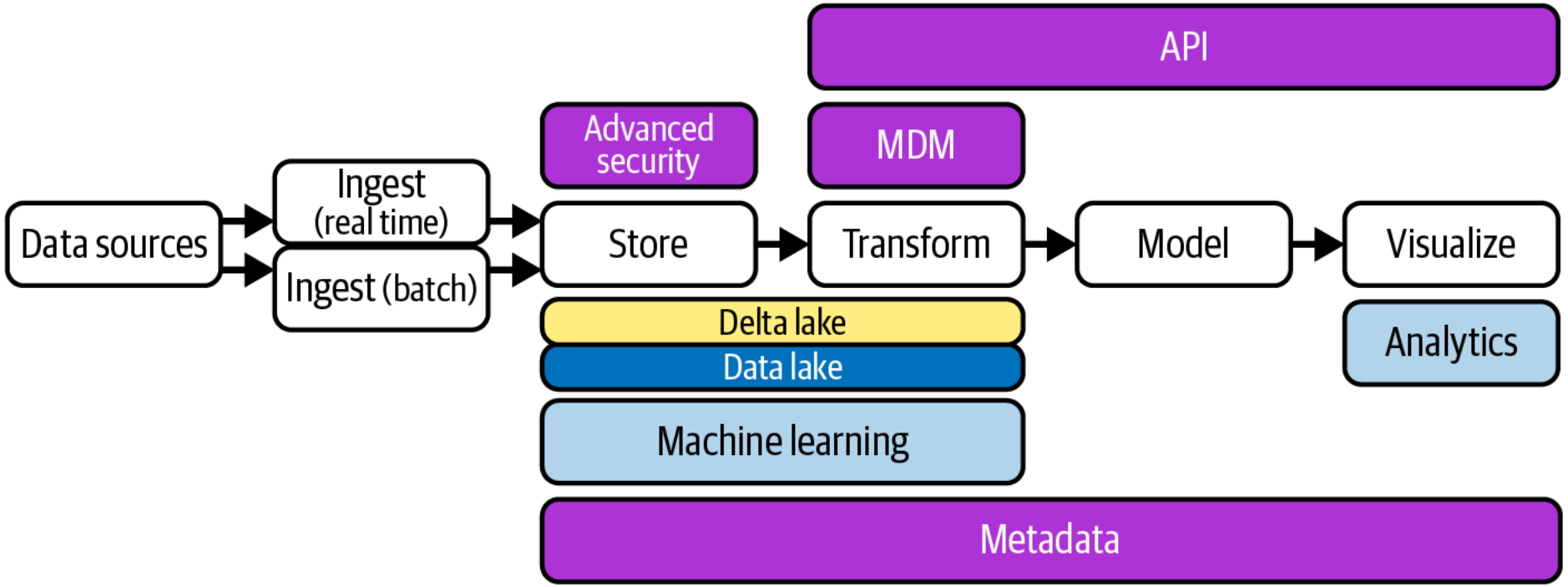
Top features:

- **Supports SQL commands INSERT, DELETE, UPDATE, and MERGE**
- ACID transactions for one table
- **Time travel (data versioning enables rollbacks, audit trail)**
- Streaming and batch unification
- Schema enforcement & schema evolution
- **Performance improvements (Data skipping, caching, Z-Order, etc)**
- Competitors: Apache Hudi, Apache Iceberg

Spark:

df.write.format("delta").save(delta_table_path)
instead of
df.write.format(“parquet").save(delta_table_path)

Data Lakehouse high-level architecture



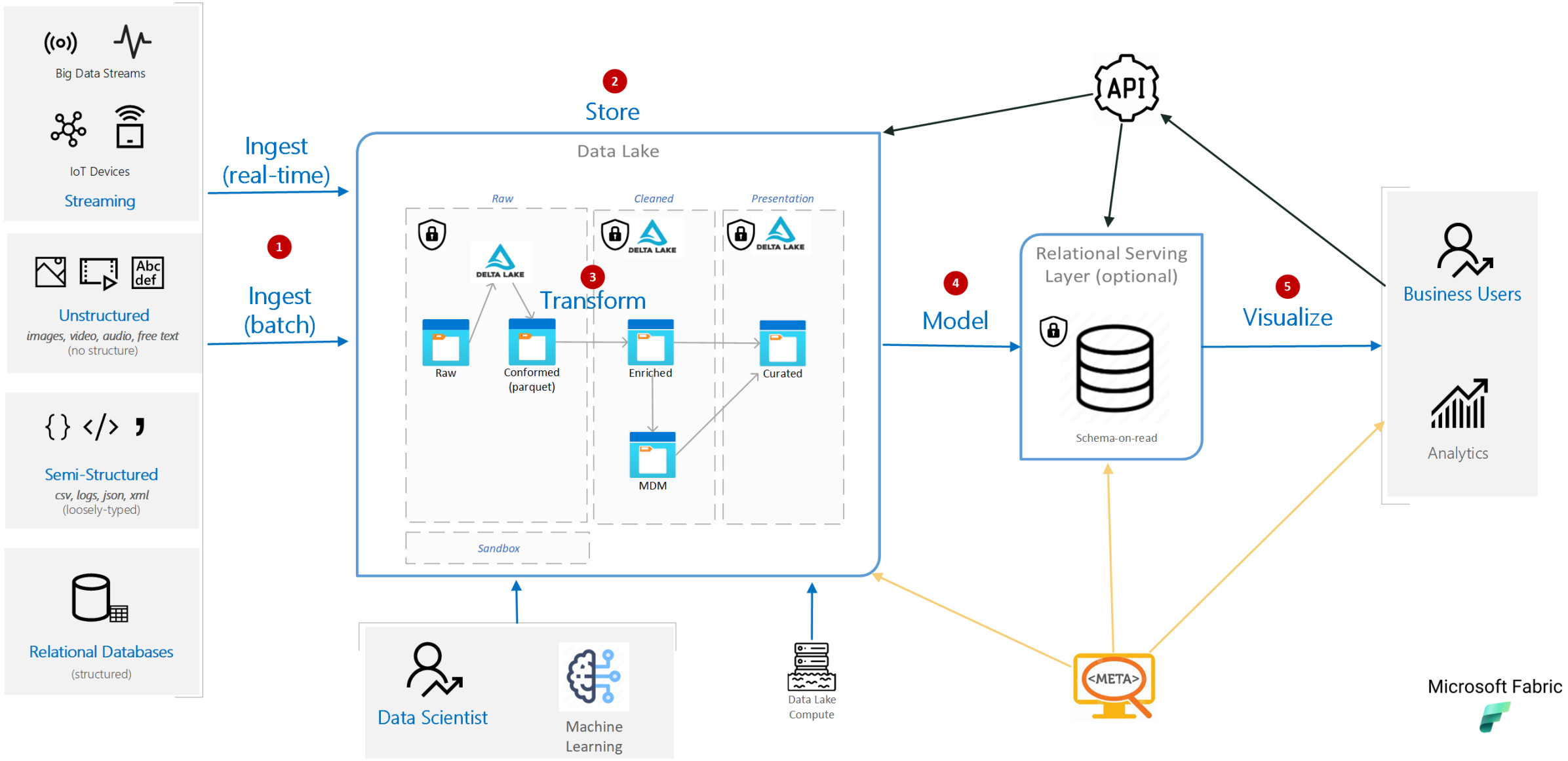
Added Delta lake, removed RDW

Use cases for Data Lakehouse

Today's data architectures commonly suffer from six problems:

- **Reliability: Keeping the data lake and warehouse consistent**
- **Data staleness: Data in warehouse is older**
- Limited support for advanced analytics: Data scientists prefer files
- Total cost of ownership: Extra cost for data copied to warehouse
- Data governance: More copies, more risk
- **Complexity: More specialized skills needed for both a data lake and RDW**

Data Lakehouse architecture



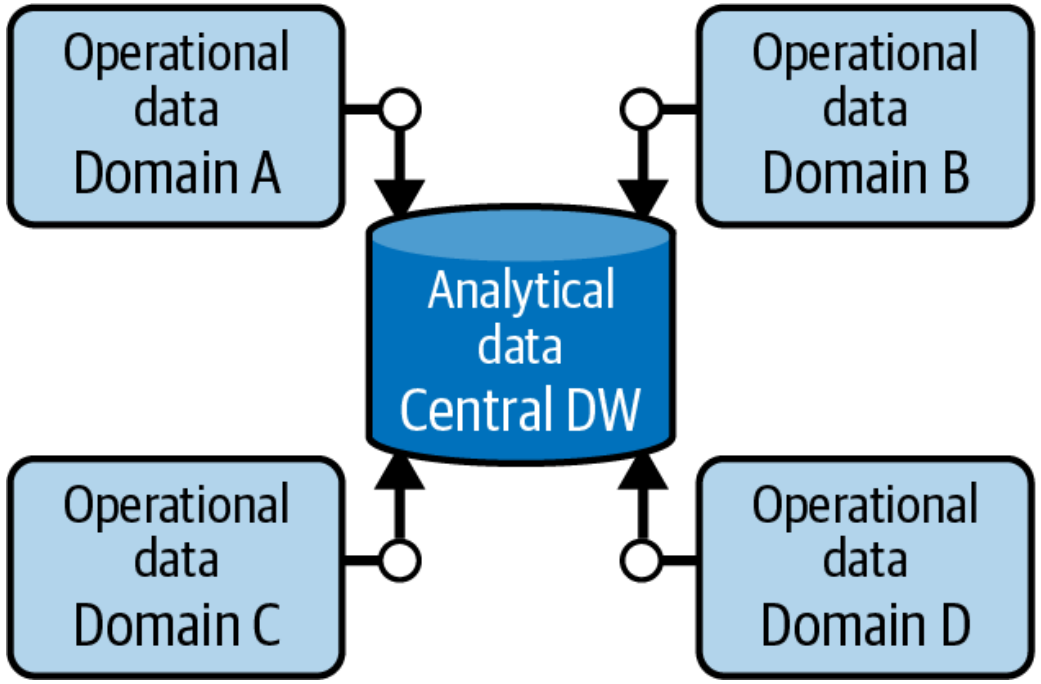
Concerns skipping relational data warehouse

- **Speed: Relational database *queries* are faster, especially with Massively Parallel Processing (MPP) and with advanced indexing, advanced statistics, caching, advanced query plan optimization, materialized views, advanced join optimization**
- **Security: No row-level security (RLS), column-level security, data-at-rest encryption, column-level encryption, Transparent Data Encryption (TDE), dynamic data masking**
- People are used to using a relational database (forced metadata layer)
- **Complexity: Metadata separate from data, file-based world**
- Concurrency: Multiple reads of a file at the same time can be slow
- Missing features: SQL Views, referential integrity, workload management, advanced auditing and compliance features (such as auditing trails, data retention policies, and compliance certifications), ACID against multiple tables
- Products must add delta lake support in order to use it

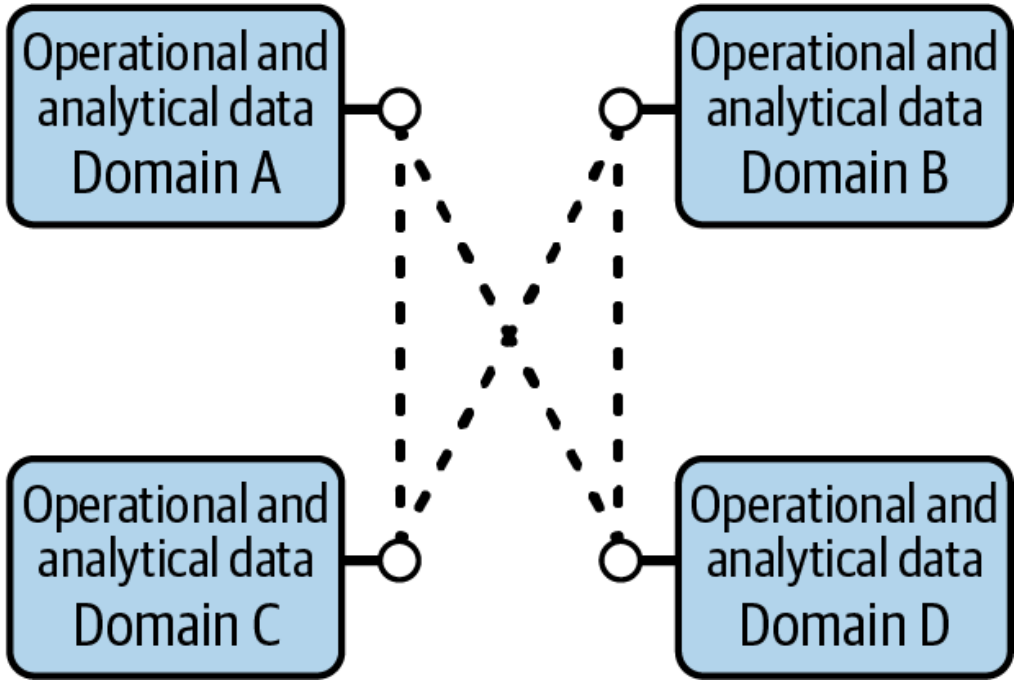
Microsoft Fabric and Power BI addresses most of these concerns. Otherwise, just put limited data in RDW that needs it

Data Mesh

Traditional centralized vs Data Mesh decentralized



MDW, Data Fabric, Data Lakehouse



Data Mesh

Credit to Zhamak Dehghani (May 2019)

Data Mesh

Data Mesh is a concept, not technology

*It is a **huge** organizational and cultural shift*

Credit to Zhamak Dehghani

Data Mesh - Overview

A data mesh is a decentralized approach to managing data, where multiple teams within a company are responsible for their own data, promoting collaboration and flexibility. By implementing data mesh principles, the quality and accuracy of data can be enhanced, resulting in increased trust among businesses to utilize data more extensively for informed decision-making.

Data Mesh Principles

#1) Domain Ownership

Decentralize and distribute responsibility to people who are closest to the data in order to support continuous change and scalability (i.e. manufacturing, sales, supplier)

#2) Data as a product

Analytical data provided by the domains are treated as a product and the consumers of that data are treated as customers (domain teams, API code, data and metadata, infrastructure)

#3) Self-serve data infrastructure as a platform

Simplify data product creation and management by automating infrastructure provisioning (i.e. storage, compute, data pipeline, access control)

#4) Federated computational governance

A collaborative data governance between domains and a central data team to define, implement and monitor global rules (i.e., interoperability, data quality, data security, regulations, data modelling)

Use cases for Data Mesh

Data mesh tries to solve four challenges with a centralized data lake/warehouse:

- Lack of ownership: who owns the data – the data source team or the infrastructure team?
- Lack of quality: the infrastructure team is responsible for quality but does not know the data well
- **Organizational scaling**: the central team becomes the bottleneck, such as with an enterprise data lake/warehouse
- **Technical scaling**: current big data solutions can’t keep up with additional data requirements

Example healthcare domains and products within them

Domain-driven design (DDD) is a long and complicated process!

Patient data domain

- Patient data analytics
- Patient engagement
- Population Health Management
- Clinical decision support systems
- Patient matching and ID resolution
- Clinical research
- Patient outcome prediction
- Patient satisfaction

Clinical data domain

- Clinical research
- Clinical decision support
- Clinical analytics
- Clinical trial management
- Imaging analysis
- Disease registries
- Real-world evidence
- Clinical trial matching

Claims data domain

- Claims analytics
- Claims management
- Fraud detection
- Payment processing
- Patient financial management
- Provider network analysis
- Claims denial management
- Health plan selection

Public health data domain

- Disease surveillance
- Health equity
- Population health management
- Environmental health tracking
- Public health research
- Infectious disease modeling
- Health behavior change
- Chronic disease management

Concerns with Data Mesh

- No standard definition of a data mesh
- **Huge investment in organizational change and technical implementation**
- Performance problem of combining data from multiple domains
- Duplication of data for performance reasons
- **Getting quality engineering people for each domain**
- Inconsistent technical implementations for the domains
- Domains don’t want to wait for a data mesh
- **Need incentives for each domain to counter extra work**
- Self-serve approach of data requests could be challenging
- Duplication of data and ingestion platform
- Creation of data silos for domains not able to join data mesh
- **Not seeing the big picture for combing data**

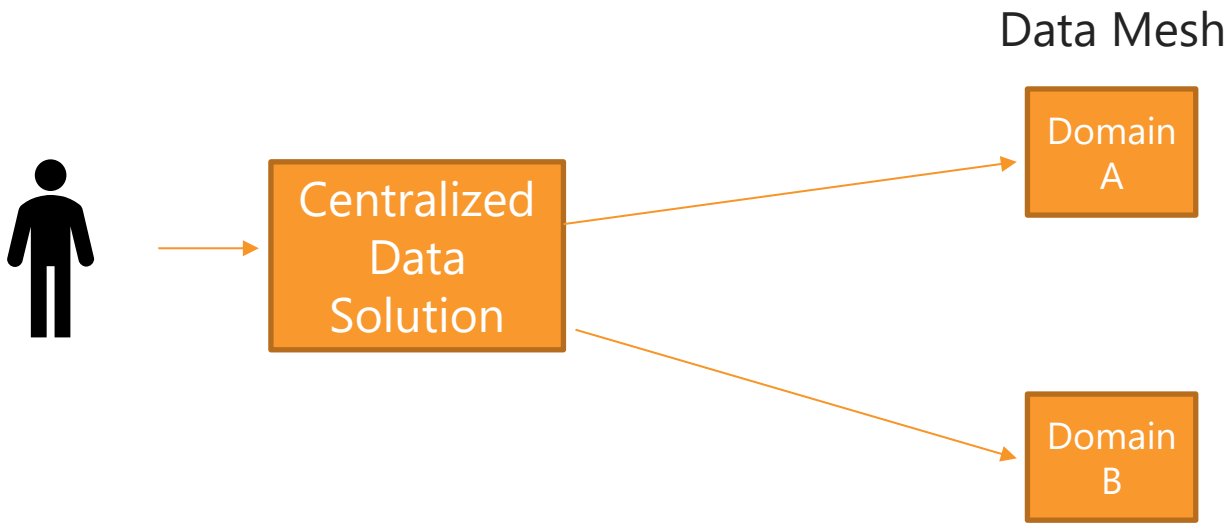
Data Mesh: Centralized vs decentralized data architecture
Data Mesh: Centralized ownership vs decentralized ownership

Data Mesh Future

This view is my own and not that of Microsoft!

In the end, I predict data mesh will become an extension to a centralized data solution for a small percentage of solutions via a hub-and-spoke model:

- Start by using new data to create new data mesh domains
- Supplement those domains with your current centralized data solution
- Slowly migrate your centralized data into data mesh domains over time
- Paves the way for a cultural shift over time



Data Mesh principles adoption estimate:

1) Domain ownership (90%)
2) Data as a product (70%)
3) Self-serve data infrastructure as a platform (30%)
4) Federated computational governance (50%)

Data Mesh concepts help with a better way of thinking how to get value out of data

When to use each architecture?

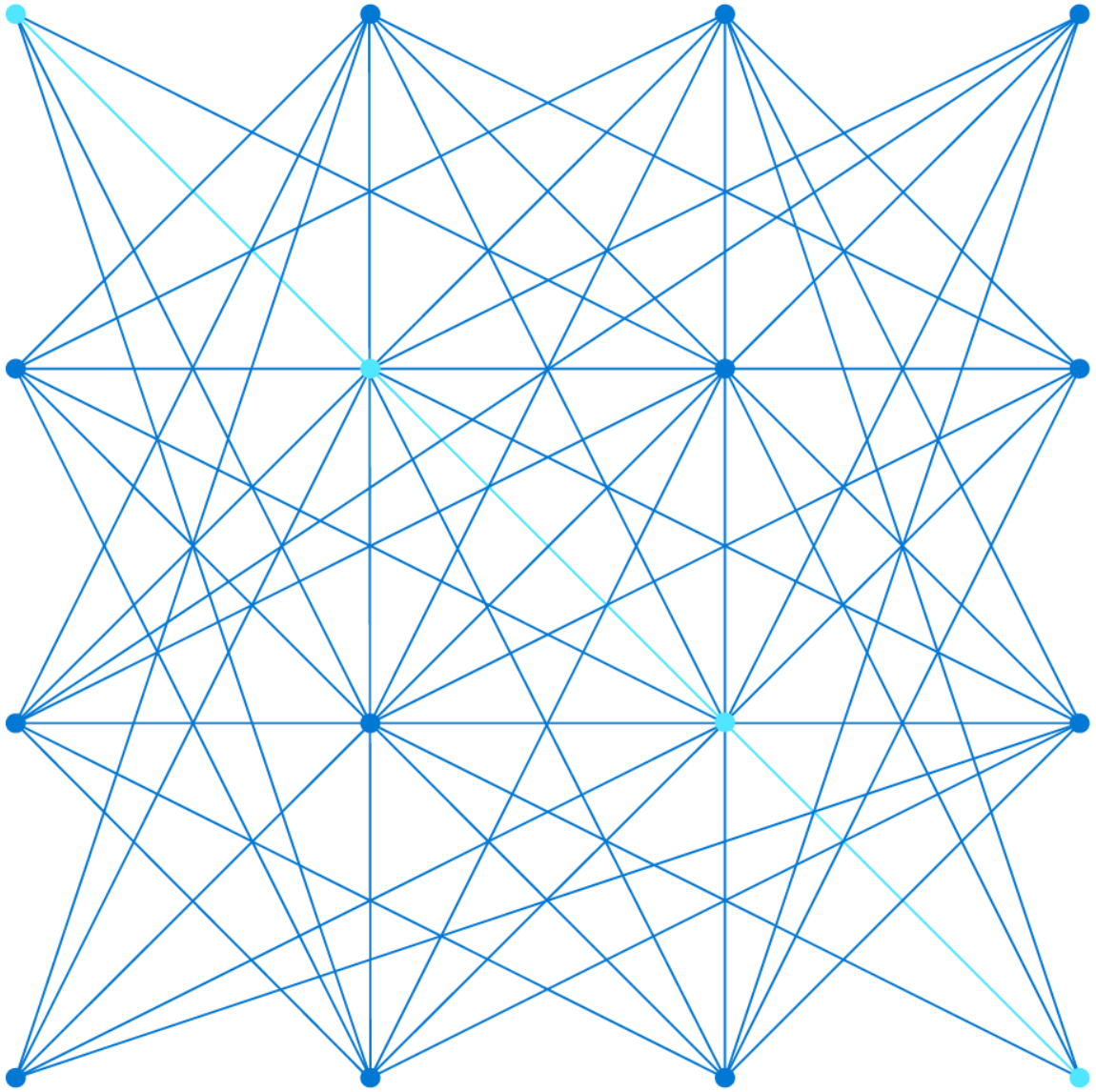
A very high-level use case for each architecture (in ascending cost and complexity):

- Modern data warehouse: Small amount of data (<1TB); if used to relational data warehouses (RDW); can skip data lake if very small amount of data
- Data fabric: Need to ingest many different data sources (size, speed, type). Large migration effort (i.e., many SP's would need to be rewritten if no RDW)
- Data lakehouse: Use it until you can't – then copy some data to RDW
- Data mesh: Very large, domain-oriented company, that is having major pain points with scalability and can afford a long timeline. Each domain will use one of the three architectures

Most companies will use pieces of each architecture to build a solution adapted to their specific needs for data (use cases) and their business capabilities.

Microsoft Azure

Data Architectures on Microsoft Fabric



Microsoft Fabric

Key:

= Lakehouse
= Warehouse
= Lakehouse table
= Warehouse table
= Shortcut
= Dataset

Spark Engine

Data Engineering

Use Spark Notebooks

Python R Scala Spark

Data pipeline

Dataflow Gen2

Write data into Lakehouse tables

SQL Engine / Polaris

Data Warehousing

Use SQL Queries & Stored Procedures

SSMS

Azure Data Studio

Full T-SQL support (see limitations)

Write data into Warehouse tables

Data pipeline

Dataflow Gen2

Microsoft Fabric: Lakehouse vs Warehouse (YouTube video)

Architectures Microsoft Fabric supports

- Modern data warehouse: Yes, use both Lakehouse and Warehouse (but really data lakehouse architecture). Could use just Warehouse if small amount of data or used to RDW
- Data fabric: Yes, using Fabric Real-Time Intelligence. Use just Warehouse for migrations
- Data lakehouse: Yes, rare case use Eventhouse or Cosmos DB
- Data mesh: Yes, with exceptions

Recap:

- Modern data warehouse: Small amount of data (<1TB); if used to relational data warehouses (RDW); can skip data lake if very small amount of data
- Data fabric: Need to ingest many different data sources (size, speed, type). Large migration effort (i.e., many SP's would need to be rewritten if no RDW)
- Data lakehouse: Use it until you can't – then copy some data to RDW
- Data mesh: Very large, domain-oriented company, that is having major pain points with scalability and can afford a long timeline. Each domain will use one of the three architectures

Data Lakehouse – Microsoft Fabric

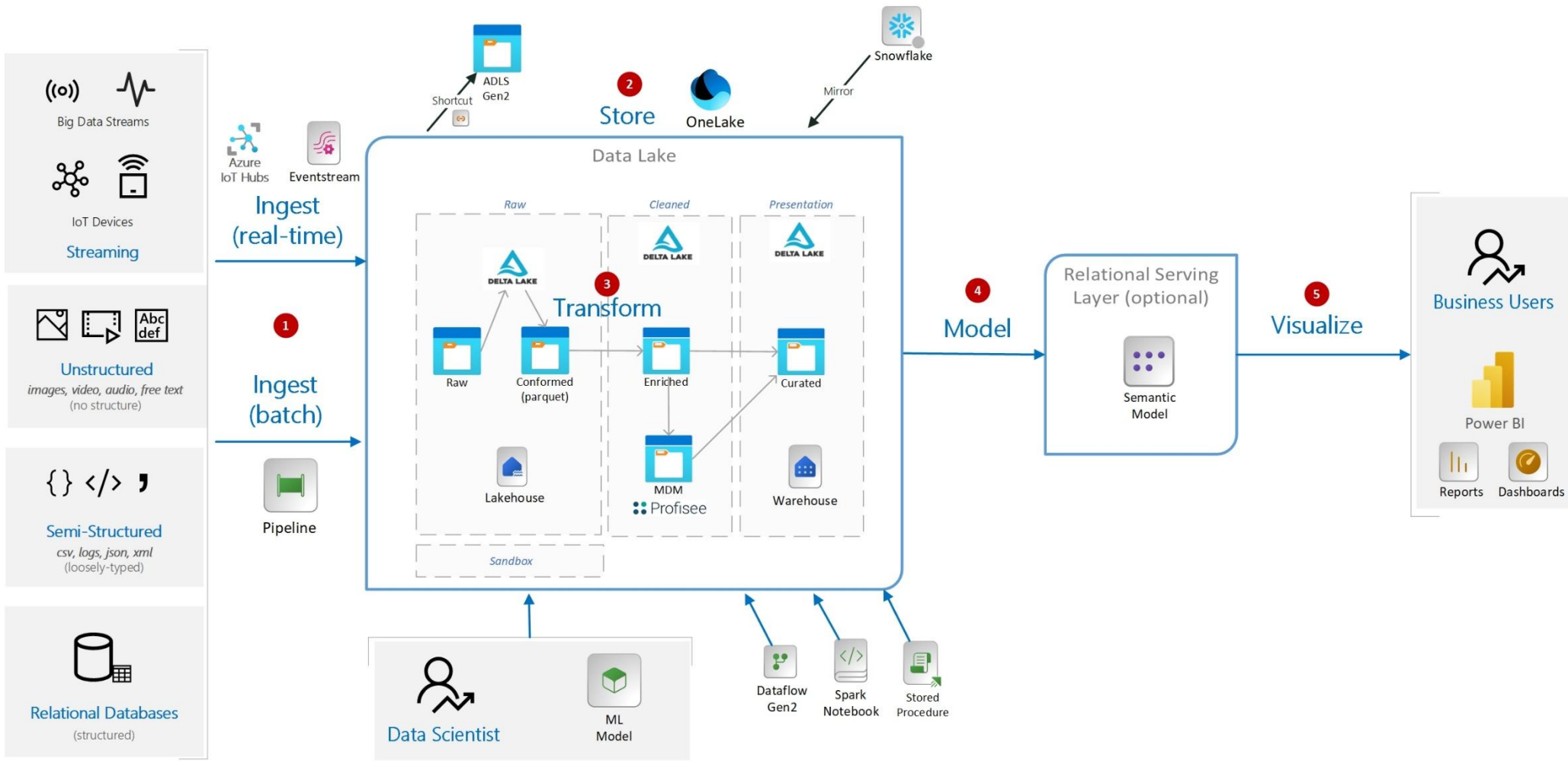
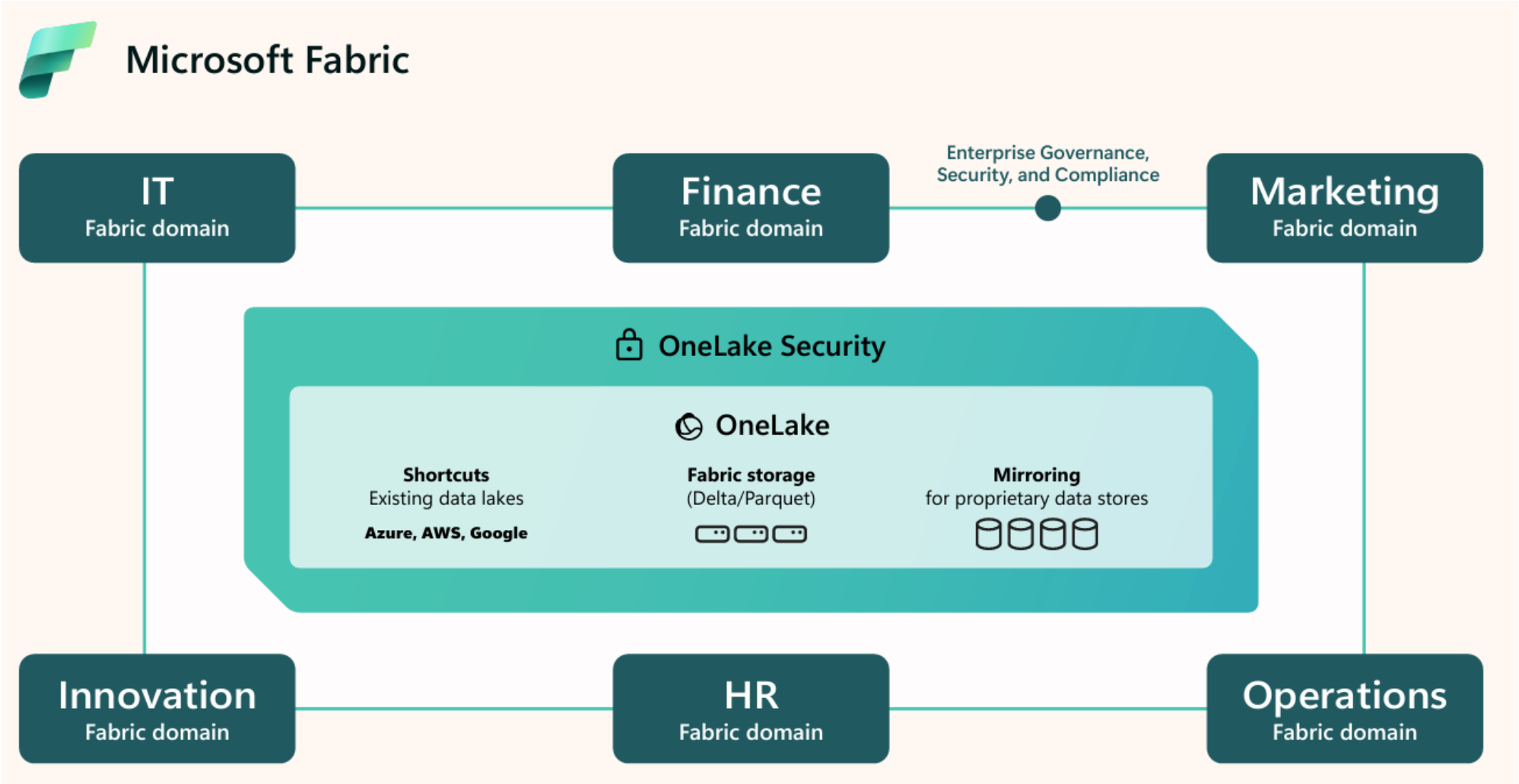


Diagram by ***JamesSerra.com***

Discover & Govern — Microsoft Purview

Platform — Microsoft Entra ID | Cost Management | Azure Key Vault | Azure Monitor | Security Center | Azure DevOps & GitHub | Azure Policy

Microsoft Fabric domains



Microsoft Fabric and Data Mesh

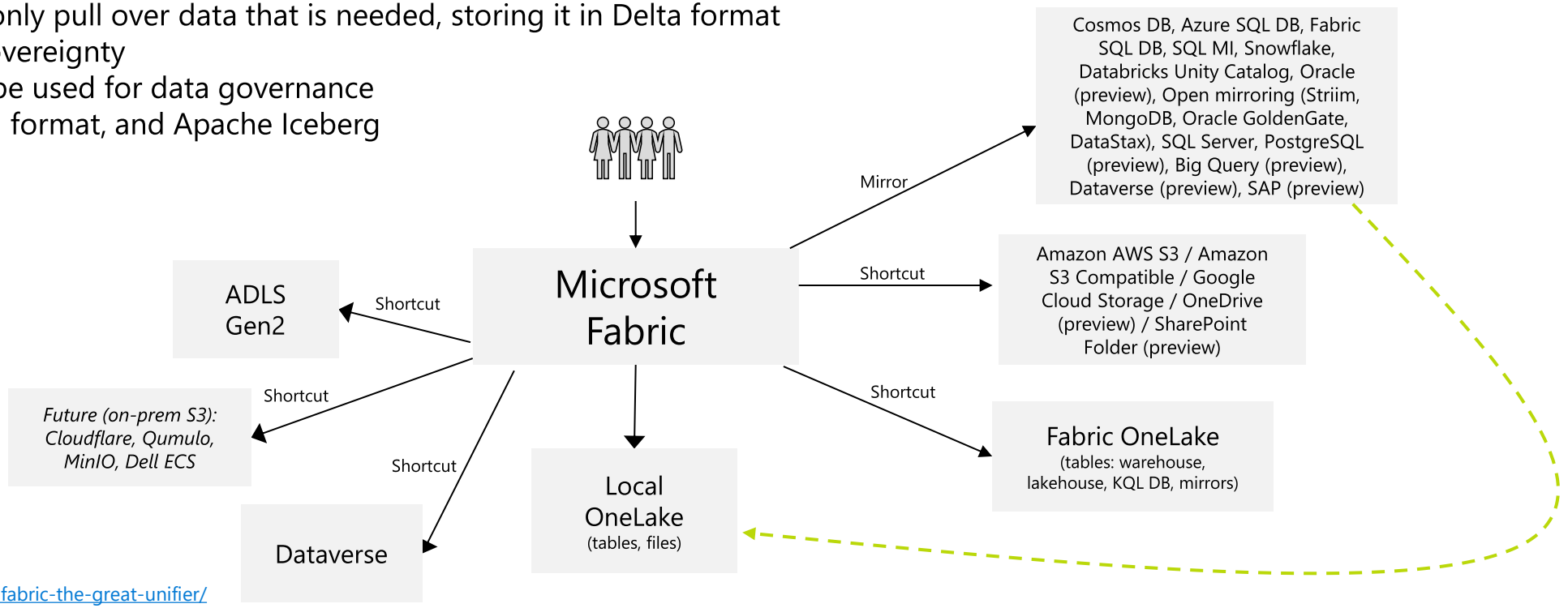
Typical exceptions to a "pure" data mesh when using Microsoft Fabric to build a data mesh:

- Use Microsoft Fabric for each domain (instead of each domain choosing their own technology)
- Use Microsoft Fabric shortcuts for data and metadata distribution (instead of API's)
- Data mesh principles not enforced (Microsoft Fabric domains can contain anything in the Fabric Workspaces)
- Having data products that simply contain data and metadata and not implementing the more complicated concept of using a data quantum as a data product (in which it holds not only the domain-oriented data and metadata but also the code that performs the necessary data transformations, the policies governing the data and the infrastructure to hold the data – Zhamak Dehghani)

Data Mesh vs Data Fabric - in Microsoft Fabric (Marthe Moengen)
Data Pancakes – Piethein Strengholt
Microsoft Fabric and Data Mesh - a perfect fit?
The data mesh challenge: closing the gap between strategy and operation at scale (Zhamak)

Think of Fabric as the great unifier

- Supplement other sources (Databricks, Snowflake) and storage (S3) and use shortcuts to give end-users the ability to easily create reports from multiple sources
- Some customers will use a competing cloud platform for various data analytics workloads, but may still want to use Fabric's business intelligence, data science, data engineering, and other capabilities on that data
- Easy button - helps end-users who are not technical to unify data from all different sources
- Microsoft is not trying to compete or replace other products/clouds
- Helps with being multi-cloud
- Great for bringing data together to train ML models
- Think of shortcuts as a light virtualization layer
- Shortcuts cache data and only pull over data that is needed, storing it in Delta format
- Shortcuts supports data sovereignty
- Microsoft Purview should be used for data governance
- Shortcuts are against Delta format, and Apache Iceberg



https://www.jamesserra.com/archive/2023/12/microsoft-fabric-the-great-unifier/

Modern Analytics, AI and Governance at Scale (MA2G)

Enterprise Data Strategy

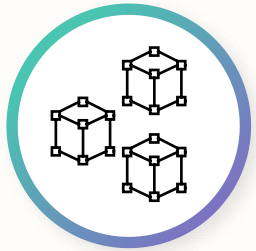
Culture Transformation

Align Process + People + Technology

Autonomous Lines of Businesses

Organization Change Management

Platform and Data Ownership



Composable Solution

Enterprise Data Governance

Data Management Foundation

Domains and Data Products

Technical Architecture

Microsoft Cloud Adoption Framework

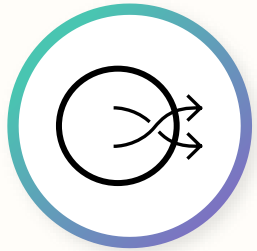
Cloud Scale Analytics

Azure Landing Zones

Azure Data Services

IPs and Accelerators

(Microsoft + Partners)



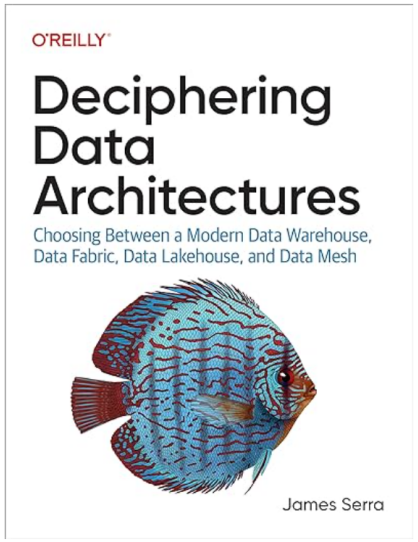
Implementation Stages

Discovery, MVPs, Phases

Modern Analytics, AI, Governance | Microsoft Azure (ebook)

Microsoft packaged solution based on learnings from our own and other clients' transformation journey

My book



Roll over image to zoom in

Deciphering Data Architectures 1st Edition, Kindle Edition

by James Serra (Author) | Format: Kindle Edition

5.0 ★★★★★ 3 ratings

#1 New Release in Data Modeling & Design

See all formats and editions

Book description | Editorial reviews

Data fabric, data lakehouse, and data mesh have recently appeared as viable alternatives to the modern data warehouse. These new architectures have solid benefits, but they're also surrounded by a lot of hyperbole and confusion. This practical book provides a guided tour of each architecture to help data professionals understand its pros and cons.

In the process, James Serra, big data and data warehousing solution architect at Microsoft, examines common data architecture concepts, including how data warehouses have had to evolve to work with data lake features. You'll learn what data lakehouses can help you achieve, and how to distinguish data mesh hype from reality. Best of all, you'll be able to determine the most appropriate data architecture for your needs. By reading this book, you'll:

- Gain a working understanding of several data architectures
- Know the pros and cons of each approach
- Distinguish data architecture theory from the reality
- Learn to pick the best architecture for your use case
- Understand the differences between data warehouses and data lakes
- Learn common data architecture concepts to help you build better solutions
- Alleviate confusion by clearly defining each data architecture
- Know what architectures to use for each cloud provider

Read entire book now with an O'Reilly subscription:
Deciphering Data Architectures (O'Reilly.com)

- Foundation
  - 1. Big Data
  - 2. Types of Data Architectures
  - 3. The Architecture Design Session
- Common Data Architecture Concepts
  - 4. The Relational Data Warehouse
  - 5. Data Lake
  - 6. Data Storage Solutions and Processes
  - 7. Approaches to Design
  - 8. Approaches to Data Modeling
  - 9. Approaches to Data Ingestion
- Data Architectures
  - 10. The Modern Data Warehouse
  - 11. Data Fabric
  - 12. Data Lakehouse
  - 13. Data Mesh Foundation
  - 14. Should You Adopt Data Mesh? Myths, Concerns, And The Future
- People, Process, and Technology
  - 15. People And Processes
  - 16. Technologies

More details

jamesserra3@gmail.com

Order on Amazon!

Q & A

James Serra, Microsoft, Data & AI Solution Architect
Email me at: jamesserra3@gmail.com
Follow me at: @JamesSerra
Link to me at: www.linkedin.com/in/JamesSerra
Visit my blog at: JamesSerra.com
(email me for the deck)

Sound off.
The mic is all yours.
Influence the product roadmap.

Join the Fabric User Panel

Share your feedback directly with our Fabric product group and researchers.

https://aka.ms/JoinFabricUserPanel

Join the SQL User Panel

Influence our SQL roadmap and ensure it meets your real-life needs

https://aka.ms/JoinSQLUserPanel

How was the session?

Complete Session Surveys in Whova for your chance to WIN PRIZES!