Data Lakehouse, Data Mesh, and Data Fabric

(the alphabet soup of data architectures)

sqlbits

James Serra

Data & AI Solution Architect

Microsoft

jamesserra3@gmail.com

Blog: JamesSerra.com

About Me

- Microsoft, Data & AI Solution Architect in Microsoft Consulting Services (MCS), now called Industry Solutions Delivery (ISD)
- At Microsoft for most of the last eight years, with a brief stop at EY
- Was previously a Data & AI Architect at Microsoft for seven years
- In IT for 35 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, Enterprise Data World conference, Big Data Conference Europe, SQL Saturdays
- Blog at JamesSerra.com
- Former SQL Server MVP
- Author of book “Reporting with Microsoft SQL Server 2012”

MVP Microsoft® Most Valuable Professional

Agenda

- Data Warehouse
- Data Lake
- Modern Data Warehouse
- Data Fabric
- Data Lakehouse
- Data Mesh

I tried to figure out all these data platform buzzwords on my own…

And it did not turn out well:

Let’s prevent that from happening…

The view in this deck are my own and not that of Microsoft!

What is a Data Warehouse and why use one?

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

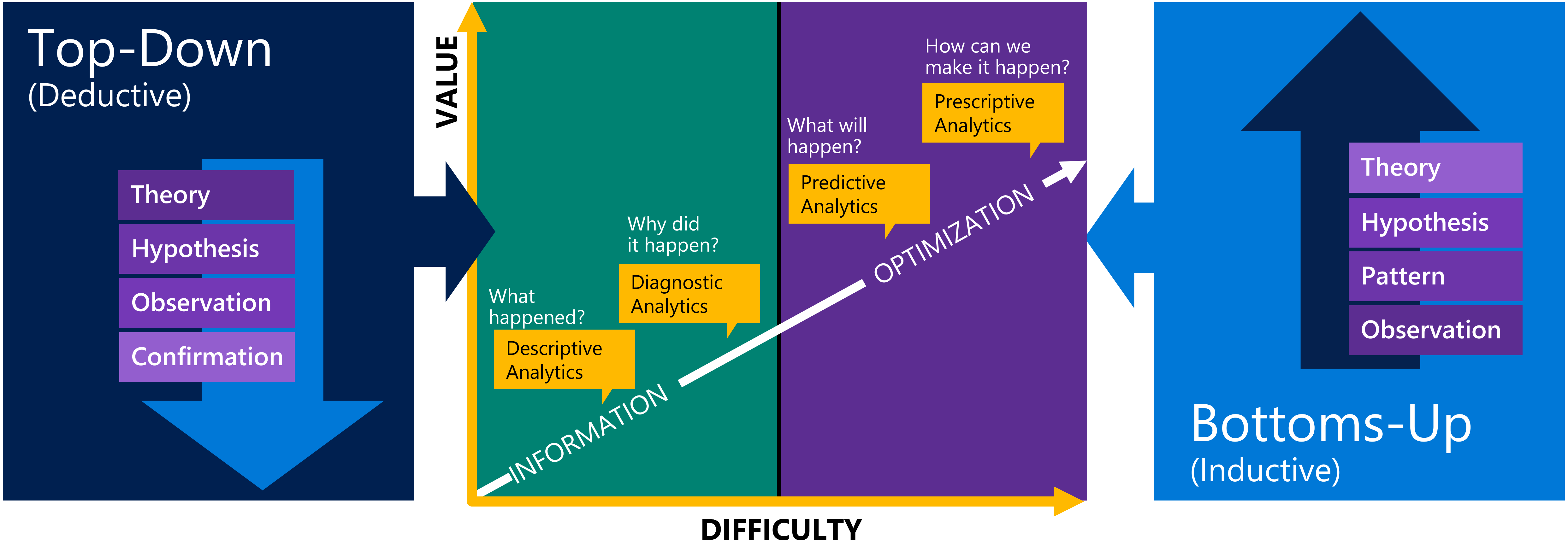
A data warehouse is where you store data from multiple data sources to be used for historical and trend analysis reporting. It acts as a central repository for many subject areas and contains the "single version of truth". It is NOT to be used for OLTP applications.

Reasons for a data warehouse:

- Reduce stress on production system
- Optimized for read access, sequential disk scans
- Integrate many sources of data
- Keep historical records (no need to save hardcopy reports)
- Restructure/rename tables and fields, model data
- Protect against source system upgrades
- Use Master Data Management, including hierarchies
- No IT involvement needed for users to create reports
- Improve data quality and plugs holes in source systems
- One version of the truth
- Easy to create BI solutions on top of it (i.e. Power BI tabular model)
- Don’t need to provide security access for many users to the production systems
- Make better business decisions by getting greater insights into your company

Why You Need a Data Warehouse

Two Approaches to getting value out of data: Top-Down + Bottoms-Up



Top-Down
(Deductive)

Theory

Hypothesis

Observation

Confirmation

VALUE

DIFFICULTY

What happened?

Descriptive Analytics

Why did it happen?

Diagnostic Analytics

What will happen?

Predictive Analytics

How can we make it happen?

Prescriptive Analytics

INFORMATION

OPTIMIZATION

Bottoms-Up
(Inductive)

Theory

Hypothesis

Pattern

Observation

- Know the questions to ask
- Lot’s of upfront work to get the data to where you can use it
- Model first (schema-on-write)

- Don’t know the questions to ask
- Little upfront work needs to be done to start using data
- Model later (schema-on-read)

Data Warehousing Uses A Top-Down Approach

Understand Corporate Strategy

Gather Requirements

Business Requirements

Technical Requirements

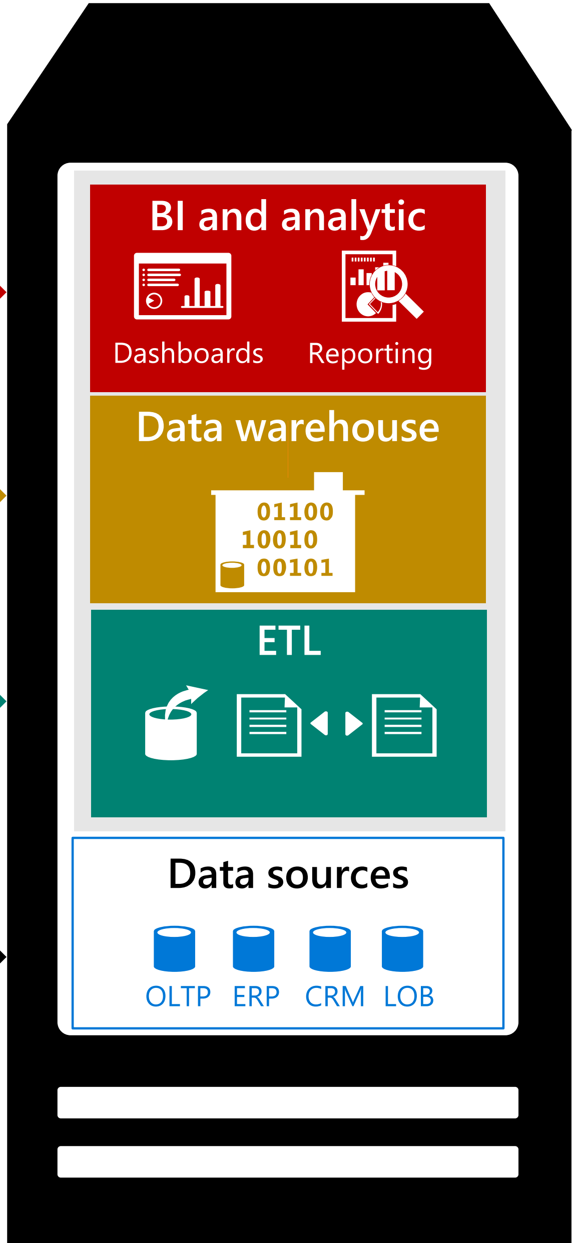
Implement Data Warehouse

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

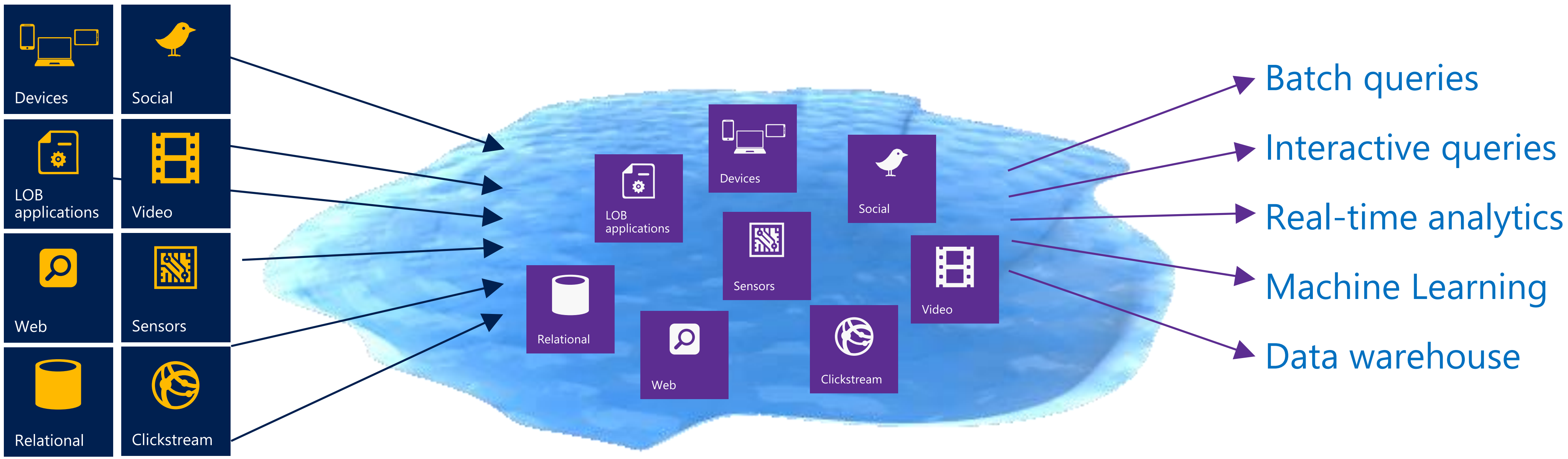
OLTP ERP CRM LOB

The “data lake” Uses A Bottoms-Up Approach

Ingest all data
regardless of requirements

Store all data
in native format without schema definition

Do analysis
Using analytic engines like Hadoop



- Batch queries
- Interactive queries
- Real-time analytics
- Machine Learning
- Data warehouse

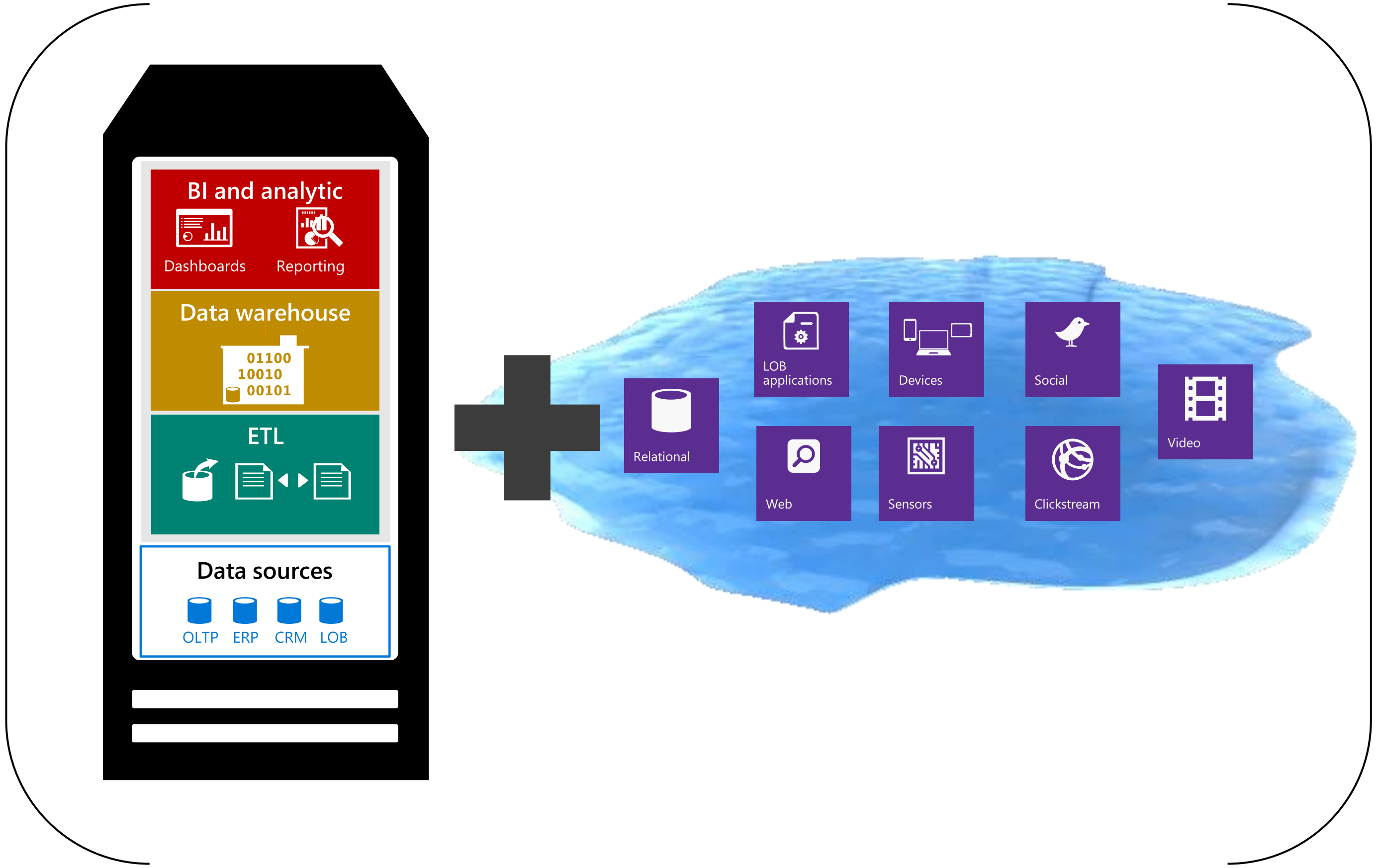
Data Lake + Data Warehouse Better Together

What happened?

Descriptive Analytics

Why did it happen?

Diagnostic Analytics



What will happen?

Predictive Analytics

How can we make it happen

Prescriptive Analytics

What is a data lake and why use one?

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

Reasons for a data lake:

- Inexpensively store unlimited data
- Centralized place for multiple subjects (single version of the truth)
- Collect all data "just in case" (data hoarding). The data lake is a good place for data that you "might" use down the road
- 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 transformations (avoiding user contention)**
- Wanting to use technologies/tools (i.e Databricks) to refine/filter data that do the refinement quicker/better than your EDW
- **Quick user access to data for power users/data scientists (allowing for faster ROI)**
- **Data exploration to see if data valuable before writing ETL and schema for relational database, or use for one-time report**
- Allows use of Hadoop tools such as ETL and extreme analytics
- Place to land IoT streaming data
- On-line archive or backup for data warehouse data (i.e. keep three years of data in DW and have older data in data lake with an external table pointing to it)
- With Hadoop/ADLS, high availability and disaster recovery built in
- It can ingest large files quickly and provide data redundancy
- ELT jobs on EDW are taking too long because of increasing data volumes and increasing rate of ingesting (velocity), so offload some of them to the Hadoop data lake
- Have a backup of the raw data in case you need to load it again due to an ETL error (and not have to go back to the source). You can keep a long history of raw data
- Allows for data to be used many times for different analytic needs and use cases
- Cost savings and faster transformations: storage tiers with lifecycle management; separation of storage and compute resources allowing multiple instances of different sizes working with the same data simultaneously vs scaling data warehouse; low-cost storage for raw data saving space on the EDW
- **Extreme performance for transformations by having multiple compute options each accessing different folders containing data**
- The ability for an end-user or product to easily access the data from any location

Data Lake with DW use cases

Data Lake

Staging & preparation

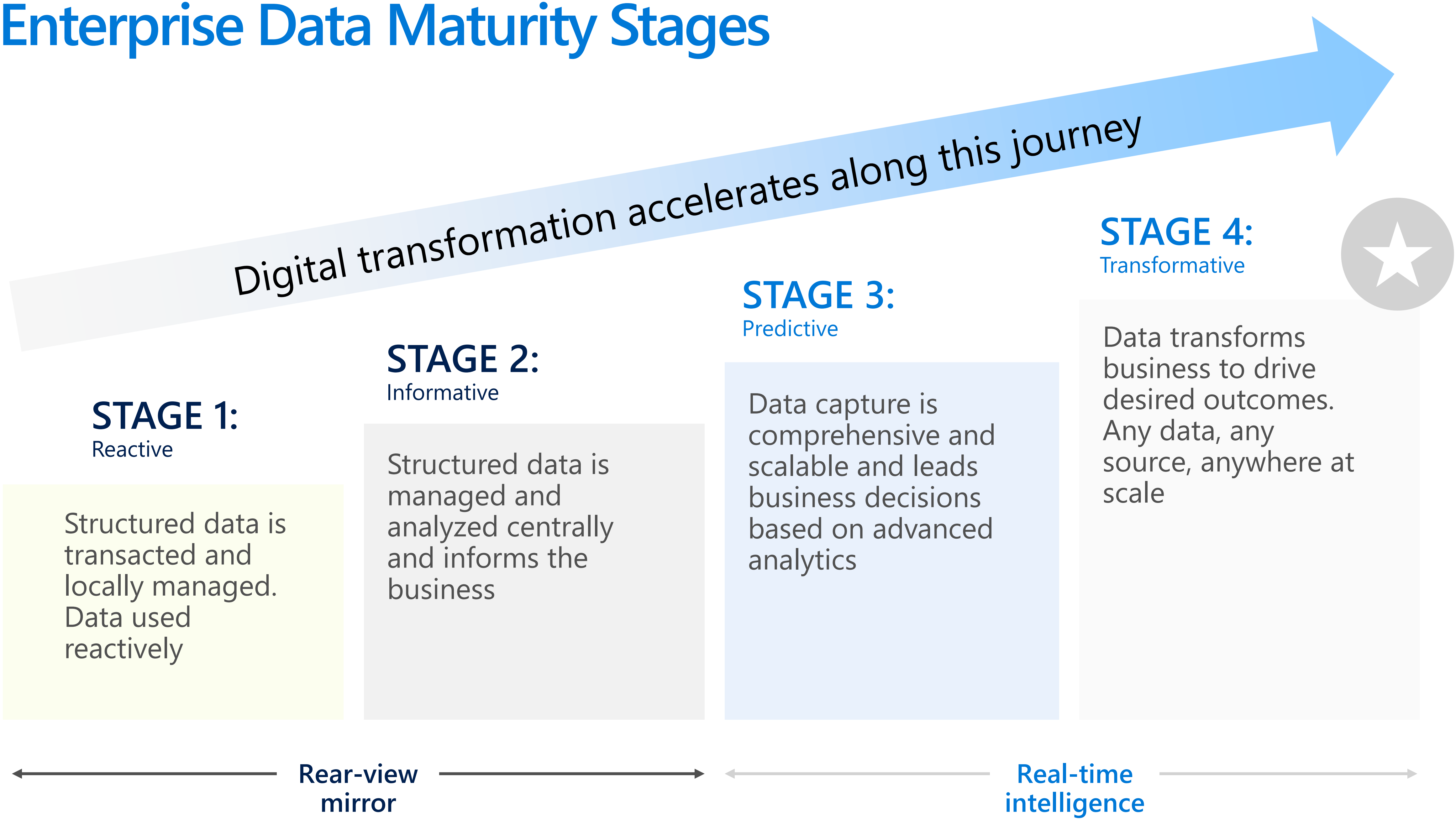
- Data scientists/Power users
- Batch processing
- Data refinement/cleaning
- ETL workloads
- Store older/backup data
- Sandbox for data exploration
- One-time reports
- Quick access to data
- Don’t know questions

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

Enterprise Data Maturity Stages



Digital transformation accelerates along this journey

STAGE 1:
Reactive

Structured data is transacted and locally managed. Data used reactively

STAGE 2:
Informative

Structured data is managed and analyzed centrally and informs the business

STAGE 3:
Predictive

Data capture is comprehensive and scalable and leads business decisions based on advanced analytics

STAGE 4:
Transformative

Data transforms business to drive desired outcomes. Any data, any source, anywhere at scale

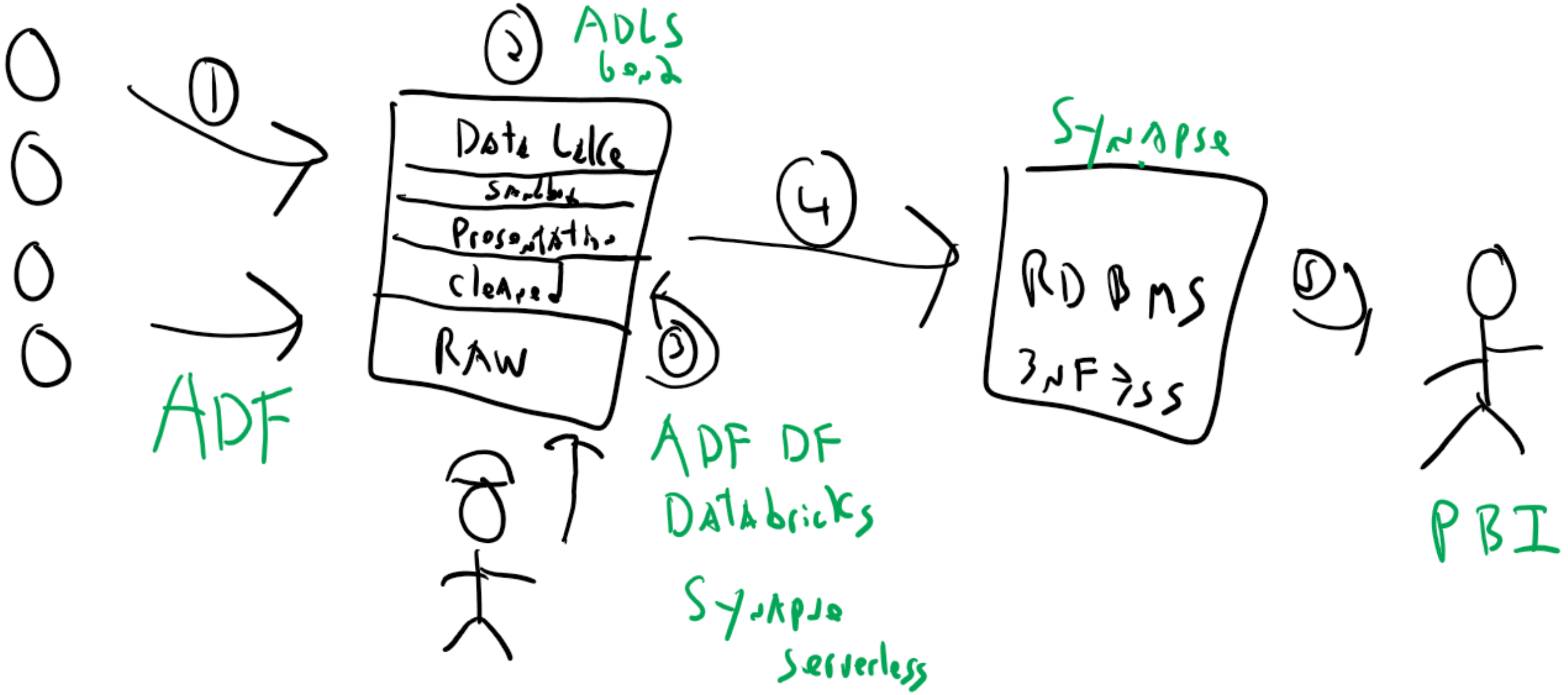
Rear-view mirror

Real-time intelligence

Modern Data Warehouse

Modern Data Warehouse (MDW)

1) IngesT
2) Store
3) Transform
4) Model
5) Visualize/ML



Data Fabric

Data Fabric adds to a modern data warehouse:

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

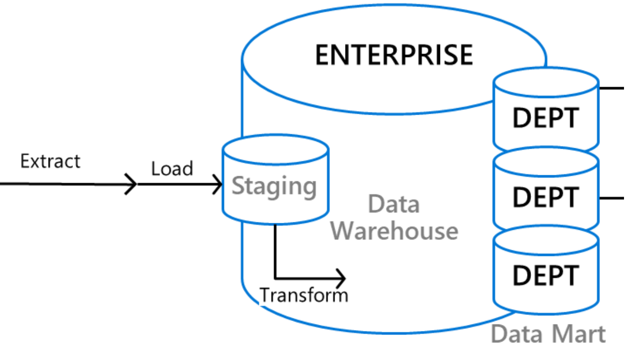
Bottom line: Additional technology to source more data, secure it, and make it available

Data Fabric defined

Data Lakehouse

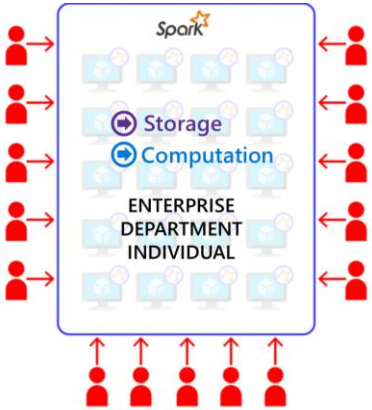
Late 1980s

Data Warehouse



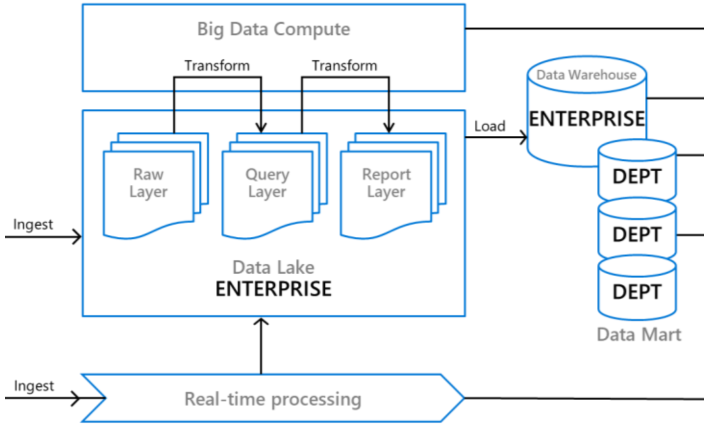
Late 2000s

Data Lake



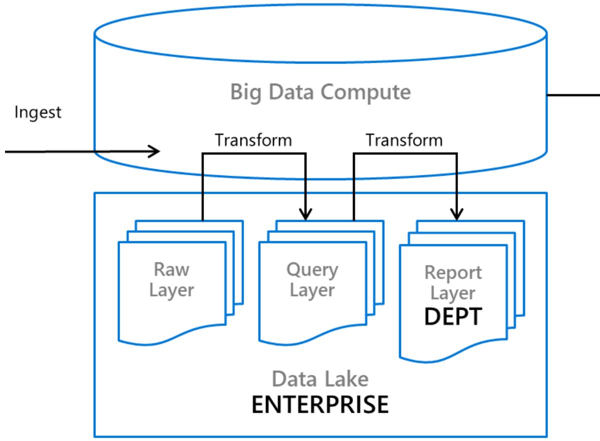
Mid 2010s

Cloud Data Platform



Early 2020s

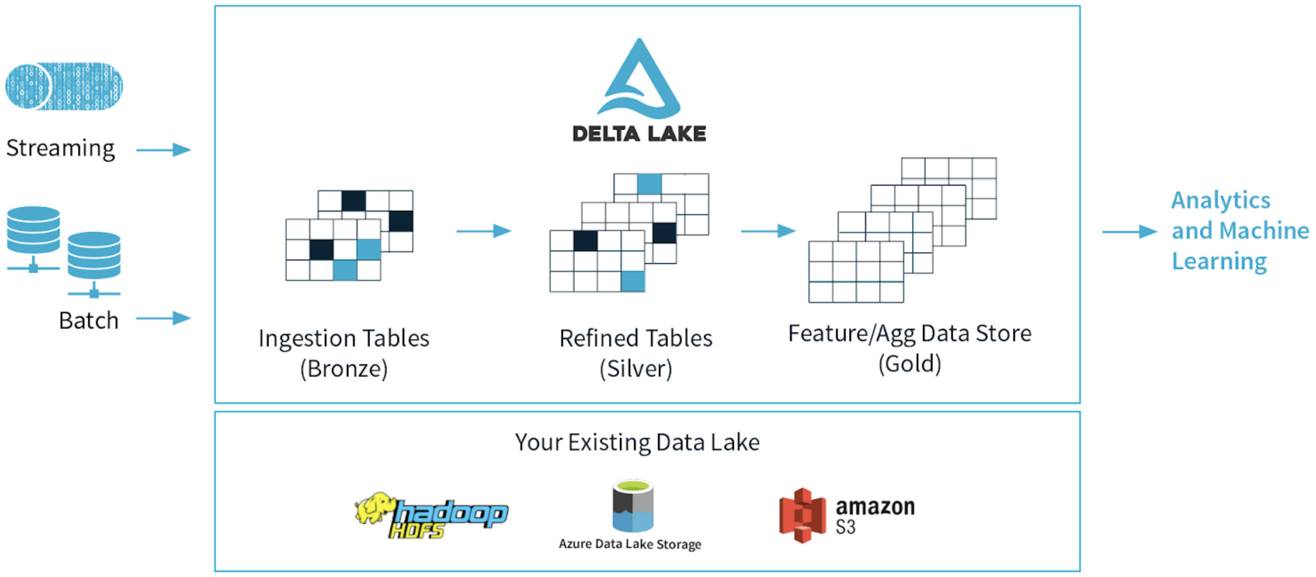
Data Lakehouse



Delta Lake

Top features:

- ACID transactions
- Time travel (data versioning enables rollbacks, audit trail)
- Streaming and batch unification
- Schema enforcement
- Upserts and deletes (MERGE)
- Performance improvement



Databricks Delta Lake

Use cases for Data Lakehouse

Today’s data architectures commonly suffer from four problems:

- Reliability: Keeping the data lake and warehouse consistent
- Data staleness: Data in warehouse is older
- Limited support for advanced analytics: Top ML systems don’t work well on warehouses
- Total cost of ownership: Extra cost for data copied to warehouse

Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics

Concerns skipping relational database

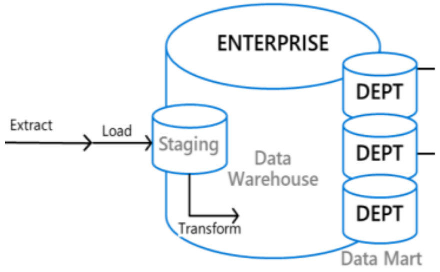
- Speed: Relational databases faster, especially Massively Parallel Processing (MPP)
- Security: No RLS, column-level, dynamic data masking
- Complexity: Metadata separate from data, file-based world
- Concurrency: Multiple reads of a file at the same time
- Missing features: Referential integrity, TDE, workload management; other features lock you into Spark
- Having to use Spark SQL instead of T-SQL
- People used to using a relational database

Azure Synapse: starting to see data lake only solutions because can use T-SQL, Power BI (speed, RLS), cost savings with Serverless

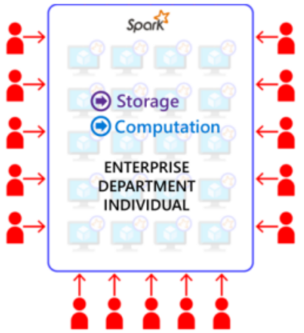
Data Lakehouse & Synapse

Data Mesh

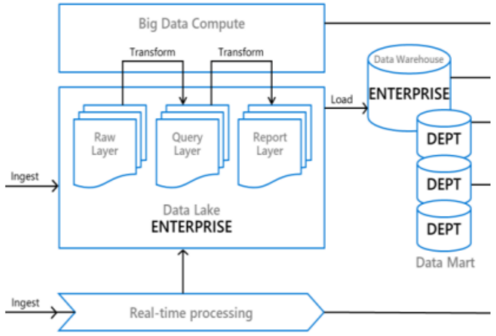
Late 1980s
Data Warehouse



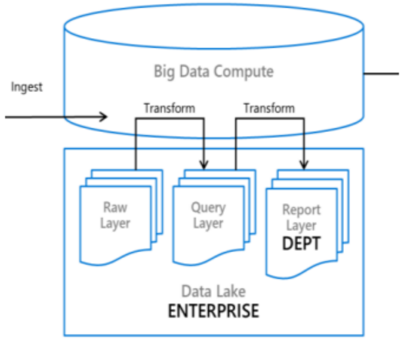
Late 2000s
Data Lake



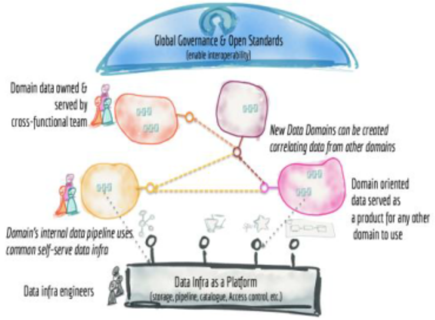
Mid 2010s
Cloud Data Platform



2020
Data Lakehouse



2021
Data Mesh??

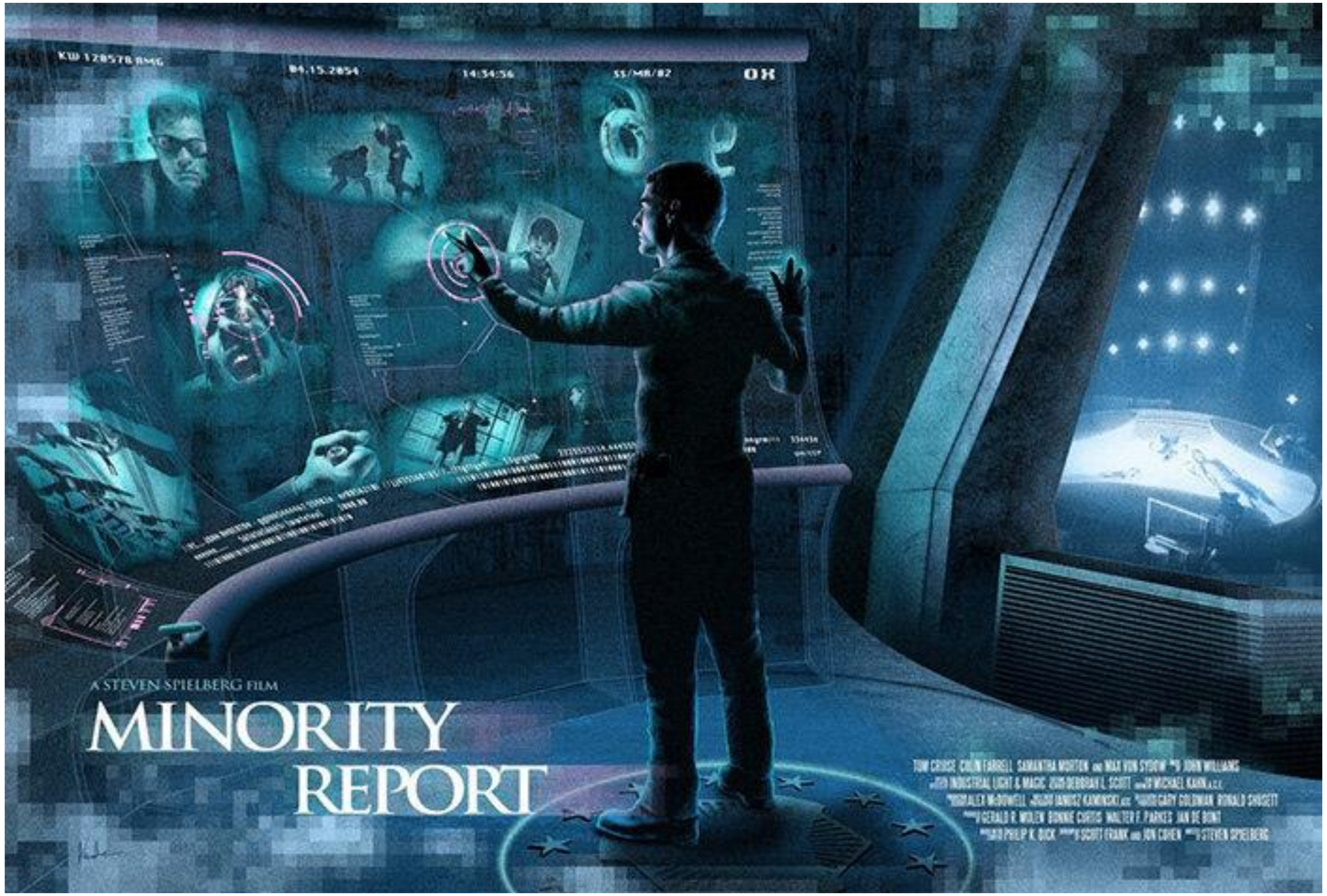

Global Governance & Open Standards
(enable interoperability)
Domain data owned & served by cross-functional team
New Data Domains can be created correlating data from other domains
Domain oriented data served as a product for any other domain to use
Domain's internal data pipeline uses common self-serve data infra
Data infra engineers
Data Infra as a Platform
(storage, pipeline, catalogue, Access control, etc.)

Centralization

Decentralization

Data Mesh in theory

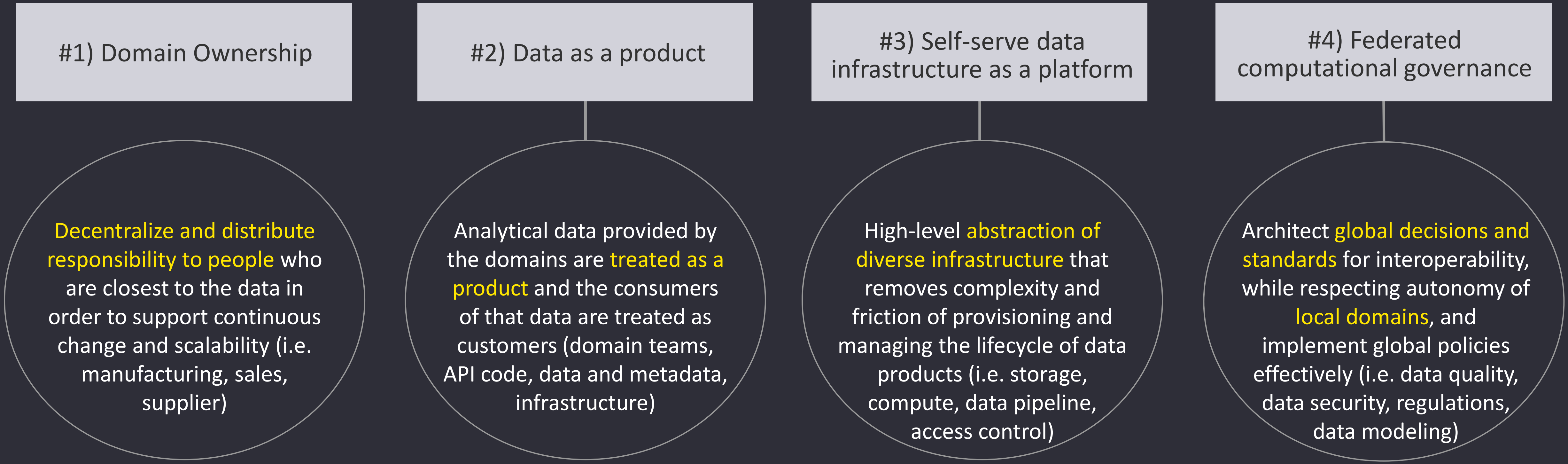
Lots of things sound great in theory…


KW 128578 RMG
04.15.2054
14:34:56
OX
A STEVEN SPIELBERG FILM
MINORITY
REPORT
TOM CRUISE COLIN FARRELL SAMANTHA MORTON AND MAX VON SYDOW JOHN WILLIAMS
INDUSTRIAL LIGHT & MAGIC DEBORAH L. SCOTT MICHAEL KAHN A.C.E.
ALEX McDOWELL JANUSZ KAMINSKI A.S.C. GARY GOLDMAN RONALD SHUSETT
GERALD R. MOLEN BONNIE CURTIS WALTER F. PARKES JAN DE BONT
PHILIP K. DICK SCOTT FRANK AND JON COHEN STEVEN SPIELBERG

Data Mesh - Overview

Data mesh is an intentionally designed distributed data architecture, under centralized governance and standardization for interoperability, enabled by a shared and harmonized self-serve data infrastructure

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

High-level abstraction of diverse infrastructure that removes complexity and friction of provisioning and managing the lifecycle of data products (i.e. storage, compute, data pipeline, access control)

#4) Federated computational governance

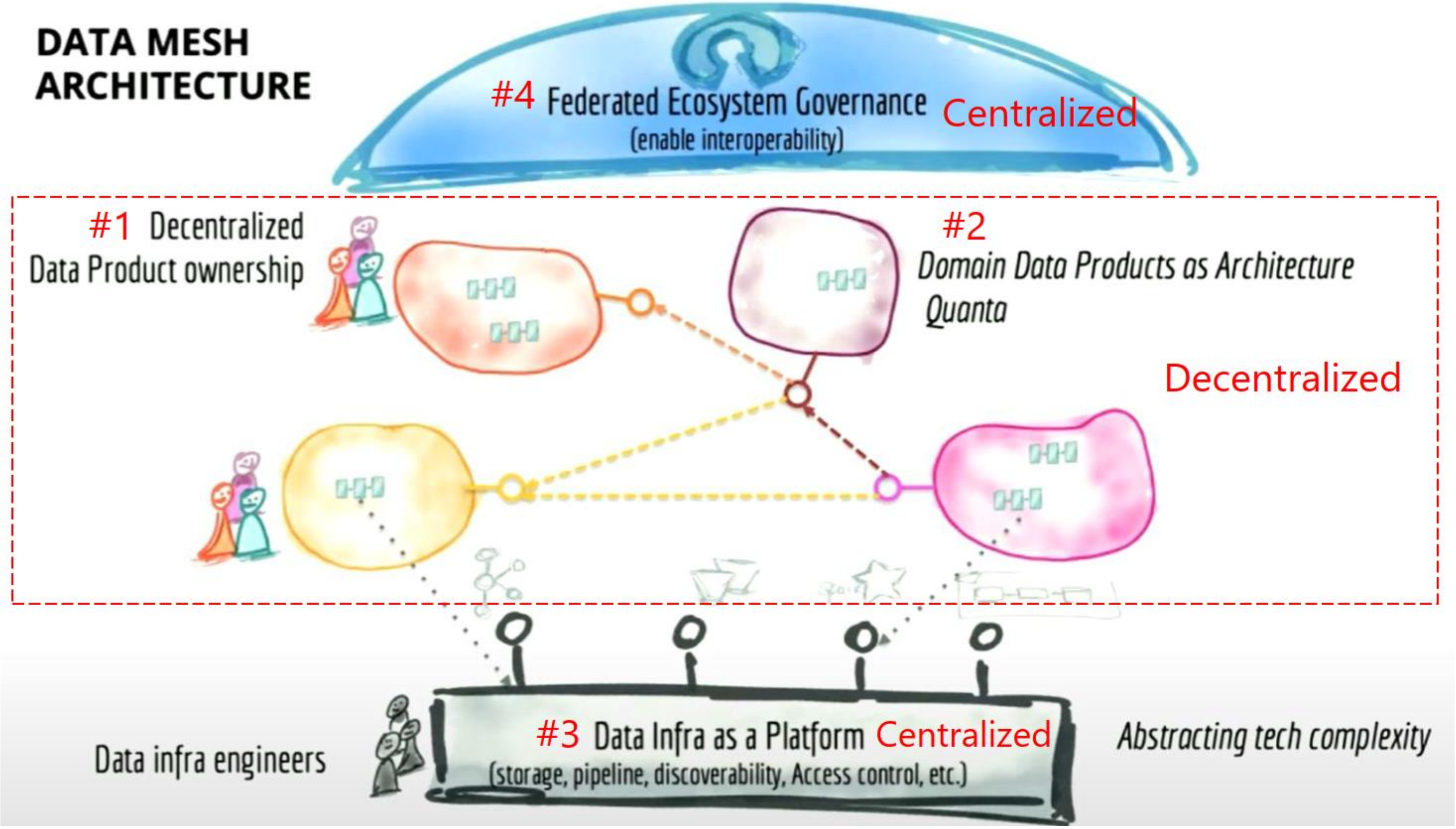
Architect global decisions and standards for interoperability, while respecting autonomy of local domains, and implement global policies effectively (i.e. data quality, data security, regulations, data modeling)

Data Mesh

Data Mesh is a concept, not a product

It's a mindset shift where you go from:

- Centralized ownership to decentralized ownership
- Pipelines as first-class concern to domain data as first-class concern
- Data as a by-product to data as a product
- A siloed data engineering team to cross-functional domain-data teams
- A centralized data lake/warehouse to an ecosystem of data products


DATA MESH ARCHITECTURE
#4 Federated Ecosystem Governance
(enable interoperability)
Centralized
#1 Decentralized Data Product ownership
#2 Domain Data Products as Architecture Quanta
Decentralized
Data infra engineers
#3 Data Infra as a Platform Centralized
(storage, pipeline, discoverability, Access control, etc.)
Abstracting tech complexity

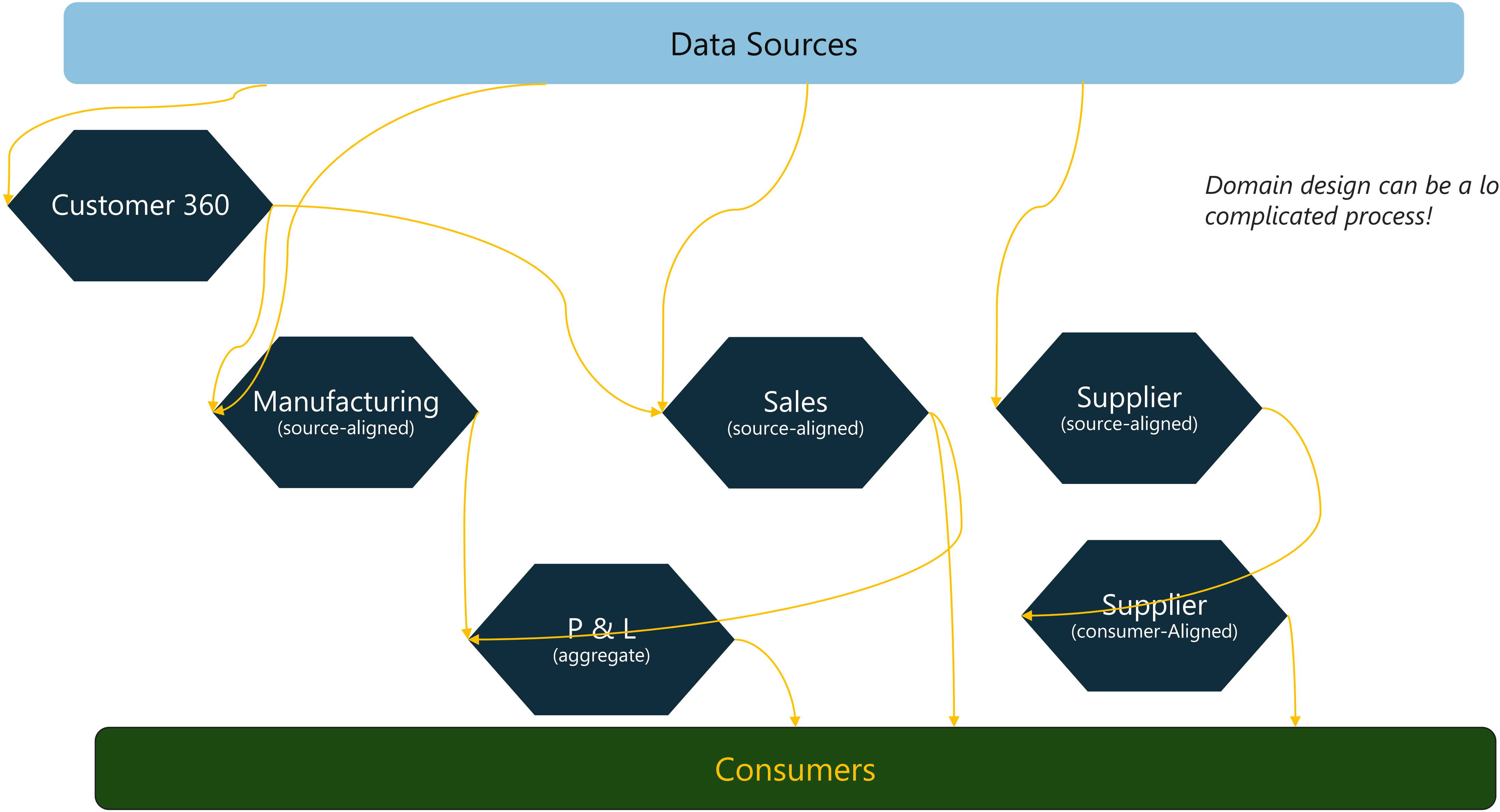
Credit to Zhamak Dehghani, Slack: data-mesh-learning.slack.com

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

Data Mesh – Logical Architecture



Domain design can be a long and complicated process!

Concerns with Data Mesh

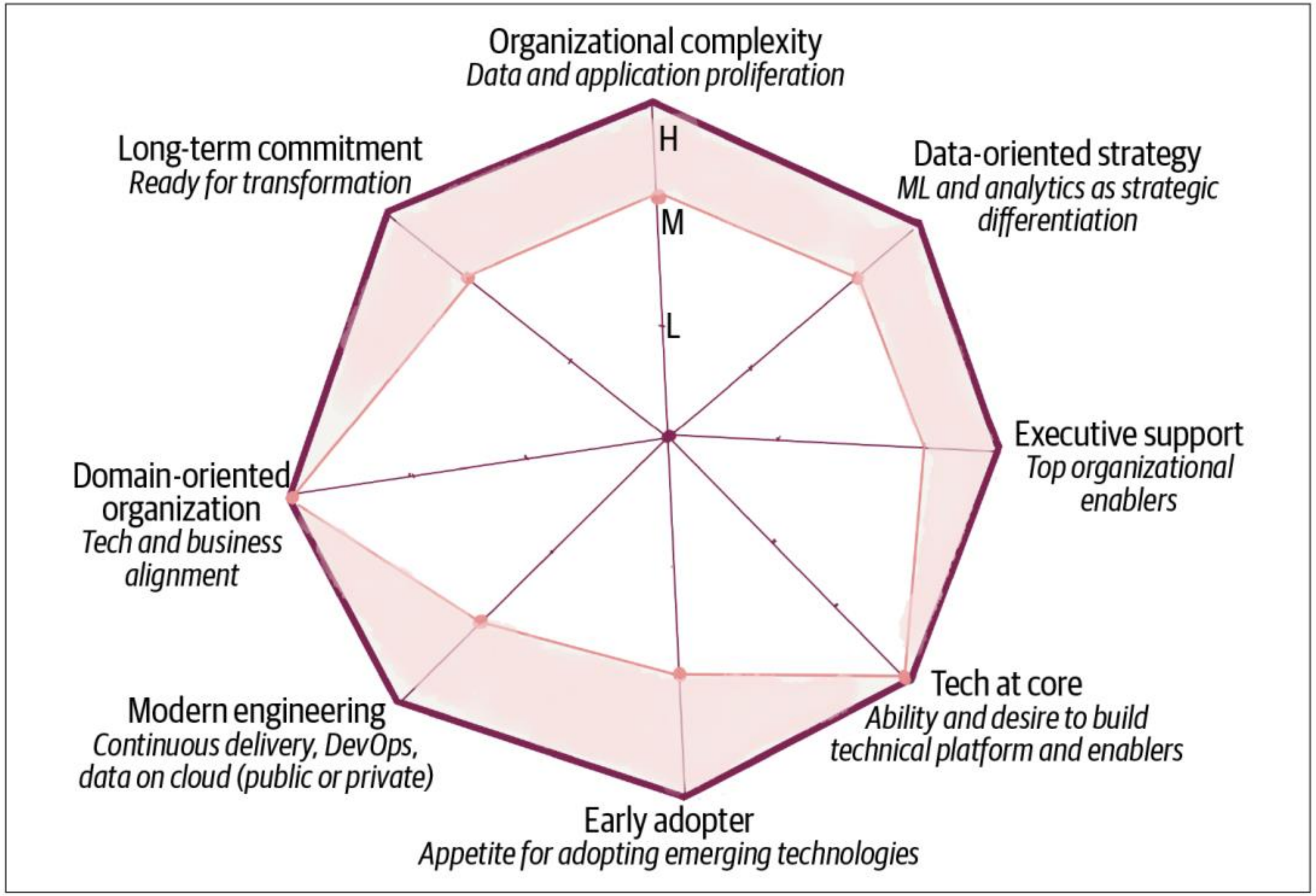
- No standard definition of a data mesh
- Huge investment in organizational change and technical implementation
- Performance of combining data from multiple domains
- Duplication of data for performance reasons
- Getting quality engineering people for each domain
- Inconsistent technical implementations 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

Key for a successful Data Mesh

- Have current pain points
- A company culture open to change
- Experience people
- Be aware of Data Mesh concerns
- Don’t just jump on the latest buzzword
- Don’t listen to vendors
- Don’t go strictly “by the data mesh book”
- Have a very long runway

Should you adopt data mesh today?

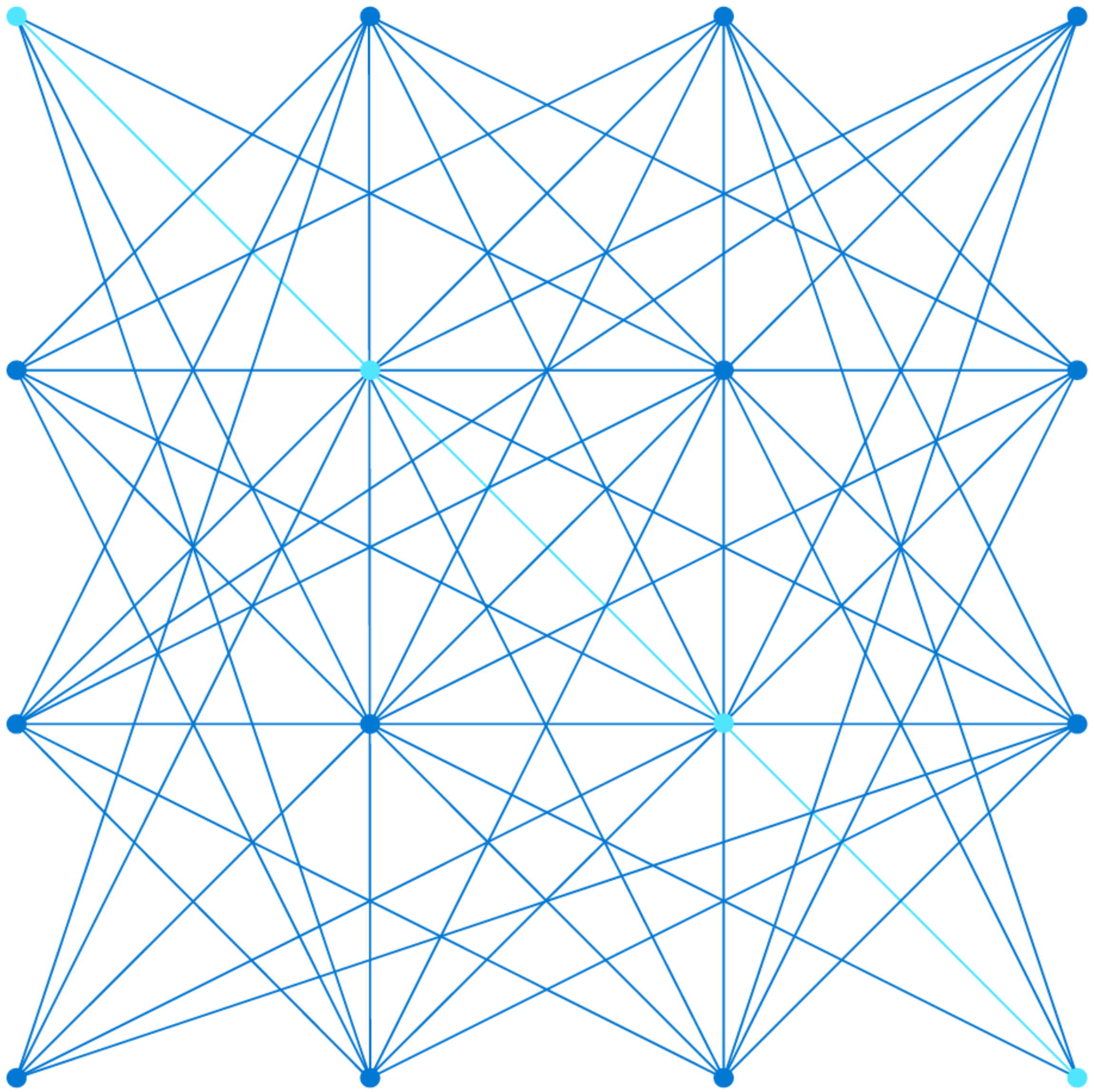


Need to score medium or high in ALL categories

From book “Data Mesh – Delivering Data-Driven Value at Scale” by Zhamak Dehghani

Microsoft Azure

Data Mesh on Azure



Enterprise Scale Analytics and AI (ESA)

Enterprise-scale is an architecture approach and reference implementation that enables effective construction and operationalization of landing zones on Azure, at scale and aligned with Azure Roadmap and Cloud Adoption Framework.

What is Enterprise Scale Analytics and AI?

A scalable analytics framework designed to enable customers building a data platform.

- *Supports multiple topologies ranging across Data Centric, Lakehouse, Data Fabric and Data Mesh*
- Based on inputs from PG and a diverse international group of specialists working with a range of customers
- Separate guidance tailored to Small-Medium and Large enterprises
- ~80% prescribed viewpoint with 20% client customization

Enterprise Scale Landing Zones is a prerequisite for Enterprise Scale Analytics since it is built on the core foundation of Enterprise Scale Landing Zones. Consisting of:

- Prescriptive architecture
- Designed by Subject Matter Experts
- Documented End to End Technical Solution
- Deployment Templates
- Operational Usage Model

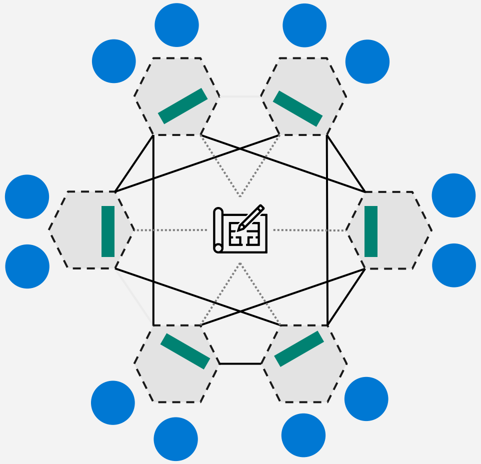
Data Mesh on Azure Resources

- Piethein Strengholt: Blog - Implementing Data Mesh on Azure (public), Blog – Data Mesh topologies (public), Book - Data Management at Scale: Best Practices for Enterprise Architecture (public)
- Cloud Adoption Framework: Azure data management and analytics scenario (public)
- Data Management & Analytics Scenario - Data Management Zone: Github (public)
- Data Management & Analytics Scenario - Data Landing Zone: Github (public)
- Enterprise-Scale - Reference Implementation: Github (public)
- Microsoft doc: A financial institution scenario for data mesh (public)

Governance Topologies : Different Approaches

Private
(do not share publicly)

Mesh Type 2

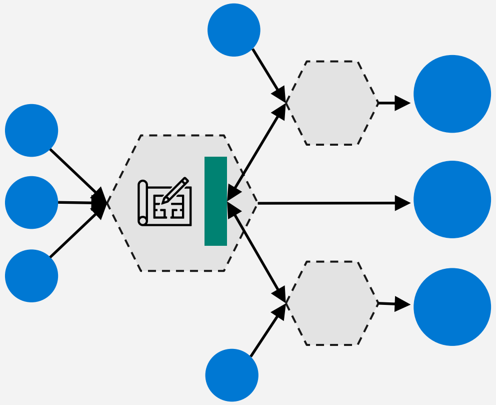


- Domains use the same technology
- Each domain has its own storage that is the same technology

Centralised
(Control)

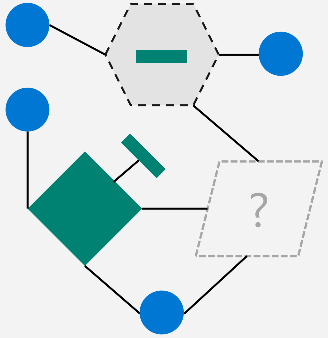
Distributed
(Agility)

Mesh Type 1



- Domains use the same technology
- Data is kept in one enterprise data lake with each domain getting its own container/folder

Mesh Type 3



- Domains can use any technology they want
- Each domain has its own storage that can be any technology

Data Fabric vs Data Mesh

If Data Fabric uses data virtualization, how is it different from Data Mesh:

- Usually only some of the data is virtualized, so still mostly centralized
- Not making data as a product (no contract with domains).
- Still have siloed data engineering team in IT

Or more specifically, violates 3 of the 4 Data Mesh principles:

1) Domain ownership
2) **Data as a product**
3) **Self-serve data infrastructure as a platform**
4) **Federated computational governance**

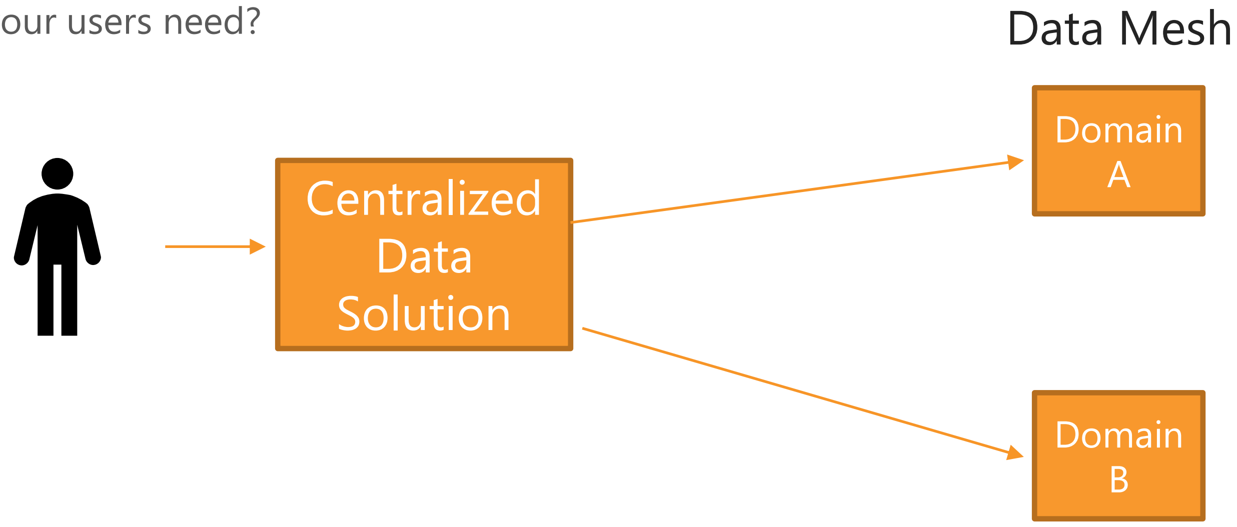
Future

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.

There will be a very small percentage of solutions that are 100% true to the data mesh concept (assuming mesh type 1 and 2 are true to the data mesh concept). *Ask ten people what a data mesh is and you will get eleven answers!*

Always ask: What do your users need?



1) **Domain ownership (90%)**
2) Data as a product (50%)
3) Self-serve data infrastructure as a platform (1%)
4) Federated computational governance (25%)

Rethinking the Data Mesh Architecture: Apply it Piecemeal (eckerson.com)

Feedback:

https://sqlb.it/?7106

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

Comparisons of Data Fabric and Data Mesh

Areas	Data Mesh	Data Fabric
Framework	Focus on data architecture	Focus on data architecture, semantic consumption, through the wide use of Ontologies
Governance	Multiple governance layers	Unified governance layer
Security	Data Products owning the domain data and applying security and governance applicable to the domain	Focuses on a comprehensive Unified Security model across the entire Data Ecosystem
Consistency	Complex mechanics to ensure consistency of data	Focused on enabling and ensuring trust by applying automatic consistency
Implementation	Is complex, even to start a small implementation due to the need of understanding and segregating domain data	By far simpler, due to the inherent use of Data Virtualization, meta data and knowledge graphs