Building a Multi-Cloud Data Fabric for Analytics

James Serra

Data Platform Architecture Lead

EY

jamesserra3@gmail.com

Blog: JamesSerra.com

About Me

- EY, Data Platform Architecture Lead
- 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

Questions to ask customer

- Can you use the cloud?
- Is this a new solution or a migration?
- What is the skillset of the developers?
- Will you use non-relational data (variety)?
- How much data do you need to store (volume)?
- Is this an OLTP or OLAP/DW solution?
- Will you have streaming data (velocity)?
- Will you use dashboards and/or ad-hoc queries?
- Will you use batch and/or interactive queries?
- How fast do the operational reports need to run?
- Will you do predictive analytics?
- Do you want to use Microsoft tools or open source?
- What are your high availability and/or disaster recovery requirements?
- Do you need to master the data (MDM)?
- Are there any security limitations with storing data in the cloud?
- Does this solution require 24/7 client access?
- How many concurrent users will be accessing the solution at peak-time and on average?
- What is the skill level of the end users?
- What is your budget and timeline?
- Is the source data cloud-born and/or on-prem born?
- How much daily data needs to be imported into the solution?
- What are your current pain points or obstacles (performance, scale, storage, concurrency, query times, etc)?
- Are you ok with using products that are in preview?
- What are your security requirements? Do you need data sovereignty?
- Is data movement a challenge?

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

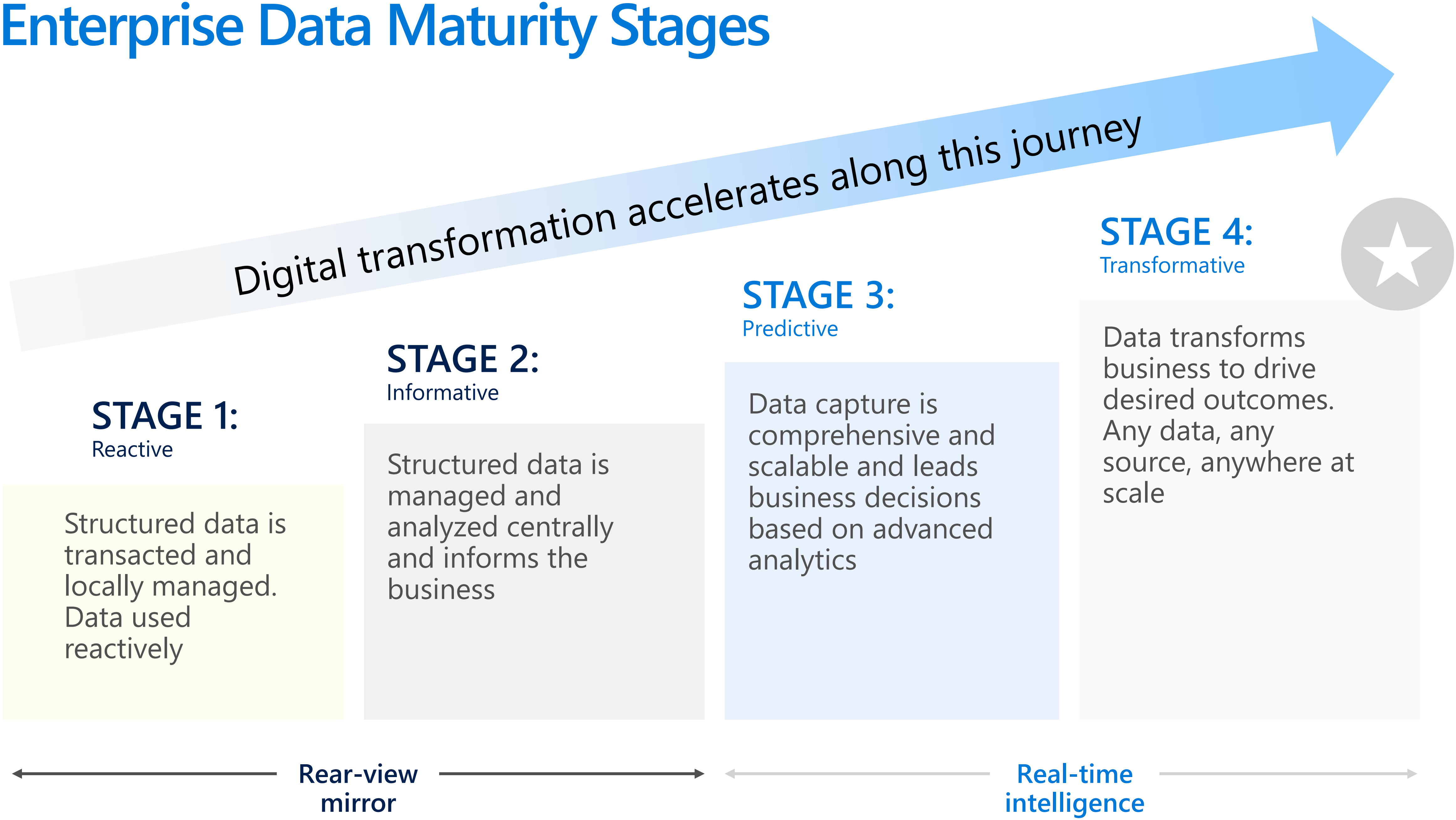
- Data scientists/Power users
- Batch processing
- Data refinement/cleaning
- ETL workloads
- Store older/backup data
- Sandbox for data exploration
- One-time reports
- Quick access to data
- Don’t know questions

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

Single-cloud vs Multi-cloud

Benefits of multi-cloud:

- Improved ability to meets SLA’s
- Reduced cost
- Reduced lock-in
- Capacity issues
- Missing features/products
- Data sovereignty

Single-cloud versus Multi-cloud | James Serra's Blog

Single-cloud vs Multi-cloud

Concerns:

- Performance
- Increasing the skillset
- Reduced interoperability
- Switching costs
- Management overhead
- Administrative complexity
- Least common denominator
- Exposure

Single-cloud versus Multi-cloud | James Serra's Blog

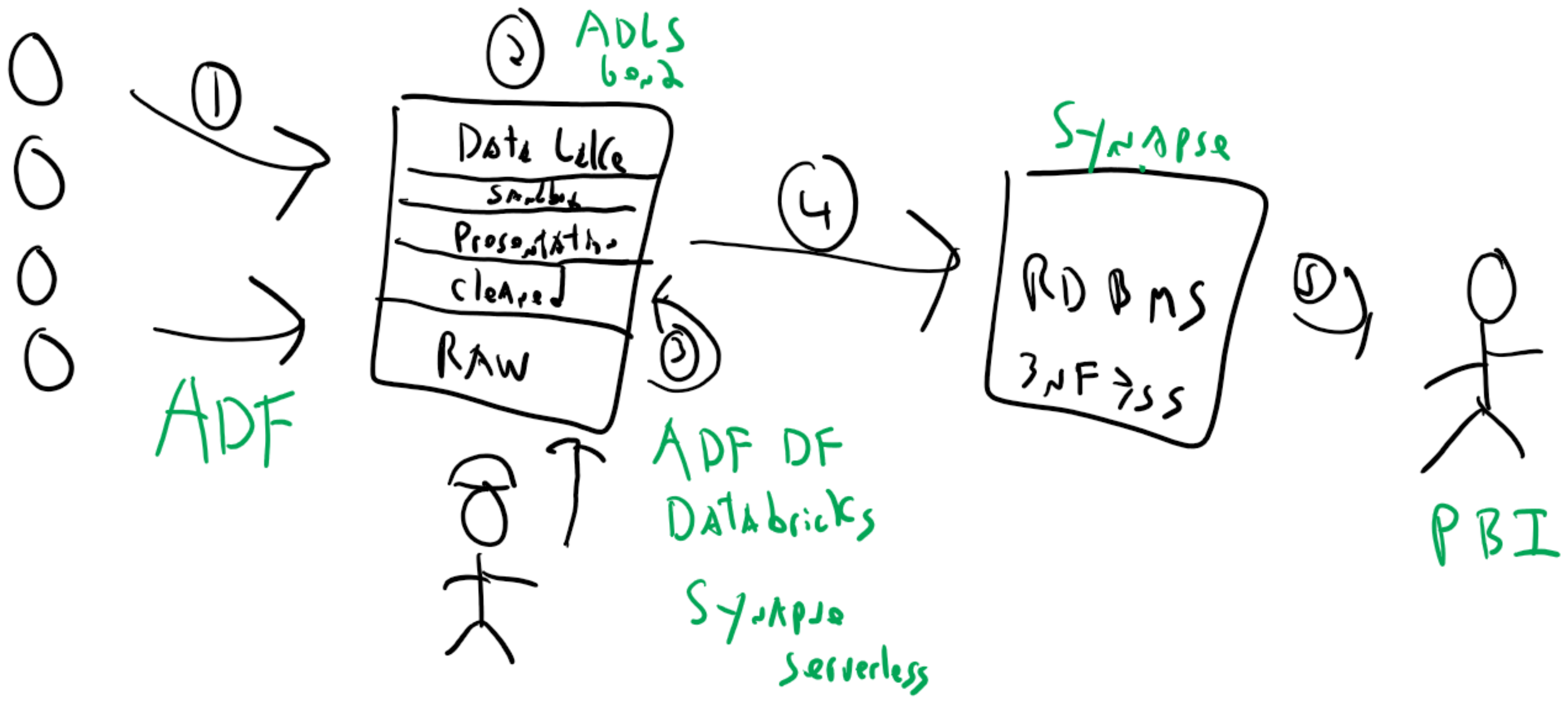
Populating a Data Warehouse

- Determine frequency of data pull (daily, weekly, etc)
- Full Extraction – All data (usually dimension tables)
- Incremental Extraction – Only data changed from last run (fact tables)
- How to determine data that has changed
  - Timestamp - Last Updated
  - Change Data Capture (CDC)
  - Partitioning by date
  - Triggers on tables
  - MERGE SQL Statement
  - Column DEFAULT value populated with date
  - 3rd-party product - striim
- Online Extraction – Data from source. First create copy of source:
  - Replication
  - Database Snapshot
  - Availability Groups
- Offline Extraction – Data from flat file

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:

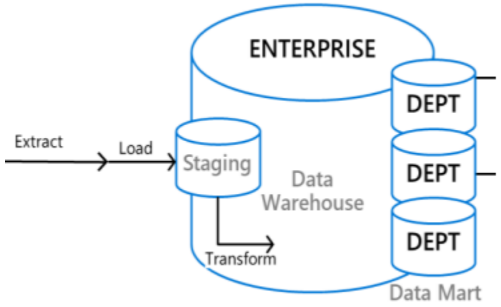
- Data access
- Data policies
- Metadata catalog/Lineage
- MDM
- Data virtualization
- Data scientist tools
- APIs
- Building blocks/Services
- Products

Data Fabric defined

Data Lakehouse

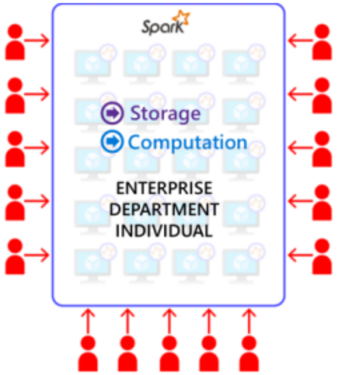
Late 1980s

Data Warehouse



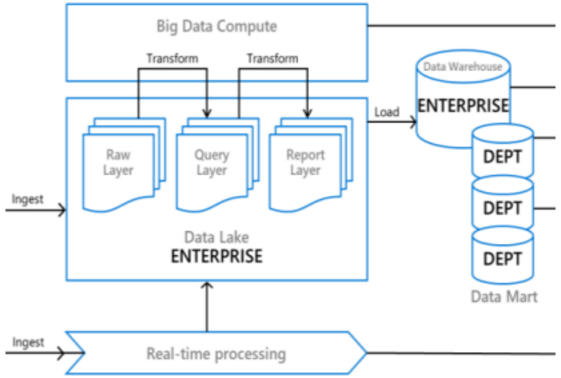
Late 2000s

Data Lake



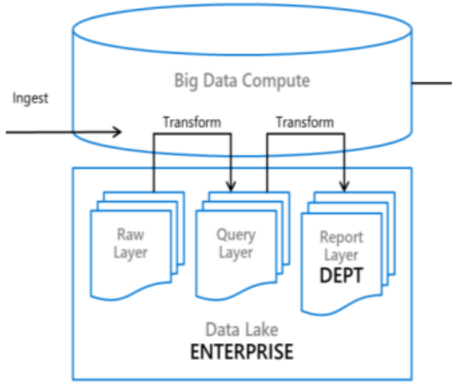
Mid 2010s

Cloud Data Platform



2020

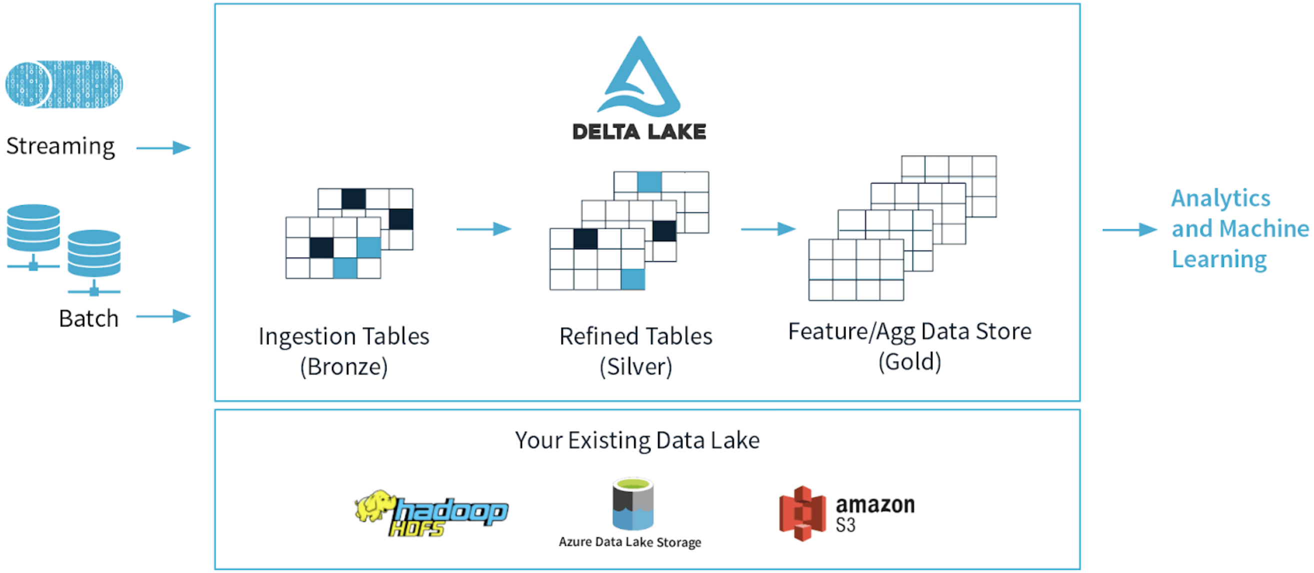
Data Lakehouse



Delta Lake

Top features:

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

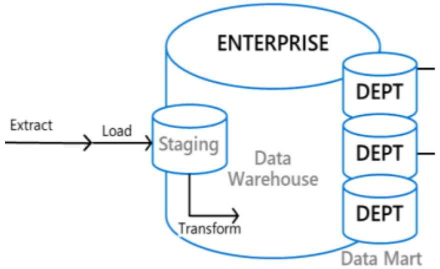
Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics

Concerns skipping relational database

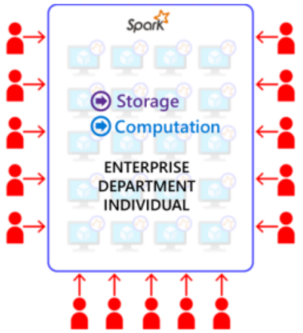
- Speed: Relational databases faster, especially MPP
- Security: No RLS, column-level, dynamic data masking
- Complexity: Metadata separate from data, file-based world
- Missing features: Referential integrity, TDE, workload management; other features require locked into Spark

Data Mesh

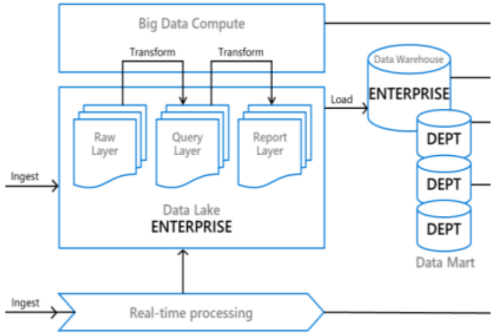
Late 1980s
Data Warehouse



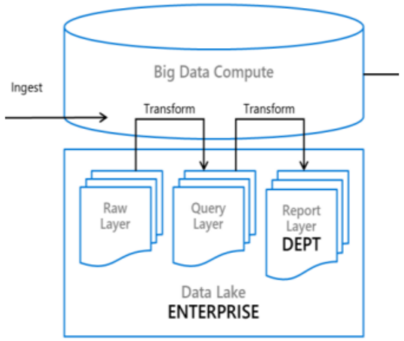
Late 2000s
Data Lake



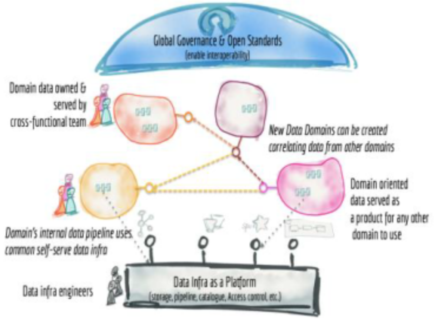
Mid 2010s
Cloud Data Platform



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.)

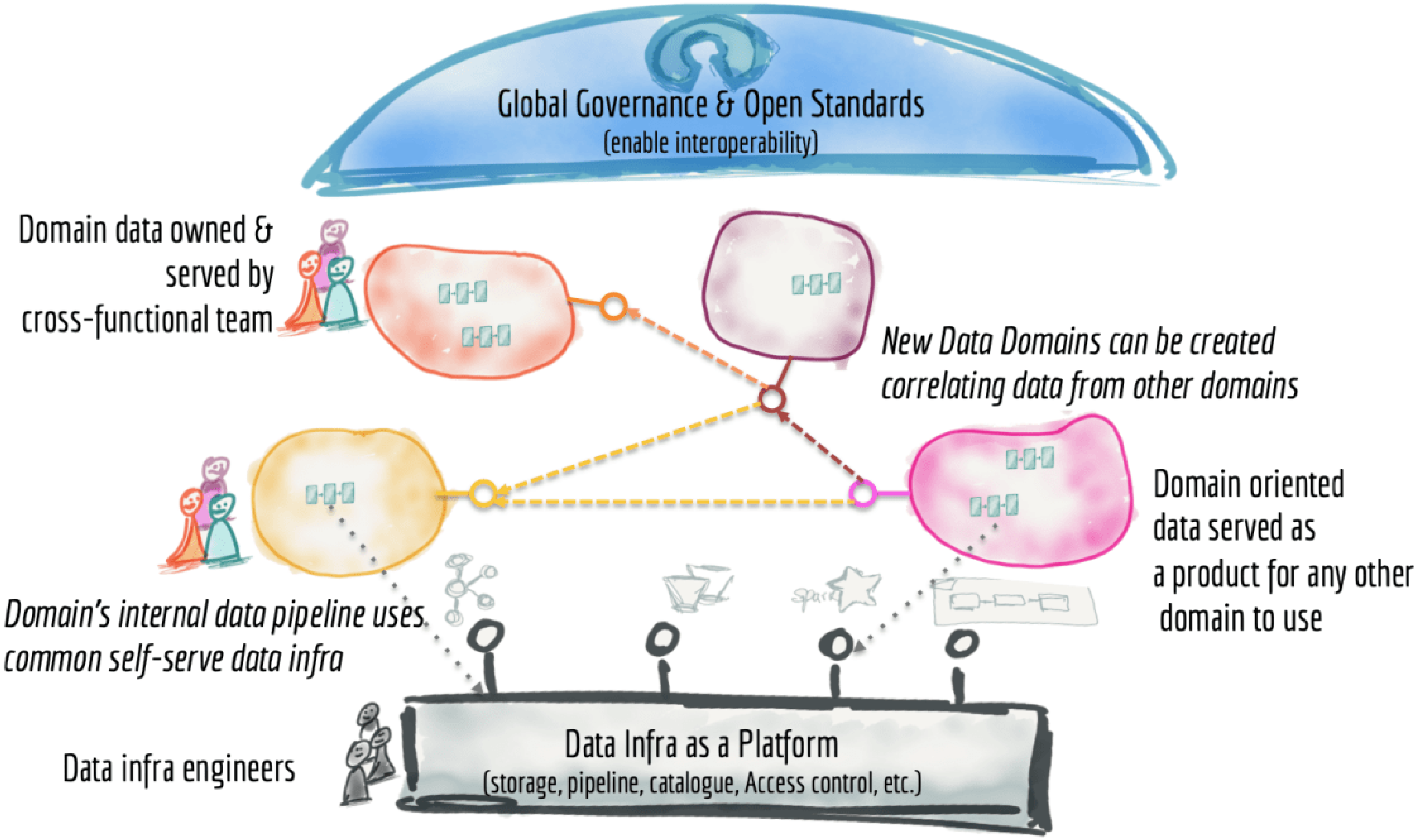
Centralization

Decentralization

Data Mesh

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



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.)
spark

Credit to Zhamak Dehghani

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

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

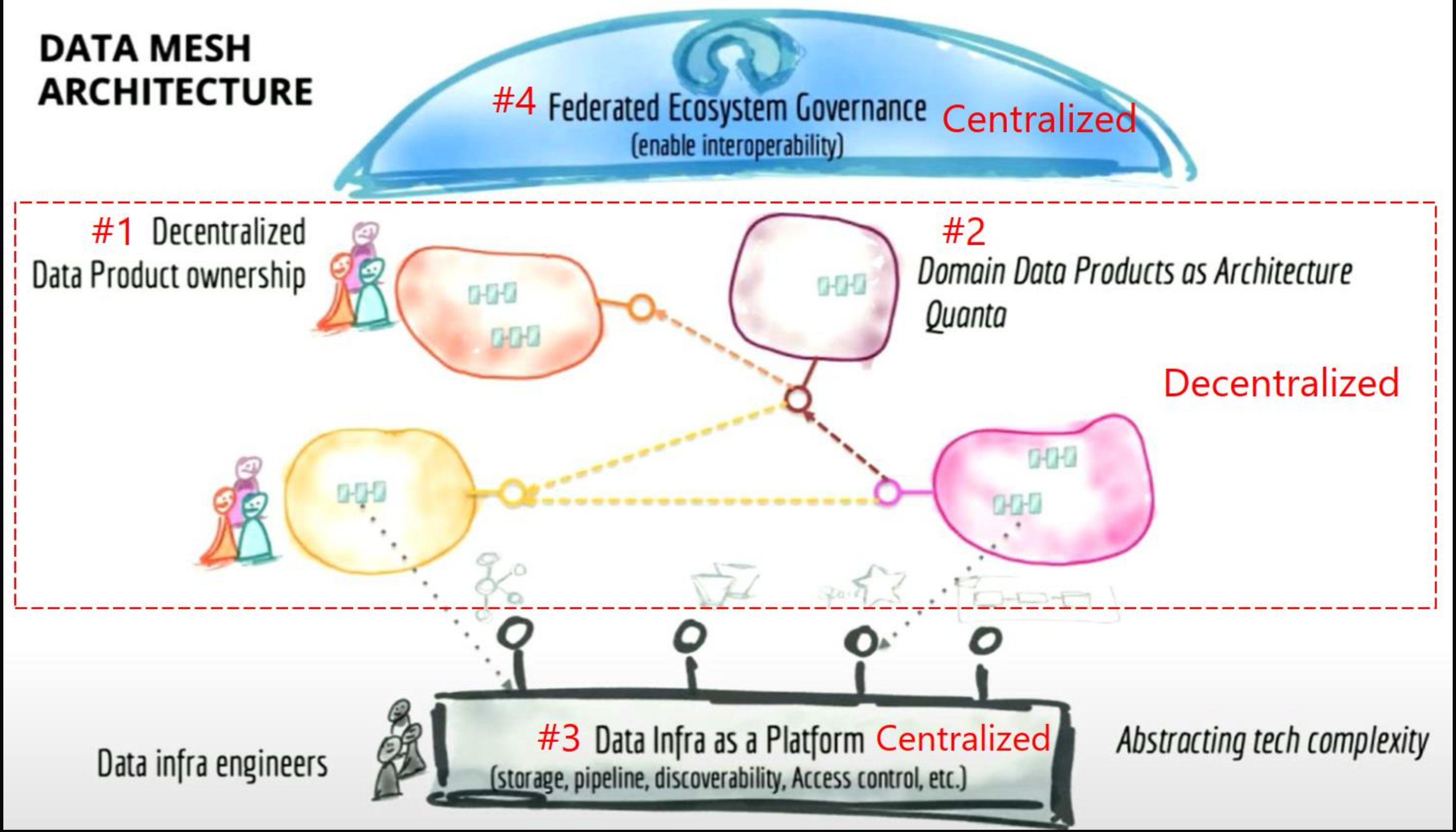
- Have current pain points
- A company culture open to change
- Experience people
- Be aware of Data Mesh concerns
- Don’t just jump on the latest buzzword
- Don’t listen to vendors
- Don’t go strictly “by the data mesh book”
- Have a very long runway

Real Data Mesh implementations

- Large banks
  - JPMC
  - Saxo Bank
  - JPMorgan Chase
- Intuit
- Adevinta
- HelloFresh
- DPG Media
- Max Schultze
- CMC Markets
- Kolibri Games

- Data Mesh Content

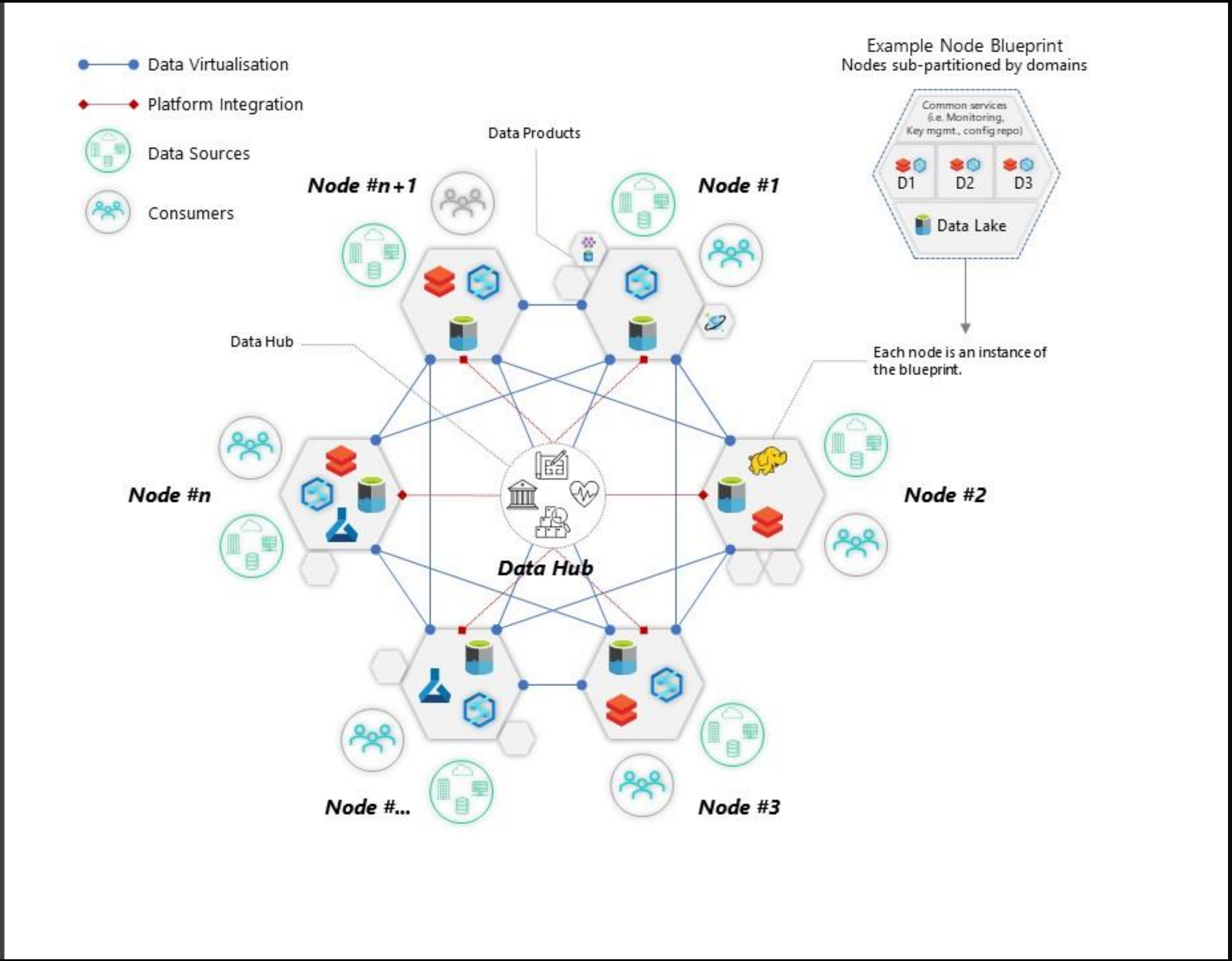
Microsoft Data Mesh


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

Microsoft Harmonized Mesh

Harmonised Mesh

- Azure Harmonised Mesh allows multiple groups within an organisation to operate their own analytics platform whilst adhering to common policies and standards.
- The central datahub hosts data catalogue, mesh wide audit capabilities, monitoring, and auxiliary services for automation.
- The central data platform group defines blueprints that encompass baseline security, policies, capabilities and standards.
- New nodes are instantiated based on these blueprints, which encompass key capabilities to enable enterprise analytics (i.e.. Storage, monitoring, key management, ELT, analytical engines, and automation)
- Node instances can be augmented to serve respective business requirements, i.e. deploying additional domains, customising domains and data products within the node.
- Nodes are typically split by either org-division, function, or region.



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

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

Q & A

James Serra, Data Platform Architecture Lead
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