What's the best Big Data Architecture for you?

Christoph Windheuser
Databricks
May 21 2026
INNOQ MeetUp Cologne

About me

- PhD in Machine Learning at ENST (Paris France), Carnegie Mellon University (Pittsburgh US) and Waseda University (Tokyo Japan)
- Different positions at SAP and Capgemini
- 4 years at ThoughtWorks, implementing Data Mesh Architectures with customers
- 4 years at Databricks, Sr. Manager Professional Services Central EMEA

The views and opinions on this talk are mine and not that of Databricks.

Agenda

- Why Big Data and Big Data Architectures (BDA)?
- What is a BDA and what are the requirements?
- A brief history of Data Architectures
- Different BDAs:
  - The Modern Data Stack
    - Data Lakehouse
    - Data Mesh
- Outlook: The Future of BDA

Agenda

- **Why Big Data and Big Data Architectures (BDA)?**
- What is a BDA and what are the requirements?
- A brief history of Data Architectures
- Different BDAs:
  - The Modern Data Stack
    - Data Lakehouse
    - Data Mesh
- Outlook: The Future of BDA

Why "Big Data"?

- Data Driven Companies are more successful!
- A Big Data Architecture is a prerequisite to Implementing AI and Agents in your organization

Data- driven companies are:

23x
more likely to grow customer base

19x
more likely to be profitable

Sources: Forrester, Forbes

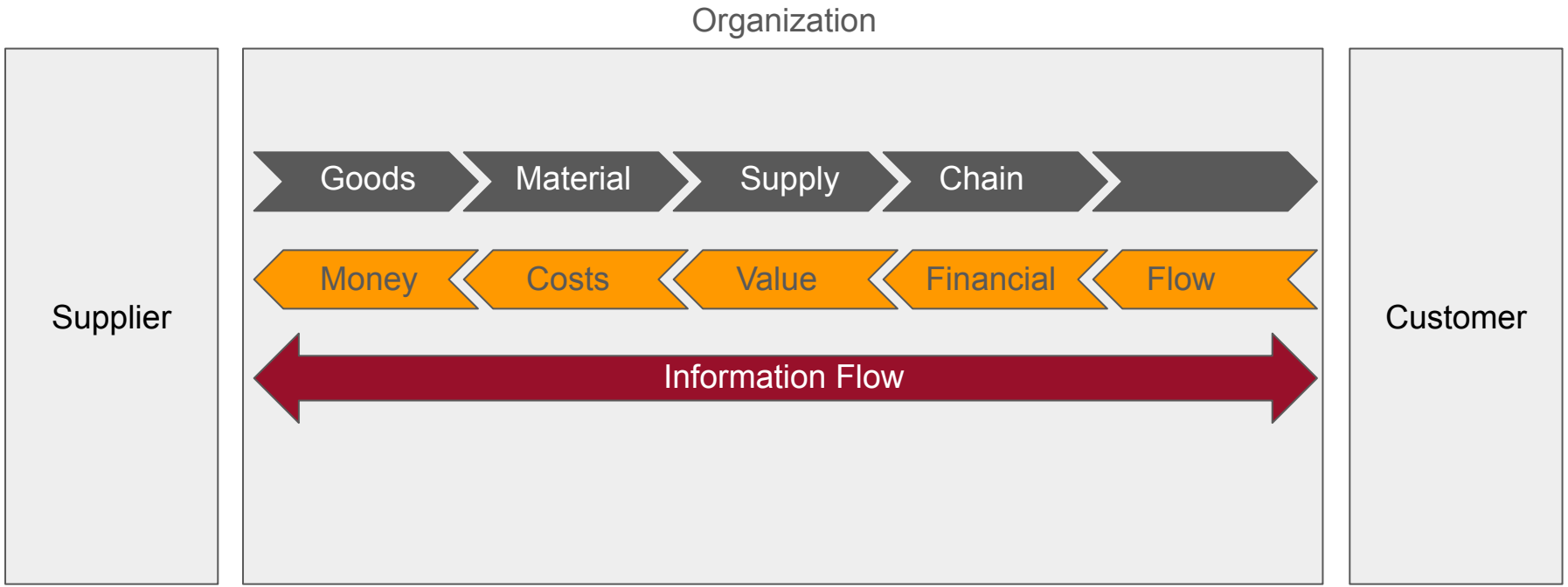
What is "Big Data"?

"Big Data" = "All Data"

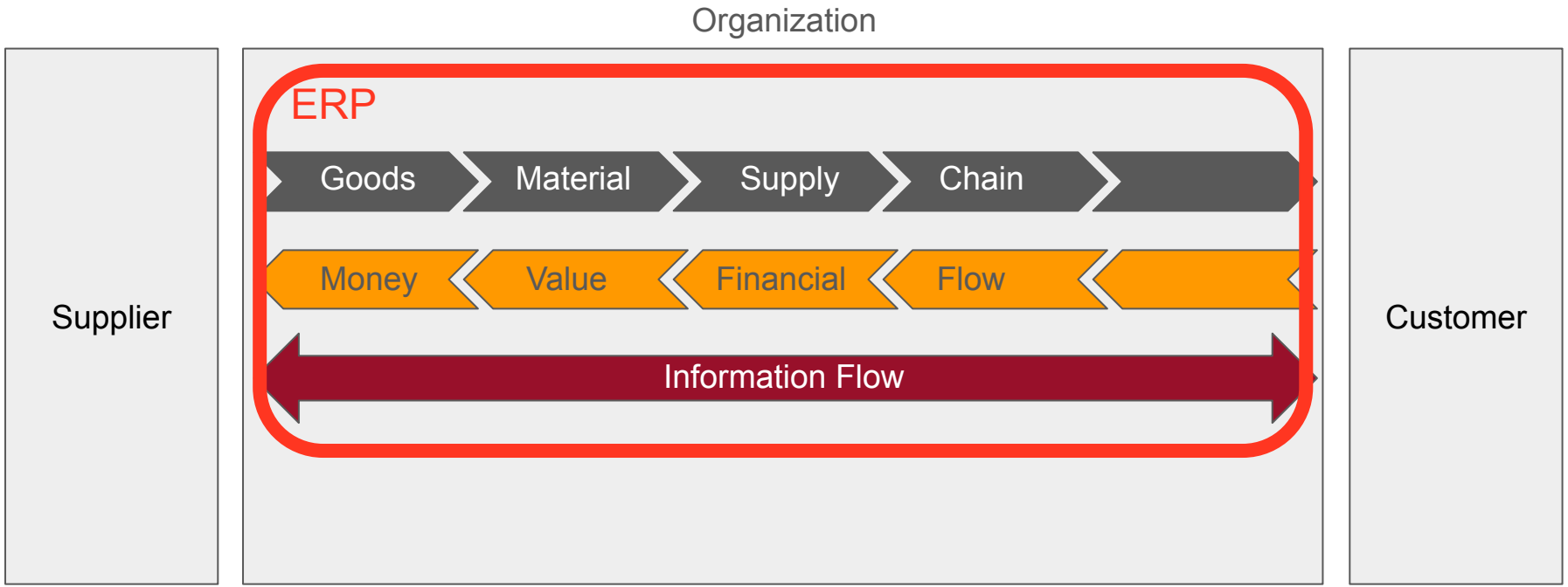
(acc. James Serra, Deciphering Data Architectures, O'Reilly)

- Structured (ex. tables)
- Semi-Structured (ex. XML, JSON)
- Unstructured (ex. text, pictures, audio, video)
- Batch Data
- Streaming Data

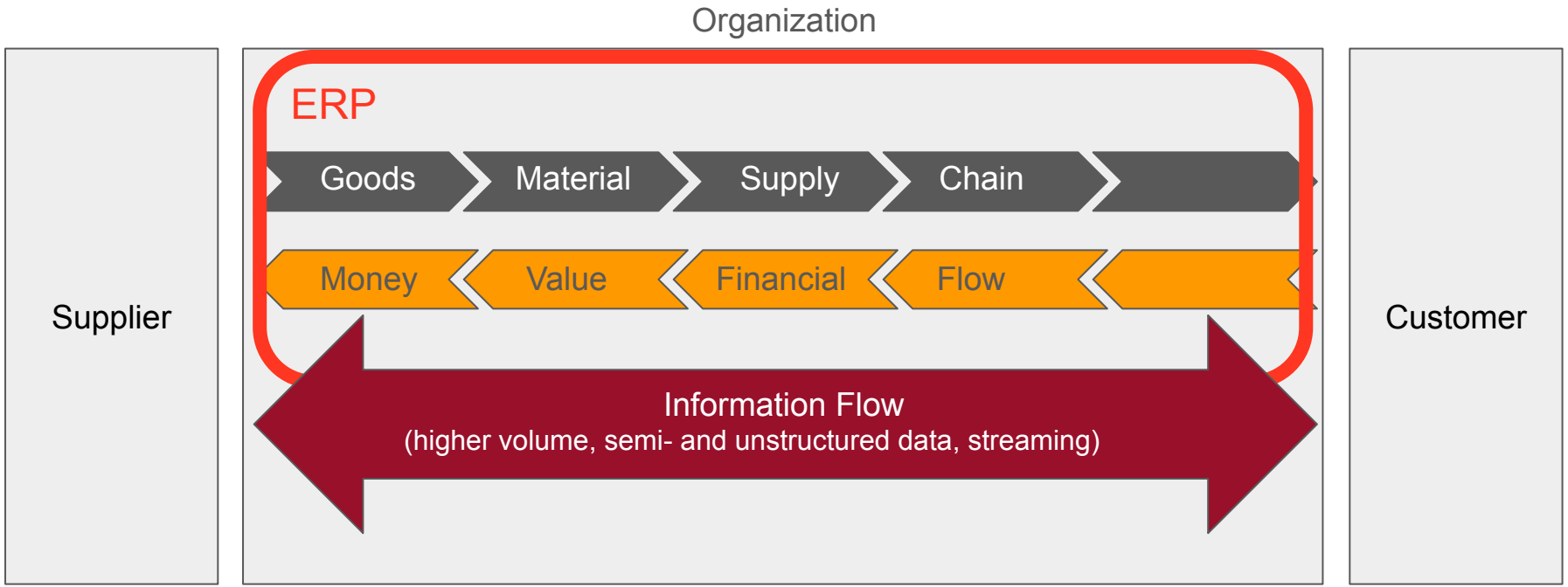
Why do we need a "Big Data Architecture" (BDA)?



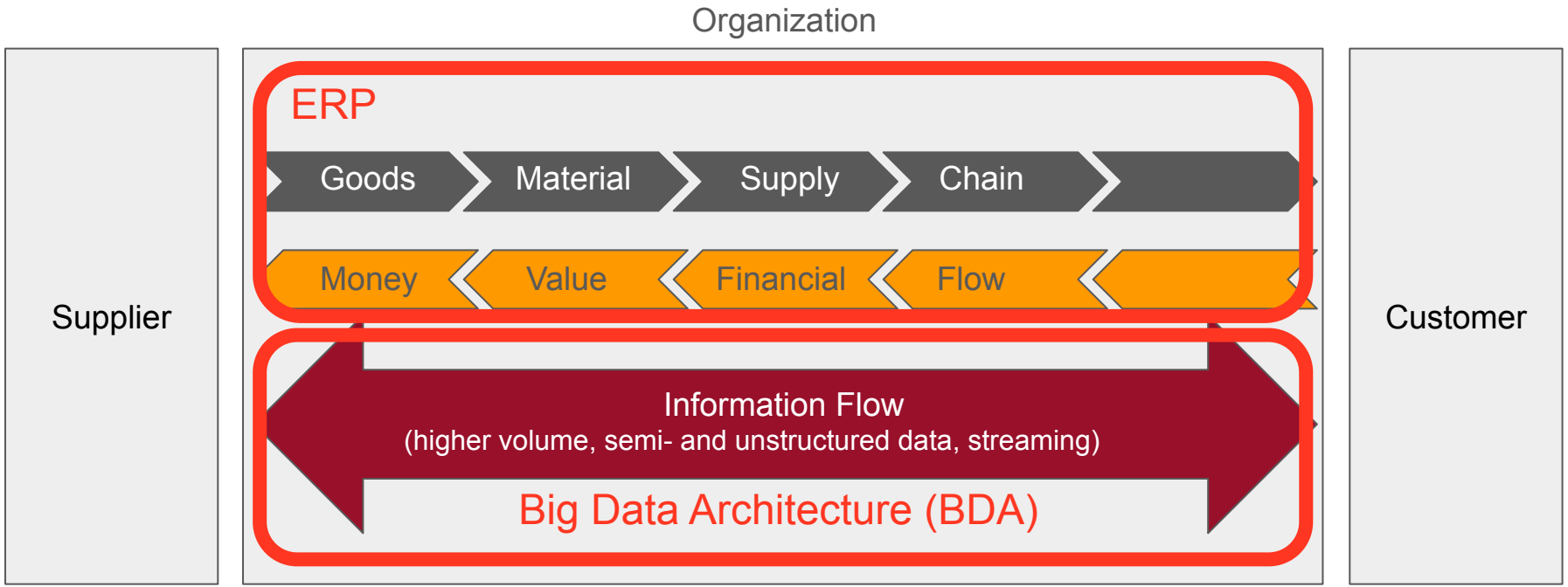
Why do we need a "Big Data Architecture" (BDA)?



Why do we need a "Big Data Architecture" (BDA)?



Why do we need a "Big Data Architecture" (BDA)?



Agenda

- Why Big Data and Big Data Architectures (BDA)?
- **What is a BDA and what are the requirements?**
- A brief history of Data Architectures
- Different BDAs:
  - The Modern Data Stack
    - Data Lakehouse
    - Data Mesh
- Outlook: The Future of BDA

What is a BDA and what are the Requirements?

A Big Data Architecture must provide the *right information at the right time at the right place for persons and applications, regardless of the type and size* of the data.

1. "Unlimited" storage
2. "Unlimited" compute
3. Capable to work with all kind of data (structured, semi- and unstructured)
4. Capable to support all use cases (analytics, streaming, DS, ML, GenAI, etc.)
5. Provides information via UI and API
6. Data must be trustworthy
7. Data must be discoverable
8. Enforces Data Government rules
9. Supports the company's data organization
10. Enables non-IT people to work with data *(data democratization)*

Agenda

- Why Big Data and Big Data Architectures (BDA)?
- What is a BDA and what are the requirements?
- **A brief history of Data Architectures**
- Different BDAs:
  - The Modern Data Stack
    - Data Lakehouse
    - Data Mesh
- Outlook: The Future of BDA

Important Milestones of Data Architectures

Data stored on tape or disks

- 1950

Edgar F. Codd proposed the Relational Database Model, SQL was invented

- 1970

RDBMS (OLTP)

- IBM
- ORACLE
- MS SQL Server

on-prem Data Warehouses (OLAP)

- Terradata
- ORACLE
- SAP BW

Cloud-based data lakes

- AWS (2006)
- GCP (2008)
- Azure (2010)
- People talking about "Big Data"

Massive Parallel Compute

- MapReduce (2003 - Google)
- Hadoop (2006 - Yahoo)
- Apache Spark (2009 - Berkeley)

The Modern Data Stack

- Data Mesh (2019)
- Lakehouse (2020)

Agenda

- Why Big Data and Big Data Architectures (BDA)?
- What is a BDA and what are the requirements?
- A brief history of Data Architectures
- **Different BDAs:**
  - The Modern Data Stack
    - Data Lakehouse
    - Data Mesh
- Outlook: The Future of BDA

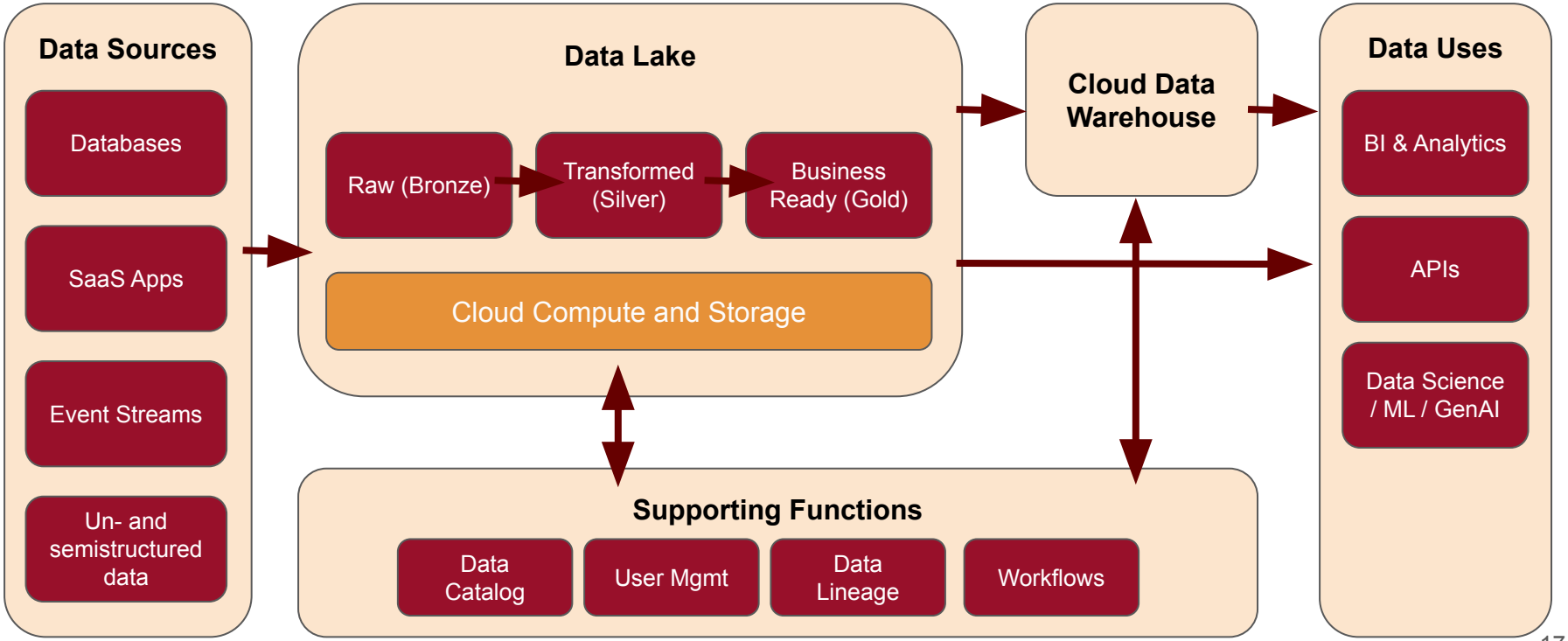
The Modern Data Stack

What is a "Modern Data Stack"?

Definition: ***A Modern Data Stack (MDS) is a collection of cloud-based tools and technologies used to gather, store, process, and analyze data in a scalable, efficient, and cost-effective way.***
(altexsoft https://www.altexsoft.com/blog/modern-data-stack/)

- Cloud-based
- Best-of-Breed
- Scalable
- Separation of Storage and Compute
- Always has a Data Lake
- Can have a Cloud-based Data Warehouse

Modern Data Stack Architecture



Characteristics of a Modern Data Stack

Pros

- Scalable
- Cost effective (consumption based)
- Universal, works with all kind of data (tabular, semi- and unstructured data)
- Supports all kind of use cases (BI & Analytics, ML, Streaming, etc.)

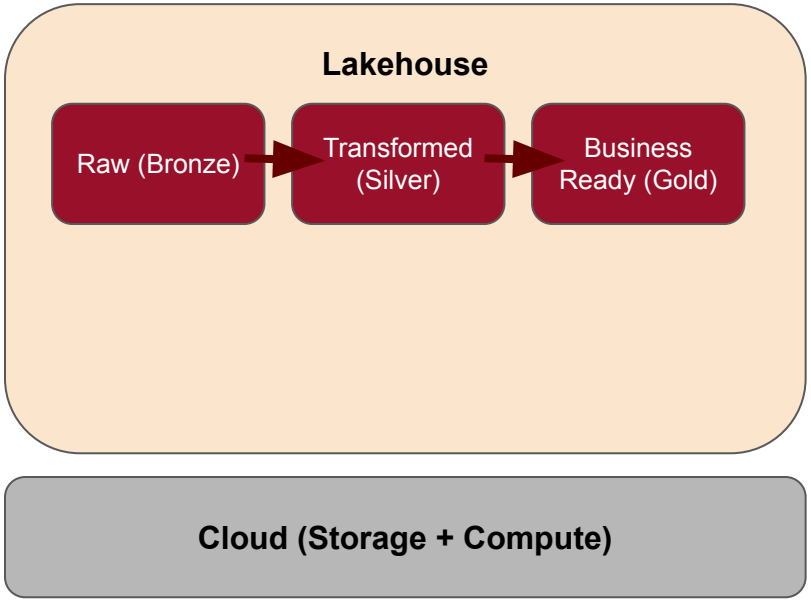
Cons

- Complex Architecture
- Separation of Data Lake and Data Warehouse can cause problems (synchronization, data stallness, no central security and data governance)

Data Lakehouse

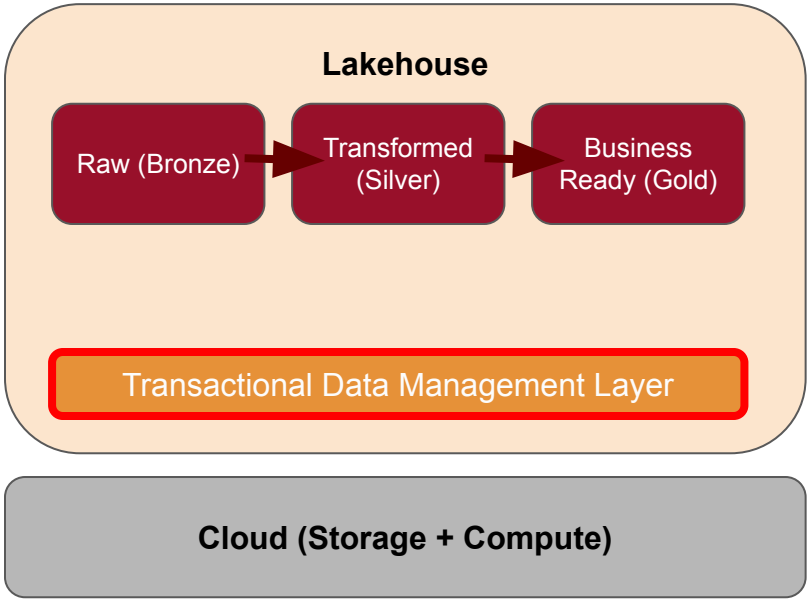
What is a Lakehouse?

A "Lake-House" is a combination of a Data Lake and a Data Warehouse. The concept was invented by Databricks in 2020.



What is a Lakehouse?

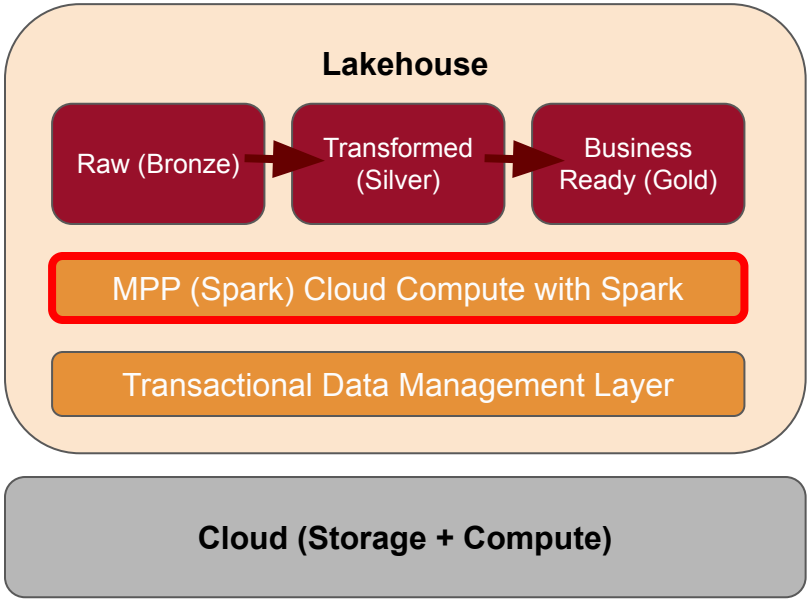
A "Lake-House" is a combination of a Data Lake and a Data Warehouse. The concept was invented by Databricks in 2020.



- Put a data management layer on top of a Data Lake with **transactional safety** (ACID properties = Atomicity, Consistency, Isolation, Durability) like Delta Lake, Apache Hudi or Apache Iceberg to get a ***Transactional Data Lake.***

What is a Lakehouse?

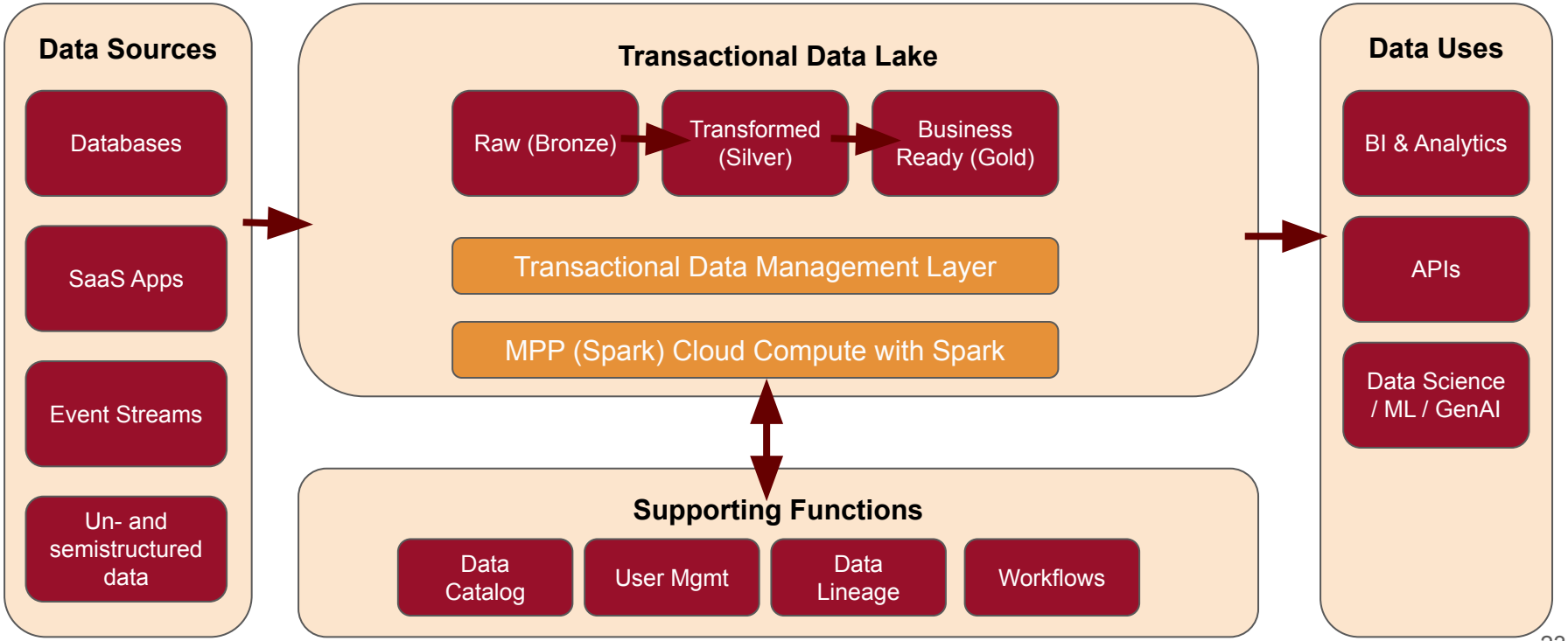
A "Lake-House" is a combination of a Data Lake and a Data Warehouse. The concept was invented by Databricks in 2020.



- Use a cloud-based ***massive parallel processing*** **(MPP)** framework like Spark SQL to get unlimited compute scalability and SQL performance comparable with state-of-the-art data warehouses.

- Put a data management layer on top of a Data Lake with **transactional safety** (ACID properties = Atomicity, Consistency, Isolation, Durability) like Delta Lake, Apache Hudi or Apache Iceberg to get a ***Transactional Data Lake.***

Lakehouse Architecture



MPP = Massive Parallel Processing

Characteristics of a Lakehouse

Pros

- Simpler architecture compared to the Modern Data Stack
- All Data in one Data Lake makes Data Governance and Metadata Management much easier
- Can process all kinds of data
- Out-of-the-box solutions available

Cons

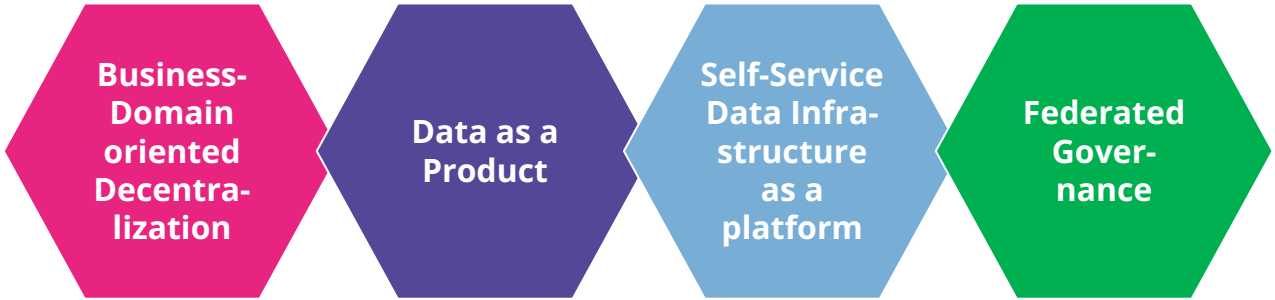
- Technology led approach, does not provide guidance to a organizational design (in particular a distributed approach for large organisations)

Data Mesh

What is a Data Mesh?

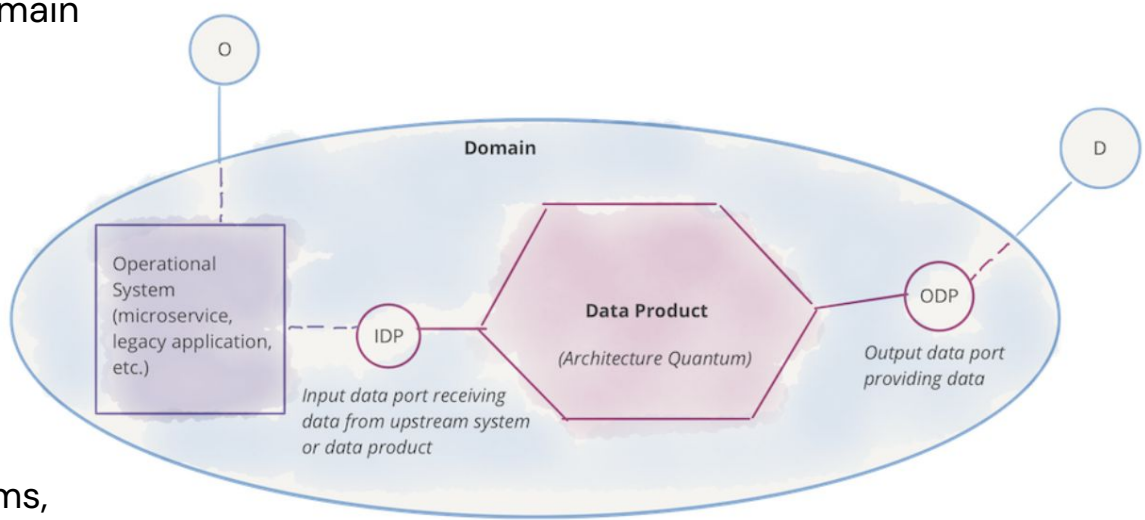
Developed 2019 by Zhamak Dehghani during her time at ThoughtWorks.

- More an organizational approach than a Big Data Architecture pattern.
- Decentralized Data organization
- Data is treated as a product



Data as a Product

- Decentralized managed by a Domain Data Product Owner
- Discoverable
- Addressable
- Secure (governed by access control)
- Trustworthy
- Interoperable
- Polyglot data ports (event streams, batch files, relational tables, graphs, etc.)



Picture from: *How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh*
Zhamak Dehghani, martinfowler.com, 2019

Characteristics of a Data Mesh

Pros

- Decentralized approach organized by Business Domains makes it interesting for large organizations
- IT or Analytics team is not longer a bottleneck
- Data as a Product is moving data ownership to business domains

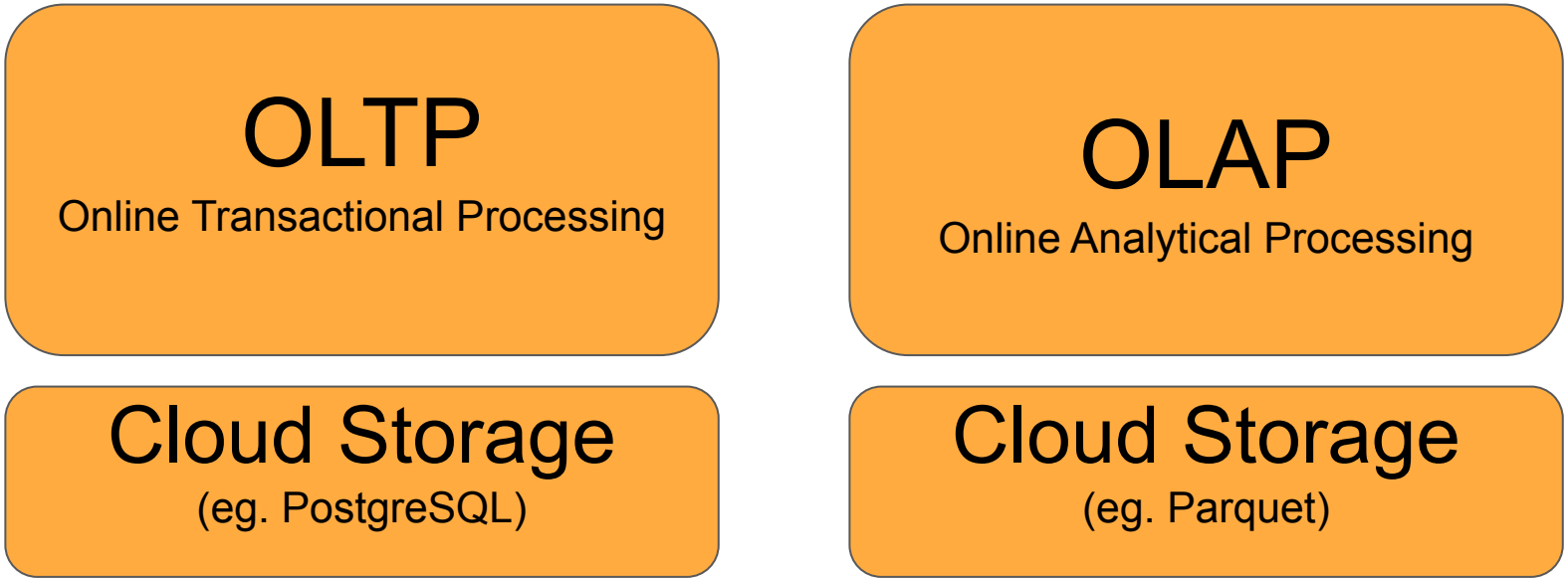
Cons

- Data Mesh does not say anything on how to implement it (technology agnostic)
- No out-of-the-box solutions from vendors available
- Implementing a Data Mesh is foremost a business trans-formation project and less a technical IT project

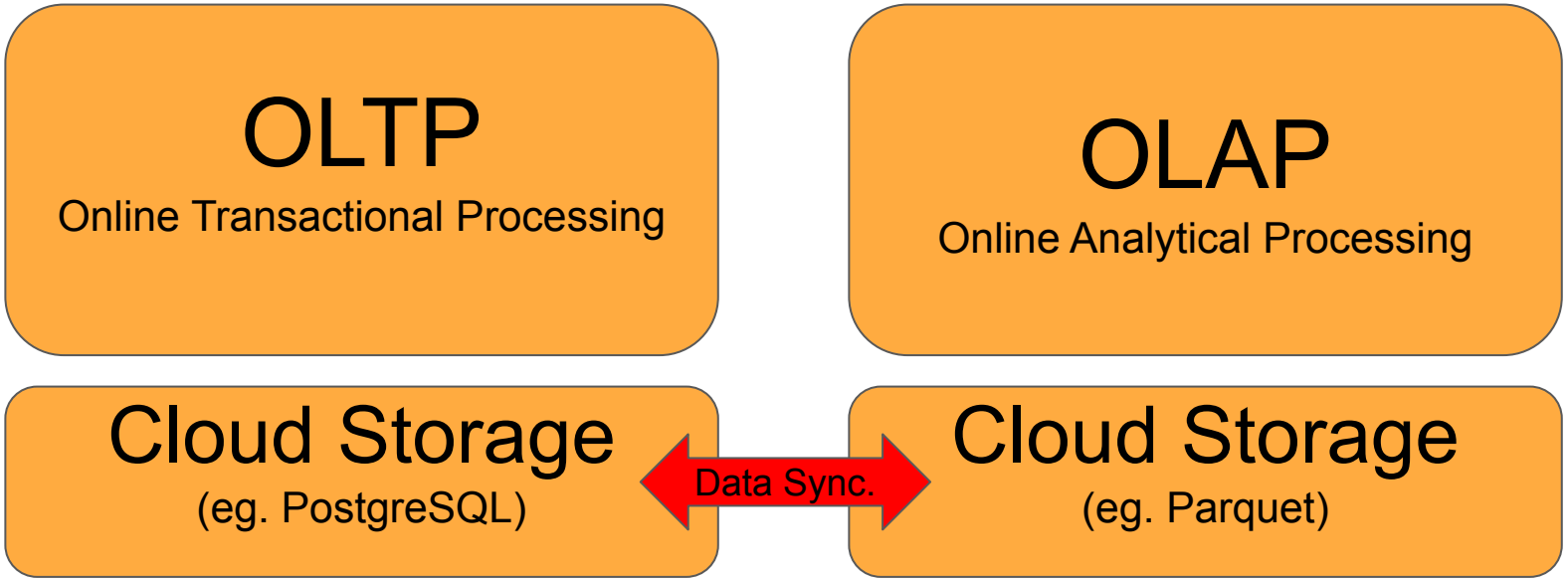
Agenda

- Why Big Data and Big Data Architectures (BDA)?
- What is a BDA and what are the requirements?
- A brief history of Data Architectures
- Different BDAs:
  - The Modern Data Stack
    - Data Lakehouse
    - Data Mesh
- **Outlook: The Future of BDA**

OLTP - OLAP


OLTP
Online Transactional Processing
OLAP
Online Analytical Processing
Cloud Storage
(eg. PostgreSQL)
Cloud Storage
(eg. Parquet)

HTAP (Hybrid Transactional/Analytical Processing - in research)



Data Democratisation: *"Talk to your Data"*

"Show me the global sales figure of our product xyz per country in the last quarter"

Use Case Example: Translation of SQL dialects for DB migrations

*SQL Scripts from **Vendor A:***

- *Script 1*
- *Script 2*
- *Script 3*
- ..
- ..
- ..
- ..
- ..
- ..
- ..
- ..
- ..
- *Script n*

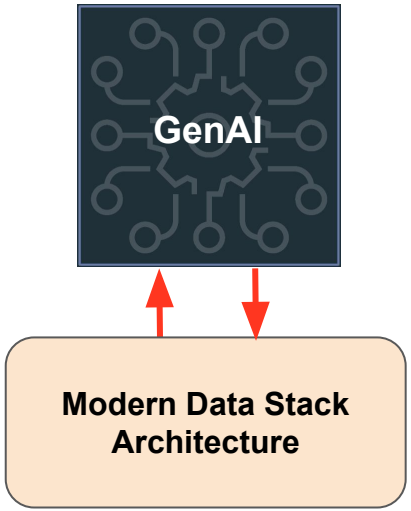
SQL-to-SQL Translation

*SQL Scripts from **Vendor B:***

- *Script 1*
- *Script 2*
- *Script 3*
- ..
- ..
- ..
- ..
- ..
- ..
- ..
- ..
- ..
- ..
- *Script n*

The MDA supports GenAI Use Cases

- Vector DBs for Embeddings (for RAGs)
- Fine-tuning of LLMs
- Company-wide Data Governance for Humans and Agents
- Coding for Data Engineering and Data Science
- ….



THANK YOU!