# The metadata challenge

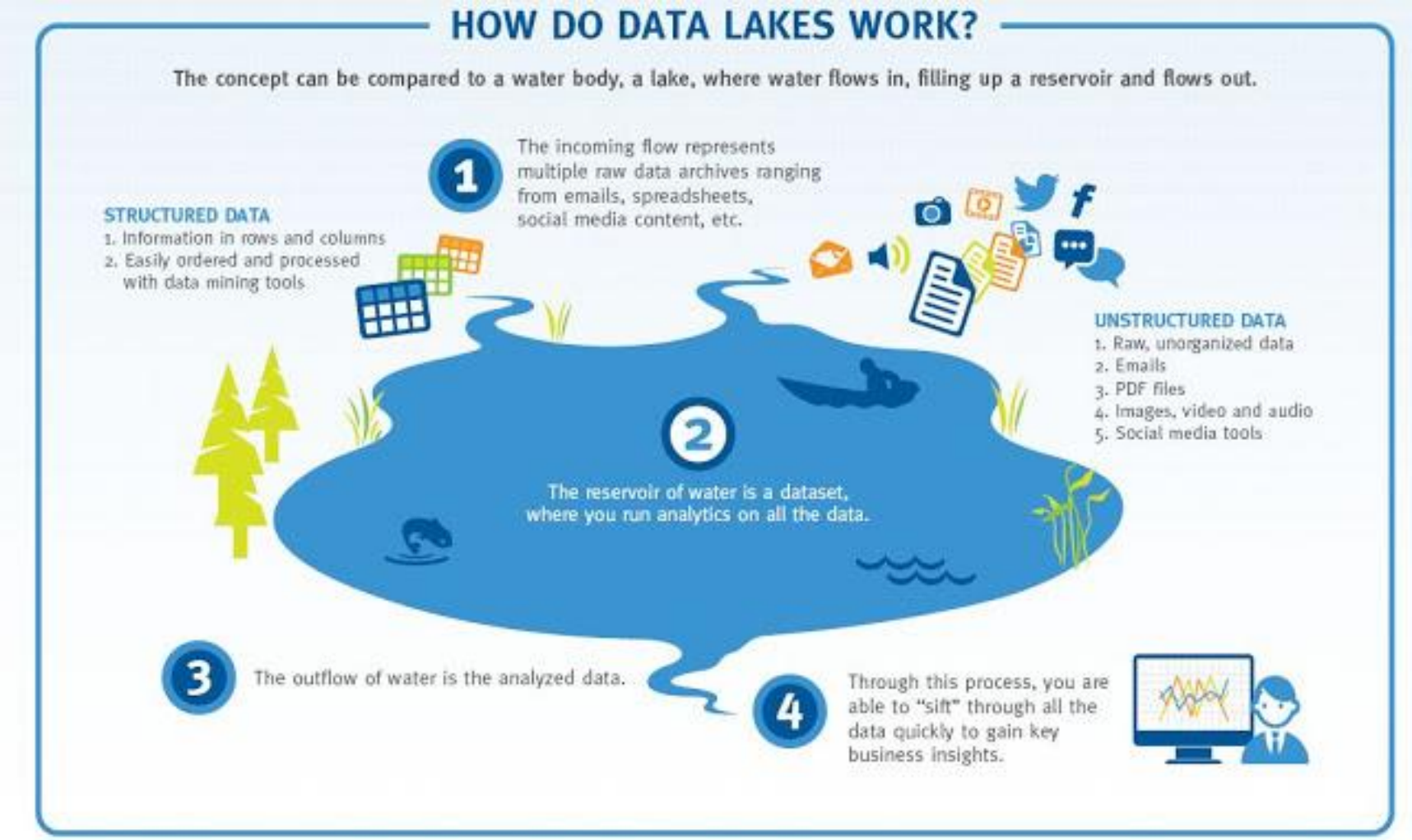
# What we are going to do

Definitions: from the data lake to the data platform

Define challenges

Discuss current solutions and future directions

# Data lake


HOW DO DATA LAKES WORK?
The concept can be compared to a water body, a lake, where water flows in, filling up a reservoir and flows out.
1
The incoming flow represents multiple raw data archives ranging from emails, spreadsheets, social media content, etc.
STRUCTURED DATA
1. Information in rows and columns
2. Easily ordered and processed with data mining tools
UNSTRUCTURED DATA
1. Raw, unorganized data
2. Emails
3. PDF files
4. Images, video and audio
5. Social media tools
2
The reservoir of water is a dataset, where you run analytics on all the data.
3
The outflow of water is the analyzed data.
4
Through this process, you are able to "sift" through all the data quickly to gain key business insights.

# Data lake

"*If you think of a datamart as a store of bottled water – cleansed and packaged and structured for easy consumption – the data lake is a large body of water in a more natural state.*"

- James Dixon, 2010

"*A large storage system for raw, heterogeneous data, fed by multiple data sources, and that allows users to explore, extract and analyze the data.*"

- Sawadogo, P., Darmont, J. **On data lake architectures and metadata management**. *J Intell Inf Syst* 56, 97–120 (2021)

"*A data lake is a central location that holds a large amount of data in its native, raw format.*"

- Databricks, 2021

# Data lake

The data lake started with the Apache Hadoop movement, using the Hadoop File System (HDFS) for cheap storage

- *Schema-on-read* architecture
- Agility of storing any data at low cost
- Eludes the problems of quality and governance

A two-tier data lake + warehouse architecture is dominant in the industry

- HDFS replaced by cloud data lakes (e.g., S3, ADLS, GCS)
- Data lake data directly accessible to a wide range of analytics engines
- A subset of data is "ETL-ed" to a data warehouse for important decision support and BI apps

Armbrust, M., Ghodsi, A., Xin, R., & Zaharia, M. (2021). **Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics**. *CIDR*.

# Data lake

## Downsides of data lakes

- Security
  - All the data is stored and managed as files
  - No fine-grained access control on the contents of files, but only coarse-grained access governing who can access what files or directories
- Quality
  - Hard to prevent data corruption and manage schema changes
  - Challenging to ensure atomic operations when writing a group of files
  - No roll-back mechanism
- Query performance
  - Formats are not optimized for fast access

## It is often said that the *lake* easily turns into a *swamp*

# Data lakehouse

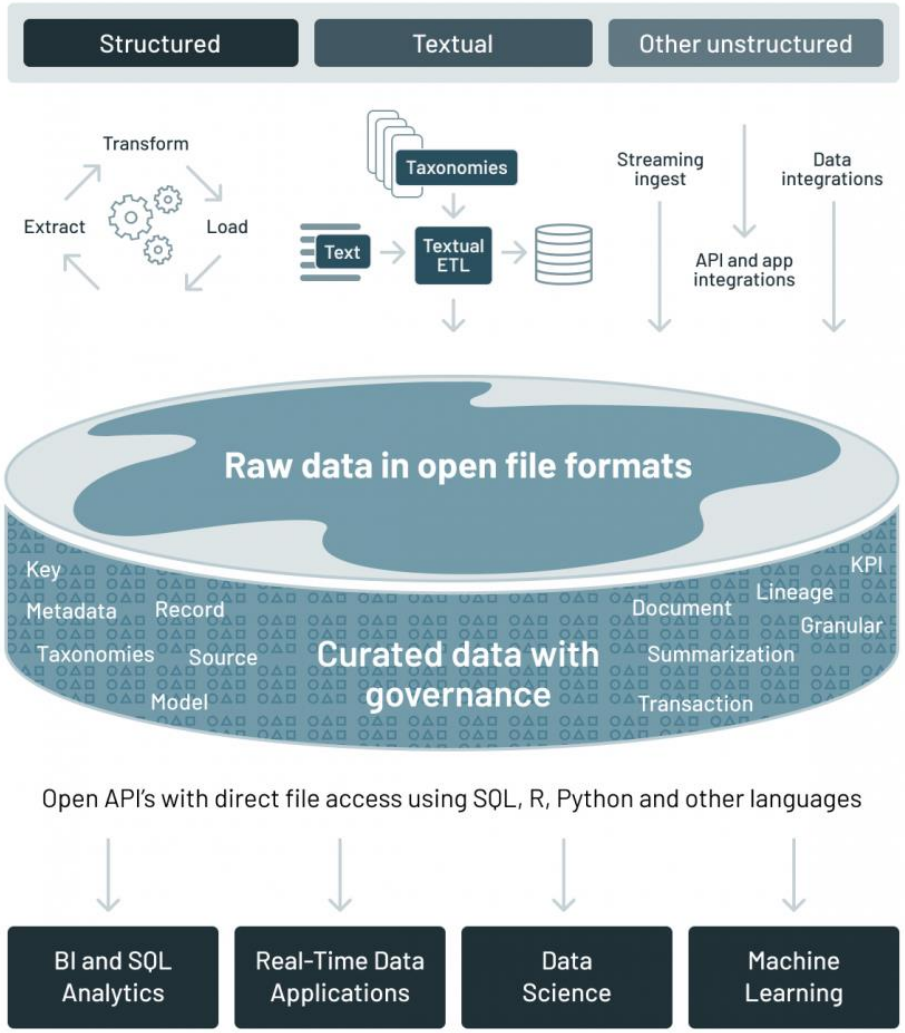
The data lakehouse enables storing all your data once in a data lake and efficiently doing AI and BI on that data directly at a massive scale

- ACID transaction support
- Schema enforcement
- Data governance
  - All processes ensuring that data meet high quality standards throughout the whole lifecycles
  - Including availability, usability, consistency, integrity, security
- Support for diverse workloads (e.g., data science, ML, SQL, analytics)

https://databricks.com/blog/2020/01/30/what-is-a-data-lakehouse.html

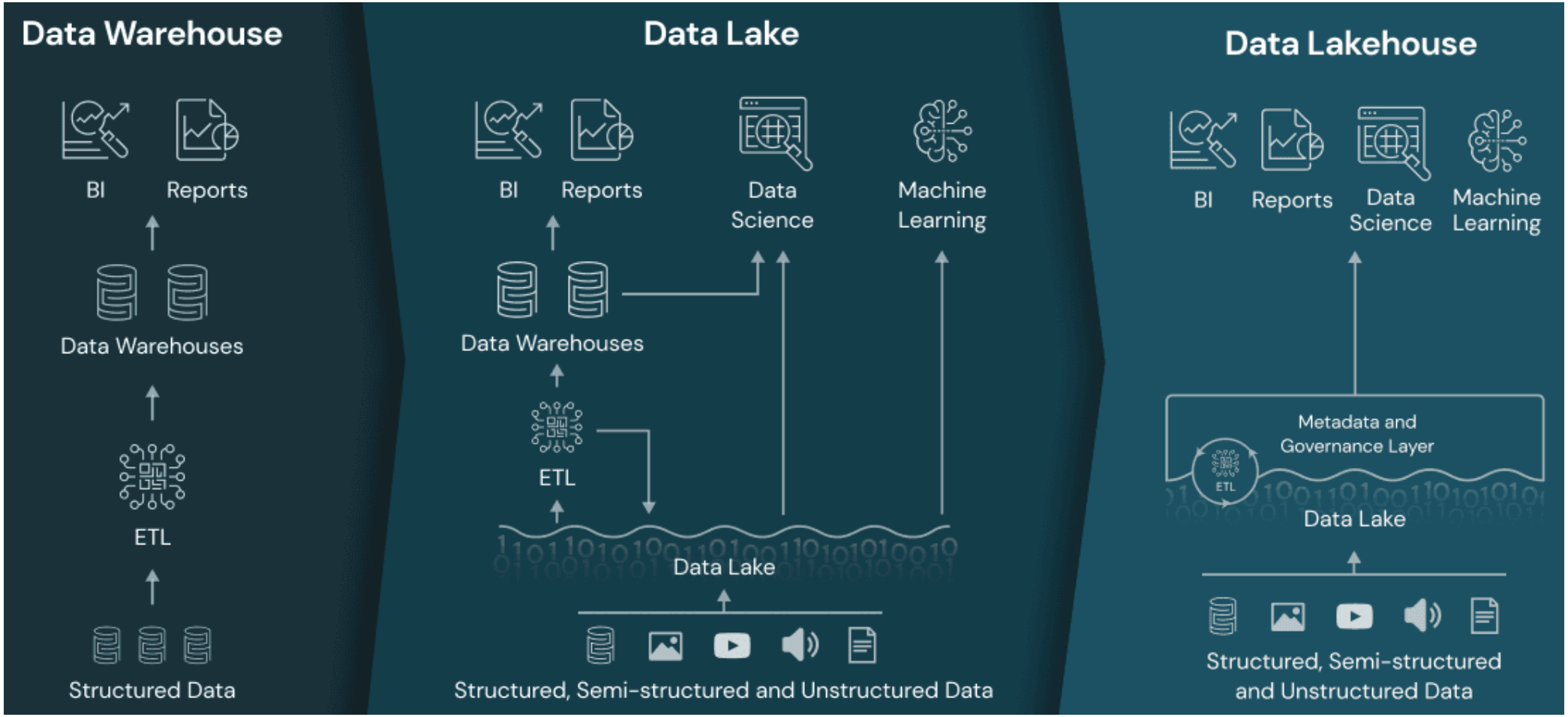
# Data lakehouse

| | Data warehouse | Data lake | Data lakehouse |
|---|---|---|---|
| **Data format** | Closed, proprietary format | **Open format** (e.g., Parquet) | Open format |
| **Types of data** | Structured data, with limited support for semi-structured data | **All types**: Structured data, semi-structured data, textual data, unstructured (raw) data | All types: Structured data, semi-structured data, textual data, unstructured (raw) data |
| **Data access** | SQL-only, no direct access to file | **Open APIs** for direct access to files with SQL, R, Python and other languages | Open APIs for direct access to files with SQL, R, Python and other languages |
| **Reliability** | **High quality**, reliable data with ACID transactions | Low quality, data swamp | High quality, reliable data with ACID transactions |
| **Governance and security** | **Fine-grained** security and governance for row/columnar level for tables | Poor governance as security needs to be applied to files | Fine-grained security and governance for row/columnar level for tables |
| **Performance** | **High** | Low | High |
| **Scalability** | Scaling becomes exponentially more expensive | **Scales** to hold any amount of data at low cost, regardless of type | Scales to hold any amount of data at low cost, regardless of type |
| **Use case support** | Limited to BI, SQL applications and decision support | Limited to machine learning | One data architecture for BI, SQL and machine learning |

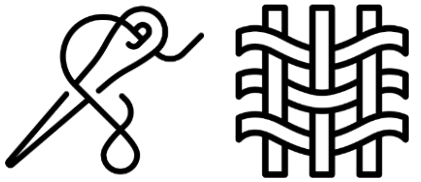



https://databricks.com/blog/2021/05/19/evolution-to-the-data-lakehouse.html

# Data lakehouse



# Data fabric



Data fabric enables frictionless access and sharing of data in a distributed data environment

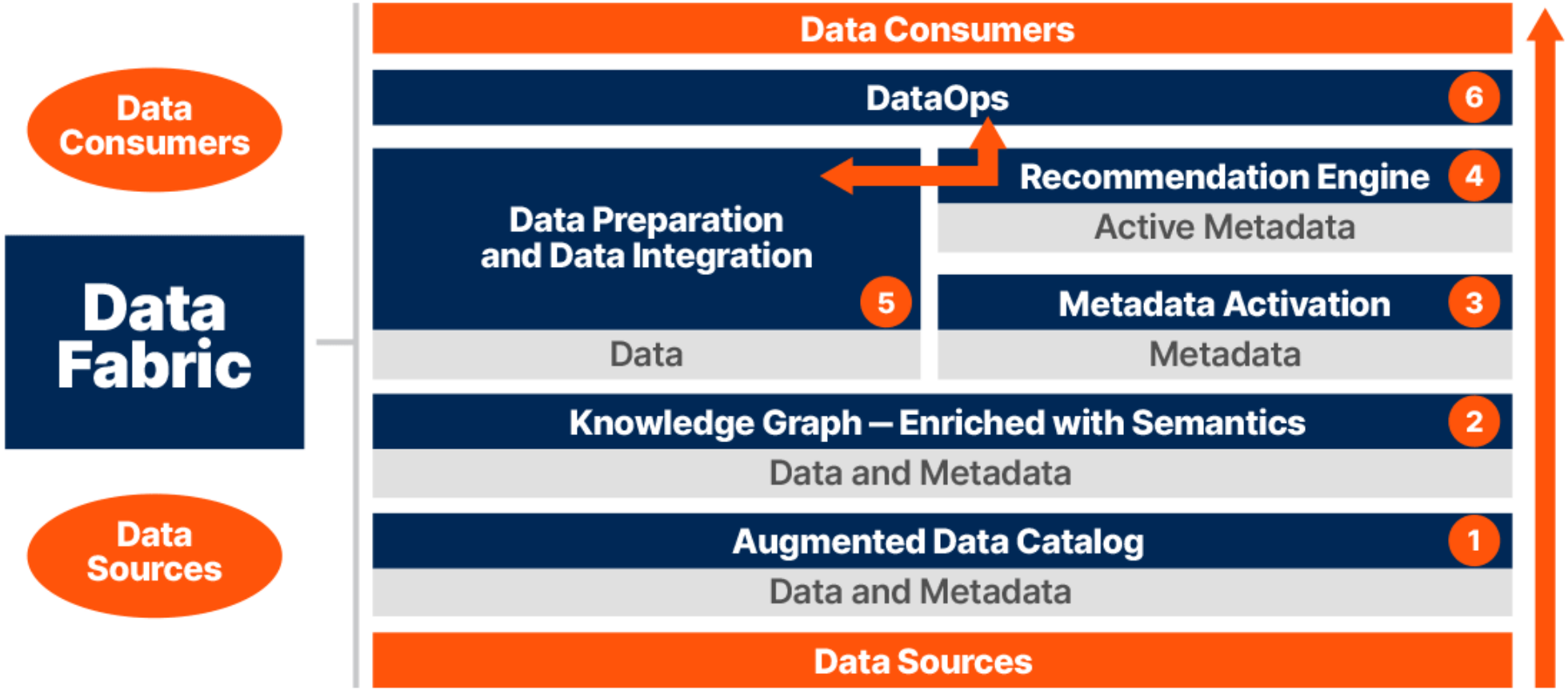
- It enables **a single and consistent data management framework**, which allows seamless data access and processing by design across otherwise siloed storage
- Leverages **both human and machine capabilities** to access data in place or support its consolidation where appropriate
- **Continuously** identifies and connects data from disparate applications to discover unique, business-relevant relationships between the available data points

It is a unified architecture with an integrated set of technologies and services

- Designed to deliver **integrated and enriched data** – at the right time, in the right method, and to the right data consumer – in support of both operational and analytical workloads
- Combines key data management technologies – such as data catalog, data governance, data integration, data pipelining, and data orchestration

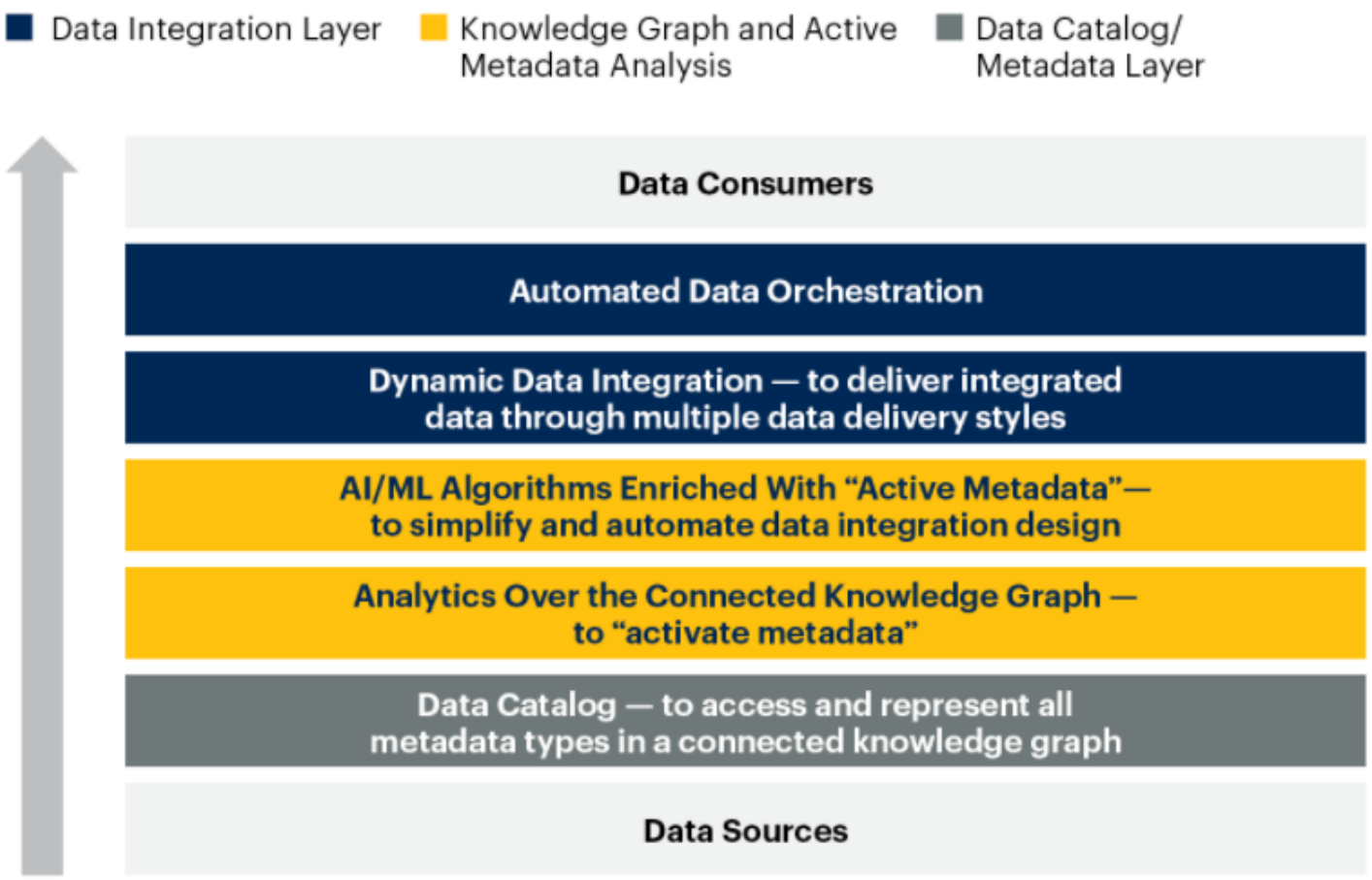
Gartner, 2019 https://www.gartner.com/en/newsroom/press-releases/2019-02-18-gartner-identifies-top-10-data-and-analytics-technolo
Gartner, 2021 https://www.gartner.com/smarterwithgartner/data-fabric-architecture-is-key-to-modernizing-data-management-and-integration
K2View Whitepaper: What is a Data Fabric? The Complete Guide, 2021

# Data fabric



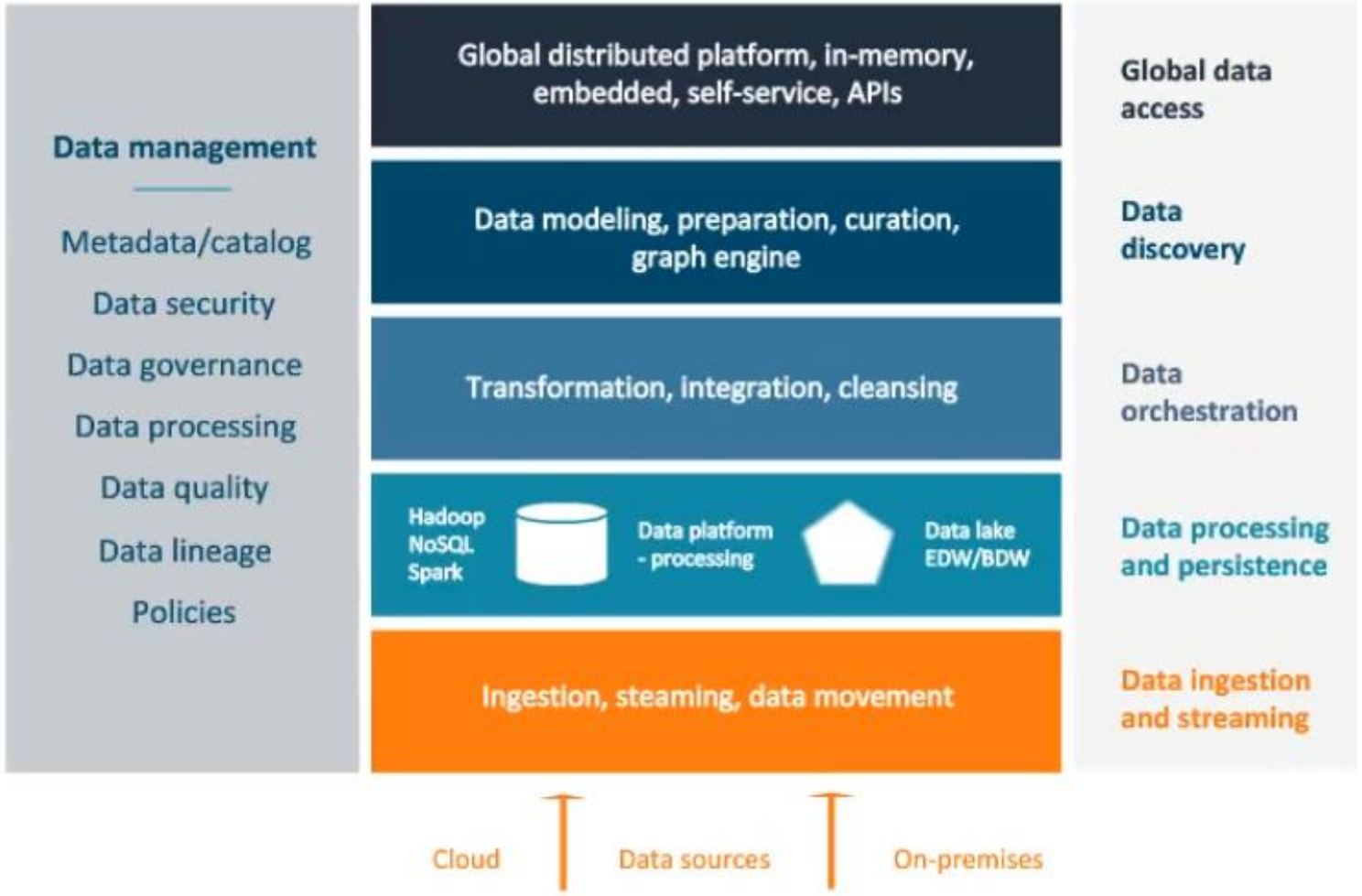

https://www.irion-edm.com/data-management-insights/gartner-data-summit-irion-representative-vendor-for-data-fabric-technology/

# Data fabric




Gartner, 2021 https://www.gartner.com/smarterwithgartner/data-fabric-architecture-is-key-to-modernizing-data-management-and-integration

# Data fabric


Data management
Metadata/catalog
Data security
Data governance
Data processing
Data quality
Data lineage
Policies
Global distributed platform, in-memory, embedded, self-service, APIs
Data modeling, preparation, curation, graph engine
Transformation, integration, cleansing
Hadoop NoSQL Spark
Data platform - processing
Data lake EDW/BDW
Ingestion, steaming, data movement
Global data access
Data discovery
Data orchestration
Data processing and persistence
Data ingestion and streaming
Cloud
Data sources
On-premises

# Data fabric

## It is a design concept

- It optimizes data management by automating repetitive tasks
- According to Gartner estimates, 25% of data management vendors will provide a complete framework for data fabric by 2024 – up from 5% today

| | |
|---|---|
| Cambridge Semantics | Anzo, AnzoGraph |
| Cloudera | Cloudera Data Platform |
| DataRobot | Paxata |
| Denodo Technologies | Denodo Platform |
| Hitachi Vantara | Lumada Data Services |
| IBM | IBM Cloud Pak for Data |
| Informatica | Informatica Intelligent Data Management |
| Infoworks | DataFoundry |
| Oracle | Oracle GoldenGate, Oracle Autonomous Data Platform, Oracle Cloud Infrastructure, Oracle Analytics Cloud |
| Qlik | Qlik Data Catalyst, Qlik Replicate, Qlik Compose for Data Warehouse, Qlik Compose for Data Lakes |
| SAP | SAP HANA, SAP Data Intelligence, SAP Information Management, SAP PowerDesigner, SAP Cloud Platform Integration |
| Solix Technologies | Solix Common Data Platform |
| Syncsort | Syncsort Connect, Syncsort Trillium, Syncsort Spectrum, Syncsort Ironstream |
| Talend | Talend Data Fabric |
| TIBCO Software | TIBCO Unify |


Gartner, 2021 https://www.gartner.com/smarterwithgartner/data-fabric-architecture-is-key-to-modernizing-data-management-and-integration

K2View, 2021 https://www.k2view.com/top-data-fabric-vendors

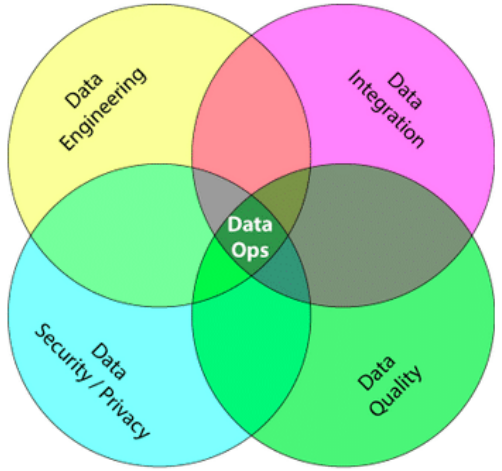
# DataOps

## From DevOps to DataOps

- *"A collaborative data management practice focused on improving the communication, integration and automation of data flows between data managers and data consumers across an organization"*
- Data analytics improved in terms of velocity, quality, predictability and scale of software engineering and deployment

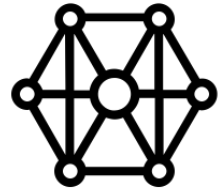

## Some key rules

- Establish progress and performance measurements at every stage
- Automate as many stages of the data flow as possible
- Establish governance discipline (*governance-as-code*)
- Design process for growth and extensibility




Gartner, 2020 https://www.gartner.com/smarterwithgartner/how-dataops-amplifies-data-and-analytics-business-value
Andy Palmer, 2015 https://www.tamr.com/blog/from-devops-to-dataops-by-andy-palmer/
William Vorhies, 2017 https://www.datasciencecentral.com/profiles/blogs/dataops-it-s-a-secret

# Data mesh



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

- Domain-oriented decentralized data ownership
  - Decentralization and distribution of responsibility to people who are closest to the data, in order to support continuous change and scalability
  - Each domain exposes its own op/analytical APIs
- Data as a product (*quantum*)
  - Products must be discoverable, addressable, trustworthy, self-describing, secure
- Self-serve data infrastructure as a platform
  - High-level abstraction of infrastructure to provision and manage the lifecycle of data products
- Federated computational governance
  - A governance model that embraces decentralization and domain self-sovereignty, interoperability through global standardization, a dynamic topology, automated execution of decisions by the platform

Zhamak Dehghani, 2019 https://martinfowler.com/articles/data-monolith-to-mesh.html
Zhamak Dehghani, 2020 https://martinfowler.com/articles/data-mesh-principles.html

# Data mesh vs Data fabric

A data fabric and a data mesh both provide an architectural framework to access data across multiple technologies and platforms

- Data fabric
  - Attempts to centralize and coordinate data management
  - Tackles the complexity of data and metadata in a smart way that works well together
- Data mesh
  - Emphasis on decentralization and data domain autonomy
  - Focuses on organizational change; it is more about people and process

They are concepts, not things

- They are *not* mutually exclusive
- They are architectural frameworks, not architectures
  - The frameworks must be adapted and customized to your needs, data, processes, and terminology

Alex Woodie, 2021 https://www.datanami.com/2021/10/25/data-mesh-vs-data-fabric-understanding-the-differences/
Dave Wells, 2021 https://www.eckerson.com/articles/data-architecture-complex-vs-complicated

# Data platform

Data fabric and mesh are different architectural frameworks for data platforms

A data platform is a complete solution for ingesting, processing, analyzing, and presenting the data generated by the systems, processes, and infrastructures of the modern digital organization

- A single platform to be used across an entire organization
- Prevent silos
- Provide actionable insights based on a holistic view of the organization’s data

# Metadata challenges

Knowledge representation

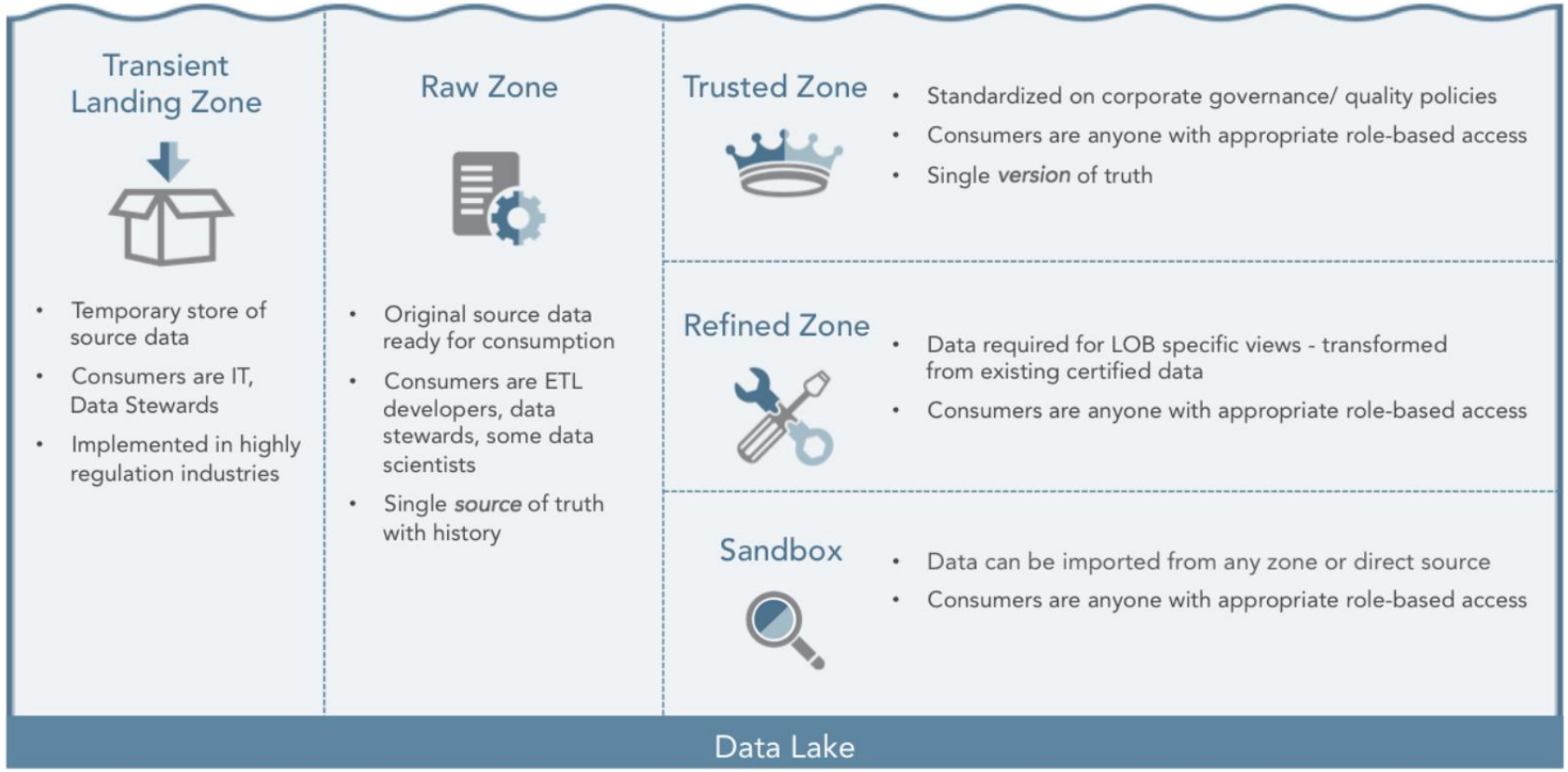
- Which metadata must be captured
- How should metadata be organized

Knowledge exploitation

- Which features do metadata enable

# Knowledge representation

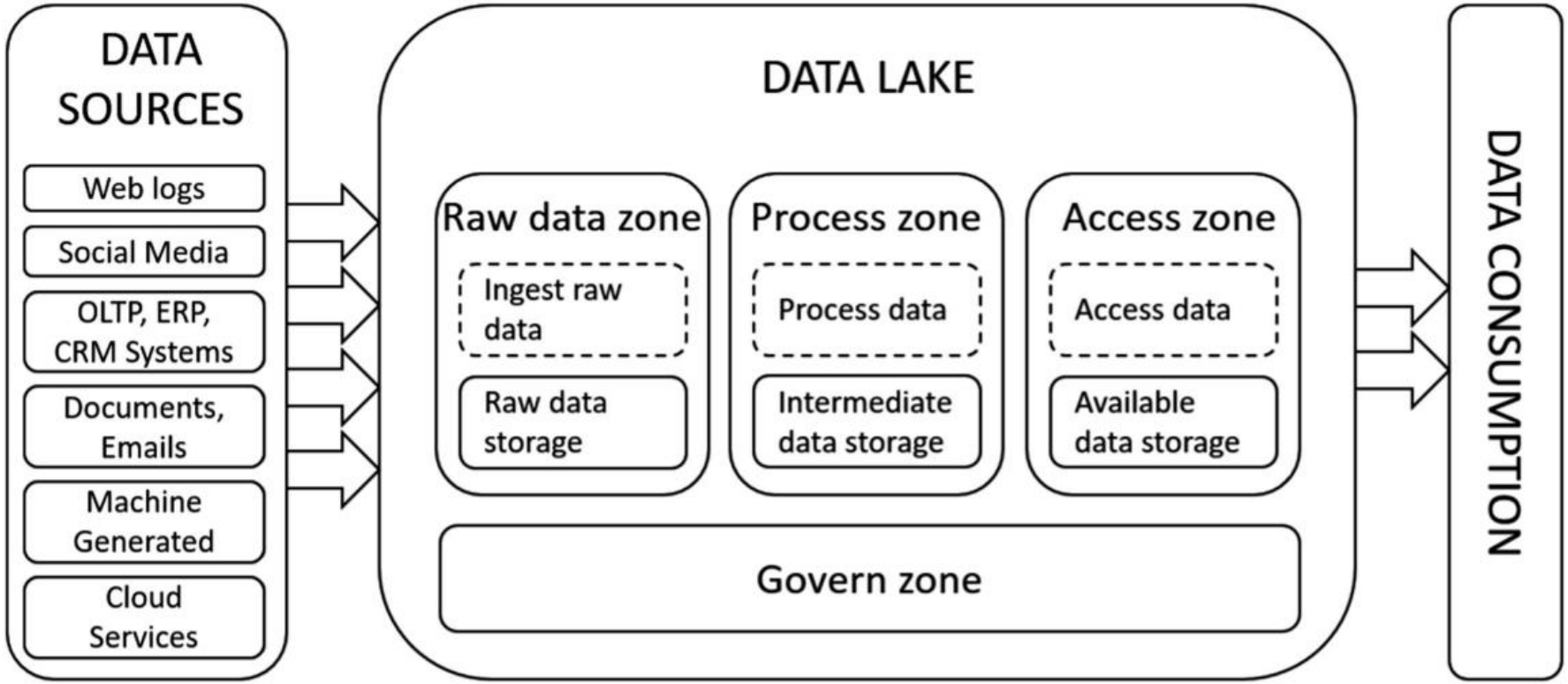
## A first layer of metadata: structuring the data lake




A. LaPlante, B. Sharma, **Architecting Data Lakes**, *O'Reilly Media*, Sebastopol, 2018.

# Knowledge representation

A first layer of metadata: structuring the data lake




F. Ravat, Y. Zhao, **Data lakes: Trends and perspectives**, in: *Proc. DEXA*, Linz, Austria, 2019, pp. 304–313.

# Knowledge representation

## A classification of functionalities enabled by metadata

- Semantic enrichment
  - Generating a description of the context of data, e.g., with tags, to make them more interpretable and understandable
- Data indexing
  - Data structures to retrieve datasets based on specific characteristics (keywords or patterns)
- Link generation and conservation
  - Detecting similarity relationships or integrating preexisting links between datasets
- Data polymorphism
  - Storing multiple representations of the same data to avoid repeating pre-processings and speed up analyses
- Data versioning
  - Support data changes while conserving previous states
- Usage tracking
  - Records the interactions between users and the data

Sawadogo, P. N., Scholly, E., Favre, C., Ferey, E., Loudcher, S., & Darmont, J. (2019, September). **Metadata systems for data lakes: models and features.** In *European conference on advances in databases and information systems* (pp. 440-451). Springer, Cham.

# Knowledge representation

## A classification of metadata

- **Technical** metadata
  - Capture the form and structure of each dataset
  - E.g.: type of data (text, JSON, Avro); structure of the data (the fields and their types)
- **Operational** metadata
  - Capture lineage, quality, profile, and provenance of the data
  - E.g.: source and target locations of data, size, number of records, and lineage
- **Business** metadata
  - Captures what it all means to the user
  - E.g.: business names, descriptions, tags, quality, and masking rules for privacy

A. LaPlante, B. Sharma, **Architecting Data Lakes**, *O'Reilly Media*, Sebastopol, 2018.

# Knowledge representation

## Another classification of metadata

- **Intra-object** metadata
  - *Properties* provide a general description of an object in the form of key-value pairs
  - *Summaries and previews* provide an overview of the content or structure of an object
  - *Semantic metadata* are annotations that help understand the meaning of data
- **Inter-object** metadata
  - *Objects groupings* organize objects into collections, each object being able to belong simultaneously to several collections
  - *Similarity links* reflect the strength of the similarity between two objects
  - *Parenthood relationships* reflect the fact that an object can be the result of joining several others
- **Global** metadata
  - *Semantic resources*, i.e., knowledge bases (ontologies, taxonomies, thesauri, dictionaries) used to generate other metadata and improve analyses
  - *Indexes*, i.e., data structures that help find an object quickly
  - *Logs*, used to track user interactions with the data lake

Sawadogo, P. N., Scholly, E., Favre, C., Ferey, E., Loudcher, S., & Darmont, J. (2019, September). **Metadata systems for data lakes: models and features**. In *European conference on advances in databases and information systems* (pp. 440-451). Springer, Cham.

# Knowledge representation

Table 1: Features provided by data lake metadata systems

| System | Type | SE | DI | LG | DP | DV | UT |
|---|---|---|---|---|---|---|---|
| SPAR (Fauduet and Peyrard, 2010) [10] | ♦♯ | ✓ | ✓ | ✓ | | | ✓ |
| Alrehamy and Walker (2015) [1] | ♦ | ✓ | | ✓ | | | |
| Terrizzano et al. (2015) [27] | ♦ | ✓ | ✓ | | | ✓ | ✓ |
| Constance (Hai et al., 2016) [11] | ♦ | ✓ | ✓ | | | | |
| GEMMS (Quix et al., 2016) [22] | ◊ | ✓ | | | | | |
| CLAMS (Farid et al., 2016) [8] | ♦ | ✓ | | | | | |
| Suriarachchi and Plale (2016) [26] | ◊ | | | | ✓ | | ✓ |
| Singh et al. (2016) [24] | ♦ | ✓ | ✓ | ✓ | ✓ | | |
| Farrugia et al. (2016) [9] | ♦ | | | ✓ | | | |
| GOODS (Halevy et al., 2016) [12] | ♦ | ✓ | ✓ | ✓ | | ✓ | ✓ |
| CoreDB (Beheshti et al., 2017) [3] | ♦ | | ✓ | | | | ✓ |
| Ground (Hellerstein et al., 2017) [13] | ◊♯ | ✓ | ✓ | | | ✓ | ✓ |
| KAYAK (Maccioni and Torlone, 2018) [17] | ♦ | ✓ | ✓ | ✓ | | | |
| CoreKG (Beheshti et al., 2018) [4] | ♦ | ✓ | ✓ | ✓ | ✓ | | ✓ |
| Diamantini et al. (2018) [5] | ◊ | ✓ | | ✓ | ✓ | | |

♦ : Data lake implementation ◊ : Metadata model
♯ : Model or implementation assimilable to a data lake

Sawadogo, P. N., Scholly, E., Favre, C., Ferey, E., Loudcher, S., & Darmont, J. (2019, September). **Metadata systems for data lakes: models and features**. In *European conference on advances in databases and information systems* (pp. 440-451). Springer, Cham.

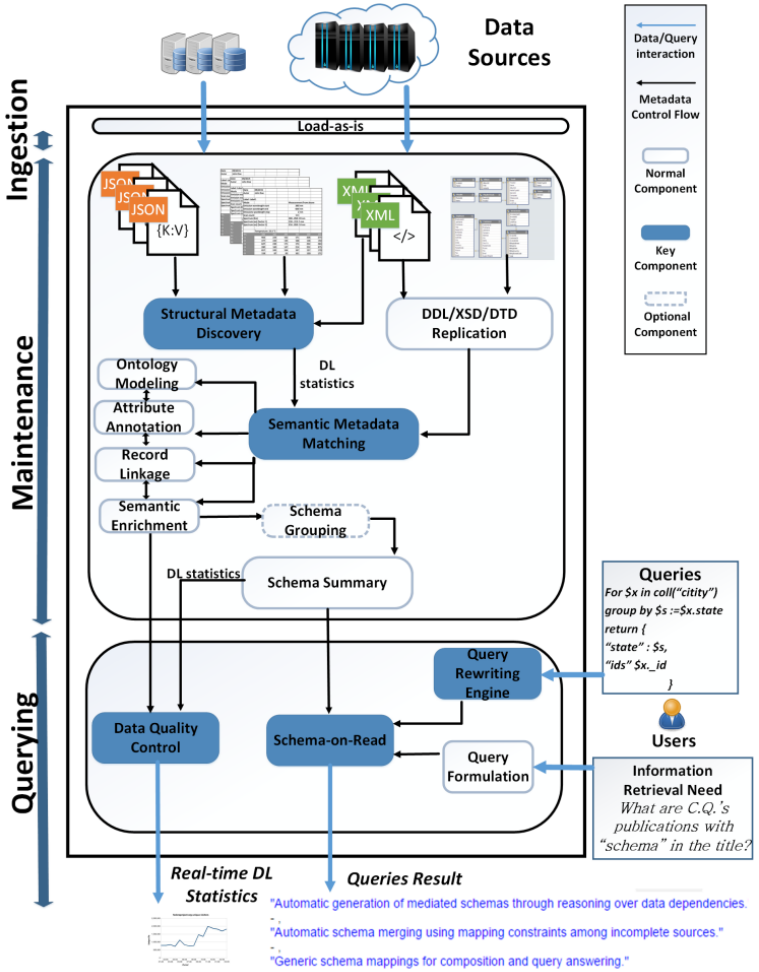
# Knowledge representation

Table 1: Features provided by data lake metadata systems

| System | Type | SE | DI | LG | DP | DV | UT |
|---|---|---|---|---|---|---|---|
| SPAR (Fauduet and Peyrard, 2010) [10] | ♦♯ | ✓ | ✓ | ✓ | | | ✓ |
| Alrehamy and Walker (2015) [1] | ♦ | ✓ | | ✓ | | | |
| Terrizzano et al. (2015) [27] | ♦ | ✓ | ✓ | | | ✓ | ✓ |
| Constance (Hai et al., 2016) [11] | ♦ | ✓ | ✓ | | | | |
| GEMMS (Quix et al., 2016) [22] | ◊ | ✓ | | | | | |
| CLAMS (Farid et al., 2016) [8] | ♦ | ✓ | | | | | |
| Suriarachchi and Plale (2016) [26] | ◊ | | | | ✓ | | ✓ |
| Singh et al. (2016) [24] | ♦ | ✓ | ✓ | ✓ | ✓ | | |
| Farrugia et al. (2016) [9] | ♦ | | | ✓ | | | |
| GOODS (Halevy et al., 2016) [12] | ♦ | ✓ | ✓ | ✓ | | ✓ | ✓ |
| CoreDB (Beheshti et al., 2017) [3] | ♦ | | ✓ | | | | ✓ |
| KA | | | | | | | ✓ |
| Diamantini et al. (2018) [5] | ◊ | ✓ | | ✓ | ✓ | | |

Few details given on metamodel and functionalities.
No metadata collected on operations.

♦ : Data lake implementation ◊ : Metadata model
♯ : Model or implementation assimilable to a data lake



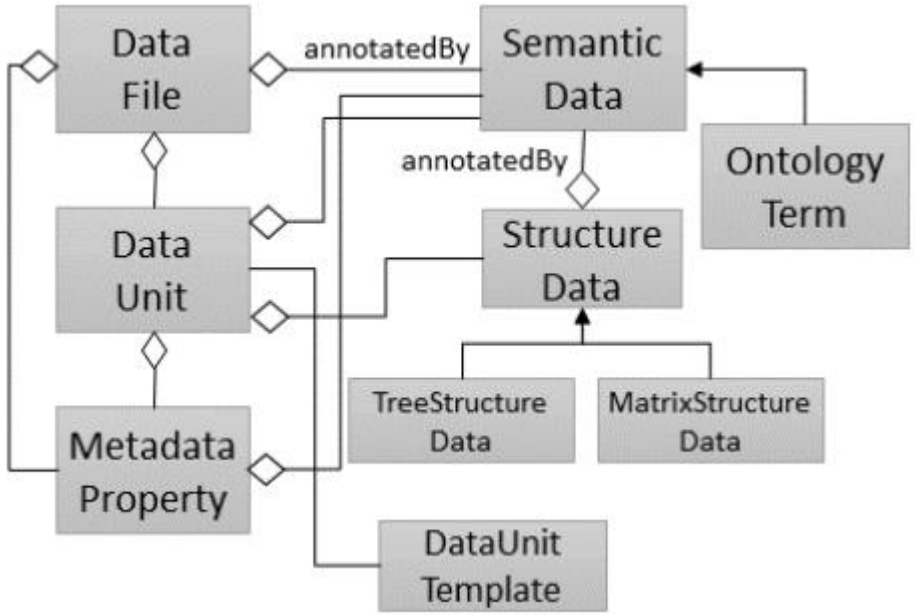

Hai, R., Geisler, S., & Quix, C. (2016, June). **Constance: An intelligent data lake system**. In *Proceedings of the 2016 international conference on management of data* (pp. 2097-2100).

# Knowledge representation

Table 1: Features provided by data lake metadata systems

| System | Type | SE | DI | LG | DP | DV | UT |
|---|---|---|---|---|---|---|---|
| SPAR (Fauduet and Peyrard, 2010) [10] | ♦♯ | ✓ | ✓ | ✓ | | | ✓ |
| Alrehamy and Walker (2015) [1] | ♦ | ✓ | | ✓ | | | |
| Terrizzano et al. (2015) [27] | ♦ | ✓ | ✓ | | | ✓ | ✓ |
| Constance (Hai et al., 2016) [11] | ♦ | ✓ | ✓ | | | | |
| GEMMS (Quix et al., 2016) [22] | ◊ | ✓ | | | | | |
| CLAMS (Farid et al., 2016) [8] | ♦ | ✓ | | | | | |
| Suriarachchi and Plale (2016) [26] | ◊ | | | | ✓ | | ✓ |
| Singh et al. (2016) [24] | ♦ | ✓ | ✓ | ✓ | ✓ | | |
| Farrugia et al. (2016) [9] | ♦ | | | ✓ | | | |
| GOODS (Halevy et al., 2016) [12] | ♦ | ✓ | ✓ | ✓ | | ✓ | ✓ |
| CoreDB (Beheshti et al., 2017) [3] | ♦ | | ✓ | | | | ✓ |
| KA | | | | | | ✓ | ✓ |
| | | | | | | | ✓ |
| Diamantini et al. (2018) [5] | ◊ | ✓ | | ✓ | ✓ | | |

No discussion about the functionalities provided.
No metadata collected on operations and agents.

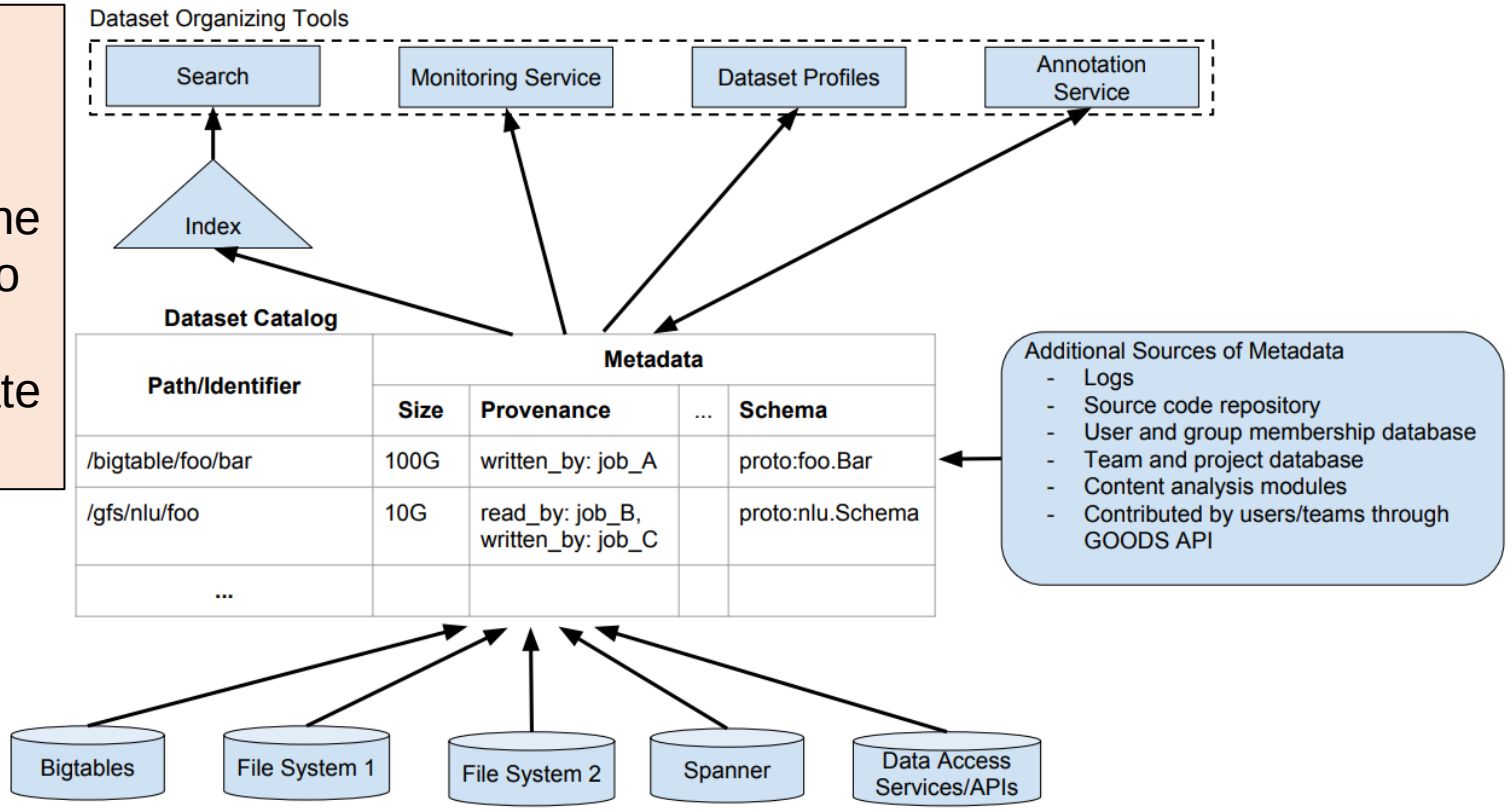



♦ : Data lake implementation ◊ : Metadata model
♯ : Model or implementation assimilable to a data lake

Quix, C., Hai, R., & Vatov, I. (2016). **GEMMS: A Generic and Extensible Metadata Management System for Data Lakes**. In *CAiSE forum* (Vol. 129).

# Knowledge representation

Crawls Google's storage systems to extract basic metadata on datasets and their relationship with other datasets.
Performs metadata inference, e.g., to determine the schema of a non-self-describing dataset, to trace the provenance of data through a sequence of processing services, or to annotate data with their semantics.




Strictly coupled with the Google platform.
Mainly focuses on object description and searches.
No formal description of the metamodel.

Farrugia et al. (2016) [9] ♦
GOODS (Halevy et al., 2016) [12] ♦ ✓
CoreDB (Beheshti et al., 2017) [3] ♦
✓
✓
✓
✓

♯ : Model or implementation assimilable to a data lake

Halevy, A. Y., Korn, F., Noy, N. F., Olston, C., Polyzotis, N., Roy, S., & Whang, S. E. (2016). **Managing Google's data lake: an overview of the Goods system**. *IEEE Data Eng. Bull.*, *39*(3), 5-14.

# Knowledge representation

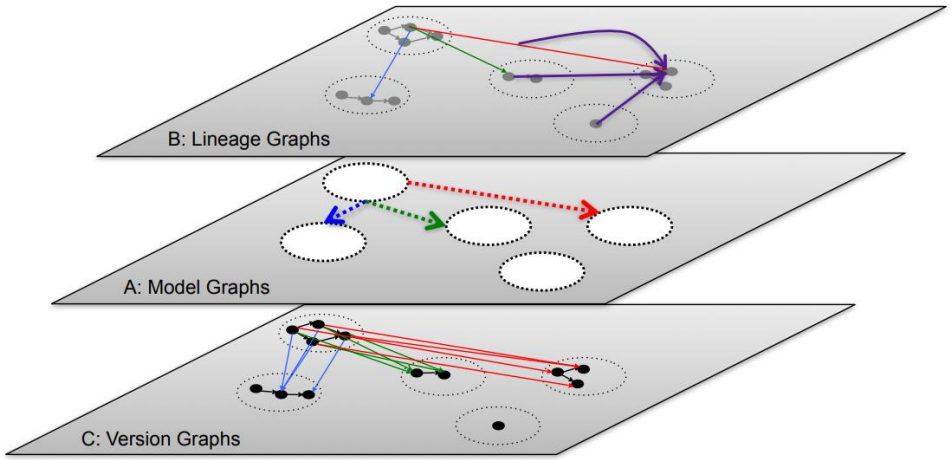
Version graphs represent data versions.
Model graphs represent application metadata, i.e., how data are interpreted for use.
Lineage graphs capture usage information.

Not enough details given to clarify which metadata are actually handled.
Functionalities are described at a high level.

Table 1: Features provided by data lake metadata systems

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | DI | LG | DP | DV | UT |
| | | | | ✓ | ✓ | | | ✓ |
| | | | | | ✓ | | | |
| | | | | ✓ | | | ✓ | ✓ |
| | | | | ✓ | | | | |
| GEMMS (Quix et al., 2016) [22] | ◊ | ✓ | | | | | | |
| | | | | | | ✓ | | ✓ |
| | | | | ✓ | ✓ | ✓ | | |
| | | | | | ✓ | | | |
| GOODS (Halevy et al., 2016) [12] | ♦ | ✓ | ✓ | | ✓ | | ✓ | ✓ |
| CoreDB (Beheshti et al., 2017) [3] | ♦ | | ✓ | | | | | ✓ |
| Ground (Hellerstein et al., 2017) [13] | ◊♯ | ✓ | ✓ | | | | ✓ | ✓ |
| KAYAK (Maccioni and Torlone, 2018) [17] | ♦ | ✓ | ✓ | | ✓ | | | |
| CoreKG (Beheshti et al., 2018) [4] | ♦ | ✓ | ✓ | | ✓ | ✓ | | ✓ |
| Diamantini et al. (2018) [5] | ◊ | ✓ | | | ✓ | ✓ | | |

♦ : Data lake implementation ◊ : Metadata model
♯ : Model or implementation assimilable to a data lake




Hellerstein, J. M., Sreekanti, V., Gonzalez, J. E., Dalton, J., Dey, A., Nag, S., ... & Sun, E. (2017, January). **Ground: A Data Context Service**. In *CIDR*.

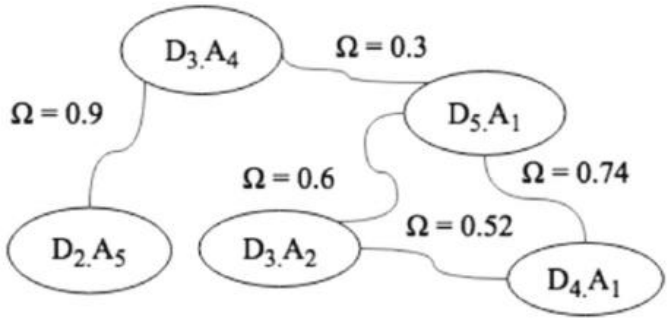
# Knowledge representation

Table 1: Features provided by data lake m

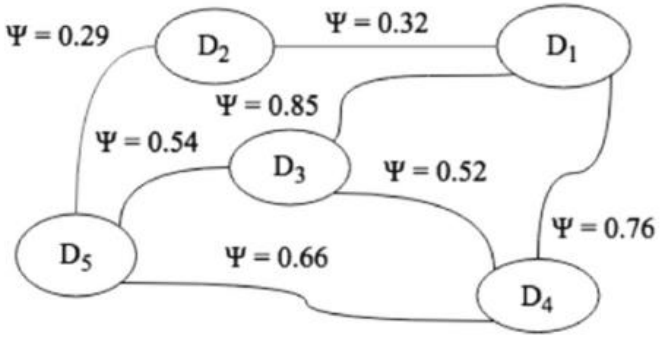
| System | Type | SE | I | | | | | |
|---|---|---|---|---|---|---|---|---|
| SPAR (Fauduet and Peyrard, 2010) [10] | ♦♯ | ✓ | | | | | | |
| Alrehamy and Walker (2015) [1] | ♦ | ✓ | | | | | | |
| Terrizzano et al. (2015) [27] | ♦ | ✓ | | | | | | |
| Constance (Hai et al., 2016) [11] | ♦ | ✓ | | | | | | |
| GEMMS (Quix et al., 2016) [22] | ◊ | ✓ | | | | | | |
| CLAMS (Farid et al., 2016) [8] | ♦ | ✓ | | | | | | |
| | | | ✓ | ✓ | ✓ | | | |
| | | | | | ✓ | | | |
| | | | ✓ | ✓ | | ✓ | ✓ | |
| CoreDB (Beheshti et al., 2017) [3] | ♦ | | ✓ | | | | ✓ | |
| Ground (Hellerstein et al., 2017) [13] | ◊♯ | ✓ | ✓ | | | ✓ | ✓ | |
| KAYAK (Maccioni and Torlone, 2018) [17] | ♦ | ✓ | ✓ | ✓ | | | | |
| CoreKG (Beheshti et al., 2018) [4] | ♦ | ✓ | ✓ | ✓ | ✓ | | ✓ | |
| Diamantini et al. (2018) [5] | ◊ | ✓ | | ✓ | ✓ | | | |

♦ : Data lake implementation ◊ : Metadata model
♯ : Model or implementation assimilable to a data lake

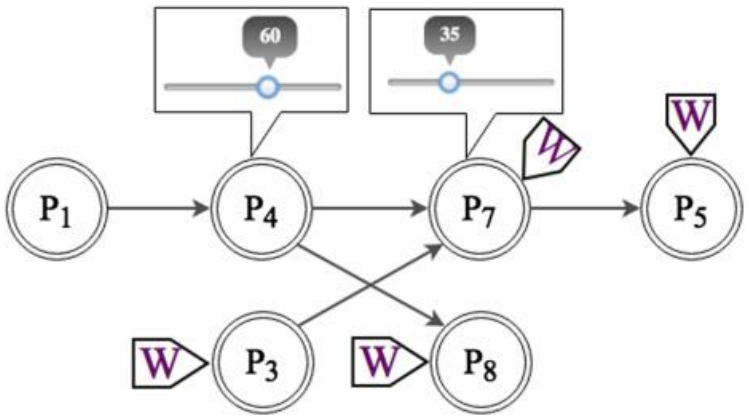
Support users in creating and optimizing the data processing pipelines.
Only goal-related metadata are collected.




(a) Affinity of attributes $A_j$




(b) Joinability of datasets $D_i$



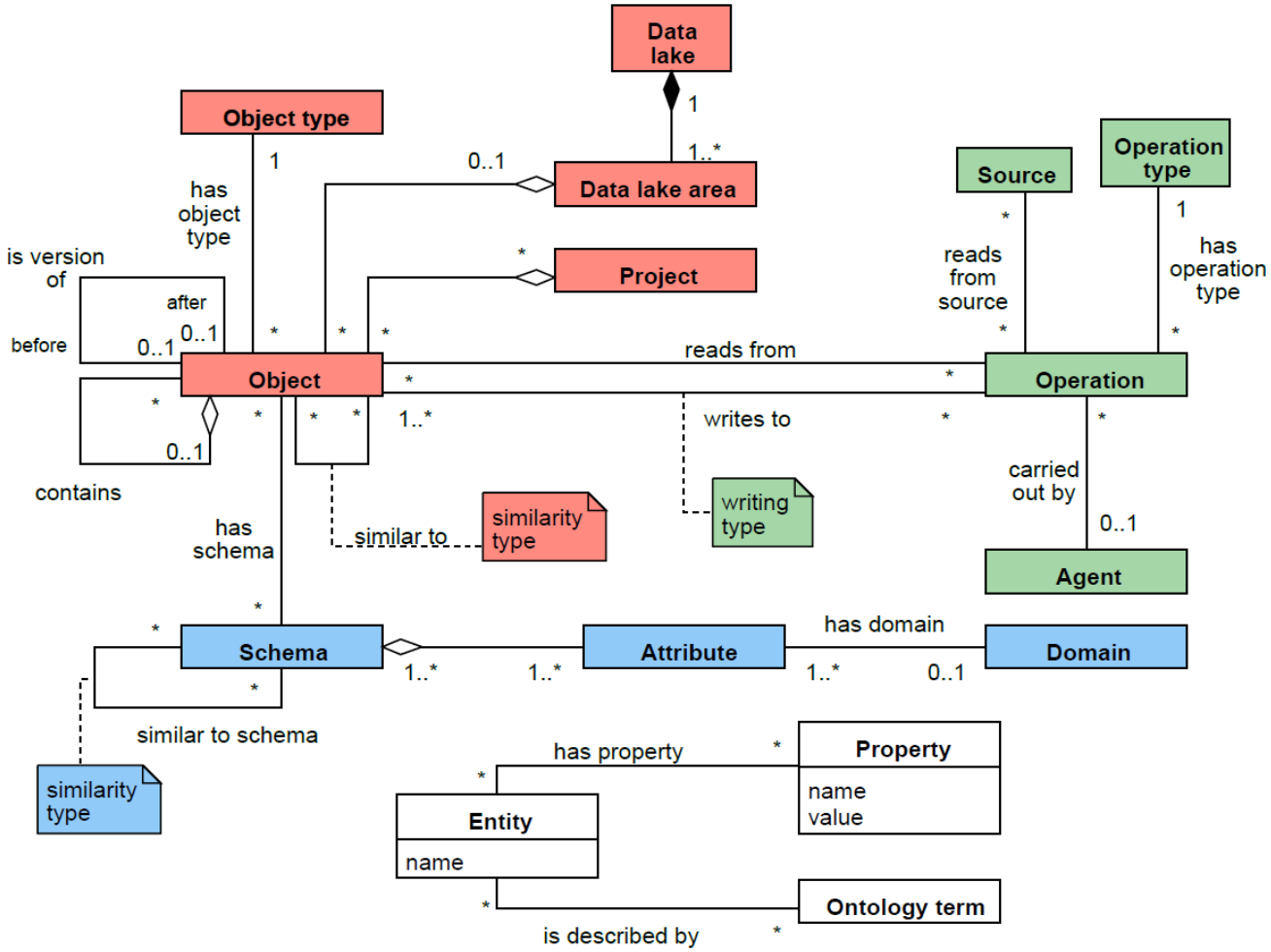

Maccioni, A., & Torlone, R. (2018, June). **KAYAK: a framework for just-in-time data preparation in a data lake**. In *International Conference on Advanced Information Systems Engineering* (pp. 474-489). Springer, Cham.

# Knowledge representation

Technical

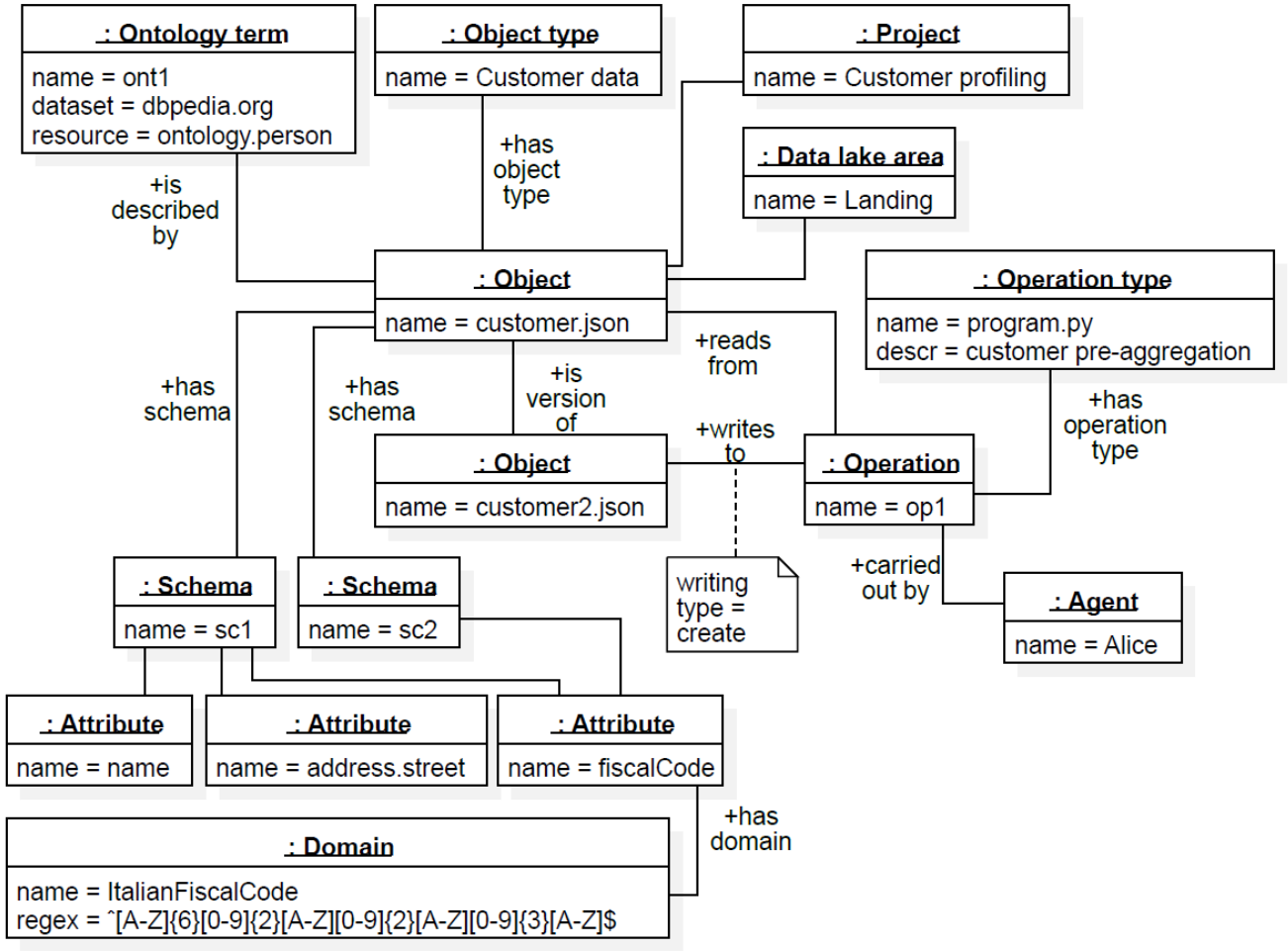
Operational

Business




Francia, M., Gallinucci, E., Golfarelli, M., Leoni, A. G., Rizzi, S., & Santolini, N. (2021). **Making data platforms smarter with MOSES**. *Future Generation Computer Systems*, 125, 299-313.

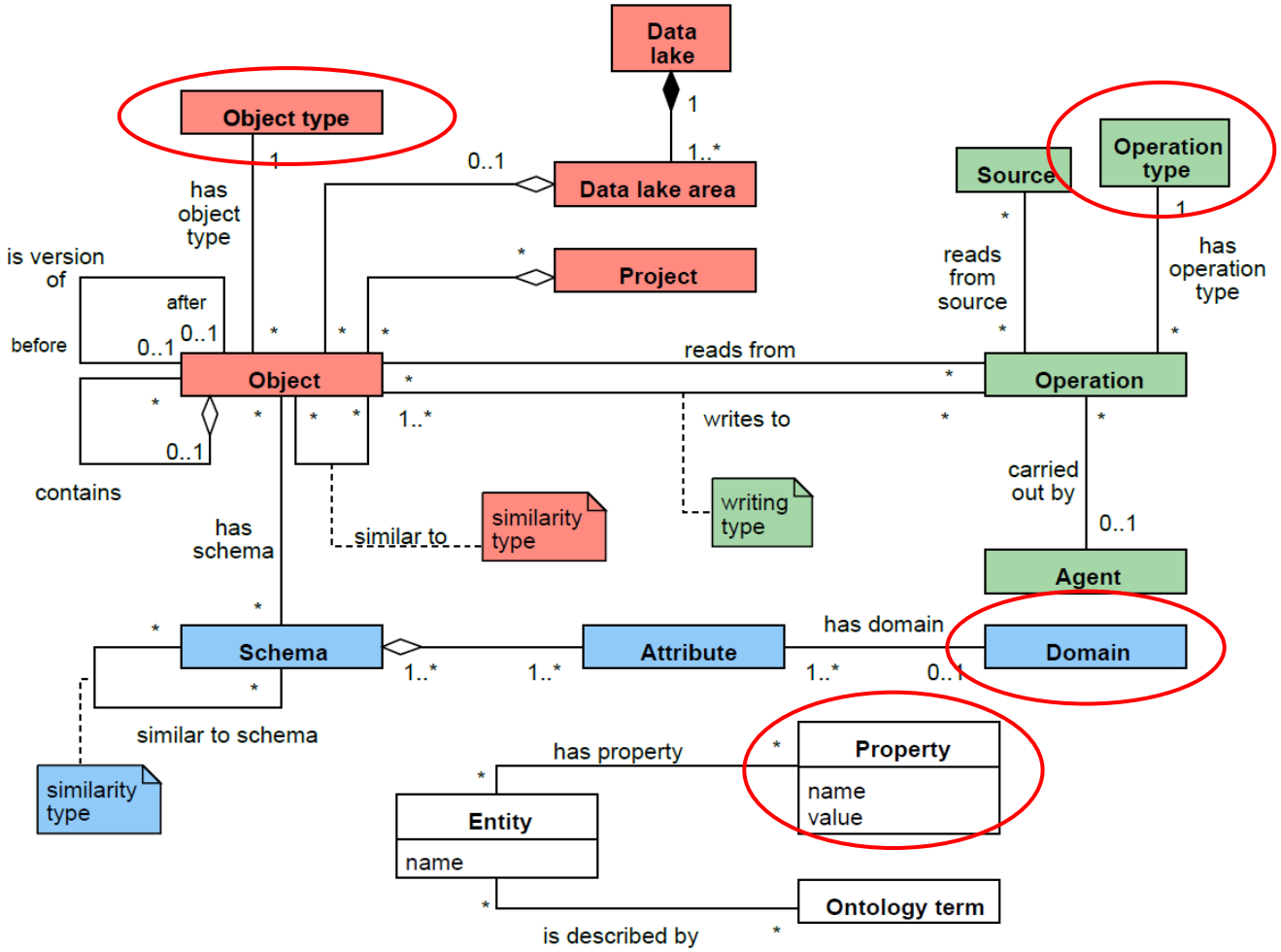
# Knowledge representation




Francia, M., Gallinucci, E., Golfarelli, M., Leoni, A. G., Rizzi, S., & Santolini, N. (2021). **Making data platforms smarter with MOSES**. *Future Generation Computer Systems*, 125, 299-313.

# Knowledge representation

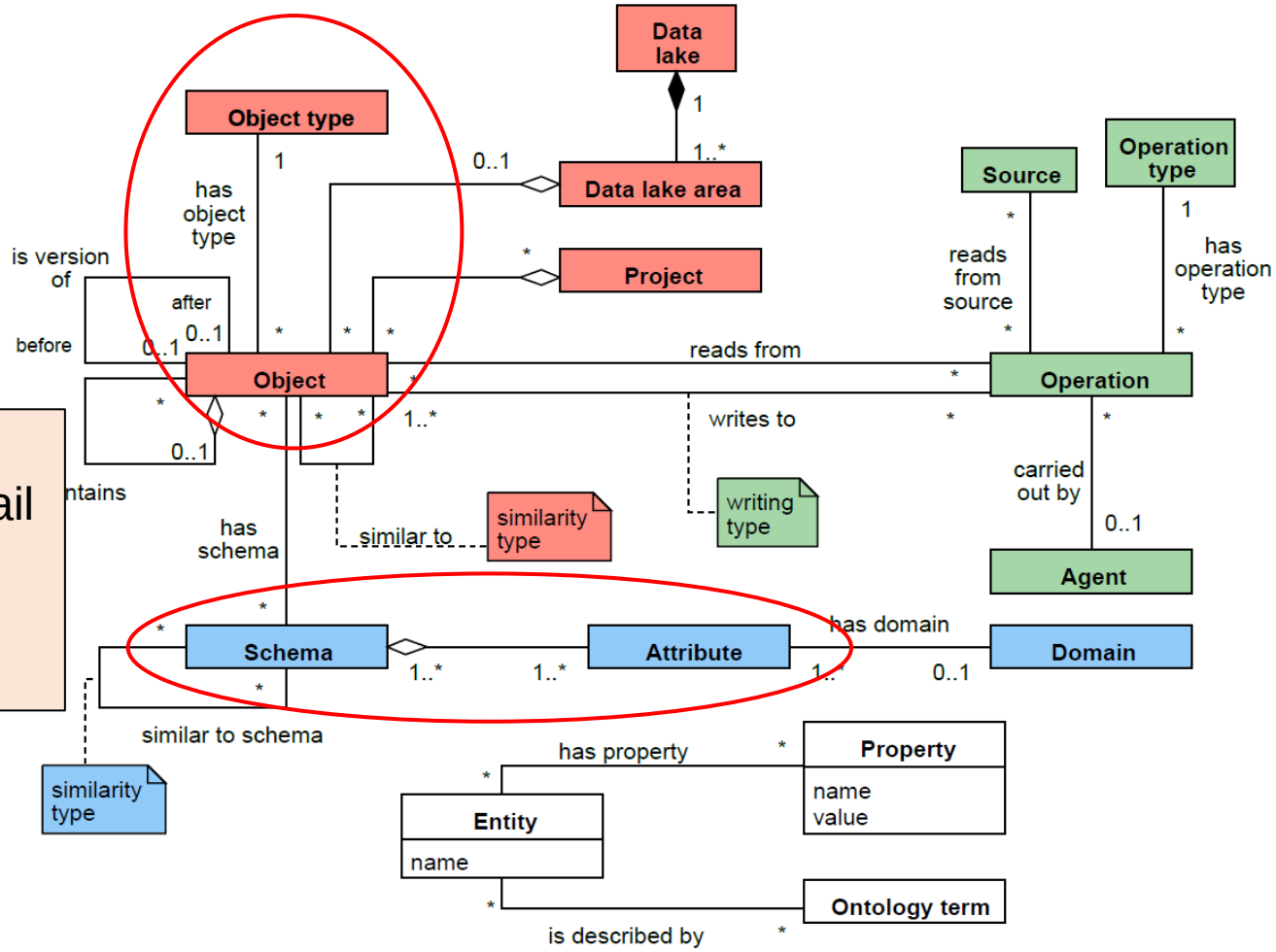
Not pre-defined
Domain-independent, extensible




Francia, M., Gallinucci, E., Golfarelli, M., Leoni, A. G., Rizzi, S., & Santolini, N. (2021). **Making data platforms smarter with MOSES**. *Future Generation Computer Systems*, 125, 299-313.

# Knowledge representation

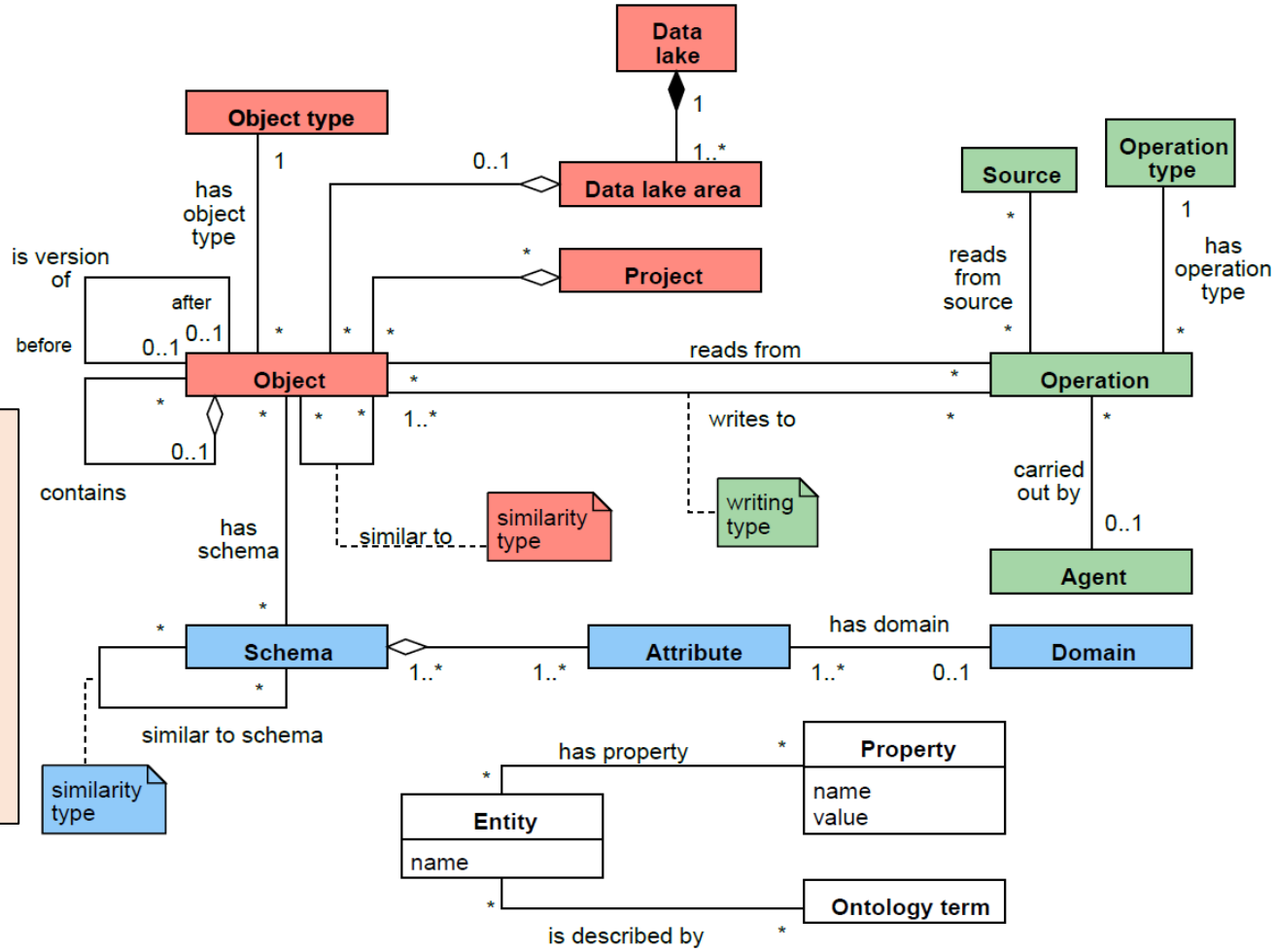
Tune the trade-off between the level of detail of the functionalities and the required computational effort




Francia, M., Gallinucci, E., Golfarelli, M., Leoni, A. G., Rizzi, S., & Santolini, N. (2021). **Making data platforms smarter with MOSES**. *Future Generation Computer Systems*, 125, 299-313.

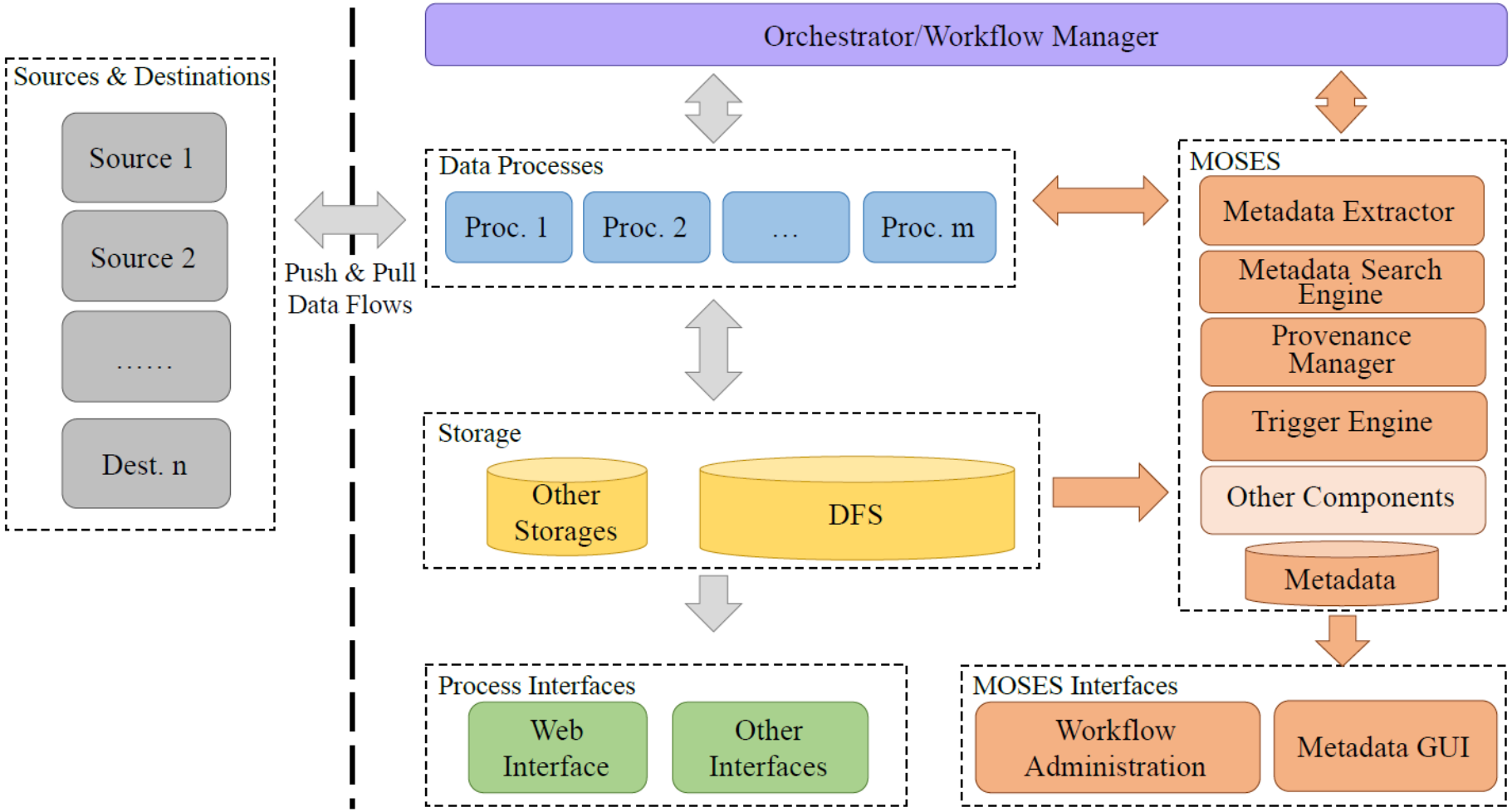
# Knowledge representation

Functionalities
- ✓ Semantic enrichment
- x Data indexing
- ✓ Link generation
- ✓ Data polymorphism
- ✓ Data versioning
- ✓ Usage tracking




Francia, M., Gallinucci, E., Golfarelli, M., Leoni, A. G., Rizzi, S., & Santolini, N. (2021). **Making data platforms smarter with MOSES**. *Future Generation Computer Systems*, 125, 299-313.

# Architectural reference



# Knowledge exploitation

Capturing the metadata

Object profiling and search

Provenance and versioning

Orchestration support

# Capturing the metadata

## Pull strategy

- The system actively collects new metadata
- Requires scheduling: when does the system activate itself?
  - Event-based (CRUD)
  - Time-based
- Requires wrappers: what does the system capture?
  - Based on data type and/or application
  - A comprehensive monitoring is practically unfeasible

## Push strategy

- The system passively receives new metadata
- Requires an API layer
- Mandatory for operational metadata

# Object profiling and search

Discoverability is a key requirement for data platforms

- Simple searches to let users locate “known” information
- Data exploration to let users uncover “unknown” information
- Common goal: identification and description of Objects

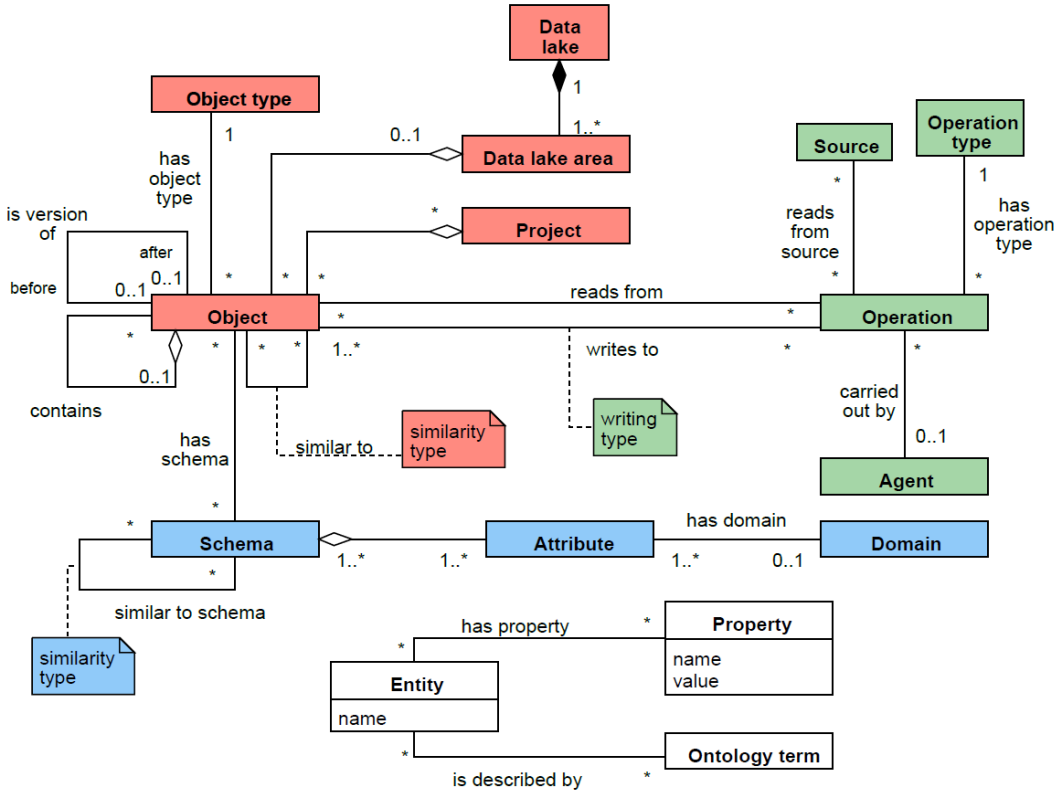
Two levels of querying

- Metadata level (most important)
- Data level (can be coupled with the first one)

# Object profiling and search

## Basic search

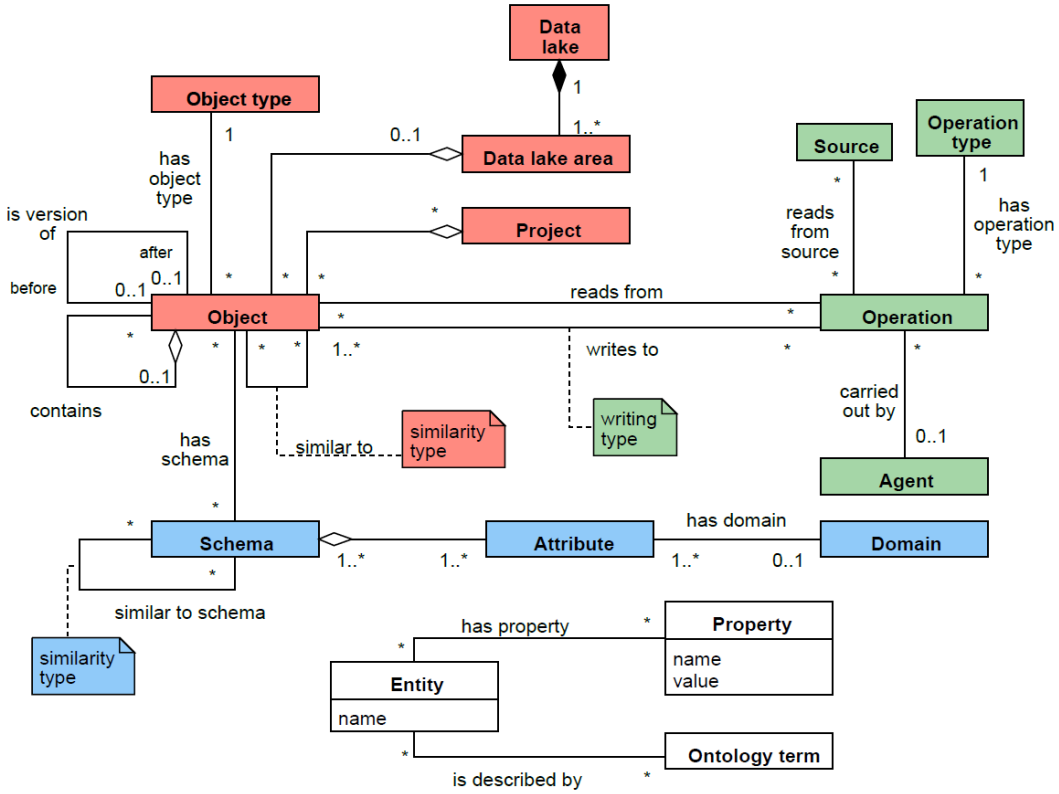
- MATCH (o:Object)-[]-(:Project {name:"ABC"})
  RETURN o
  - Return all objects of a given project
- MATCH (o:Object)-[]-(d:DataLakeArea)
  WHERE d.name = "Landing"
  AND o.name LIKE "2021_%"
  AND o.size < 100.000
  RETURN o
  - Return small objects with a given name pattern in the landing area



# Object profiling and search

## Schema-driven search

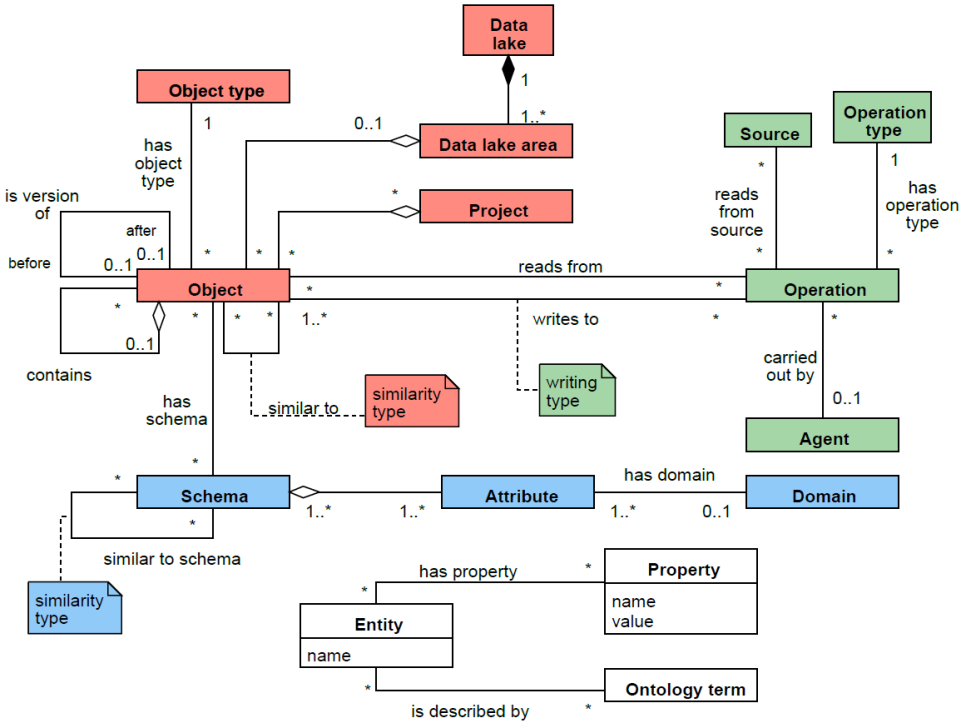
- MATCH (o:Object)-[]-(:Schema)-[]-(a:Attribute),
  (a)-[]-(:Domain {name: "FiscalCode"})
  RETURN o
  - Return objects that contain information referring to a given Domain



# Object profiling and search

## Provenance-driven search

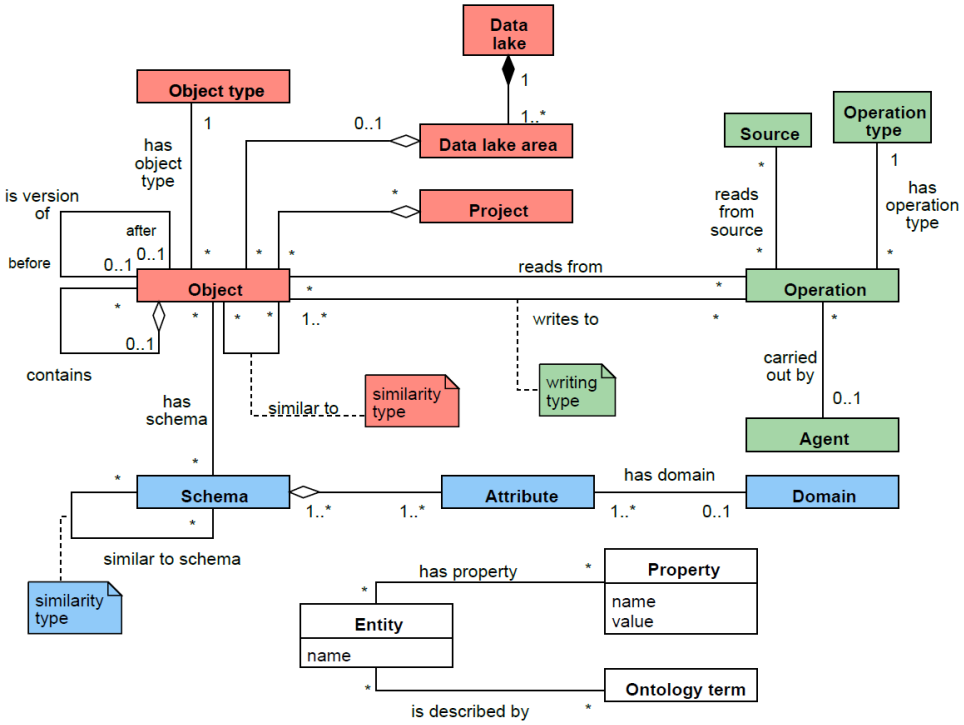
- MATCH (obj1:Object)-[:readsFrom]-(o:Operation)-[:writesTo]-(obj2:Object)
  CREATE (obj1)-[:ancestorOf]->(obj2)
- MATCH (:Object {id:123})-[:ancestorOf*]-(obj:Object)
  RETURN obj
  - Discover objects obtained from a given ancestor
- MATCH (obj:Object)-[:ancestorOf*]-(:Object {id:123})
  RETURN obj
  - Discover object(s) from which another has originated
- Example: a ML team wants to use datasets that were publicized as *canonical* for certain domains, but they find these datasets being too “groomed” for ML
  - Provenance links can be used to browse upstream and identify the less-groomed datasets that were used to derive the canonical datasets



# Object profiling and search

## Similarity-driven search

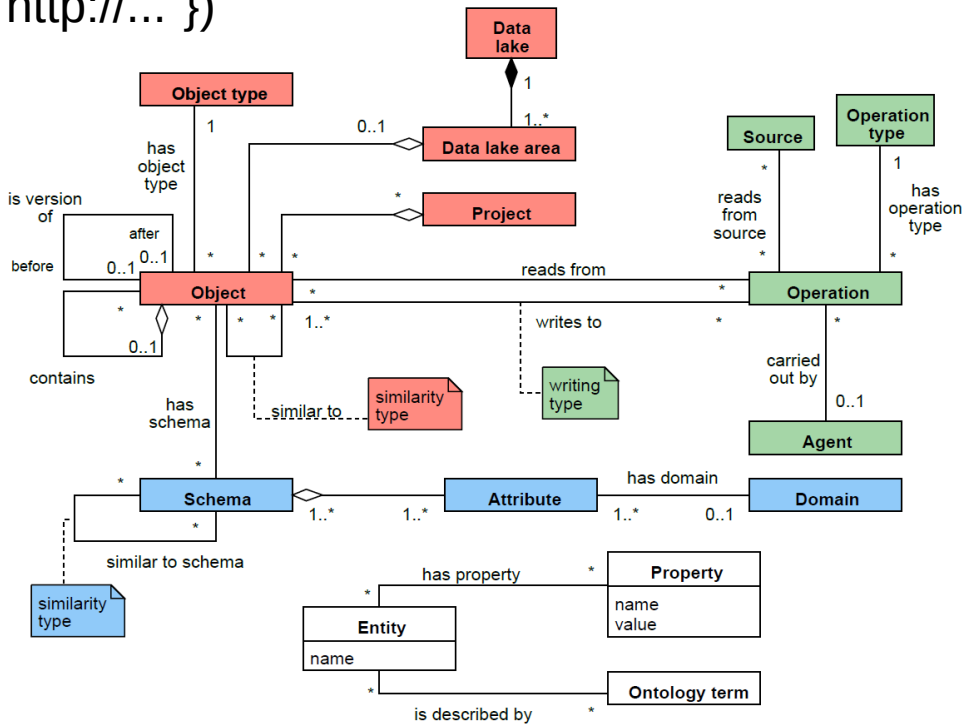
- MATCH (:Object {id:123})-[r:similarTo]-(o:Object)
  WHERE r.similarityType="affinity"
  RETURN o
  - Discover datasets to be merged in a certain query
- MATCH (:Object {id:123})-[r:similarTo]-(o:Object)
  WHERE r.similarityType="joinability"
  RETURN o
  - Discover datasets to be joined in a certain query
- Group similar objects and enrich the search results
  - List the main objects from each group
  - Restrict the search to the objects of a single group



# Object profiling and search

## Semantics-driven search

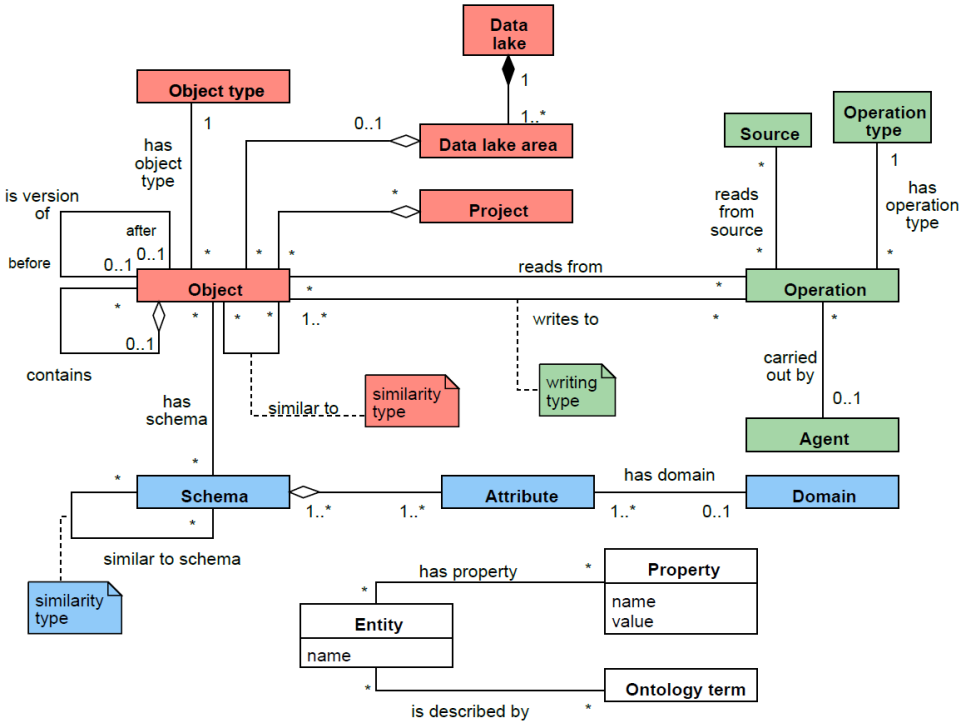
- MATCH (o:Object)-[:isDescribedBy]-(:OntologyTerm {uri:"http://..."})
  RETURN o
- MATCH (o:Object)-[*]-(any),
  (any)-[:isDescribedBy]-(:OntologyTerm {uri:"http://..."})
  RETURN o
  - Search objects without having any knowledge of their physical or intensional properties, but simply exploiting their traceability to a certain semantic concept



# Object profiling and search

## Profiling

- MATCH (o:Object)-[]-(:OntologyType {name:"Table"}),
  (o)-[]-(s:Schema)-[]-(a:Attribute),
  (o)-[r:similarTo]-(o2:Object),
  (o)-[:ancestorOf]-(o3:Object),
  (o4:Object)-[:ancestorOf]-(o)
  RETURN o, s, a, r, o2, o3, o4
  - Shows an object's properties, list the relationships with other objects in terms of similarity and provenance
  - Compute a representation of the intensional features that mostly characterize a group of objects (see slides on schema heterogeneity)



# Provenance and versioning

Provenance: metadata pertaining to the history of a data item

- Any information that describes the production process of an end product
- Encompasses meta-data about entities, data, processes, activities, and persons involved in the production process
- Essentially, it describes a transformation pipeline, including the origin of objects and the operations they are subject to

J.Wang, D. Crawl, S. Purawat, M. H. Nguyen, I. Altintas, **Big data provenance: Challenges, state of the art and opportunities**, in: *Proc. BigData*, Santa Clara, CA, USA, 2015, pp. 2509–2516.

M. Herschel, R. Diestelk¨amper, H. Ben Lahmar, **A survey on provenance: What for? What form? What from?**, *VLDB J.* 26 (6) (2017) 881–906.

# Provenance and versioning

## Several use cases

- Supply chain
  - Assess food's quality and gain trust in food products
- Scientific experiments
  - **Experiments on big data scales are not easily repeatable** (e.g., 100 PB workloads @CERN, SKA)
  - Ensuring the reproducibility and accessibility of the scientific results is essential
- Complex data processing
  - The development of complex data processing pipelines is iterative/incremental
  - Provenance analysis leads to faster and higher-quality analysis, debugging, and refinement

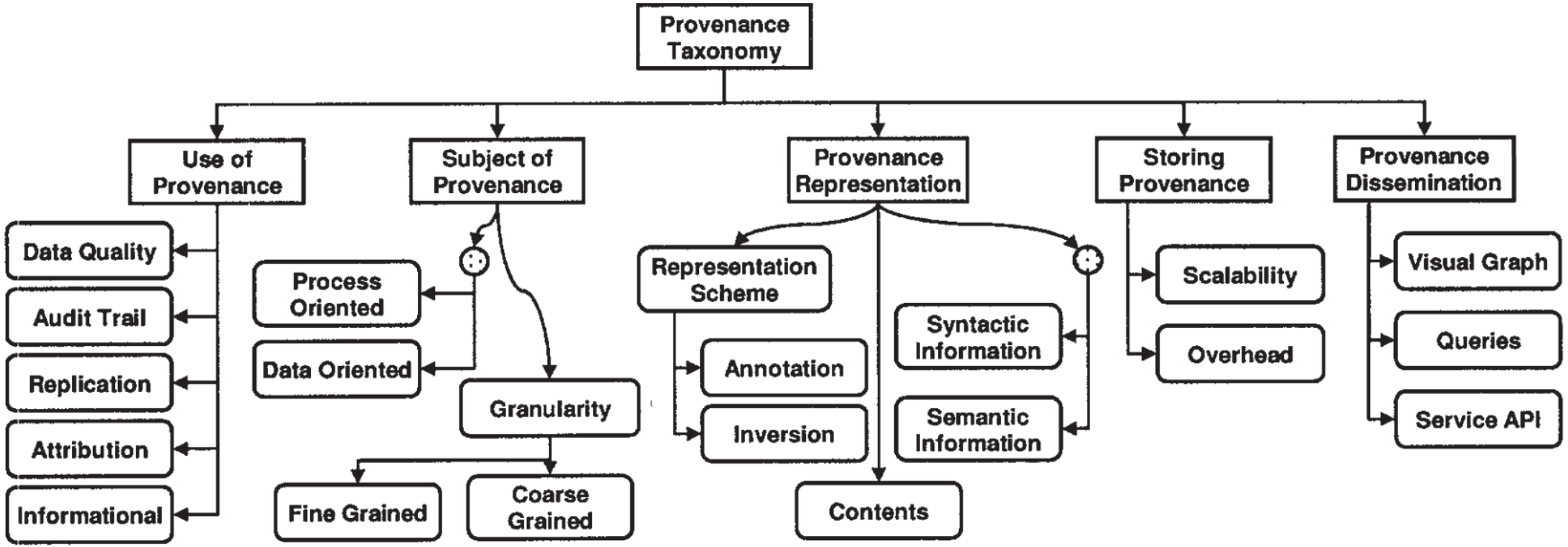
New, S.: **The transparent supply chain**. *Harvard Bus. Rev.88*, 1–5(2010)

Cranmer, K., Heinrich, L., Jones, R., South, D.M.: **Analysis preservation in ATLAS**. *J. Physi*.664(3) (2015).

Bourhis, P., Deutch, D., Moskovitch, Y.: **POLYTICS: provenance-based analytics of data-centric applications**. In: *IEEE International Conference on Data Engineering (ICDE)*, pp. 1373–1374(2017)

# Provenance and versioning

A taxonomy of provenance techniques




Y. Simmhan, B. Plale, D. Gannon, **A survey of data provenance in e-science**,*SIGMOD Rec*. 34 (3) (2005) 31–36.

# Provenance and versioning

## Provenance functionalities (activated by metadata)

- Data quality
  - Monitoring accuracy, precision, and recall of produced objects to notify the data scientist when a transformation pipeline is not behaving as expected
- Debugging
  - Inferring the cause of pipeline failures is challenging and requires an investigation of the overall processing history, including input objects and the environmental settings
- Reproducibility
  - Re-execution of all or part of the operations belonging to a pipeline
- Trustworthiness
  - Help data scientists to trust the objects produced by tracing them back to their sources and storing the agents who operated on those objects
- Versioning
  - Marking a generated object and its versions (e.g., due to changes in a database schema) helps in identifying relevant objects along with their semantic versions, and to operate with legacy objects

# Provenance and versioning

An important aspect is the granularity of provenance

- Fine-grained provenance is typically used for single vertical applications
  - It requires to collect huge amounts of detailed information to enable a very detailed tracing
- Coarse-grained provenance is appropriate to ensure a broad coverage of highly heterogeneous transformations possibly involving several applications and datasets

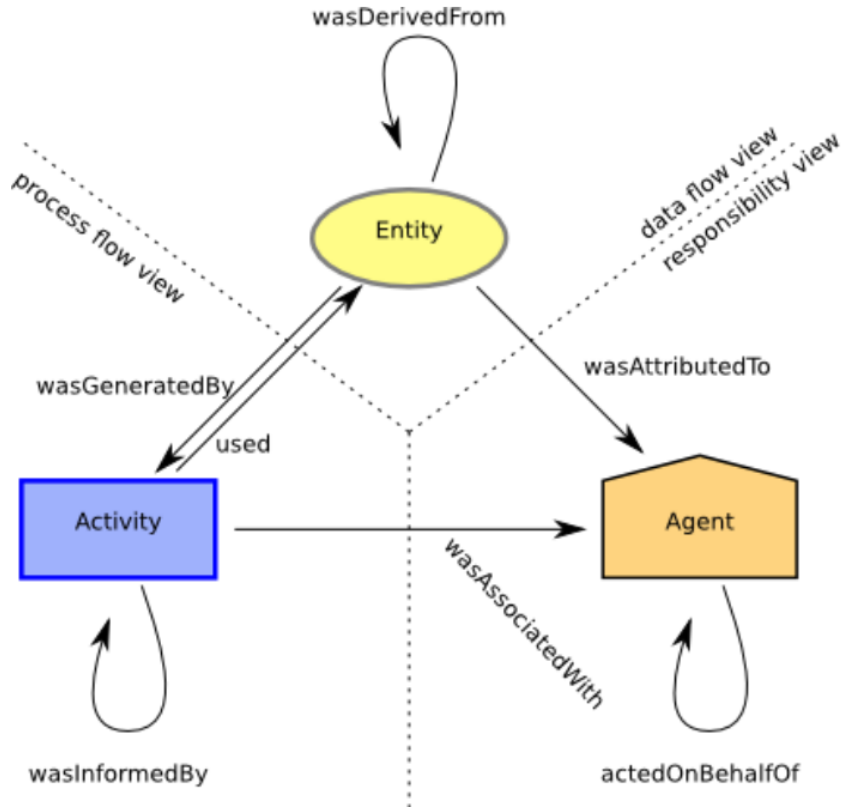
Choosing a granularity is the result of a trade-off between accuracy and computational effort

- Storing only the name and the version of a clustering algorithm enables an approximate reproducibility of the results
- Storing all its parameters makes this functionality much more accurate

# Provenance and versioning

PROV: a standard for provenance modeling

- Several tools exists for managing PROV metadata
  - https://openprovenance.org/services/view/translator
  - https://lucmoreau.github.io/ProvToolbox/
  - https://prov.readthedocs.io/en/latest/
- Compliance with PROV ensures integration with existing tools for querying and visualization




L. Moreau, P. T. Groth, **Provenance: An Introduction to PROV**, *Synthesis Lectures on the Semantic Web: Theory and Technology*, Morgan & Claypool Publishers, 2013.

# Provenance and versioning

## Some current research directions

- Expand PROV to better suite big data scenarios
  - Y. Gao, X. Chen and X. Du, **A Big Data Provenance Model for Data Security Supervision Based on PROV-DM Model**, in *IEEE Access*, vol. 8, pp. 38742-38752, 2020.
- Define provenance-based approaches to measure the quality of big data
  - Taleb, I., Serhani, M.A., Bouhaddioui, C. et al. **Big data quality framework: a holistic approach to continuous quality management**. *J Big Data* 8, 76 (2021).
- An outline of the challenges, including granularity identification, integration, security concerns
  - A. Chacko and S. D. Madhu Kumar, **Big data provenance research directions**, *TENCON 2017 - 2017 IEEE Region 10 Conference*, 2017, pp. 651-656, doi: 10.1109/TENCON.2017.8227942.
- Blockchain-based provenance systems
  - Dang, T. K., & Duong, T. A. (2021). **An effective and elastic blockchain-based provenance preserving solution for the open data**. *International Journal of Web Information Systems*.
  - Ruan, P., Dinh, T. T. A., Lin, Q., Zhang, M., Chen, G., & Ooi, B. C. (2021). **LineageChain: a fine-grained, secure and efficient data provenance system for blockchains**. *The VLDB Journal*, *30*(1), 3-24.

# Orchestration support

The orchestrator is the component in charge of controlling the execution of computation activities

- Either through a regular scheduling of the activities
- Or by triggering a process in response to a certain event

Several entities (either processes or human beings) can cover this role to activate some data processes

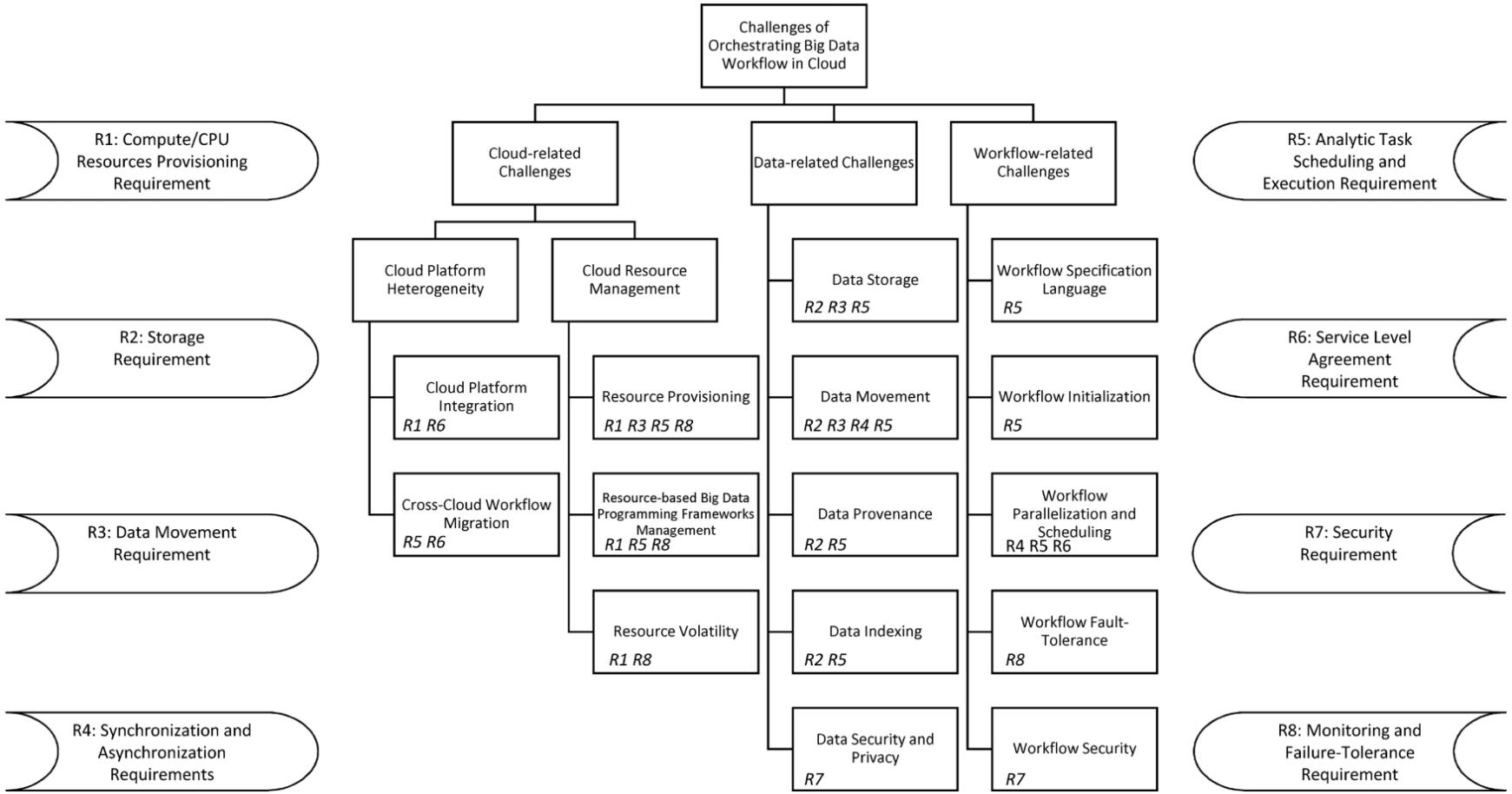
# Orchestration support

## Orchestration functionalities (activated by metadata)

- Dynamic/condition-based behavior
  - Decide *what* data process should be activated under different conditions
  - Decide *how* to tune the parameters in case of parametric data processes
- Triggering
  - Decide *when* to trigger a certain data process
- Scoping
  - Assess the trustworthiness of objects to decide *if* a certain data process should be activated or not
- Resource estimation/prediction
  - Decide the optimal amount of resources required to terminate successfully while leaving sufficient resources to the other concurrent process, based on previous executions and current settings
  - Negotiate the resources with the cluster's resource manager

# Orchestration support

Orchestration requirements & challenges




Barika, M., Garg, S., Zomaya, A. Y., Wang, L., Moorsel, A. V., & Ranjan, R. (2019). **Orchestrating big data analysis workflows in the cloud: research challenges, survey, and future directions**. *ACM Computing Surveys (CSUR)*, *52*(5), 1-41.

# Orchestration support

## Orchestration requirements

- R1 Compute/CPU resource provisioning
  - Determine the right amount of resources
  - Continuously monitor and manage them in a dynamic execution environment
- R2 Storage
  - Choose the right cloud storage resource, data location, and format (if the application is parametric)
- R3 Data movement
  - Dynamically transfer large datasets between compute and storage resources
- R4 Synchronization and asynchronization
  - Manage the control and data flow dependencies across analytics tasks

Barika, M., Garg, S., Zomaya, A. Y., Wang, L., Moorsel, A. V., & Ranjan, R. (2019). **Orchestrating big data analysis workflows in the cloud: research challenges, survey, and future directions**. *ACM Computing Surveys (CSUR)*, *52*(5), 1-41.

# Orchestration support

## Orchestration requirements

- R5 Analytic task scheduling and execution
  - Scheduling and coordinating the execution of workflow tasks across diverse sets of big data programming models
  - Tracking and capturing provenance of data
- R6 Service Level Agreement
  - Executions may need to meet user-defined QoS requirements (e.g., a strict execution deadline)
- R7 Security
  - Beyond standard encryption approaches: private (anonymous) computation, verification of outcomes in multi-party settings, placement of components according to security policies
- R8 Monitoring and Failure-Tolerance
  - Ensure that everything is streamlined and executed as anticipated
  - As failures could happen at any time, handle those failures when they occur or predicting them before they happen

Barika, M., Garg, S., Zomaya, A. Y., Wang, L., Moorsel, A. V., & Ranjan, R. (2019). **Orchestrating big data analysis workflows in the cloud: research challenges, survey, and future directions**. *ACM Computing Surveys (CSUR)*, *52*(5), 1-41.

# Orchestration support

## Orchestration challenges

- Cloud Platform Heterogeneity
  - **Integration** (different APIs, virtualization formats, pricing policies, hardware/software configurations)
  - **Workflow Migration** (e.g., to aspire to specific QoS features in the target cloud or better price)
- Cloud Resource Management
  - **Resource Provisioning** (selecting the right configuration of virtual resources; the resource configuration search space grows exponentially, and the problem is often NP-complete)
  - **Resource-based Big Data Programming Frameworks Management** (automatically select the configurations for both IaaS-level resource and PaaS-level framework to consistently accomplish the anticipated workflow-level SLA requirements, while maximizing the utilization of cloud datacenter resources)
  - **Resource Volatility** (at different levels: VM-level, big data progressing framework-level and workflow task-level)

Barika, M., Garg, S., Zomaya, A. Y., Wang, L., Moorsel, A. V., & Ranjan, R. (2019). **Orchestrating big data analysis workflows in the cloud: research challenges, survey, and future directions**. *ACM Computing Surveys (CSUR)*, *52*(5), 1-41.

# Orchestration support

## Orchestration challenges

- Data-related
  - **Storage** (where the data will be residing, which data format will be used)
  - **Movement** (minimize transfer rates, exploit *data locality* in task-centric or worker-centric way)
  - **Provenance** (trade-off expressiveness with overhead)
  - **Indexing** (which dataset is worth indexing and how)
  - **Security and Privacy** (cryptography, access control, integrity, masking, etc.)

Barika, M., Garg, S., Zomaya, A. Y., Wang, L., Moorsel, A. V., & Ranjan, R. (2019). **Orchestrating big data analysis workflows in the cloud: research challenges, survey, and future directions**. *ACM Computing Surveys (CSUR)*, *52*(5), 1-41.

# Orchestration support

## Orchestration challenges

- Workflow-related
  - **Specification Language** (devising a high level, technology-/cloud-independent workflow language)
  - **Initialization** (subdivision into fragments considering dependencies, constraints, etc.)
  - **Parallelization and Scheduling** (with super-workflows defined at application and task level)
  - **Fault-Tolerance** (thing can go wrong at workflow-, application-, and cloud-level)
  - **Security** (securing workflow logic and computation)

Barika, M., Garg, S., Zomaya, A. Y., Wang, L., Moorsel, A. V., & Ranjan, R. (2019). **Orchestrating big data analysis workflows in the cloud: research challenges, survey, and future directions**. *ACM Computing Surveys (CSUR)*, *52*(5), 1-41.

# Metadata challenge - Conclusions

Some predictions for 2022

- Data lakes and data warehouses will become indistinguishable
  - Analytics will merge with SQL-based systems within data platforms
- Universal standards for governance, lineage, and metrics will begin to emerge
- Knowledge graphs will be in high demand
- The next generation of data sharing will require domain-oriented governance within (and between) organizations

Bob Muglia, 2021, https://towardsdatascience.com/the-future-of-the-modern-data-stack-2de175b3c809