

Smart Data Platform Management

Challenges and Applications

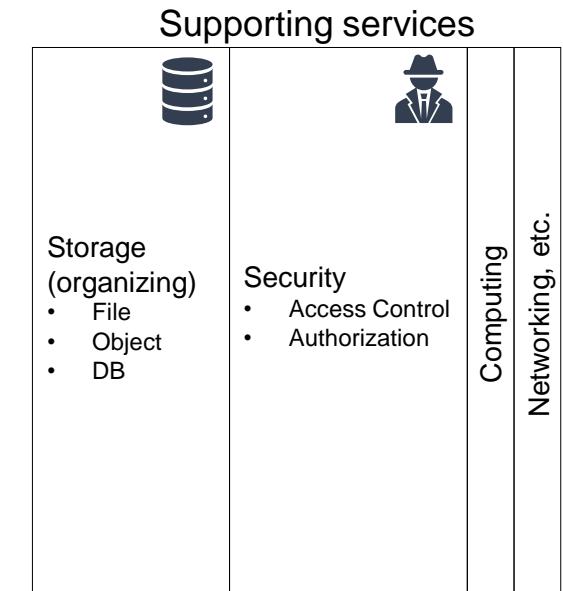
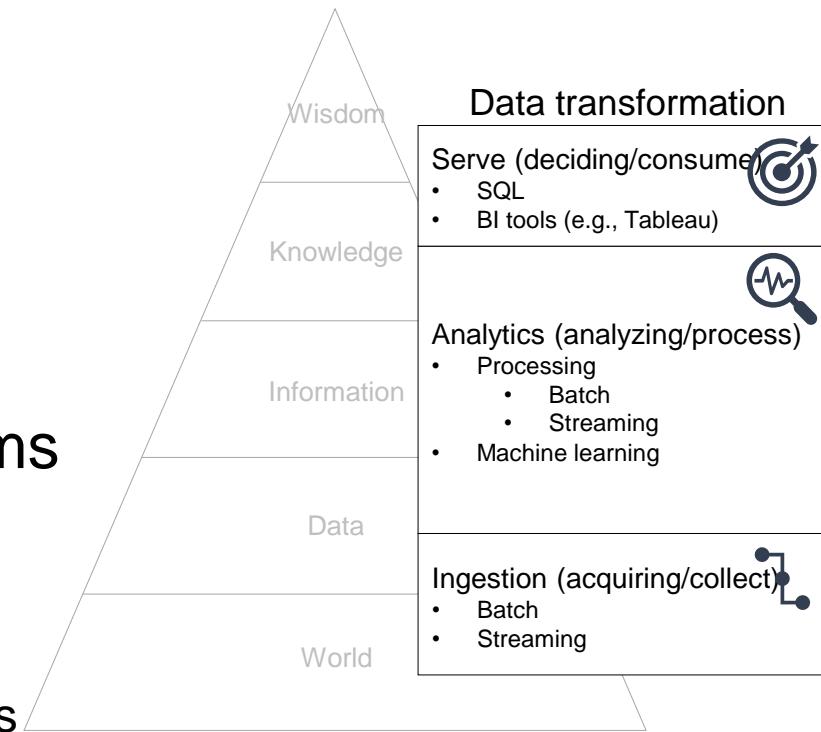
Data platform

We have services

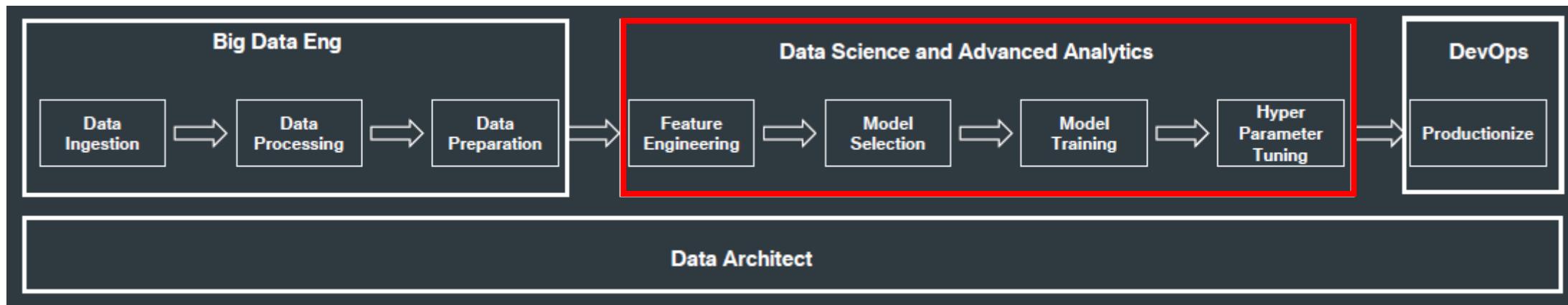
- To transform data
- To support the transformation

The (DIKW) pyramid abstracts many techniques and algorithms

- Standardization
- Integration
- Orchestration
- Accessibility through APIs



Data platform



Data platform: related job positions

Data platform engineer

- Orchestrate the successful implementation of cloud technologies within the data infrastructure of their business
- Solid understanding of impact database types and implementation
- Responsible for purchasing decisions for cloud services and approval of data architectures

Data architect

- Team members who understand all aspects of a data platform's architecture
- Work closely with the data platform engineers to create data workflows
- Responsible for designing and testing new database architectures and planning both data and architecture migrations

Data pipeline engineer

- Responsible for planning, architecting, and building large-scale data processing systems

Data analyst

- Analyze data systems, creating automated systems for retrieving data from the data platform
- Cloud data analysts are more commonly members of the business user population

Data scientist

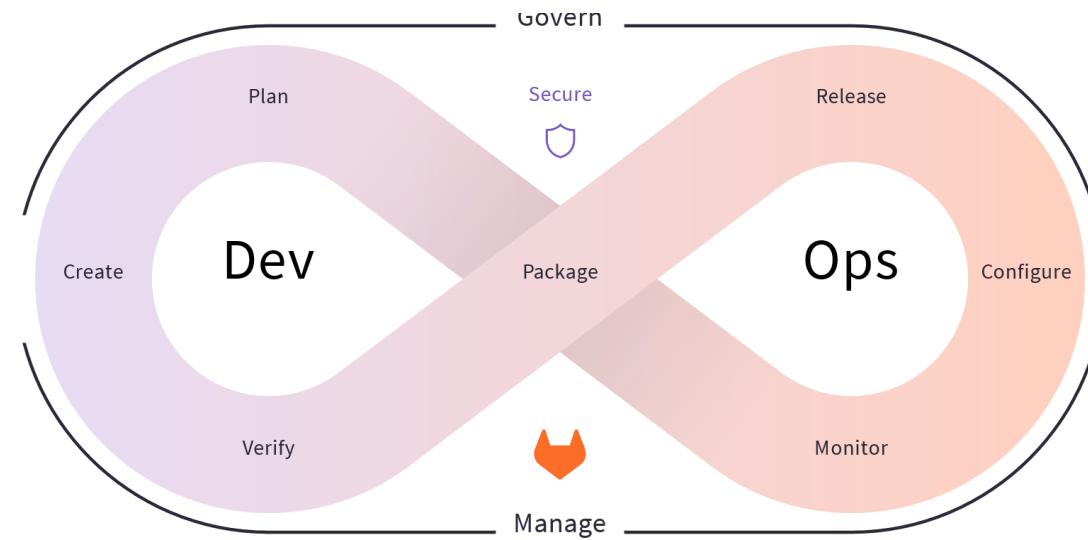
- Analyze and interpret complex digital data
- Work with new technologies (e.g., machine learning) to deepen the business' understanding and gain new insights

From DevOps...

DevOps combines development and operations to increase the efficiency, speed, and security of software development and delivery compared to traditional processes.

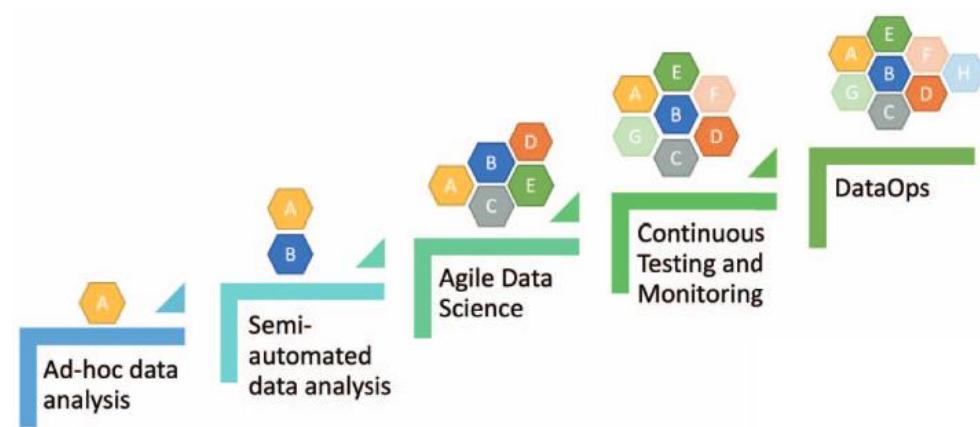
DevOps practices enable software development (dev) and operations (ops) teams to accelerate delivery through automation, collaboration, fast feedback, and iterative improvement

<https://about.gitlab.com/topics/devops/> (accessed 2023-06-03)



... to DataOps

DataOps refers to a general process aimed to shorten the end-to-end data analytic life-cycle time by introducing automation in the data collection, validation, and verification process



Case	Use cases at Ericsson	Interviewed Experts	
		ID	Role
A	Automated data collection for data analytics	R4	Senior Data Scientist
B	Building data pipelines	R1	Integration and Operations Professional
C	Toolkit for Network Analytics	R2	Analytics System Architect
D	Building CI pipelines for Data Scientist team	R7	Data Scientist
E	Tracking the Software Version	R5	Senior Customer Support Engineer
F	Testing the Software Quality	R6	Developer Customer Support
G	KPI Analysis Software	R3	Senior Data Engineer
H	Building data pipelines for CI and CD data	R8	Program Manager

Munappy, A. R., Mattos, D. I., Bosch, J., Olsson, H. H., & Dakkak, A. (2020, June). From ad-hoc data analytics to dataops. In *Proceedings of the International Conference on Software and System Processes* (pp. 165-174).

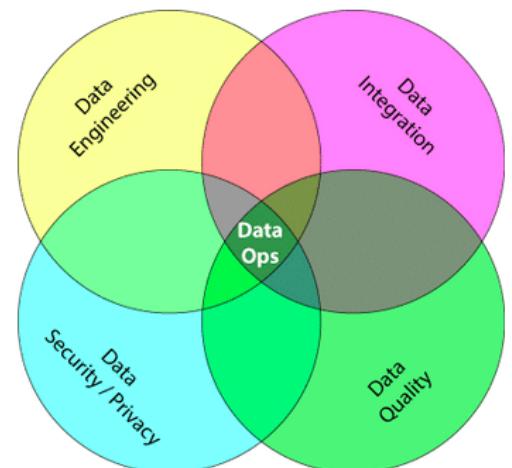
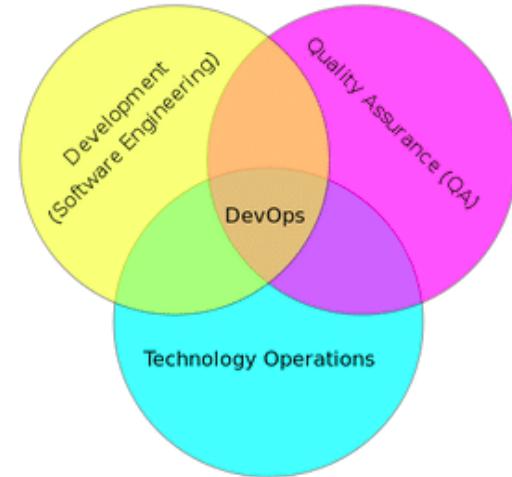
DataOps

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 fabric

“vision for data management [...] that seamlessly connects different clouds, whether they are private, public, or hybrid environments.” (2016)

Frictionless access and sharing of data in a distributed data environment

- Enables a **single and consistent data management framework**, which allows seamless data access and processing by design across otherwise siloed storage
- Leverages **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**

<https://cloud.netapp.com/hubfs/Data-Fabric/Data%20Fabric%20WP%20April%202017.pdf> (accessed 2023-06-23)

Gartner, 2019 <https://www.gartner.com/en/newsroom/press-releases/2019-02-18-gartner-identifies-top-10-data-and-analytics-technologies>

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

- **Catalog all your data:** including business glossary, data lineage, and data quality metrics
- **Enable self-service capabilities:** data discovery, search, and consumption of data-as-a-product
- **Provide a knowledge graph:** Visualizing how data is interconnected, deriving additional actionable insights
- **Provide intelligent (smart) information integration:** automating the process of bringing data from disparate sources alike in their data integration and transformation, including AI/ML-based automated generation, adjustments, and enforcement of rules and policies
- **Derive insight from metadata:** Orchestrating an end-to-end unified lifecycle of data management, integration, data engineering, and data governance end to end
- **Enforce local and global data rules/policies:** Including AI/ML-based automated generation, adjustments, and enforcement of rules and policies
- **Manage an end-to-end unified lifecycle:** Implementing a coherent and consistent lifecycle end to end of all Data Fabric tasks across various platforms, personas, and organizations
- **Enforce data and AI governance:** Broadening the scope of traditional data governance to include AI artefacts, for example, AI models, pipelines

Is this brand new?

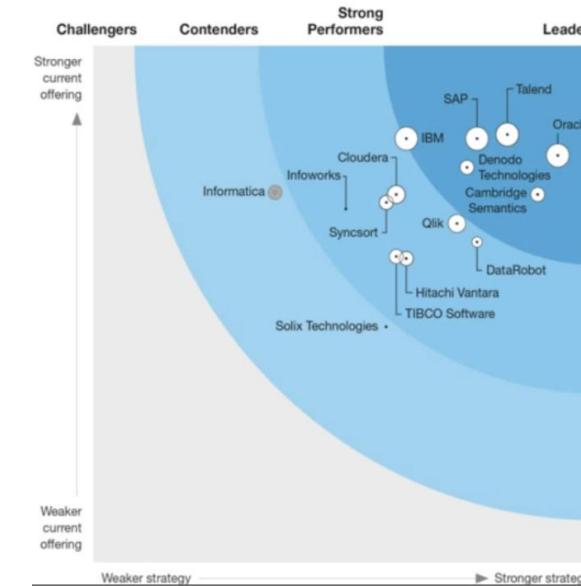
S

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

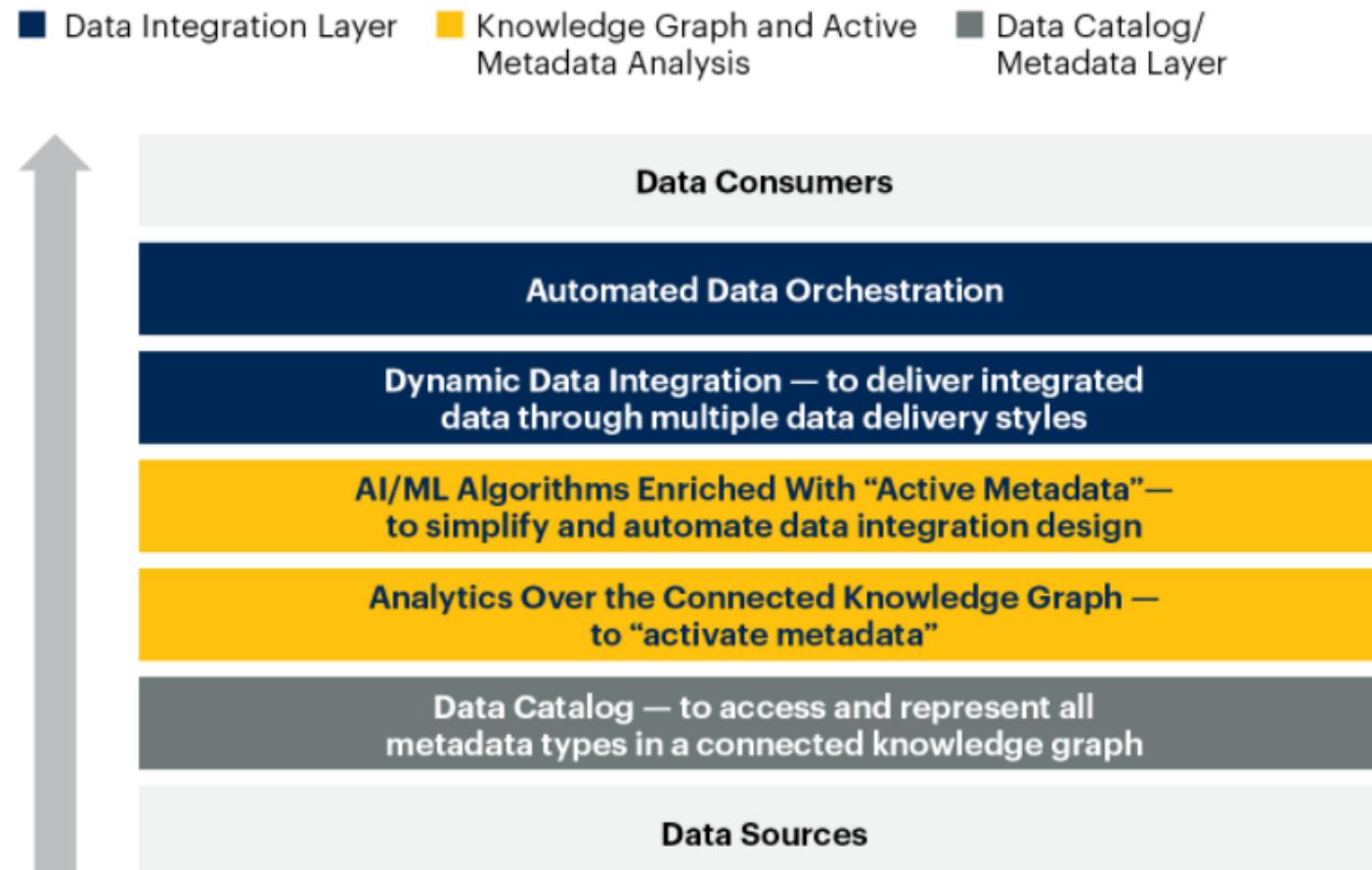
Cambridge Semantics	Anzo, AnzoGraph
Cloudera	Cloudera Data Platform
DataRobot	Paxata
Denodo Technologies	Denodo Platform
Hitachi Vantara	Lumada Data Services
IBM	IBM Cloud Pak for Data
Informatica	Informatica Intelligent Data Management
Infoworks	DataFoundry
Oracle	Oracle GoldenGate, Oracle Autonomous 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>

Data fabric



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

Data mesh

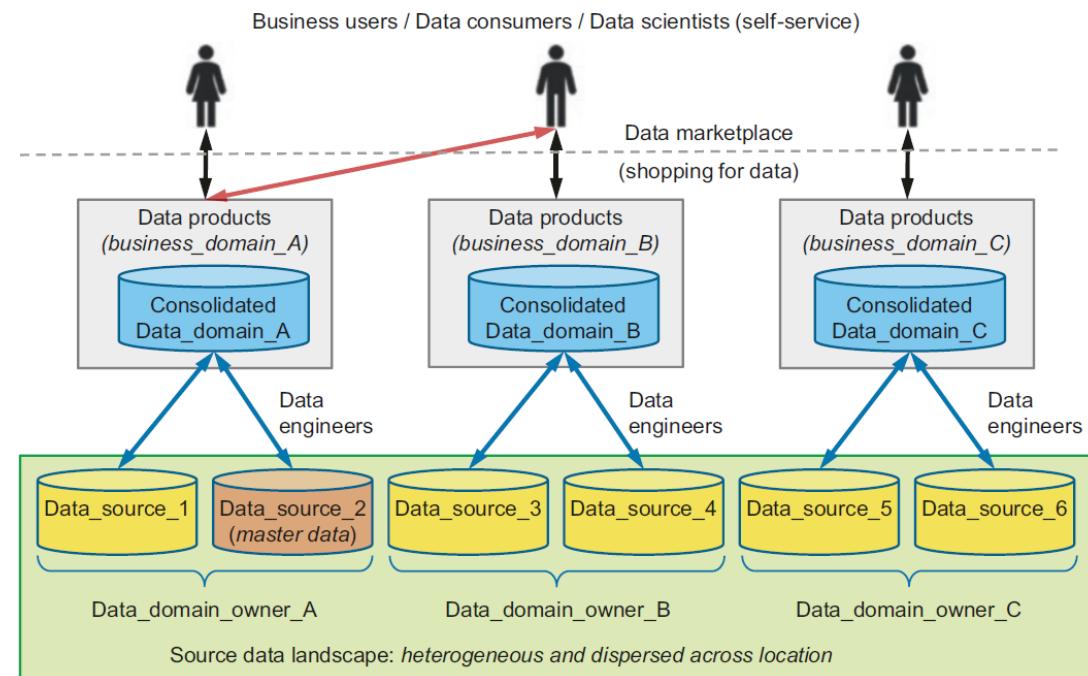
Distributed data architecture, under centralized governance and standardization for interoperability, enabled by a shared and harmonized self-serve data infrastructure

- 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

Data mesh

Data Mesh organizes data around **business domain owners** and transforms relevant data assets (data sources) to **data products** that can be consumed by distributed business users from various business domains or functions

- Data products are created, governed, and used in an **autonomous, decentralized**, and self-service manner
- **Self-service capabilities**, which we have already referenced as a Data Fabric capability, enable business organizations to entertain a data marketplace with shopping-for-data characteristics



What makes data a product?

A **data product** is raw data transformed into a business context

- Data products are registered in **knowledge catalog** through specifications (XML, JSON, etc.)
- Main features
 - **Data product description:** The data product needs to be well described
 - **Access methods:** for example, REST APIs, SQL, NoSQL, etc., and where to find the data asset
 - **Policies and rules:** who is allowed to consume the data product for what purpose
 - **SLAs:** agreements regarding the data product availability, performance characteristics, functions, cost of data product usage
 - **Defined format:** A data product needs to be described using a defined format
 - **Cataloged:** All data products need to be registered in the knowledge catalog. Data products need to be searchable and discoverable by potential data product consumers and business user
- Data products themselves are not stored in the knowledge catalog

Data mesh vs data fabric

They are design 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
 - Gartner estimates 25% of data management vendors will provide a complete data fabric solution by 2024 – up from 5% today

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>

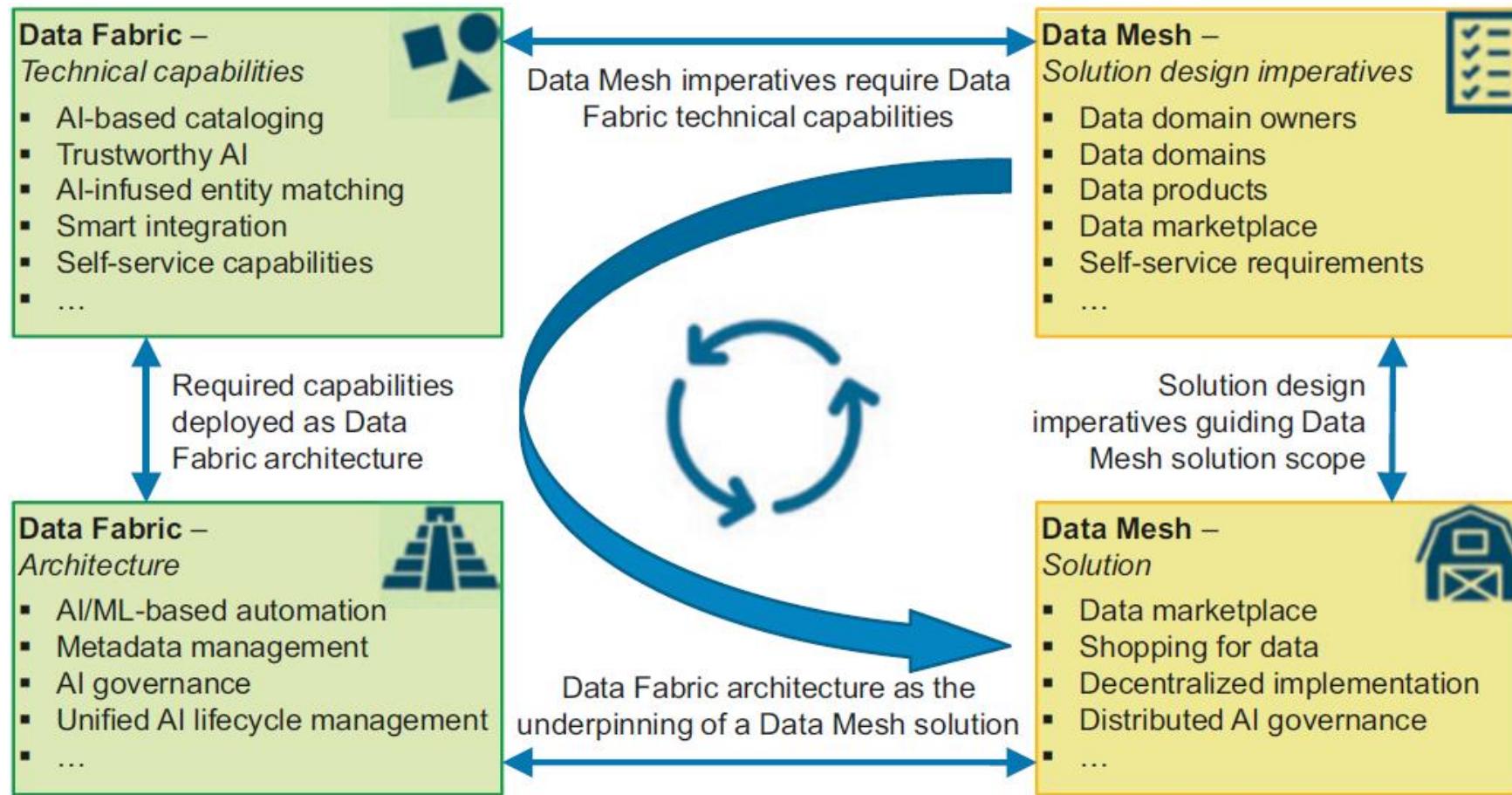
Data mesh vs data fabric

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
 - Focus on the architectural, technical capabilities, and intelligent analysis to produce active metadata supporting a smarter, AI-infused system to orchestrate various data integration styles
- **Data mesh**
 - Emphasis on decentralization and data domain autonomy
 - Focuses on organizational change; it is more about people and process
 - Data are primarily organized around domain owners who create business-focused data products, which can be aggregated and consumed across distributed consumers

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 mesh vs data fabric



Data mesh vs data fabric

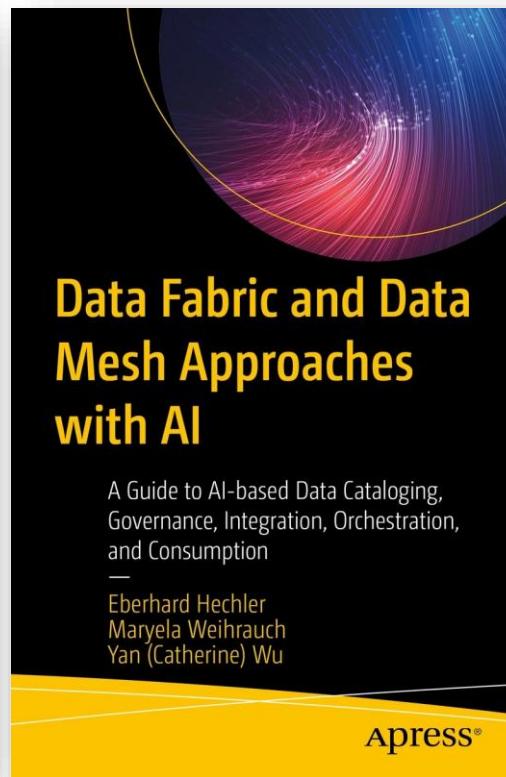
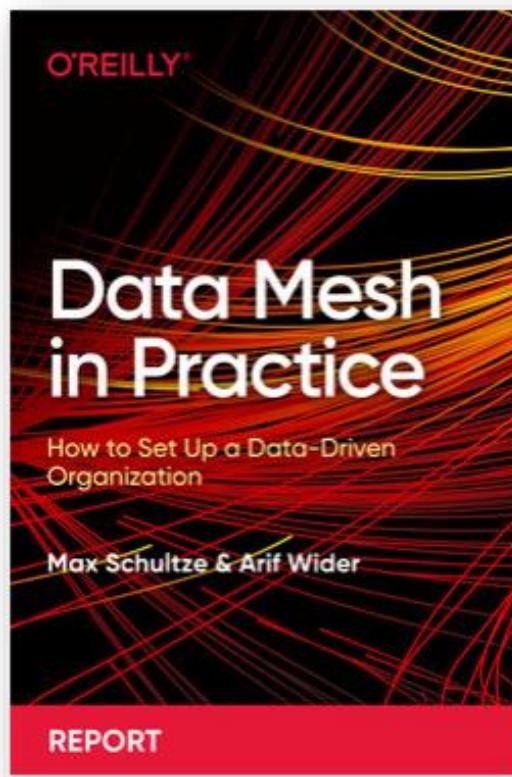
Data Fabric and Mesh are the results from the data architecture evolution

- **Many capabilities were in existence already long before** the terms were coined

Take away:

- Abstract the “building blocks” of such platforms
- Let them evolve according to scalability and flexibility requirements

(Some) References



Metadata Challenges

Metadata challenges

- Lacking smart support to govern the complexity of data and transformations
- Data transformations must be governed to prevent DP turning into a swamp
 - Amplified in data science, with data scientists prevailing data architects
 - Leverage descriptive metadata and maintenance to keep control over 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 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

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

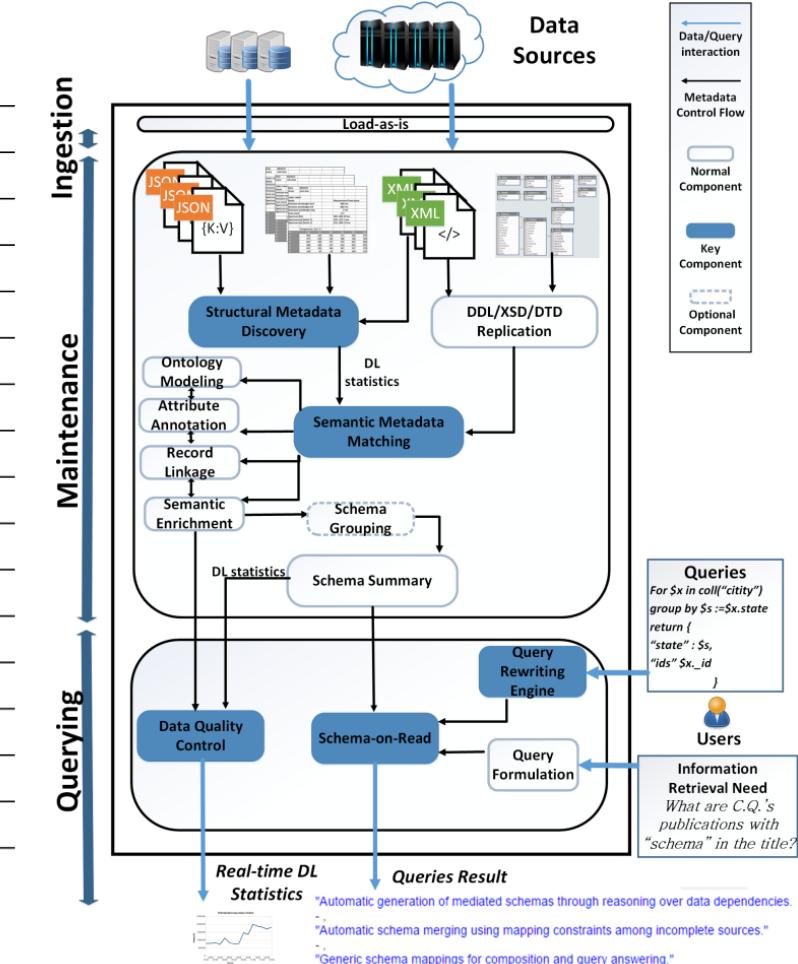
Table 1: Features provided by data lake metadata systems

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]	◆		✓			✓	
Few details given on metamodel and functionalities. KA No metadata collected on operations.							
Diamantini et al. (2018) [5]	◊	✓		✓	✓		

◆ : 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).



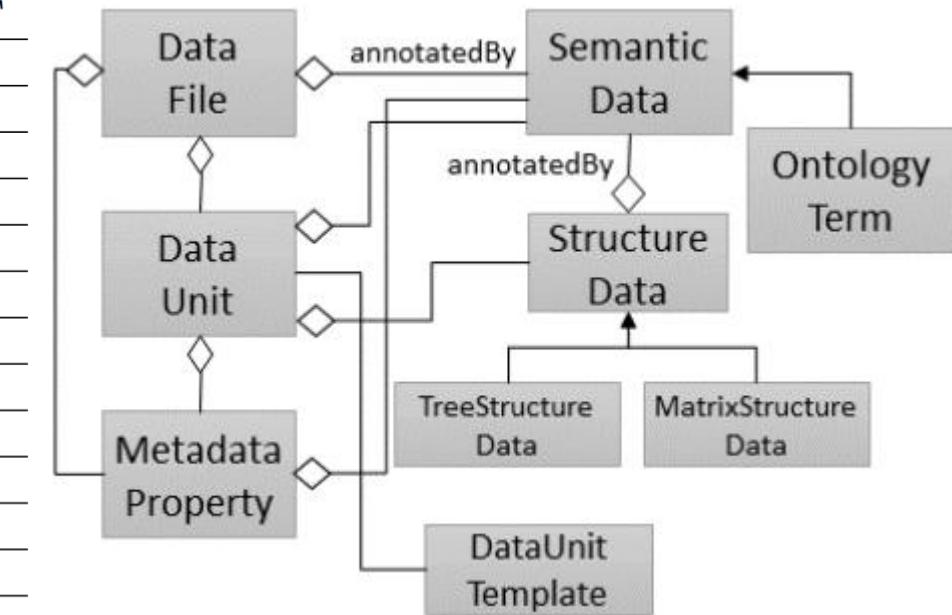
Knowledge representation

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								
No discussion about the functionalities provided. No metadata collected on operations and agents.								
Diamantini et al. (2018) [5]	◊		✓		✓	✓		

- ◆ : 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.

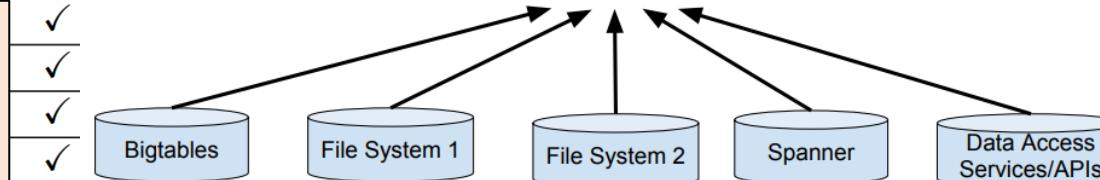
Farrugia et al. (2016) [9] ♦

GOODS (Halevy et al., 2016) [12] ♦ ✓

CoreDB (Beheshti et al., 2017) [3] ♦

Strictly coupled with the Google platform.
Mainly focuses on object description and searches.

No formal description of the metamodel.



: 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

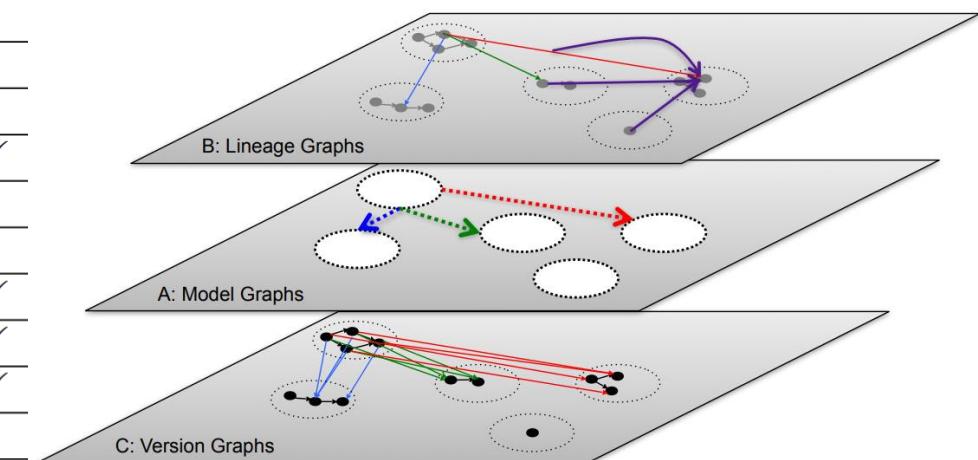
Table 1: Features provided by data lake metadata systems

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

	DI	LG	DP	DV	UT
GEMMS (Quix et al., 2016) [22] ◇	✓			✓	
Not enough details given to clarify which metadata are actually handled.			✓	✓	
Functionalities are described at a high level.		✓	✓		
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]	◆	✓	

Support users in creating and optimizing
the data processing pipelines.

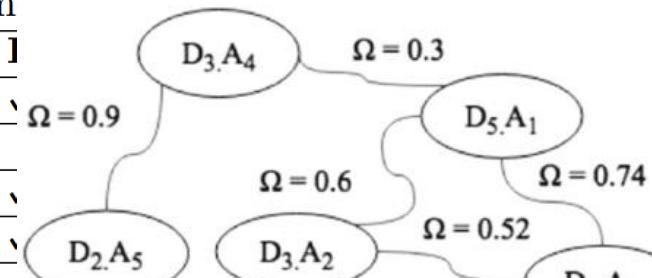
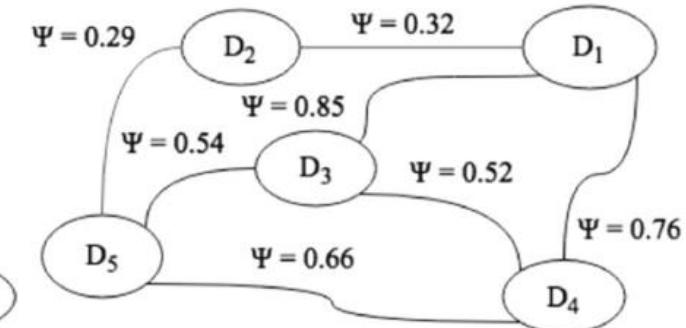
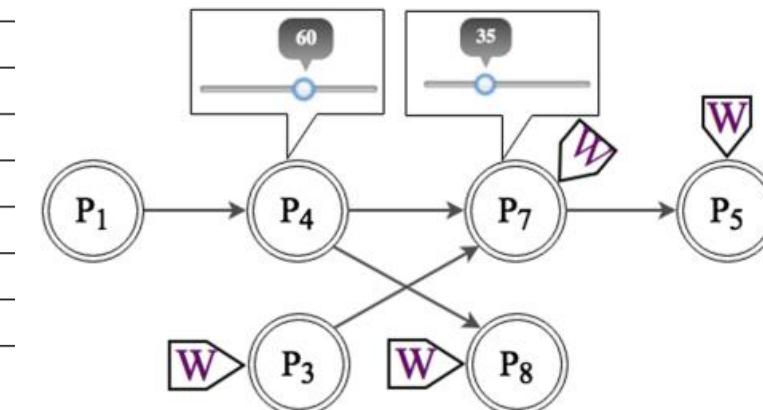
Only goal-related metadata are collected.

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

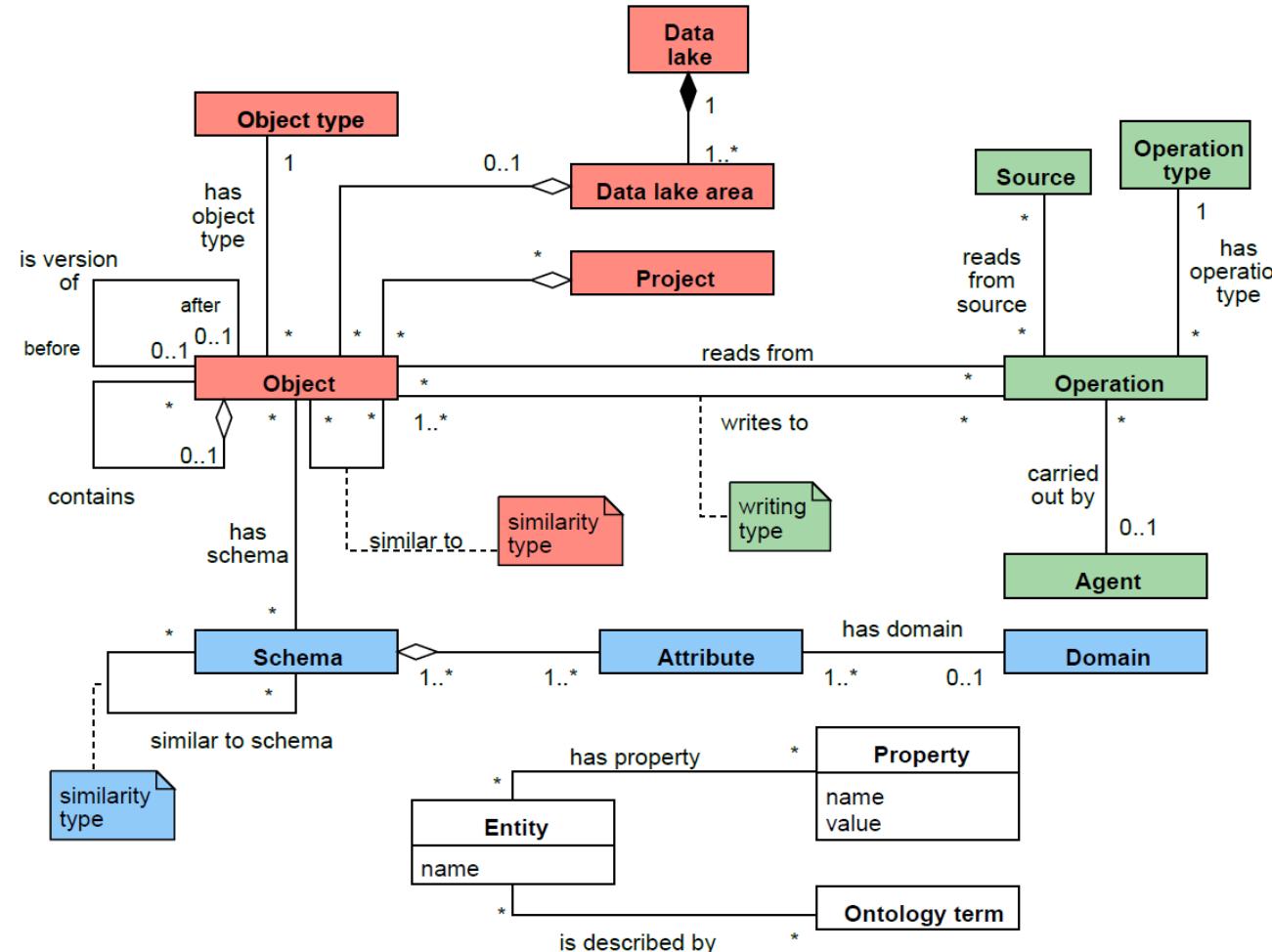
‡ : Model or implementation assimilable to a data lake

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.

(a) Affinity of attributes A_j (b) Joinability of datasets D_i 

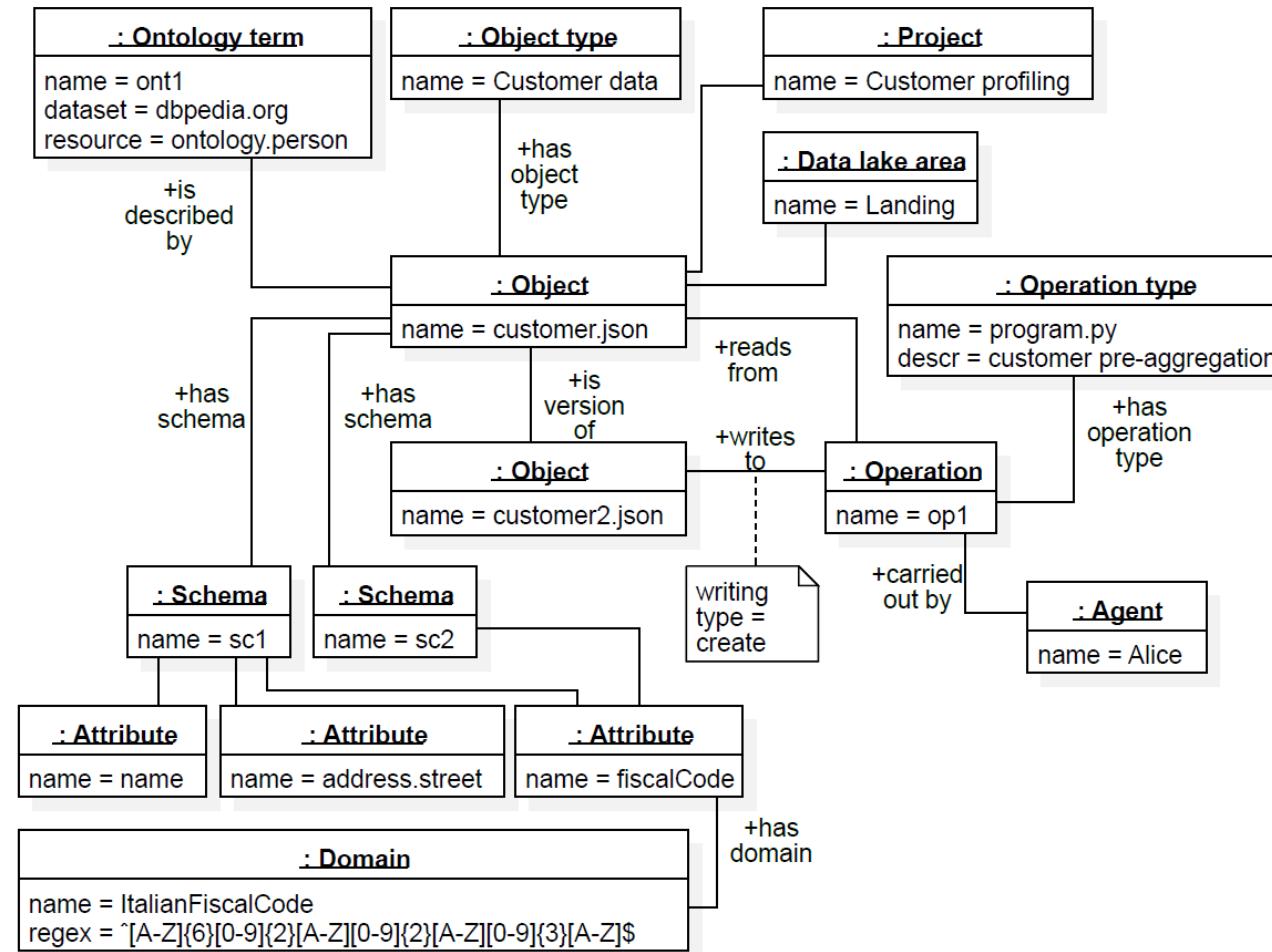
Knowledge representation

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.

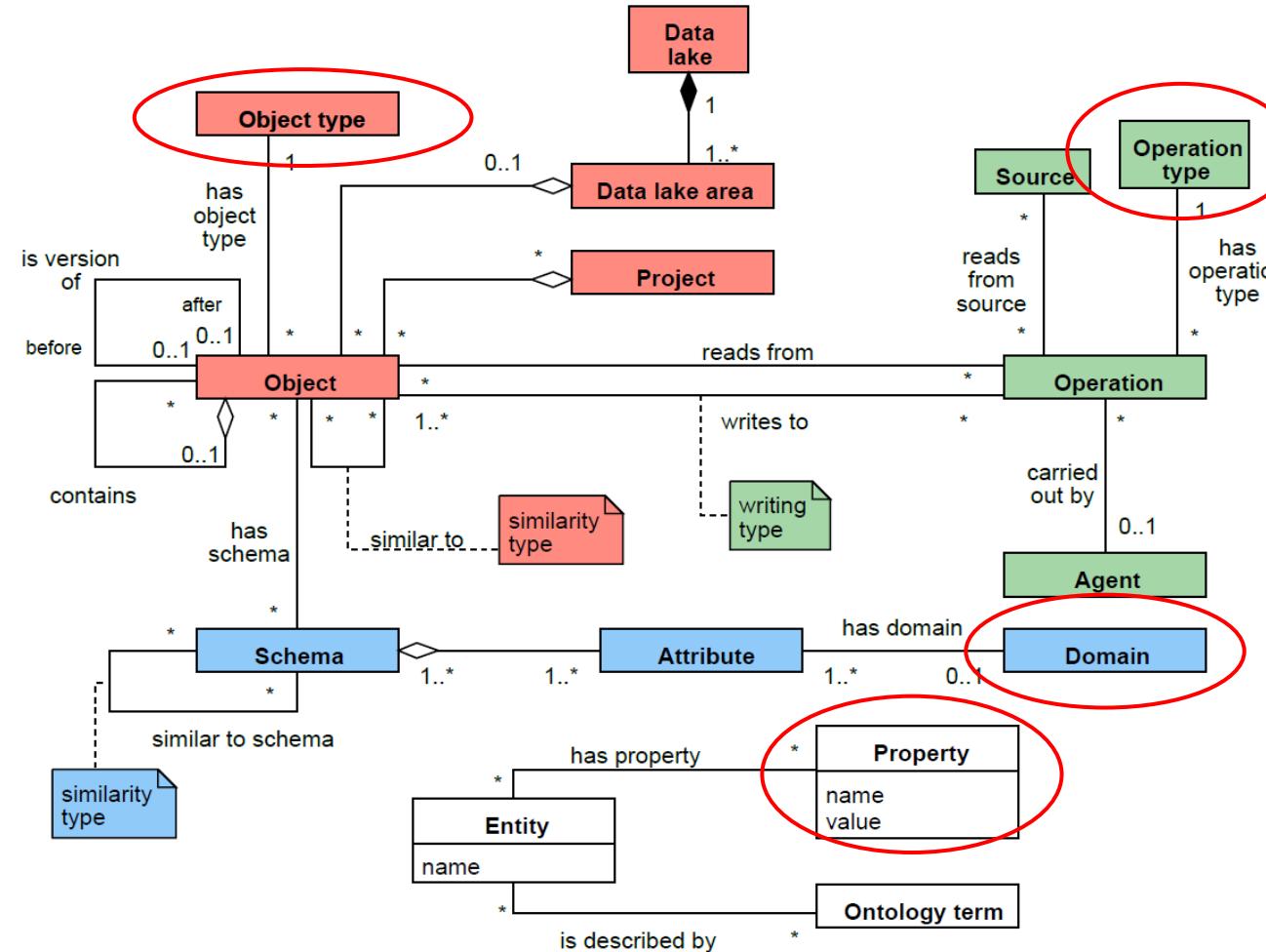
Knowledge representation



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.

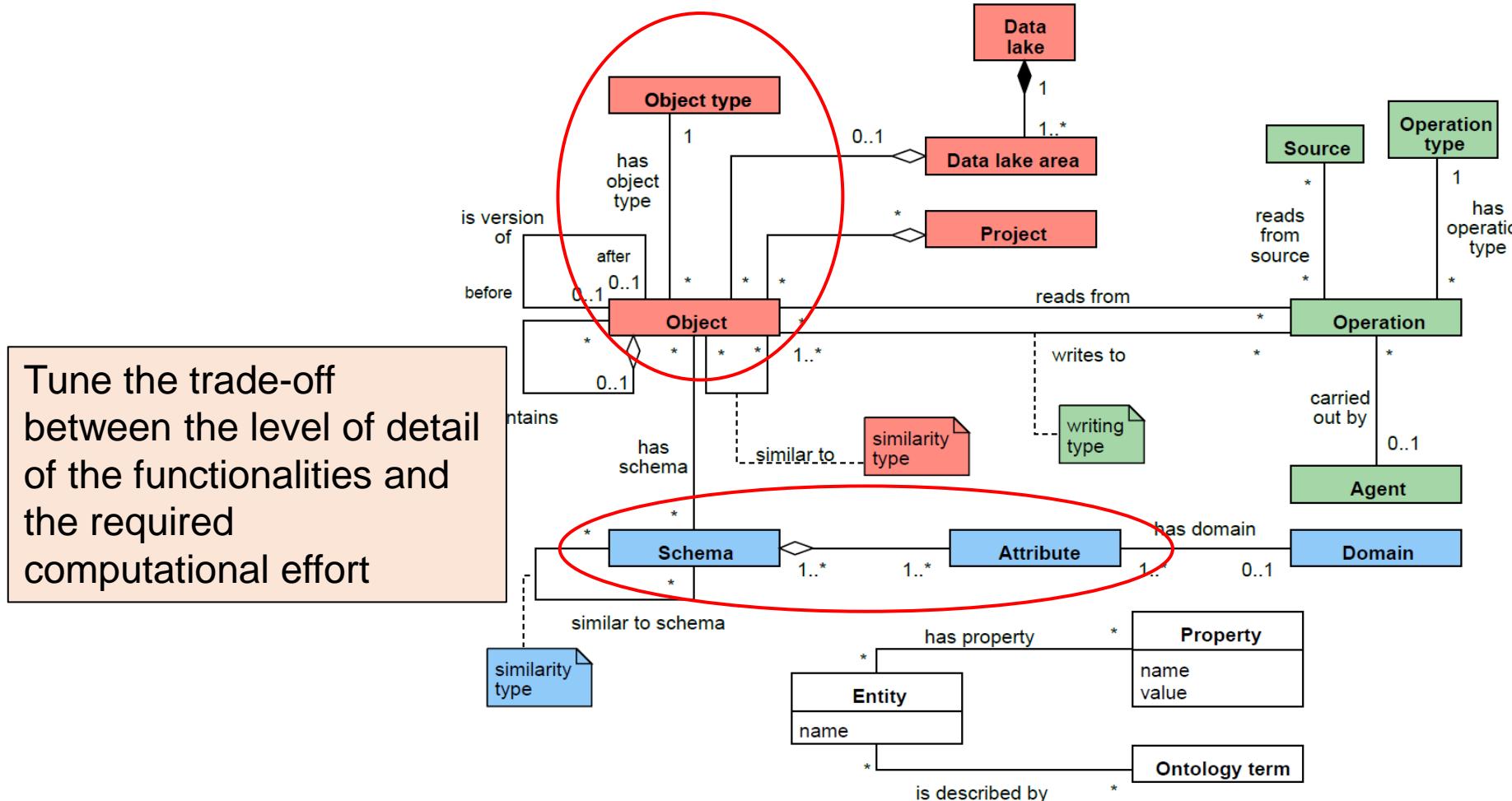
Knowledge representation

Not pre-defined
Domain-independent,
extensible



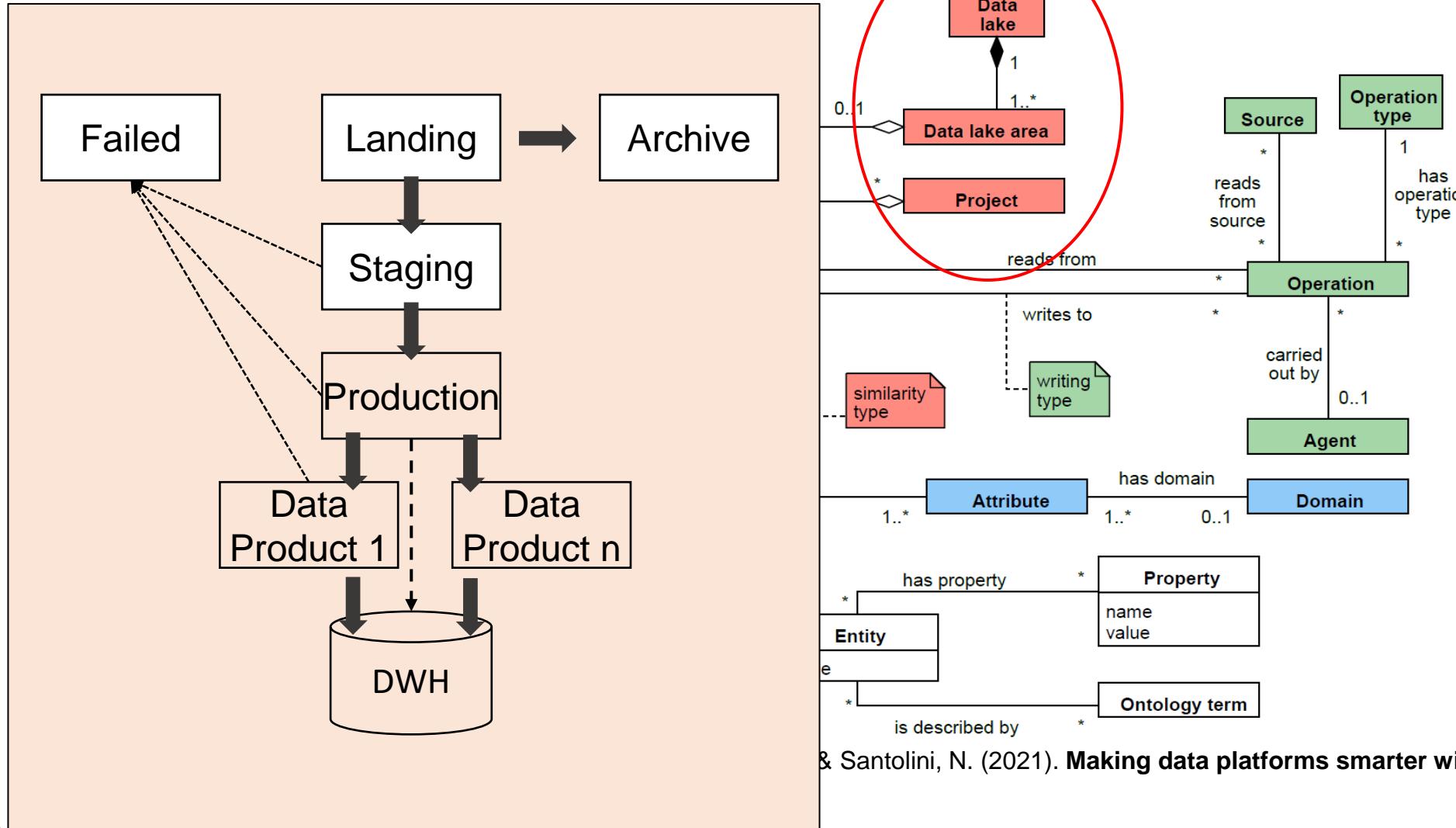
Francia, M., Gallinucci, E., Golfarelli, M., Leoni, A. G., Rizzi, S., & Santolini, N. (2021). Making data platforms smarter with MOSES. *Future Generation Computer Systems*, 125, 299-313.

Knowledge representation

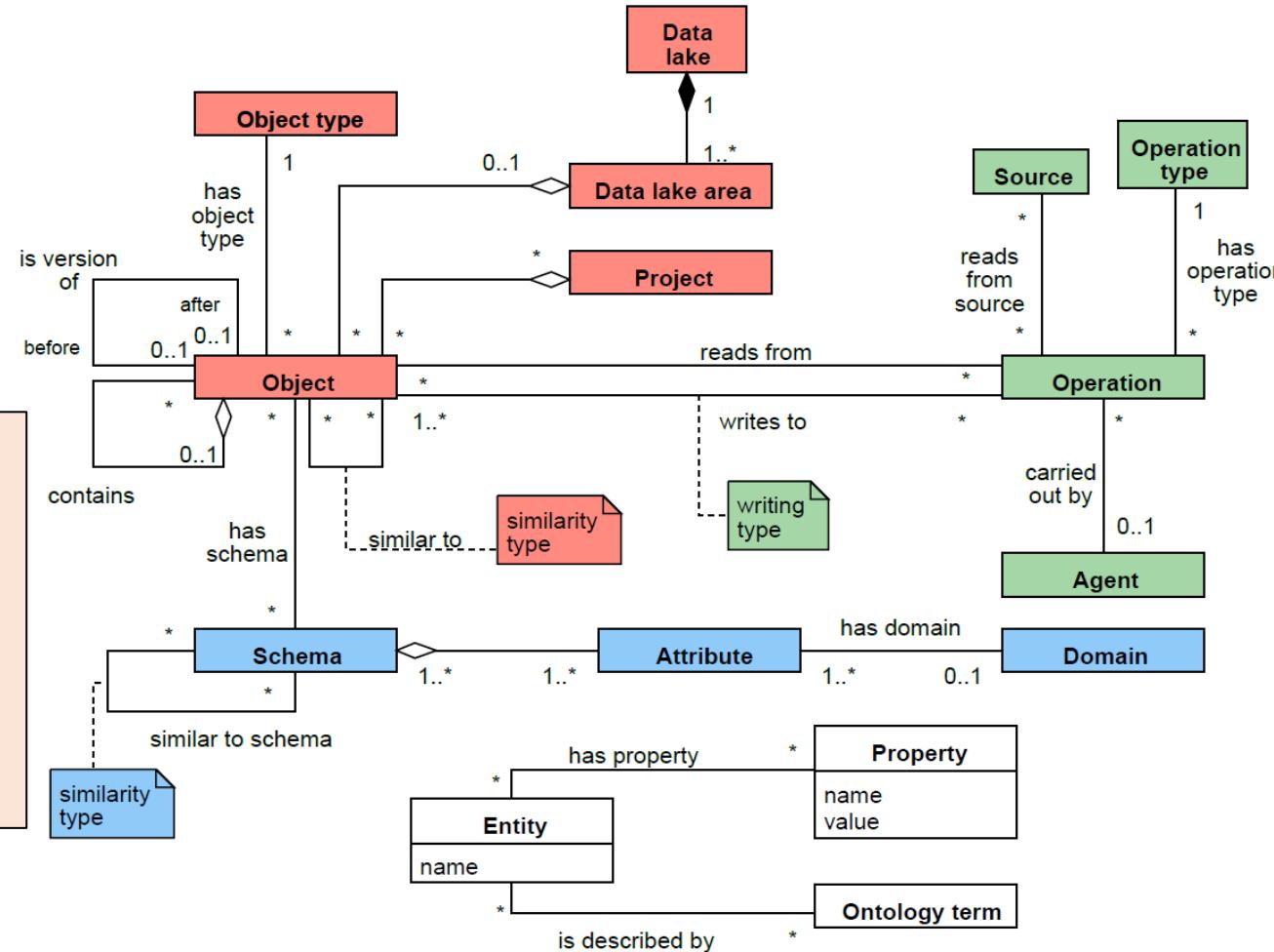


Francia, M., Gallinucci, E., Golfarelli, M., Leoni, A. G., Rizzi, S., & Santolini, N. (2021). **Making data platforms smarter with MOSES**. *Future Generation Computer Systems*, 125, 299-313.

Knowledge representation



Knowledge representation



Functionalities

- ✓ Semantic enrichment
 - ✗ 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.

Knowledge representation

How would you implement the meta-model?

The Property Graph Data Model

Born in the database community

- Meant to be queried and processed
- **THERE IS NO STANDARD!**

Two main constructs: nodes and edges

- Nodes represent entities,
- Edges relate pairs of nodes, and may represent different types of relationships

Nodes and edges might be labeled,

and may have a set of properties represented as attributes (key-value pairs)***

Further assumptions:

- Edges are directed,
- Multi-graphs are allowed

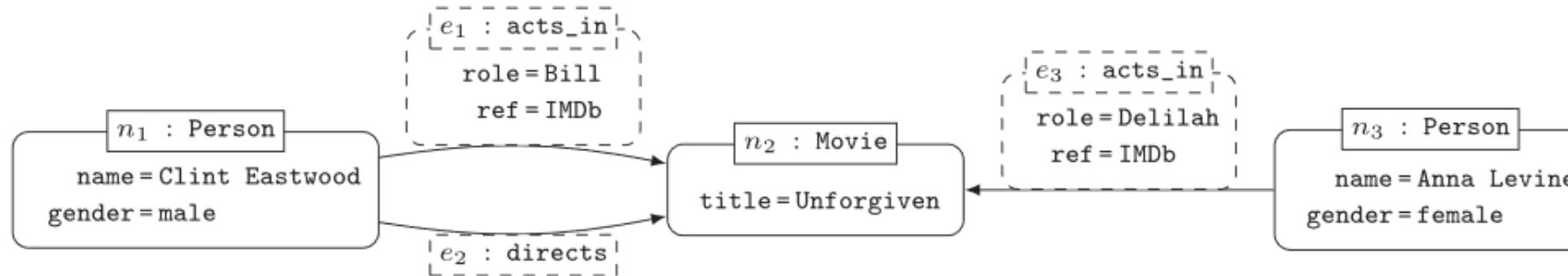
*** Note: in some definitions (the least) edges are not allowed to have attributes

Formal Definition

Definition 2.3 (Property graph). A property graph G is a tuple $(V, E, \rho, \lambda, \sigma)$, where:

- (1) V is a finite set of *vertices* (or *nodes*).
- (2) E is a finite set of *edges* such that V and E have no elements in common.
- (3) $\rho : E \rightarrow (V \times V)$ is a total function. Intuitively, $\rho(e) = (v_1, v_2)$ indicates that e is a directed edge *from* node v_1 *to* node v_2 in G .
- (4) $\lambda : (V \cup E) \rightarrow Lab$ is a total function with *Lab* a set of labels. Intuitively, if $v \in V$ (respectively, $e \in E$) and $\rho(v) = \ell$ (respectively, $\rho(e) = \ell$), then ℓ is the label of node v (respectively, edge e) in G .
- (5) $\sigma : (V \cup E) \times Prop \rightarrow Val$ is a partial function with *Prop* a finite set of properties and *Val* a set of values. Intuitively, if $v \in V$ (respectively, $e \in E$), $p \in Prop$ and $\sigma(v, p) = s$ (respectively, $\sigma(e, p) = s$), then s is the value of property p for node v (respectively, edge e) in the property graph G .

Example of Property Graph



$$V = \{n_1, n_2, n_3\}$$

$$E = \{e_1, e_2, e_3\}$$

$$\rho(e_3) = (n_3, n_2)$$

$$\begin{array}{l} \wedge(n_3) = \text{Person} \\ \lambda(e_2) = \text{directs} \end{array}$$

$$\begin{array}{l} \wedge(e_1) = \text{acts_in} \\ \lambda(e_3) = \text{acts_in} \end{array}$$

$$\begin{aligned} \sigma(n_1, \text{gender}) &= \text{male} \\ \sigma(n_2, \text{title}) &= \text{Unforgiven} \\ \sigma(n_3, \text{name}) &= \text{Anna Levine} \\ \sigma(n_3, \text{gender}) &= \text{female} \\ \sigma(e_1, \text{role}) &= \text{Bill} \\ \sigma(e_1, \text{ref}) &= \text{IMDb} \\ \sigma(e_3, \text{role}) &= \text{Delilah} \\ \sigma(e_3, \text{ref}) &= \text{IMDb} \end{aligned}$$

Traversal Navigation

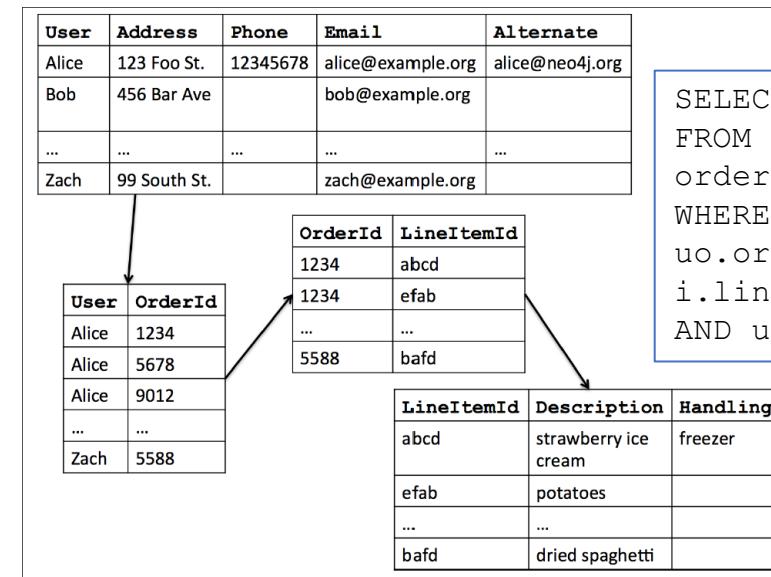
We define the graph traversal pattern as: “the ability to rapidly traverse structures to an arbitrary depth (e.g., tree structures, cyclic structures) and with an arbitrary path description (e.g. friends that work together, roads below a certain congestion threshold)” [Marko Rodriguez]

Totally opposite to set theory (on which relational databases are based on)

- Sets of elements are operated by means of the relational algebra

Traversing Data in a RDBMS

In the relational theory, it is equivalent to joining data (schema level) and select data (based on a value)



```
SELECT *
FROM user u, user_order uo,
orders o, items i
WHERE u.user = uo.user AND
uo.orderId = o.orderId AND
i.lineItemId = i.lineItemId
AND u.user = 'Alice'
```

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

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-processing 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.

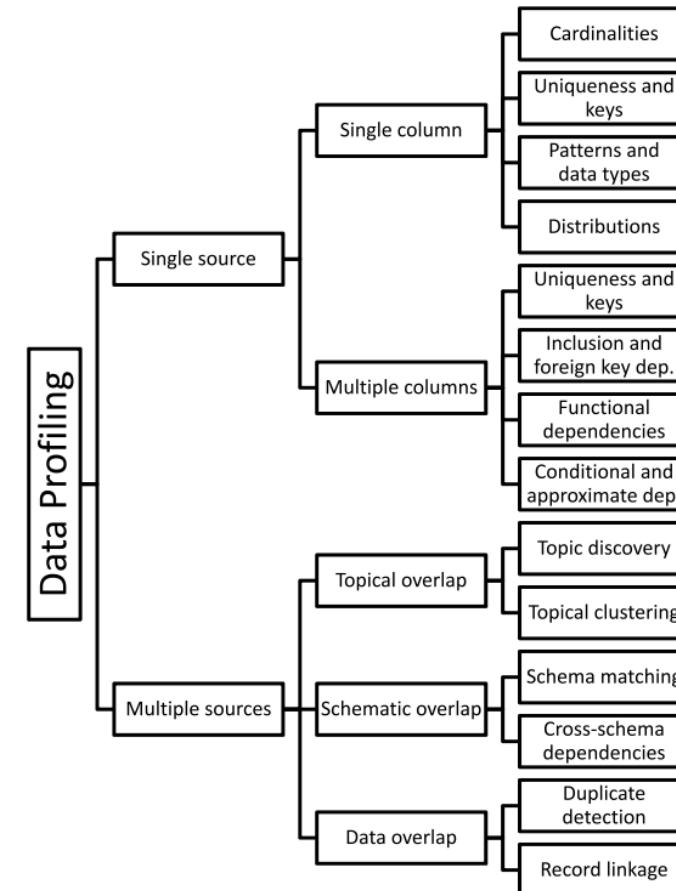
Managing data platforms

- Data provenance
- Compression
- Data profiling
- Entity resolution
- Data versioning
- ...

Data profiling

Data profiling

- A broad range of methods to efficiently analyze a given data set
- E.g., in a **relational** scenario, **tables** of a relational database are **scanned** to derive **metadata**, such as data types **and value patterns**, completeness and uniqueness of columns, **keys and foreign keys**, and occasionally **functional dependencies** and association rules



Naumann, Felix. "Data profiling revisited." *ACM SIGMOD Record* 42.4 (2014): 40-49.

Data profiling

Use cases

- **Query optimization**
 - Performed by DBMS to support query optimization with statistics about tables and columns
 - Profiling results can be used to estimate the selectivity of operators and the cost of a query plan
- **Data cleansing** (typical use case is profiling data)
 - Prepare a cleansing process by revealing errors (e.g., in formatting), missing values or outliers
- **Data integration and analytics**

Challenges?

Naumann, Felix. "Data profiling revisited." *ACM SIGMOD Record* 42.4 (2014): 40-49.

Data profiling

Challenges

- The results of data profiling are **computationally complex** to discover
 - E.g., discovering keys/dependencies usually involves some sorting step for each considered column
- Verification of **complex constraints on column combinations** in a database
 - What is the complexity of this task?

a	b	c	d
1	1	2	2
1	2	1	4

Complexity

- Given a table with columns $C = \{ a, b, c, d \}$
- To extract the (distinct) cardinality of each column, I will consider $|C|$ columns
(a), (b), (c), (d)
- To extract the correlations between pairs of columns, I will consider $\binom{|C|}{2}$ groups
(a, b), (a, c), (a, d), (b, c), (c, d), (c, d)
- Extracting the relationships among all possible groups of columns generalizes to $\sum_{n=1}^{|C|} \binom{|C|}{n} = 2^{|C|} - 1$ groups

Naumann, Felix. "Data profiling revisited." *ACM SIGMOD Record* 42.4 (2014): 40-49.

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

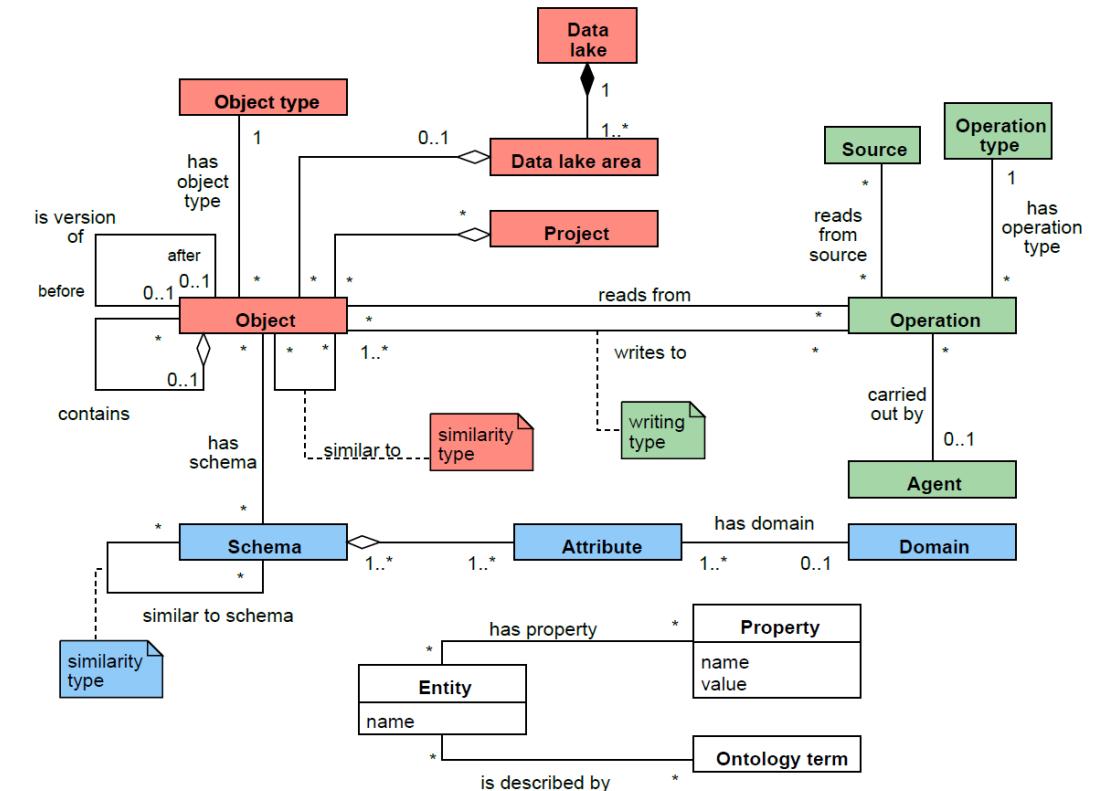
Two levels of querying

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

Object profiling and search

Basic search

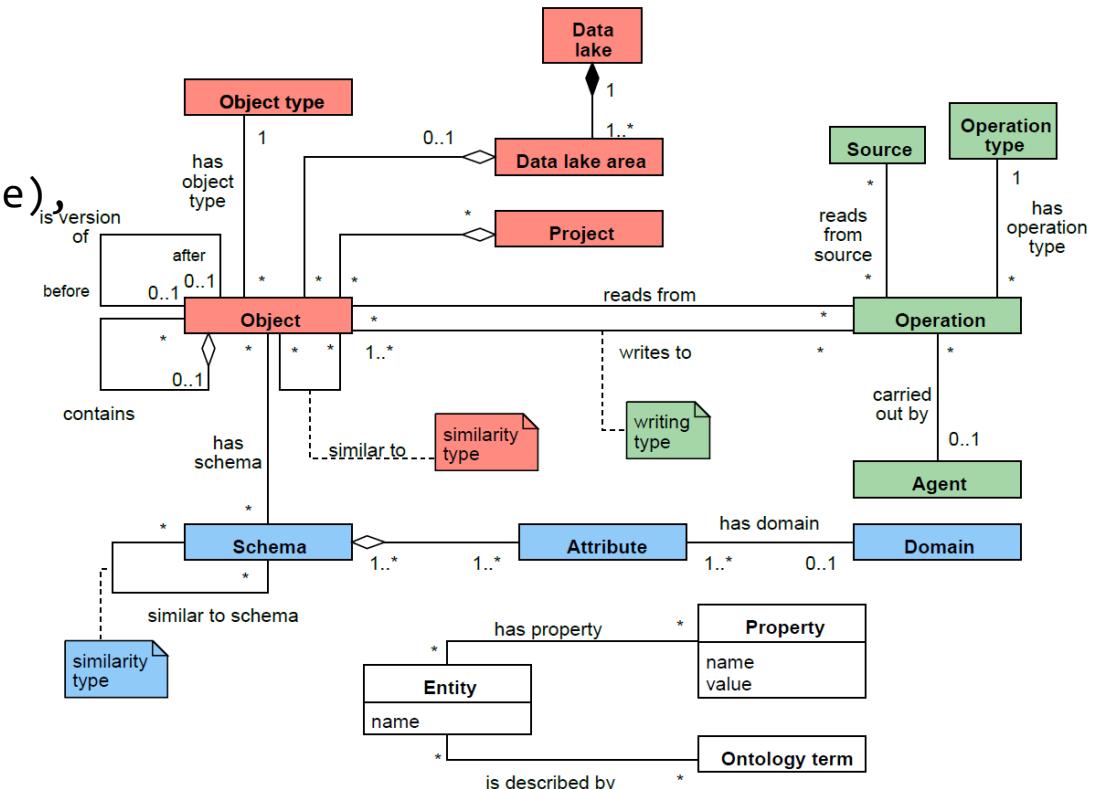
- MATCH (o:Object)-[]-(:Project {name:"ABC"})
RETURN o
 - Return all objects of a given project
- MATCH (o:Object)-[]-(d:DataLakeArea)
WHERE d.name = "Landing"
AND o.name LIKE "2021_%"
AND o.size < 100.000
RETURN o
 - Return small objects with a given name pattern in the landing area



Object profiling and search

Schema-driven search

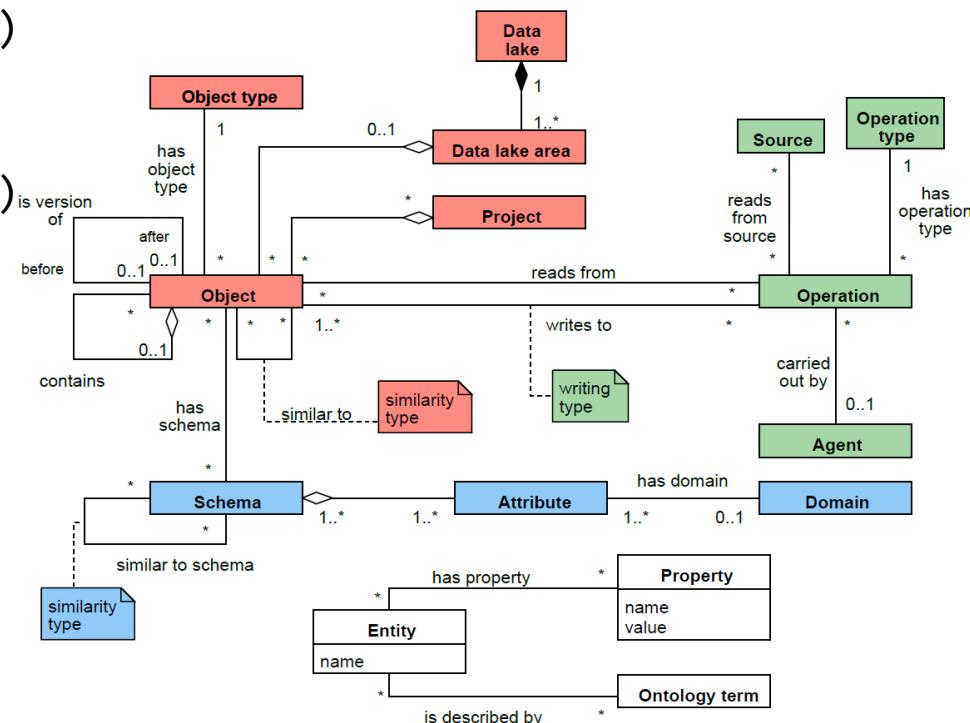
- MATCH (o:Object)-[]-(:Schema)-[]-(a:Attribute),
 (a)-[]-(:Domain {name: "FiscalCode"})
 RETURN o
 - Return objects that contain information referring to a given Domain



Object profiling and search

Provenance-driven search

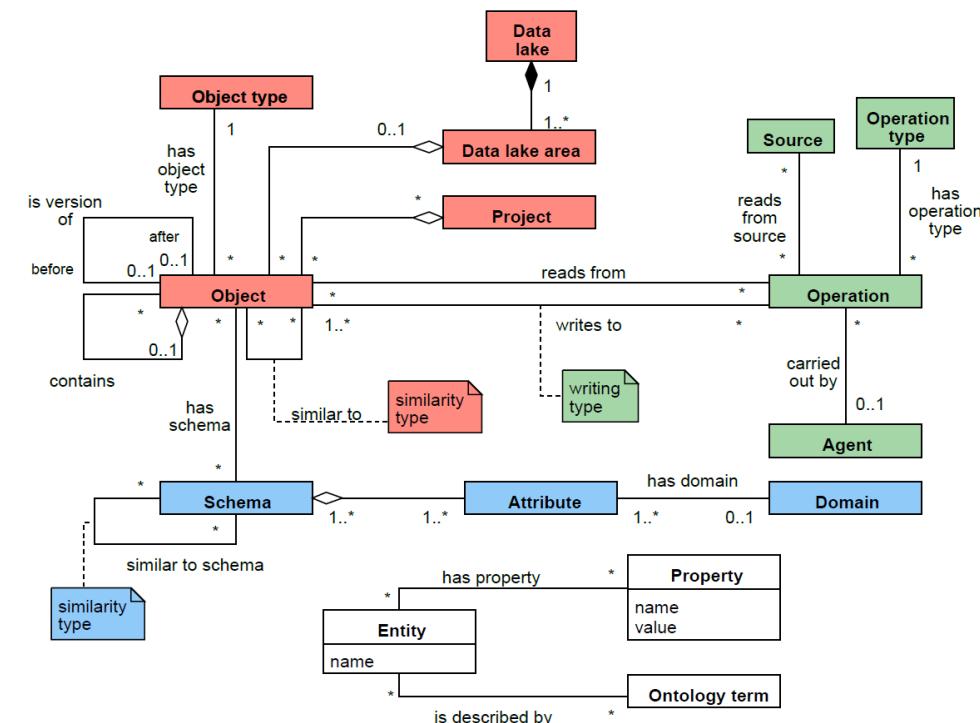
- MATCH (obj1:Object)-[:readsFrom]-(o:Operation)-[:writesTo]-(obj2:Object)
CREATE (obj1)-[:ancestorOf]->(obj2)
- MATCH (:Object {id:123})-[:ancestorOf*]-(obj:Object)
RETURN obj
 - Discover objects obtained from a given ancestor
- MATCH (obj:Object)-[:ancestorOf*]-(:Object {id:123})
RETURN obj
 - Discover 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

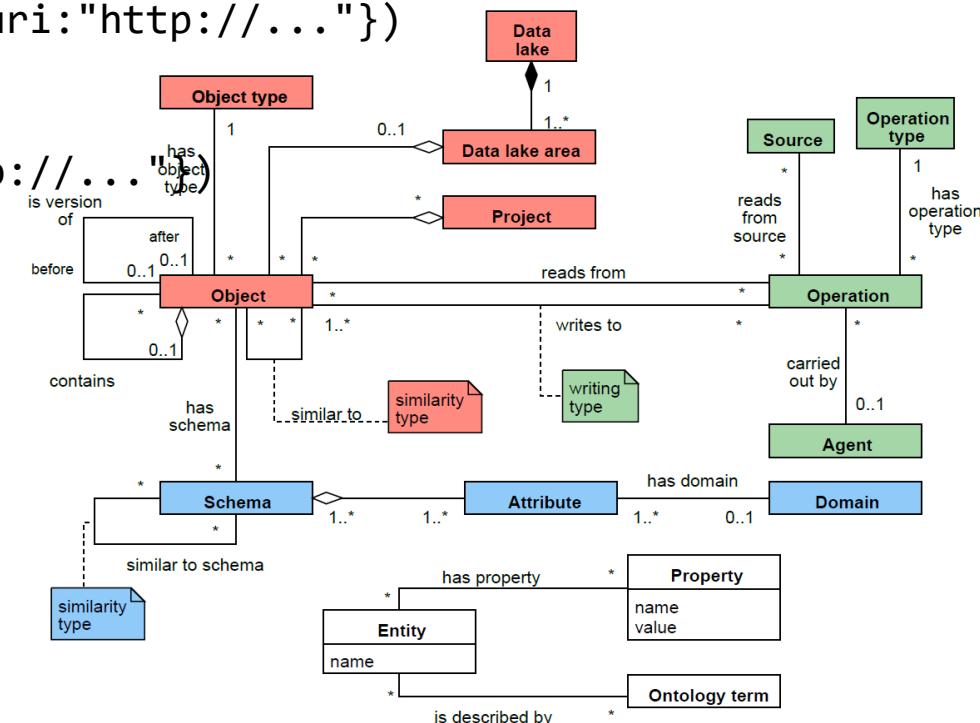
- MATCH (:Object {id:123})-[r:similarTo]-(o:Object)
WHERE r.similarityType="affinity"
RETURN o
 - Discover datasets to be merged in a certain query
- MATCH (:Object {id:123})-[r:similarTo]-(o:Object)
WHERE r.similarityType="joinability"
RETURN o
 - Discover datasets to be joined in a certain query
- 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

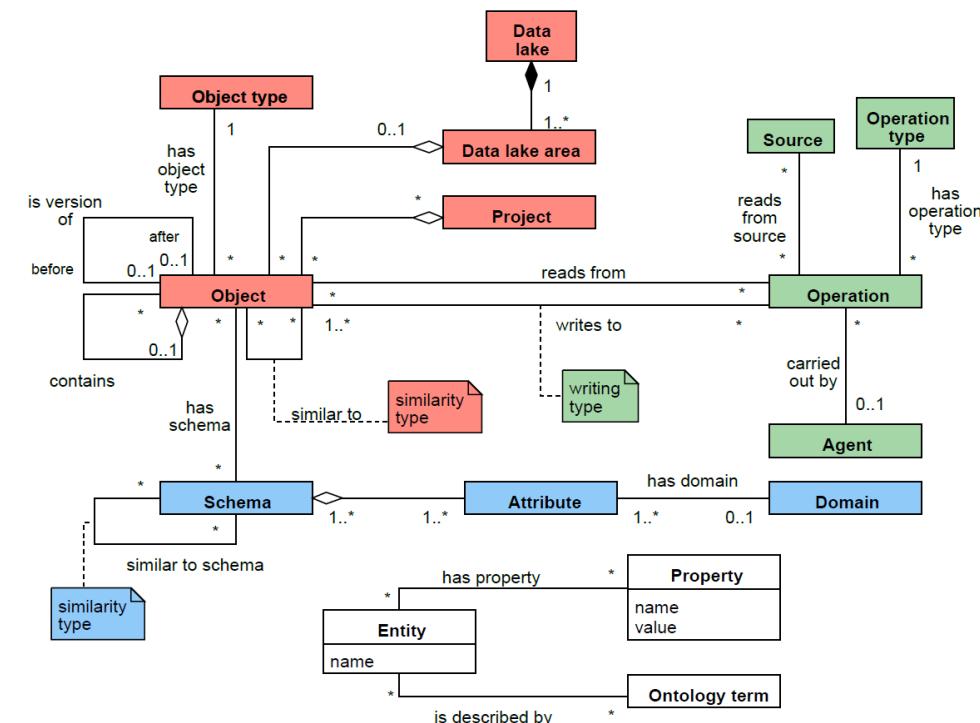
Profiling

- ```

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. Diestelkämper, H. Ben Lahmar, **A survey on provenance: What for? What form? What from?**, *VLDB J.* 26 (6) (2017) 881–906.

# Data provenance

Provenance (also referred to as lineage, pedigree, parentage, genealogy)

- The description of the origins of data and the process by which it arrived at the database
- Not only data products (e.g., tables, files), but also the processes that created them

## Use cases

- Business domain. *Users traditionally work with an organized data schema, where the structure and semantics of the data in use is shared across the corporation or even B2B. Yet, a large proportion of businesses deal with bad quality data. Sources of bad data need to be identified and corrected to avoid costly errors in business forecasting.*
- Scientific/research domain. *Data used in the scientific field can be ad hoc and driven by individual researchers or small communities. The scientific field is moving towards more collaborative research and organizational boundaries are disappearing. Sharing data and metadata across organizations is essential, leading to a convergence on common schemes to ensure compatibility. Issues of trust, quality, and copyright of data are significant when using third-party data in such a loosely connected network.*

Simmhan, Yogesh L., Beth Plale, and Dennis Gannon. "A survey of data provenance techniques." *Computer Science Department, Indiana University, Bloomington IN 47405* (2005): 69.

# Data provenance

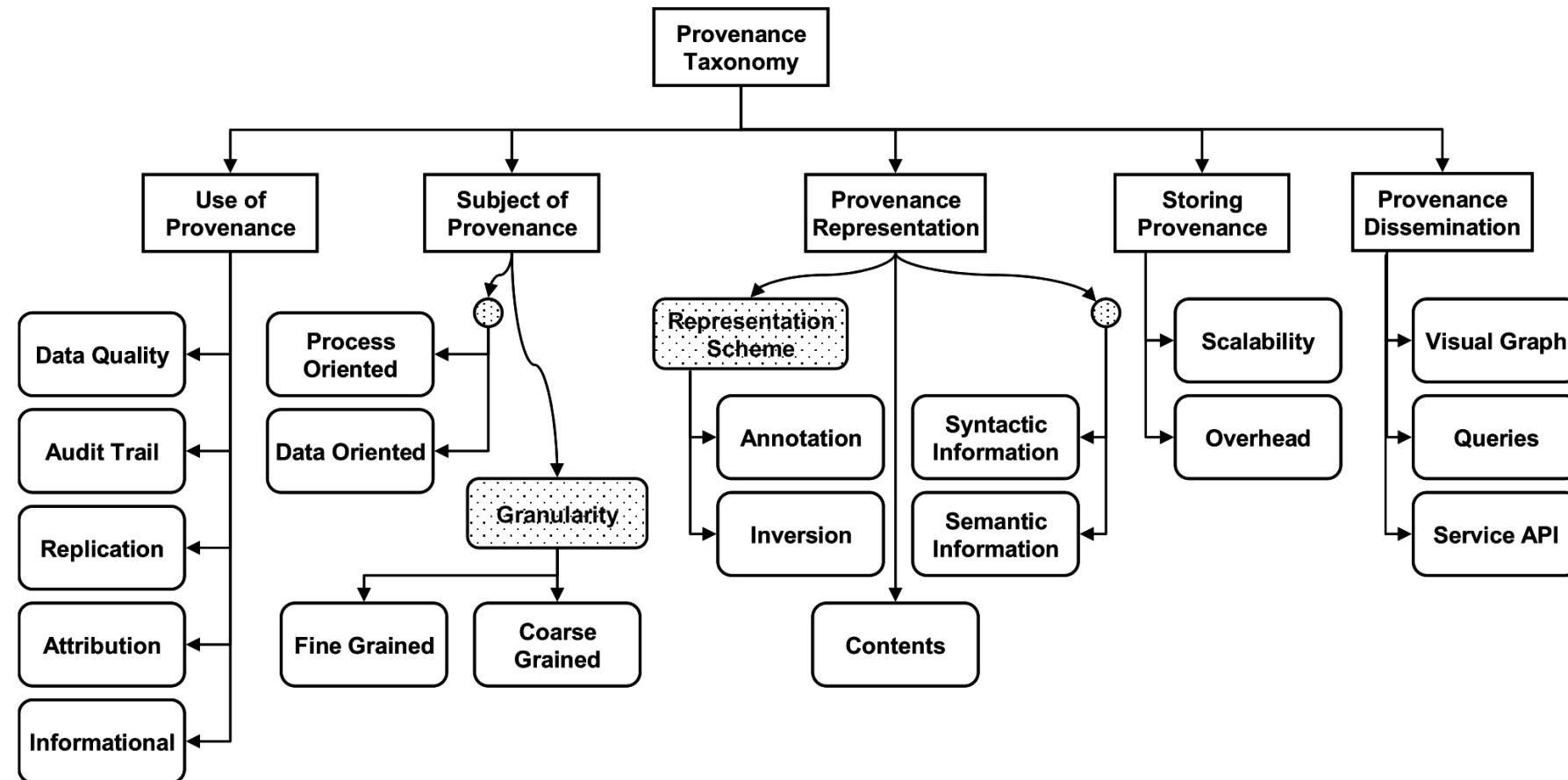
Astronomers are creating an international Virtual Observatory

- A **federation** of all the world significant astronomical **data resources** coupled with **provision of the computational resources** needed to exploit the data scientifically
- Astronomy changed from being an individualistic to a **collective enterprise**
- Telescope time is devoted/allocated to systematic sky surveys and analysis is performed using data from the archives
- Astronomers are **increasingly relying on data that they did not take themselves**
- Raw data bear **many instrumental signatures that must be removed** in the process of generating data products



Mann, Bob. "Some data derivation and provenance issues in astronomy." *Workshop on Data Derivation and Provenance, Chicago*. 2002.  
[https://www.esa.int/Science\\_Exploration/Space\\_Science/Webb/Webb\\_inspects\\_the\\_heart\\_of\\_the\\_Phantom\\_Galaxy](https://www.esa.int/Science_Exploration/Space_Science/Webb/Webb_inspects_the_heart_of_the_Phantom_Galaxy) (accessed 2022-08-01)

# Data provenance



Simmhan, Yogesh L., Beth Plale, and Dennis Gannon. "A survey of data provenance techniques." *Computer Science Department, Indiana University, Bloomington IN 47405* (2005): 69.

# Data provenance

## Granularity

- **Fine-grained** (instance level): tracking data items (e.g., a tuple in a dataset) transformations
- **Coarse-grained** (schema-level): tracking dataset transformations

## Queries

- **Where** provenance: given some output, which inputs did the output come from?
- **How** provenance: given some output, how were the inputs manipulated?
- **Why** provenance: given some output, why was data generated?
  - E.g., in the form of a proof tree that locates source data items contributing to its creation

Simmhan, Yogesh L., Beth Plale, and Dennis Gannon. "A survey of data provenance techniques." *Computer Science Department, Indiana University, Bloomington IN 47405* (2005): 69.

Ikeda, Robert, and Jennifer Widom. *Data lineage: A survey*. Stanford InfoLab, 2009.

# 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

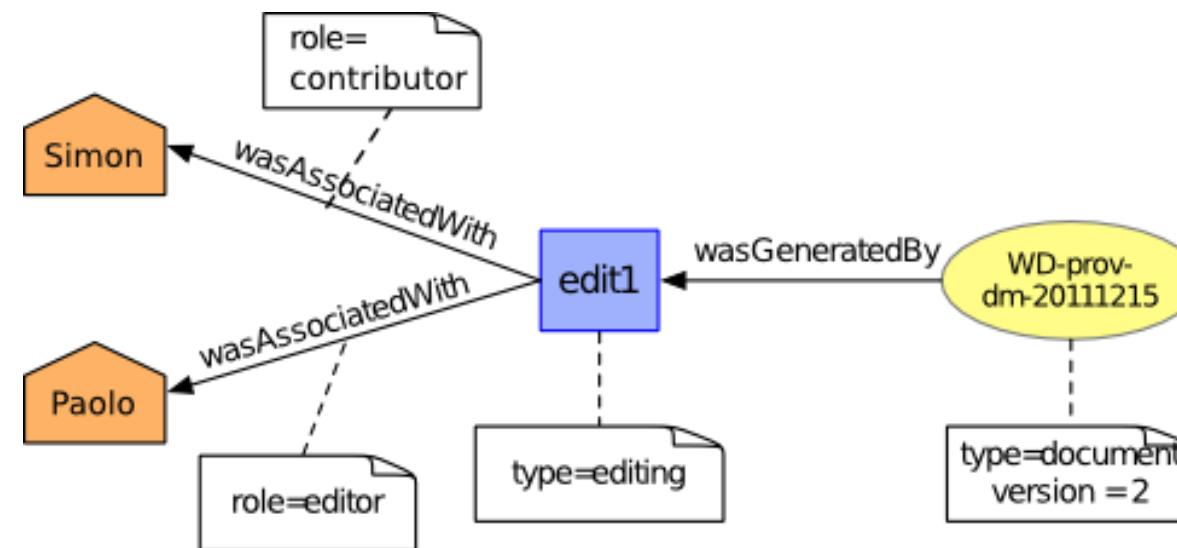
Choosing a granularity is the result of a trade-off between accuracy and computational effort

- 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

# Data provenance

## Data provenance, an example of data management

- Metadata pertaining to the history of a data item
- Pipeline including the origin of objects and operations they are subjected to
- We have a standard: <https://www.w3.org/TR/prov-dm/>



<https://www.w3.org/TR/prov-dm/>

# Data provenance

## Entity

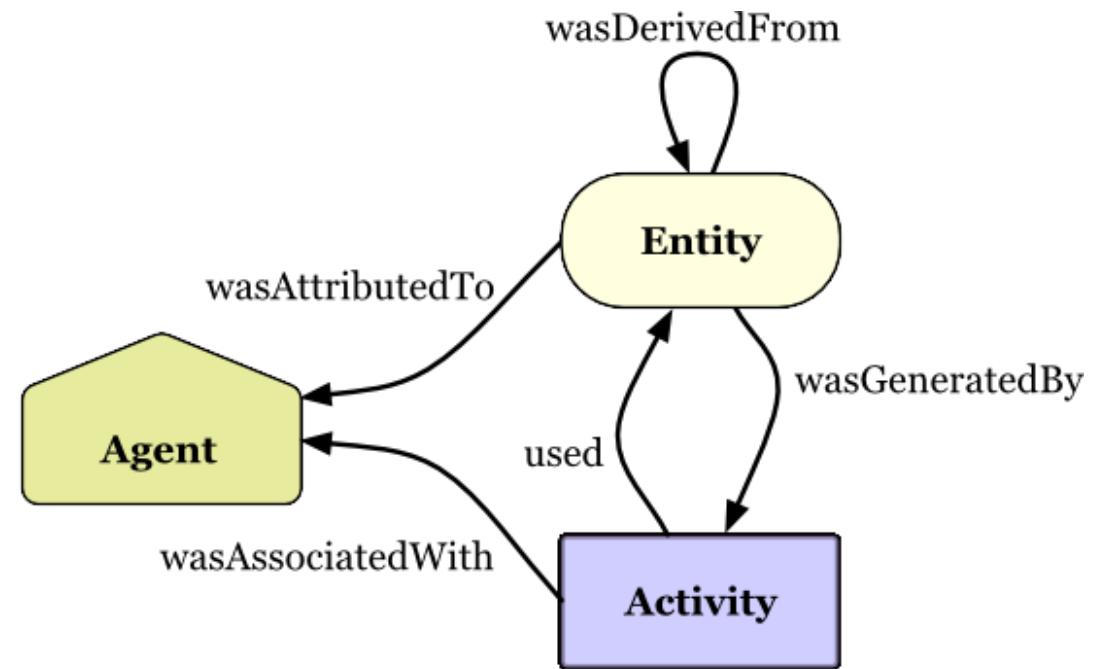
- Physical/conceptual things

## Activity

- Dynamic aspects of the world, such as actions
- How entities come into existence, often making use of previously existing entities

## Agent

- A person, a piece of software
- Takes a role in an activity such that the agent can be assigned some degree of responsibility for the activity taking place

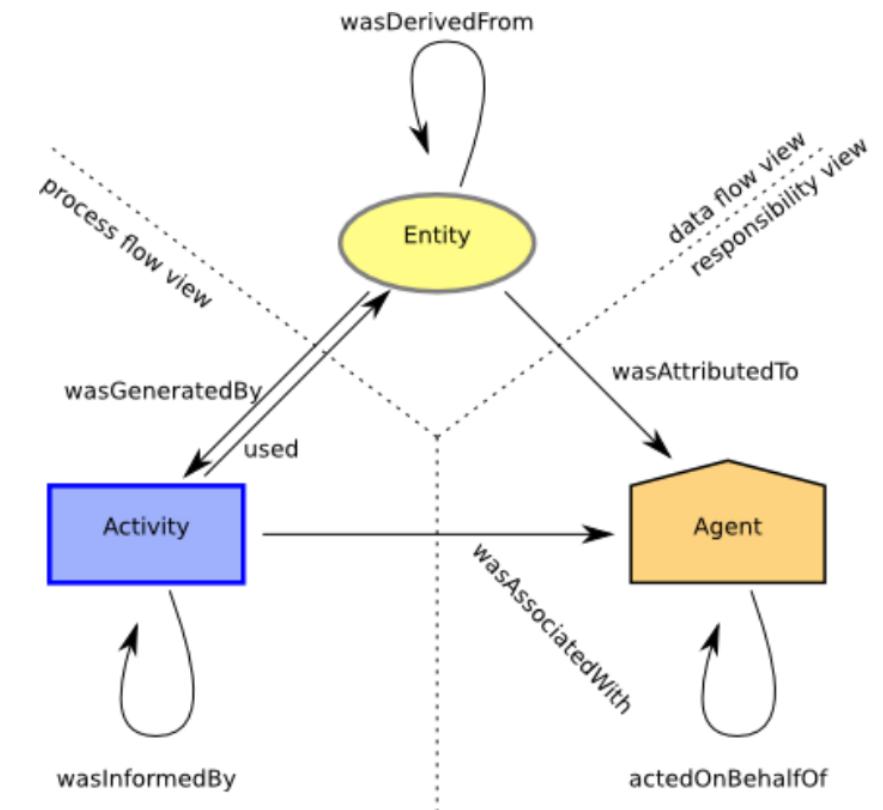


<https://www.w3.org/TR/2013/NOTE-prov-primer-20130430/>

# 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

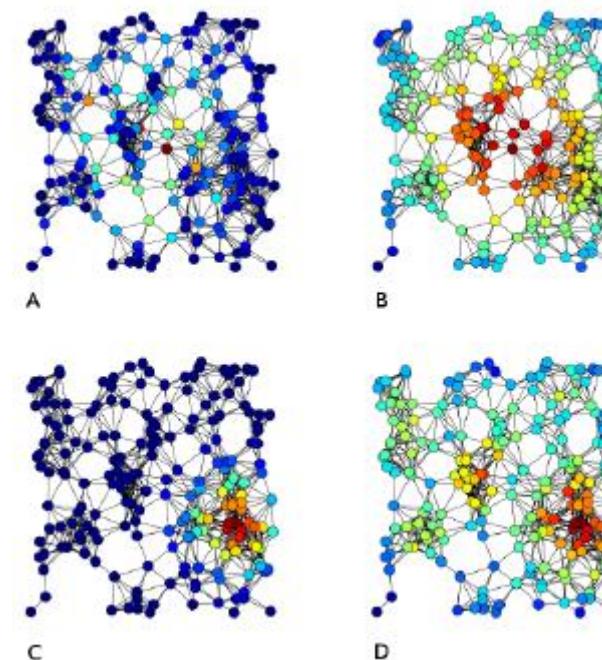
## Provenance functionalities (activated by metadata)

- **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

# Graph DB and Centrality Measures

## Measures of centrality

- **Betweenness centrality (A)**
  - Number of shortest paths between two nodes that pass from a certain node
- **Closeness centrality (B)**
  - Sum of distances to all other nodes.
- **Eigenvector centrality (C)**
  - The score of a node is influenced by score of adjacent nodes (Page rank)
- **Degree centrality (D)**
  - Number of adjacent nodes



# 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

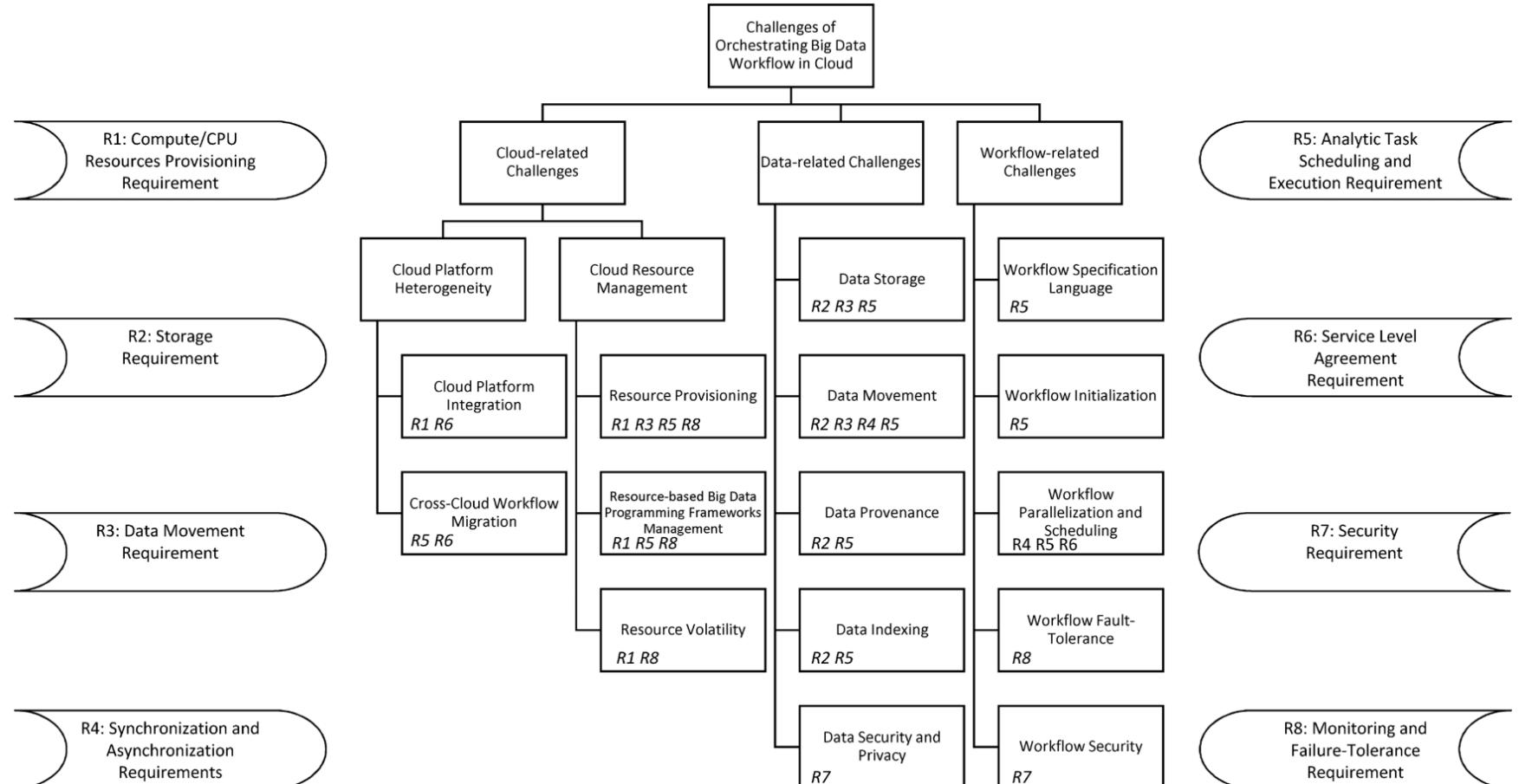
# Orchestration support

## 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)

# 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

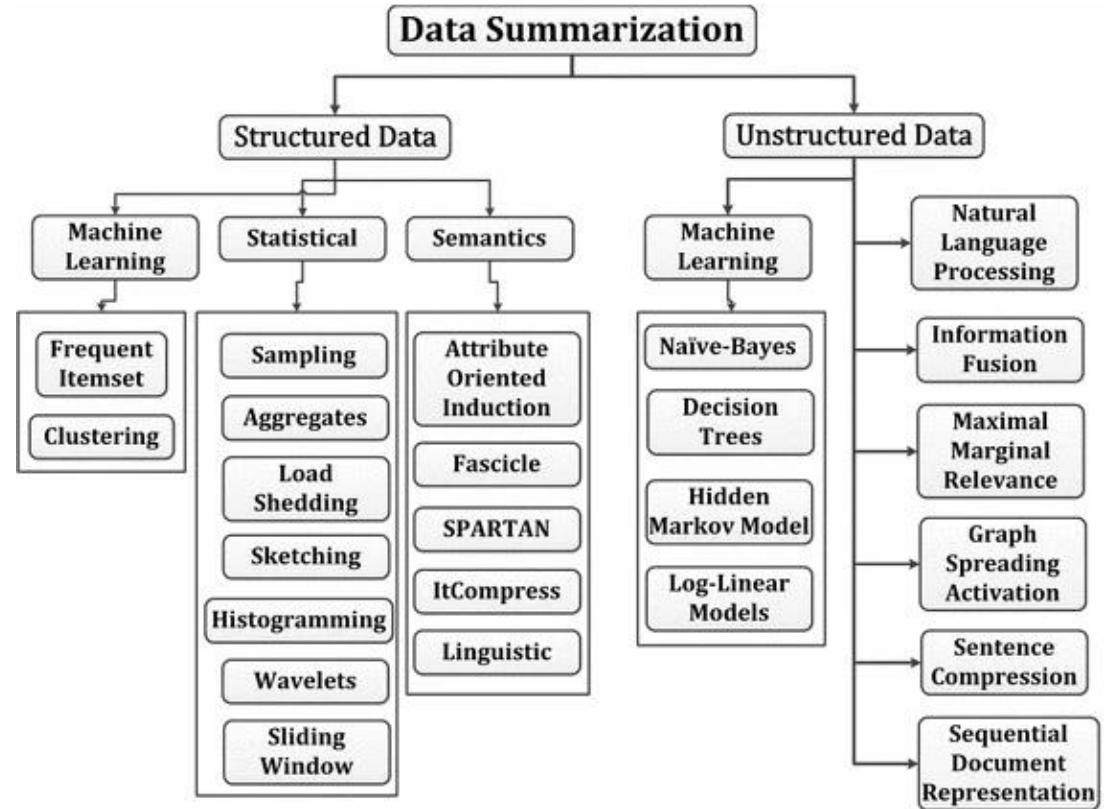
- Workflow-related
  - **Specification Language** (devising a high level, technology-/cloud-independent workflow language)
  - **Initialization** (subdivision into fragments considering dependencies, constraints, etc.)
  - **Parallelization and Scheduling** (with super-workflows defined at application 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.

# Compression

## Summarization / compression

- Present a concise representation of a dataset in a comprehensible and informative manner



Ahmed, Mohiuddin. "Data summarization: a survey." *Knowledge and Information Systems* 58.2 (2019): 249-273.

# Entity resolution

## Entity resolution

- (also known as entity matching, linking)
- Find records that refer to the same entity across different data sources (e.g., data files, books, websites, and databases)

| ID  | Name                     | Telephone    | Address                       | Items Purchased  |
|-----|--------------------------|--------------|-------------------------------|------------------|
| 233 | Angelica J. Jordan       | 334-555-0178 | 111 Spring Ln, Greenville, AL | 5556, 7611       |
| 452 | Angie Jordan             | 202-555-5477 | 45 Krakow St, Washington, DC  | 2297             |
| 699 | Andrew Jordan            | 334-555-0178 | 111 Spring Ln, Greenville, AL | 1185, 2299, 3720 |
| 720 | Angie Jrodon             |              |                               | 5556             |
| 821 | Angelica Jeffries Jordan | 202-555-5477 | 397 Hope Blvd, Greenville, AL | 7611             |

Papadakis, George, et al. "Blocking and filtering techniques for entity resolution: A survey." *ACM Computing Surveys (CSUR)* 53.2 (2020): 1-42.

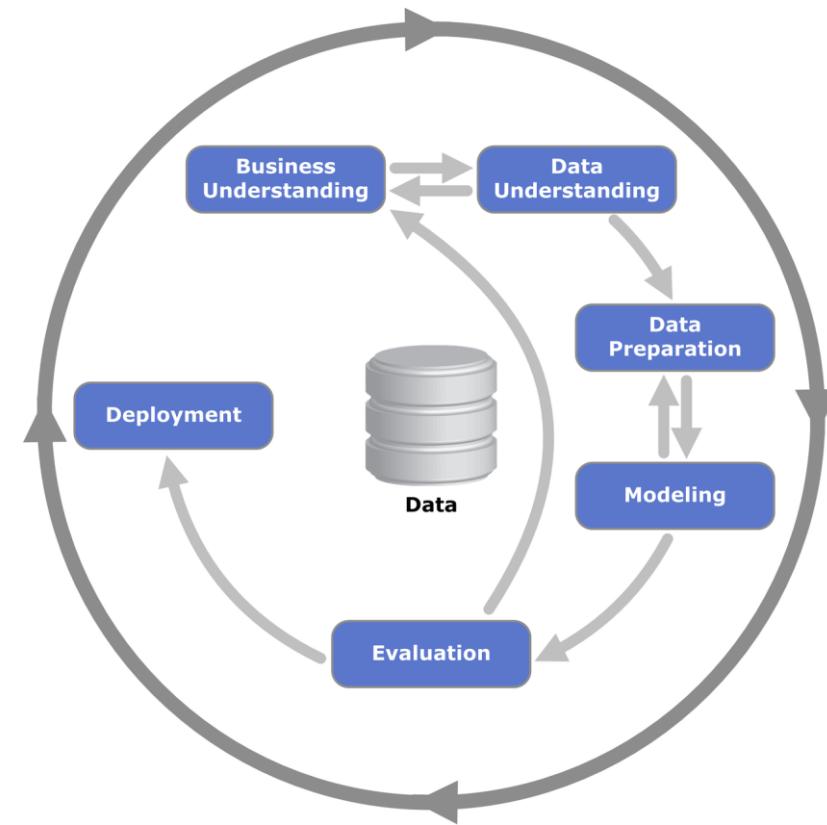
# Data versioning

## Version control

- A class of systems responsible for managing changes to computer programs, documents, or data collections
- Changes are identified by a number/letter code, termed the revision/version number

However, data pipelines are not only about code but also about

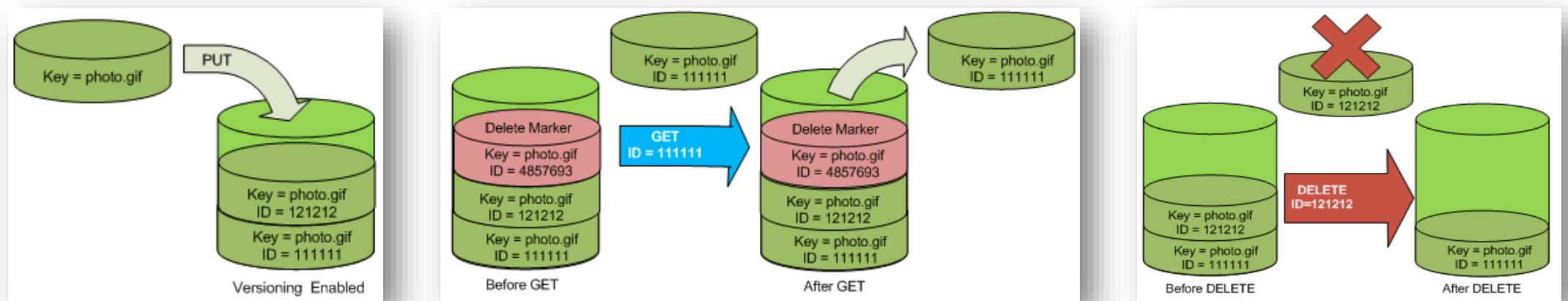
- Model Version control
- Data Version Control
- Model Parameter Tracking
- Model Performance Comparison



# Data versioning

Support CRUD (Create, Read, Update, Delete) operations with versions

E.g., on AWS (PUT, GET, DELETE), what about update?



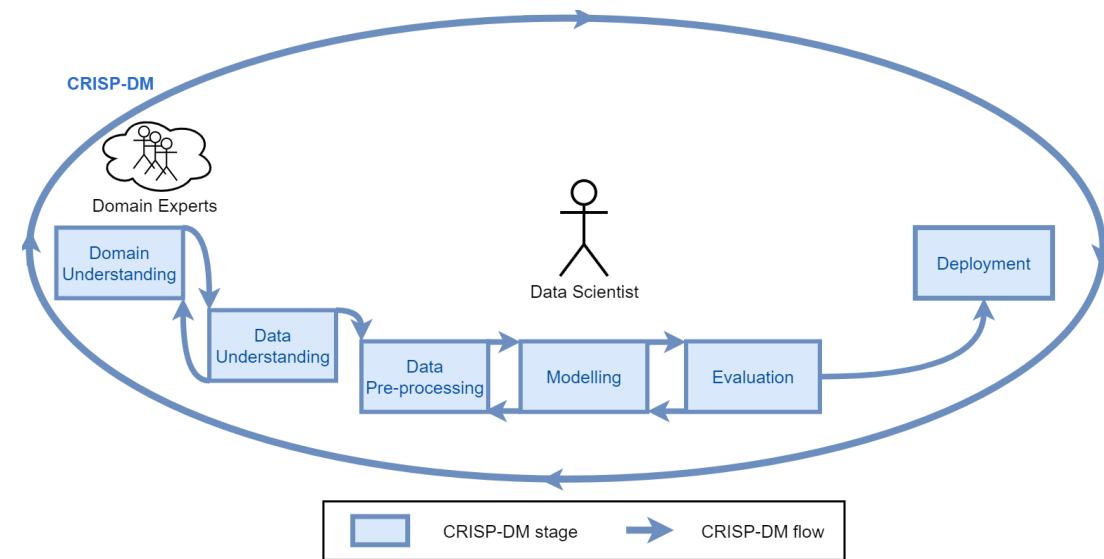
<https://docs.aws.amazon.com/AmazonS3/latest/userguide/versioning-workflows.html> (accessed 2022-08-01)

# Tuning Data Pipelines

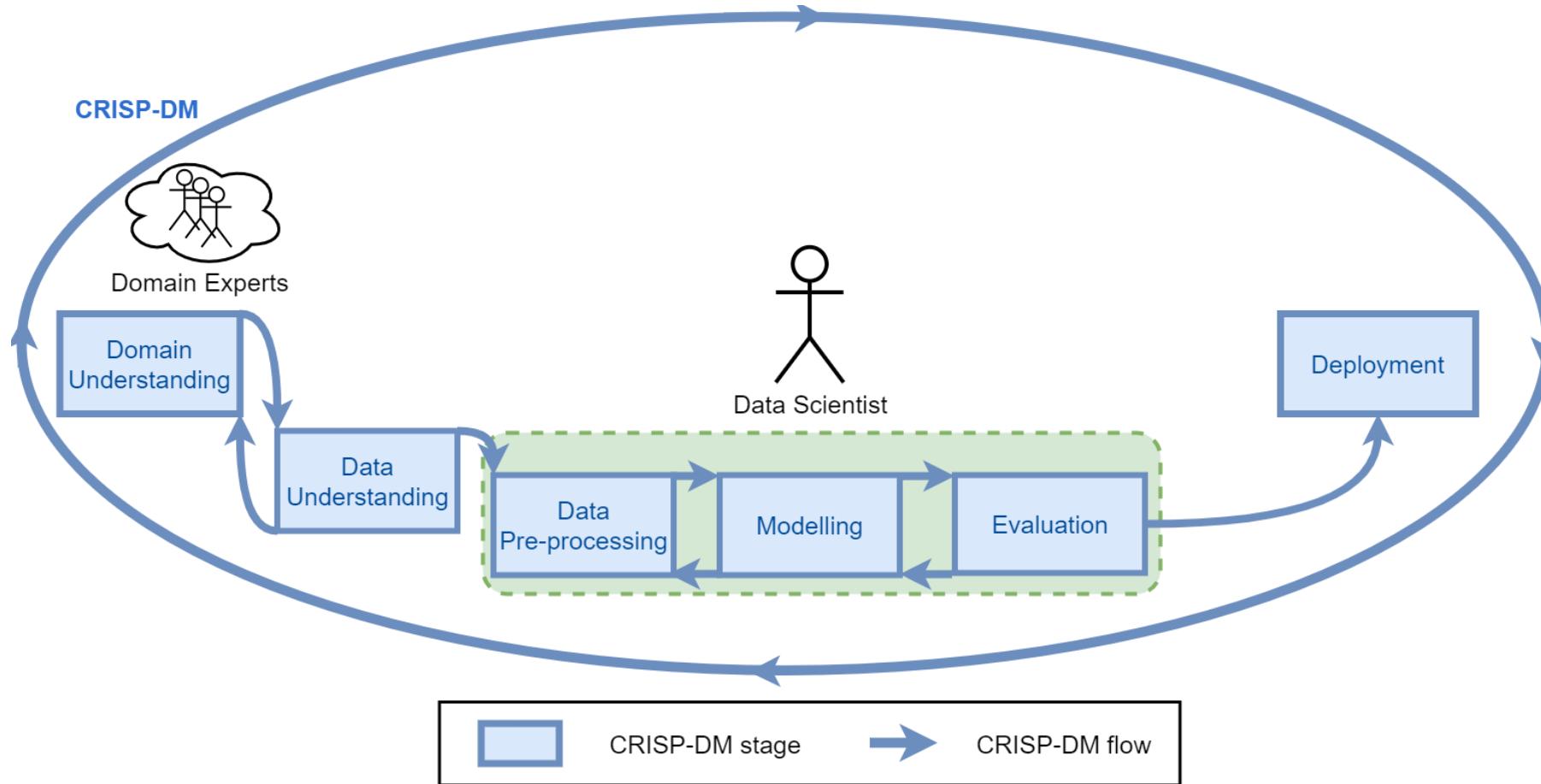
# CRISP-DM

The **CRoss Industry Standard Process for Data Mining (CRISP-DM)** is a process model that serves as the base for a data science process. It has six sequential phases:

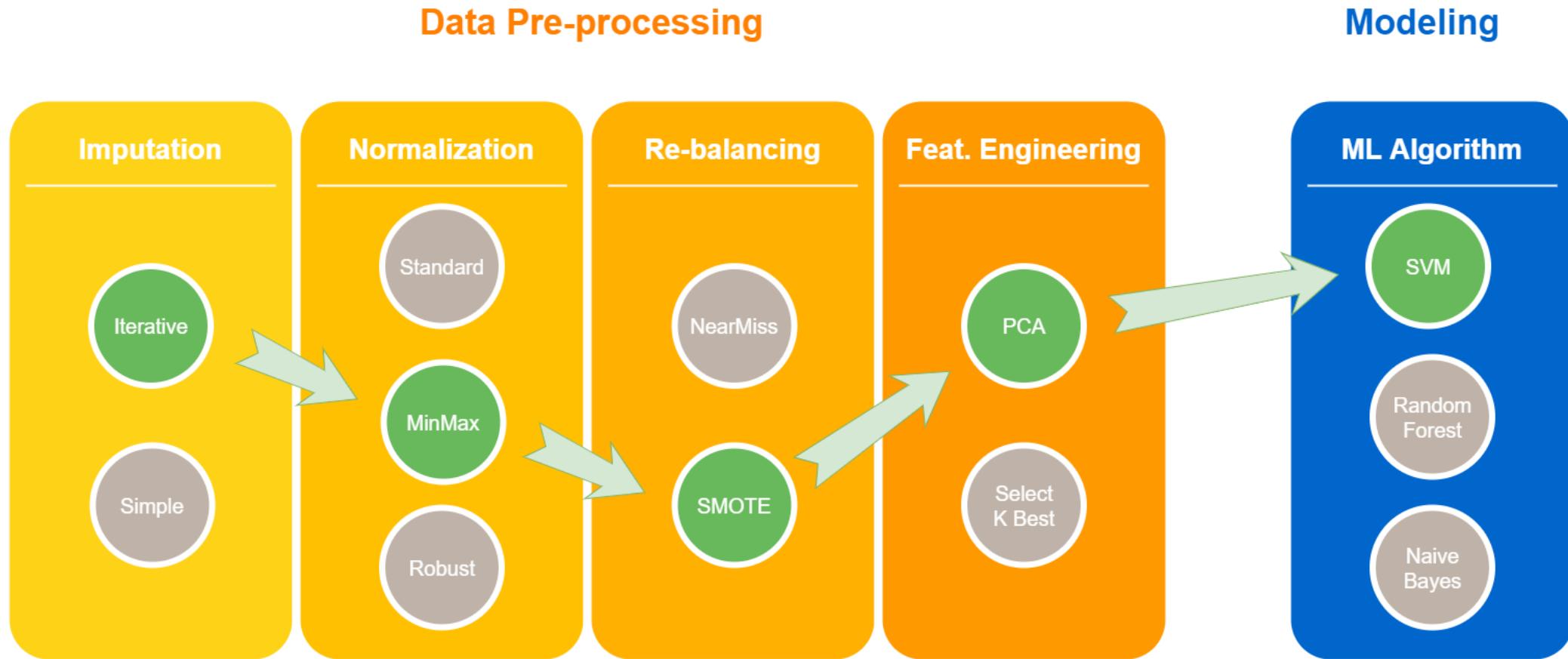
1. Business understanding – What does the business need?
2. Data understanding – What data do we have / need? Is it clean?
3. Data preparation – How do we organize the data for modeling?
4. Modeling – What modeling techniques should we apply?
5. Evaluation – Which model best meets the business objectives?
6. Deployment – How do stakeholders access the results?



# Pipelines for ML tasks



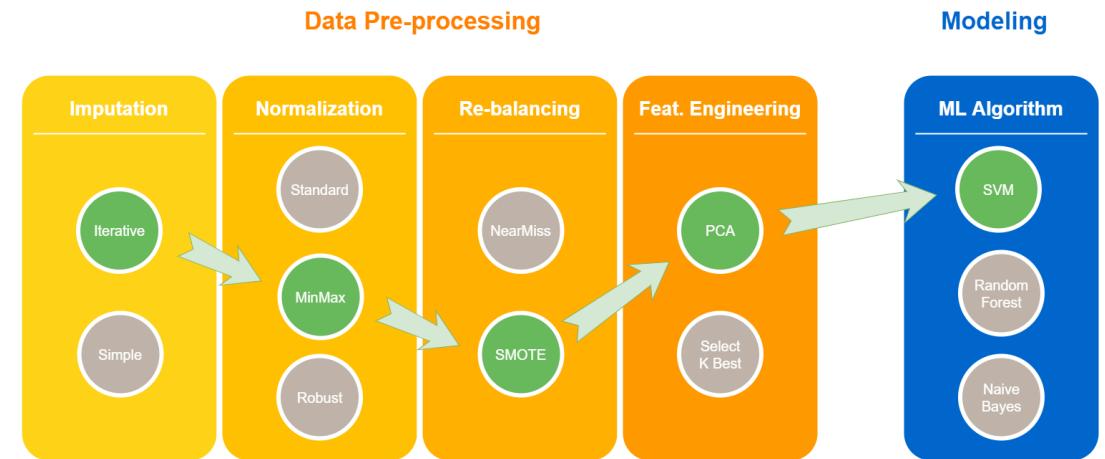
# Pipelines for ML tasks



# Pipelines for ML tasks

Tuning pipelines is hard

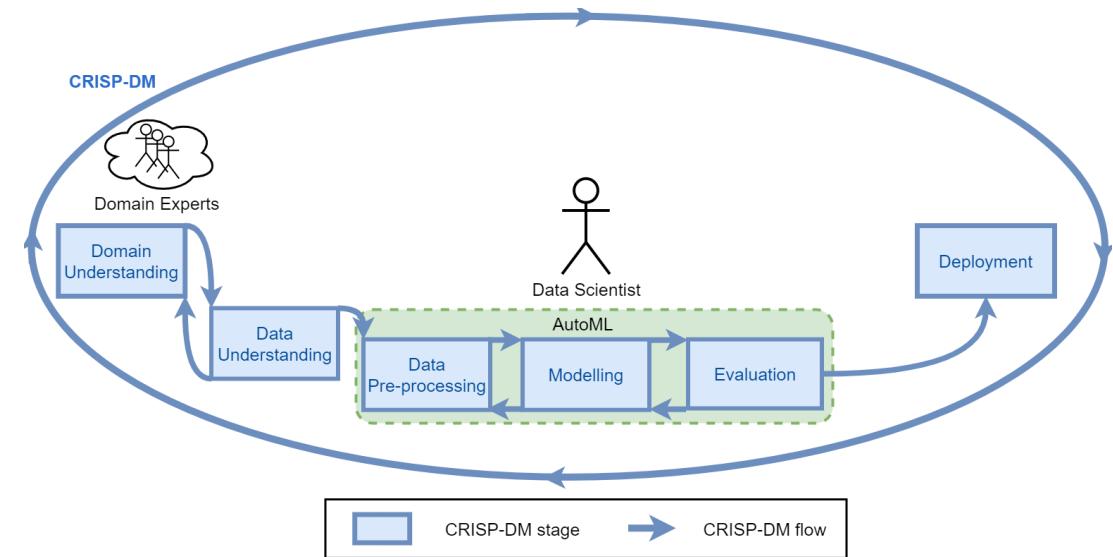
- At each **step**, a technique must be selected
- For each technique, a set of **hyper-parameters** must be set
- Each **hyper-parameter has its own search space**



# AutoML

AutoML aims at automating the ML pipeline instantiation:

- it is difficult to consider all the constraints together;
- it is not transparent;
- it doesn't allow a proper knowledge augmentation.



Thornton, et al. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 847-855).

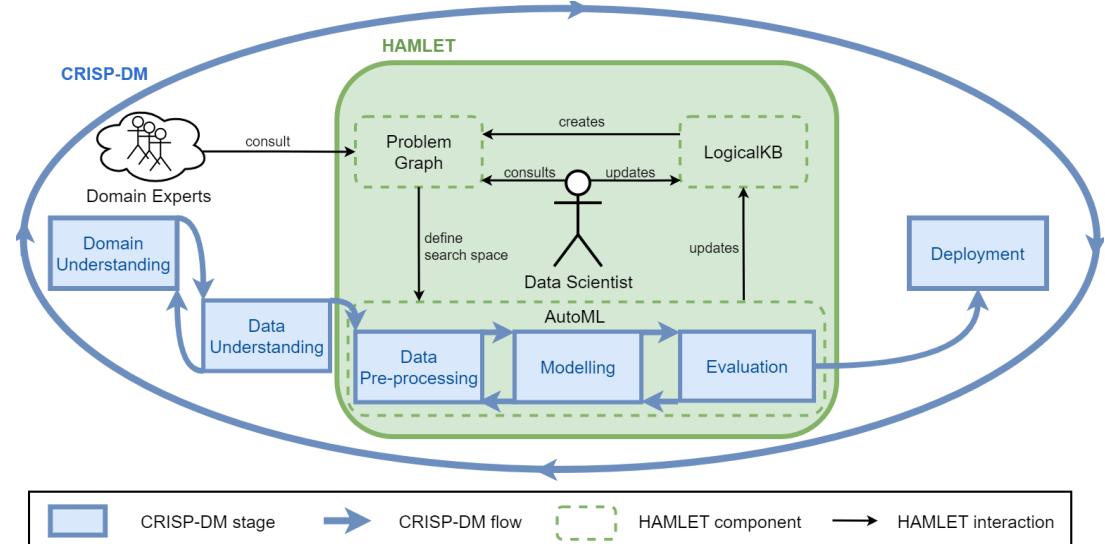
Feurer, Matthias, et al. "Auto-sklearn 2.0: Hands-free automl via meta-learning." The Journal of Machine Learning Research 23.1 (2022): 11936-11996.

# HAMLET

## HAMLET: Human-centric AutoML via Logic and Argumentation

### HAMLET leverages :

- Logic to give a structure to the knowledge;
- Argumentation to deal with inconsistencies, and revise the results.



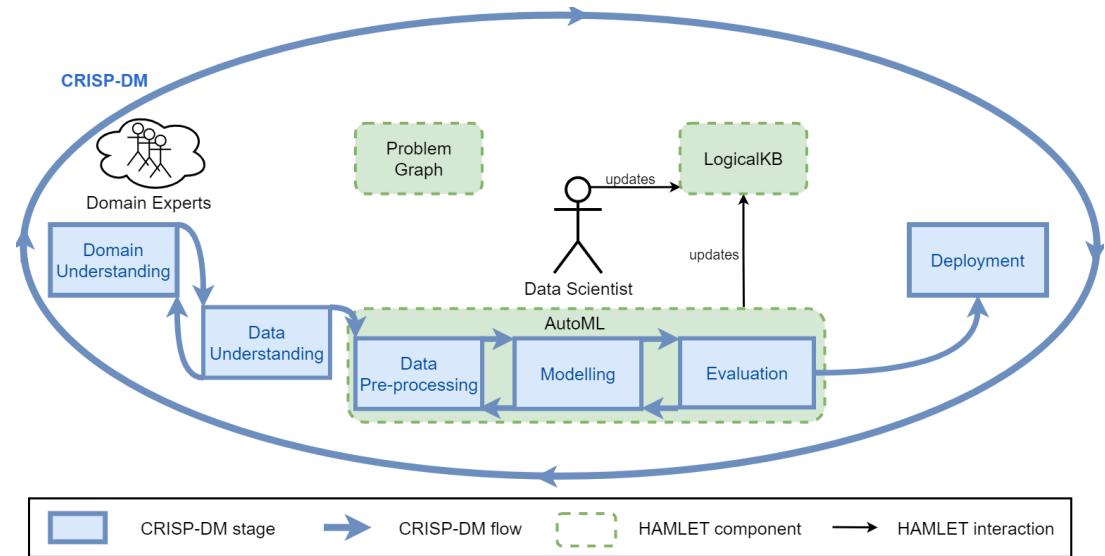
<https://github.com/QueueInc/HAMLET>

Francia M., Giovanelli J., and Pisano P. "HAMLET: A framework for Human-centered AutoML via Structured Argumentation." Future Generation Computer Systems 142 (2023): 182-194.

# HAMLET

The LogicalKB enables:

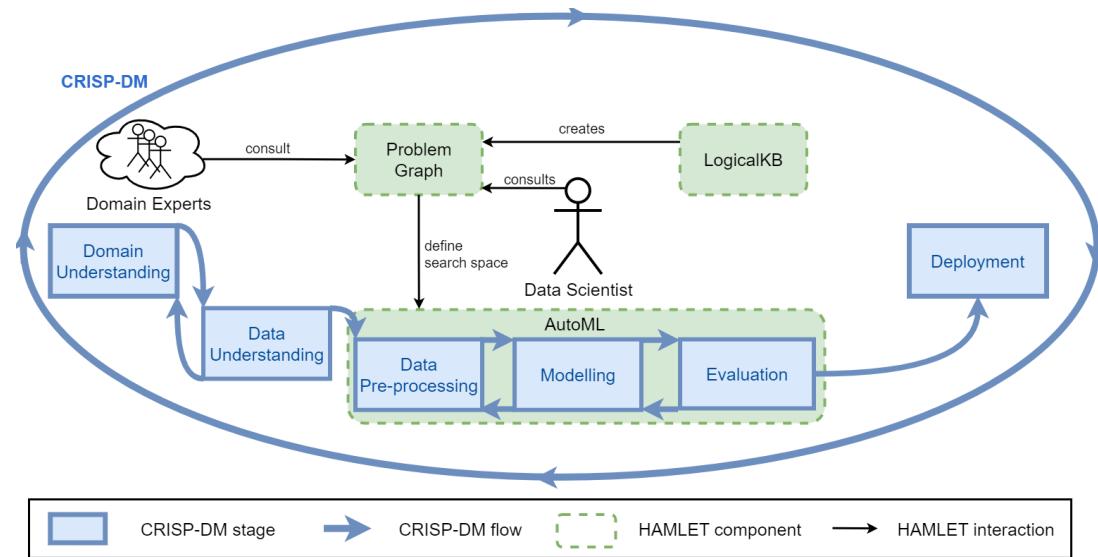
- the Data Scientist to structure the ML constraints;
- the AutoML tool to encode the explored results



# HAMLET

The Problem Graph allows to:

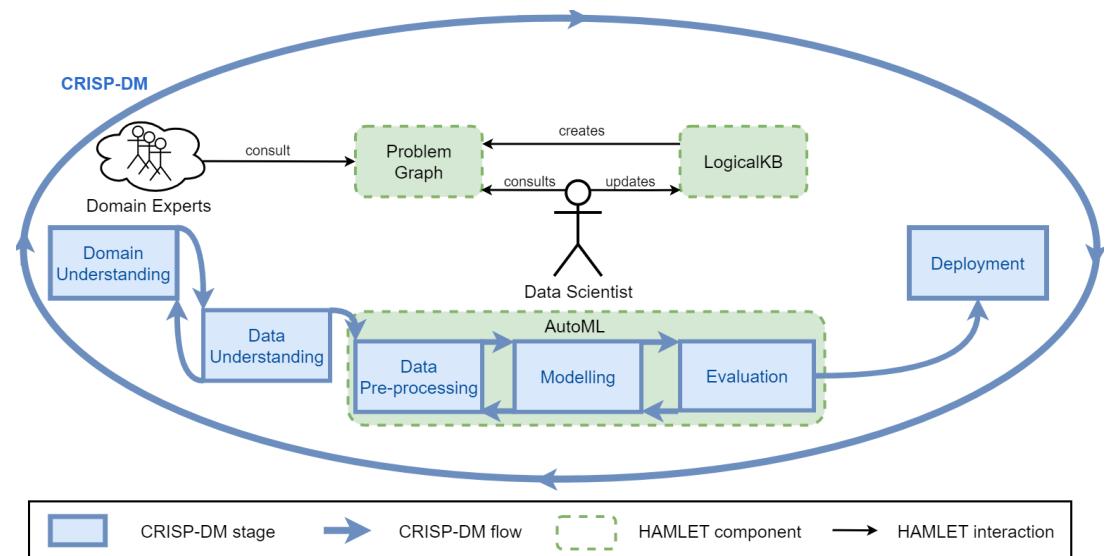
- consider all the ML constraints together;
- set up the AutoML search space;
- discuss and argument about the results.



# HAMLET

The Data Scientist iterates on:

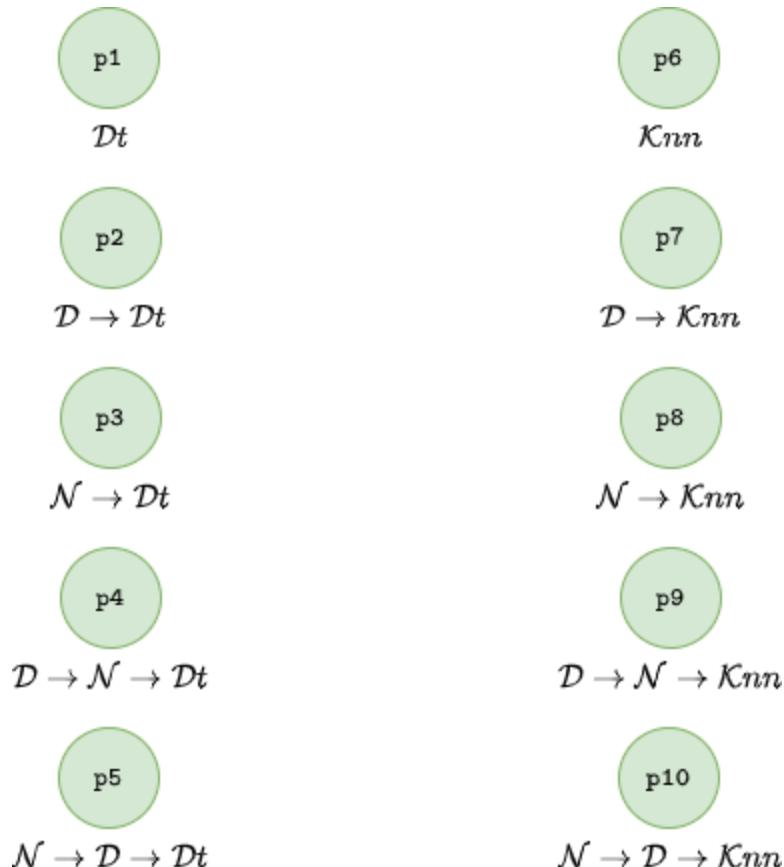
1. editing the LogicalKB;
2. consulting the Problem Graph;
3. running the AutoML tool;
4. discussing the AutoML insights.



# KB and Problem Graph

```
Declare steps pipeline
s1 : ⇒ step(D).
s2 : ⇒ step(N).
s3 : ⇒ step(Cl).

Declare classification algorithms
a1 : ⇒ algorithm(Cl, Dt).
a2 : ⇒ algorithm(Cl, Knn).
```

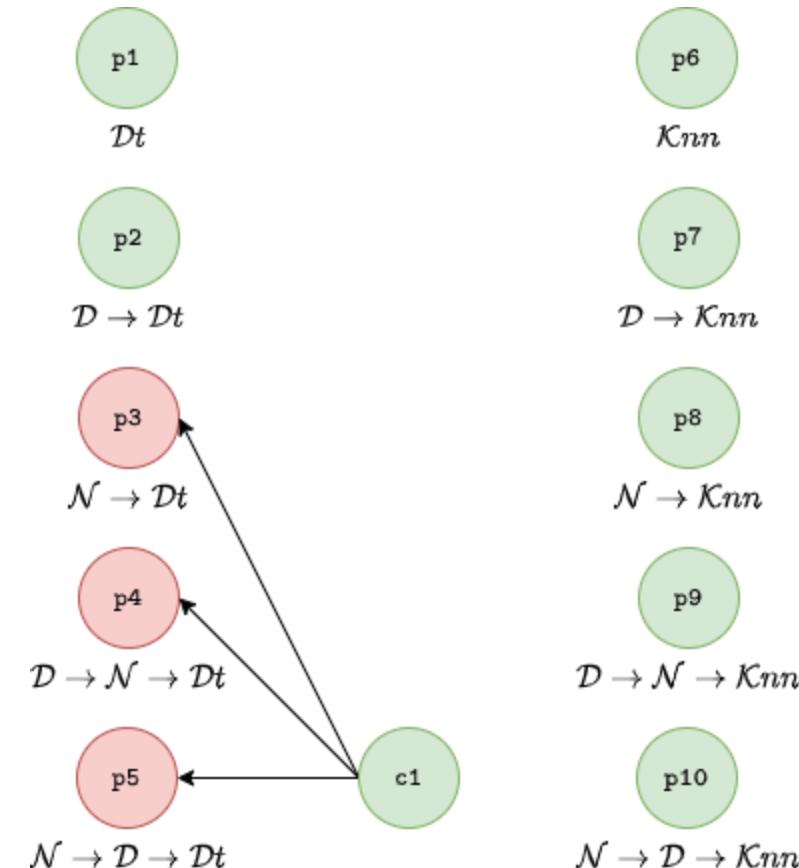


# KB and Problem Graph

```
Declare steps pipeline
s1 : ⇒ step(D).
s2 : ⇒ step(N).
s3 : ⇒ step(Cl).

Declare classification algorithms
a1 : ⇒ algorithm(Cl, Dt).
a2 : ⇒ algorithm(Cl, Knn).
```

```
Forbid Normalization when using DT
c1 : ⇒ forbidden(<N>, Dt).
```



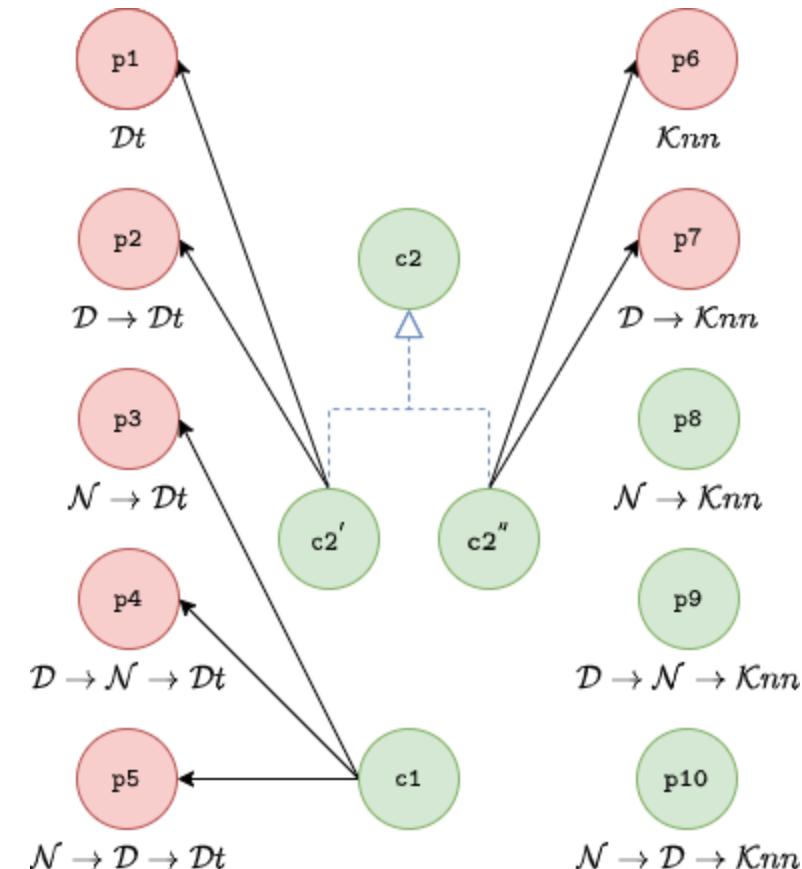
# KB and Problem Graph

```
Declare steps pipeline
s1 : ⇒ step(D).
s2 : ⇒ step(N).
s3 : ⇒ step(Cl).
```

```
Declare classification algorithms
a1 : ⇒ algorithm(Cl, Dt).
a2 : ⇒ algorithm(Cl, Knn).
```

```
Forbid Normalization when using DT
c1 : ⇒ forbidden(<N>, Dt).
```

```
Mandatory Normalization in Classification Pipelines
c2 : ⇒ mandatory(<N>, Cl).
```



# KB and Problem Graph

```
Declare steps pipeline
```

```
s1 : => step(D).
```

```
s2 : => step(N).
```

```
s3 : => step(Cl).
```

```
Declare classification algorithms
```

```
a1 : => algorithm(Cl, Dt).
```

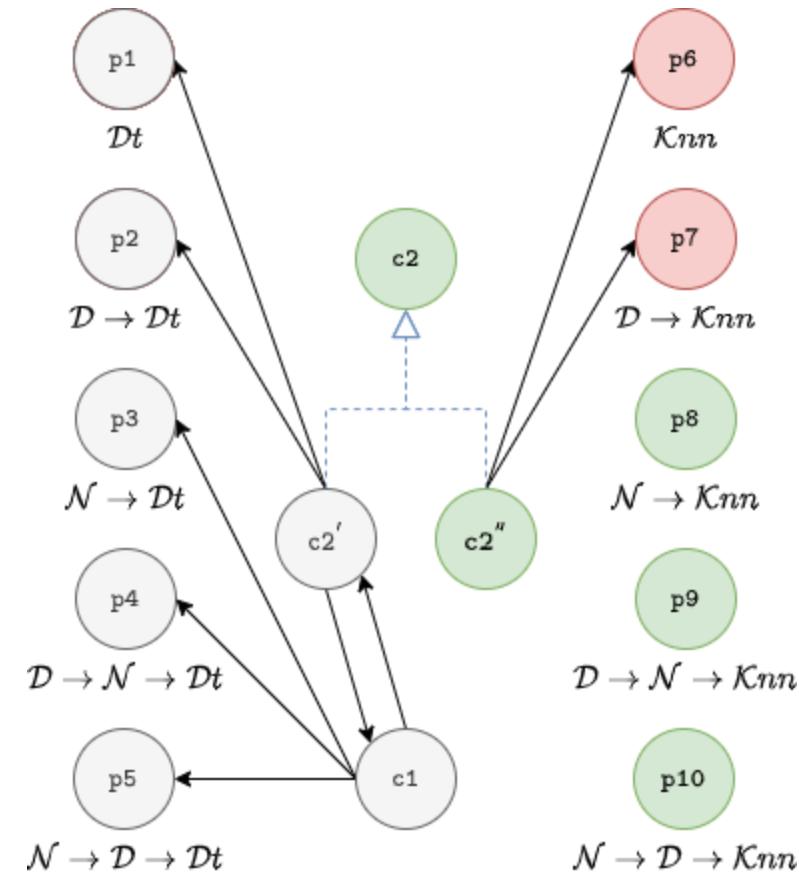
```
a2 : => algorithm(Cl, Knn).
```

```
Forbid Normalization when using DT
```

```
c1 : => forbidden(<N>, Dt).
```

```
Mandatory Normalization in Classification Pipelines
```

```
c2 : => mandatory(<N>, Cl).
```



# KB and Problem Graph

```
Declare steps pipeline
```

```
s1 : => step(D).
```

```
s2 : => step(N).
```

```
s3 : => step(Cl).
```

```
Declare classification algorithms
```

```
a1 : => algorithm(Cl, Dt).
```

```
a2 : => algorithm(Cl, Knn).
```

```
Forbid Normalization when using DT
```

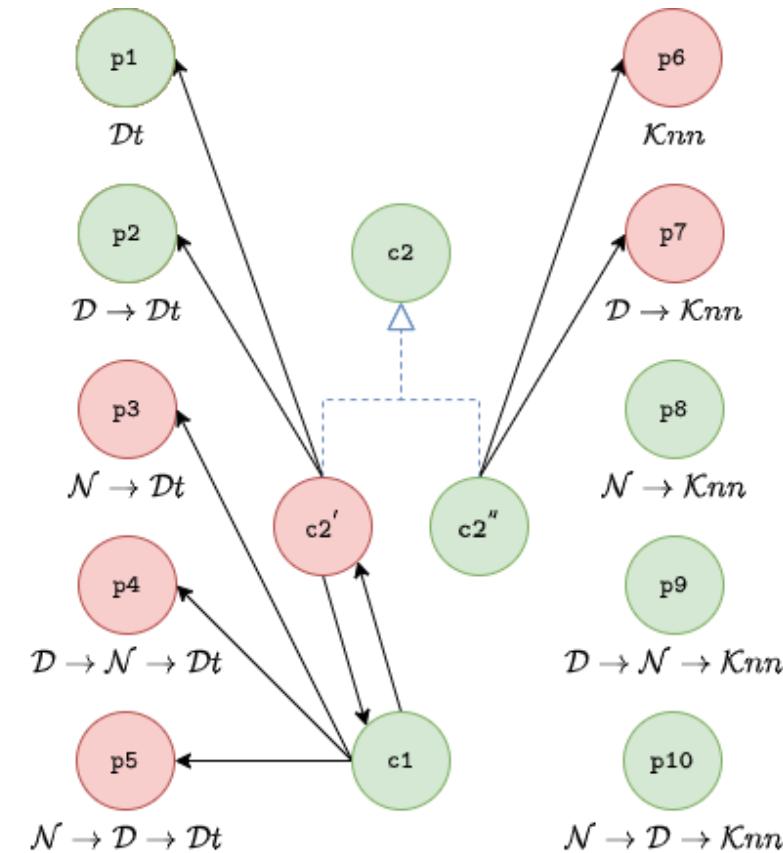
```
c1 : => forbidden(<N>, Dt).
```

```
Mandatory Normalization in Classification Pipelines
```

```
c2 : => mandatory(<N>, Cl).
```

```
Resolve conflict between c1 and c2
```

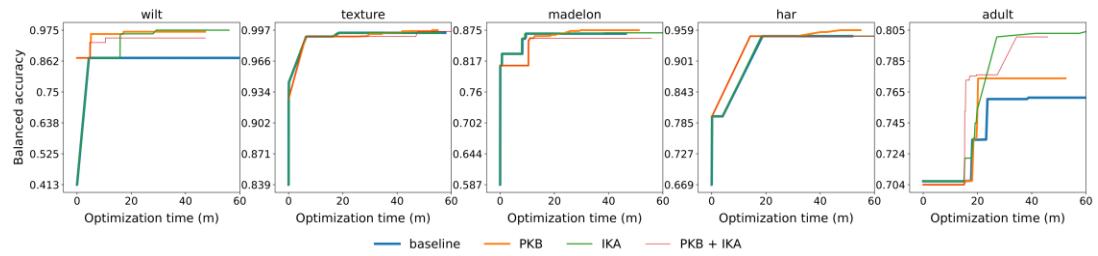
```
sup (c1, c2).
```



# Evaluation

## Settings:

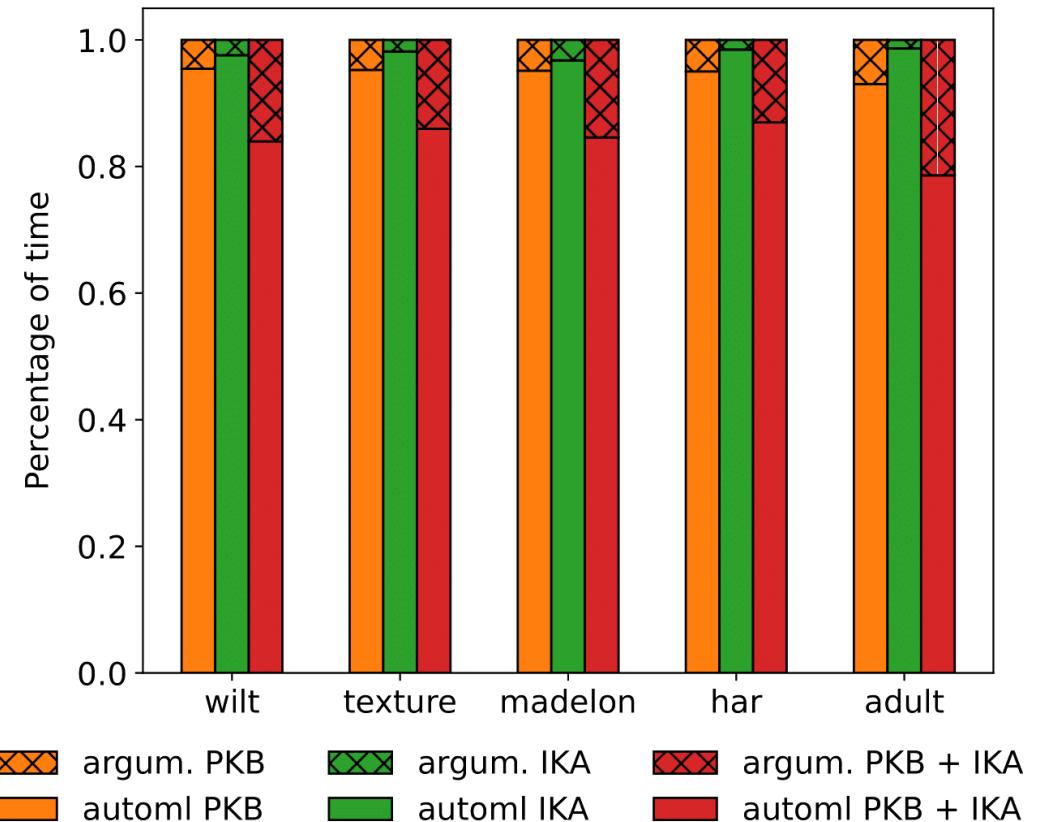
- **Baseline:** 1 optimization it. of 60 mins;
- **PKB** (Preliminary Knowledge Base): 1 optimization it. of 60 mins with non-empty LogicalKB;
- **IKA** (Iterative Knowledge Augmentation): 4 optimization it. of 15 mins with empty LogicalKB;
- **PKB + IKA:** 4 optimization it. of 15 mins with non-empty LogicalKB.



# Evaluation

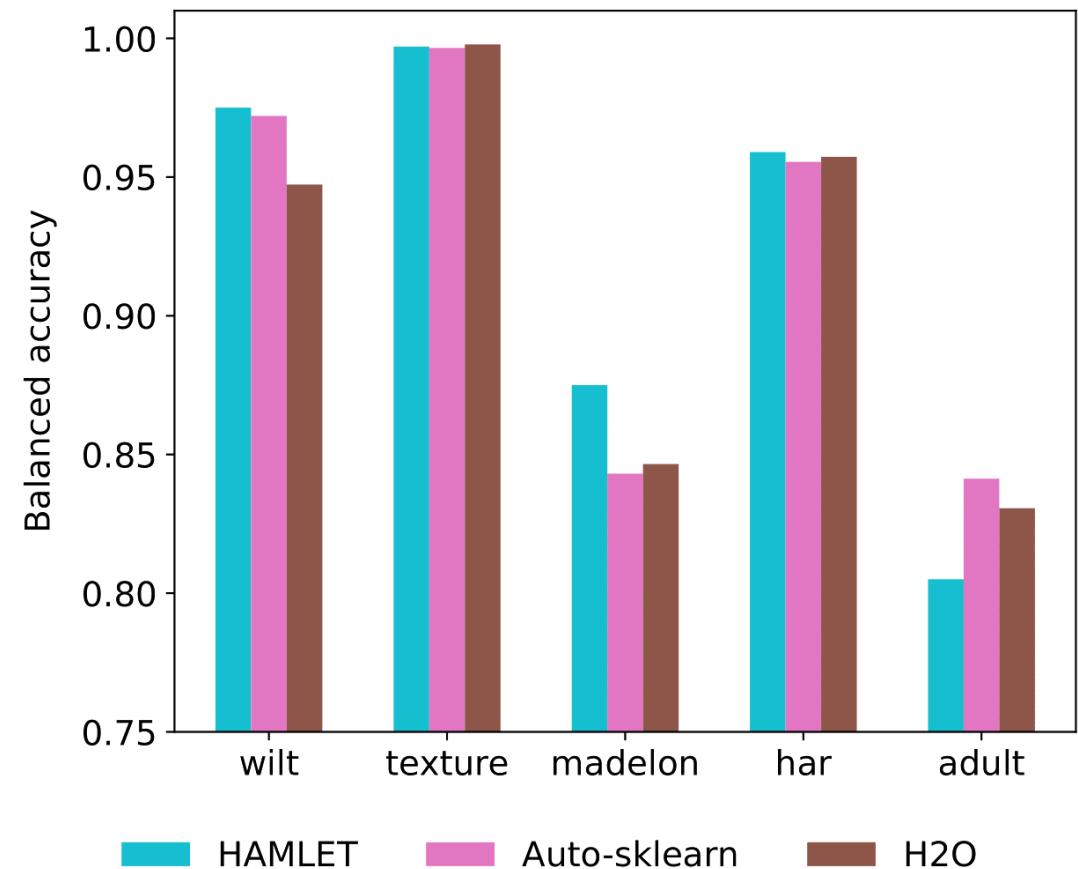
## Settings:

- **Baseline**: 1 optimization it. of 60 mins;
- **PKB** (Preliminary Knowledge Base): 1 optimization it. of 60 mins with non-empty LogicalKB;
- **IKA** (Iterative Knowledge Augmentation): 4 optimization it. of 15 mins with empty LogicalKB;
- **PKB + IKA**: 4 optimization it. of 15 mins with non-empty LogicalKB.



# Evaluation

Comparison with AutoML tools



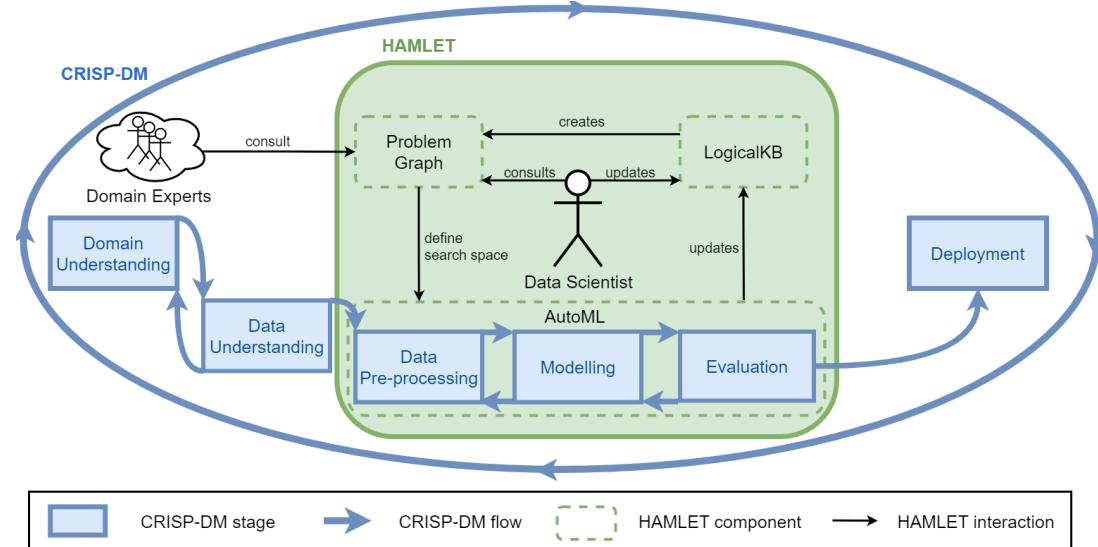
# HAMLET

## Key features:

- knowledge injection;
- representation via an human- and machine-readable medium;
- insight discovery;
- dealing with possible arising inconsistencies.

## Future directions:

- make constraints fuzzy;
- improve recommendation algorithm;
- enhance HAMLET with meta-learning;
- manage cross-cutting constraints (e.g., ethic, legal).



# Advanced Analytics

Applications and Challenges

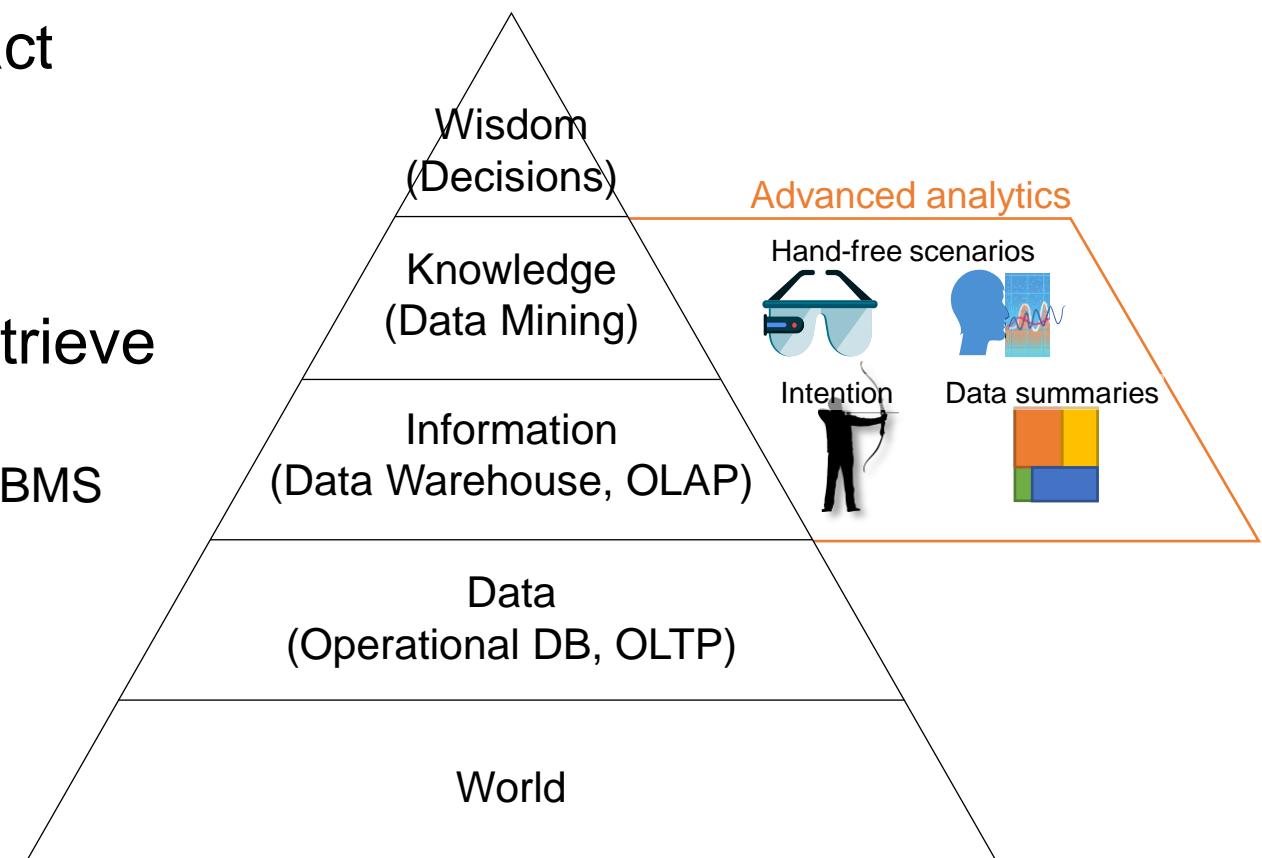
# Advanced Analytics

High availability and accessibility attract new data scientists

- High competence in business domain
- Low competence in computer science

Since the '70s, relational queries to retrieve data

- Comprehension of formal languages and DBMS
- Advanced analytics (semi-automatic transformation)
  - “Information” and “Knowledge” levels



# Advanced Analytics

Many problems to address:

- Query recommendation based on contextual data
  - E.g., augmented reality and digital twins
- Definition of interest
- Diversification
- Compression
- Natural Language and Vocalization

# Application scope

Enable analytics through augmented reality [1]

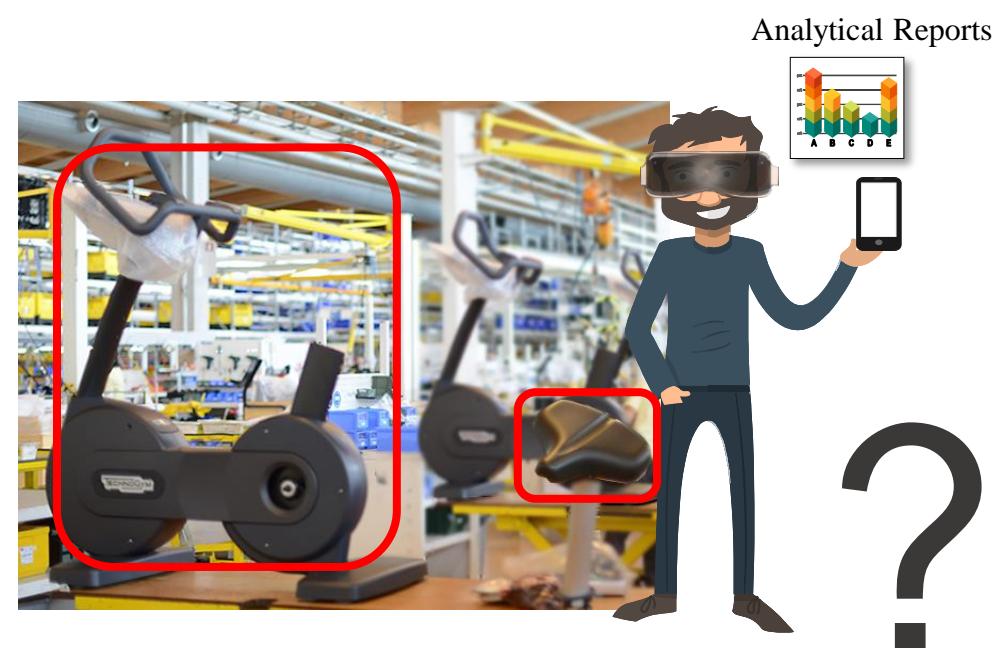
- E.g., an inspector analyzing production rates

Sense the context through augmented devices

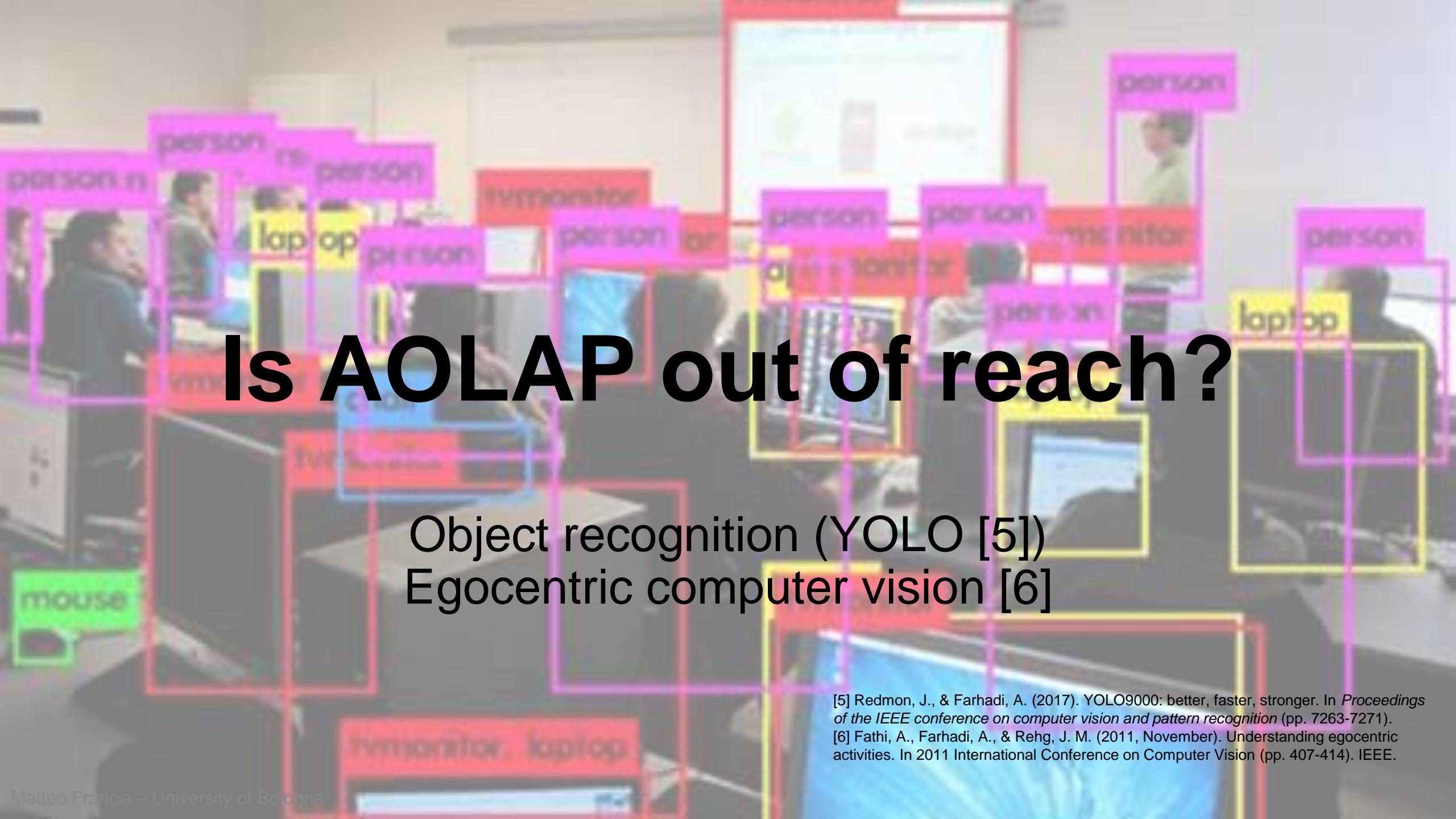
- E.g., smart glasses
- Detect interaction and engagement [1]

Produce analytical reports

- Relevant to the sensed context
- Cardinality constraint
- Near real-time



[1] Francia, Matteo, Matteo Golfarelli, and Stefano Rizzi. "A-BI+: a framework for Augmented Business Intelligence." *Information Systems* 92 (2020): 101520.  
[2] Yu-Chuan Su, Kristen Grauman: Detecting Engagement in Egocentric Video. ECCV (5) 2016: 454-471



# Is AOLAP out of reach?

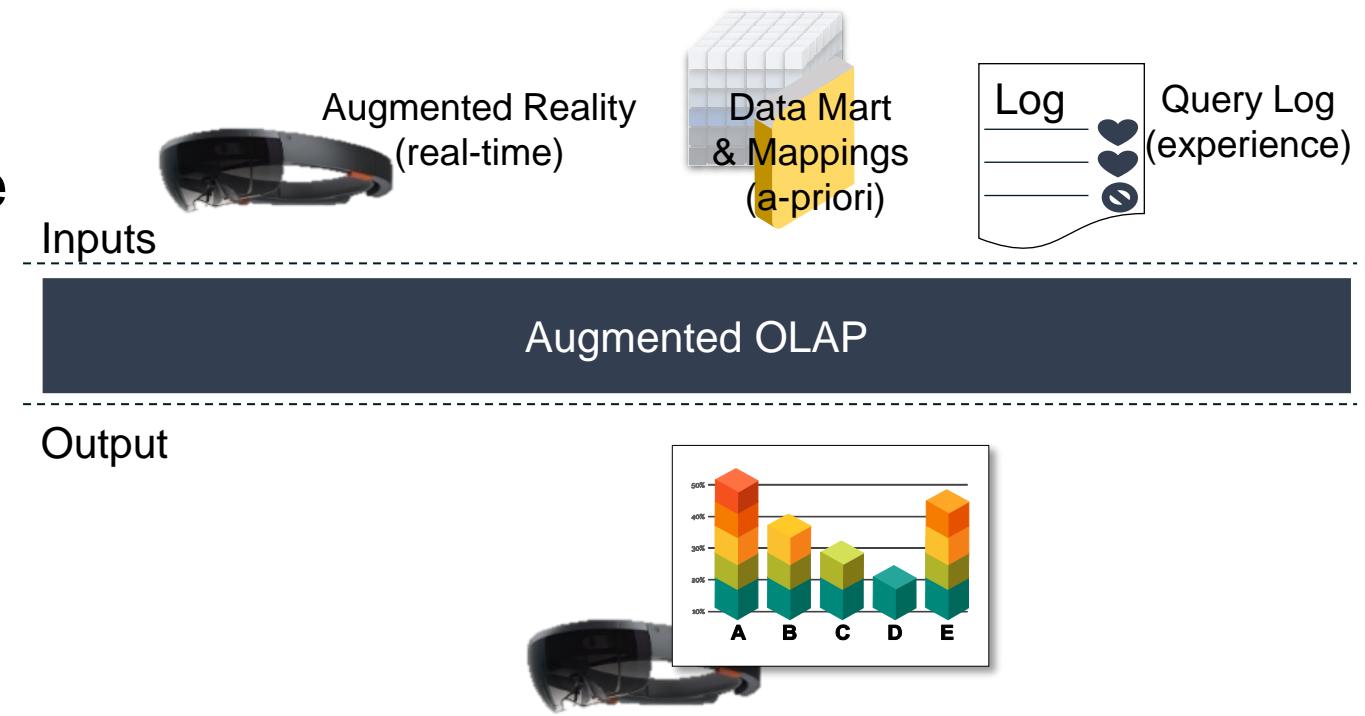
Object recognition (YOLO [5])  
Egocentric computer vision [6]

- [5] Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263-7271).  
[6] Fathi, A., Farhadi, A., & Rehg, J. M. (2011, November). Understanding egocentric activities. In *2011 International Conference on Computer Vision* (pp. 407-414). IEEE.

# Augmented OLAP

## Augmented OLAP, a 3D marriage

- Augmented reality
  - Real-time information [2]
- Business intelligence
  - OLAP: get data facts
- Recommendation
  - Pick relevant data facts



[2] Angelo Croatti, Alessandro Ricci: Towards the Web of Augmented Things. ICSA Workshops 2017: 80-87

# What can we sense?

Data Mart: repository of multidimensional cubes

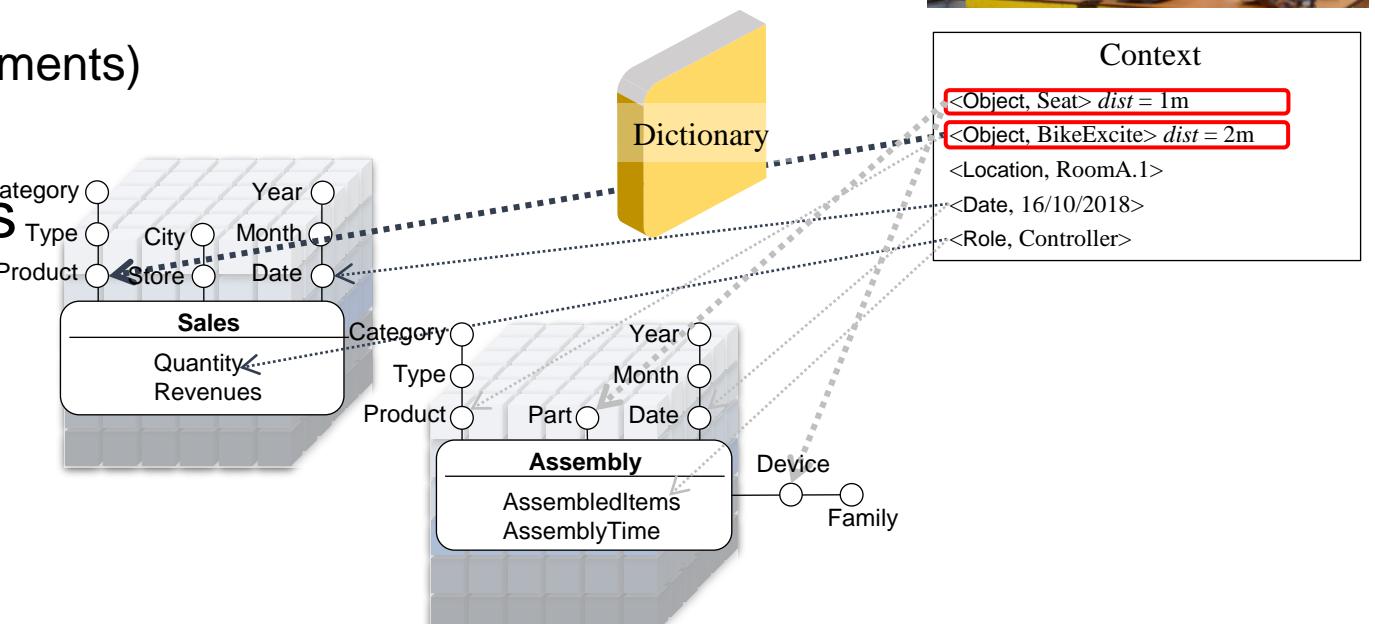
- Cubes representing business facts

Data dictionary

- What we can recognize (i.e., md-elements)
- **Context**: subset of md-elements

Mappings to sets of md-elements

- A-priori interest



# Recommendation

## Context interpretation

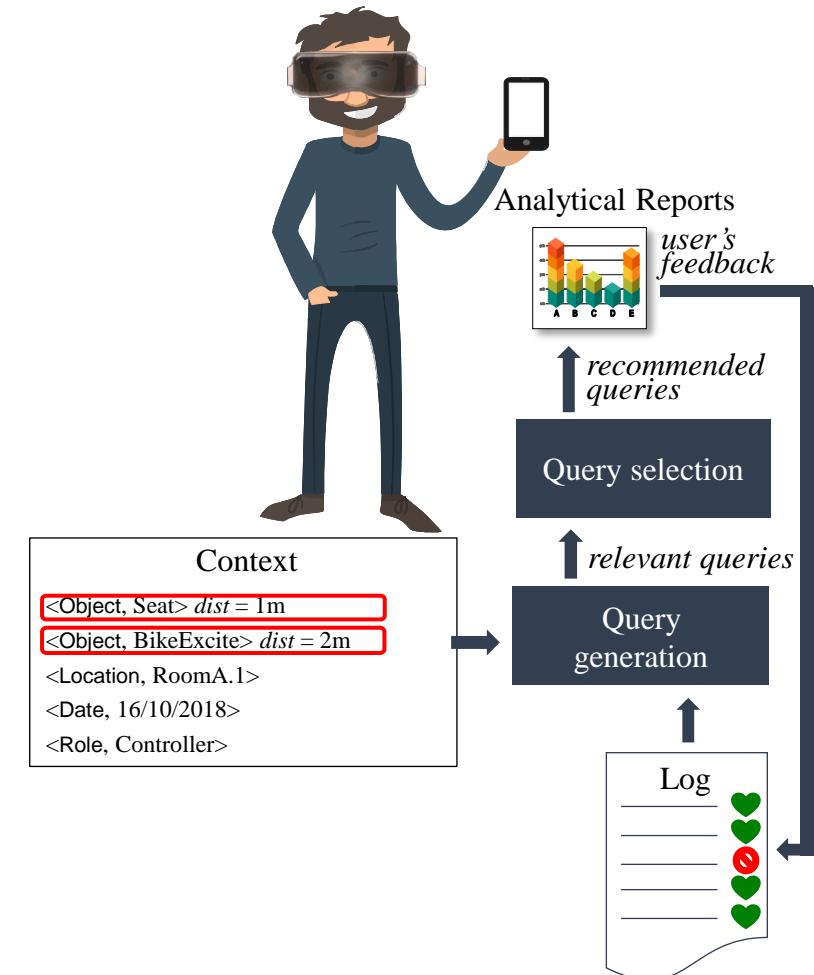
- Given context  $T$  over the data dictionary
- Project  $T$  to an **image of fragments**  $I$  through mappings
  - **Fragment**: intuitively a “small” query

## Add the log

- Get queries with positive feedback from *similar* contexts
  - Enrich  $I$  to  $I^*$  with *unperceived* elements from  $T$
- Each fragment has *contextual and log relevance*

## Query generation

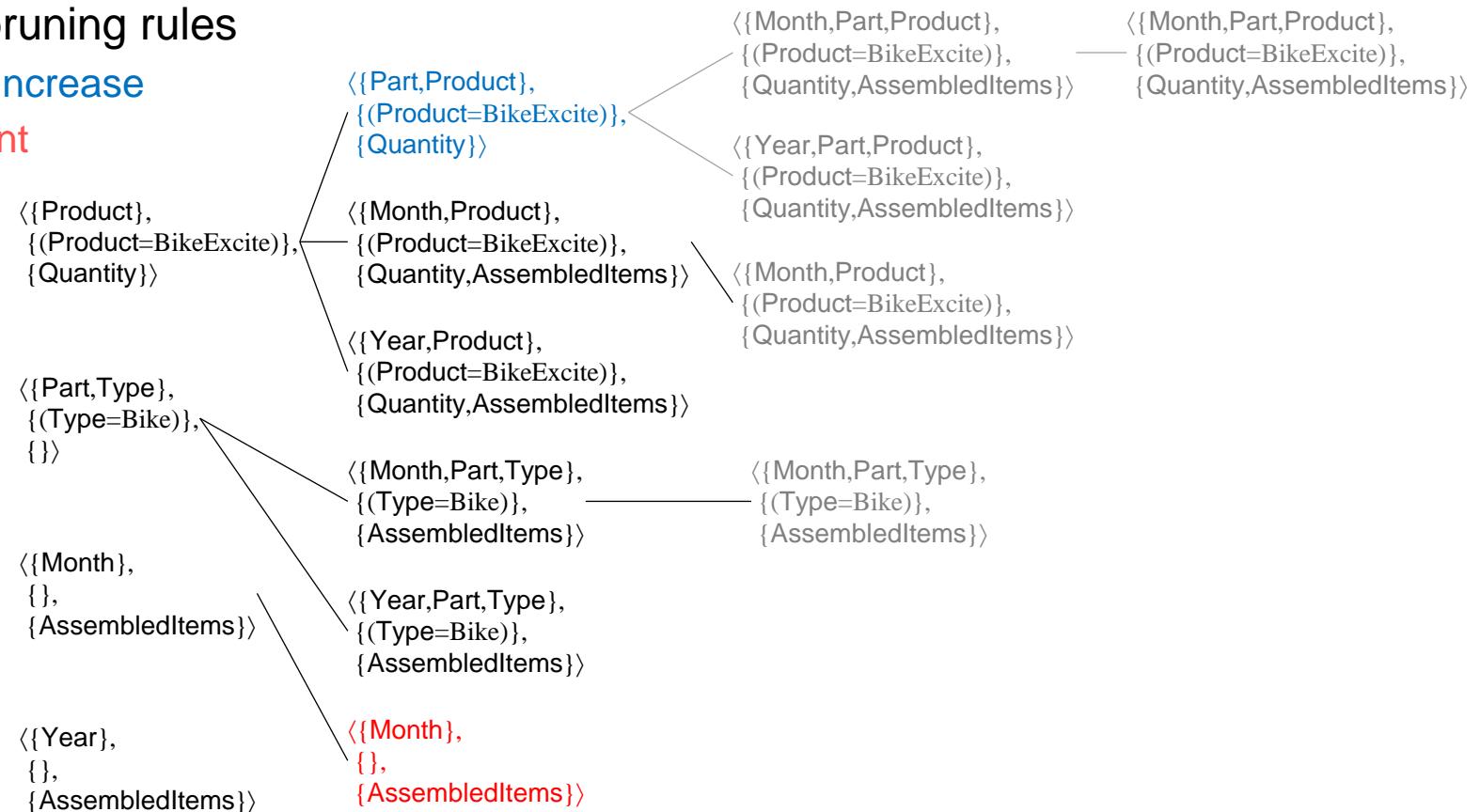
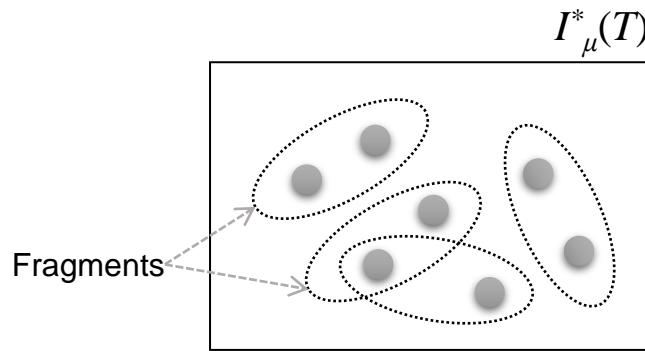
- Cannot directly translate  $I^*$  into a well-formed query
- High cardinality  $I^*$  = hardly interpretable “monster query”



# Query generation

Generate queries from image  $I^*(T)$  of fragments

- Each fragment is a query
- Depth-first exploration with pruning rules
  - Query cardinality can only increase
  - Some queries are redundant



# Query selection

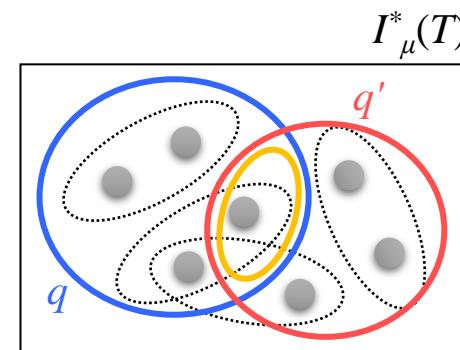
Given #queries ( $rq$ ), maximize the covered fragments and minimize their overlapping

$$rel_T(Q) = \sum_{\emptyset \neq Q' \subseteq Q} sim(Q') \cdot \frac{\sum_{q \in Q'} rel_T(q)}{|Q'|} \cdot (-1)^{|Q'|+1}$$

E.g., given two queries  $q$  and  $q'$

$$rel(q) + rel(q') - sim(q, q') * (rel(q) + rel(q')) / 2$$

- Weighted Maximum Coverage Problem (NP-hard)
- Greedy: iteratively pick query maximizing  $rel_T$ 
  - Only a few query are retrieved, not expensive



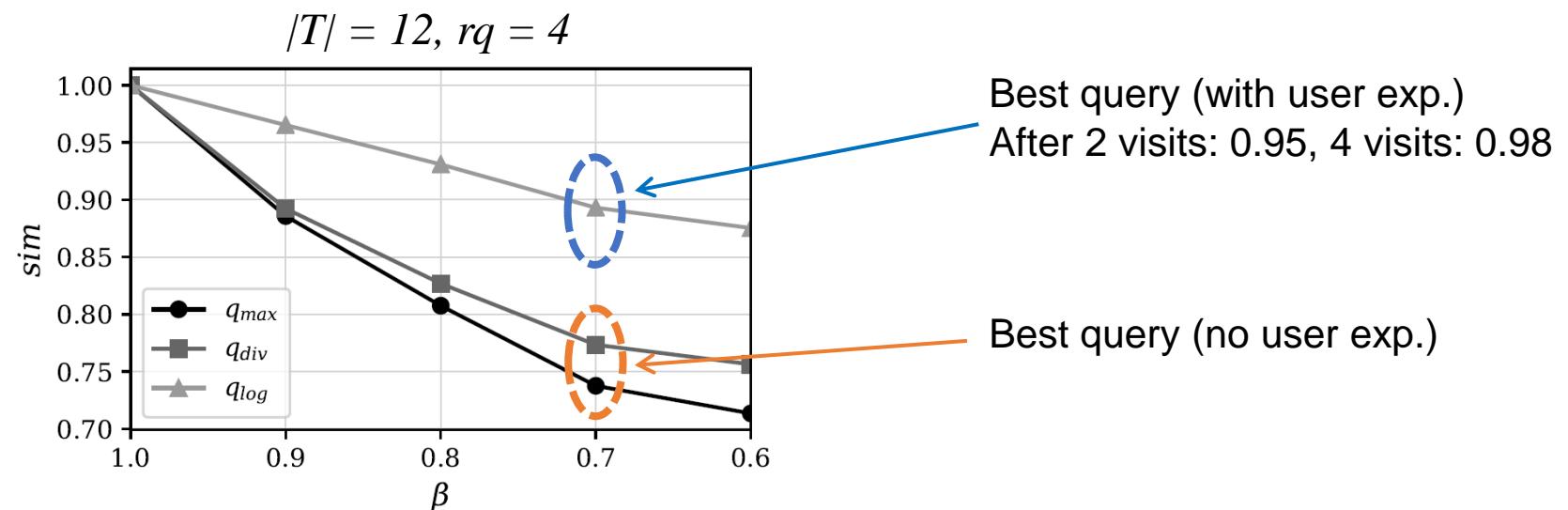
# Effectiveness

## Test set up

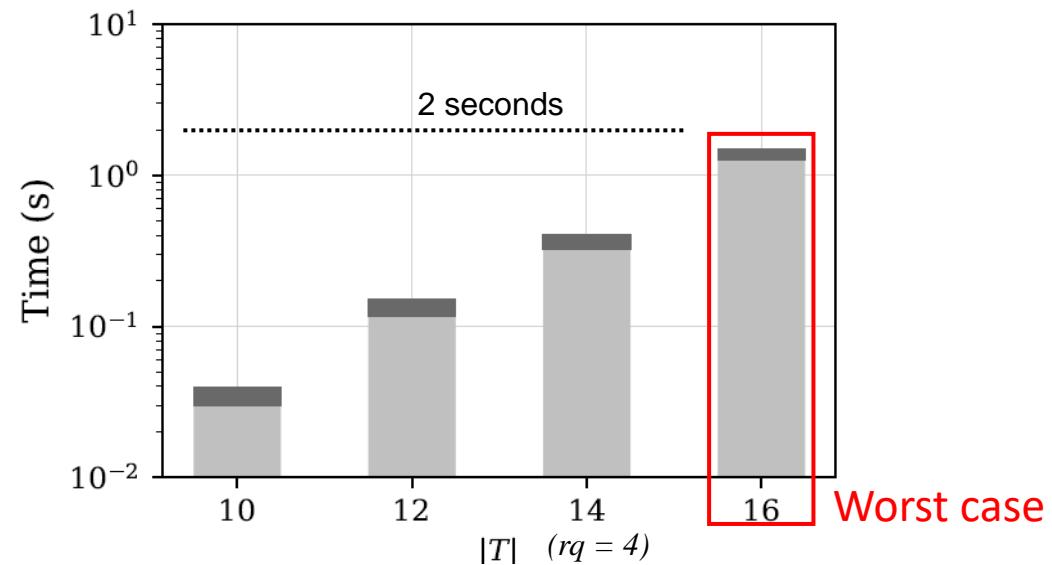
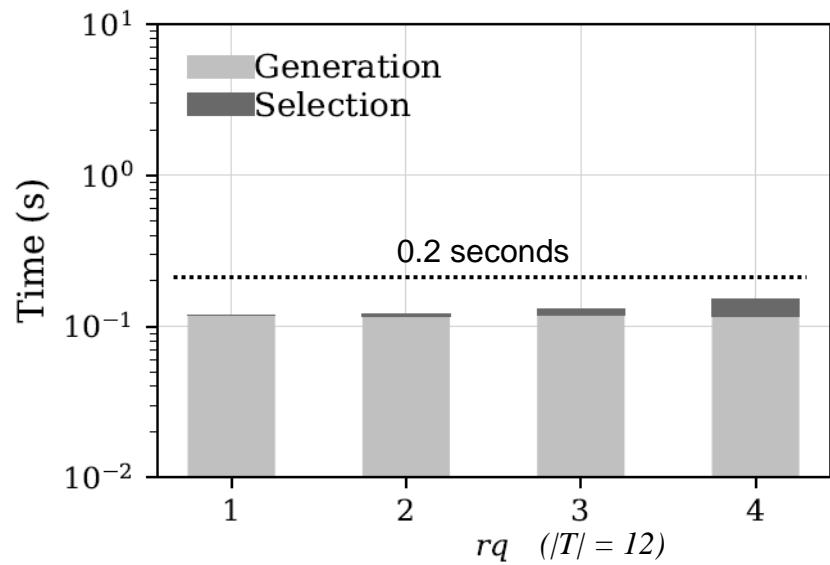
- Cube with  $10^9$  md-elements
- Simulate user moving inside a factory

Given **fixed context** and query target

- Assess similarity of the proposed query in **similar contexts**
- $\beta$ : context similarity
- $sim$ : proposed/target query similarity



# Efficiency



# Research directions

## Analytics in augmented reality

- Support analytical queries in hand-free scenarios
- Recommend relevant data facts from a real-world context

## Research directions

- Provide (fast) query previews
  - Estimate the execution time of each query
  - Address query caching and multi-query optimization issues
- Correlate context-awareness to data quality [3]
  - Relevance, amount, and completeness [4]

[3] Stephanie Watts, Ganesan Shankaranarayanan, Adir Even: Data quality assessment in context: A cognitive perspective. Decis. Support Syst. 48(1): 202-211 (2009)

[4] Diane M. Strong, Yang W. Lee, Richard Y. Wang: Data Quality in Context. Commun. ACM 40(5): 103-110 (1997)

# Motivation

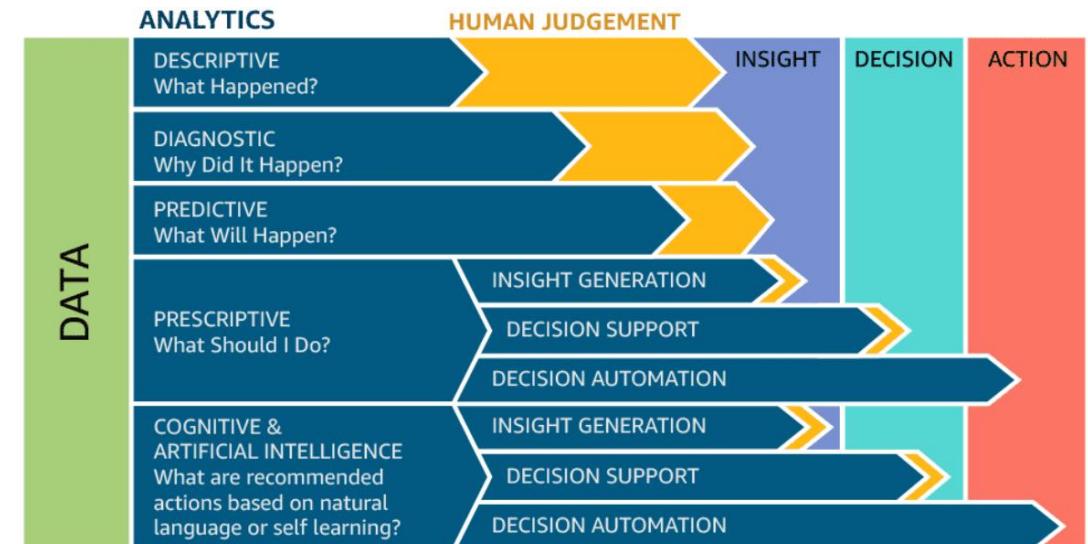
Enable analytics through natural language

OLAP provides low-level operators [1]

- Users need to have knowledge on the multidimensional model...
- ... or even programming skills

We introduce COOL (COnversational OLap) [3]

- Translate natural language into formal queries

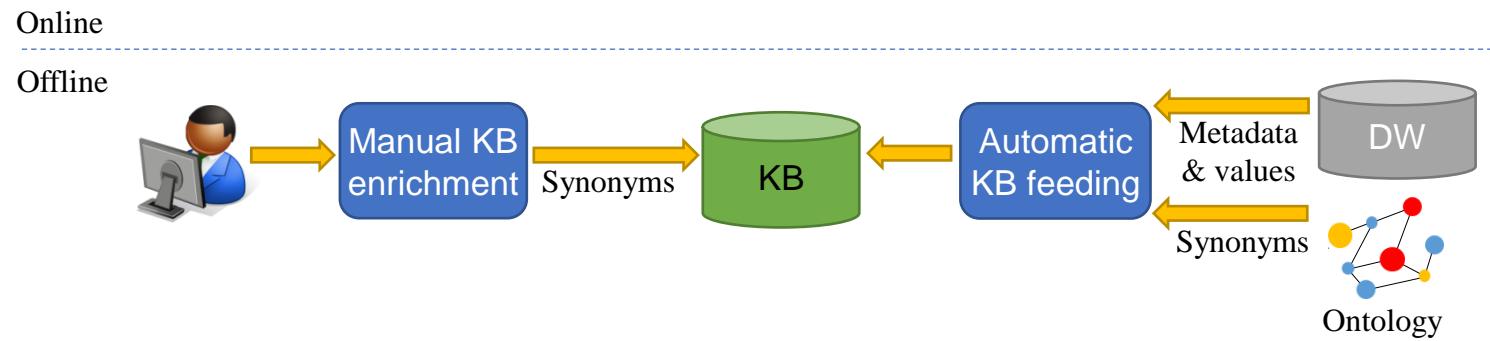


[1] Panos Vassiliadis, Patrick Marcel, Stefano Rizzi: Beyond roll-up's and drill-down's: An intentional analytics model to reinvent OLAP. *Information Systems*. (2019)

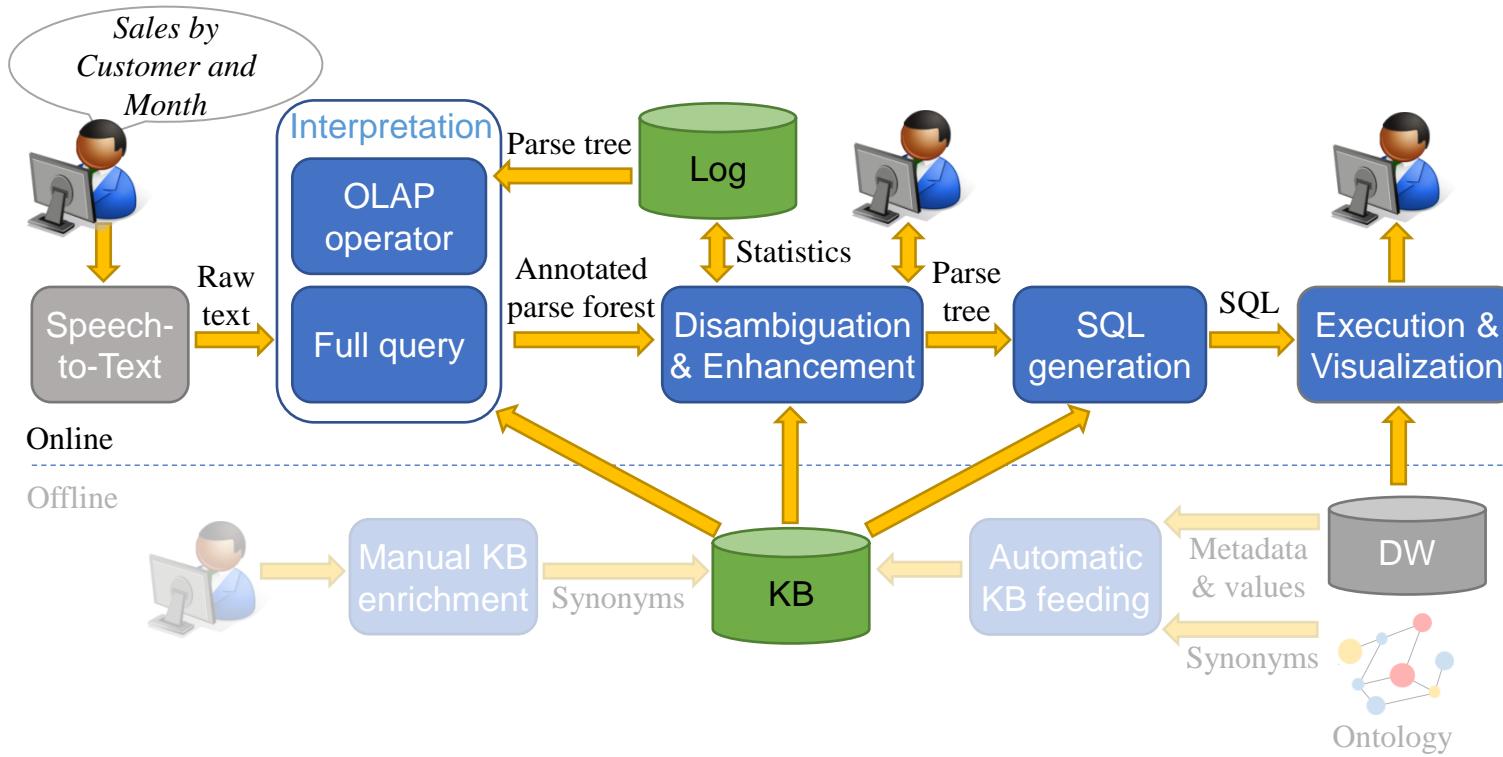
[2] Matteo Francia, Matteo Golfarelli, Stefano Rizzi: A-BI+: A framework for Augmented Business Intelligence. *Information Systems*. (2020)

[3] Matteo Francia, Enrico Gallinucci, Matteo Golfarelli: COOL: A Framework for Conversational OLAP. *Information Systems*. (2021)

# COOL: architecture

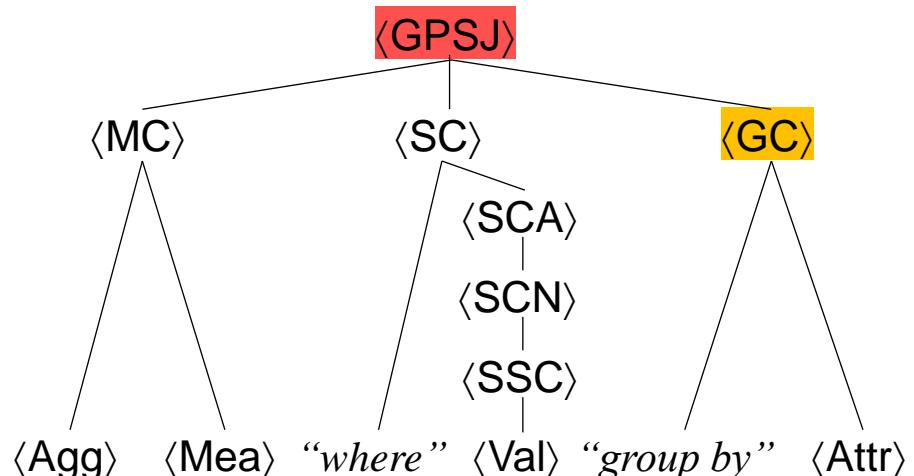


# COOL: architecture



# COOL: interpretation

```
<GPSJ> ::= <MC><GC><SC>
<MC> ::= (<Agg><Mea> | <Cnt><Fct>)+
<GC> ::= "group by" <Attr>+
<SC> ::= "where" <SCA>
<SCA> ::= <SCN> "and" <SCA> | <SCN>
<SCN> ::= "not" <SSC> | <SSC>
<SSC> ::= <Attr><Cop><Val> | <Attr><Val> | <Val>
<Cop> ::= "=" | "<>" | ">" | "<" | ">=" | "<="
<Agg> ::= "sum" | "avg" | "min" | "max"
<Cnt> ::= "count" | "count distinct"
<Fct> ::= Domain-specific facts
<Mea> ::= Domain-specific measures
<Attr> ::= Domain-specific attributes
<Val> ::= Domain-specific values
```



$M_1 = \langle \text{avg}, \text{UnitSales}, \text{where}, 2019, \text{group by}, \text{Region} \rangle$

$T = \text{«return the average sales in 2019 per store region»}$

# Effectiveness

40 users with heterogeneous OLAP skills

- Asked to translate (Italian) analytic goals into English
- Users provided good feedback on the interface...
- ... as well as on the interpretation accuracy

| OLAP Familiarity | Full Query |          | OLAP operator |          |
|------------------|------------|----------|---------------|----------|
|                  | Accuracy   | Time (s) | Accuracy      | Time (s) |
| Low              | 0.91       | 141      | 0.86          | 102      |
| High             | 0.91       | 97       | 0.92          | 71       |

# COOL in Action!

COOL: Conversational OLAP

b business intelligence group

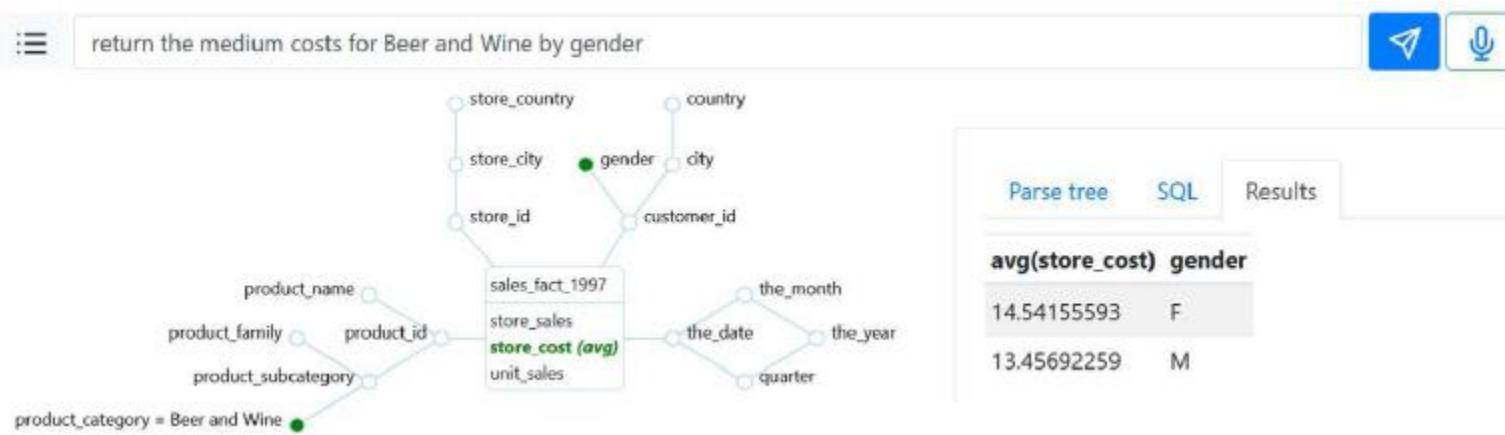
```
graph TD; SALESFACT[SALES_FACT_1997] --> STORECOUNTRY[STORE_COUNTRY]; SALESFACT --> STORECITY[STORE_CITY]; SALESFACT --> STOREID[STORE_ID]; SALESFACT --> CUSTOMERID[CUSTOMER_ID]; SALESFACT --> THEMONTH[THE_MONTH]; SALESFACT --> QUARTER[QUARTER]; SALESFACT --> THEDATE[THE_DATE]; SALESFACT --> THEYEAR[THE_YEAR]; STORECOUNTRY --- STORECITY; STORECITY --- STOREID; CUSTOMERID --- GENDER; CUSTOMERID --- STOREID; GENDER --- CITY; CITY --- STORECOUNTRY; PRODUCTNAME[PRODUCT_NAME] --- PRODUCTID[PRODUCT_ID]; PRODUCTID --- PRODUCTSUBCATEGORY[PRODUCT_SUBCATEGORY]; PRODUCTSUBCATEGORY --- PRODUCTCATEGORY[PRODUCT_CATEGORY]; PRODUCTCATEGORY --- PRODUCTNAME; PRODUCTFAMILY[PRODUCT_FAMILY] --- PRODUCTID; PRODUCTSUBCATEGORY --- PRODUCTID; PRODUCTID --- STOREID; THEMONTH --- THEYEAR; THEMONTH --- QUARTER; THEYEAR --- QUARTER; THEMONTH --- THEDATE; THEYEAR --- THEDATE; THEMONTH --- THEYEAR;
```

Full query mode

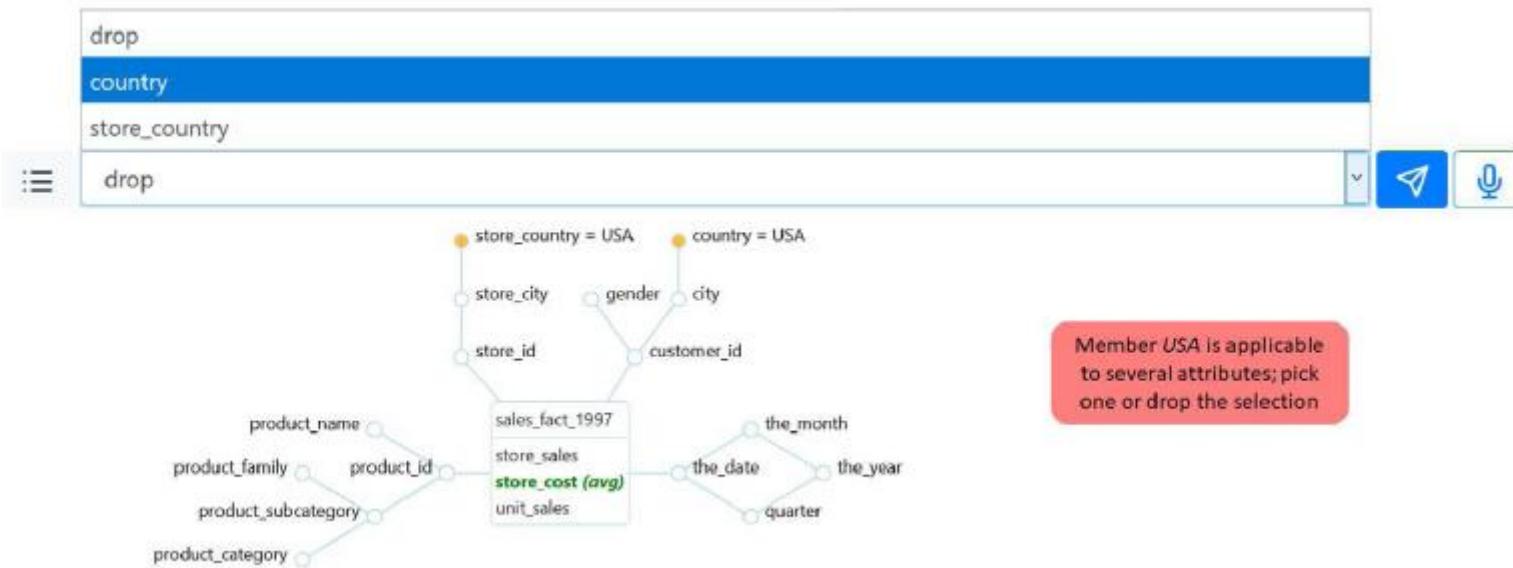
Please insert your query

≡ Send Microphone Trash

# COOL in Action!



# COOL in Action!



# Research directions

## COOL (Conversational OLAP)

- Support the translation of a natural language conversation into an OLAP session
- Analyze data without requiring technological skills
  - Add conversational capabilities to Augmented OLAP

## Towards an end-to-end conversational solution

- Create **query summaries** that can be returned as short vocal messages
- Identify **insights** out of a large amount of data
- Identify the “right” **storytelling** and user-system interaction