ICDE2023
Anaheim, CA April 3 – 7, 2023

Amalur: Data Integration Meets Machine Learning

Rihan Hai, Christos Koutras, Andra Ionescu, Ziyu Li, Wenbo Sun, Jessie van Schijndel, Yan Kang, Asterios Katsifodimos

Take-away

Q: Can we use data integration metadata to improve the effectiveness and efficiency of ML model training?

Scope

Theory

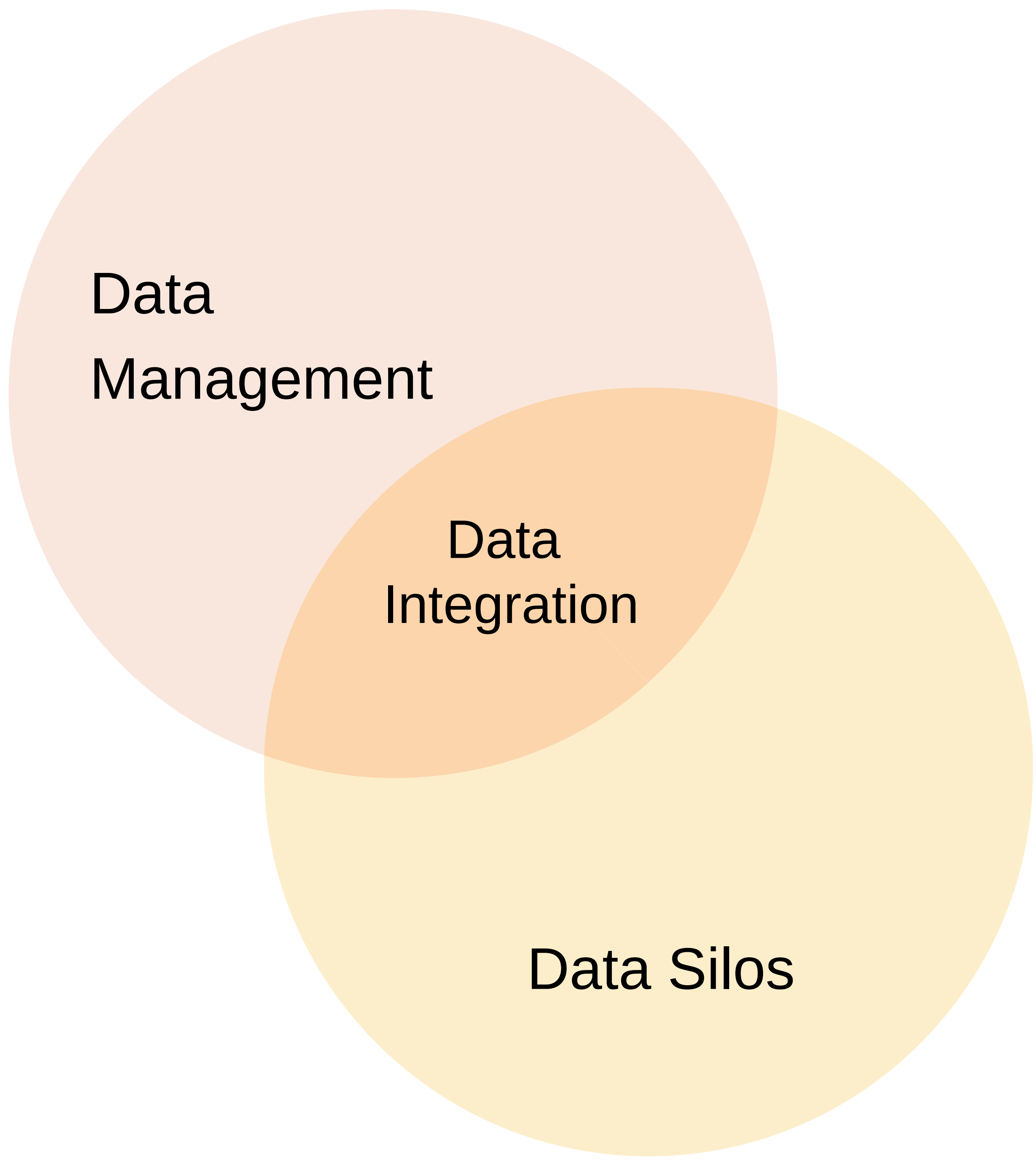
Systems

Scope

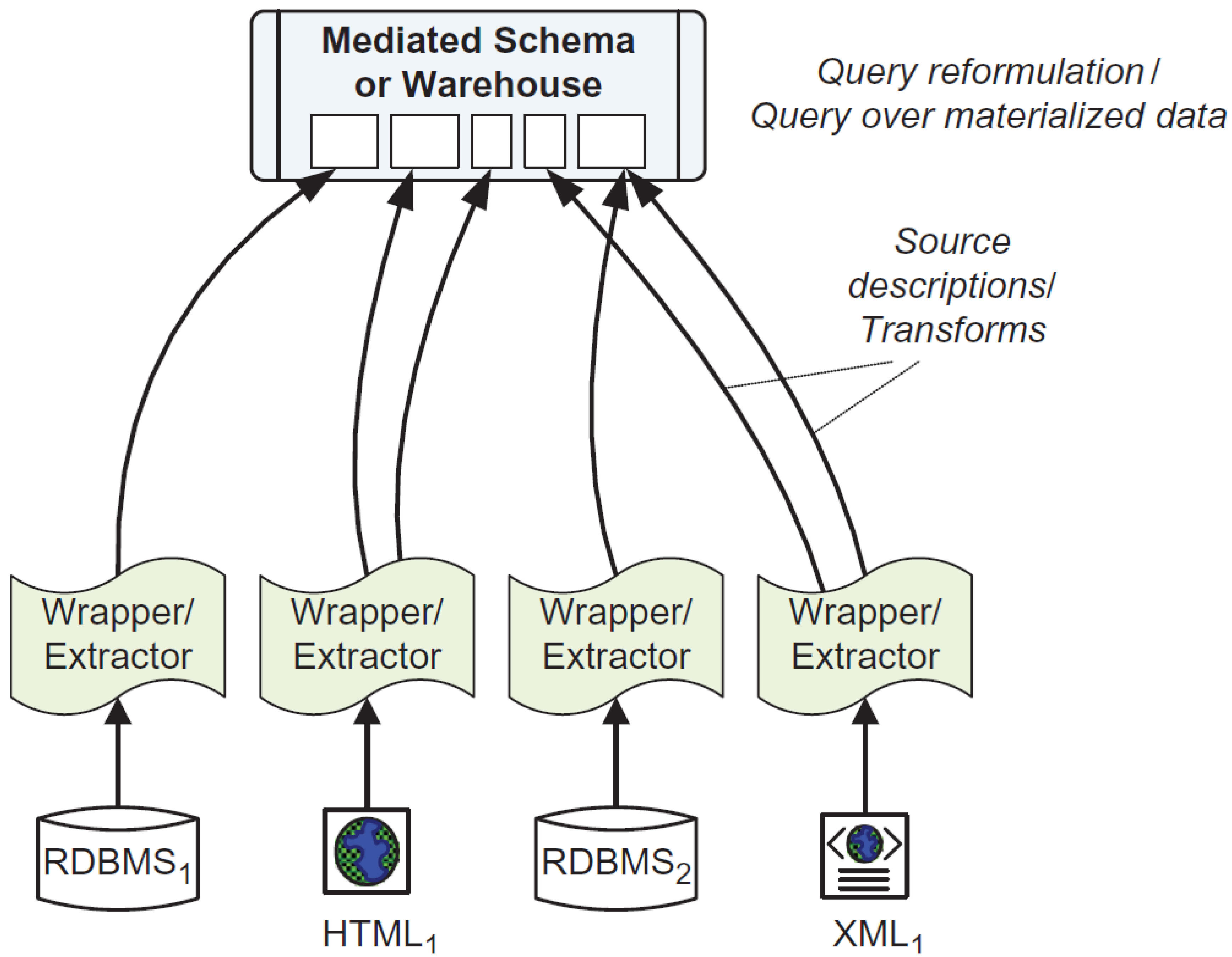
Theory

Systems

Data integration -- in a nutshell



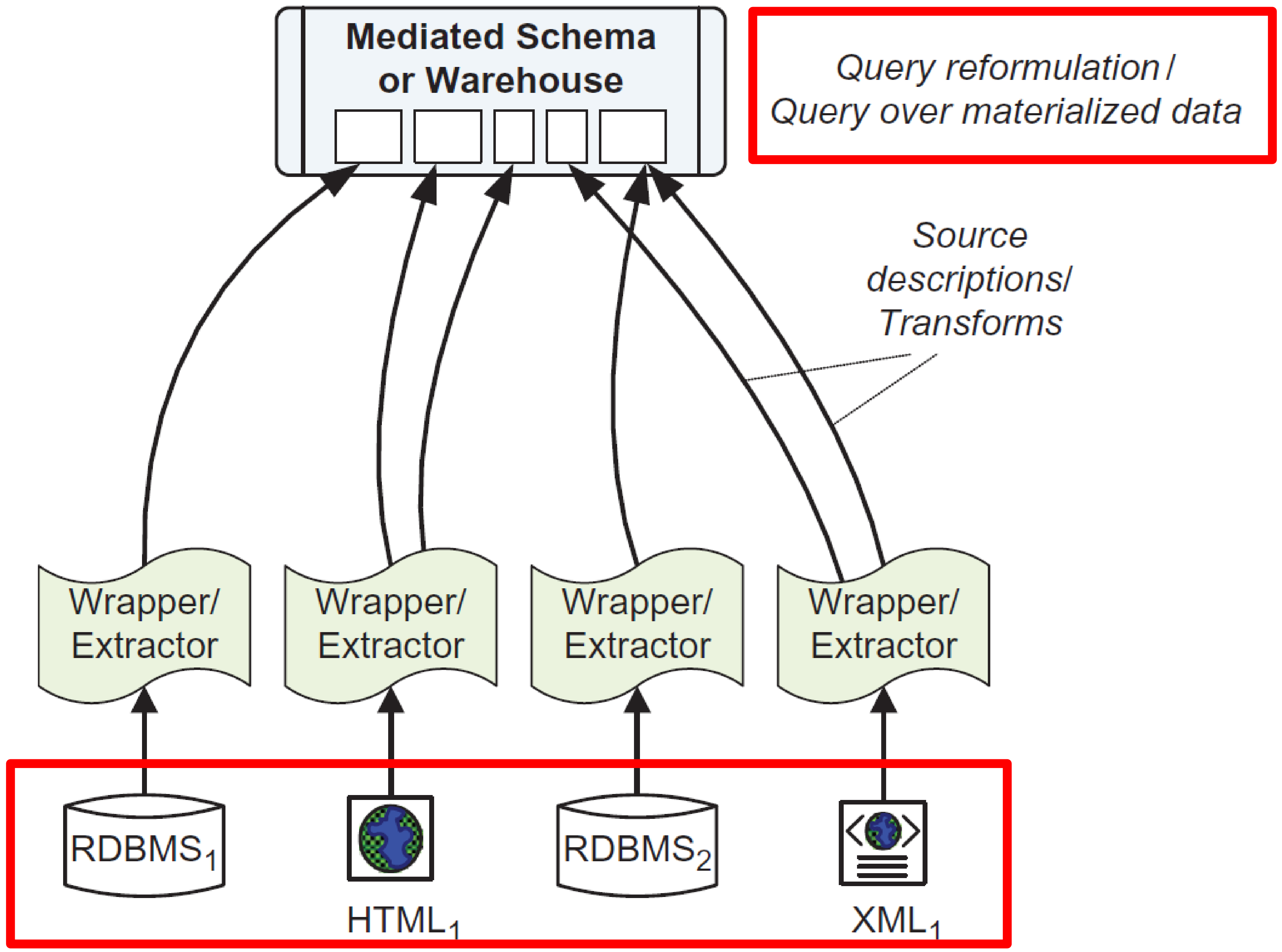
Data integration -- System architecture



Basic architecture of a general-purpose data integration system

Principles of Data Integration Alon Halevy, AnHai Doan, and Zachary

Data integration -- System architecture



Basic architecture of a general-purpose data integration system

Principles of Data Integration Alon Halevy, AnHai Doan, and Zachary

Data integration -- Schema matching

	mortality	pname	age	resting HR
0	0	Jack	20	60
1	0	Sam	35	58
2	0	Ruby	22	65
3	1	Jane	37	70

Hospital Amsterdam S_1

	mortality	name	age	oxygen	date diagnosed
0	1	Rose	45	95	1/4/21
1	0	Castiel	20	97	3/8/22
2	1	Jane	37	92	11/5/21

Hospital Delft S_2

Data integration -- Schema merging

Target table schema
(a.k.a. global schema, mediated schema)

Hospital T

mortality	age	resting HR	oxygen

	mortality	pname	age	resting HR
0	0	Jack	20	60
1	0	Sam	35	58
2	0	Ruby	22	65
3	1	Jane	37	70

Hospital Amsterdam S_1

	mortality	name	age	oxygen	date diagnosed
0	1	Rose	45	95	1/4/21
1	0	Castiel	20	97	3/8/22
2	1	Jane	37	92	11/5/21

Hospital Delft S_2

Data integration -- Schema mapping

Target table schema

mortality	age	resting HR	oxygen

Hospital T

	mortality	pname	age	resting HR
0	0	Jack	20	60
1	0	Sam	35	58
2	0	Ruby	22	65
3	1	Jane	37	70

Hospital Amsterdam S_1

	mortality	name	age	oxygen	date diagnosed
0	1	Rose	45	95	1/4/21
1	0	Castiel	20	97	3/8/22
2	1	Jane	37	92	11/5/21

Hospital Delft S_2

Principles of Data Integration *Aln Halevy, AnHai Doan, and Zachary*

Data integration -- Entity resolution

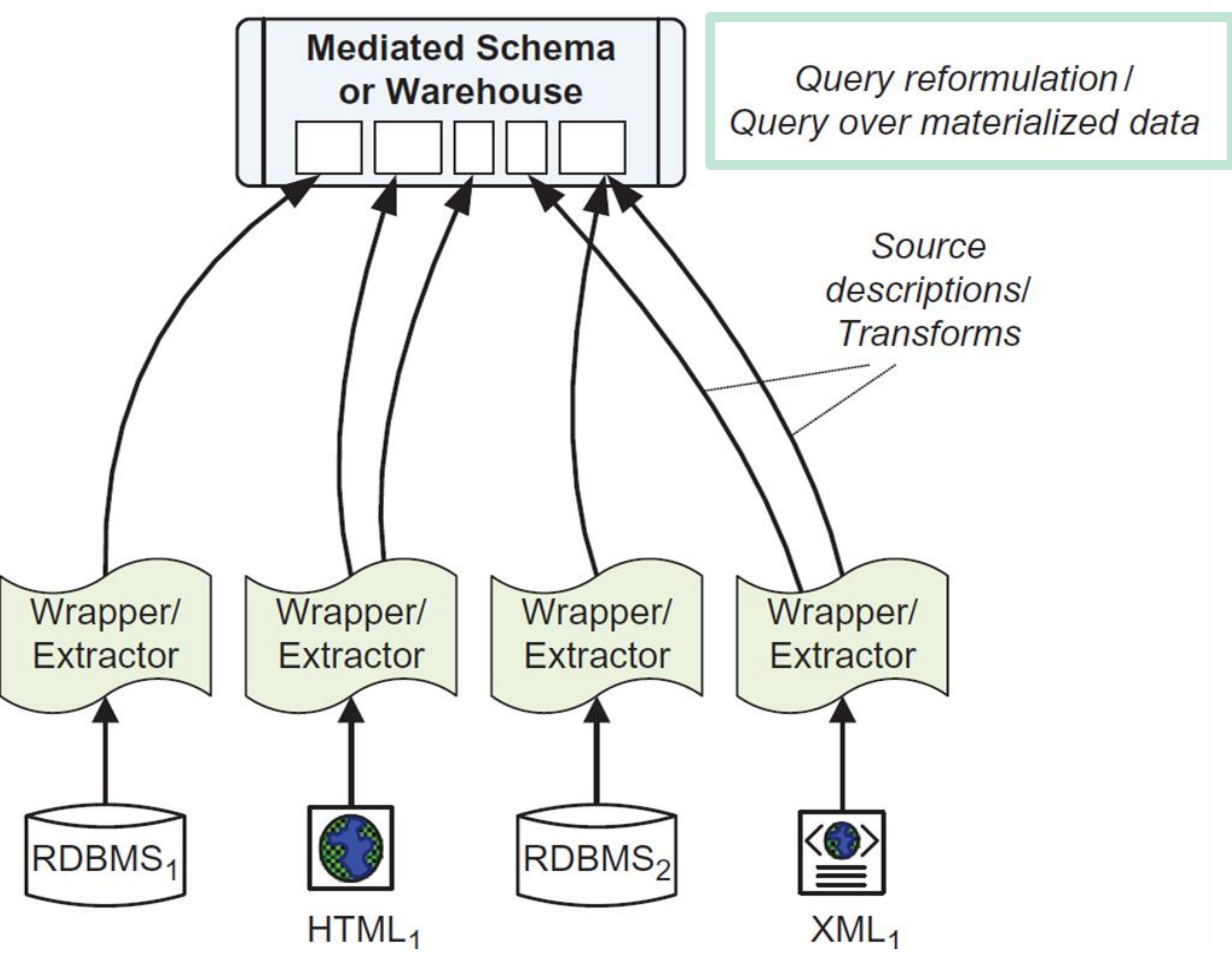
	mortality	pname	age	resting HR
0	0	Jack	20	60
1	0	Sam	35	58
2	0	Ruby	22	65
3	1	Jane	37	70

Hospital Amsterdam S_1

	mortality	name	age	oxygen	date diagnosed
0	1	Rose	45	95	1/4/21
1	0	Castiel	20	97	3/8/22
2	1	Jane	37	92	11/5/21

Hospital Delft S_2

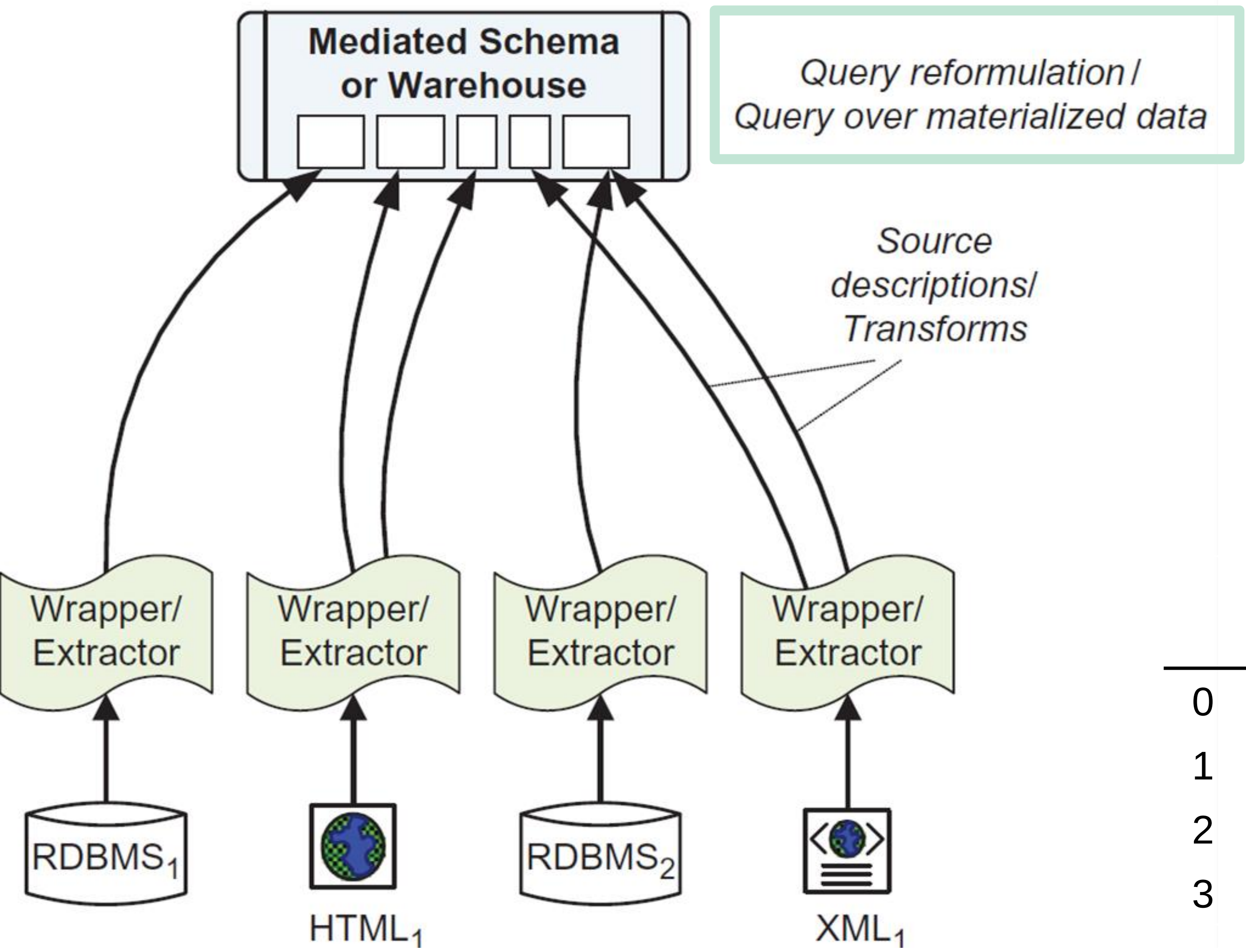
Data integration -- Query reformulation



Basic architecture of a general-purpose data integration system

Principles of Data Integration *Alon Halevy, AnHai Doan, and Zachary*

Data integration -- Query reformulation



Basic architecture of a general-purpose data integration system

Query q

```
SELECT count(*)
FROM T
WHERE oxygen<94
```

Hospital *T*

mortality	age	resting HR	oxygen

	mortality	name	age	resting HR
0	0	Jack	20	60
1	0	Sam	35	58
2	0	Ruby	22	65
3	1	Jane	37	70

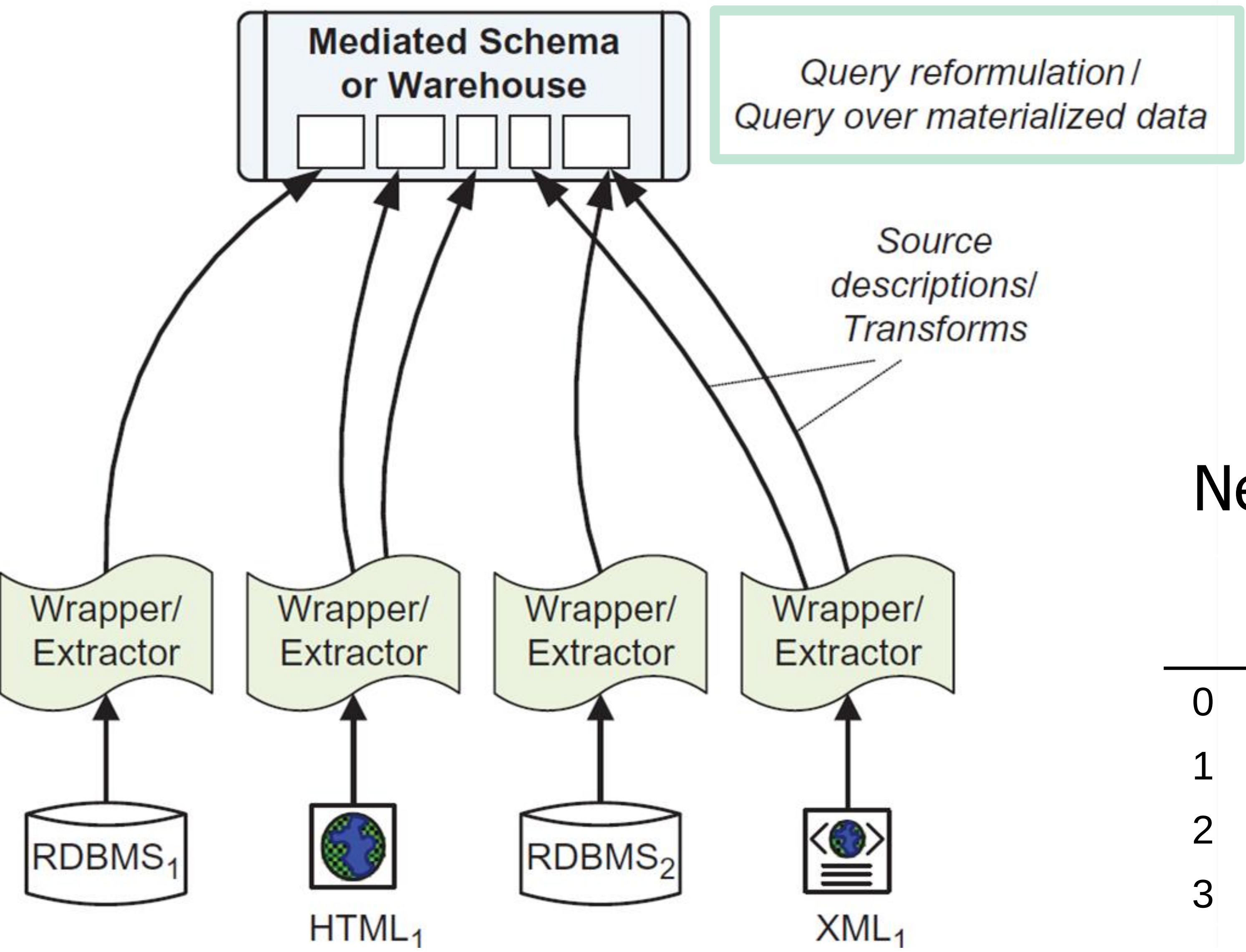
Hospital Amsterdam S_1

	mortality	name	age	oxygen	date diagnosed
0	1	Rose	45	95	1/4/21
1	0	Castiel	20	97	3/8/22
2	1	Jane	37	92	11/5/21

Hospital Delft S_2

Principles of Data Integration Alon Halevy, AnHai Doan, and Zachary

Data integration -- Query reformulation



Basic architecture of a general-purpose data integration system

Query q

```
SELECT count(*)
FROM T
WHERE oxygen<94
```

Hospital *T*

mortality	age	resting HR	oxygen

New query q_1

	mortality	name	age	resting HR
0	0	Jack	20	60
1	0	Sam	35	58
2	0	Ruby	22	65
3	1	Jane	37	70

Hospital Amsterdam S_1

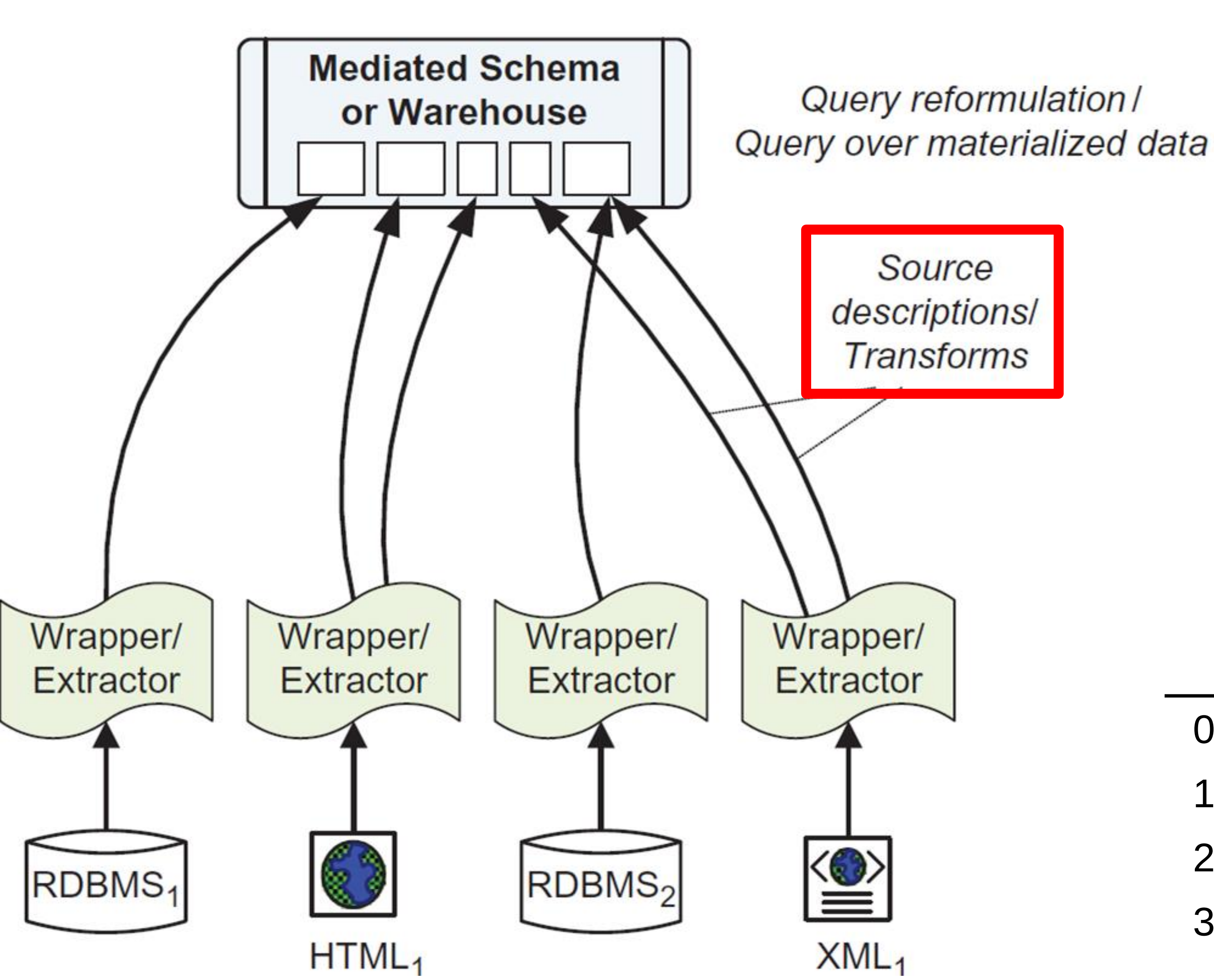
New query q_2

	mortality	name	age	oxygen	date diagnosed
0	1	Rose	45	95	1/4/21
1	0	Castiel	20	97	3/8/22
2	1	Jane	37	92	11/5/21

Hospital Delft S_2

Principles of Data Integration *Alon Halevy, AnHai Doan, and Zachary*

Data integration -- Schema mapping



Basic architecture of a general-purpose data integration system

Hospital *T*

mortality	age	resting HR	oxygen

	mortality	name	age	resting HR
0	0	Jack	20	60
1	0	Sam	35	58
2	0	Ruby	22	65
3	1	Jane	37	70

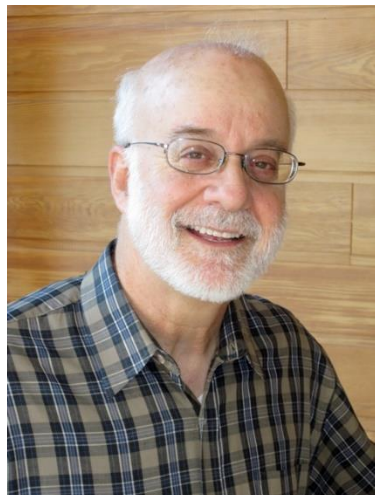
Hospital Amsterdam S_1

	mortality	name	age	oxygen	date diagnosed
0	1	Rose	45	95	1/4/21
1	0	Castiel	20	97	3/8/22
2	1	Jane	37	92	11/5/21

Hospital Delft S_2

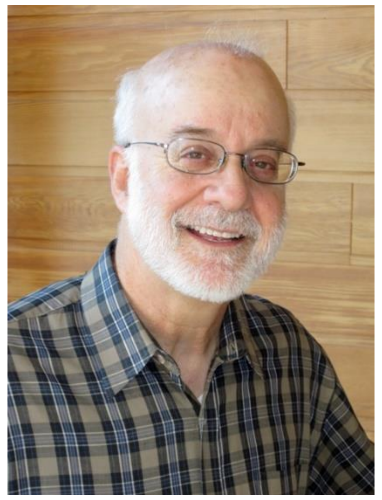
Principles of Data Integration Alon Halevy, AnHai Doan, and Zachary

Mapping Languages – First-order Logic



Ronald Fagin
IBM Fellow

Mapping Languages – First-order Logic



Ronald Fagin
IBM Fellow

“ ***Tgds*** are one of the two major types of *database dependencies* ”

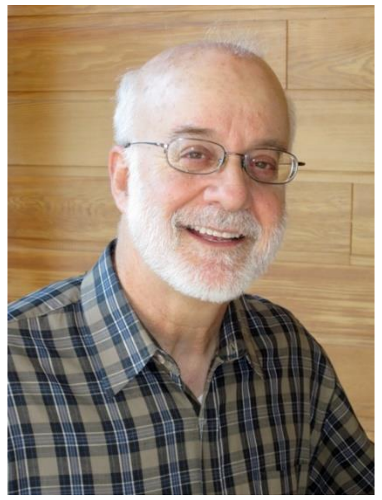
Mapping Languages – First-order Logic

Source-to-target Tuple generating dependencies (s-t tgds) [Beeri and Vardi, 1984]

Definition 3.3.1 ([Fagin, 2009]). *Tuple-generating dependencies (tgds)* are formulas of the form

$$\forall x(\varphi(\mathbf{x}) \rightarrow \exists \mathbf{y}\ \psi(\mathbf{x}, \mathbf{y})), \text{ where}$$

(1) $\varphi(\mathbf{x})$ is a conjunction of atomic formulas, all with variables among the variables in $\mathbf{x}$.
(2) every variable in x appears in $\varphi(\mathbf{x})$ (but not necessarily in $\psi(\mathbf{x}, \mathbf{y})$).
(3) $\psi(\mathbf{x}, \mathbf{y})$ is a conjunction of atomic formulas, all with variables among the variables in $\mathbf{x}$ and $\mathbf{y}$.



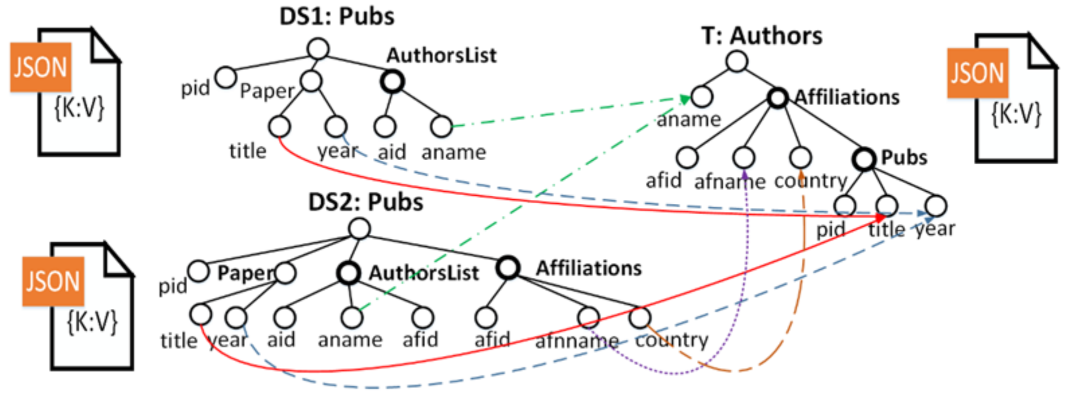
Ronald Fagin
IBM Fellow

“ ***Tgds*** are one of the two major types of *database dependencies* ”

The world of schema mapping is far beyond this…

- **More Definitions**
  - Constants & labeled nulls
  - Source and target instances
  - Homomorphism, universal solutions, canonical solution, core, certain answers
  - GAV, LAV and GLAV
  - Closed World Assumption vs. Open World Assumption
  - Term (plain term), function symbol
  - Skolemization, normalization, implication and Logical Equivalence problems
  - …
- **More People & Topics**
  - **Maurizio Lenzerini** (DI & Ontology related Rewriting)
  - **Alin Deutsch** (Query rewriting, EGDs)
  - **Lucian Popa** (Data Exchange)
  - **Phokion G. Kolaitis** (Nested Mappings)
  - **Balder Ten Cate, Wang-chiew Tan** (Data Examples)
  - **Renée Miller** (Data Exchange, Open Data Integration)
  - …

Schema mappings for nested data?



Expressive power

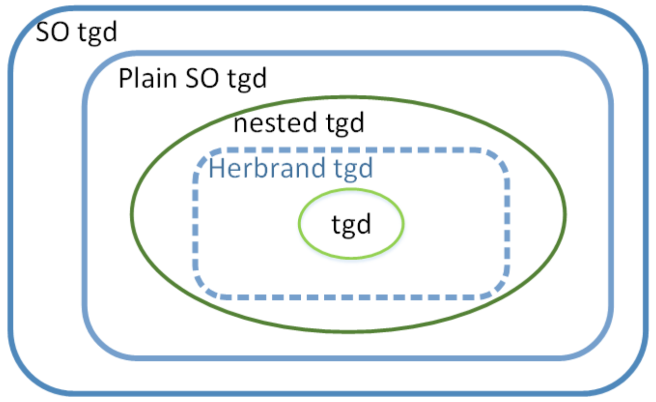
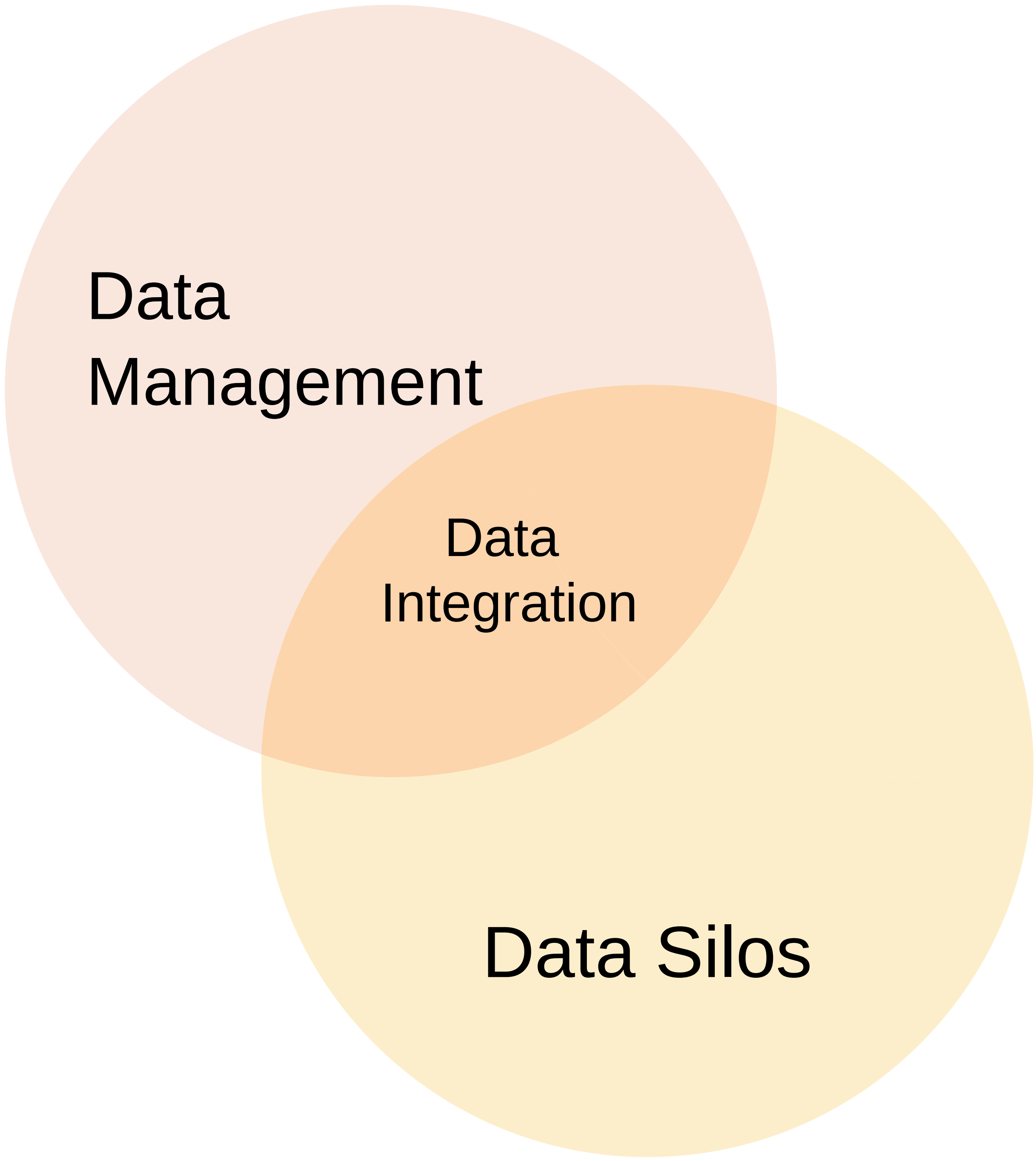


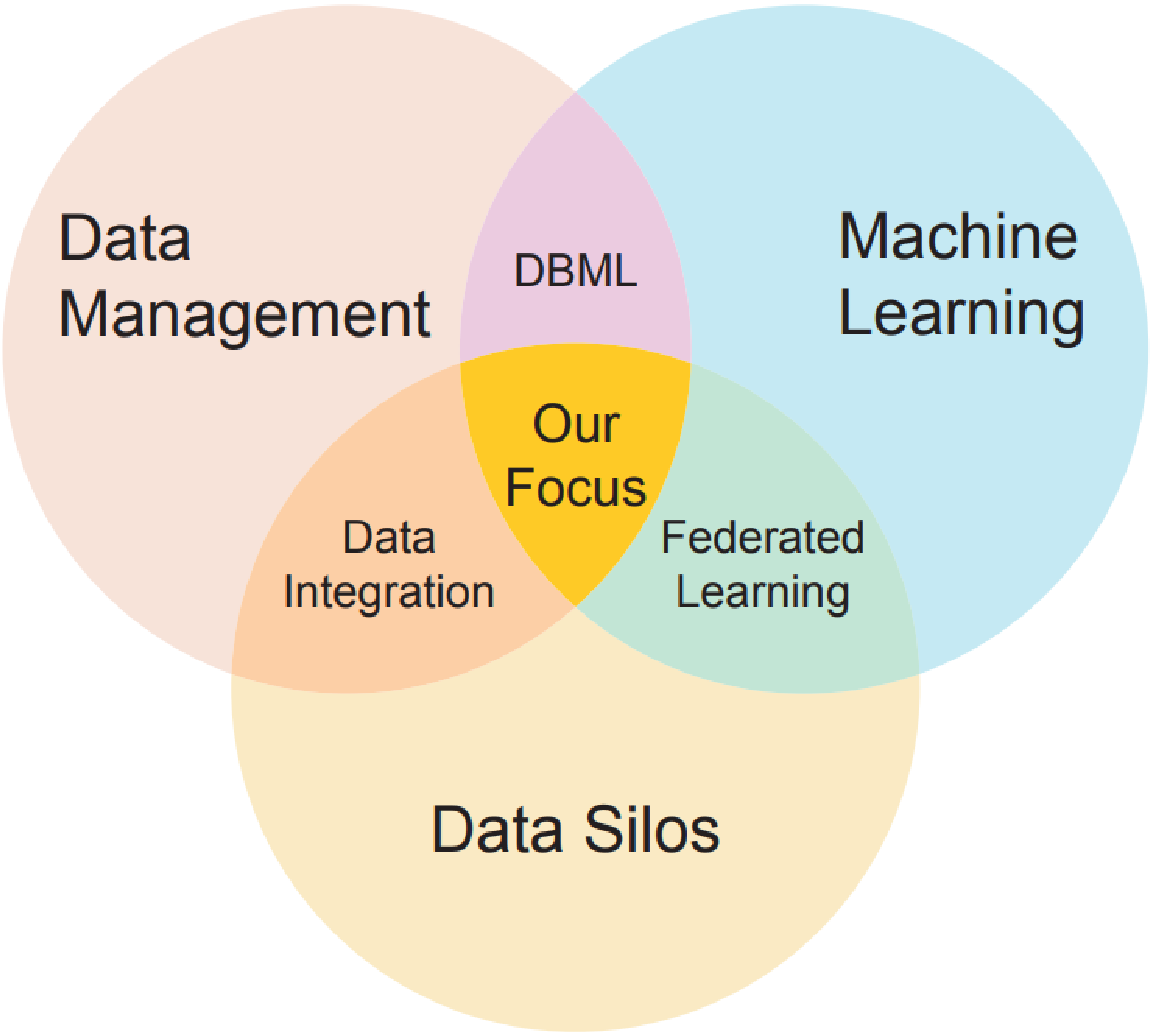
Table 3.1: Comparison of structural and reasoning properties

Structural/Reasoning properties	Tgds	Nested tgds	Plain SO tgds	SO tgds
Closure under target homomorphisms	Yes	Yes	Yes	No
Admitting universal solutions	Yes	Yes	Yes	Yes
Allowing for conjunctive query rewriting	Yes	Yes	Yes	Yes
Decidable for implication problem	Yes	Yes	Unknown	No
Decidable for logical equivalence problem	Yes	Yes	Unknown	No

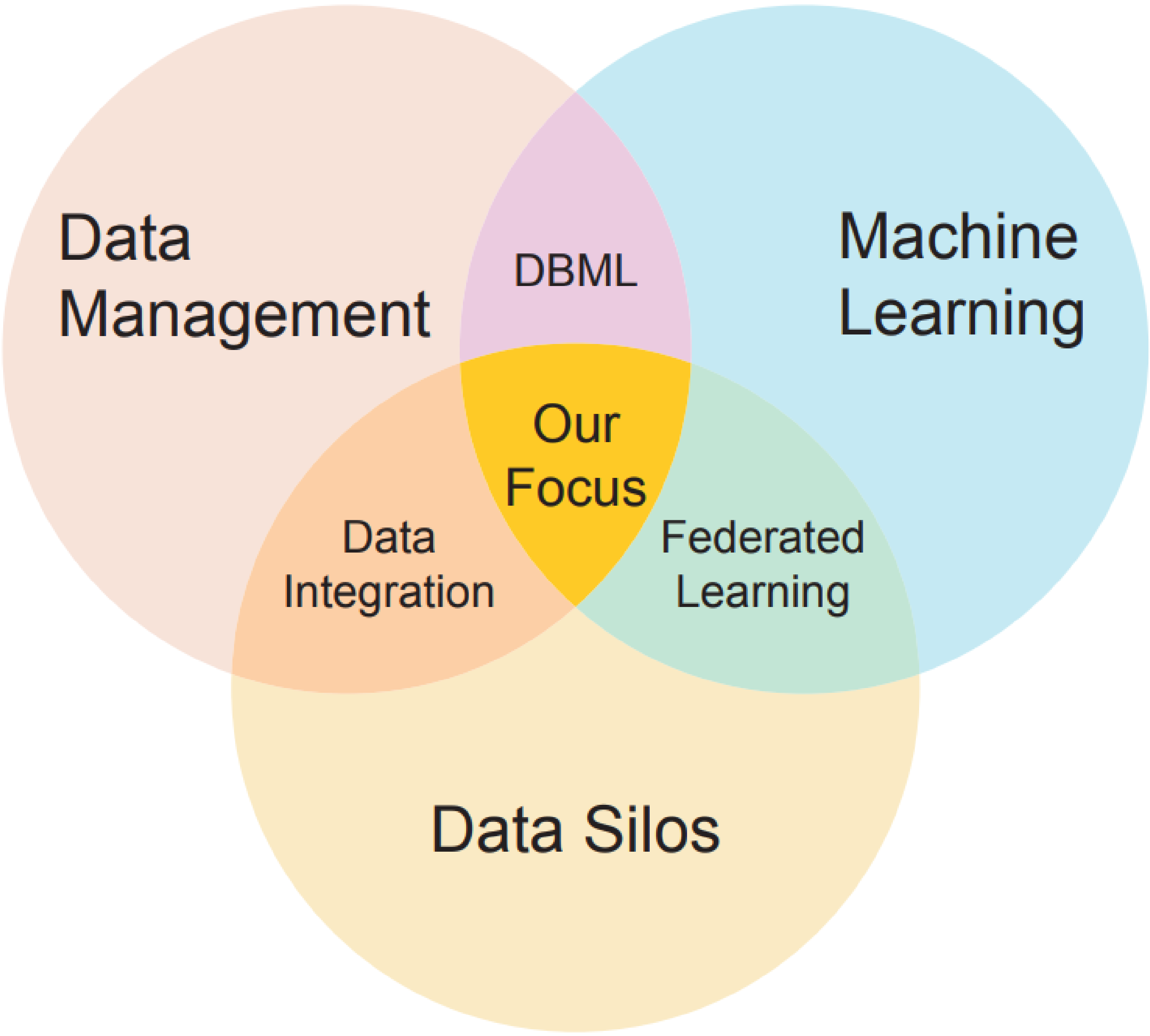
Adding Machine Learning



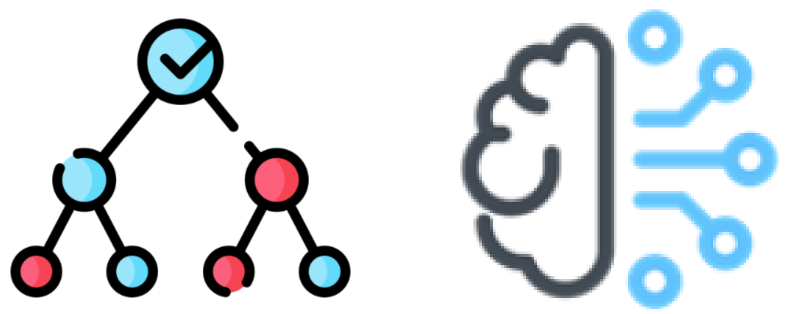
Adding Machine Learning



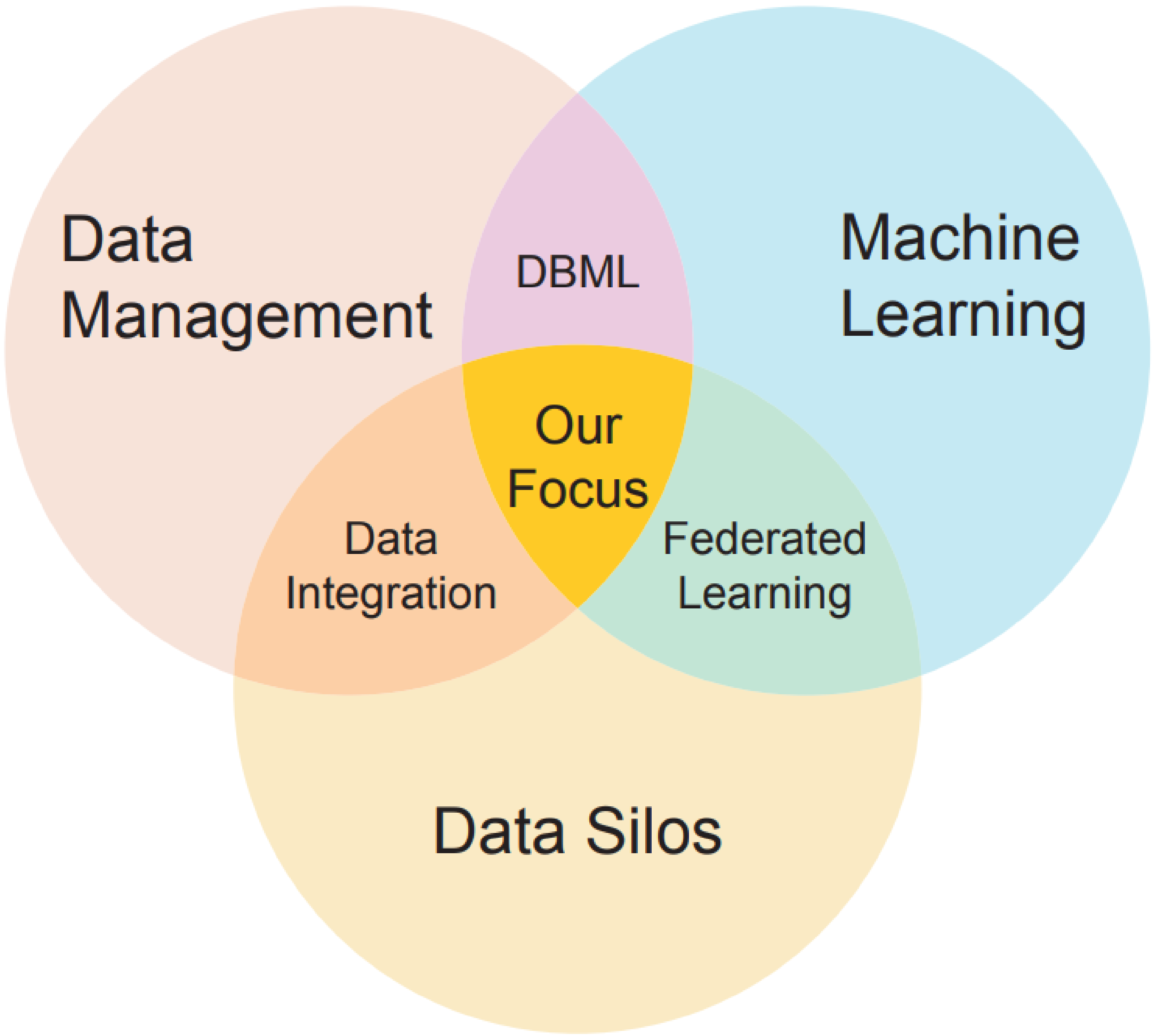
Adding Machine Learning



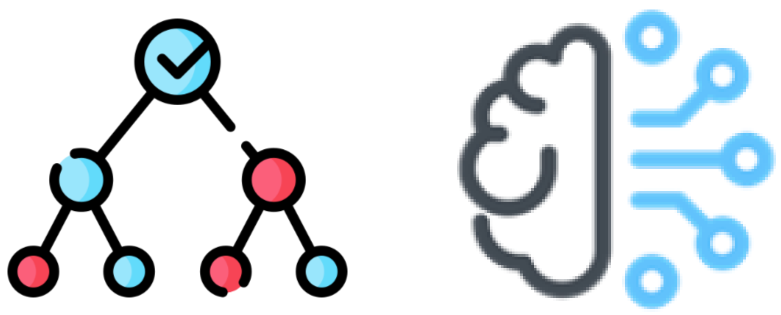
ML models

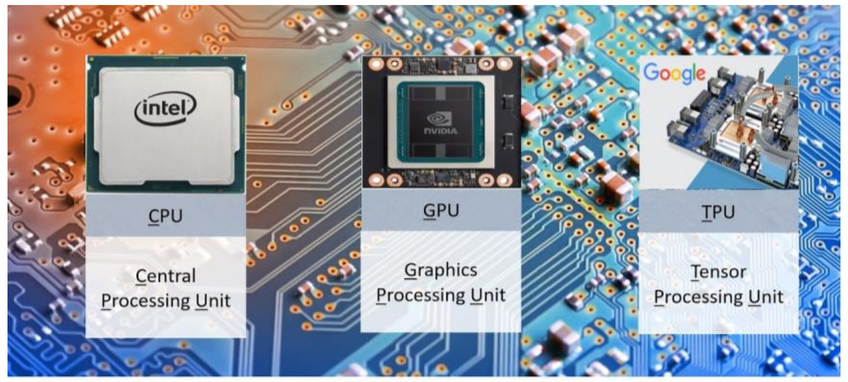


Adding Machine Learning



ML models





New hardware

Represent **dataset relationships in ML use cases**

Step.1: Formal description of ML use cases with database dependencies

"***Tgds*** are one of the two major types of *database dependencies*"

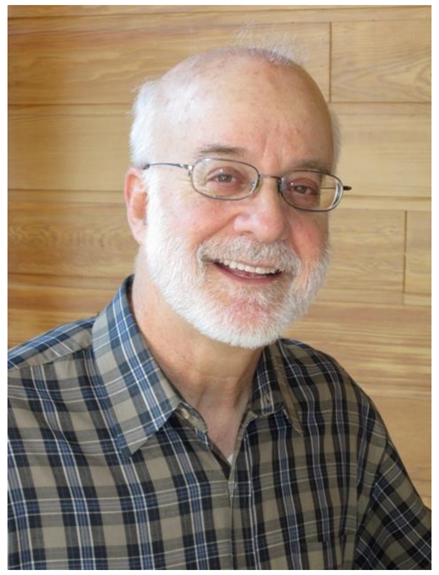


Table 1: Four example data integration scenarios for feature augmentation and federated learning

No.	Dataset Relationship	Schema mappings	Example use cases
1	Full outer join	$m_1: \forall m,n,a,hr,o,dd\ (S_1(m,n,a,hr) \wedge S_2(m,n,a,o,dd) \rightarrow T(m,a,hr,o))$ $m_2: \forall m,n,a,hr\ (S_1(m,n,a,hr) \rightarrow \exists o\ T(m,a,hr,o))$ $m_3: \forall m,n,a,o,dd\ (S_2(m,n,a,o,dd) \rightarrow \exists hr\ T(m,a,hr,o))$	Feature augmentation, Federated learning, …
2	Inner join	$m_1: \forall m,n,a,hr,o,dd\ (S_1(m,n,a,hr) \wedge S_2(m,n,a,o,dd) \rightarrow T(m,a,hr,o))$	Feature augmentation, (Vertical) federated learning, …
3	Left join	$m_1: \forall m,n,a,hr,o,dd\ (S_1(m,n,a,hr) \wedge S_2(n,a,o,dd) \rightarrow T(m,a,hr,o))$ $m_2: \forall m,n,a,hr\ (S_1(m,n,a,hr) \rightarrow \exists o\ T(m,a,hr,o))$	Feature augmentation, (Vertical) federated learning, …
4	Union	$m_2: \forall m,n,a,hr,o\ (S_1(m,n,a,hr,o) \rightarrow T(m,a,hr,o))$ $m_3: \forall m,n,a,hr,o,dd\ (S_2(m,n,a,hr,o,dd) \rightarrow T(m,a,hr,o))$	Data sample augmentation, (Horizontal) federated learning, …

Transform dataset relationships in ML use cases

Step.1: Formal description of ML use cases with database dependencies

Step.2: Transform metadata into matrices

Column matching (schema mapping)

$$m_1 : \forall m, n, a, hr, o, dd \ (S_1(m, n, a, hr) \wedge S_2(m, n, a, o, dd) \ \longrightarrow T(m, a, hr, o))$$

$$\boldsymbol{M_1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{M_2} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(a) Mapping matrix

Row matching (entity resolution)

S_1	3	1	Jane	37	70	
S_2	2	1	Jane	37	92	11/5/21

$$\boldsymbol{I_1} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{I_2} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

(b) Indicator matrix

Scope

Theory

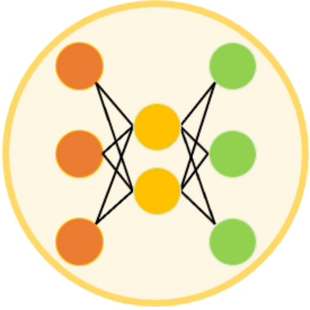
Systems

Amalur: Model Lake

Source: https://www.behance.net/gallery/38395965/Mother-Earth

Web Information Systems (WIS) group

Data Lake



Model Zoo

arXiv > cs > arXiv:2106.09592

Computer Science > Databases

[Submitted on 17 Jun 2021 (v1), last revised 17 Feb 2023 (this version, v2)]

Data Lakes: A Survey of Functions and Systems

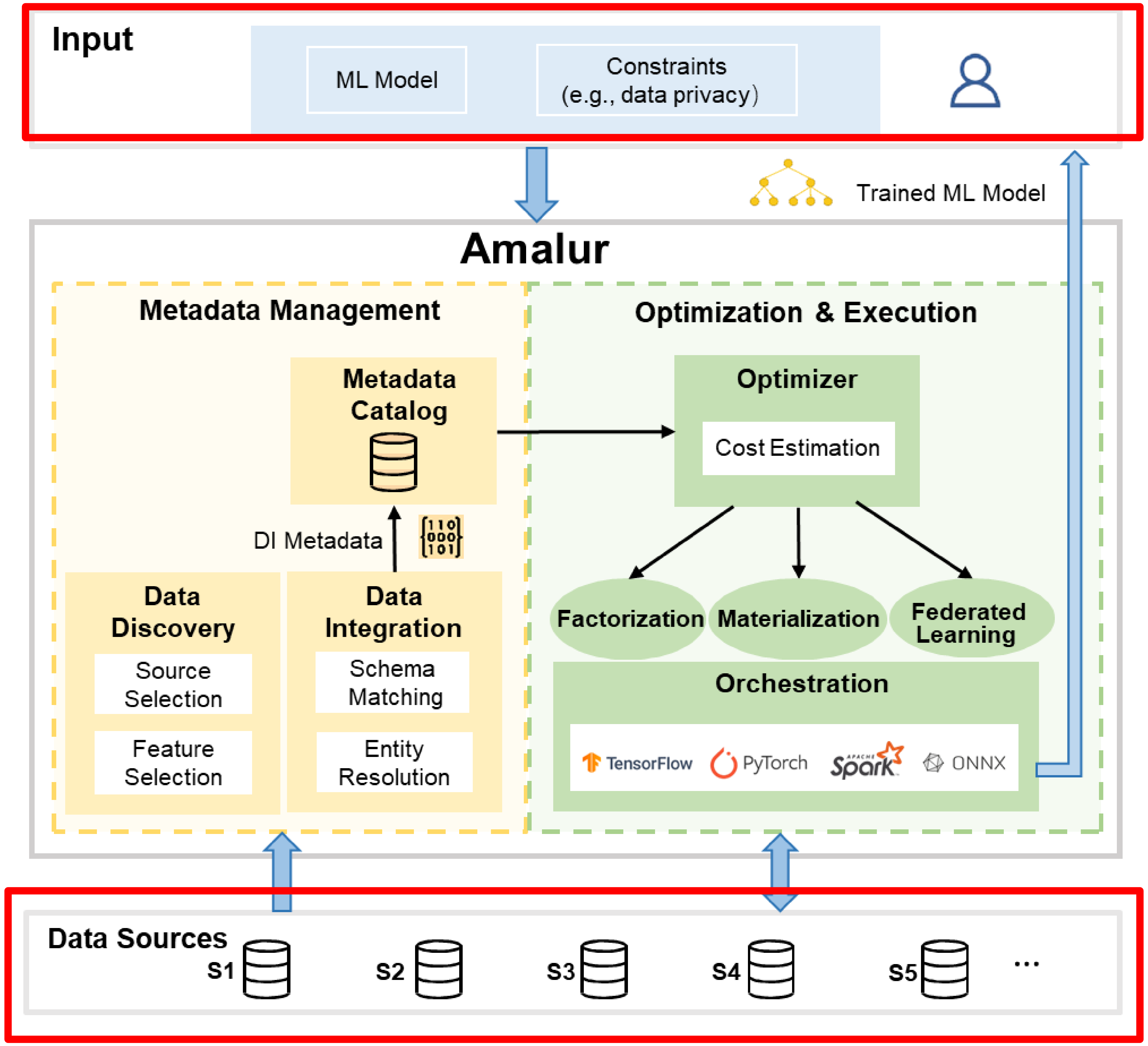
Rihan Hai, Christos Koutras, Christoph Quix, Matthias Jarke

Macaroni: Crawling & Enriching Metadata from Public Model Zoos

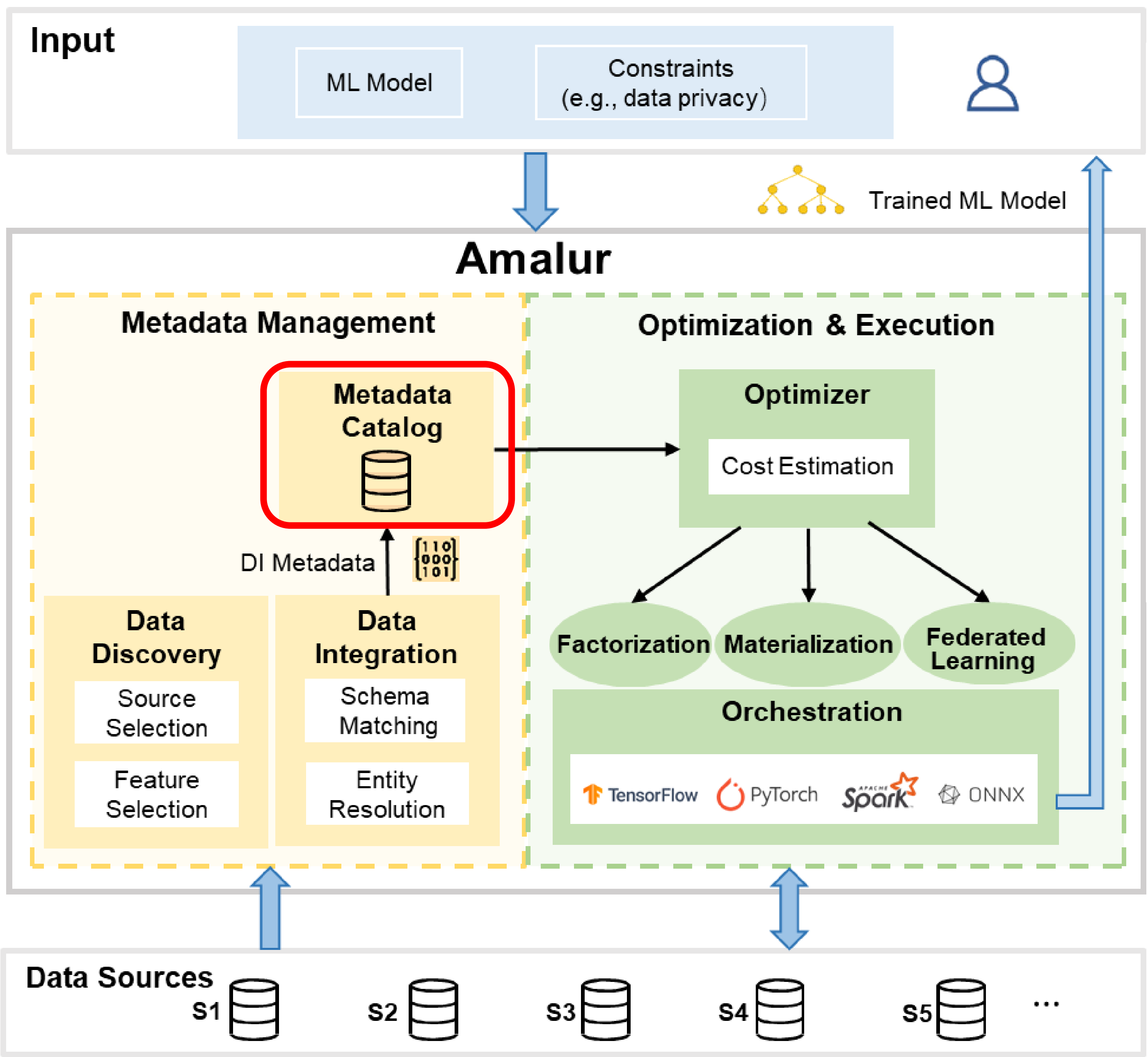
Ziyu Li, Henk Kant, Rihan Hai, Asterios Katsifodimos, and Alessandro Bozzon

Delft University of Technology

Amalur: Model Lake



Amalur: Workflow



Column matching (schema mapping)

$$m_1: \forall m,n,a,hr,o,dd\ (S_1(m,n,a,hr) \wedge S_2(m,n,a,o,dd) \longrightarrow T(m,a,hr,o))$$

$$M_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \qquad M_2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(a) Mapping matrix

Row matching (entity resolution)

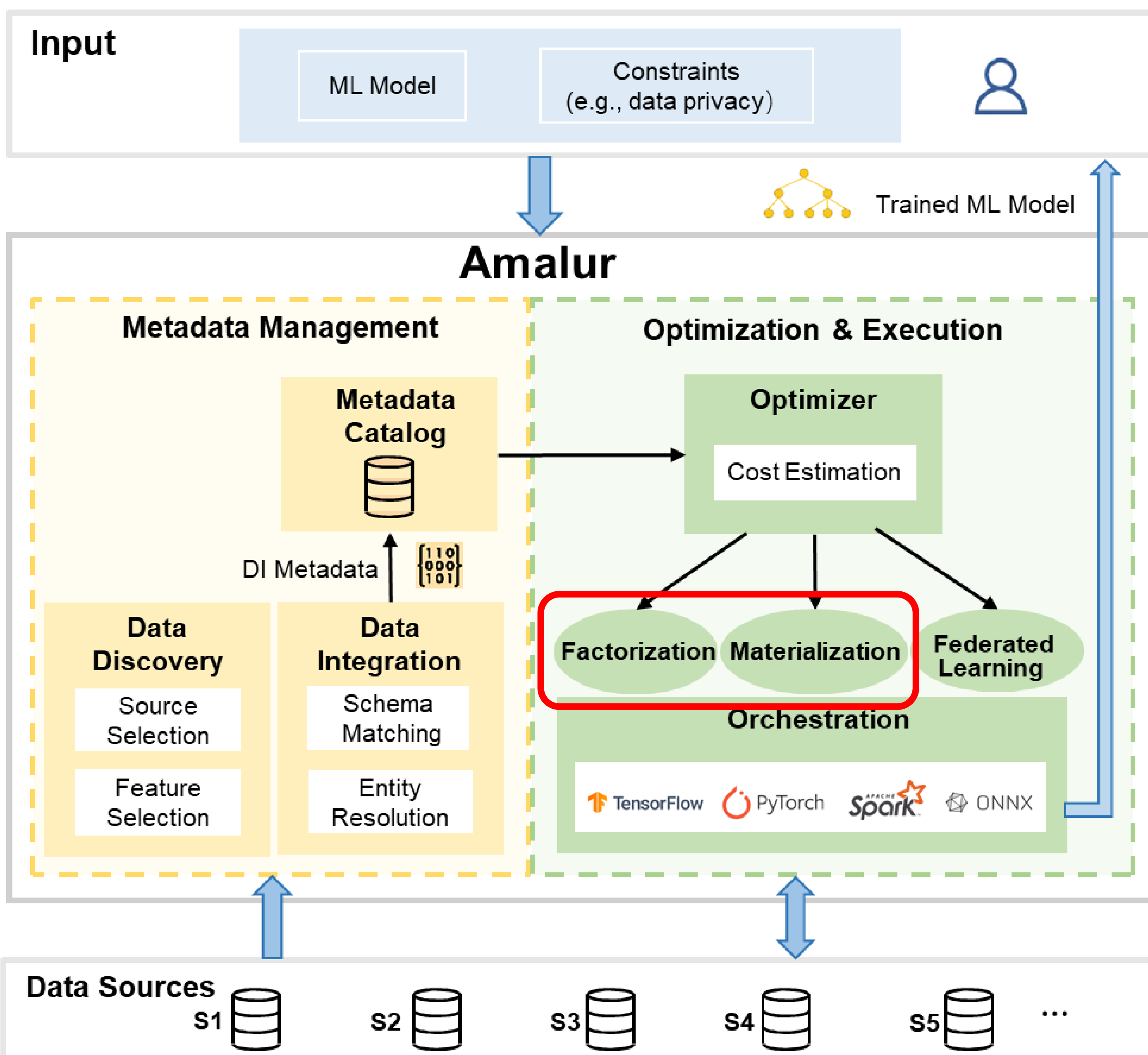
S_1	3	1	Jane	37	70	
S_2	2	1	Jane	37	92	11/5/21

$$I_1 \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad I_2 \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

(b) Indicator matrix

Representing DI metadata as tensors

Amalur: Workflow



Column matching (schema mapping)

$m_1 : \forall m, n, a, hr, o, dd\ (S_1(m, n, a, hr) \land S_2(m, n, a, o, dd) \longrightarrow T(m, a, hr, o))$

$$M_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad M_2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(a) Mapping matrix

Row matching (entity resolution)

S_1	3	1	Jane	37	70	
S_2	2	1	Jane	37	92	11/5/21

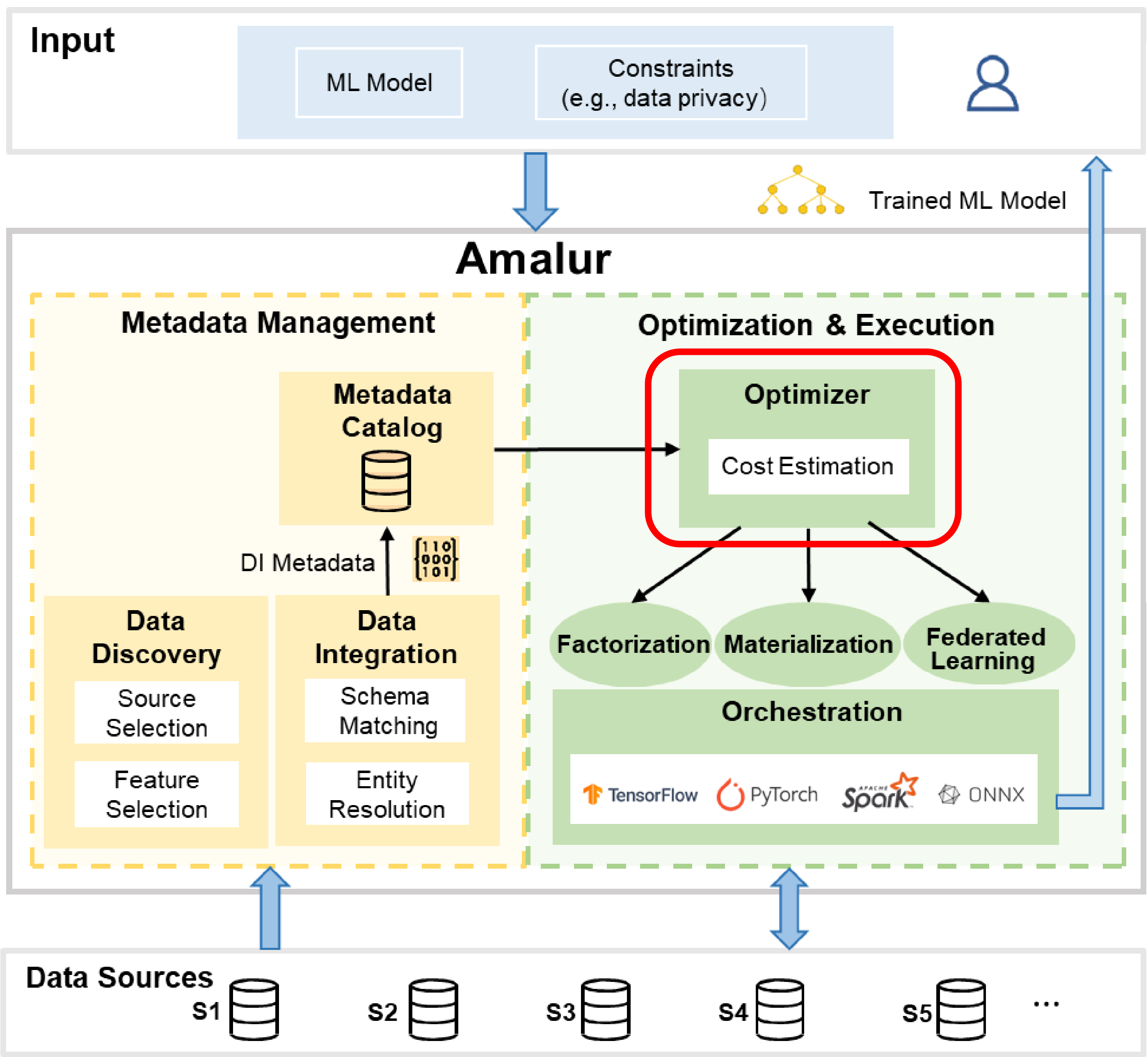
$$I_1 \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad I_2 \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

(b) Indicator matrix

Factorize: $TX \rightarrow T_1X + (T_2 \circ R_2)X$

Materialize: $T \rightarrow I_1D_1M_1^T + (I_2D_2M_2^T \circ R_2)$

Amalur: Workflow



Column matching (schema mapping)

$m_1: \forall m,n,a,hr,o,dd\ (S_1(m,n,a,hr) \wedge S_2(m,n,a,o,dd) \longrightarrow T(m,a,hr,o))$

$$M_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad M_2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(a) Mapping matrix

Row matching (entity resolution)

S_1	3	1	Jane	37	70	
S_2	2	1	Jane	37	92	11/5/21

$$I_1 \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad I_2 \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

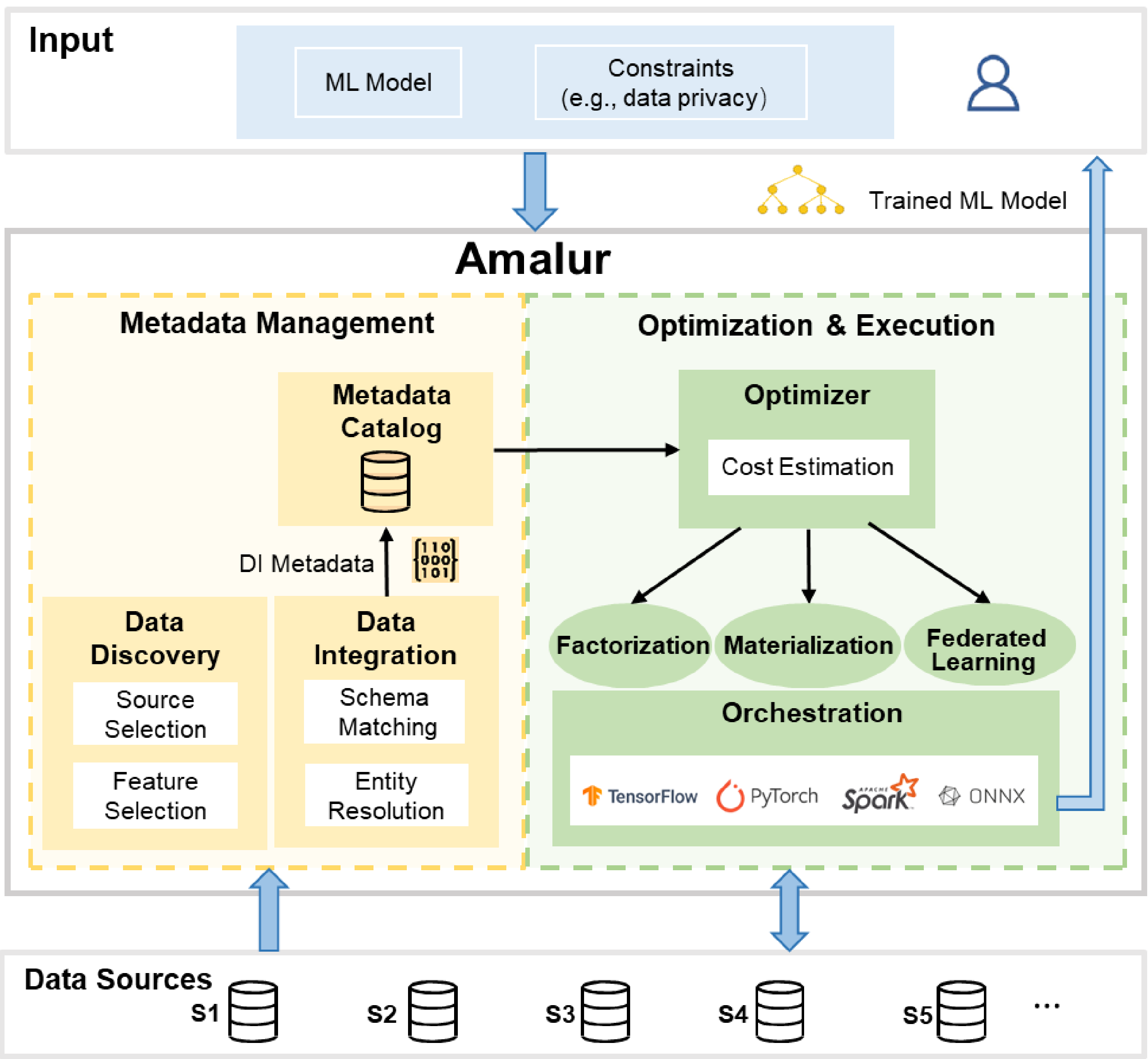
(b) Indicator matrix

Factorize: $TX \rightarrow T_1X + (T_2 \circ R_2)X$

Materialize: $T \rightarrow I_1D_1M_1^T + (I_2D_2M_2^T \circ R_2)$

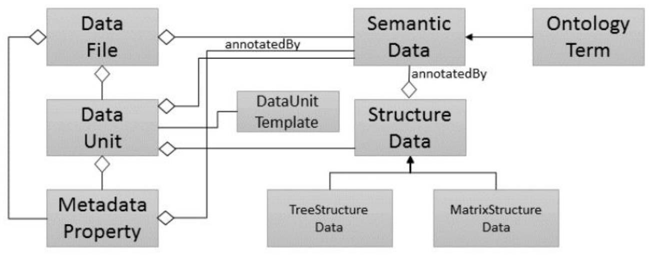
The whole picture

Model Lake Amalur

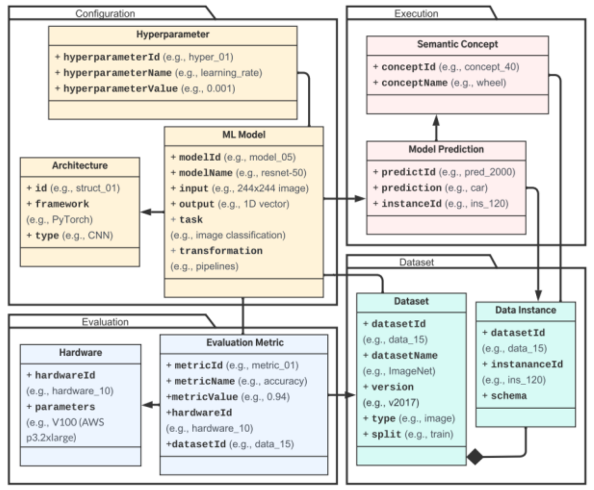


Metadata Representations

Data



ML model



Conceptual

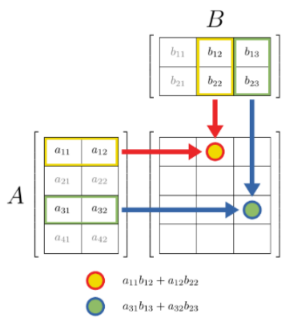
FO logic

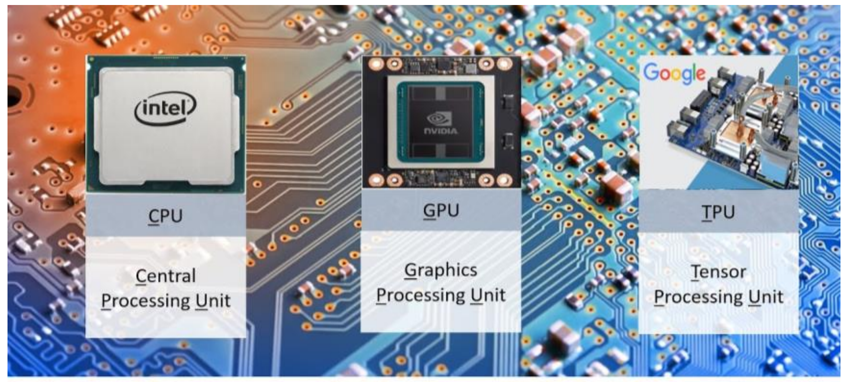
$$m_1:\ \forall s,n,se,a,p,d\ (S_1(s,n,se,a) \land S_2(s,n,se,p,d) \rightarrow T(s,se,a,p))$$

Matrices

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Logical





Physical

Vision

Data Integration meets Machine Learning

Q: Can we use data integration metadata to improve the effectiveness and efficiency of ML model training?

Data Integration

Data Lakes

Data Discovery

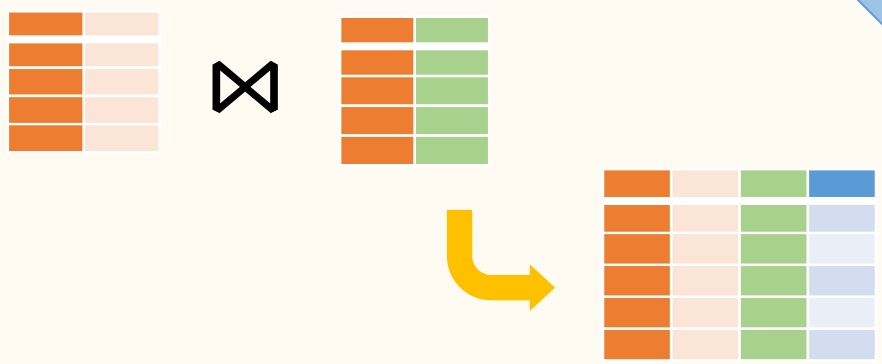
Data Cleaning

...

Dataset relationships as First-order Logic

Data dependencies:

$\forall a, b, c\, (S_1(a,b) \land S_2(a,c) \rightarrow \exists d\; T(a,b,c,d))$

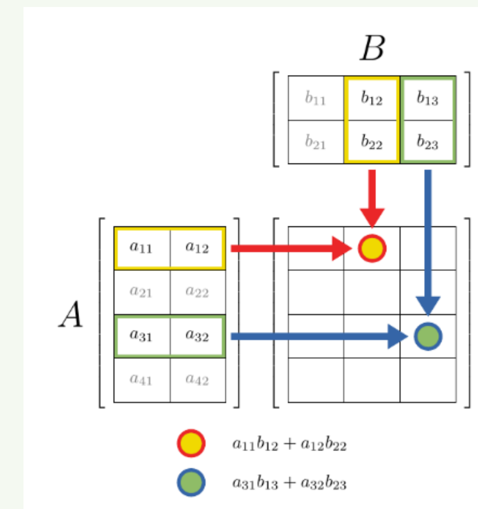


?

ML models & Linear Algebra

Matrix multiplication:

AB



Federated Learning

Transfer Learning

Embeddings

Transformers

...

We’re hiring
-- Work, live, love at Delft

- **DBML, data lakes, and federated learning**
  - PhD students
  - postdoctoral researcher

Contact: R.Hai@tudelft.nl

- **Scalable & Consistent Function-as-a-Service Systems**
  - 2 PhD students
  - 1 postdoctoral researcher

Contact: a.katsifodimos@tudelft.nl