VADALOG

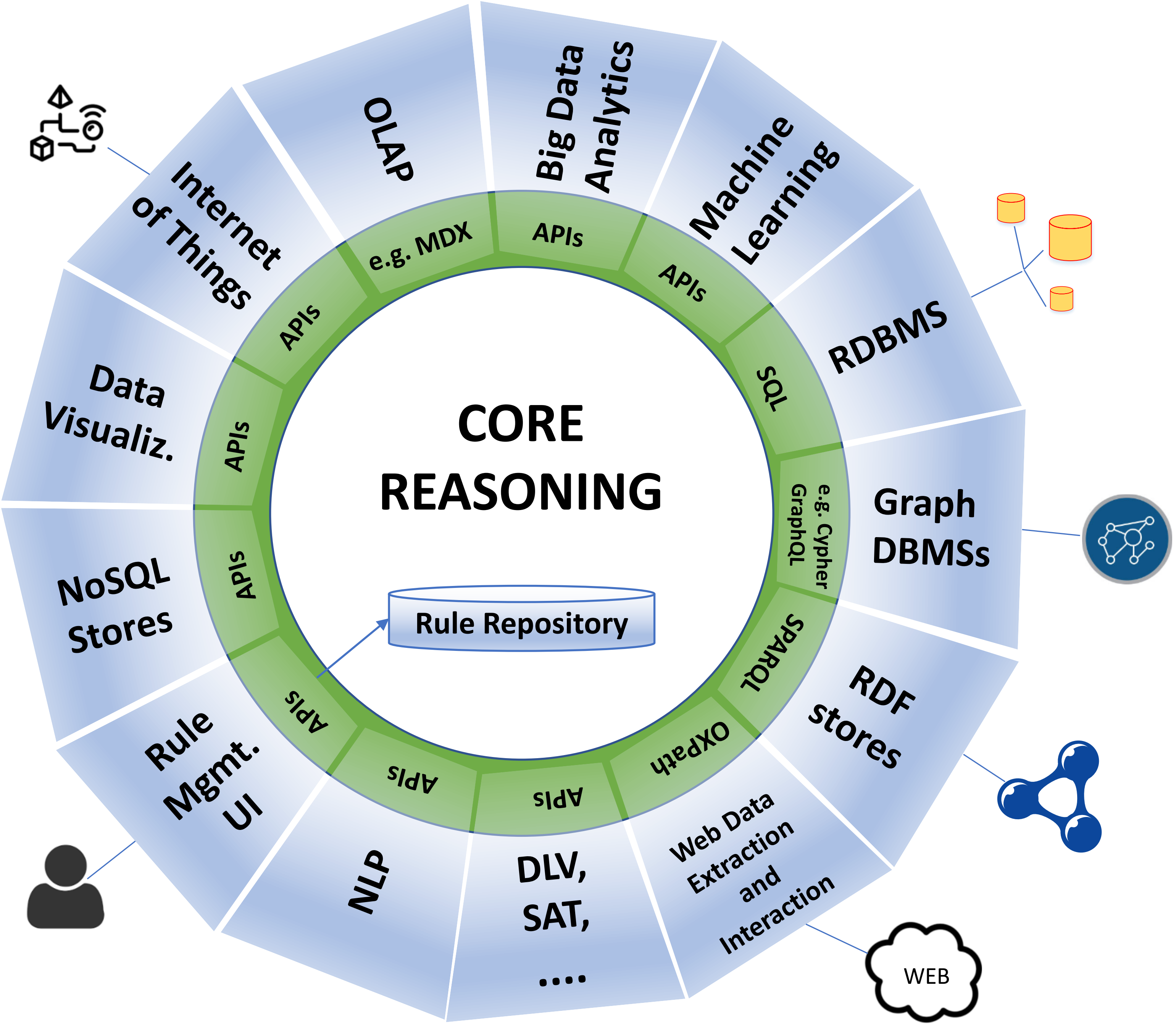
A Swift Logic for Big Data and a System Combining Datalog Reasoning with Machine Learning

Georg Gottlob

Univ. of Oxford and TU Wien

VADALOG

DeepReason.ai



CORE REASONING

Rule Repository

OLAP

e.g. MDX

Big Data Analytics

APIs

Machine Learning

APIs

RDBMS

SQL

Graph DBMSs

e.g. Cypher GraphQL

RDF stores

SPARQL

Web Data Extraction and Interaction

OXPath

WEB

DLV, SAT, ...

APIs

NLP

APIs

Rule Mgmt. UI

APIs

NoSQL Stores

APIs

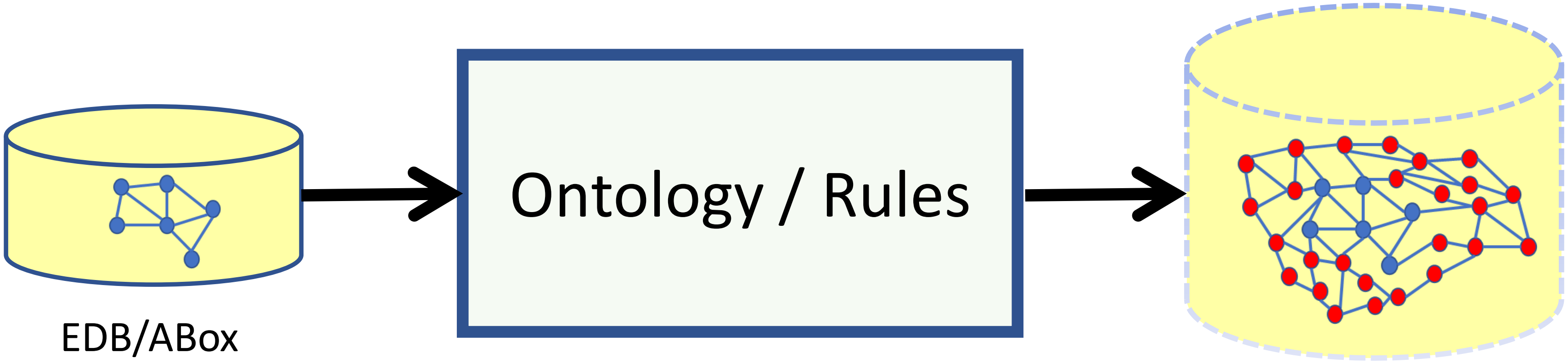
Data Visualiz.

APIs

Internet of Things

APIs

Reasoning in Knowledge Graphs



EDB/ABox

Ontology / Rules

EDB+IDB

Many still think that DLs or graph databases suffice. However:

Reasoning tasks are required that cannot be expressed by description logics, and cannot be reasonably managed by relational DBMS, nor by graph DBMS.

Example: Wikidata Marriage Intervals

[Krötzsch DL 2017]

Wikidata contains the statement :

Taylor was married to Burton starting from 1964 and ending 1974

This can be represented in relational DB or Datalog-notation by :

```
married(taylor,burton,1964,1974)
```

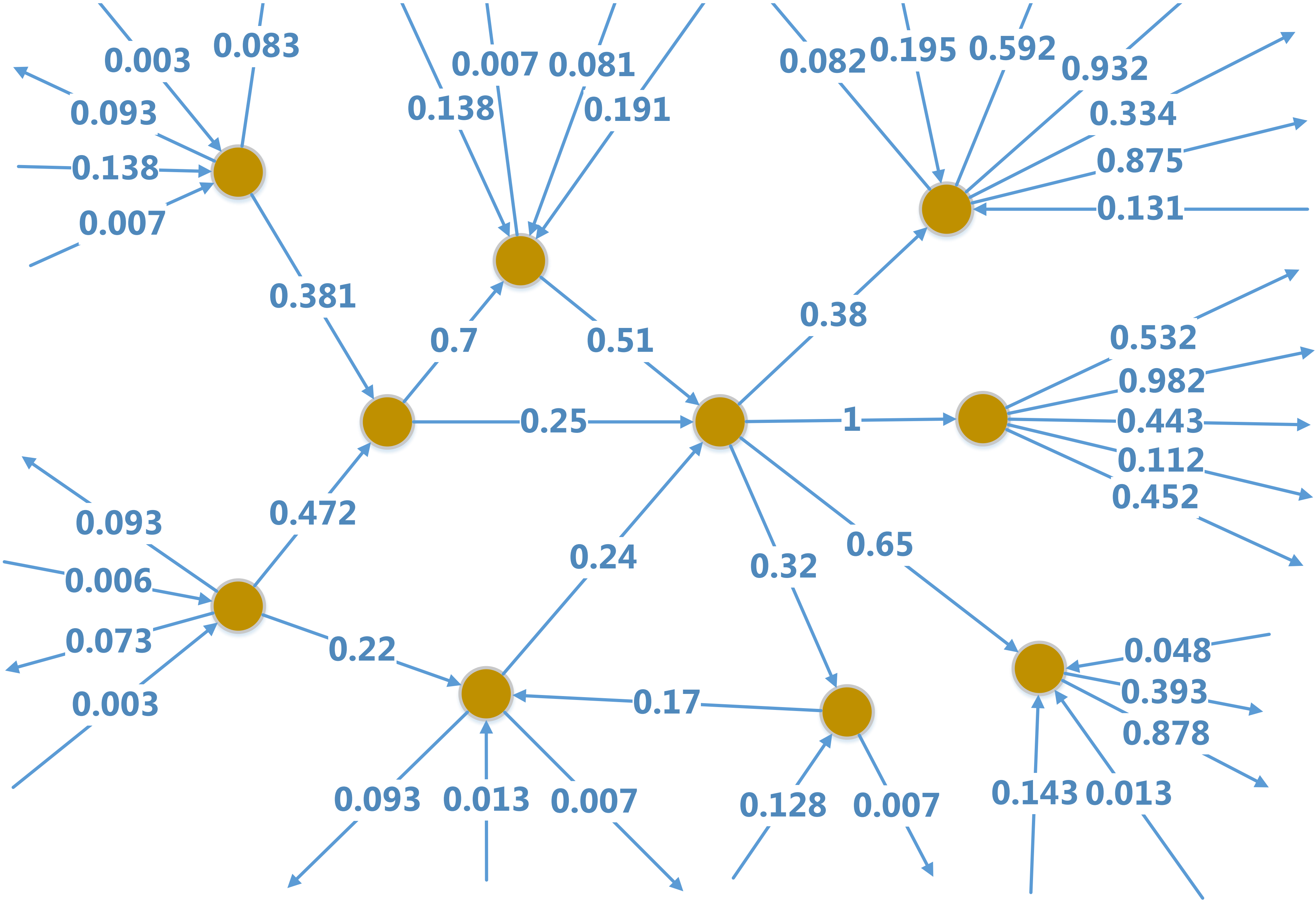
Symmetry rule for marriage intervals in Datalog:

```
∀ u,v,x,y. married(u,v,x,y) → married(v,u,x,y)
```

This cannot be expressed in DLs!

Note: In what follows, we will often omit universal quantifiers.

Example: Controlling Companies



Example: Controlling Companies

x controls y if

x directly holds over 50% of y, or

x controls a set of companies that jointly hold over 50% of y

```
                                    company(x) → control(x,x),
control(x,y), own(y,z,w), v=msum(w,⟨y⟩), v>0.5 → control(x,z).
```

This cannot be expressed in DLs and only clumsily in SQL and Graph DBMS!

Example: My Creditworthiness

Example: My Creditworthiness

up to £10,000

£8,500

£12,000

up to EUR 10,000

up to EUR 20,000

£500

£ 8,000

£ 12,500

EUR 14,000

My Explanation

A machine-learning program has “reasonably” learned:

> ***People who live in a joint household with someone who does not pay their bills are likely to fail repaying their own debts.***

This ethically questionable rule was applied to incomplete and wrong data:

- Before I bought the house there was a tenant who indeed did not pay his bills (*tons of unpaid bills & overdue notices in my mailbox*).
- The tenant had moved out before I moved in, but the Credit Rating Agency did not know, and simply assumed he still lived there.

ML should be complemented and, where necessary, overruled by domain-specific “expert rules” that express domain knowledge:

- **A new house-owner is most likely unrelated to a previous tenant.**
- **If a house is bought by somebody who did not live there previously, and now lives there, then the previous occupiers have most likely moved out. [→Verify!]**
- **If someone has closed their bank account without opening a new one then it is likely that the person has moved out of the country.**

...

Automatically accessing outside sources such as the Land Register and/or Social Networks may help.

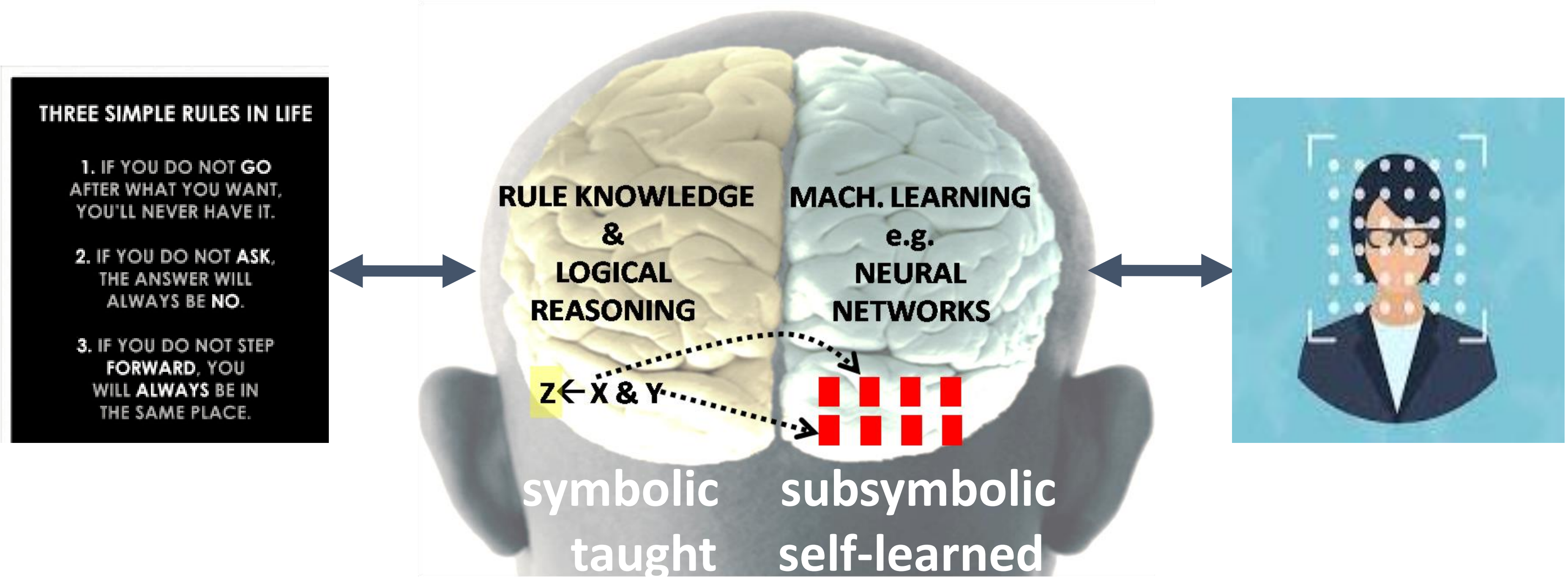
→ Data extraction from external sources is a requirement for KGMS.

Knowledge Graph Management Systems

KGMS combine the power of rule-based reasoning with machine learning over Big Data:

KGMS = KBMS + Big Data + Analytics

Misusing the lateralization thesis for illustration

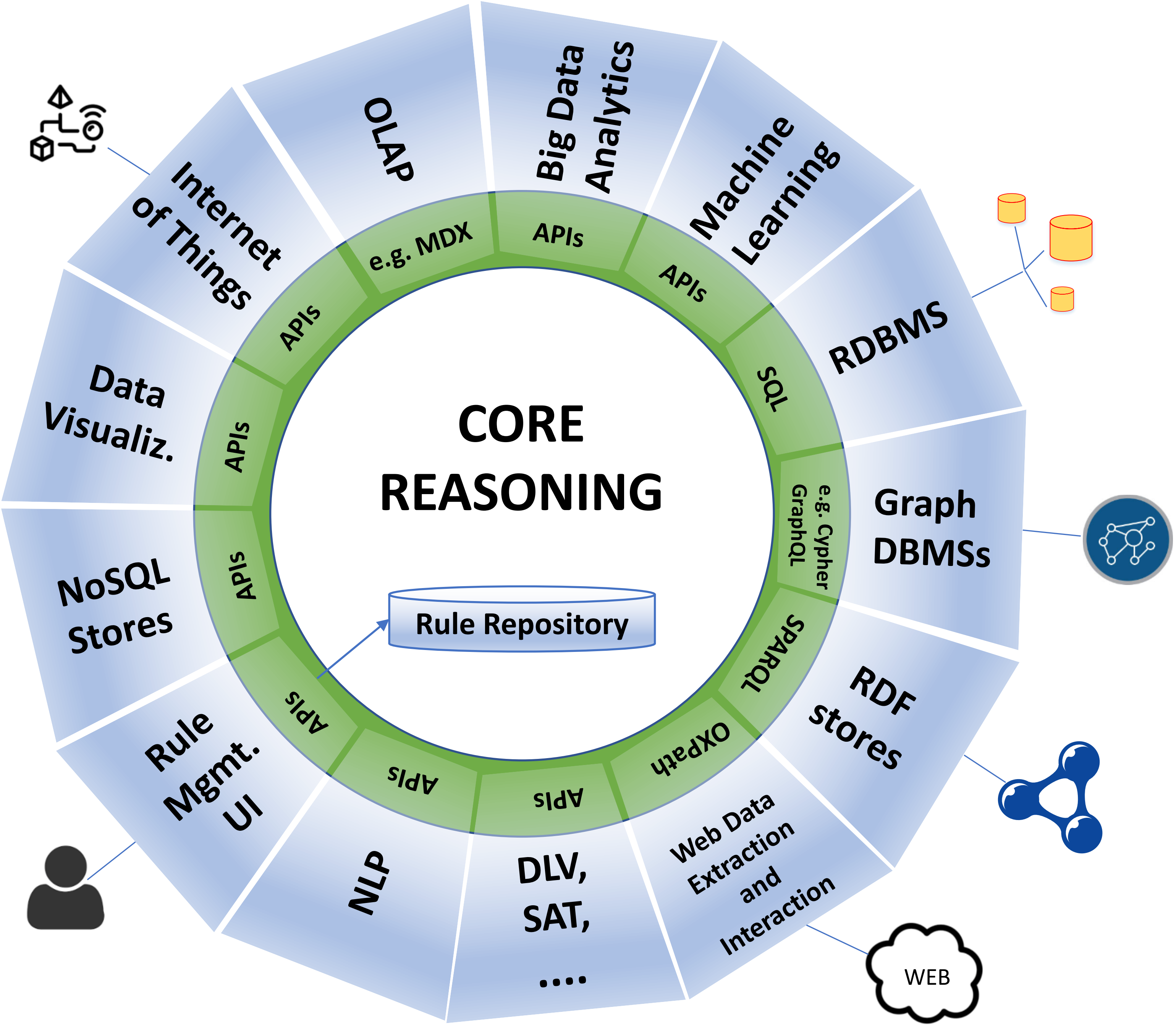


THREE SIMPLE RULES IN LIFE

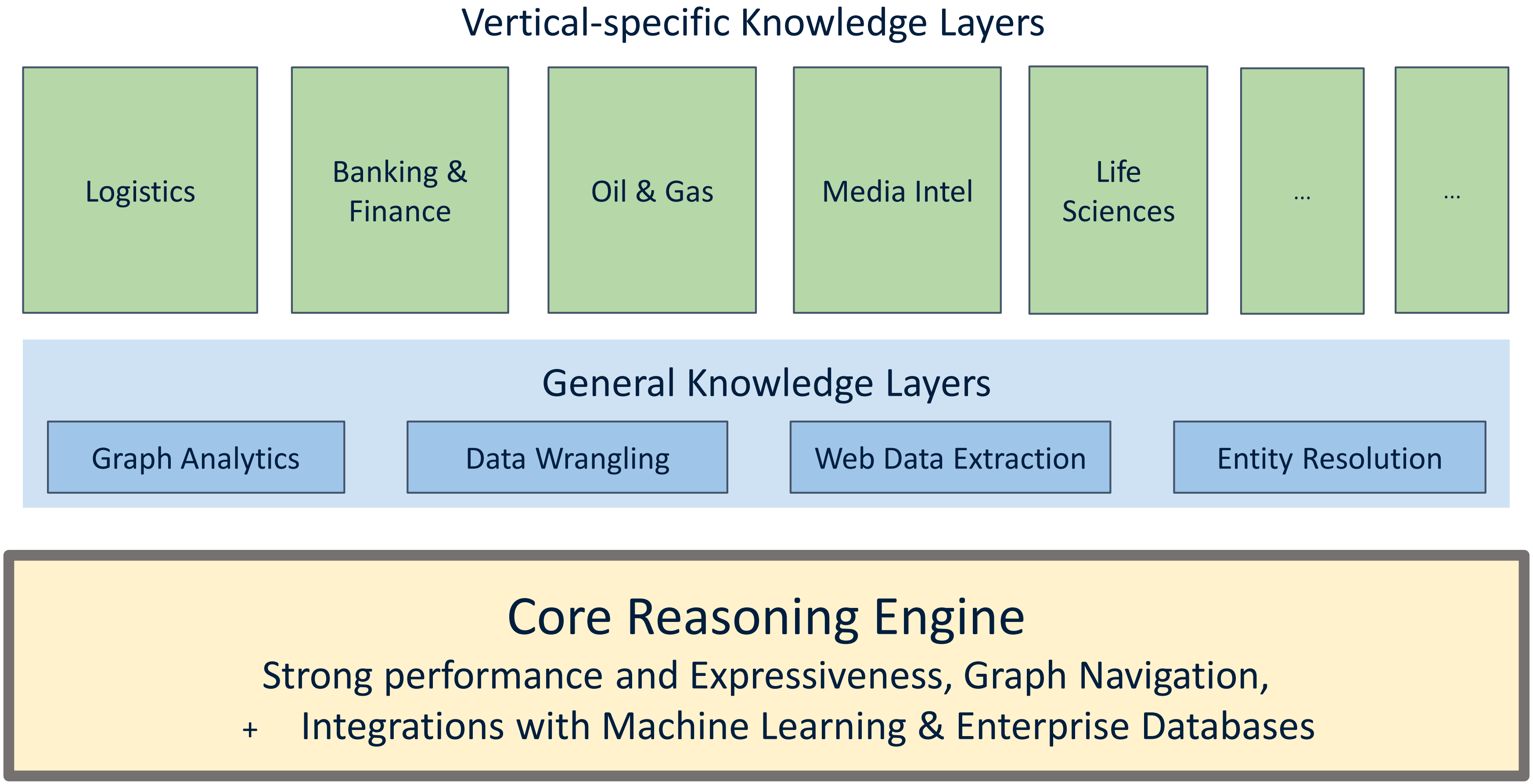
1. IF YOU DO NOT GO AFTER WHAT YOU WANT, YOU'LL NEVER HAVE IT.

2. IF YOU DO NOT ASK, THE ANSWER WILL ALWAYS BE NO.

3. IF YOU DO NOT STEP FORWARD, YOU WILL ALWAYS BE IN THE SAME PLACE.


CORE
REASONING
Rule Repository
OLAP
e.g. MDX
Big Data Analytics
APIs
Machine Learning
APIs
RDBMS
SQL
Graph DBMSs
e.g. Cypher GraphQL
RDF stores
SPARQL
Web Data Extraction and Interaction
OXPath
WEB
DLV, SAT, ...
APIs
NLP
APIs
Rule Mgmt. UI
APIs
NoSQL Stores
APIs
Data Visualiz.
APIs
Internet of Things
APIs

Knowledge Layers



Vertical-specific Knowledge Layers

Logistics

Banking & Finance

Oil & Gas

Media Intel

Life Sciences

…

…

General Knowledge Layers

Graph Analytics

Data Wrangling

Web Data Extraction

Entity Resolution

Core Reasoning Engine

Strong performance and Expressiveness, Graph Navigation,

+ Integrations with Machine Learning & Enterprise Databases

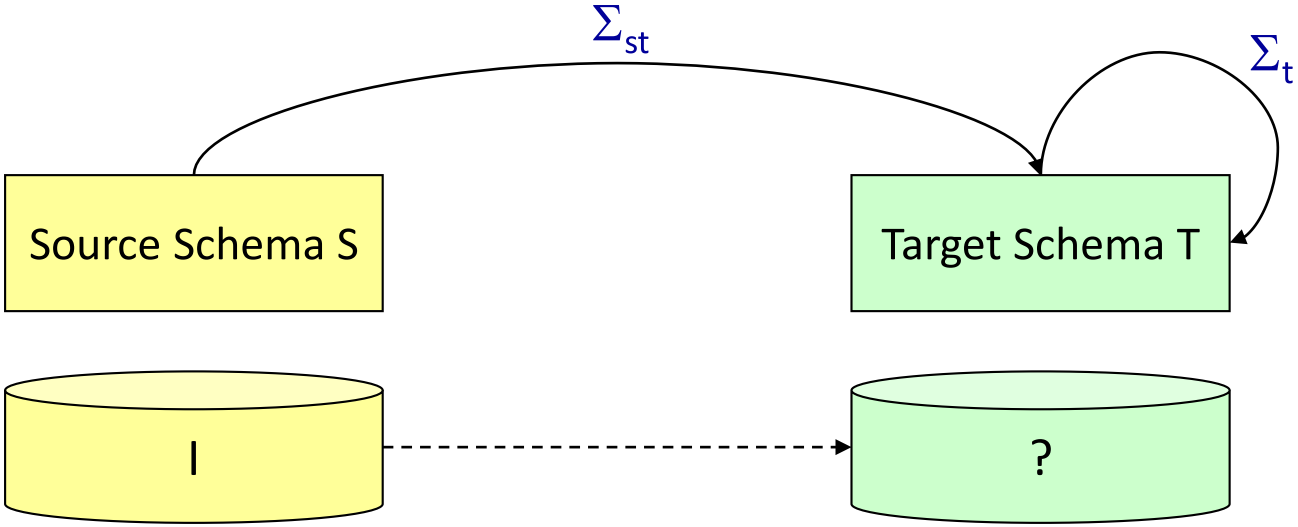
Vadalog: The Core Reasoning Language

Core Vadalog = full Datalog + restricted use of ∃ + stratif. negation + ⊥

Why existential quantifiers in rule heads?

- Data exchange, data integration
- Data extraction
- Reasoning with RDF → Wikidata example
- Ontology querying (DL-Lite, EL, etc.)
- Automated product configuration
- Conceptual Modeling (e.g., UML)

Data Exchange, Data Provisioning, Data Wrangling



Σ_{st}

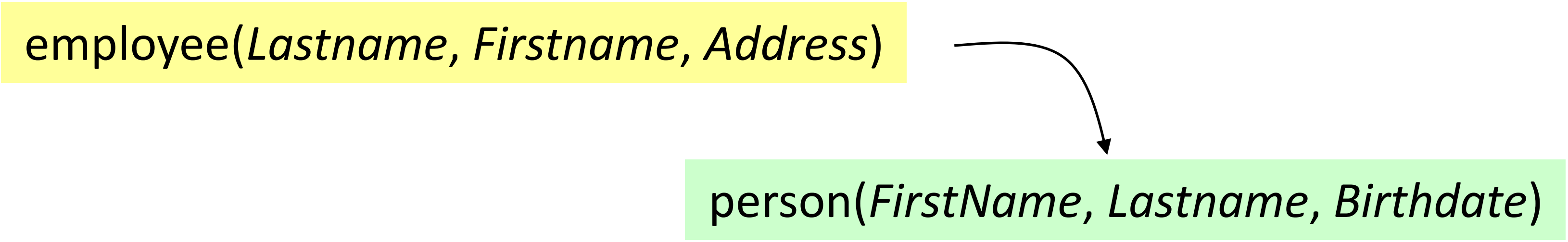
Σ_{t}

Source Schema S

Target Schema T

I

?



employee(*Lastname*, *Firstname*, *Address*)

person(*FirstName*, *Lastname*, *Birthdate*)

$$\mathtt{employee(X,Y,Z)} \rightarrow \exists \mathtt{W}\ \mathtt{person(Y,X,W)}$$

[Fagin, Kolaitis, Miller & Popa, 2003]; [Arenas et al., 2014]

Data Extraction

lixto
DELIVERING COMPETITIVE ADVANTAGE

DIADEM | domain-centric intelligent automated data extraction methodology

Wrapidity

PRODUCT	PRICE
Toshiba_Protege_cx	480
Dell_25416	360
Dell_23233	470
Acer_78987	390

Data Extraction

lixto
DELIVERING COMPETITIVE ADVANTAGE

DIADEM domain-centric intelligent automated data extraction methodology

Wrapidity

T_1	T_2
PRODUCT	PRICE
Toshiba_Protege_cx	480
Dell_25416	360
Dell_23233	470
Acer_78987	390

Data Extraction

we need object creation...

T_1	T_2
PRODUCT	PRICE
Toshiba_Protege_cx	480
Dell_25416	360
Dell_23233	470
Acer_78987	390

Data Extraction

$$\begin{array}{ll}
\texttt{table}(T_1), & \\
\texttt{table}(T_2), & \\
\texttt{sameColor}(T_1,T_2), & \\
\texttt{isNeighbourRight}(T_1,T_2) \rightarrow & \\
 & \exists T\ \texttt{tablebox}(T), \\
 & \quad \texttt{contains}(T,T_1), \\
 & \quad \texttt{contains}(T,T_2).
\end{array}$$

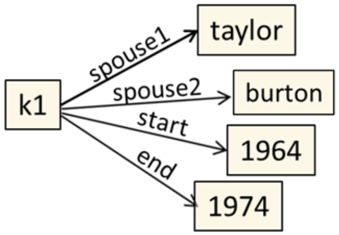
T_1	T_2
PRODUCT	PRICE
Toshiba_Protege_cx	480
Dell_25416	360
Dell_23233	470
Acer_78987	390

Reasoning with RDF – Object Creation

```
married(taylor,burton,1964,1974)
```

In the **RDF**-like “graph” notation this tuple is broken up into several triples (here represented as logical facts):

```
  spouse1(k1,taylor),
  spouse2(k1,burton),
  start(k1,1964),
  end(k1,1974)
```

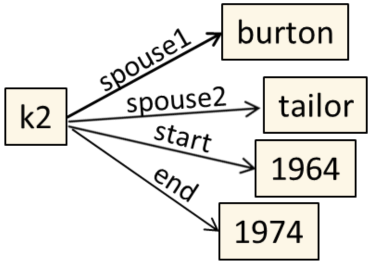


```
 ∀ u,v,x,y. married(u,v,x,y) → married(v,u,x,y)
```

This symmetry rule for marriage intervals now becomes:

```
spouse1(u,y1) ∧ spouse2(u,y2) ∧ start(u,y3) ∧ end(u,y4) →
∃v.spouse(v,y1) ∧ spouse1(v,y2) ∧ start(v,y3) ∧ end(v,y4)
```

```
  spouse1(k2,burton),
  spouse2(k2,taylor),
  start(k2,1964),
  end(k2,1974)
```



Description Logics & Ontological Reasoning

The DL-Lite Family

Popular family of DLs with low (AC_0) data complexity

DL-Lite TBox	First-Order Representation (Datalog$^{\pm}$)
DL-Lite$_{core}$	
$professor \sqsubseteq \exists teachesTo$	$\forall X\ professor(X) \rightarrow \exists Y\ teachesTo(X,Y)$
$professor \sqsubseteq \neg student$	$\forall X\ professor(X) \wedge student(X) \rightarrow \bot$
DL-Lite$_R$ (OWL 2 QL)	
$hasTutor^{-} \sqsubseteq teachesTo$	$\forall X \forall Y\ hasTutor(X,Y) \rightarrow teachesTo(Y,X)$
DL-Lite$_F$	
funct($hasTutor$)	$\forall X \forall Y \forall Z\ hasTutor(X,Y) \wedge hasTutor(X,Z) \rightarrow Y = Z$

[Calvanese, De Giacomo, Lembo, Lenzerini & Rosati, J. Autom. Reasoning 2007]

Datalog[∃]: Full Datalog augmented with ∃-quantifier

Unfortunately:

Theorem: Reasoning ($KB \vDash q$) with Datalog[∃] is undecidable.

[Beeri & Vardi, 1981]; [J. Mitchell 1983] [Chandra & Vardi 1985];

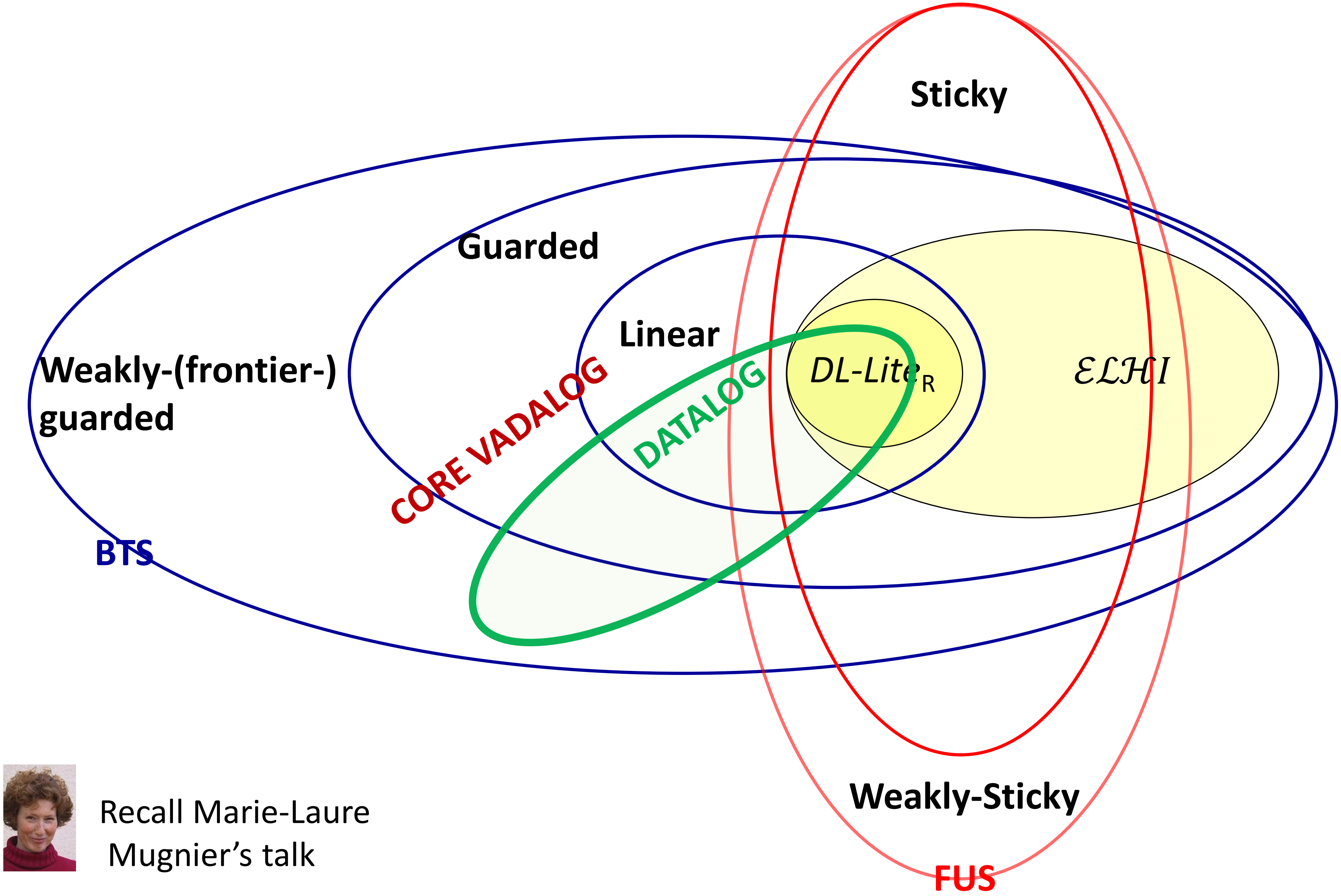
[Calì, G., & Kifer, 2008]; [Baget, Leclère & Mugnier, 2010]

Finding expressive decidable fragments has become a topic of intensive research over the last 10 years.

Datalog$^{\pm}$: Datalog[∃,⊥,¬strat, …] subject to syntactic restrictions.

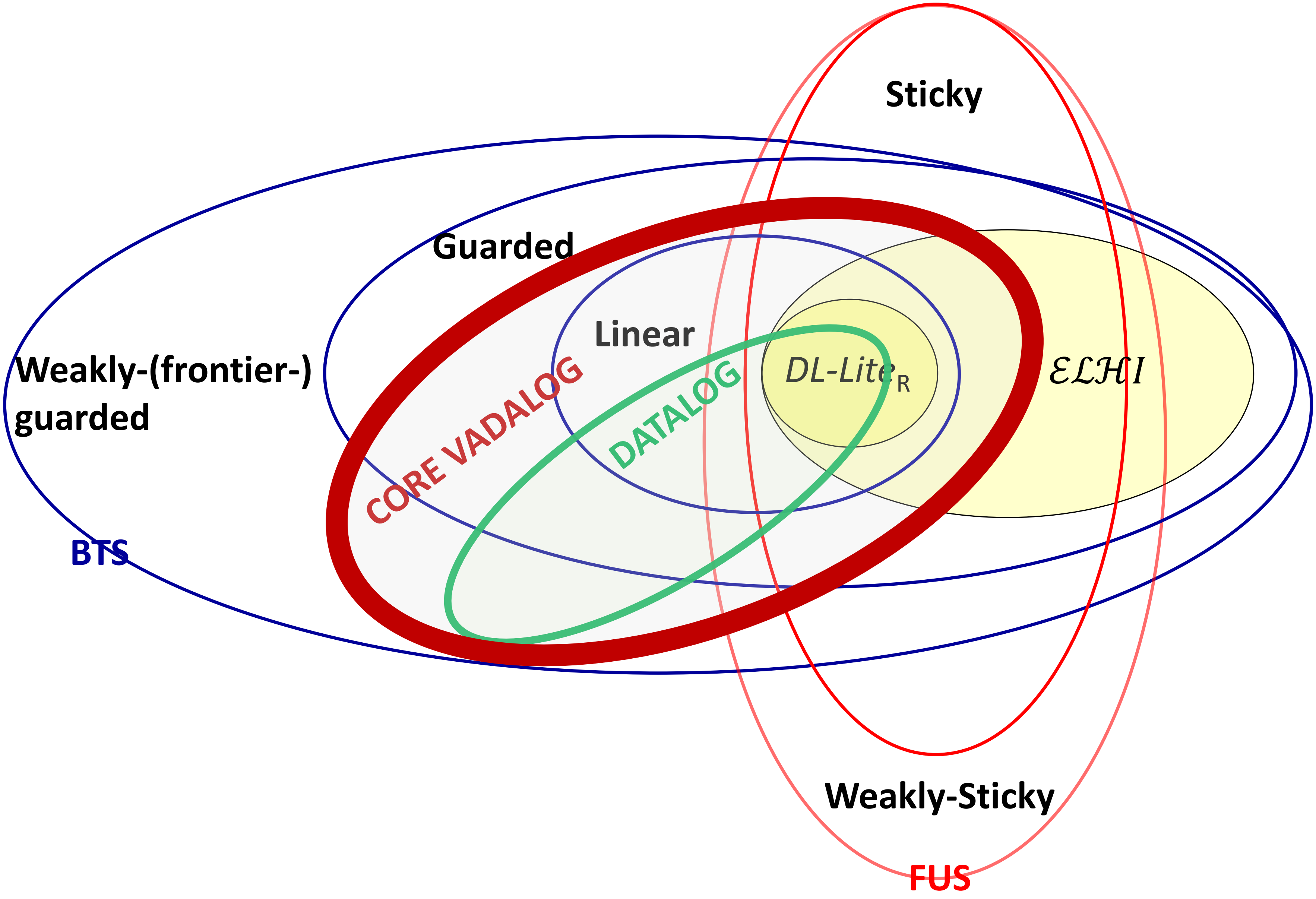
Vadalog: member of the Datalog$^{\pm}$ family admitting efficient reasoning methods.

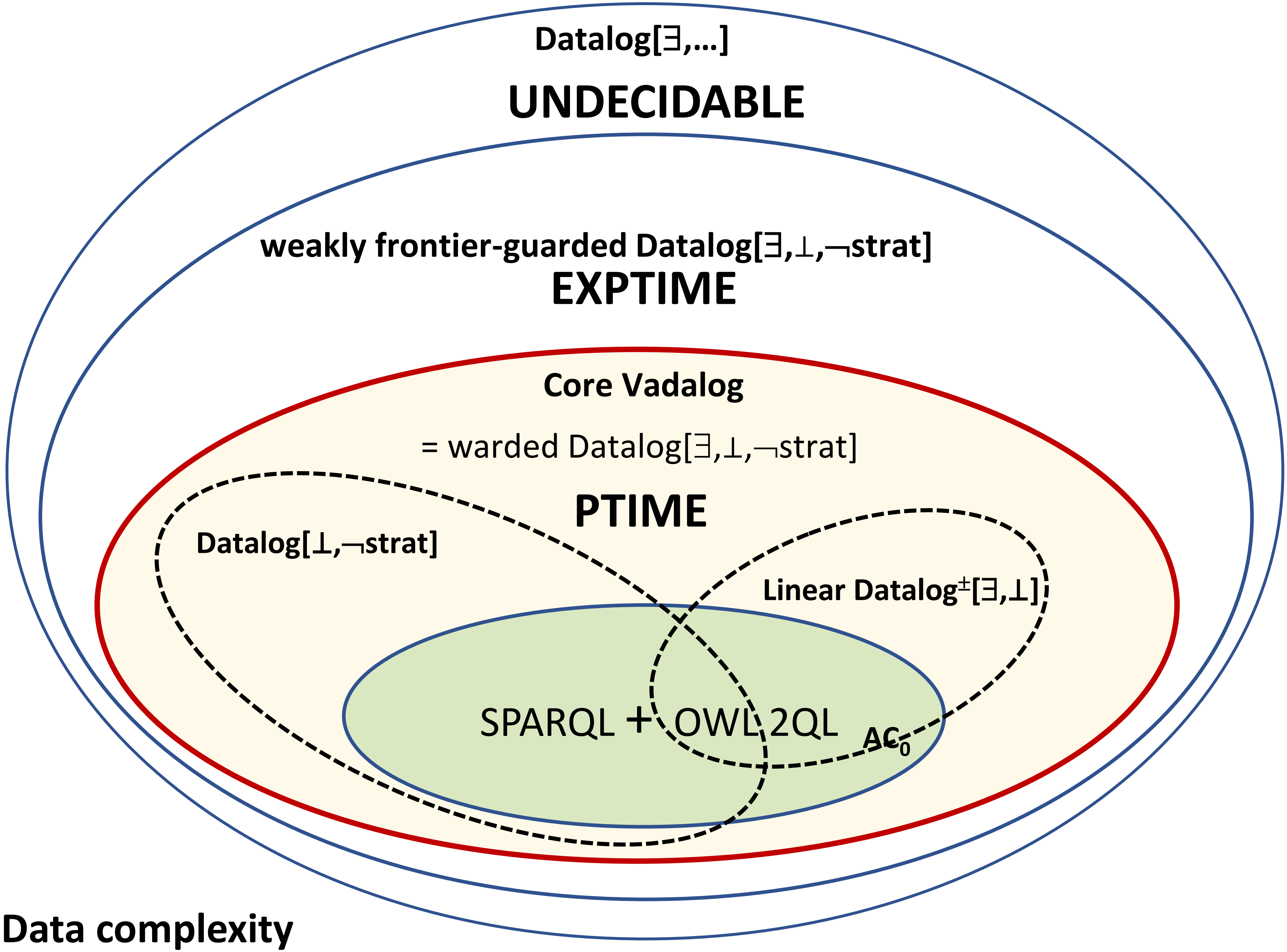
Main Decidable Datalog$^{\pm}$ Languages

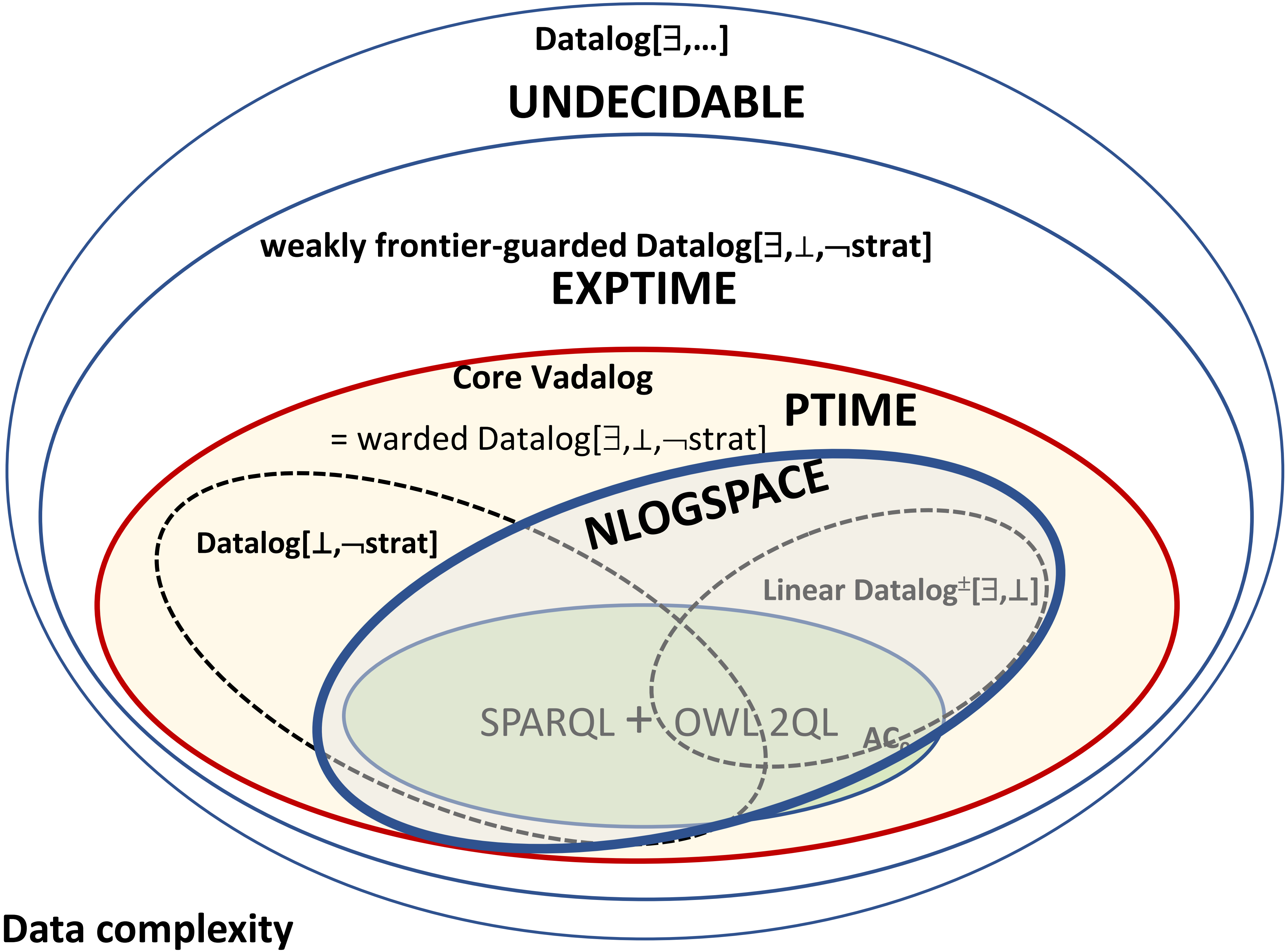


Recall Marie-Laure
Mugnier’s talk

Main Decidable Datalog$^{\pm}$ Languages




Datalog[∃,…]
UNDECIDABLE
weakly frontier-guarded Datalog[∃,⊥,¬strat]
EXPTIME
Core Vadalog
= warded Datalog[∃,⊥,¬strat]
PTIME
Datalog[⊥,¬strat]
Linear Datalog±[∃,⊥]
SPARQL + OWL 2QL
AC0
Data complexity


Datalog[∃,…]
UNDECIDABLE
weakly frontier-guarded Datalog[∃,⊥,¬strat]
EXPTIME
Core Vadalog
PTIME
= warded Datalog[∃,⊥,¬strat]
NLOGSPACE
Datalog[⊥,¬strat]
Linear Datalog±[∃,⊥]
SPARQL + OWL 2QL
AC0
Data complexity

Affected Positions in a Datalog$^{\pm}$ Program

Those positions of a predicate,
where existential values (Skolem terms) can “flow in”

$$P(X,Y),\ S(Y,Z)\ \rightarrow\ \exists W\ T(Y,X,W)$$

$$T(X,Y,Z)\ \rightarrow\ \exists W\ P(W,Z)$$

$$P(X,Y)\ \rightarrow\ \exists Z\ Q(X,Z)$$

Affected Positions in a Datalog$^{\pm}$ Program

Those positions of a predicate,
where existential values (Skolem terms) can “flow in”

$P(\underline{X},Y), S(Y,Z) \rightarrow \exists W\ T(Y,X,\underline{W})$

$T(X,Y,\underline{Z}) \rightarrow \exists W\ P(\underline{W},Z)$

$P(\underline{X},Y) \rightarrow \exists Z\ Q(X,\underline{Z})$

Affected Positions

T[3], P[1], Q[2]

Affected Positions in a Datalog$^{\pm}$ Program

Those positions of a predicate,
where existential values (Skolem terms) can “flow in”

$$P(\underline{X},\underline{Y}),\ S(Y,Z) \rightarrow \exists W\ T(Y,\underline{X},\underline{W})$$

$$T(X,\underline{Y},\underline{Z}) \rightarrow \exists W\ P(\underline{W},\underline{Z})$$

$$P(\underline{X},\underline{Y}) \rightarrow \exists Z\ Q(\underline{X},\underline{Z})$$

Affected Positions

T[3], P[1], Q[2]

T[2], P[2], Q[1]

Dangerous Variables in Rule Bodies

Head variables that, in the body, occur only in affected positions.

$P(\boxed{\underline{X}},\underline{Y}),\ S(Y,Z) \rightarrow \exists W\ T(Y,\underline{X},\underline{W})$

$T(X,\underline{Y},\boxed{\underline{Z}}) \rightarrow \exists W\ P(\underline{W},\underline{Z})$

$P(\boxed{\underline{X}},\underline{Y}) \rightarrow \exists Z\ Q(\underline{X},\underline{Z})$

Affected Positions

T[3], P[1], Q[2]

T[2], P[2], Q[1]

VADALOG is based on **Warded Rules**

A Datalog$^{\pm}$ program is **warded** if for each rule body:

- all dangerous variables jointly occur in a single „ward“ atom, and
- this ward shares only *unaffected* variables with the other body-atoms

$$P(\boxed{\underline{X}},\underline{Y}),\ S(Y,Z) \rightarrow \exists W\ T(Y,\underline{X},\underline{W})$$

$$T(X,\underline{Y},\boxed{\underline{Z}}) \rightarrow \exists W\ P(\underline{W},\underline{Z})$$

$$P(\boxed{\underline{X}},\underline{Y}) \rightarrow \exists Z\ Q(\underline{X},\underline{Z})$$

Affected Positions

T[3], P[1], Q[2]

T[2], P[2], Q[1]

Core Vadalog = warded Datalog[∃,⊥,¬strat]

Clearly, Datalog is contained in Vadalog

Examples of Warded Datalog$^{\pm}$ Rules

1. **Symmetry rule for marriage intervals:**

$$
\begin{array}{l}
\texttt{spouse1}(\underline{\texttt{x}},\texttt{y1}) \land \texttt{spouse2}(\underline{\texttt{x}},\texttt{y2}) \land \\
\texttt{start}(\underline{\texttt{x}},\texttt{y3}) \land \texttt{end}(\underline{\texttt{x}},\texttt{y4}) \rightarrow \\
\qquad\qquad \exists \texttt{v}.\ \texttt{spouse2}(\underline{\texttt{v}},\texttt{y1}) \land \texttt{spouse1}(\underline{\texttt{v}},\texttt{y2}) \land \\
\qquad\qquad\qquad \texttt{start}(\underline{\texttt{v}},\texttt{y3}) \land \texttt{end}(\underline{\texttt{v}},\texttt{y4})
\end{array}
$$

2. **OWL 2 QL description logic:**

DL-Lite TBox	First-Order Representation (Datalog$^{\pm}$)
DL-Lite$_{core}$	
$\mathit{professor} \sqsubseteq \exists \mathit{teachesTo}$	$\forall X\ \mathit{professor}(X) \rightarrow \exists Y\ \mathit{teachesTo}(X,Y)$
$\mathit{professor} \sqsubseteq \neg \mathit{student}$	$\forall X\ \mathit{professor}(X) \land \mathit{student}(X) \rightarrow \perp$
DL-Lite$_{R}$ (OWL 2 QL)	
$\mathit{hasTutor}^{-} \sqsubseteq \mathit{teachesTo}$	$\forall X \forall Y\ \mathit{hasTutor}(X,Y) \rightarrow \mathit{teachesTo}(Y,X)$

Language Features (selection)

Data types and associated operations & expressions:
integer, float, string, Boolean, date, sets.

Monotonic aggregations: min, max, sum, prod, count
work even in presence of recursion while preserving monotonicity of set-containment

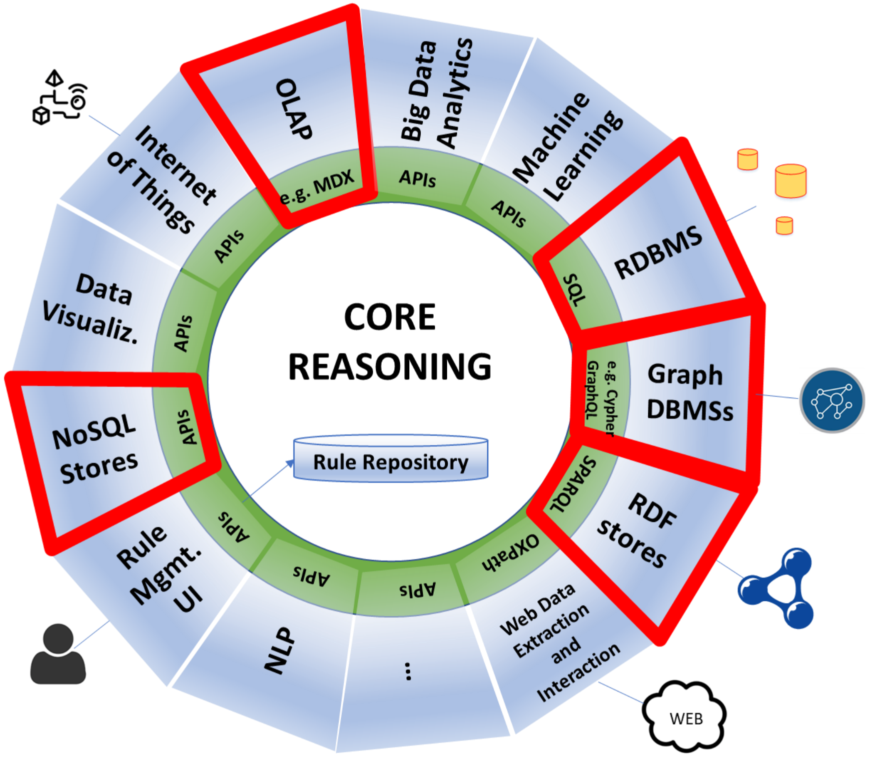
Example: Company Control

```
own(x,y,w), w>0.5 → control(x,y);
control(x,y),own(y,z,w),
  v=msum(w,⟨y⟩), v>0.5 → control(x,z).
```

Probabilistic reasoning: facts and rules can be adorned with probabilities. Marginal probabilities for derived facts will be computed assuming independence.

Equality (EGDs, functional dependencies) if non-conflicting.

Database Interface



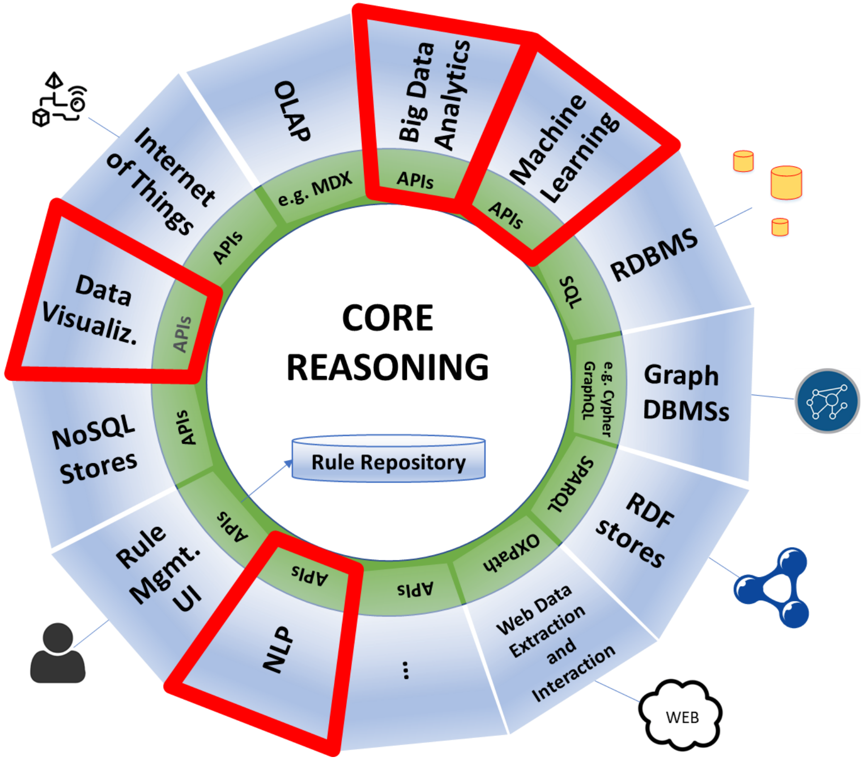
```
@bind("Own", "rdbms", "companies.ownerships").


@qbind("Own", "graphDB", "MATCH (a)-[o:Owns]->(b) RETURN a,b,o.weight").
```

Cypher query (Neo4j)

```
@bind("q","data source", "schema","table").
@update("q",{1,3,4,5}).
```

Machine Learning, Big Data Analytics, NLP & Data Visualization



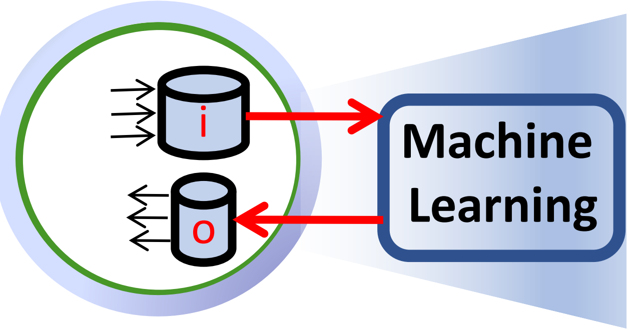
We are currently experimenting with different tools and different types of interfaces and interactions.

WEKA The University of Waikato

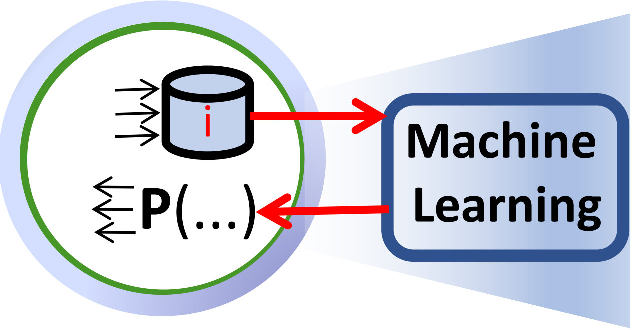
R

TensorFlow

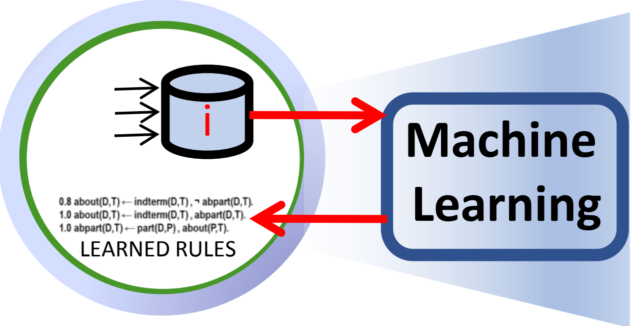
Interaction Model 1



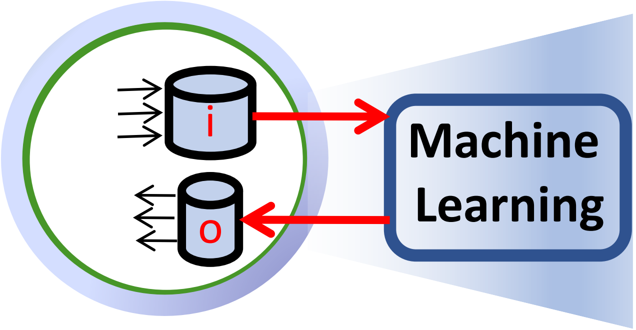
Interaction Model 2



Interaction Model 3

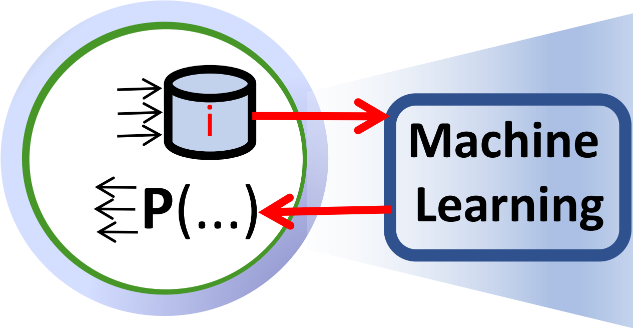

i
0.8 about(D,T) ← indterm(D,T) , ¬ abpart(D,T).
1.0 about(D,T) ← indterm(D,T) , abpart(D,T).
1.0 abpart(D,T) ← part(D,P) , about(P,T).
LEARNED RULES
Machine Learning

Interaction Model 1



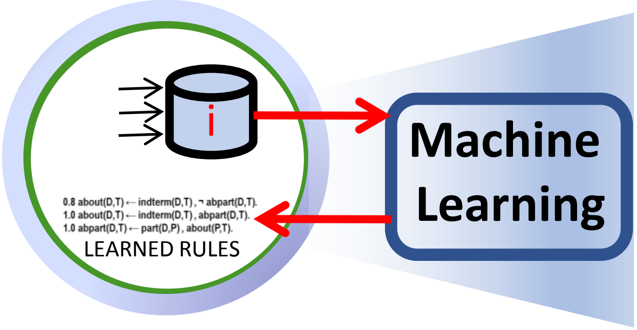
- We prepare a relation as ML input.
- ML sw classifies facts and sends them into the core reasoning system.

Interaction Model 2



- ML package acts as a special predicate.
- Called by the core reasoning system.

Interaction Model 3



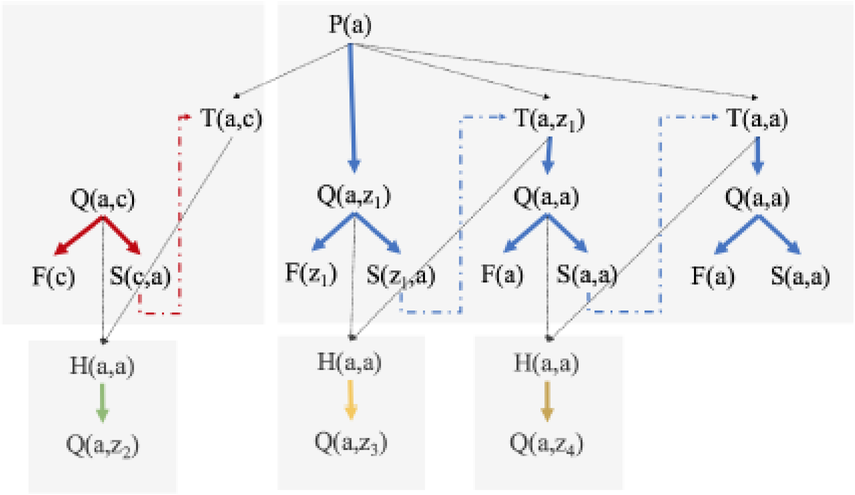
0.8 about(D,T) ← indterm(D,T) , ¬ abpart(D,T).
1.0 about(D,T) ← indterm(D,T) , abpart(D,T).
1.0 abpart(D,T) ← part(D,P) , about(P,T).

- ML sw learns rules.
- Rules are translated into probabilistic Vadalog rules.

Core Algorithms

$$D = \{\mathrm{P}(a), \mathrm{Q}(a, c)\}$$

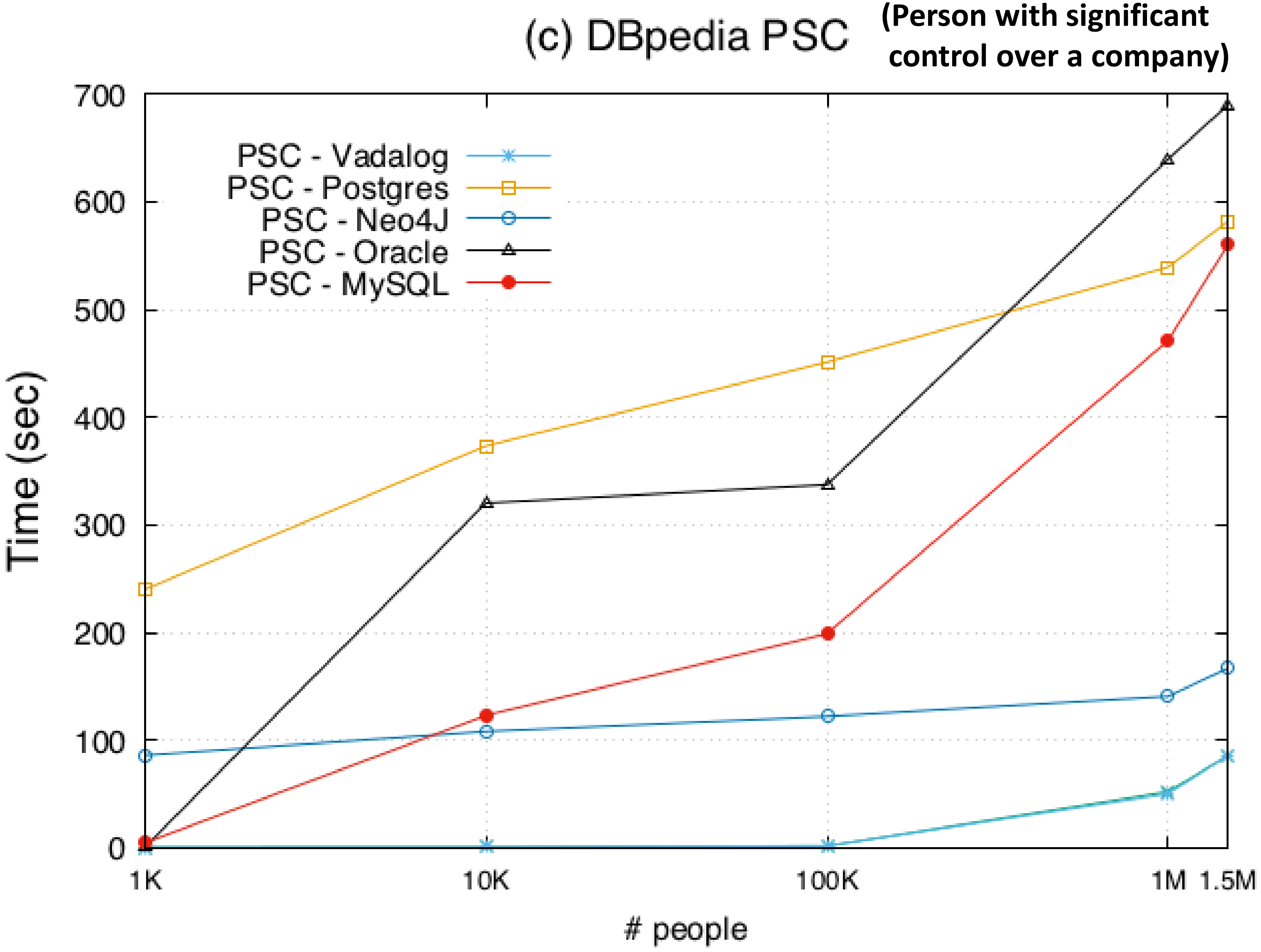
$$\begin{aligned}
1 &: \mathrm{P}(x) \rightarrow \exists z\, \underline{\mathrm{Q}}(x, \hat{z})\\
2 &: \mathrm{Q}(x, \hat{y}) \rightarrow \mathrm{S}(\hat{y}, x)\\
3 &: \mathrm{S}(\hat{x}, y), \mathrm{P}(y) \rightarrow \mathrm{T}(y, \hat{x})\\
4 &: \mathrm{T}(x, \hat{y}), \mathrm{Q}(z, \hat{y}) \rightarrow \underline{\mathrm{H}}(x, z)\\
5 &: \mathrm{T}(x, \hat{y}) \rightarrow \underline{\mathrm{Q}}(x, x)\\
6 &: \mathrm{Q}(x, \hat{y}) \rightarrow \underline{\mathrm{F}}(\hat{y})\\
7 &: \mathrm{H}(x, x) \rightarrow \exists z\, \underline{\mathrm{Q}}(x, \hat{z})\\
8 &: \mathrm{P}(x) \rightarrow \exists z\, \mathrm{T}(x, \hat{z}).
\end{aligned}$$



- Bottom-up chase processing with „aggressive“ termination strategy
- Top-down query processing (currently under implementation)
- Advanced program rewriting and optimization techniques
- Efficient & highly scalable cache managmt., query plan optimization
- Recent evaluation shows the system is extremely competitive

Performance

(Person with significant control over a company)



Crucial Question

We have a powerful language and system for reasoning with rules Over „Big Data“ and can interact with machine learning.

But are there actually real problems that can be solved with a reasonable number of rules?

Crucial Question

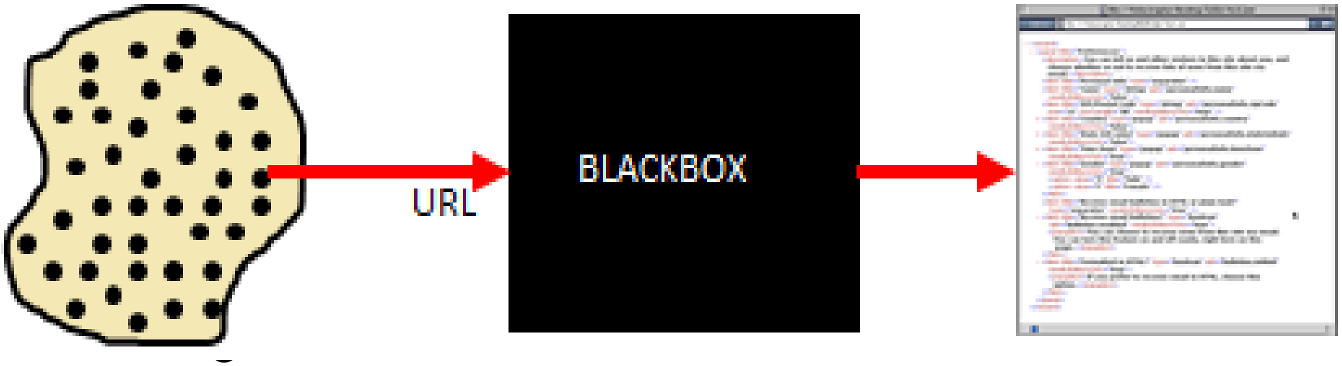
We have a powerful language and system for reasoning with rules Over „Big Data“ and can interact with machine learning.

But are there actually real problems that can be solved with a reasonable number of rules?

Yes, there are many!, for example:

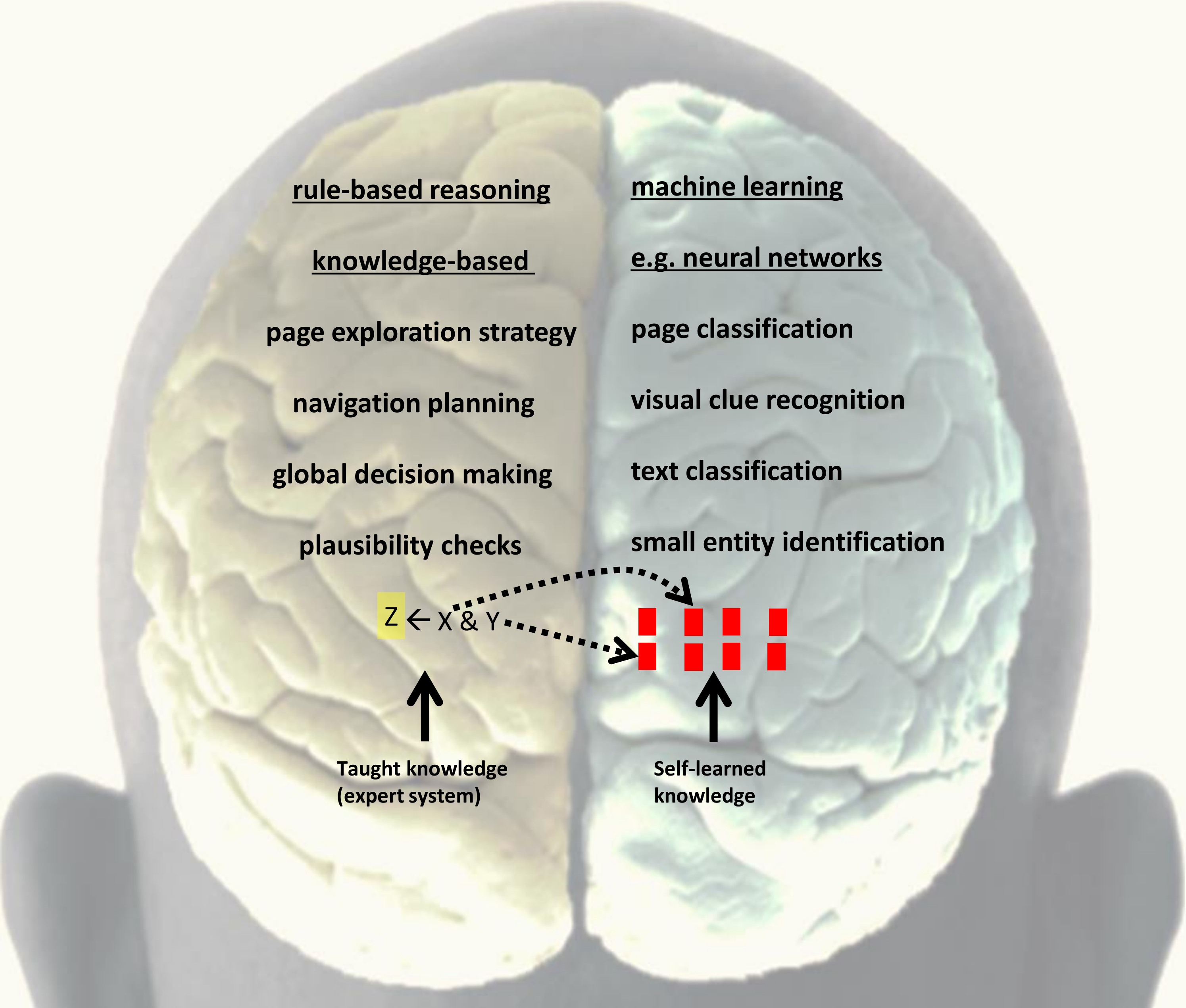
- Banks: Fraud detection, [current project]
- Banks: Creditworthiness
- Logistics: Supply chain risks
- Security companies: Detection of critical events
- Fully automated Web data extraction: The **DIADEM project**
- … and 10000 more.

DIADEM



Application domain with thousands of websites

Application-relevant Structured data (XML or RDF)



rule-based reasoning

knowledge-based

page exploration strategy

navigation planning

global decision making

plausibility checks

machine learning

e.g. neural networks

page classification

visual clue recognition

text classification

small entity identification

Z ← X & Y

Taught knowledge
(expert system)

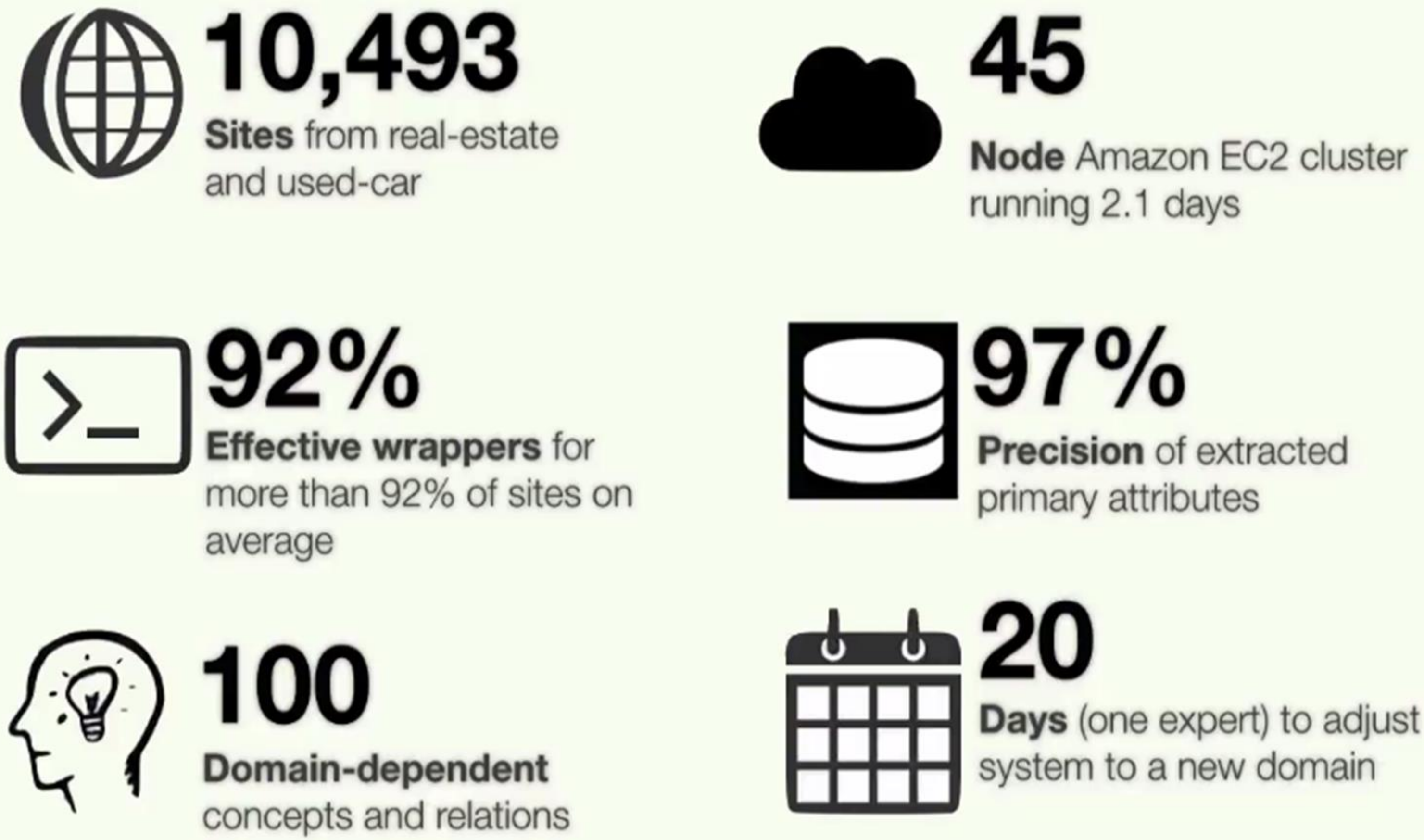
Self-learned
knowledge

Rough Idea: Knowledge via Rules

Use “expert” rules that analyze Web pages and interact with them

- **Ontological rules** (how do entities relate to each other)
  - a flat is a real-estate property
  - a house is a real-estate property
  - a real-estate property has a number of rooms
  - a price consists of a number and a currency
- **Phenomenological rules** (how do entities manifest themselves on the Web?)
  - the text chunk closest to an input field is with high prob. its descriptor.
  - each sales item is described in a “convex” (usual. rectangular) region.
- **Site exploration rules:**
  - before filling a field try to leave it empty
  - rules for handling next-page links
- **Other types of rules**

Evaluation on 10k+ Sites



10,493
Sites from real-estate and used-car

45
Node Amazon EC2 cluster running 2.1 days

92%
Effective wrappers for more than 92% of sites on average

97%
Precision of extracted primary attributes

100
Domain-dependent concepts and relations

20
Days (one expert) to adjust system to a new domain

Domains considered so far (since 2014)

- **Real estate UK**
- **Real estate US**
- **Used cars**
- **Consumer electronics**
- **Restaurant chains**
- **Restaurants in the ‘Open Web’**
- **Jobs (from company Web sites)**
- **News**
- **Companies**

Commercial Impact

DIADEM | domain-centric intelligent automated data extraction methodology

ERC Advanced Grant DIADEM
+ ERC Proof of Concept Grant
EXTRALYTICS

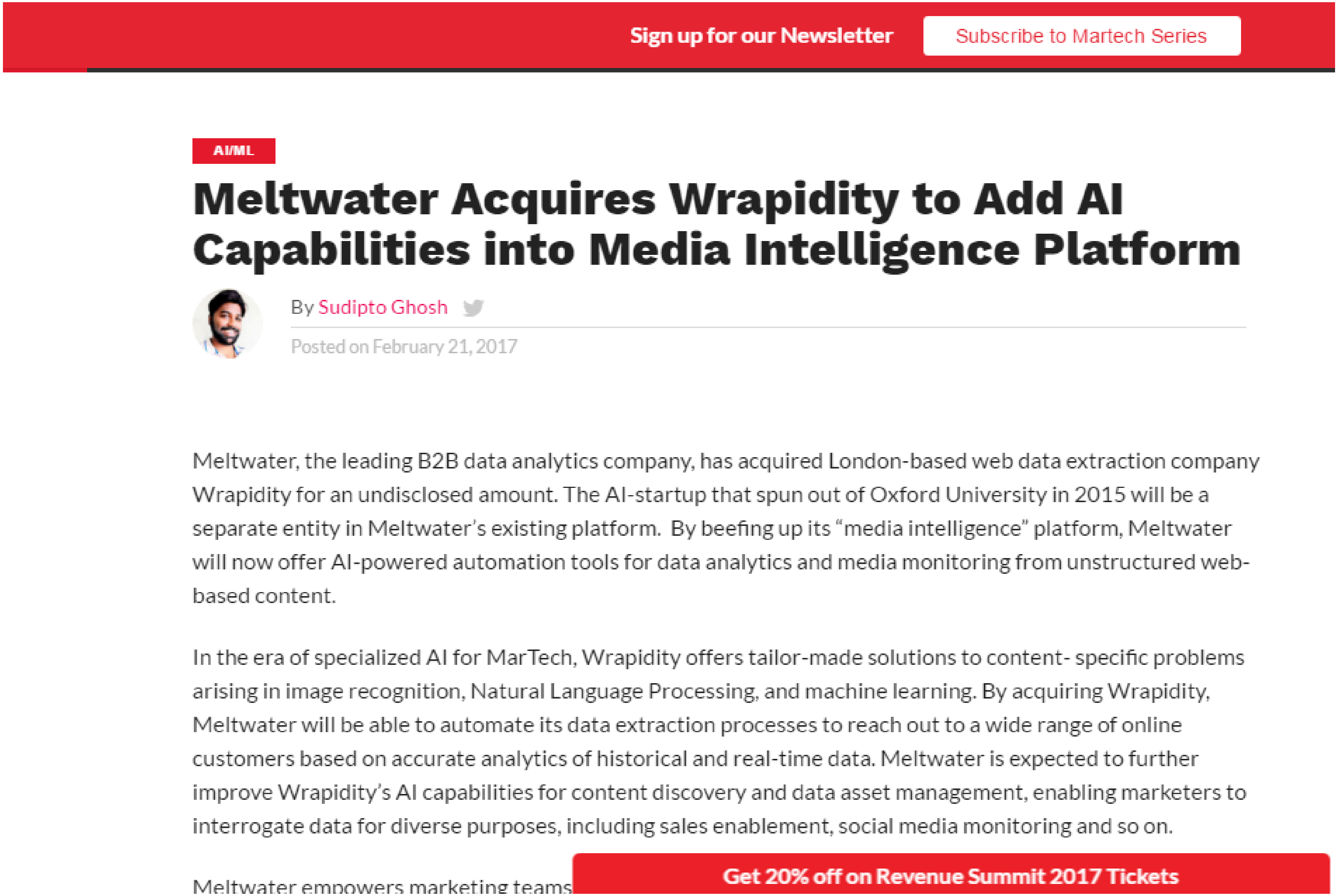
→

Wrapidity

Founded February 2015
operating initially in Oxford
now in London

Two possibilities:

- Build up company with large client portfolio
- Sell technology, IP & software to strategic partner



AI/ML

Meltwater Acquires Wrapidity to Add AI Capabilities into Media Intelligence Platform

By Sudipto Ghosh

Posted on February 21, 2017

Meltwater, the leading B2B data analytics company, has acquired London-based web data extraction company Wrapidity for an undisclosed amount. The AI-startup that spun out of Oxford University in 2015 will be a separate entity in Meltwater's existing platform. By beefing up its "media intelligence" platform, Meltwater will now offer AI-powered automation tools for data analytics and media monitoring from unstructured web-based content.

In the era of specialized AI for MarTech, Wrapidity offers tailor-made solutions to content- specific problems arising in image recognition, Natural Language Processing, and machine learning. By acquiring Wrapidity, Meltwater will be able to automate its data extraction processes to reach out to a wide range of online customers based on accurate analytics of historical and real-time data. Meltwater is expected to further improve Wrapidity's AI capabilities for content discovery and data asset management, enabling marketers to interrogate data for diverse purposes, including sales enablement, social media monitoring and so on.

Meltwater empowers marketing teams

REFERENCES

KNOWLEDGE GRAPHS & THE VADALOG SYSTEM:

Luigi Bellomarini, Georg Gottlob, Andreas Pieris, Emanuel Sallinger: Swift Logic for Big Data and Knowledge Graphs. International Joint Conference on Artificial Intelligence (IJCAI) 2017.

Luigi Bellomarini, Emanuel Sallinger, Georg Gottlob: The Vadalog System: Datalog-based Reasoning for Knowledge Graphs. Proceedings of the VLDB Endowment (PVLDB) 11(9) 2018.

Luigi Bellomarini, Daniele Fakhoury, Georg Gottlob, Emanuel Sallinger:
Knowledge Graphs and Enterprise AI: The Promise of an Enabling Technology. ICDE 2019: 26-37

CORE VADALOG, SPARQL, WARDED RULES, COMPLEXITY, THEORY:

Gerald Berger, Georg Gottlob, Andreas Pieris, Emanuel Sallinger: The Space-Efficient Core of Vadalog. PODS 2019: 270-284

M.Arenas, G.Gottlob, A.Pieris: Expressive languages for querying the semantic web. ACM Trans. Database Syst. 43(3): 13:1-13:45 (2018)

Georg Gottlob, Sebastian Rudolph, Mantas Simkus: Expressiveness of guarded existential rule languages. PODS 2014: 27-38

Leopoldo E. Bertossi, Georg Gottlob, Reinhard Pichler: Datalog: Bag Semantics via Set Semantics. ICDT 2019: 16:1-16:19

DATALOG+/- :

Cali, Gottlob, & Kifer: Taming the Infinite Chase: Query Answering under Expressive Relational Constraints. Journal of Artificial Intelligence Research 48, 2013

A.Calì, G.Gottlob, T.Lukasiewicz: A general Datalog-based framework for tractable query answering over ontologies. Journal of Web Semantics 14: 5, 2012

A.Cali, G.Gottlob, A.Pieris. Towards More Expressive Ontology Languages: the Query-Answering Problem. Artificial Intelligence 193 2012.

G.Gottlob, T.Lukasiewicz, M.-V. Martinez, G.Simari: Query answering under probabilistic uncertainty in Datalog± ontologie Annals of Mathematics and Artificial Intelligence 69:1, 2013.

DIADEM WEB DATA EXTRACTION SYSTEM:

Tim Furche, Georg Gottlob, Giovanni Grasso, Xiaonan Guo, Giorgio Orsi, Christian Schallhart, Cheng Wang:
DIADEM: Thousands of Websites to a Single Database. PVLDB 7(14): 1845-1856 (2014)

Thank You!

DIADEM DEMO: → wrapidity.com