Knowledge Graphs in Action

Georg Gottlob

Emanuel Sallinger

AI Summer School 2023

3 July 2023

Knowledge Graphs in Action

Part 1: Modern Knowledge Graphs and Vadalog

Georg Gottlob

UNIVERSITY OF OXFORD

TU WIEN

AI Summer School 2023

Center for Artificial Intelligence and Machine Learning

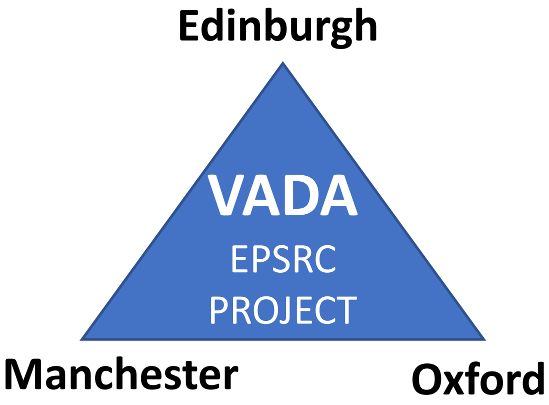
asai austrian society for artificial intelligence

3 July 2023

Knowledge Graphs in Action

Part 1: Modern Knowledge Graphs and Vadalog

Georg Gottlob



AI Summer School 2023

3 July 2023

Knowledge Graphs as Large “World” KBs

Cyc [Lenat &Guha 1989]

W: “comprehensive ontology and knowledge base of everyday common sense knowledge”.

Freebase [Bollacker et al. 2007] W: “online collection of structured data harvested from many sources, including user-submitted wiki contributions”.

Google Knowledge Graph [Singhal 2012] **+ K.Vault** [Dong et al. 2014]

W: “KB used by Google to enhance its search engine's search results with semantic-search information gathered from a wide variety of sources”.

DBpedia [Auer et al. 2007]. **Yago** [Suchanek et al 2007]

both generate structured ontologies from Wikipedia.

Wikidata [Vrandečić 2012, Krötzsch+V. 2014] open knowledge base that can be read and edited by both humans and machines.

More Specialized Knowledge Graphs

Facebook Knowledge Graph: Social graph with people, places and things + information from Wikipedia

Amazon Knowledge Graph: Started as product categorization ontology

Wolfram KB: World facts + mathematics

Factual: Businesses & places

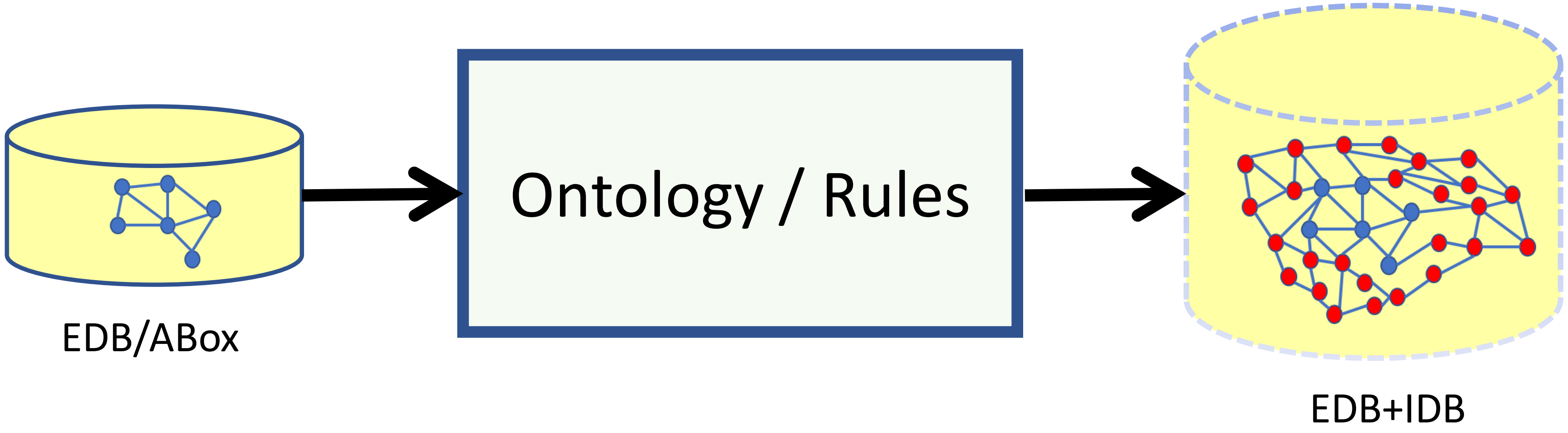
Megagon (Recruit Inst.): People, skills, recruiting

Central Banks: Company register – ownership graph

Credit Rating Agencies …

Thousands of medium to large size companies now want their own corporate knowledge graph. This not just for semantic indexing and search, but for advanced reasoning tasks on top of machine learning.

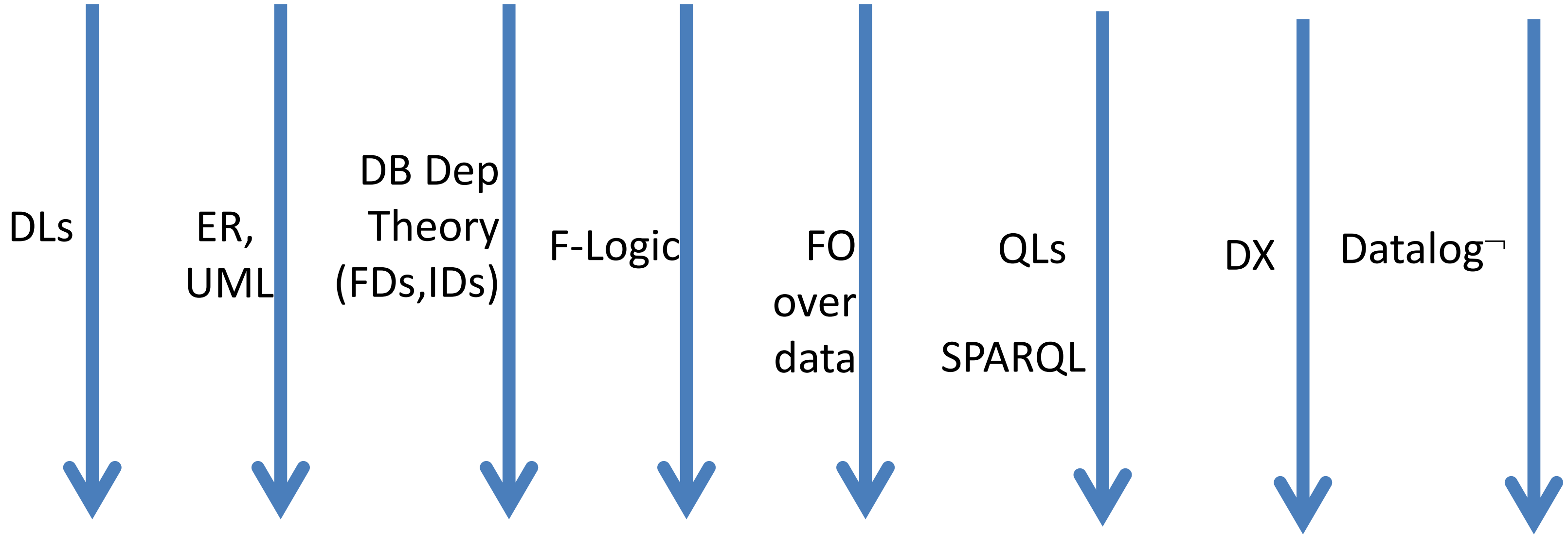
Reasoning in Knowledge Graphs



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.

DLs, Logic, KR, Databases, and all that



DLs

ER,
UML

DB Dep
Theory
(FDs,IDs)

F-Logic

FO
over
data

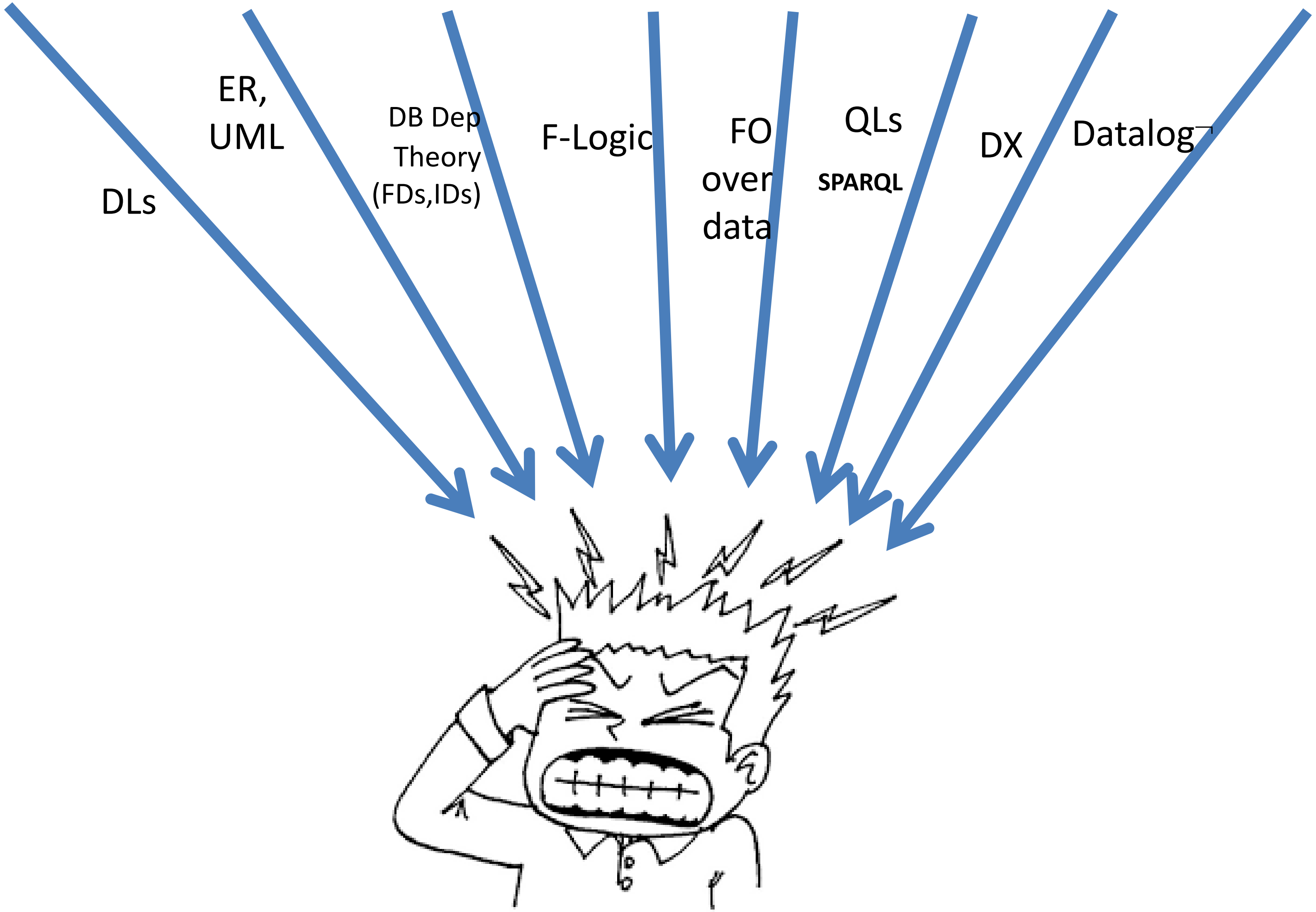
QLs

SPARQL

DX

Datalog$^{\neg}$

DLs, Logic, KR, Databases, and all that



DLs

ER, UML

DB Dep Theory (FDs,IDs)

F-Logic

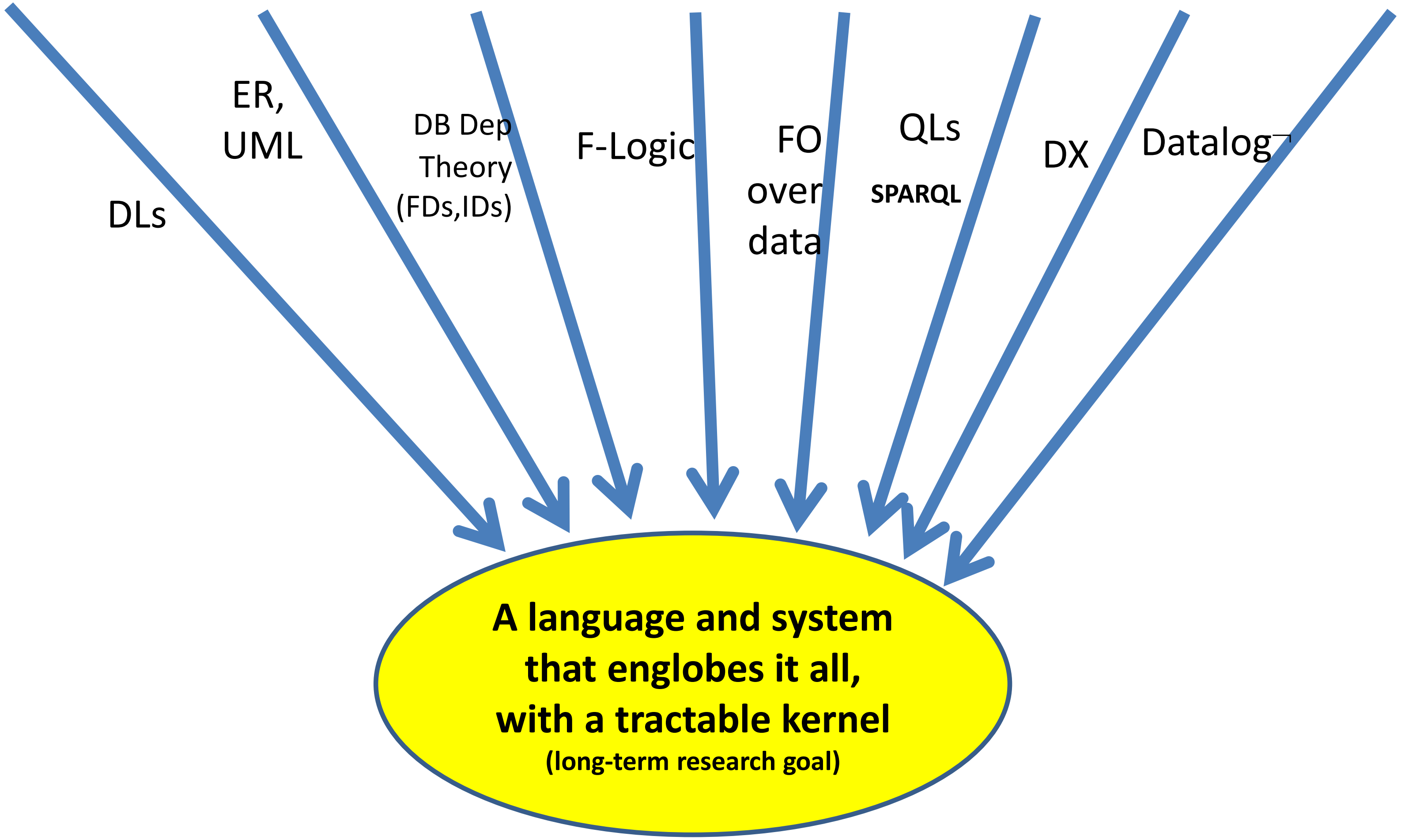
FO over data

QLs SPARQL

DX

Datalog¬

DLs, Logic, KR, Databases, and all that



A language and system that englobes it all, with a tractable kernel

(long-term research goal)

A Simple Example

married(Schneider, Meyen, 1966, 1975),
married(Schneider, Biasini, 1975, 1981),
married(Niven, Rollo, 1940, 1946),
married(Niven, Genberg, 1948, 1983),
married(Taylor, Hilton, 1950, 1951),
married(Hilton, Taylor, 1950, 1951),
married(Taylor, Wilding, 1952, 1957),
married(Taylor, Todd, 1957, 1958),
married(Taylor, Fisher 1959, 1964),
married(Taylor, Burton,1964,1974),
married(Taylor, Burton, 1975,1976),
married(Taylor, Warner, 1976, 1982),
married(Taylor, Fortensky, 1991,1996),
married(..... , ... , ... , ...)

Marriage Database

Q: married(Taylor, Burton, X? ,Y?)

A: { (1964,1974), (1975,1976) }

Q: married(Burton, Taylor ,X? ,Y?)

A: {} ✗

A Simple Example

married(Schneider, Meyen, 1966, 1975),
married(Schneider, Biasini, 1975, 1981),
married(Niven, Rollo, 1940, 1946),
married(Niven, Genberg, 1948, 1983),
married(Taylor, Hilton, 1950, 1951),
married(Hilton, Taylor, 1950, 1951),
married(Taylor, Wilding, 1952, 1957),
married(Taylor, Todd, 1957, 1958),
married(Taylor, Fisher 1959, 1964),
married(Taylor, Burton,1964,1974),
married(Taylor, Burton, 1975,1976),
married(Taylor, Warner, 1976, 1982),
married(Taylor, Fortensky, 1991,1996),
married(..... , ... , ... , ...)

Marriage Database

Q: married(Taylor, Burton ,X? ,Y?)

A: { (1964,1974), (1975,1976) }

Q: married(Burton, Taylor, X? ,Y?)

A: {} ✗

Let us add the rule:

married(u,v,x,y) →married(v,u,x,y)

Q: married(Taylor, Burton ,X? ,Y?)

A: { (1964,1974), (1975,1976) } ✓

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)
```

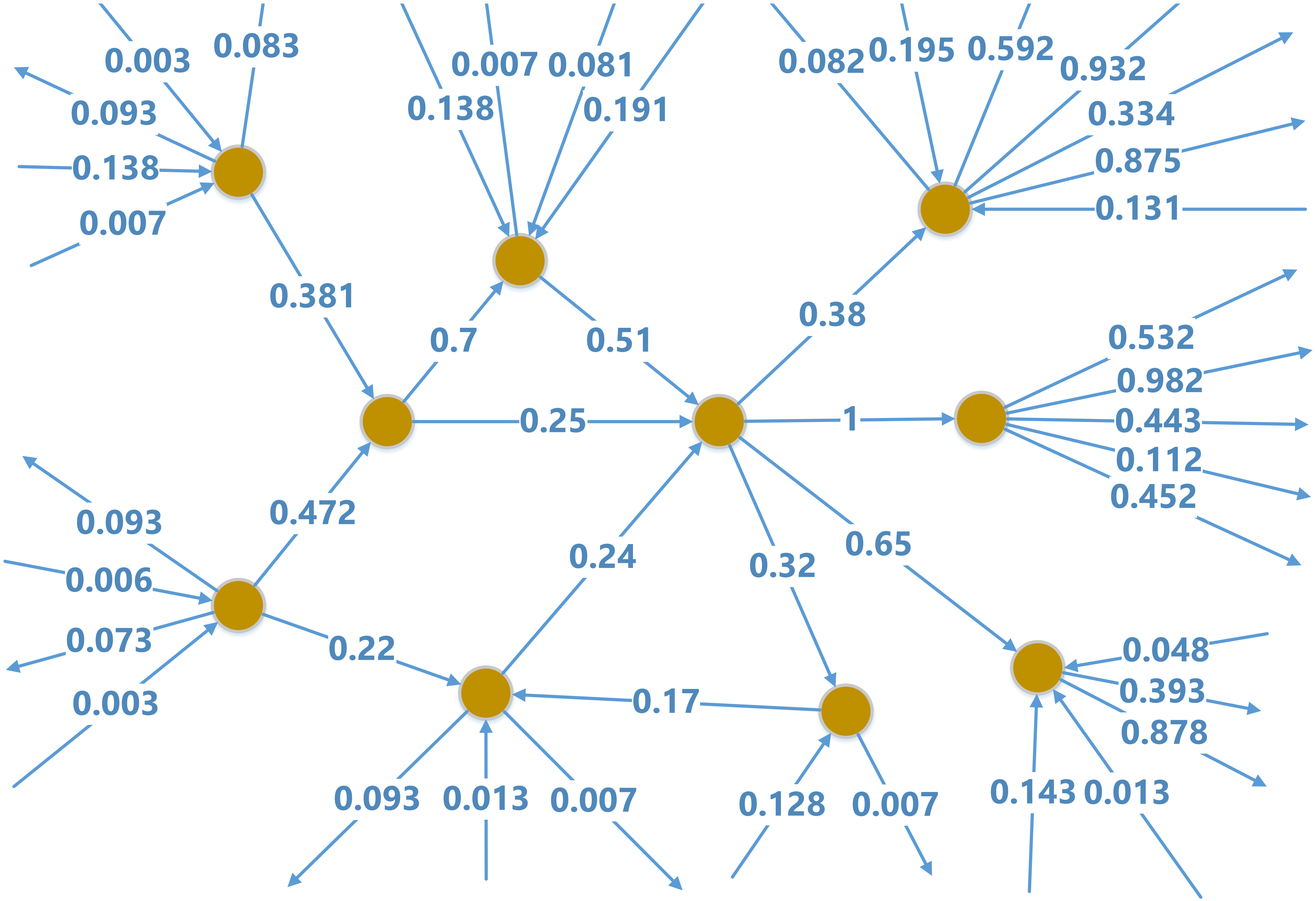
Symmetry rule for marriage intervals in Datalog:

```
∀ 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)  → own(x,x,1).
                                 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).
```

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

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 <u>wrong data</u>.

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 <u>wrong data</u>.

A human credit rating expert would instead use of the rule:

If property owners move into their recently bought one-family property, then the previous occupiers have <u>most likely</u> moved out.

(Such updates are often missing in the database)

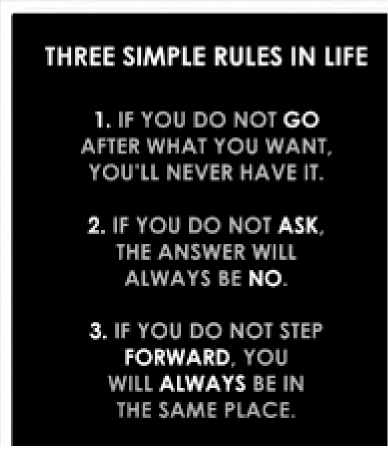
This rule can be used to update the database <u>before</u> applying machine learning.

Knowledge Graph Management Systems (KGMS)

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

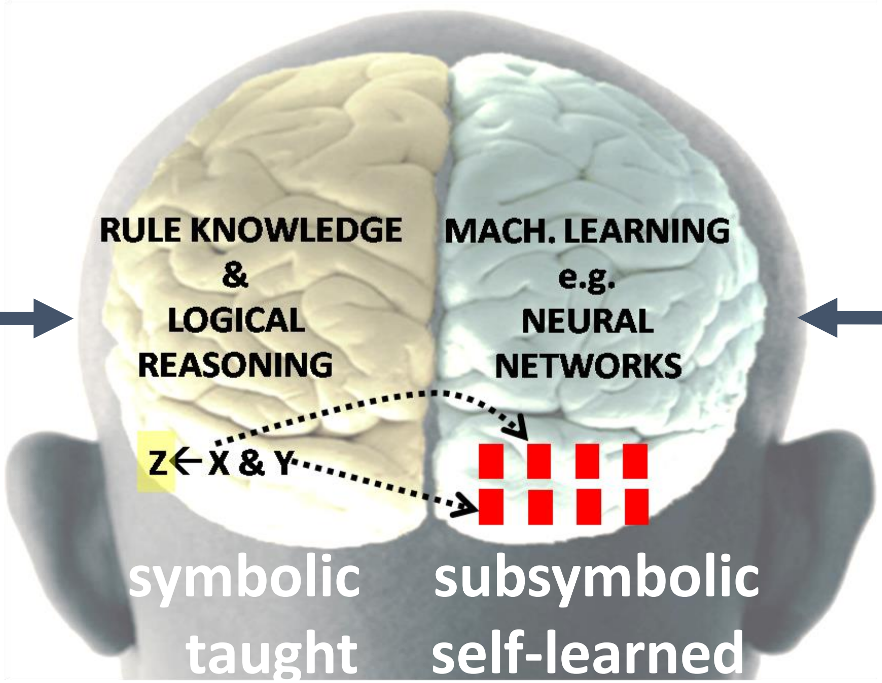


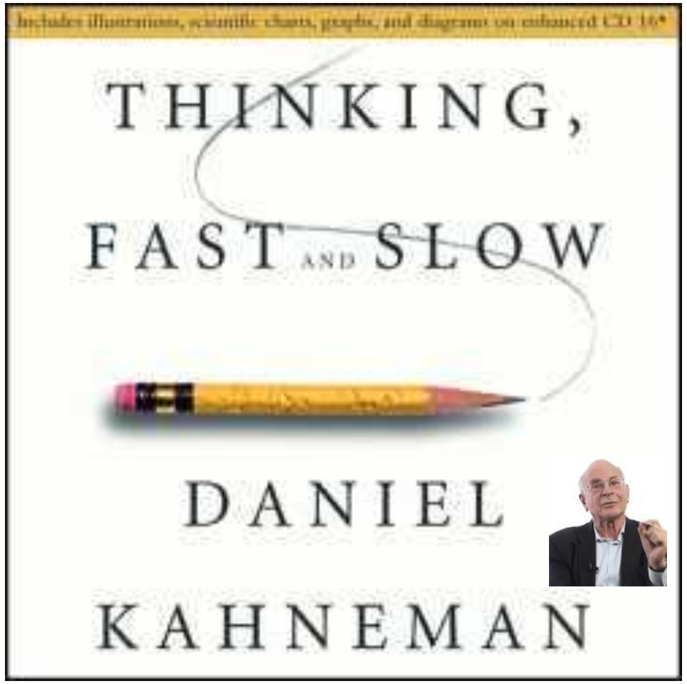
THREE SIMPLE RULES IN LIFE

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.



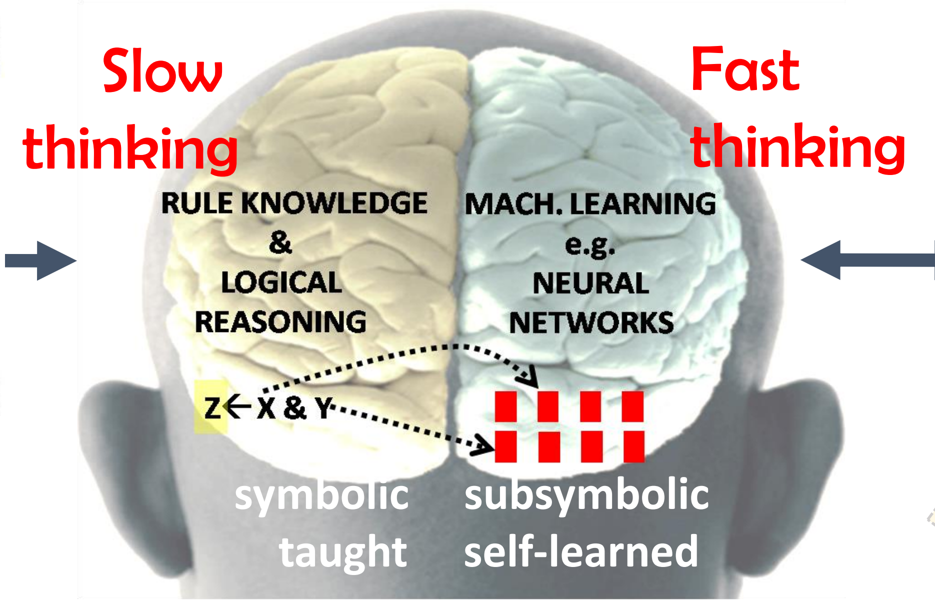


Includes illustrations, scientific charts, graphs, and diagrams on enhanced CD 16*

THINKING, FAST AND SLOW

DANIEL KAHNEMAN

Grandma: “Fly agarics are poisonous mushrooms. If you eat a poisonous mushroom, you may die”.



Slow thinking

Fast thinking

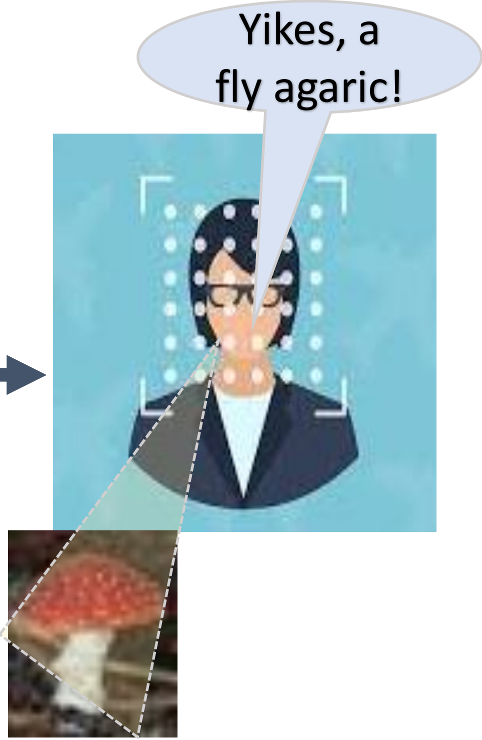
RULE KNOWLEDGE & LOGICAL REASONING

MACH. LEARNING e.g. NEURAL NETWORKS

Z← X & Y

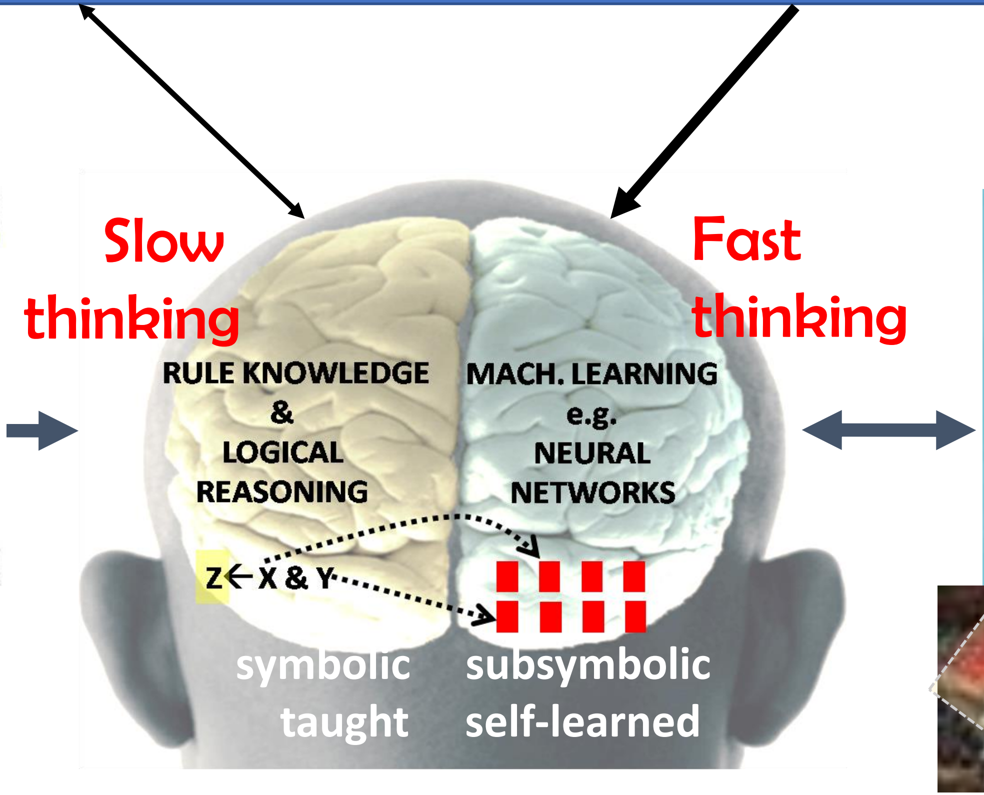
symbolic taught

subsymbolic self-learned



PRE-TRAINED LARGE LANGUAGE MODELS (LLMs)

Grandma: “Fly agarics are poisonous mushrooms. If you eat a poisonous mushroom, you may die”.



Slow thinking

RULE KNOWLEDGE & LOGICAL REASONING

Z← X & Y

symbolic taught

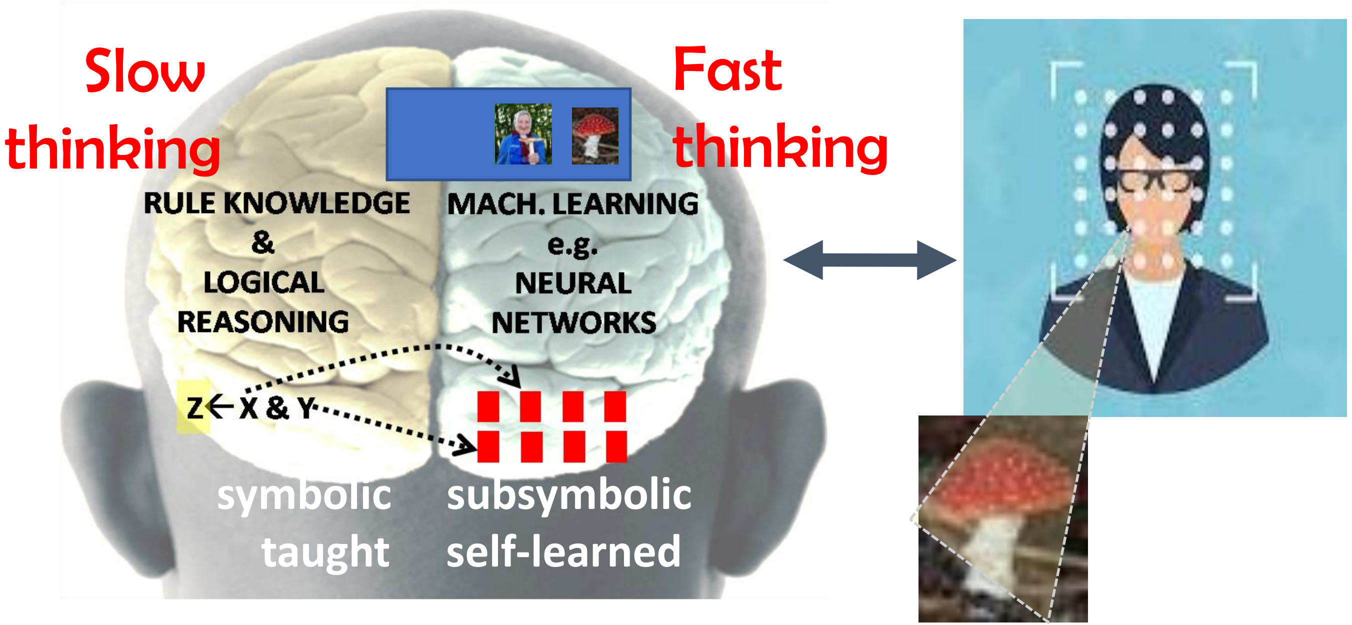
Fast thinking

MACH. LEARNING e.g. NEURAL NETWORKS

subsymbolic self-learned

How I imagine it could work, and why LLMs need to be integrated with a KG

PRE-TRAINED LARGE LANGUAGE MODELS (LLMs)



Slow thinking

Fast thinking

RULE KNOWLEDGE & LOGICAL REASONING

MACH. LEARNING e.g. NEURAL NETWORKS

Z← X & Y

symbolic taught

subsymbolic self-learned

Desiderata for KGMS According to our Philosophy

No extra permanent data repository or database/DBMS

- Uses (possible multiple) <u>existing</u> company data repositories/databases
- Can query and update these – streaming into main memory for reasoning
- No data migration necessary

Multiple data models possible.

- Relational, graph, RDF, …
- Reasoning engine interprets all data relationally (by Datalog facts)

High expressive power of reasoning language; express at least:

- Full Datalog with full recursion and stratified negation
- Graph navigation
- Aggregate functions
- Description logics such as: DL-Lite (OWL 2 QL), EL, F-Logic Lite
- SPARQL under RDFS or OWL 2 QL Entailment Regimes

Good complexity and scalability

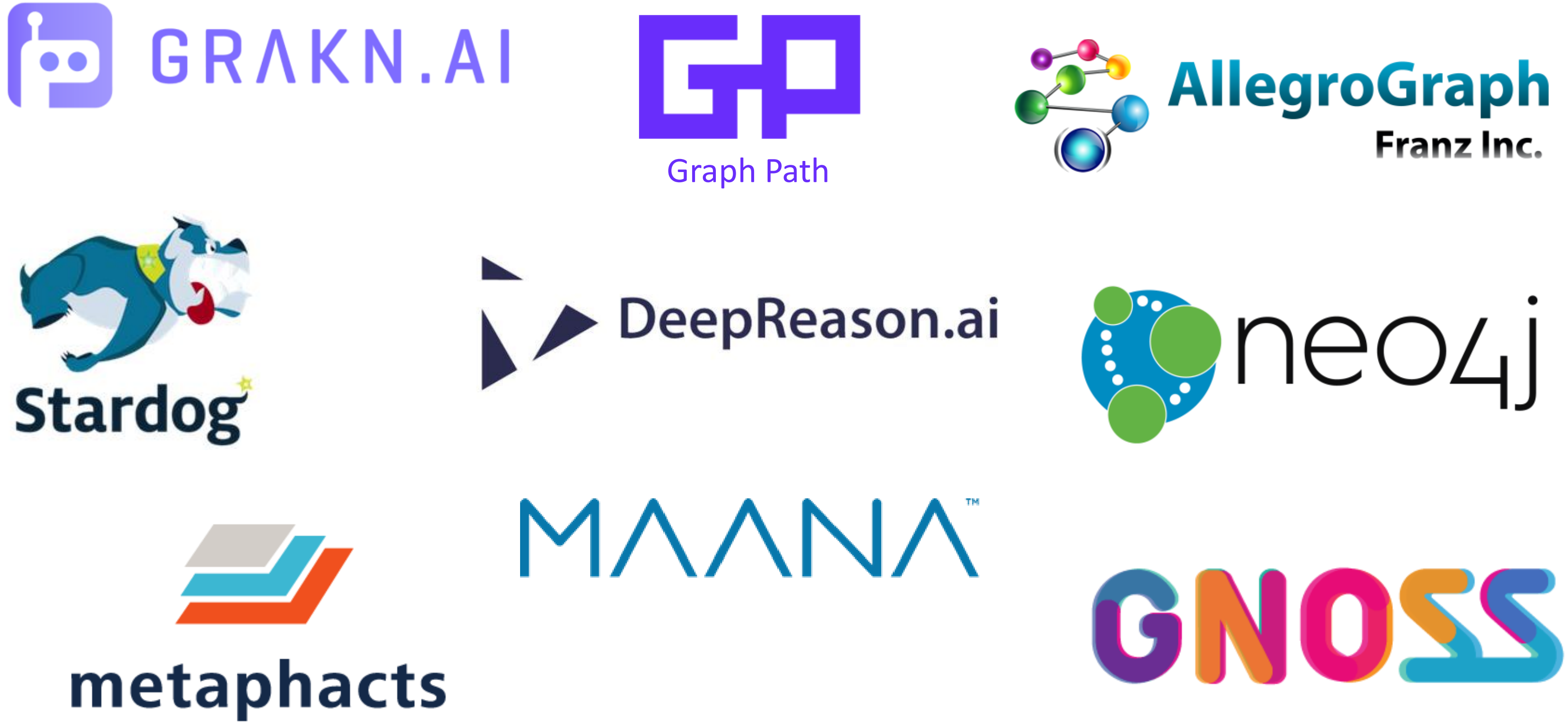
- Tractability guarantee for main formalism
- Highly efficient, and highly parallelizable language fragments

Support for machine learning, analytics, LLMs, and collaborative filtering

- APIs to standard ML and analytics packages and LLMs (do not reinvent the wheel)
- Provide system support for graph analysis (e.g. balanced separators), and typical functions such as *argmin* (with grad. desc.), *eigenvector*, *pagerank*, *simrank*, etc.

Knowledge Graph Management Systems

a diverse new field – many systems with different capabilities



Analysis along many dimensions possible

AllegroGraph Franz Inc.	Graph database supporting SPARQL and Prolog reasoning
GRAKN.AI	Apache Cassandra-based KGMS providing schema support based on the Entity Relationship model
	Knowledge Graph-as-a-Service
VADALOG	Data source-agnostic KGMS supporting ontological and recursive reasoning based on Datalog
neo4j	Leading graph database system
Stardog	RDF-based unifying data-integration platform
metaphacts	SPARQL 1.1-graph database-based end-user-oriented platform
MAANA	Azure-based computation-focused platform
GNOSS	RDF and OWL-based metadata management solution.

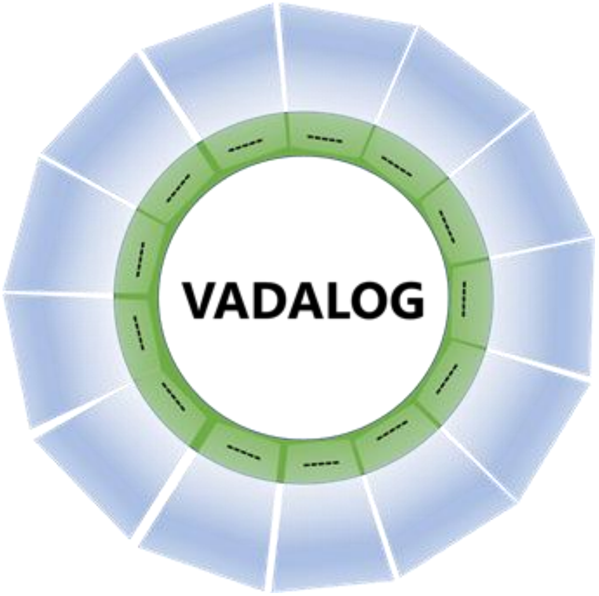
Migration necessary?

AllegroGraph Franz Inc.	Graph database supporting SPARQL and Prolog reasoning
GRAKN.AI	Apache Cassandra-based KGMS providing schema support based on the Entity Relationship model
GP	Knowledge Graph-as-a-Service
VADALOG	Data source-agnostic KGMS supportin recursive reasoning based on Datalog uses existing company DBMS for permanent storage
neo4j	Leading graph database system
Stardog	RDF-based unifying data-integration p uses existing company DBMS for permanent storage
metaphacts	SPARQL 1.1-graph database-based end-user-oriented platform
MAANA	Azure-based computation-focused platform
GNOSS	RDF and OWL-based metadata management solution.

Principle Data Format / Backend

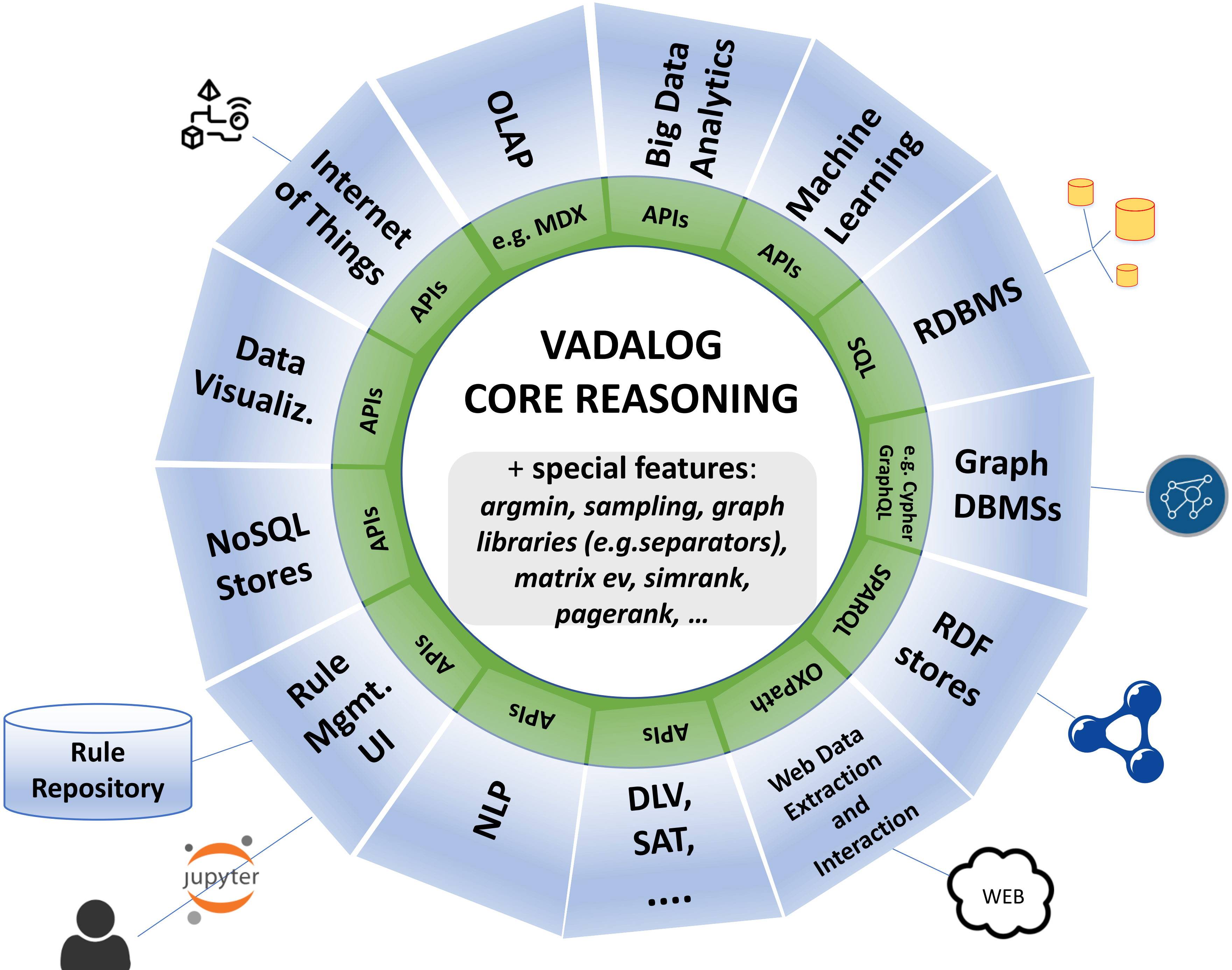
AllegroGraph Franz Inc.	Graph database sup nd Prolog reasoning	Graph
GRAKN.AI	Apache Cassandra-based KGMS providing sc based on the Entity Relationship model	Cassandra
	Knowledge Graph-as-a-Service	
VADALOG	Data source-agnostic KGMS supporting ontc recursive reasoning based on Datalog	Multiple
neo4j	Leading graph datab	Graph
Stardog	F data-integration platform	RDF
metaphacts	S database-based end-user-oriented P	RDF
MAANA	Azure-based computation-focused platform	Azure
GNOSS	I ed metadata management solution.	RDF

Vadalog KGMS Being Built at Oxford



Current Team Members

- VADA = **V**alue-**A**dded **DA**ta
- General architecture of VADALOG system
- Core reasoning language VADALOG = Warded Datalog + extensions
- Connectivity: Some plug-ins



VADALOG

CORE REASONING

+ special features:

argmin, sampling, graph libraries (e.g.separators), matrix ev, simrank, pagerank, …

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.)
- Data anonymization
- Duplicate handling
- Automated product configuration
- Conceptual Modeling (e.g., UML)

Object Creation
e.g. in web data extraction

DIADEM domain centric intelligent automated data extraction methodology

Wrapidity

PRODUCT	PRICE
Toshiba_Protege_cx	480
Dell_25416	360
Dell_23233	470
Acer_78987	390

Object Creation
e.g. in web data extraction

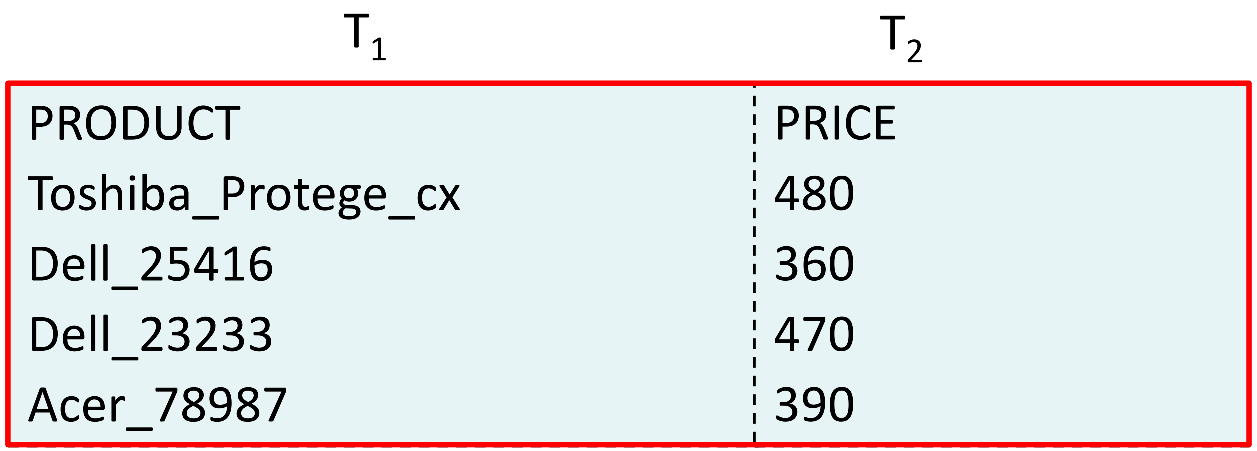
T_1	T_2
PRODUCT	PRICE
Toshiba_Protege_cx	480
Dell_25416	360
Dell_23233	470
Acer_78987	390

Object Creation
e.g. in web data extraction

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

Object Creation
e.g. in web data extraction

```
table($T_1$),
table($T_2$),
sameColor($T_1$,$T_2$),
isNeighbourRight($T_1$,$T_2$) $\rightarrow$
                    $\exists$T tablebox(T),
                       contains(T,$T_1$),
                       contains(T,$T_2$).
```



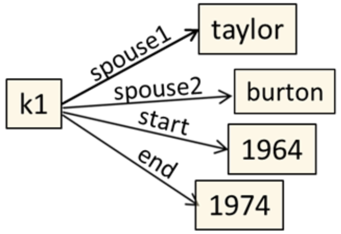
T_1	T_2
PRODUCT	PRICE
Toshiba_Protege_cx	480
Dell_25416	360
Dell_23233	470
Acer_78987	390

Reasoning with RDF – Foreign Key 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)
```

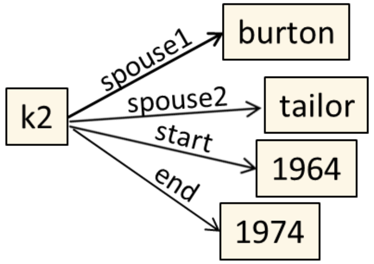


```
∀ u,v,x,y. married(u,v,x,y) → married(v,u,x,y)
```

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$_F$	
$\mathrm{funct}(hasTutor)$	$\forall X \forall Y \forall Z\ hasTutor(X,Y) \wedge hasTutor(X,Z) \rightarrow Y = Z$

[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/tractable 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.

Which are the Main Decidable Datalog$^{\pm}$ Languages?

- Guardedness: one body-atom that contains all the 8-variables

  supervisorOf(S,E), emp(E) → emp(S)

- Linearity: there exists only one atom in the body

  person(P) → ∃F fatherOf(F,P)
  fatherOf(F,P) → person(F)

- Weak-guardedness: guard only those variables that are **affected**, i.e., that may unify with null values [Calì & Kifer, 2008]

  Nontrivial example: F-logic Lite (→next page)

Nontrivial example of a weakly guarded set of rules:

F-Logic Lite, by [Cali & Kifer], VLDB 2006

(1) $\mathsf{member}(V, T) \leftarrow \mathsf{type}(O, A, T), \mathsf{data}(O, A, V).$

(2) $\mathsf{sub}(C_1, C_2) \leftarrow \mathsf{sub}(C_1, C_3), \mathsf{sub}(C_3, C_2).$

(3) $\mathsf{member}(O, C_1) \leftarrow \mathsf{member}(O, C), \mathsf{sub}(C, C_1).$

(4) $V = W \leftarrow \mathsf{data}(O, A, V), \mathsf{data}(O, A, W), \mathsf{funct}(A, O).$

Note that this is the only EGD in this axiomatization.

(5) $\mathsf{data}(O, A, V) \leftarrow \mathsf{mandatory}(A, O).$

Note that this is a TGD with an existential variable in the head (variable V; quantifiers are omitted).

(6) $\mathsf{type}(O, A, T) \leftarrow \mathsf{member}(O, C), \mathsf{type}(C, A, T).$

(7) $\mathsf{type}(C, A, T) \leftarrow \mathsf{sub}(C, C_1), \mathsf{type}(C_1, A, T).$

(8) $\mathsf{type}(C, A, T) \leftarrow \mathsf{type}(C, A, T_1),\ \mathsf{sub}(T_1, T).$

(9) $\mathsf{mandatory}(A, C) \leftarrow \mathsf{sub}(C, C_1), \mathsf{mandatory}(A, C_1).$

(10) $\mathsf{mandatory}(A, O) \leftarrow$
$\mathsf{member}(O, C), \mathsf{mandatory}(A, C).$

(11) $\mathsf{funct}(A, C) \leftarrow \mathsf{sub}(C, C_1),\ \mathsf{funct}(A, C_1).$

(12) $\mathsf{funct}(A, O) \leftarrow \mathsf{member}(O, C), \mathsf{funct}(A, C).$

(1) member(V, T) $\leftarrow$ type(O, A, T), data(O, A, V).

(2) sub(C_1, C_2) $\leftarrow$ sub(C_1, C_3), sub(C_3, C_2).

(3) member(O, C_1) $\leftarrow$ member(O, C), sub(C, C_1).

(4) $V = W$ $\leftarrow$ data(O, A, V), data(O, A, W), funct(A, O).

Note that this is the only EGD in this axiomatization.

(5) data(O, A, V) $\leftarrow$ mandatory(A, O).

Note that this is a TGD with an existential variable in the head (variable V; quantifiers are omitted).

(6) type(O, A, T) $\leftarrow$ member(O, C), type$_0$(C, A, T).

(7) type$_0$(C, A, T) $\leftarrow$ sub(C, C_1), type$_0$(C_1, A, T).

(8) type$_0$(C, A, T) $\leftarrow$ type$_0$(C, A, T_1), sub(T_1, T).

(9) mandatory$_0$(A, C) $\leftarrow$ sub(C, C_1), mandatory$_0$(A, C_1).

(10) mandatory(A, O) $\leftarrow$
member(O, C), mandatory$_0$(A, C).

(11) funct$_0$(A, C) $\leftarrow$ sub(C, C_1), funct$_0$(A, C_1).

(12) funct(A, O) $\leftarrow$ member(O, C), funct$_0$(A, C).

(1) $\mathsf{member}(V,T) \leftarrow \mathsf{type}(O,A,T), \mathsf{data}(O,A,V).$

(2) $\mathsf{sub}(C_1,C_2) \leftarrow \mathsf{sub}(C_1,C_3), \mathsf{sub}(C_3,C_2).$

(3) $\mathsf{member}(O,C_1) \leftarrow \mathsf{member}(O,C), \mathsf{sub}(C,C_1).$

(5) $\mathsf{data}(O,A,V) \leftarrow \mathsf{mandatory}(A,O).$

Note that this is a TGD with an existential variable in the head (variable V; quantifiers are omitted).

(6) $\mathsf{type}(O,A,T) \leftarrow \mathsf{member}(O,C), \mathsf{type}_0(C,A,T).$

(7) $\mathsf{type}_0(C,A,T) \leftarrow \mathsf{sub}(C,C_1), \mathsf{type}_0(C_1,A,T).$

(8) $\mathsf{type}_0(C,A,T) \leftarrow \mathsf{type}_0(C,A,T_1),\ \mathsf{sub}(T_1,T).$

(9) $\mathsf{mandatory}_0(A,C) \leftarrow \mathsf{sub}(C,C_1), \mathsf{mandatory}_0(A,C_1).$

(10) $\mathsf{mandatory}(A,O) \leftarrow$
$\mathsf{member}(O,C), \mathsf{mandatory}_0(A,C).$

(11) $\mathsf{funct}_0(A,C) \leftarrow \mathsf{sub}(C,C_1),\ \mathsf{funct}_0(A,C_1).$

(12) $\mathsf{funct}(A,O) \leftarrow \mathsf{member}(O,C), \mathsf{funct}_0(A,C).$

(1) $\mathsf{member}(V,T) \leftarrow \mathsf{type}(O,A,T), \mathsf{data}(O,A,V)$.

(2) $\mathsf{sub}(C_1,C_2) \leftarrow \mathsf{sub}(C_1,C_3), \mathsf{sub}(C_3,C_2)$.

(3) $\mathsf{member}(O,C_1) \leftarrow \mathsf{member}(O,C), \mathsf{sub}(C,C_1)$.

(5) $\mathsf{data}(O,A,V) \leftarrow \mathsf{mandatory}(A,O)$.

Note that this is a TGD with an existential variable in the head (variable V; quantifiers are omitted).

(6) $\mathsf{type}(O,A,T) \leftarrow \mathsf{member}(O,C), \mathsf{type}_0(C,A,T)$.

(7) $\mathsf{type}_0(C,A,T) \leftarrow \mathsf{sub}(C,C_1), \mathsf{type}_0(C_1,A,T)$.

(8) $\mathsf{type}_0(C,A,T) \leftarrow \mathsf{type}_0(C,A,T_1),\ \mathsf{sub}(T_1,T)$.

(9) $\mathsf{mandatory}_0(A,C) \leftarrow \mathsf{sub}(C,C_1), \mathsf{mandatory}_0(A,C_1)$.

(10) $\mathsf{mandatory}(A,O) \leftarrow$
$\mathsf{member}(O,C), \mathsf{mandatory}_0(A,C)$.

(11) $\mathsf{funct}_0(A,C) \leftarrow \mathsf{sub}(C,C_1),\ \mathsf{funct}_0(A,C_1)$.

(12) $\mathsf{funct}(A,O) \leftarrow \mathsf{member}(O,C), \mathsf{funct}_0(A,C)$.

(1) $\mathsf{member}(V, T) \leftarrow \mathsf{type}(O, A, T), \mathsf{data}(O, A, V).$

(2) $\mathsf{sub}(C_1, C_2) \leftarrow \mathsf{sub}(C_1, C_3), \mathsf{sub}(C_3, C_2).$

(3) $\mathsf{member}(O, C_1) \leftarrow \mathsf{member}(O, C), \mathsf{sub}(C, C_1).$

(5) $\mathsf{data}(O, A, V) \leftarrow \mathsf{mandatory}(A, O).$

Note that this is a TGD with an existential variable in the head (variable V; quantifiers are omitted).

(6) $\mathsf{type}(O, A, T) \leftarrow \mathsf{member}(O, C), \mathsf{type}_0(C, A, T).$

(7) $\mathsf{type}_0(C, A, T) \leftarrow \mathsf{sub}(C, C_1), \mathsf{type}_0(C_1, A, T).$

(8) $\mathsf{type}_0(C, A, T) \leftarrow \mathsf{type}_0(C, A, T_1),\ \mathsf{sub}(T_1, T).$

(9) $\mathsf{mandatory}_0(A, C) \leftarrow \mathsf{sub}(C, C_1), \mathsf{mandatory}_0(A, C_1).$

(10) $\mathsf{mandatory}(A, O) \leftarrow$
$\mathsf{member}(O, C), \mathsf{mandatory}_0(A, C).$

(11) $\mathsf{funct}_0(A, C) \leftarrow \mathsf{sub}(C, C_1),\ \mathsf{funct}_0(A, C_1).$

(12) $\mathsf{funct}(A, O) \leftarrow \mathsf{member}(O, C), \mathsf{funct}_0(A, C).$

(1) $\mathsf{member}(V, T) \leftarrow \mathsf{type}(O, A, T), \mathsf{data}(O, A, V).$

(2) $\mathsf{sub}(C_1, C_2) \leftarrow \mathsf{sub}(C_1, C_3), \mathsf{sub}(C_3, C_2).$

(3) $\mathsf{member}(O, C_1) \leftarrow \mathsf{member}(O, C), \mathsf{sub}(C, C_1).$

(5) $\mathsf{data}(O, A, V) \leftarrow \mathsf{mandatory}(A, O).$

Note that this is a TGD with an existential variable in the head (variable V; quantifiers are omitted).

(6) $\mathsf{type}(O, A, T) \leftarrow \mathsf{member}(O, C), \mathsf{type}_0(C, A, T).$

(7) $\mathsf{type}_0(C, A, T) \leftarrow \mathsf{sub}(C, C_1), \mathsf{type}_0(C_1, A, T).$

(8) $\mathsf{type}_0(C, A, T) \leftarrow \mathsf{type}_0(C, A, T_1),\ \mathsf{sub}(T_1, T).$

(9) $\mathsf{mandatory}_0(A, C) \leftarrow \mathsf{sub}(C, C_1), \mathsf{mandatory}_0(A, C_1).$

(10) $\mathsf{mandatory}(A, O) \leftarrow$
$\mathsf{member}(O, C), \mathsf{mandatory}_0(A, C).$

(11) $\mathsf{funct}_0(A, C) \leftarrow \mathsf{sub}(C, C_1),\ \mathsf{funct}_0(A, C_1).$

(12) $\mathsf{funct}(A, O) \leftarrow \mathsf{member}(O, C), \mathsf{funct}_0(A, C).$

(1) $\mathsf{member}(V, T) \leftarrow \mathsf{type}(O, A, T), \mathsf{data}(O, A, V).$

(2) $\mathsf{sub}(C_1, C_2) \leftarrow \mathsf{sub}(C_1, C_3), \mathsf{sub}(C_3, C_2).$

(3) $\mathsf{member}(O, C_1) \leftarrow \mathsf{member}(O, C), \mathsf{sub}(C, C_1).$

(5) $\mathsf{data}(O, A, V) \leftarrow \mathsf{mandatory}(A, O).$

Note that this is a TGD with an existential variable in the head (variable V; quantifiers are omitted).

(6) $\mathsf{type}(O, A, T) \leftarrow \mathsf{member}(O, C), \mathsf{type}_0(C, A, T).$

(7) $\mathsf{type}_0(C, A, T) \leftarrow \mathsf{sub}(C, C_1), \mathsf{type}_0(C_1, A, T).$

(8) $\mathsf{type}_0(C, A, T) \leftarrow \mathsf{type}_0(C, A, T_1),\ \mathsf{sub}(T_1, T).$

(9) $\mathsf{mandatory}_0(A, C) \leftarrow \mathsf{sub}(C, C_1), \mathsf{mandatory}_0(A, C_1).$

(10) $\mathsf{mandatory}(A, O) \leftarrow$
$\mathsf{member}(O, C), \mathsf{mandatory}_0(A, C).$

(11) $\mathsf{funct}_0(A, C) \leftarrow \mathsf{sub}(C, C_1),\ \mathsf{funct}_0(A, C_1).$

(12) $\mathsf{funct}(A, O) \leftarrow \mathsf{member}(O, C), \mathsf{funct}_0(A, C).$

(1) $\mathsf{member}(V, T) \leftarrow \mathsf{type}(O, A, T), \mathsf{data}(O, A, V).$
(2) $\mathsf{sub}(C_1, C_2) \leftarrow \mathsf{sub}(C_1, C_3), \mathsf{sub}(C_3, C_2).$
(3) $\mathsf{member}(O, C_1) \leftarrow \mathsf{member}(O, C), \mathsf{sub}(C, C_1).$

(5) $\mathsf{data}(O, A, V) \leftarrow \mathsf{mandatory}(A, O).$
Note that this is a TGD with an existential variable in the head (variable V; quantifiers are omitted).
(6) $\mathsf{type}(O, A, T) \leftarrow \mathsf{member}(O, C), \mathsf{type}_0(C, A, T).$
(7) $\mathsf{type}_0(C, A, T) \leftarrow \mathsf{sub}(C, C_1), \mathsf{type}_0(C_1, A, T).$
(8) $\mathsf{type}_0(C, A, T) \leftarrow \mathsf{type}_0(C, A, T_1),\ \mathsf{sub}(T_1, T).$
(9) $\mathsf{mandatory}_0(A, C) \leftarrow \mathsf{sub}(C, C_1), \mathsf{mandatory}_0(A, C_1).$
(10) $\mathsf{mandatory}(A, O) \leftarrow$
$\mathsf{member}(O, C), \mathsf{mandatory}_0(A, C).$
(11) $\mathsf{funct}_0(A, C) \leftarrow \mathsf{sub}(C, C_1),\ \mathsf{funct}_0(A, C_1).$
(12) $\mathsf{funct}(A, O) \leftarrow \mathsf{member}(O, C), \mathsf{funct}_0(A, C).$

(1) $\mathsf{member}(V, T) \leftarrow \mathsf{type}(O, A, T), \mathsf{data}(O, A, V).$

(2) $\mathsf{sub}(C_1, C_2) \leftarrow \mathsf{sub}(C_1, C_3), \mathsf{sub}(C_3, C_2).$

(3) $\mathsf{member}(O, C_1) \leftarrow \mathsf{member}(O, C), \mathsf{sub}(C, C_1).$

(5) $\mathsf{data}(O, A, V) \leftarrow \mathsf{mandatory}(A, O).$

Note that this is a TGD with an existential variable in the head (variable V; quantifiers are omitted).

(6) $\mathsf{type}(O, A, T) \leftarrow \mathsf{member}(O, C), \mathsf{type}_0(C, A, T).$

(7) $\mathsf{type}_0(C, A, T) \leftarrow \mathsf{sub}(C, C_1), \mathsf{type}_0(C_1, A, T).$

(8) $\mathsf{type}_0(C, A, T) \leftarrow \mathsf{type}_0(C, A, T_1),\ \mathsf{sub}(T_1, T).$

(9) $\mathsf{mandatory}_0(A, C) \leftarrow \mathsf{sub}(C, C_1), \mathsf{mandatory}_0(A, C_1).$

(10) $\mathsf{mandatory}(A, O) \leftarrow$
$\mathsf{member}(O, C), \mathsf{mandatory}_0(A, C).$

(11) $\mathsf{funct}_0(A, C) \leftarrow \mathsf{sub}(C, C_1),\ \mathsf{funct}_0(A, C_1).$

(12) $\mathsf{funct}(A, O) \leftarrow \mathsf{member}(O, C), \mathsf{funct}_0(A, C).$

FINISHED!

(1) $\mathsf{member}(V, T) \leftarrow \mathsf{type}(O, A, T), \boxed{\mathsf{data}(O, A, V)}.$

(2) $\mathsf{sub}(C_1, C_2) \leftarrow \mathsf{sub}(C_1, C_3), \mathsf{sub}(C_3, C_2).$

(3) $\mathsf{member}(O, C_1) \leftarrow \boxed{\mathsf{member}(O, C)}, \mathsf{sub}(C, C_1).$

(5) $\mathsf{data}(O, A, V) \leftarrow \boxed{\mathsf{mandatory}(A, O).}$

Note that this is a TGD with an existential variable in the head (variable V; quantifiers are omitted).

(6) $\mathsf{type}(O, A, T) \leftarrow \boxed{\mathsf{member}(O, C)}, \mathsf{type}_0(C, A, T).$

(7) $\mathsf{type}_0(C, A, T) \leftarrow \mathsf{sub}(C, C_1), \mathsf{type}_0(C_1, A, T).$

(8) $\mathsf{type}_0(C, A, T) \leftarrow \mathsf{type}_0(C, A, T_1),\ \mathsf{sub}(T_1, T).$

(9) $\mathsf{mandatory}_0(A, C) \leftarrow \mathsf{sub}(C, C_1), \mathsf{mandatory}_0(A, C_1).$

(10) $\mathsf{mandatory}(A, O) \leftarrow$
$\boxed{\mathsf{member}(O, C)}, \mathsf{mandatory}_0(A, C).$

(11) $\mathsf{funct}_0(A, C) \leftarrow \mathsf{sub}(C, C_1),\ \mathsf{funct}_0(A, C_1).$

(12) $\mathsf{funct}(A, O) \leftarrow \boxed{\mathsf{member}(O, C)}, \mathsf{funct}_0(A, C).$

Stratified Weakly Guarded Datalog$^{\pm}$

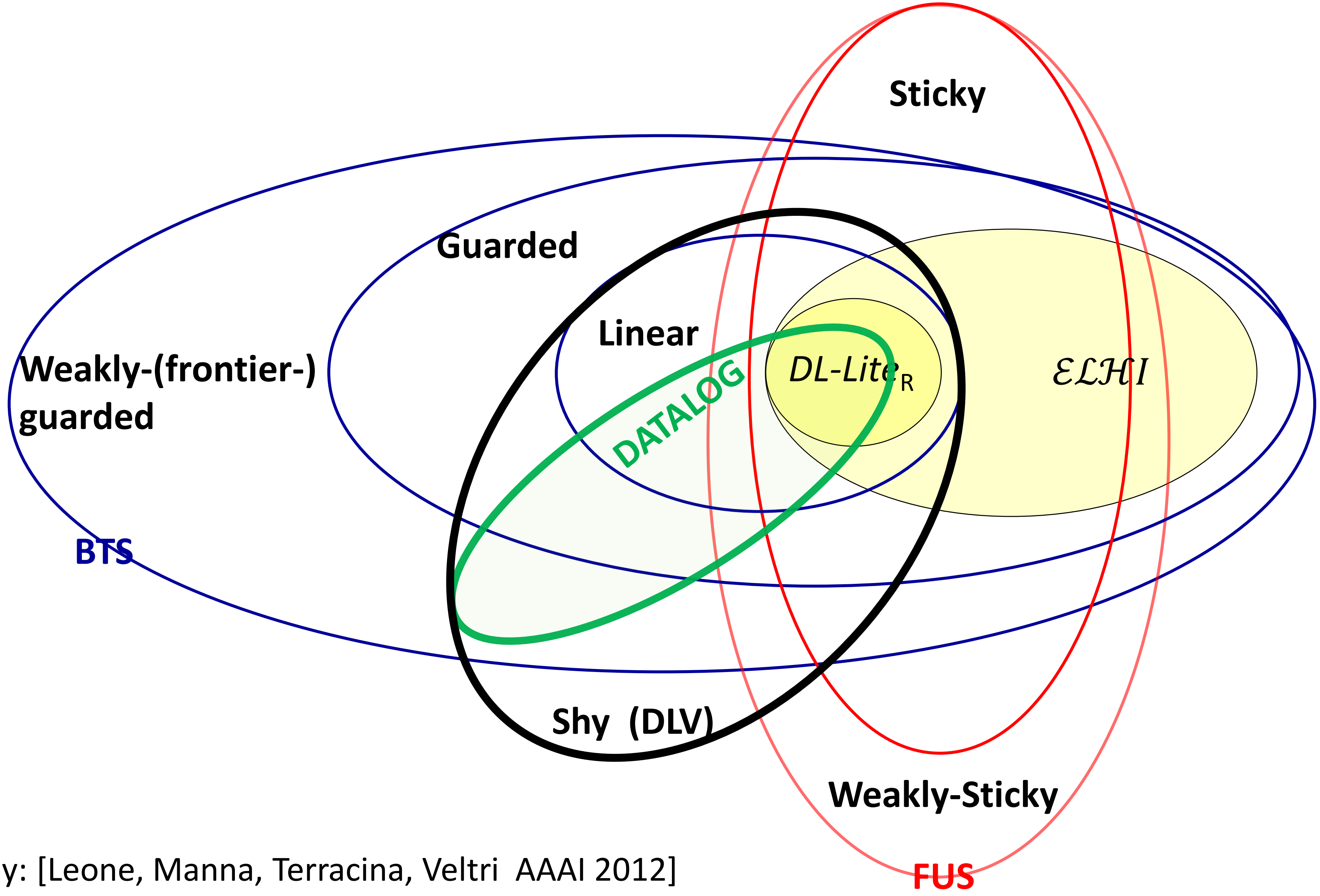
Modal Logics	DL Lite	ER, UML	DB Dep Theory (FDs,IDs)	F-Logic Lite	FO over data	CORE-SPARQL	DX (w.a.)	Datalog$^{\neg}$
✓	✓	☑	☑	✓	✓	✓	✓	✓

Legend:

✓**: captures fully**

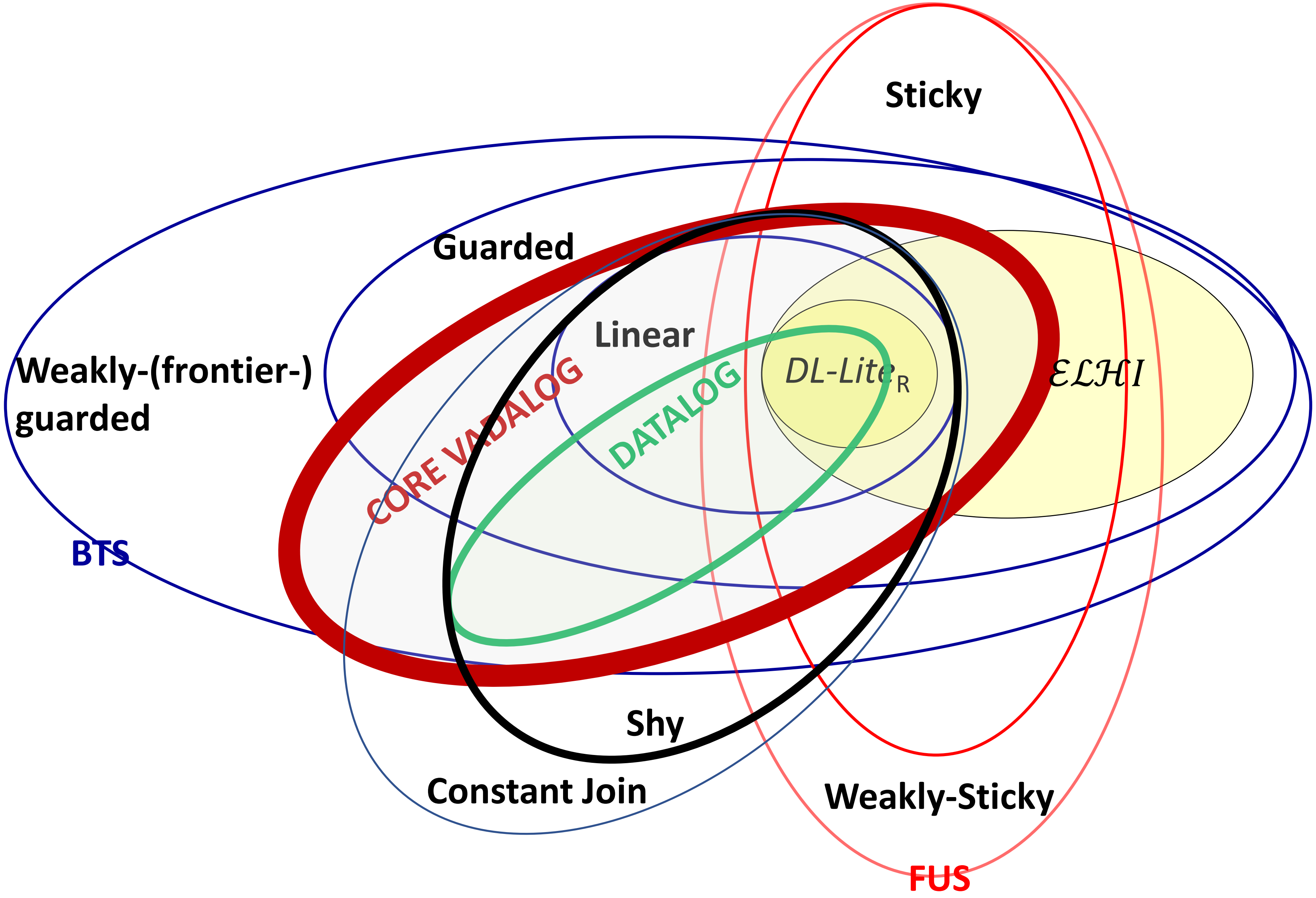
☑ **: captures large relevant decidable fragments**

Main Decidable Datalog$^{\pm}$ Languages

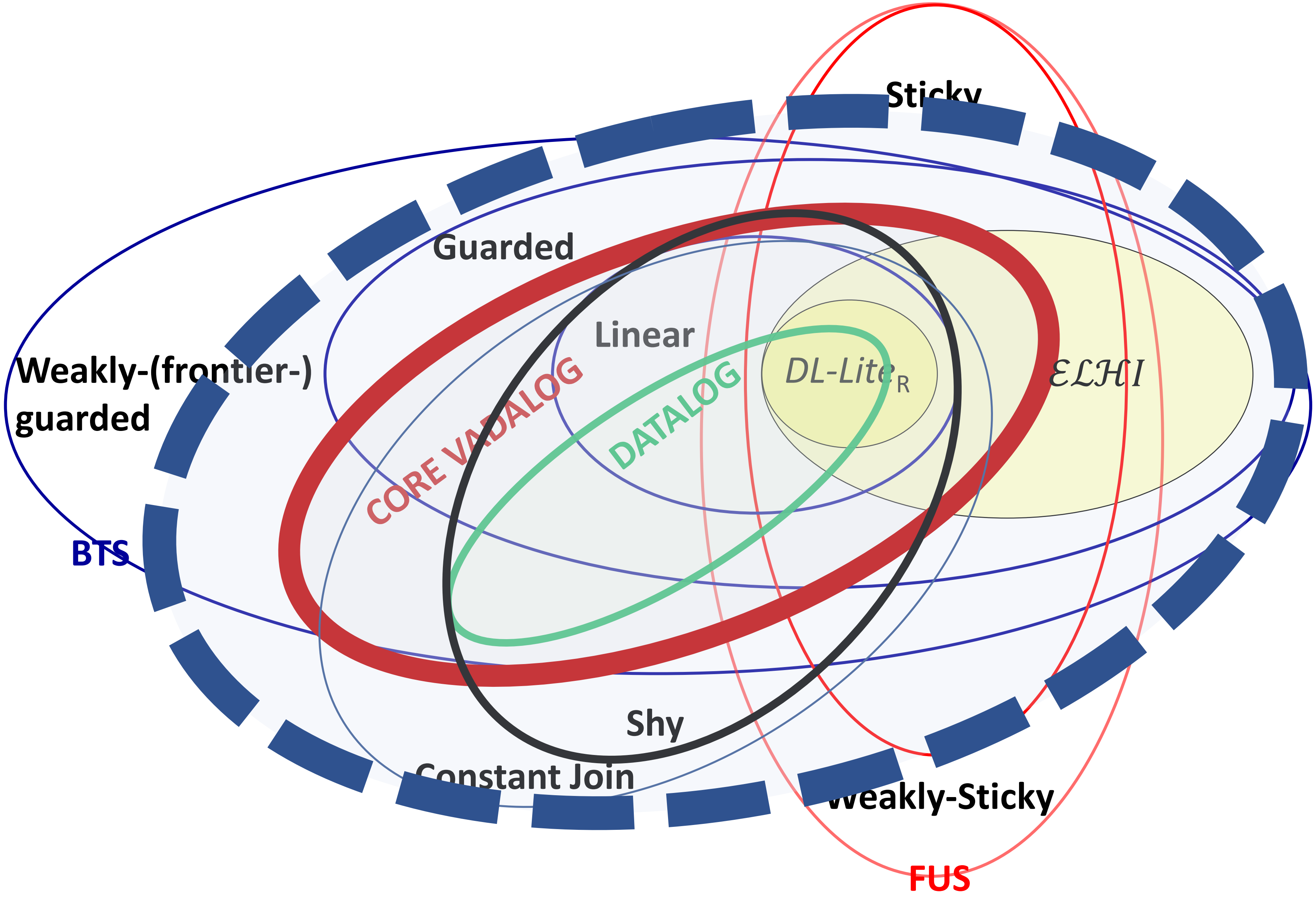


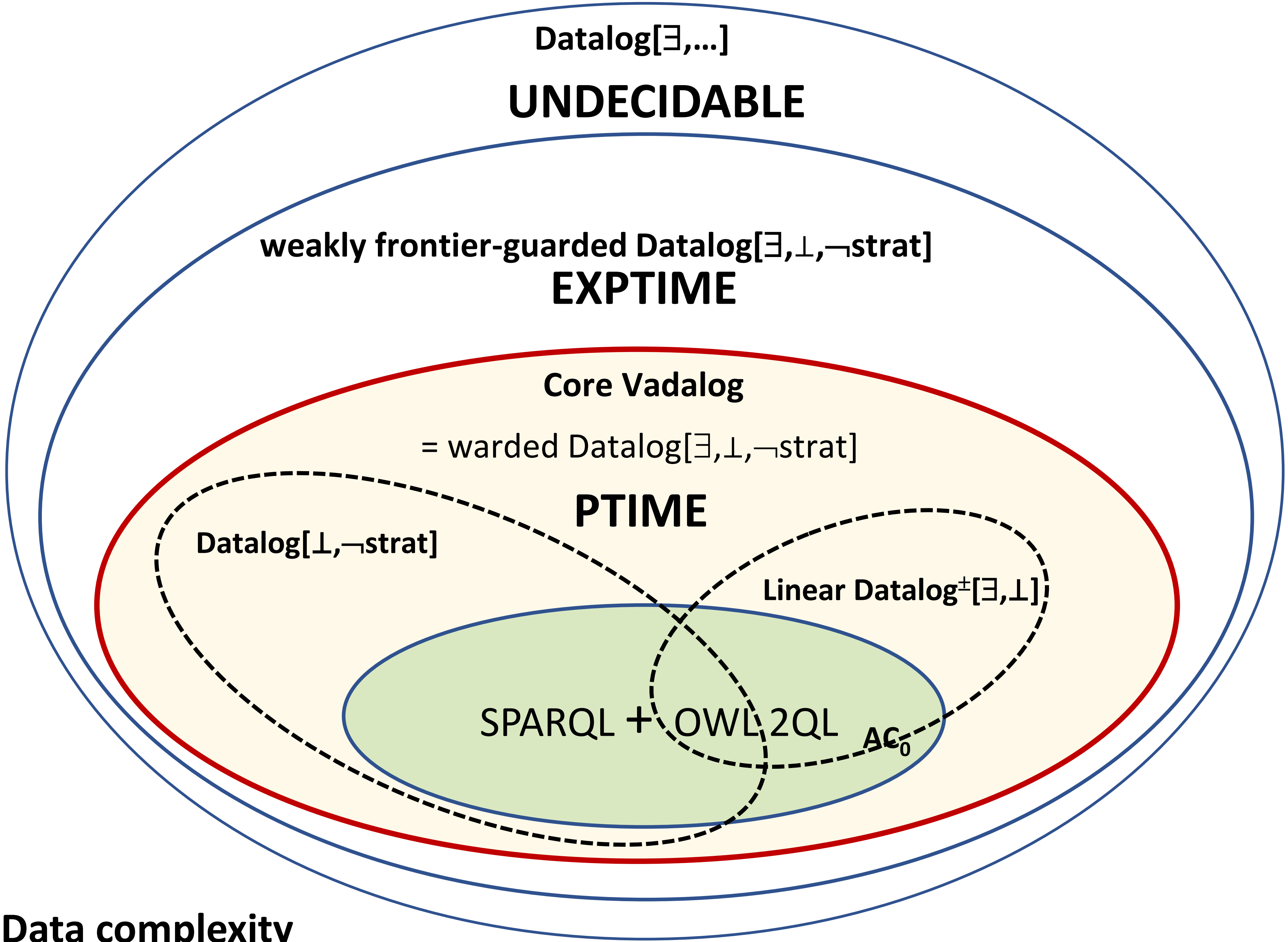
Shy: [Leone, Manna, Terracina, Veltri AAAI 2012]

Main Decidable Datalog$^{\pm}$ Languages



Future Plan – Challenge: PTIME Data Complexity





Datalog[∃,…]

UNDECIDABLE

weakly frontier-guarded Datalog[∃,⊥,¬strat]

EXPTIME

Core Vadalog

= warded Datalog[∃,⊥,¬strat]

PTIME

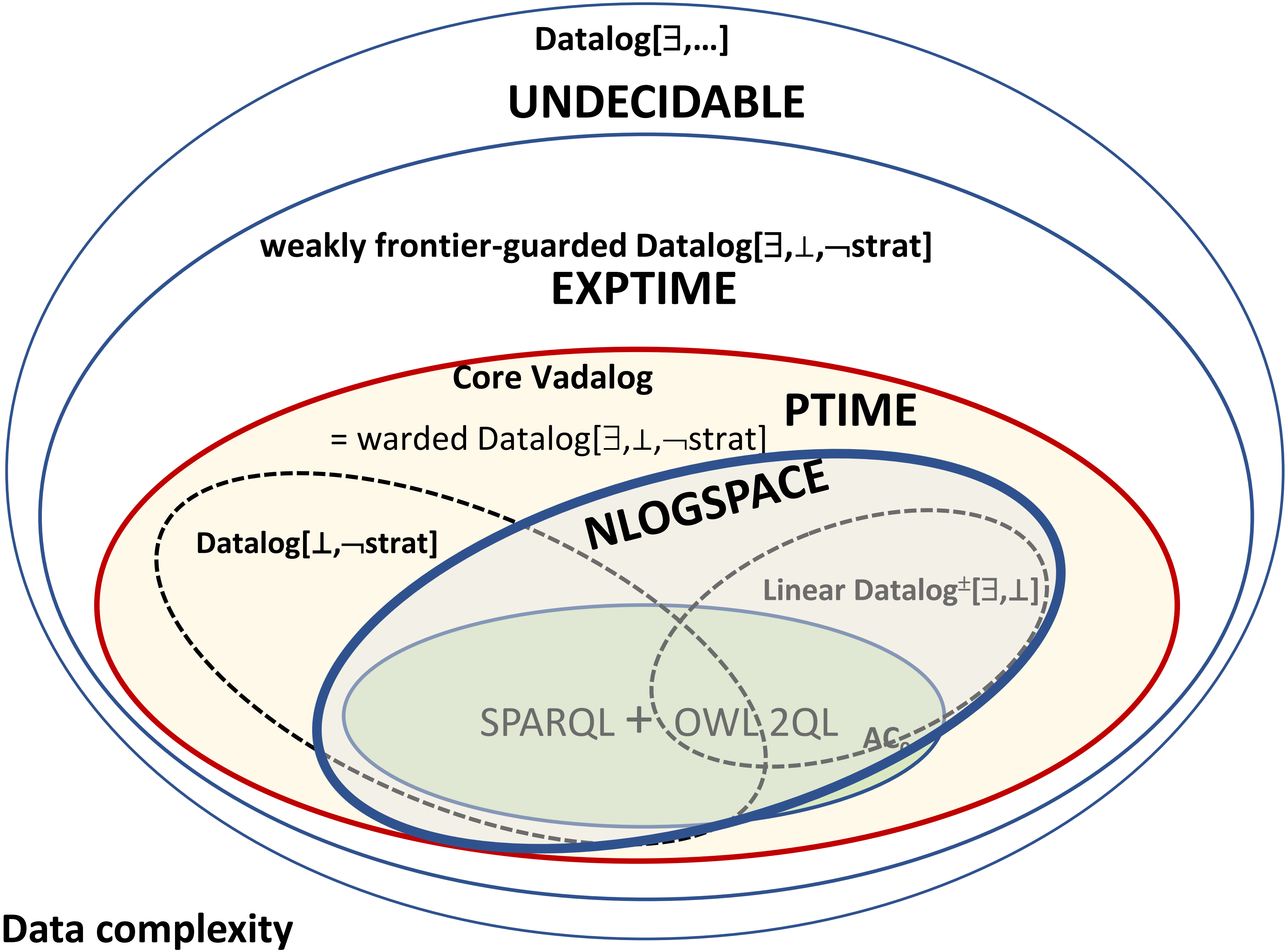
Datalog[⊥,¬strat]

Linear Datalog$^{\pm}$[∃,⊥]

SPARQL + OWL 2QL

AC_0

Data complexity



Datalog[∃,...]

UNDECIDABLE

weakly frontier-guarded Datalog[∃,⊥,¬strat]

EXPTIME

Core Vadalog

PTIME

= warded Datalog[∃,⊥,¬strat]

NLOGSPACE

Datalog[⊥,¬strat]

Linear Datalog$^{\pm}$[∃,⊥]

SPARQL + OWL 2QL

AC$_0$

Data complexity

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]

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

1. **Symmetry rule for marriage intervals (RDF):**

$$
\begin{aligned}
&\mathtt{spouse1}(\underline{\mathtt{x}},\mathtt{y1}) \wedge \mathtt{spouse2}(\underline{\mathtt{x}},\mathtt{y2}) \wedge \\
&\mathtt{start}(\underline{\mathtt{x}},\mathtt{y3}) \wedge \mathtt{end}(\underline{\mathtt{x}},\mathtt{y4}) \rightarrow \\
&\qquad\qquad \exists \mathtt{v}.\ \mathtt{spouse2}(\underline{\mathtt{v}},\mathtt{y1}) \wedge \mathtt{spouse1}(\underline{\mathtt{v}},\mathtt{y2}) \wedge \\
&\qquad\qquad\qquad \mathtt{start}(\underline{\mathtt{v}},\mathtt{y3}) \wedge \mathtt{end}(\underline{\mathtt{v}},\mathtt{y4})
\end{aligned}
$$

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

DL-Lite TBox	Representation in Vadalog
DL-Lite$_{core}$	
professor v 9*teachesTo*	8X *professor*(X) → **9**Y *teachesTo*(X,Y)
professor v :*student*	8X *professor*(X) ∧ *student*(X) → **?**
DL-Lite$_{R}$ (OWL 2 QL)	
hasTutor^{i} v *teachesTo*	8X8Y *hasTutor*(X,Y) → *teachesTo*(Y,X)

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

3. Creation of tuple-identifiers for bag semantics

R(a,b)
R(a,b)
R(a,c)
R(b,c)

S(a,c)
S(a,d)
S(b,c)

R(x,y,u) & S(x,z) → T(x)

T(a)
T(a)
T(a)
T(a)
T(a)
T(a)
T(b)

Bag semantics for Datalog by [Mumick & Shmueli 1992]

In case of recursion, possibly infinitely many duplicates…

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

use tuple-identifiers

$R(a,b,u_1)$
$R(a,b,u_2)$
$R(a,c,u_3)$
$R(b,c,u_4)$

$S(a,c,u_5)$
$S(a,d,u_6)$
$S(b,c,u_7)$

$$R(x,y,v) \,\&\, S(x,z,w) \rightarrow \exists u\, T(x,u)$$

$T(a,u_8)$
$T(a,u_9)$
$T(a,u_{10})$
$T(a,u_{11})$
$T(a,u_{12})$
$T(a,u_{13})$
$T(b,u_{14})$

This simple trick does the job:

Theorem: Datalog bag semantics can be faithfully emulated in Vadalog by using TIDs. [Bertossi, G.Pichler, ICDT'19]

Corollary: Deciding multiplicity of tuple is PTIME (data complexity)

Theorem

Vadalog has polynomial-time data complexity and can express:

- Datalog with full recursion and stratified negation

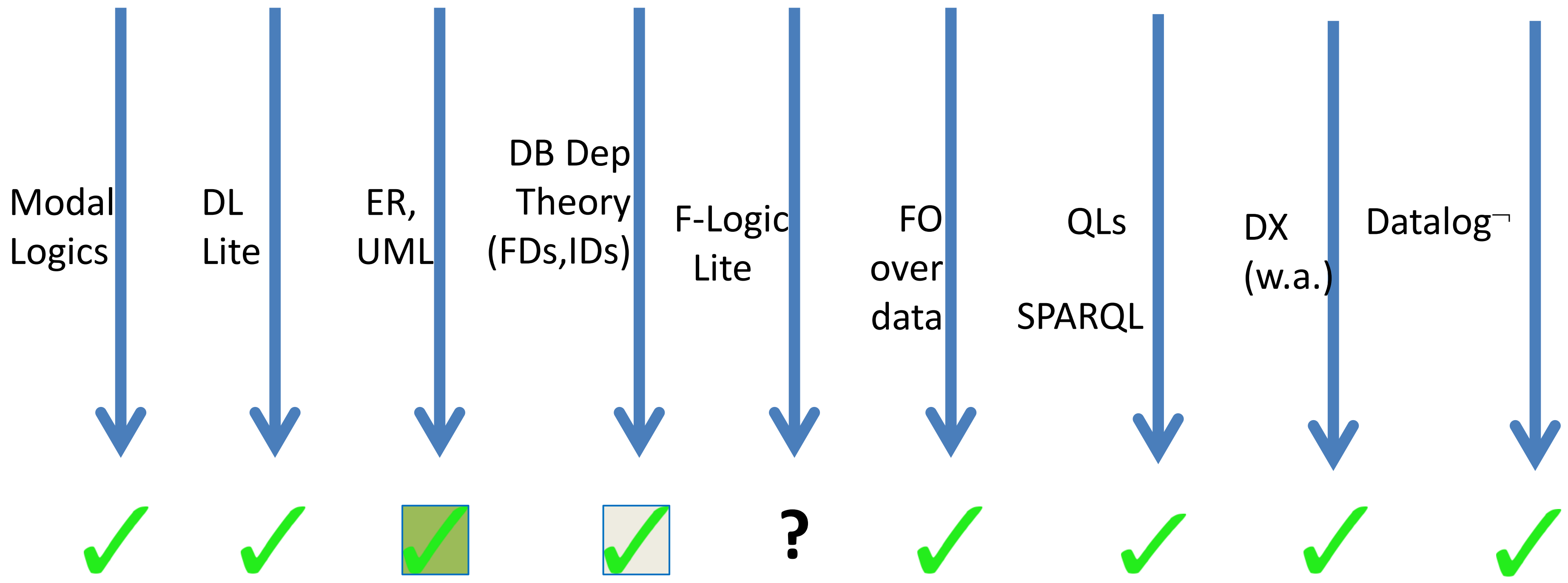
- Description logics: DL-Lite Family, in particular, OWL 2 QL, EL, F-Logic Lite

- Datalog under the bag semantics

- SPARQL under RDFS and OWL 2 QL Entailment Regimes

Moreover: All queries of iBench can be expressed in Vadalog!

VADALOG

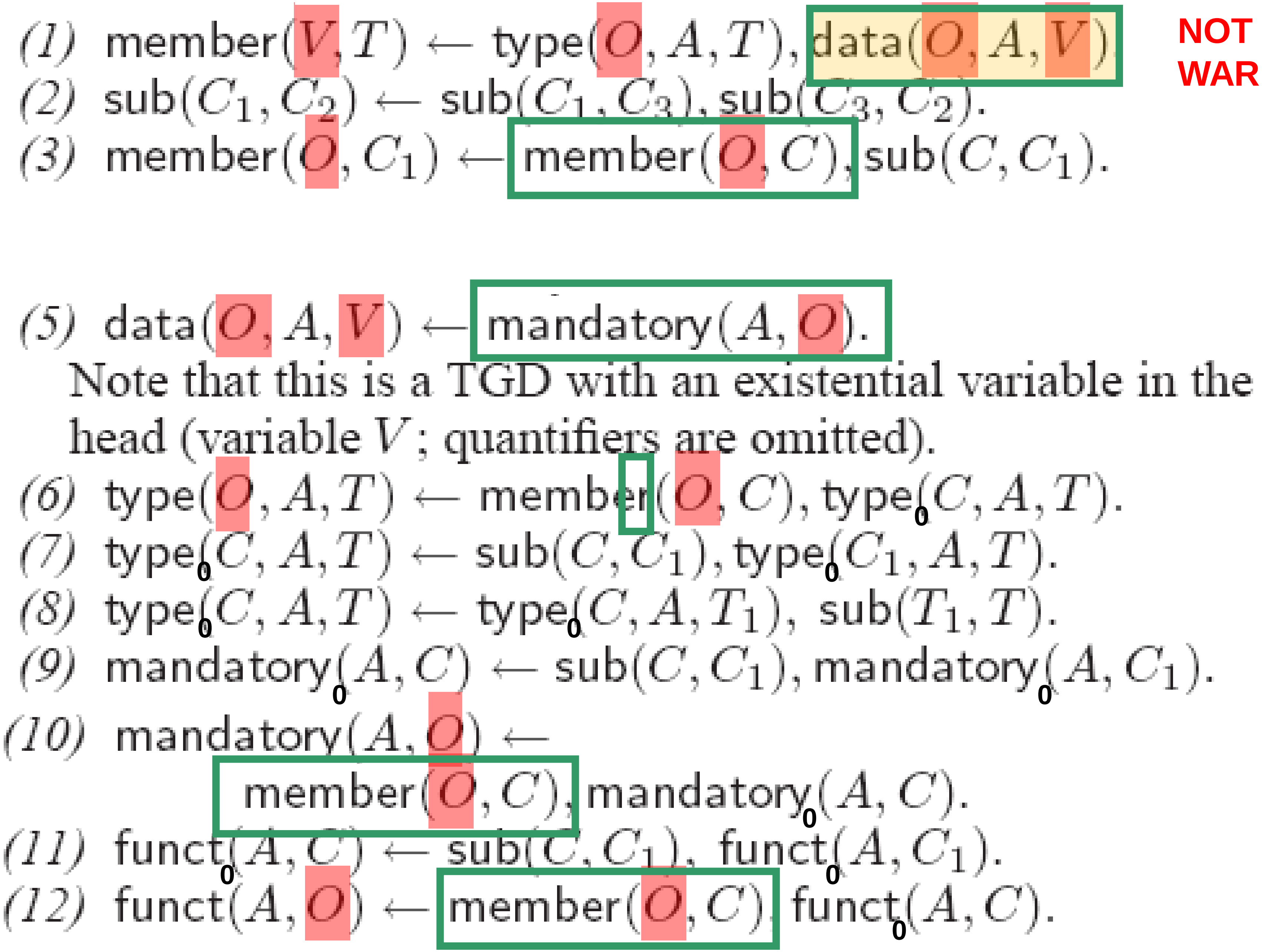


Modal Logics	DL Lite	ER, UML	DB Dep Theory (FDs,IDs)	F-Logic Lite	FO over data	QLs SPARQL	DX (w.a.)	Datalog$^\neg$
✓	✓	☑	☑	?	✓	✓	✓	✓

Legend:

✓: **captures fully**

☑ **: captures large relevant decidable fragments**



(1) $\mathsf{member}(V, T) \leftarrow \mathsf{type}(O, A, T), \boxed{\mathsf{data}(O, A, V)}$

(2) $\mathsf{sub}(C_1, C_2) \leftarrow \mathsf{sub}(C_1, C_3), \mathsf{sub}(C_3, C_2).$

(3) $\mathsf{member}(O, C_1) \leftarrow \boxed{\mathsf{member}(O, C)}, \mathsf{sub}(C, C_1).$

(5) $\mathsf{data}(O, A, V) \leftarrow \boxed{\mathsf{mandatory}(A, O).}$

Note that this is a TGD with an existential variable in the head (variable V; quantifiers are omitted).

(6) $\mathsf{type}(O, A, T) \leftarrow \mathsf{member}(O, C), \mathsf{type}_0(C, A, T).$

(7) $\mathsf{type}_0(C, A, T) \leftarrow \mathsf{sub}(C, C_1), \mathsf{type}_0(C_1, A, T).$

(8) $\mathsf{type}_0(C, A, T) \leftarrow \mathsf{type}_0(C, A, T_1), \ \mathsf{sub}(T_1, T).$

(9) $\mathsf{mandatory}_0(A, C) \leftarrow \mathsf{sub}(C, C_1), \mathsf{mandatory}_0(A, C_1).$

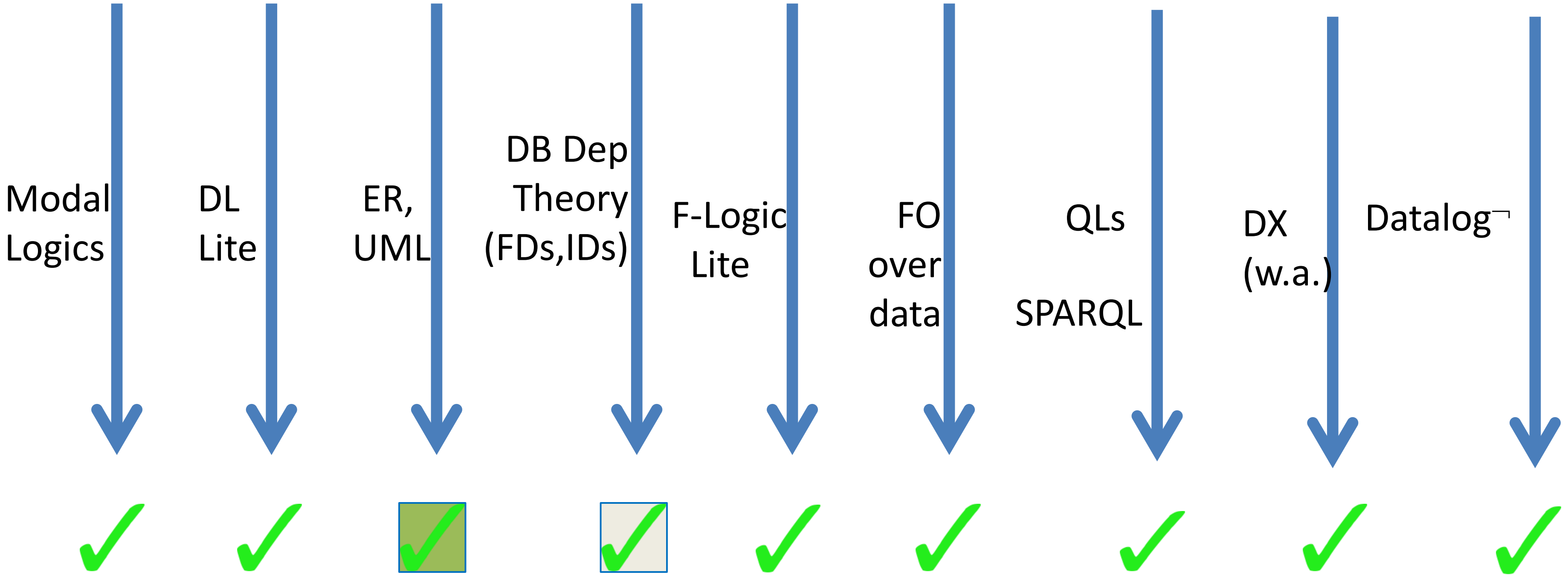
(10) $\mathsf{mandatory}(A, O) \leftarrow$
$\boxed{\mathsf{member}(O, C)}, \mathsf{mandatory}_0(A, C).$

(11) $\mathsf{funct}_0(A, C) \leftarrow \mathsf{sub}(C, C_1), \ \mathsf{funct}_0(A, C_1).$

(12) $\mathsf{funct}(A, O) \leftarrow \boxed{\mathsf{member}(O, C)}, \mathsf{funct}_0(A, C).$

NOT A WARD!

VADALOG



Modal Logics ✓

DL Lite ✓

ER, UML ☑

DB Dep Theory (FDs,IDs) ☑

F-Logic Lite ✓

FO over data ✓

QLs SPARQL ✓

DX (w.a.) ✓

$\text{Datalog}^{\neg}$ ✓

Legend:

✓: captures fully

☑ : captures large relevant decidable fragments

Further 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 weights. Marginal weights for derived facts will be computed assuming independence.

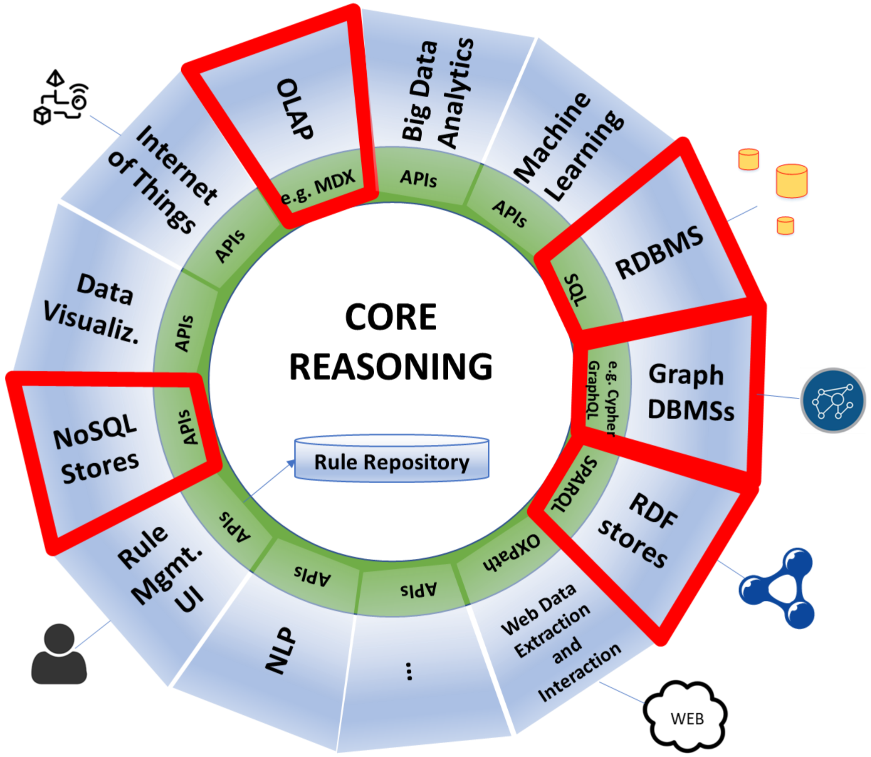
Equality (EGDs, functional dependencies) if non-conflicting.

Rules can be uncertain

```
@weight(0.6)  company(C) → ∃C1 own(C,C1).
@weight(0.5)  own(C,S), holding(C) → subsidiary(S).
```

- A Soft Vadalog rule has a **weight**

- Similar to Markov Logic Network, but Soft Vadalog
  - ➢ is not full First Order Logic
  - ➢ allows recursive definitions
  - ➢ has unrestricted domain



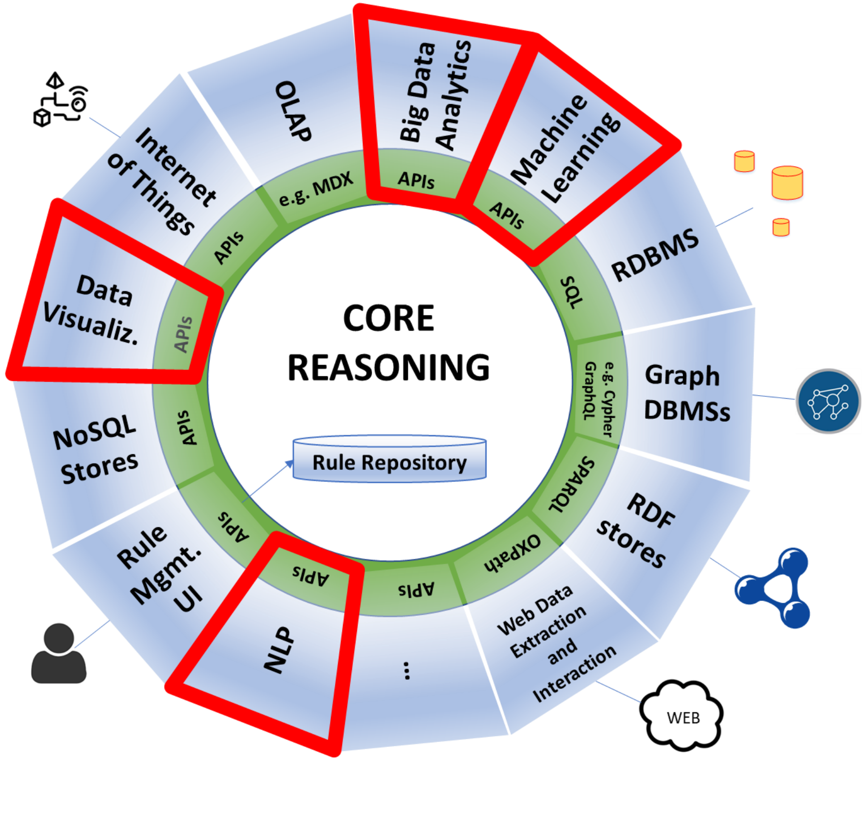
Database Interface

```
@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

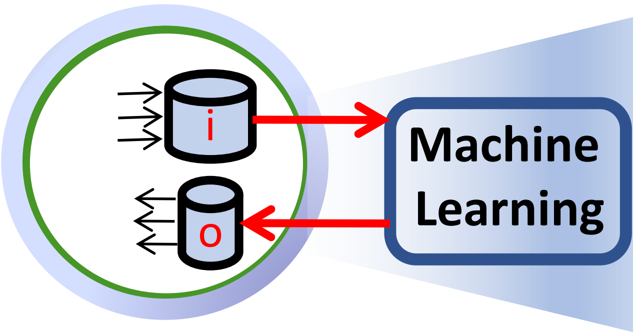
We are currently experimenting with different tools and different types of interfaces and interactions.

WEKA The University of Waikato

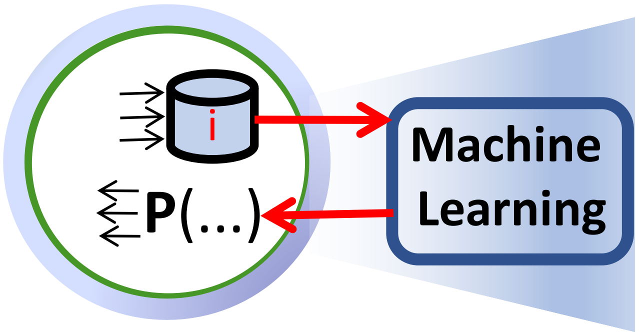
R

TensorFlow

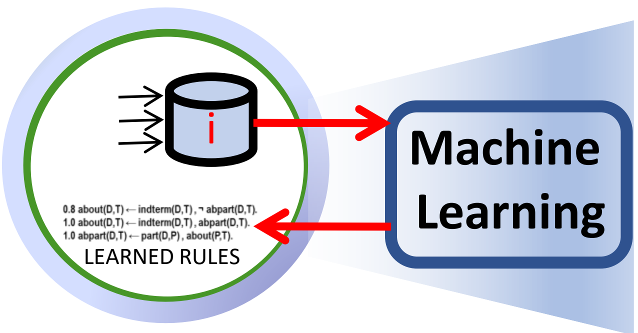
Interaction Model 1



Interaction Model 2

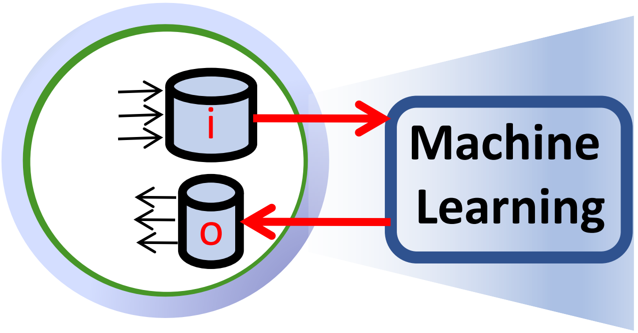


Interaction Model 3



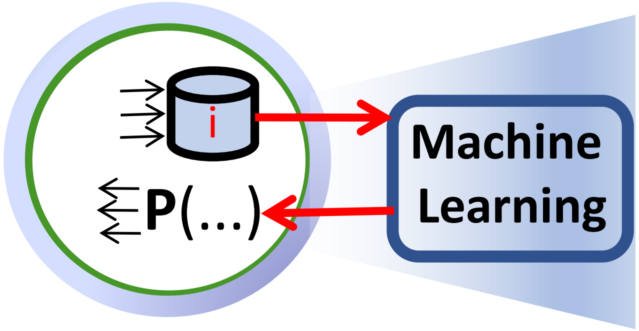
0.8 about(D,T) ← indterm(D,T) , ¬ abpart(D,T).
1.0 about(D,T) ← indterm(D,T) , abpart(D,T).
1.0 abpart(D,T) ← part(D,P) , about(P,T).
LEARNED RULES

Interaction Model 1



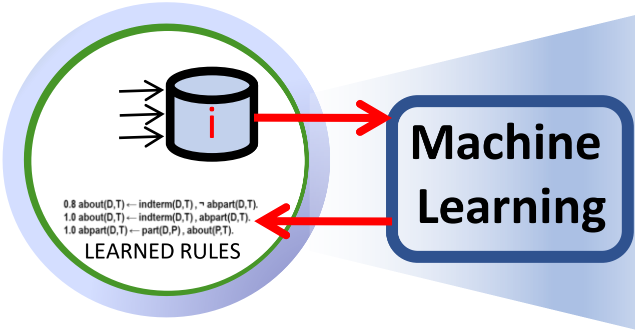
- We prepare a relation as ML input.
- ML sw classifies facts and sends them into the core reasoning system.

Interaction Model 2

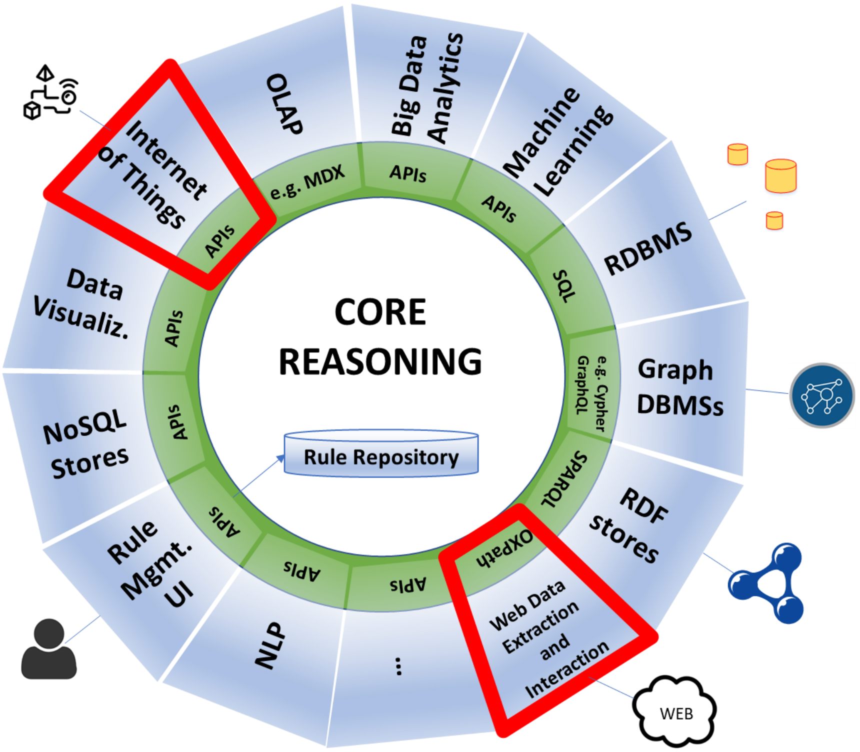


- ML package acts as a special predicate.
- Called by the core reasoning system.

Interaction Model 3


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

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



Web Data Extraction & IoT

Interfacing KG to OXPath;
Binding OXPath to Datalog

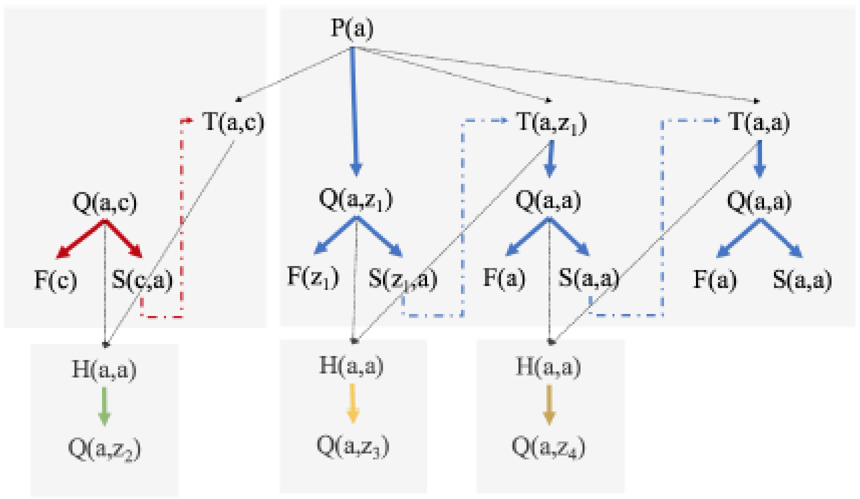
```
@qbind("Own", "oxpath",
            "doc(‘http://company_register.com/ownerships’)
            /descendant::field()[1]/{$1}
            /following::a[.#=’Search’]/{click/}
            /(//a[.#=’Next’]/ {click/})*
            //div[@class=’c’]: [./span[1]:][./span[3]:]")
```

[Furche, T., Gottlob, G., Grasso, G., Schallhart, C., & Sellers, A. (2013).
OXPath: A language for scalable data extraction, automation, and crawling on the deep web.
The VLDB Journal, *22*(1), 47-72. ".]

Core Algorithms

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

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



For more details see Luigi Bellomarini, Emanuel Sallinger, Georg Gottlob: The Vadalog System: *Datalog-based Reasoning for Knowledge Graphs*. PVLDB 11(9) 2018

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

PAPER ON THE VADALOG LANGUAGE

- Marcelo Arenas, Georg Gottlob, Andreas Pieris: *Expressive languages for querying the semantic web*.
  ACM TODS 13:1-45, 2018.

PAPERS ON 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*.
  PVLDB 11(9) 2018.

…

Some Applications

with two special partners/customers

Banca d’Italia
Central Bank of Italy

Collaboration

1. Company Control	**new approaches to classical problems – when does a company control another company?**
2. Close Links	**understanding whether companies are “too close” in terms of mutual stock participation for different purposes, e.g., for loan granting**
3. Detection of Family Business	**identifying families along with their ownerships, i.e., considering the family as the elementary control unit**
4. Anonymization of Confidential Data	**deciding whether a dataset respects complex confidentiality criteria (e.g., ISTAT) before publication and, if not, make it anonymous**
5. Hybrid Data Science Pipelines	**with different data sources, machine learning frameworks, programming languages, …**

… more applications that we cannot talk about at this point

Meltwater

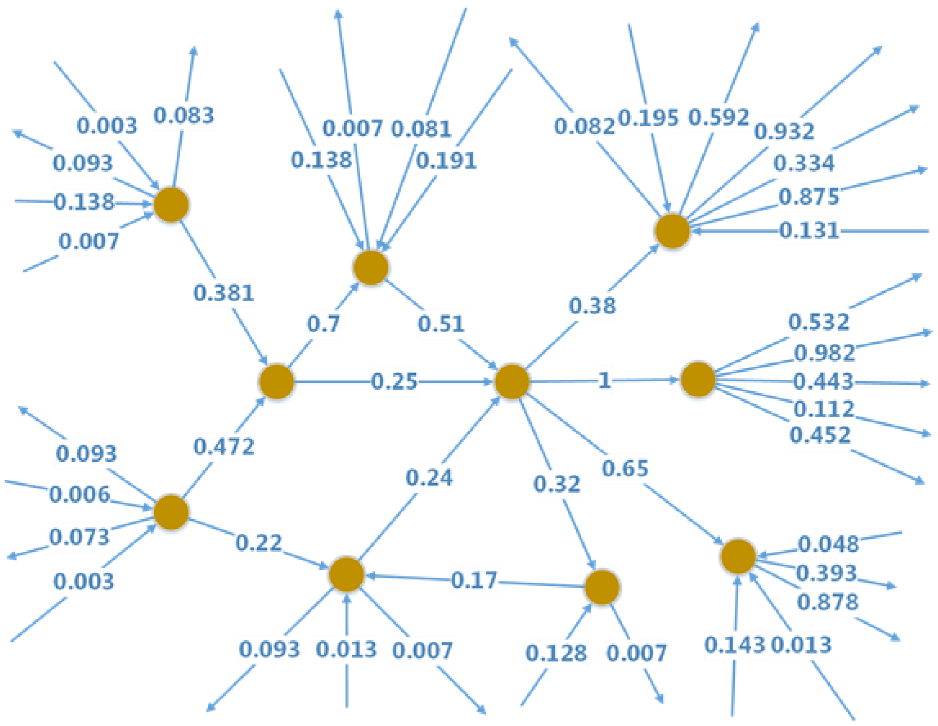
Collaboration

1. **Entity Resolution**
2. **Similarity in Bipartite Graphs**
3. **Knowledge Graph Support**
4. **Computing Higher-Level Events and Signals on KG**
5. **Fact Enrichment and Verificiation on KGs**

… more applications that we cannot talk about at this point

1 Car producer & 1 Supermarket Chain, Joint project with Univ. of Appl. Science Upper Austria in Steyr

Risk estimation in supply chains



```
sells(S,B,P)  &
P>0.5  →  depends(S,B).

sells(S,B′,P′)  &
depends(B′,B)   &
V=msum(P′,⟨B′>)  &
V>0.5  →  depends(S,B).
```

Thank You!

Thank You!

Knowledge Graphs in Action

Part 2: Theory to Practice

Emanuel Sallinger

TU WIEN | Knowledge Graph Lab

UNIVERSITY OF OXFORD

AI Summer School 2023

AIML Center for Artificial Intelligence and Machine Learning

asai austrian society for artificial intelligence

3 July 2023

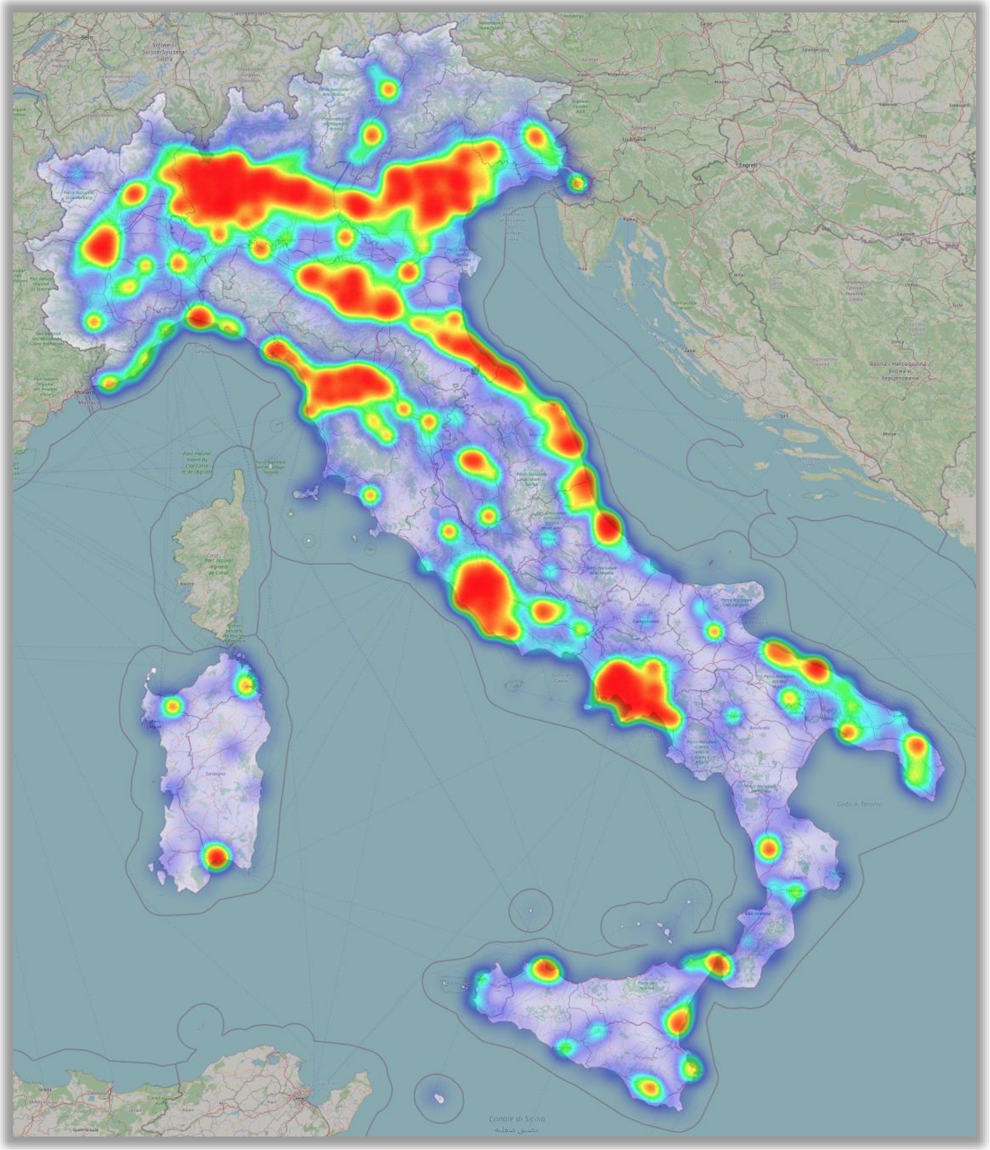
Economic Impact of Crises

BANCA D'ITALIA

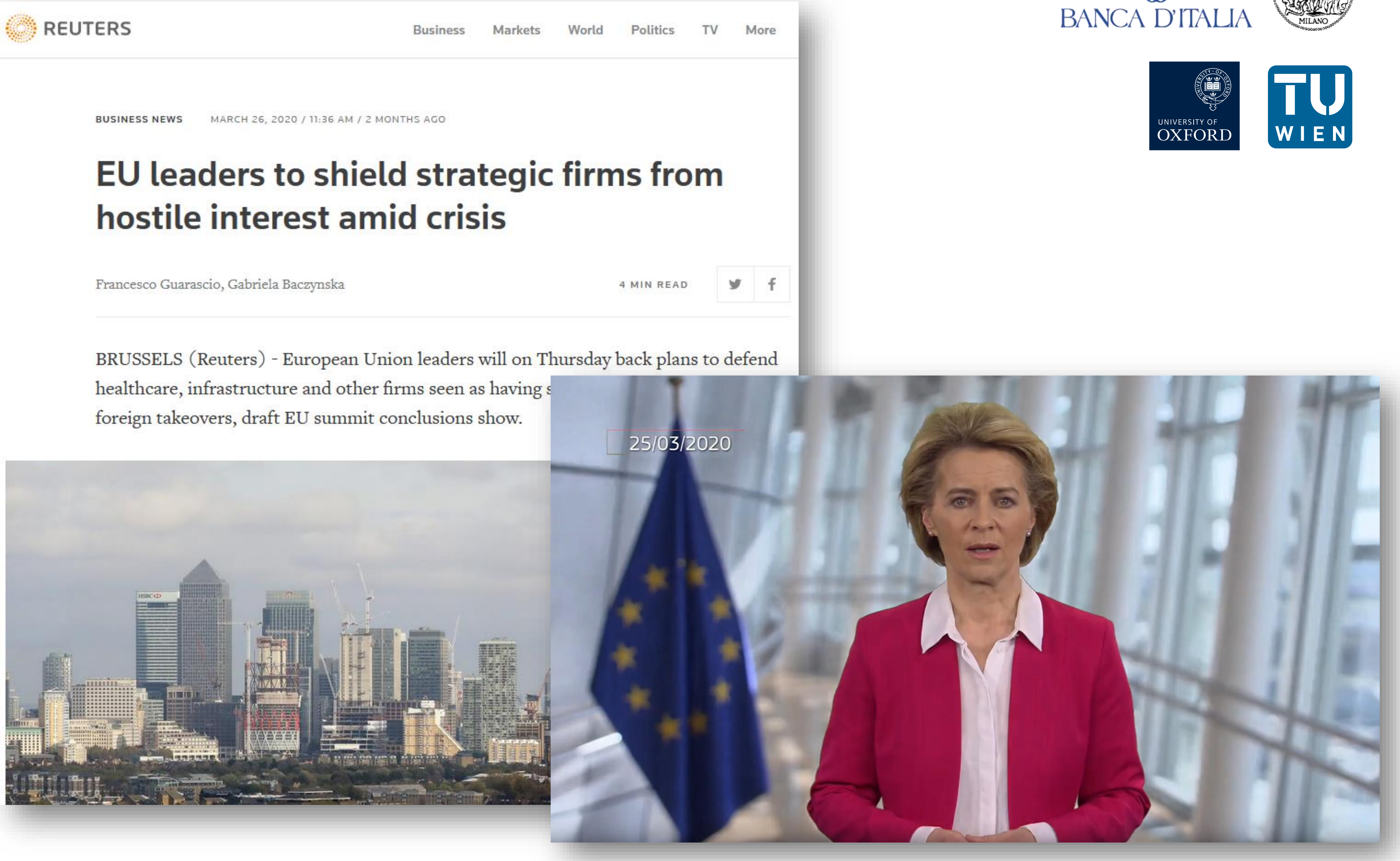
POLITECNICO MILANO

UNIVERSITY OF OXFORD

TU WIEN



Hostile Takeovers



BUSINESS NEWS MARCH 26, 2020 / 11:36 AM / 2 MONTHS AGO

EU leaders to shield strategic firms from hostile interest amid crisis

Francesco Guarascio, Gabriela Baczynska 4 MIN READ

BRUSSELS (Reuters) - European Union leaders will on Thursday back plans to defend healthcare, infrastructure and other firms seen as having s
foreign takeovers, draft EU summit conclusions show.

25/03/2020

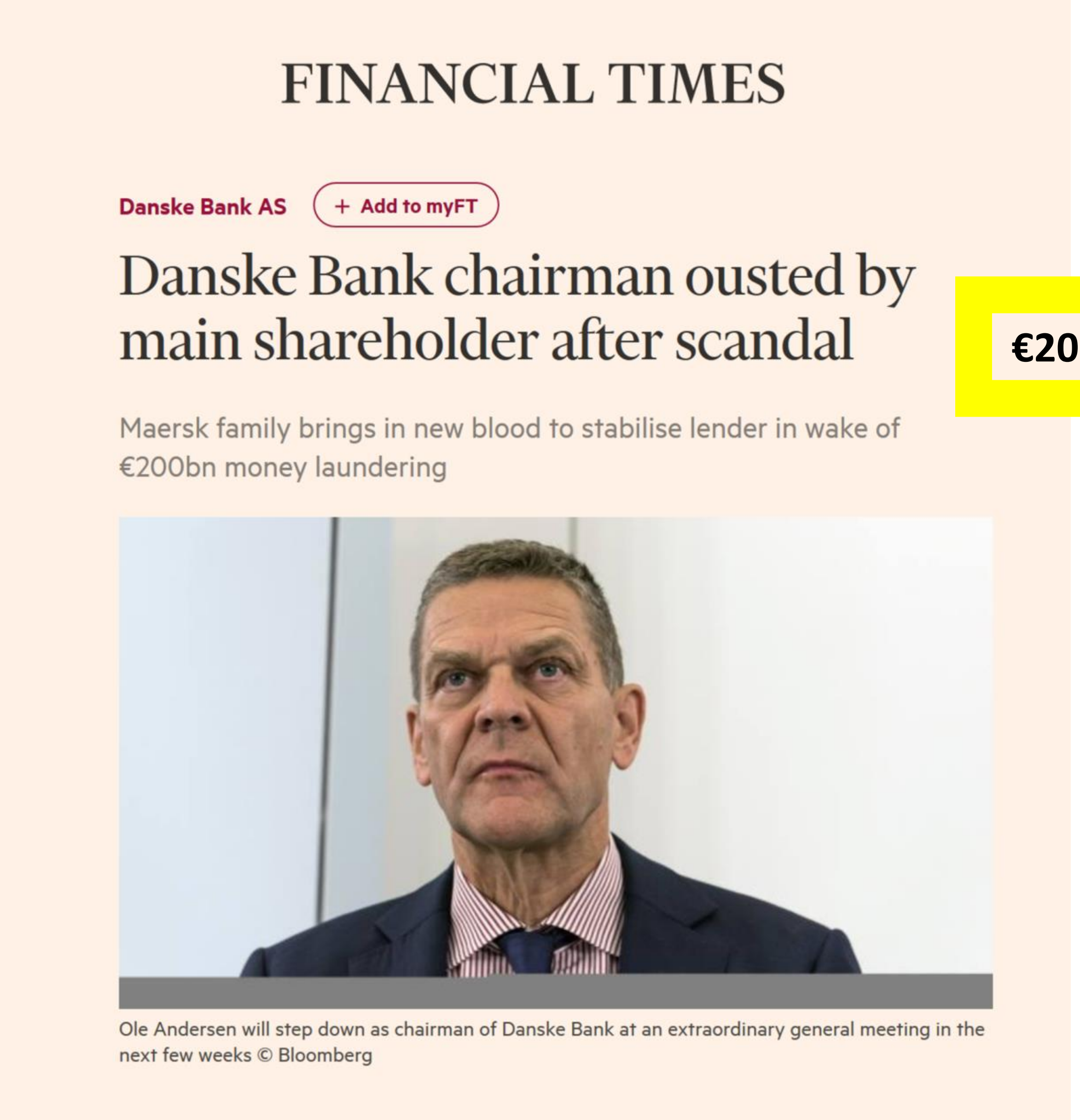
BANCA D'ITALIA

POLITECNICO MILANO

UNIVERSITY OF OXFORD

TU WIEN

Anti-Money Laundering



FINANCIAL TIMES

Danske Bank AS + Add to myFT

Danske Bank chairman ousted by main shareholder after scandal

Maersk family brings in new blood to stabilise lender in wake of €200bn money laundering

Ole Andersen will step down as chairman of Danske Bank at an extraordinary general meeting in the next few weeks © Bloomberg

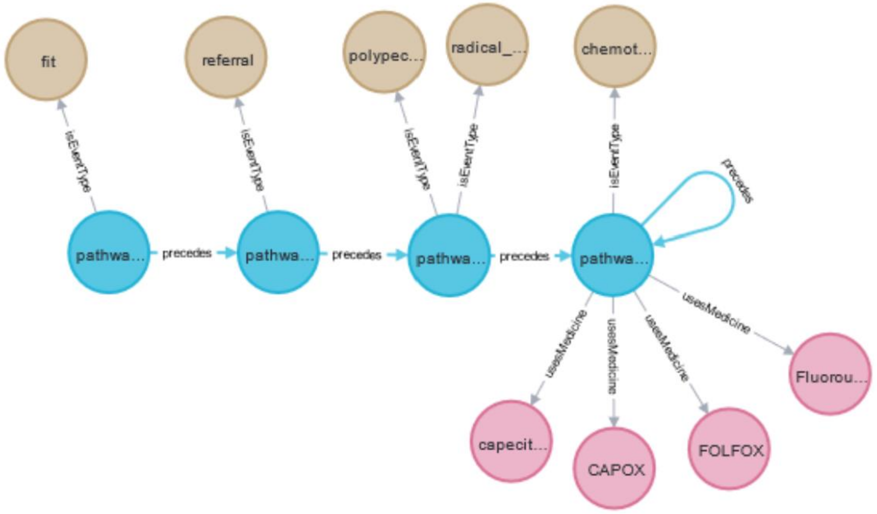
€200bn money laundering

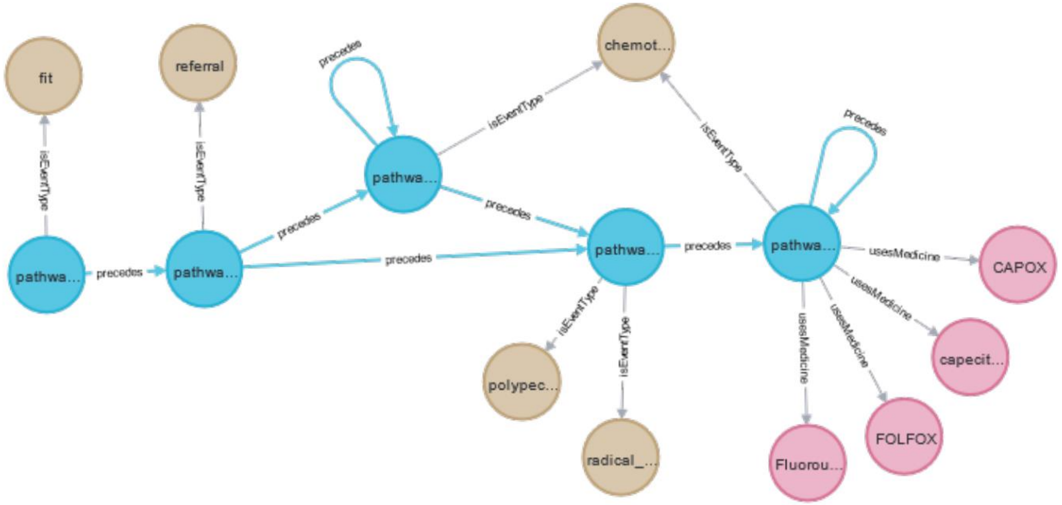
Cancer Pathways

UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF OXFORD BIG DATA INSTITUTE HEALTH DATA SCIENCE

TU WIEN





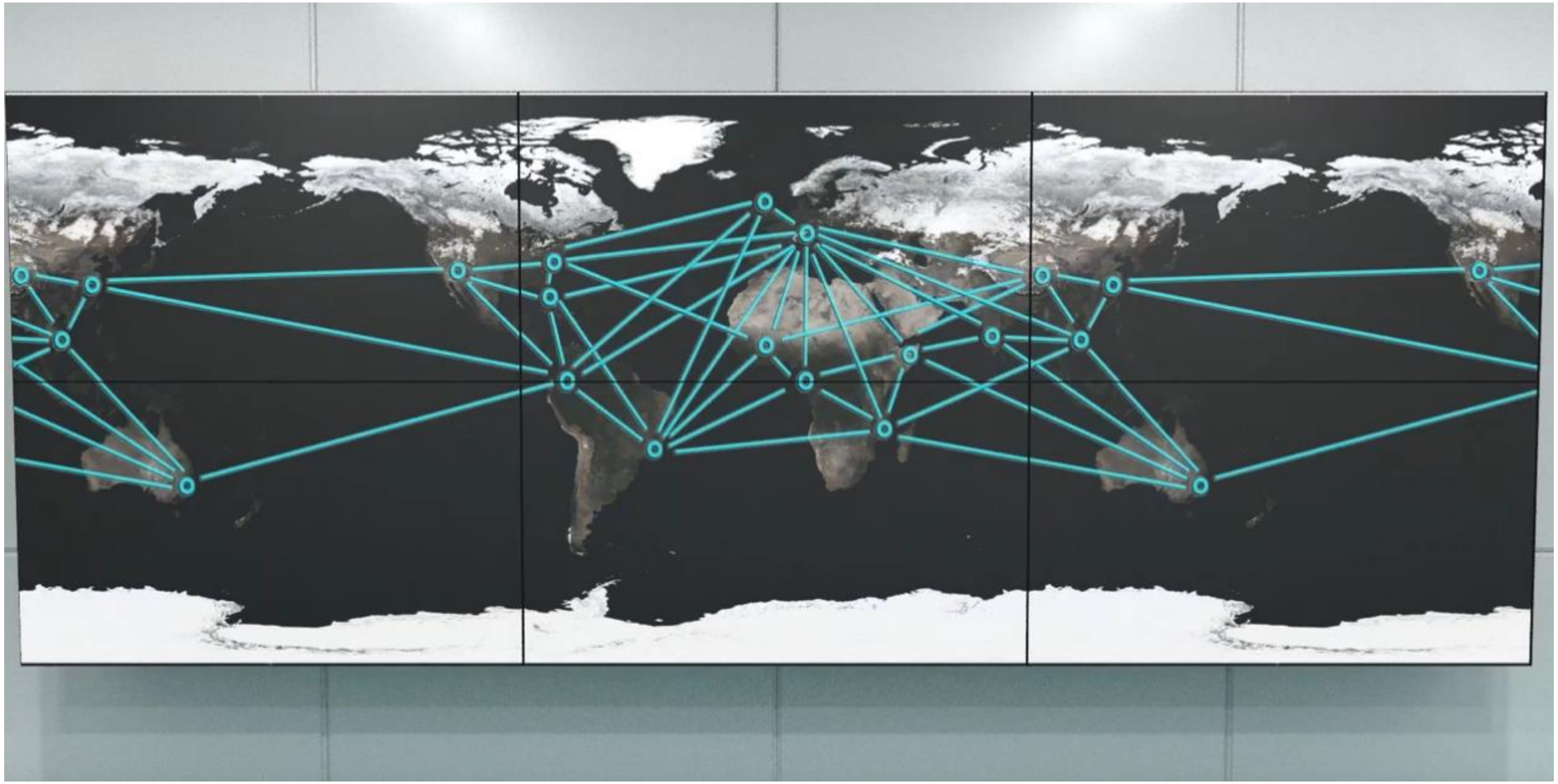
Supply Chains

UNIVERSITY OF OXFORD

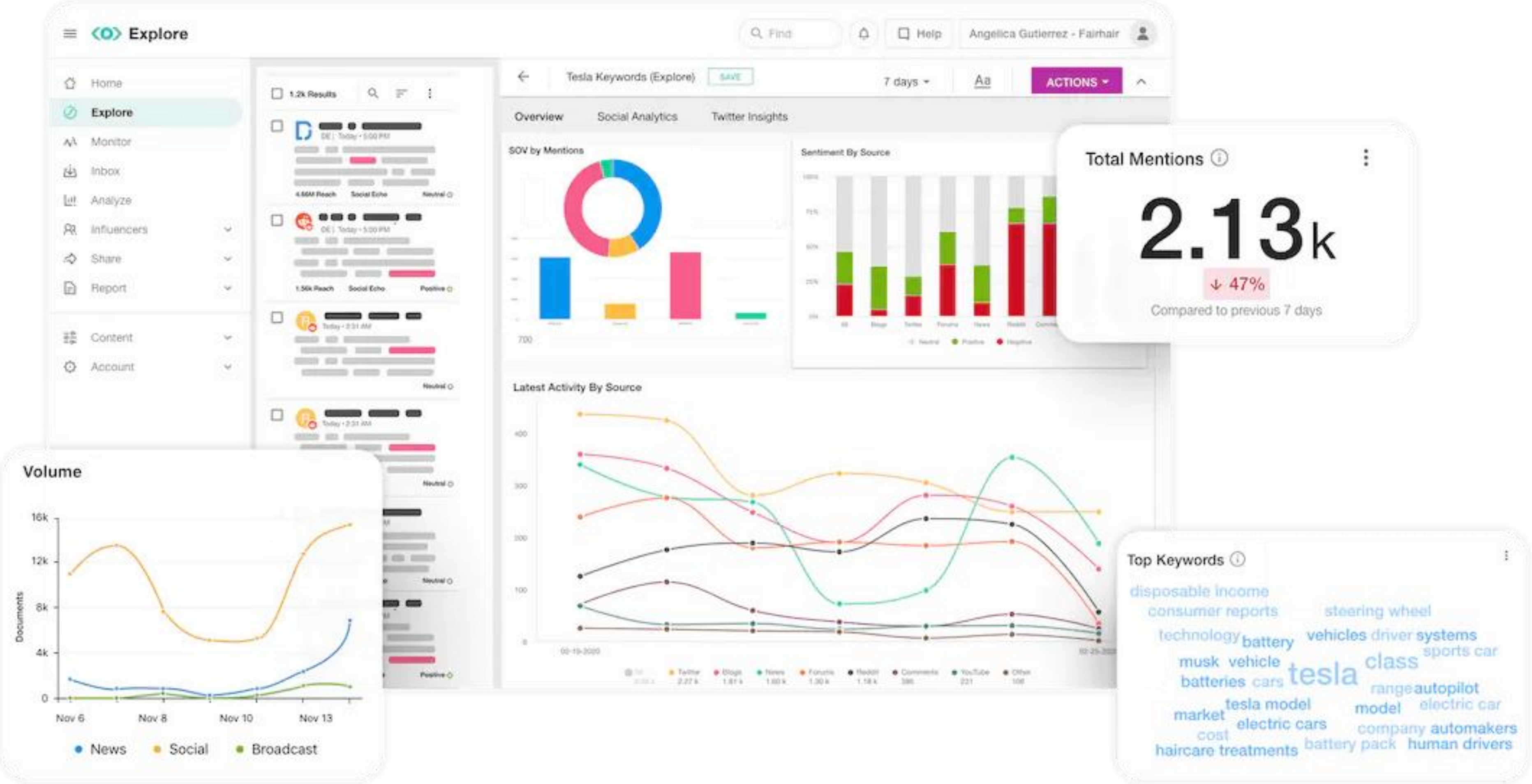
OBERÖSTERREICH

TU WIEN

JRC LIVE



Media Intelligence



Open Knowledge Network

Innovation Sprint

OPEN KNOWLEDGE NETWORK ROADMAP:

POWERING THE NEXT DATA REVOLUTION

SEPTEMBER 2022

Baby Formula Supply Chain Issue

Individual

Image source: https://www.newsweek.com/

Baby Formula
Supply Chain Issue

👤 Individual

Supply Chain
+ Finance

Image source: https://www.newsweek.com/

Baby Formula Supply Chain Issue

Individual

Supply Chain + Finance

Image source: https://www.newsweek.com/

PPP / US CARES Effectiveness

Small Business

COVID Impact + Finance

Image source: https://shutterstock.com/

Baby Formula Supply Chain Issue

Individual

Supply Chain + Finance

Image source: https://www.newsweek.com/

PPP / US CARES Effectiveness

Small Business

COVID Impact + Finance

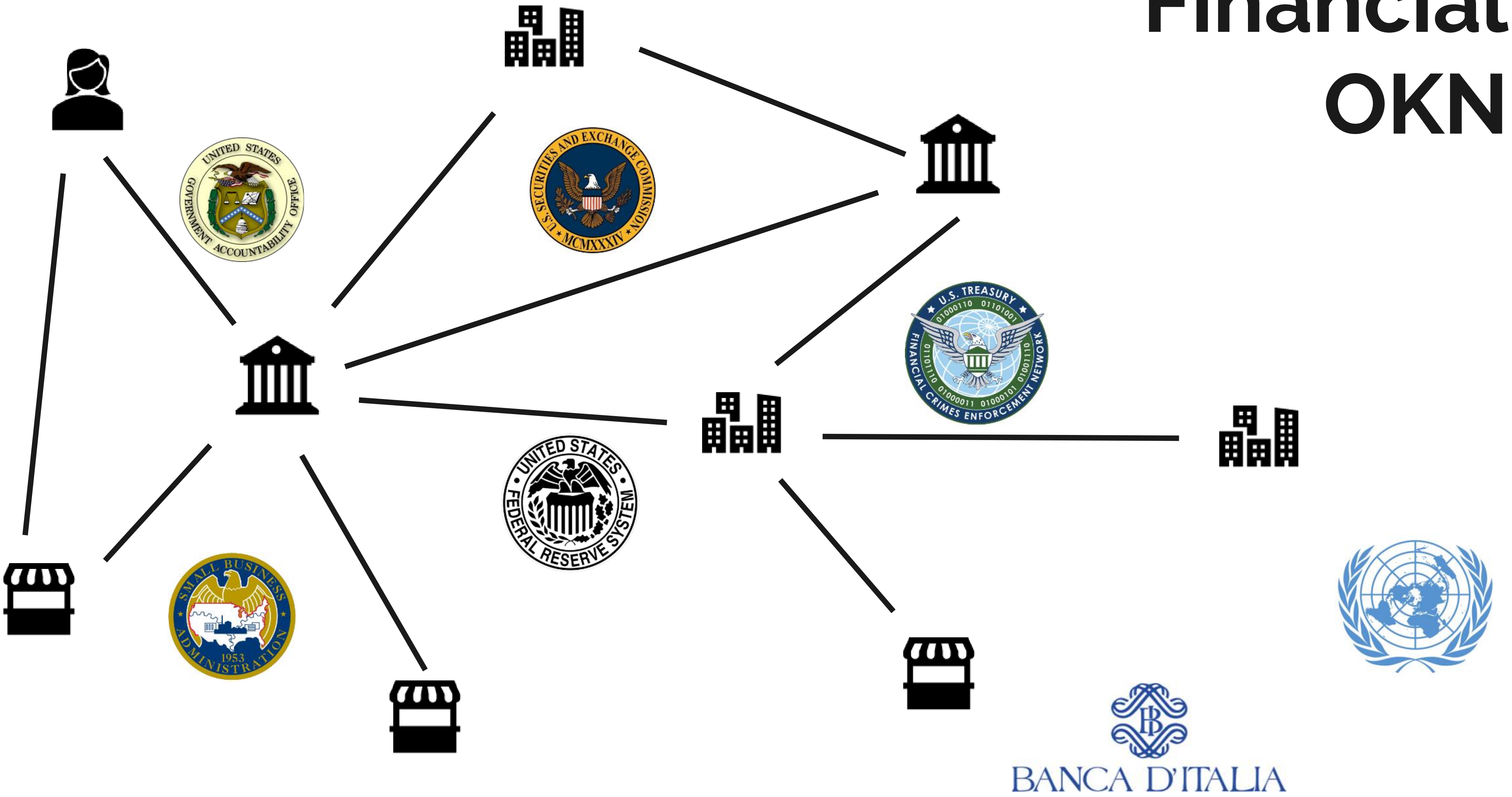
Image source: https://shutterstock.com/

Climate Finance

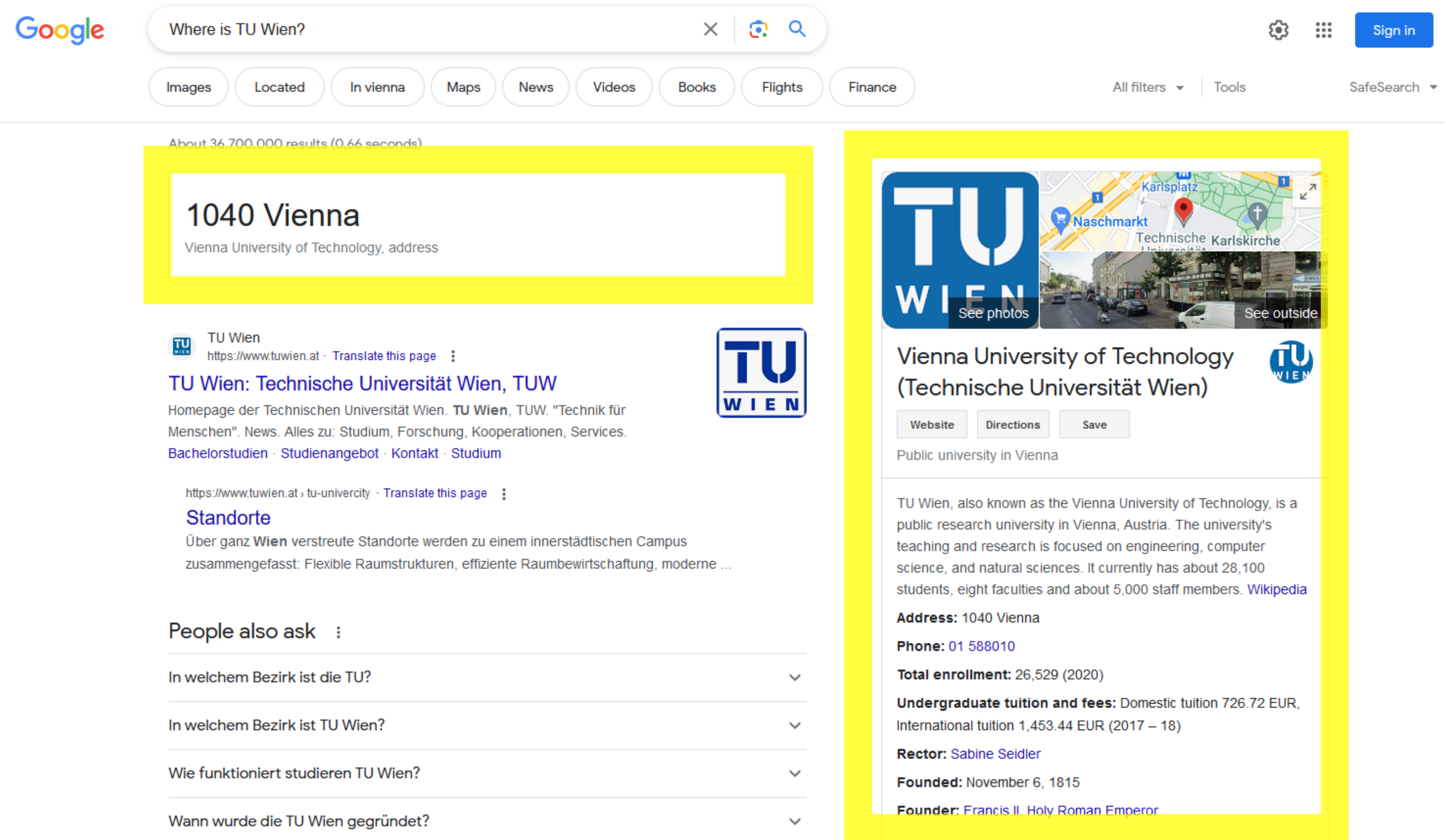
Environmental Models + Finance

Image source: https://www.lse.ac.uk/

Financial OKN


UNITED STATES GOVERNMENT ACCOUNTABILITY OFFICE
U.S. SECURITIES AND EXCHANGE COMMISSION MCMXXXIV
U.S. TREASURY FINANCIAL CRIMES ENFORCEMENT NETWORK
UNITED STATES FEDERAL RESERVE SYSTEM
SMALL BUSINESS ADMINISTRATION 1953
BANCA D'ITALIA

“Where is the TU Wien?”



About 36,700,000 results (0.66 seconds)

1040 Vienna

Vienna University of Technology, address

TU Wien
https://www.tuwien.at · Translate this page

TU Wien: Technische Universität Wien, TUW

Homepage der Technischen Universität Wien. **TU Wien**, TUW. "Technik für Menschen". News. Alles zu: Studium, Forschung, Kooperationen, Services.

Bachelorstudien · Studienangebot · Kontakt · Studium

https://www.tuwien.at › tu-univercity · Translate this page

Standorte

Über ganz **Wien** verstreute Standorte werden zu einem innerstädtischen Campus zusammengefasst: Flexible Raumstrukturen, effiziente Raumbewirtschaftung, moderne ...

People also ask

In welchem Bezirk ist die TU?

In welchem Bezirk ist TU Wien?

Wie funktioniert studieren TU Wien?

Wann wurde die TU Wien gegründet?

Vienna University of Technology (Technische Universität Wien)

Public university in Vienna

TU Wien, also known as the Vienna University of Technology, is a public research university in Vienna, Austria. The university's teaching and research is focused on engineering, computer science, and natural sciences. It currently has about 28,100 students, eight faculties and about 5,000 staff members. Wikipedia

Address: 1040 Vienna

Phone: 01 588010

Total enrollment: 26,529 (2020)

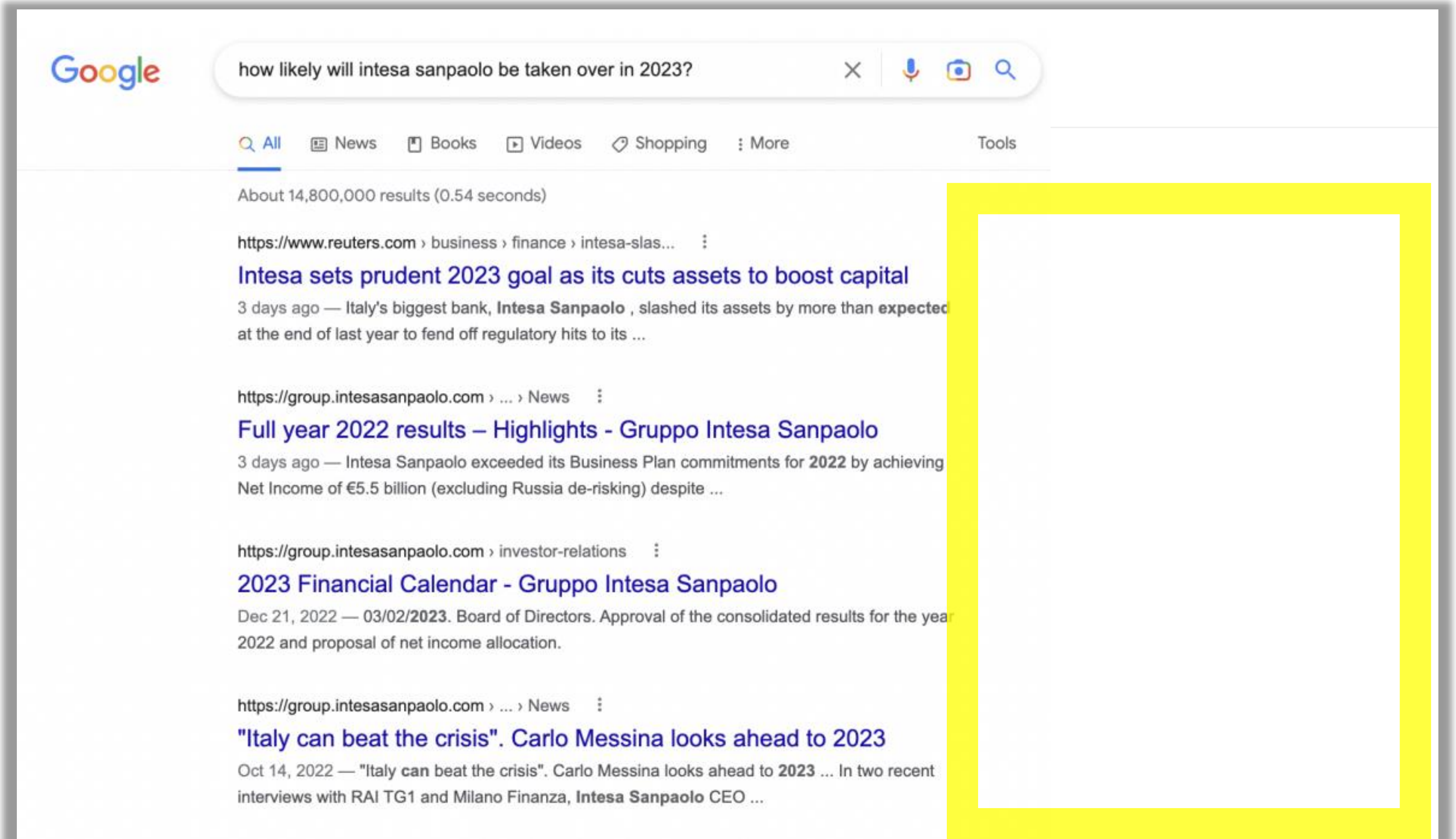
Undergraduate tuition and fees: Domestic tuition 726.72 EUR, International tuition 1,453.44 EUR (2017 – 18)

Rector: Sabine Seidler

Founded: November 6, 1815

Founder: Francis II, Holy Roman Emperor

“How likely will the largest bank in Italy be taken over in 2023?"



how likely will intesa sanpaolo be taken over in 2023?

About 14,800,000 results (0.54 seconds)

https://www.reuters.com › business › finance › intesa-slas...

Intesa sets prudent 2023 goal as its cuts assets to boost capital

3 days ago — Italy's biggest bank, **Intesa Sanpaolo** , slashed its assets by more than **expected** at the end of last year to fend off regulatory hits to its ...

https://group.intesasanpaolo.com › ... › News

Full year 2022 results – Highlights - Gruppo Intesa Sanpaolo

3 days ago — Intesa Sanpaolo exceeded its Business Plan commitments for **2022** by achieving Net Income of €5.5 billion (excluding Russia de-risking) despite ...

https://group.intesasanpaolo.com › investor-relations

2023 Financial Calendar - Gruppo Intesa Sanpaolo

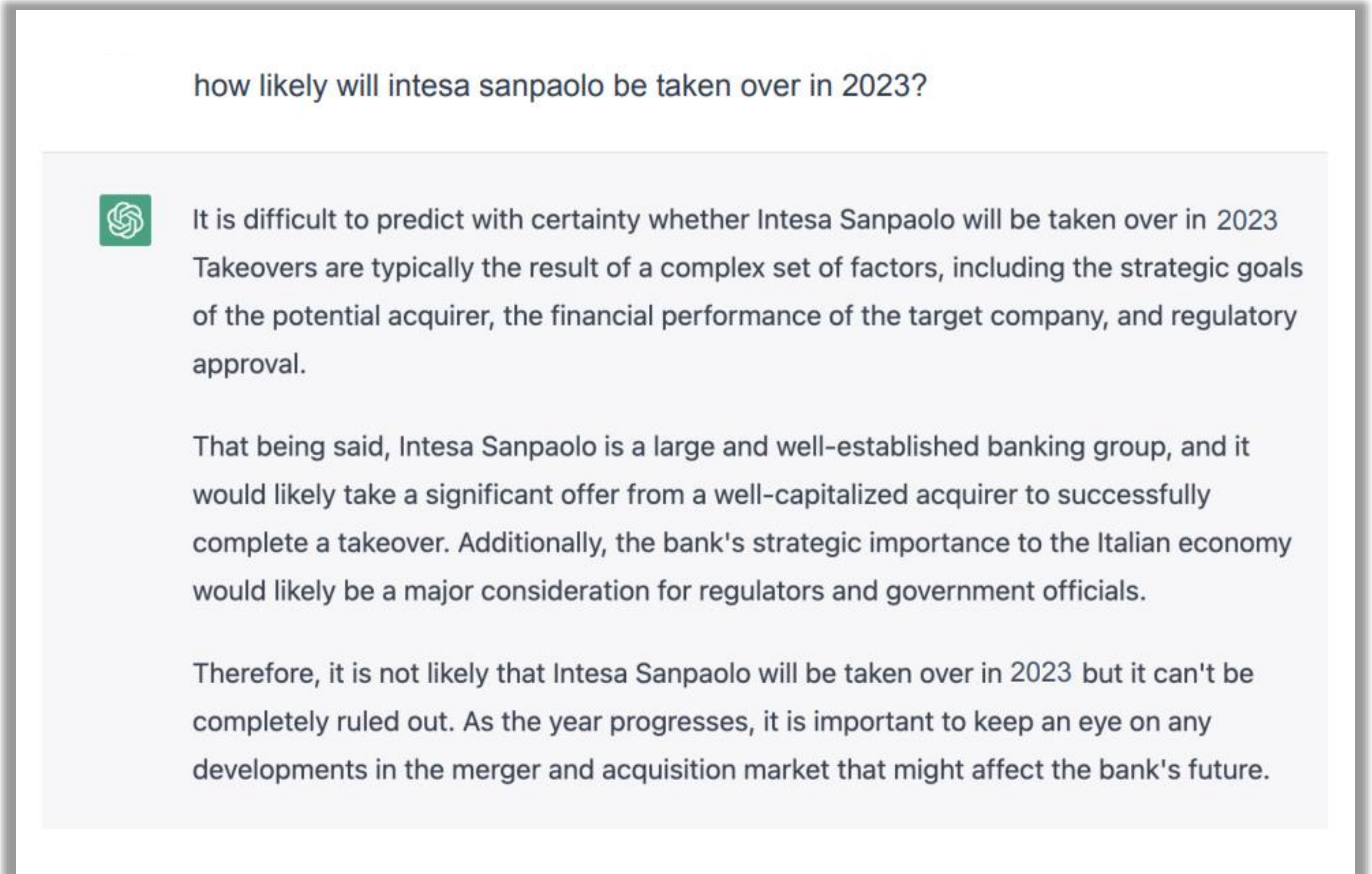
Dec 21, 2022 — 03/02/**2023**. Board of Directors. Approval of the consolidated results for the yea 2022 and proposal of net income allocation.

https://group.intesasanpaolo.com › ... › News

"Italy can beat the crisis". Carlo Messina looks ahead to 2023

Oct 14, 2022 — "Italy **can** beat the crisis". Carlo Messina looks ahead to **2023** ... In two recent interviews with RAI TG1 and Milano Finanza, **Intesa Sanpaolo** CEO ...

“How likely will the largest bank in Italy be taken over in 2023?"



how likely will intesa sanpaolo be taken over in 2023?

It is difficult to predict with certainty whether Intesa Sanpaolo will be taken over in 2023 Takeovers are typically the result of a complex set of factors, including the strategic goals of the potential acquirer, the financial performance of the target company, and regulatory approval.

That being said, Intesa Sanpaolo is a large and well-established banking group, and it would likely take a significant offer from a well-capitalized acquirer to successfully complete a takeover. Additionally, the bank's strategic importance to the Italian economy would likely be a major consideration for regulators and government officials.

Therefore, it is not likely that Intesa Sanpaolo will be taken over in 2023 but it can't be completely ruled out. As the year progresses, it is important to keep an eye on any developments in the merger and acquisition market that might affect the bank's future.

? ? ? ? ?

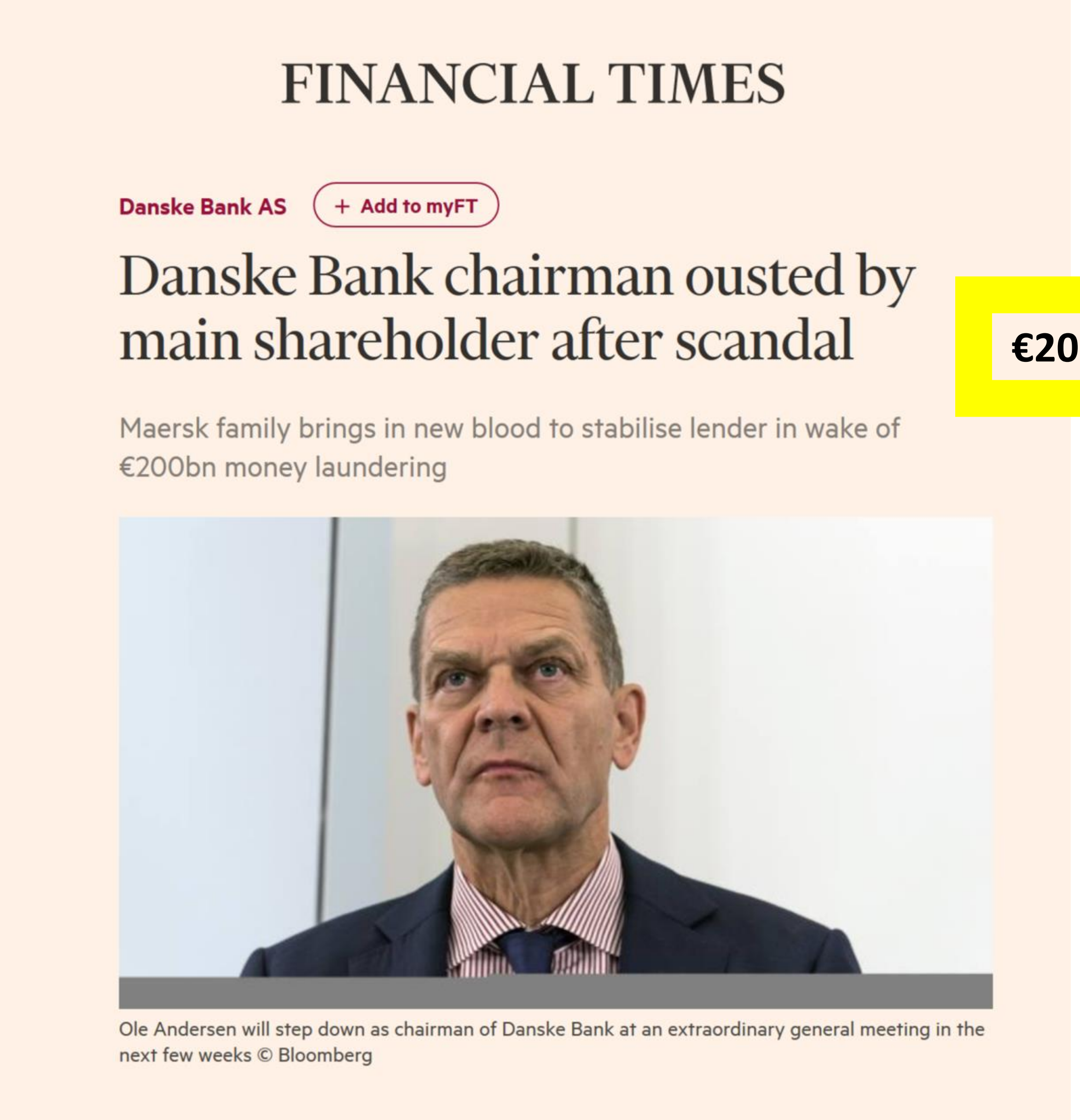
Knowledge Graphs

1. The **technology** used by Google and others

2. A meeting place of **databases**, **data science** and **Artificial Intelligence** research

3. A **skillset** to solve fascinating problems

Anti-Money Laundering



FINANCIAL TIMES

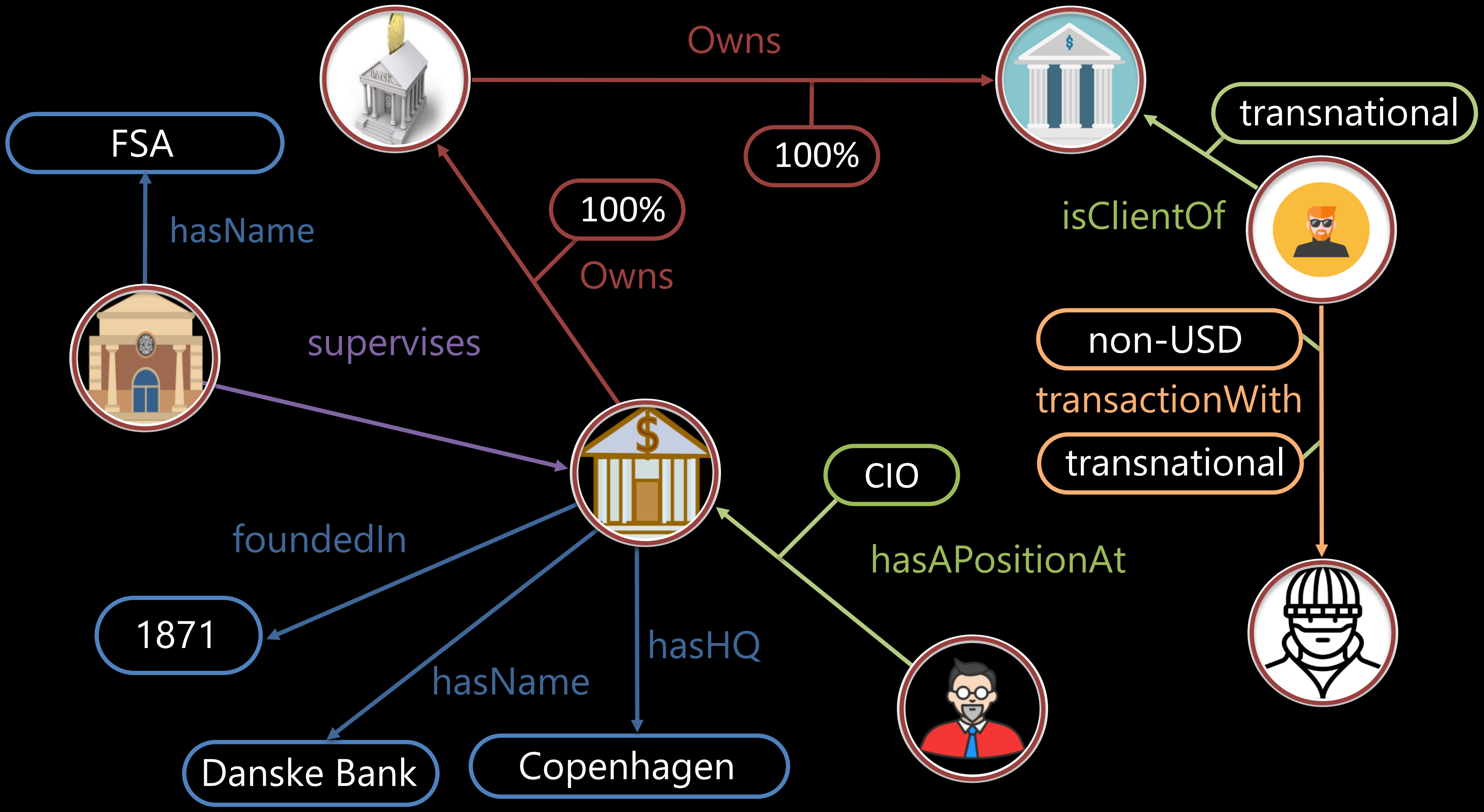
Danske Bank AS + Add to myFT

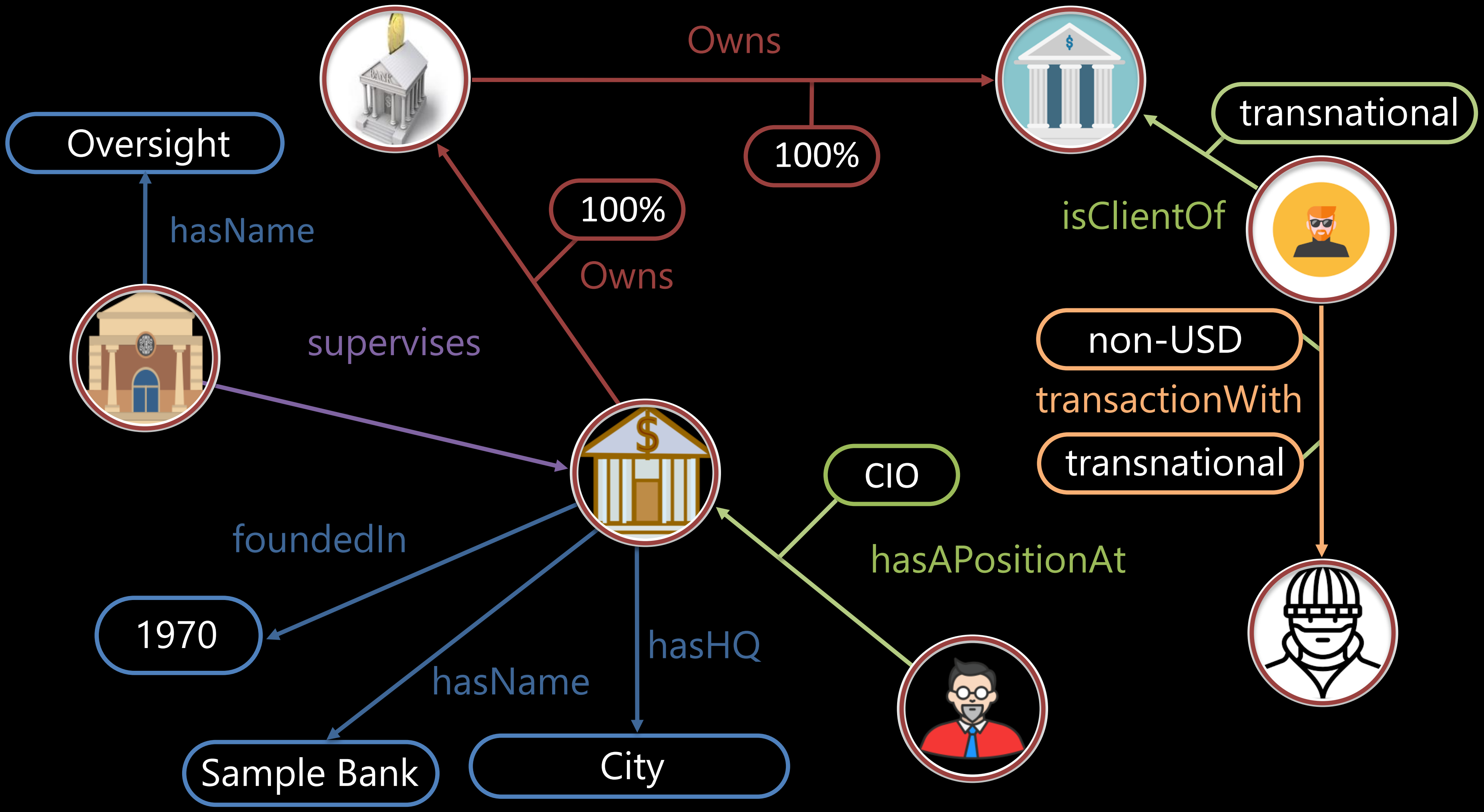
Danske Bank chairman ousted by main shareholder after scandal

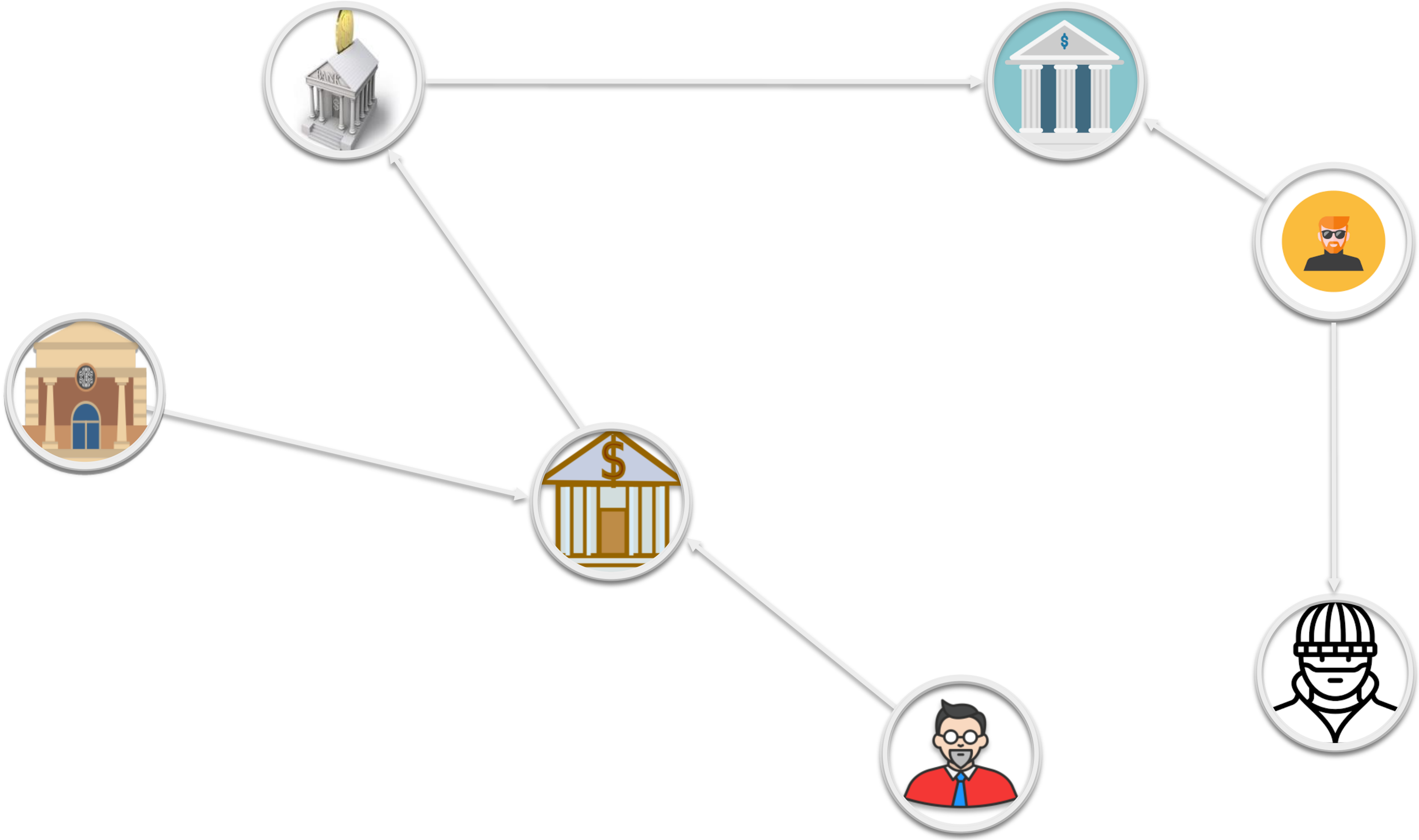
Maersk family brings in new blood to stabilise lender in wake of €200bn money laundering

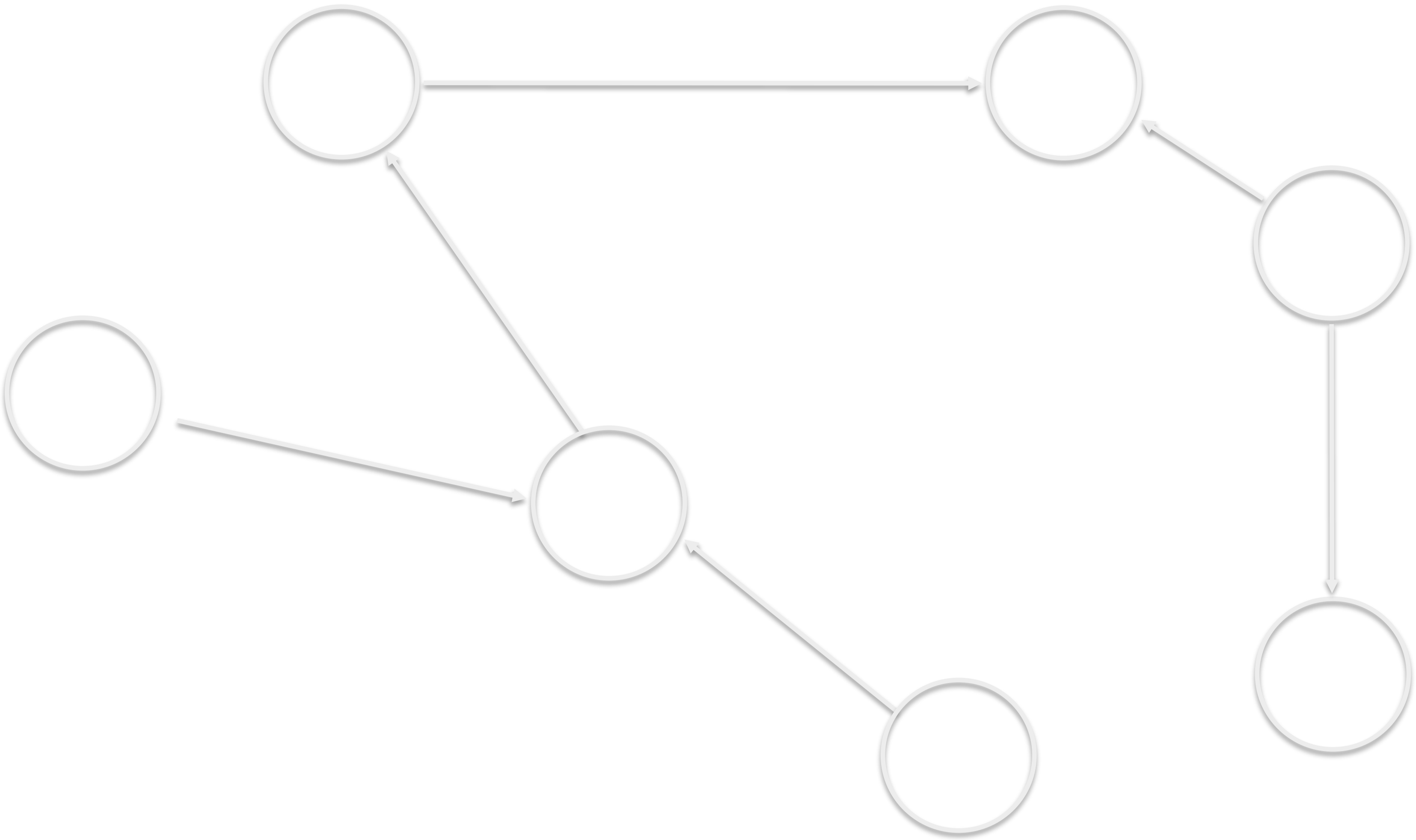
Ole Andersen will step down as chairman of Danske Bank at an extraordinary general meeting in the next few weeks © Bloomberg

€200bn money laundering


Owns
100%
transnational
FSA
hasName
100%
Owns
isClientOf
supervises
non-USD
transactionWith
transnational
CIO
hasAPositionAt
foundedIn
1871
hasHQ
hasName
Danske Bank
Copenhagen

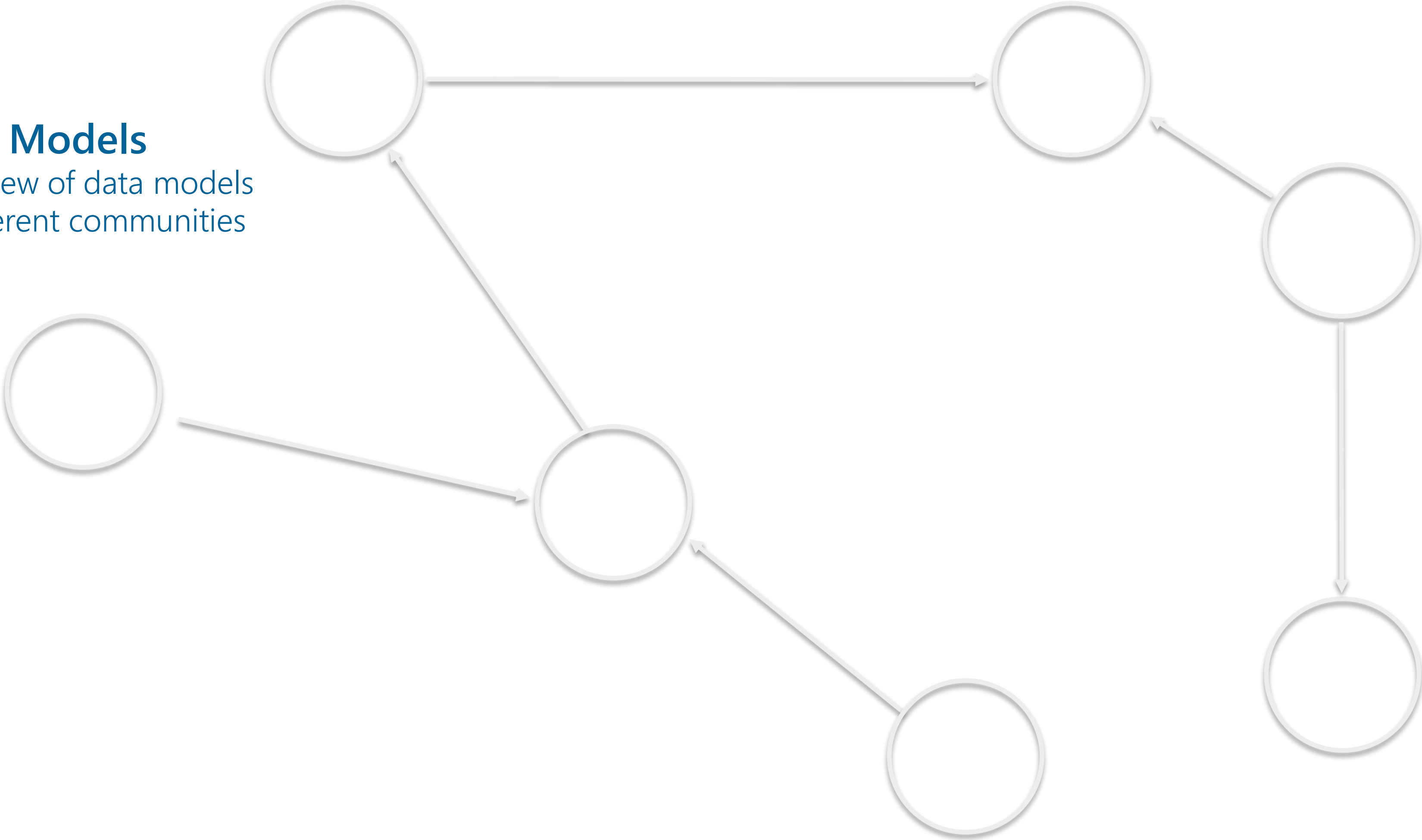

Owns
100%
transnational
Oversight
hasName
100%
Owns
isClientOf
supervises
non-USD
transactionWith
transnational
CIO
hasAPositionAt
foundedIn
1970
hasHQ
hasName
Sample Bank
City

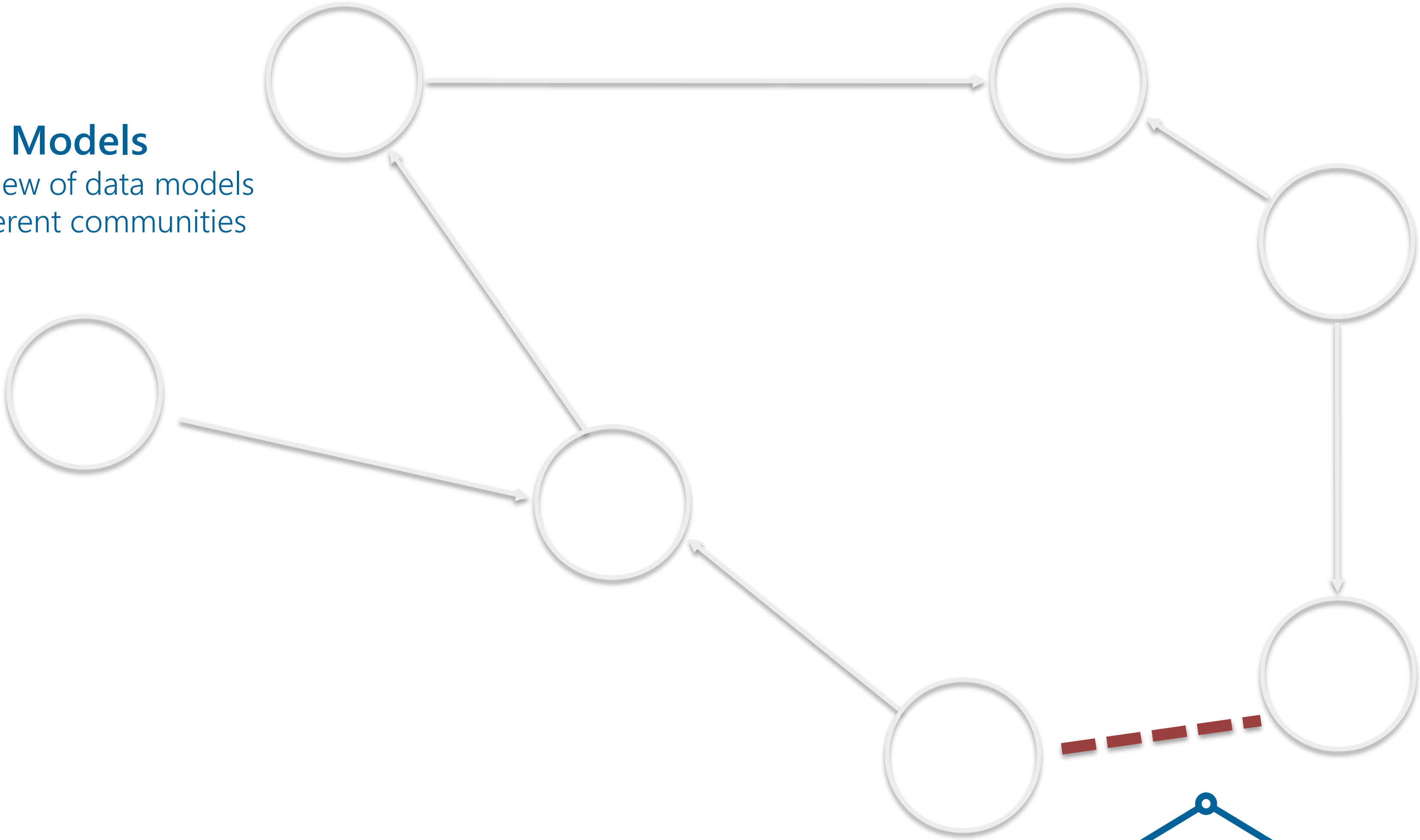




Data Models

Overview of data models in different communities



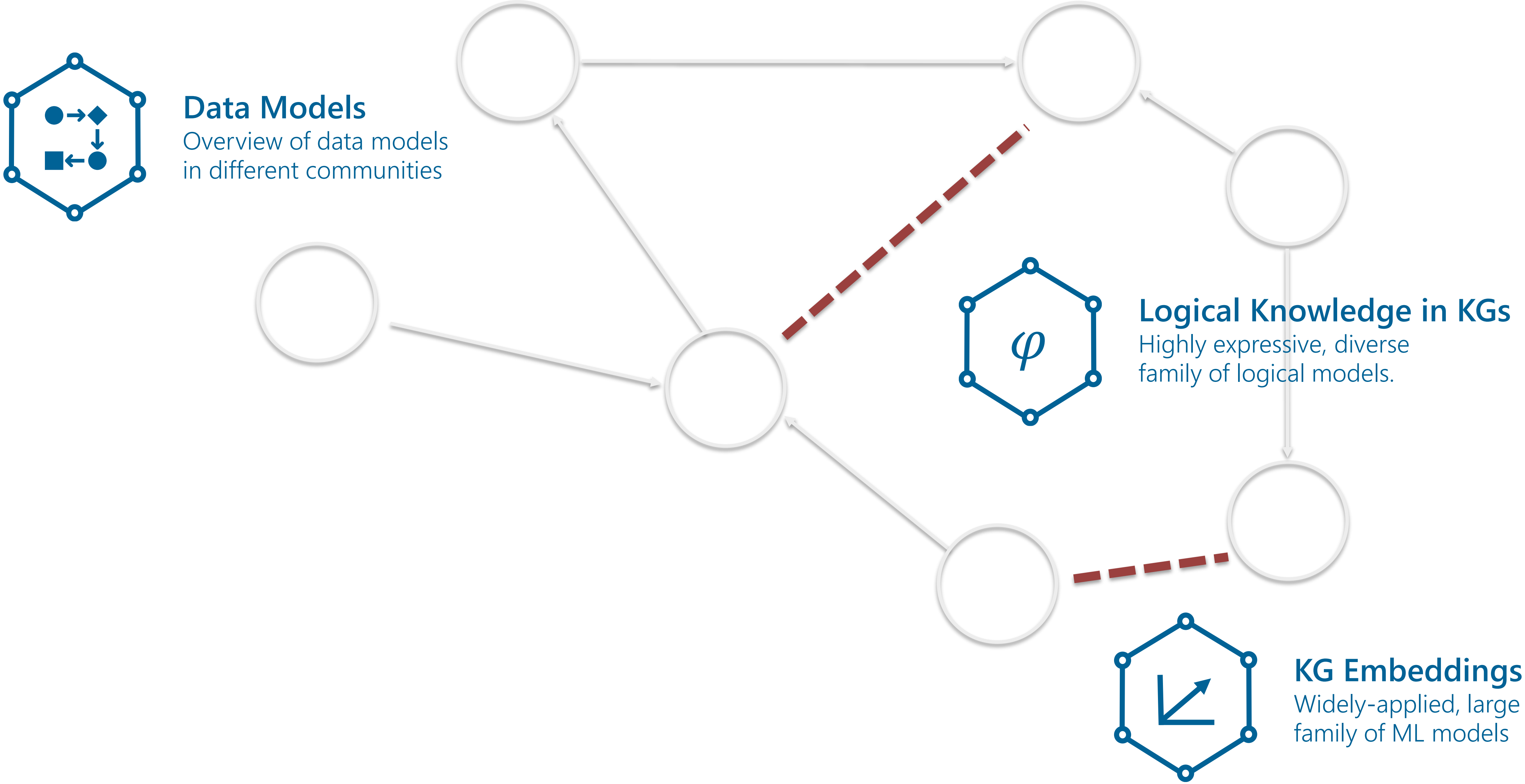


Data Models

Overview of data models
in different communities

KG Embeddings

Widely-applied, large
family of ML models



Data Models

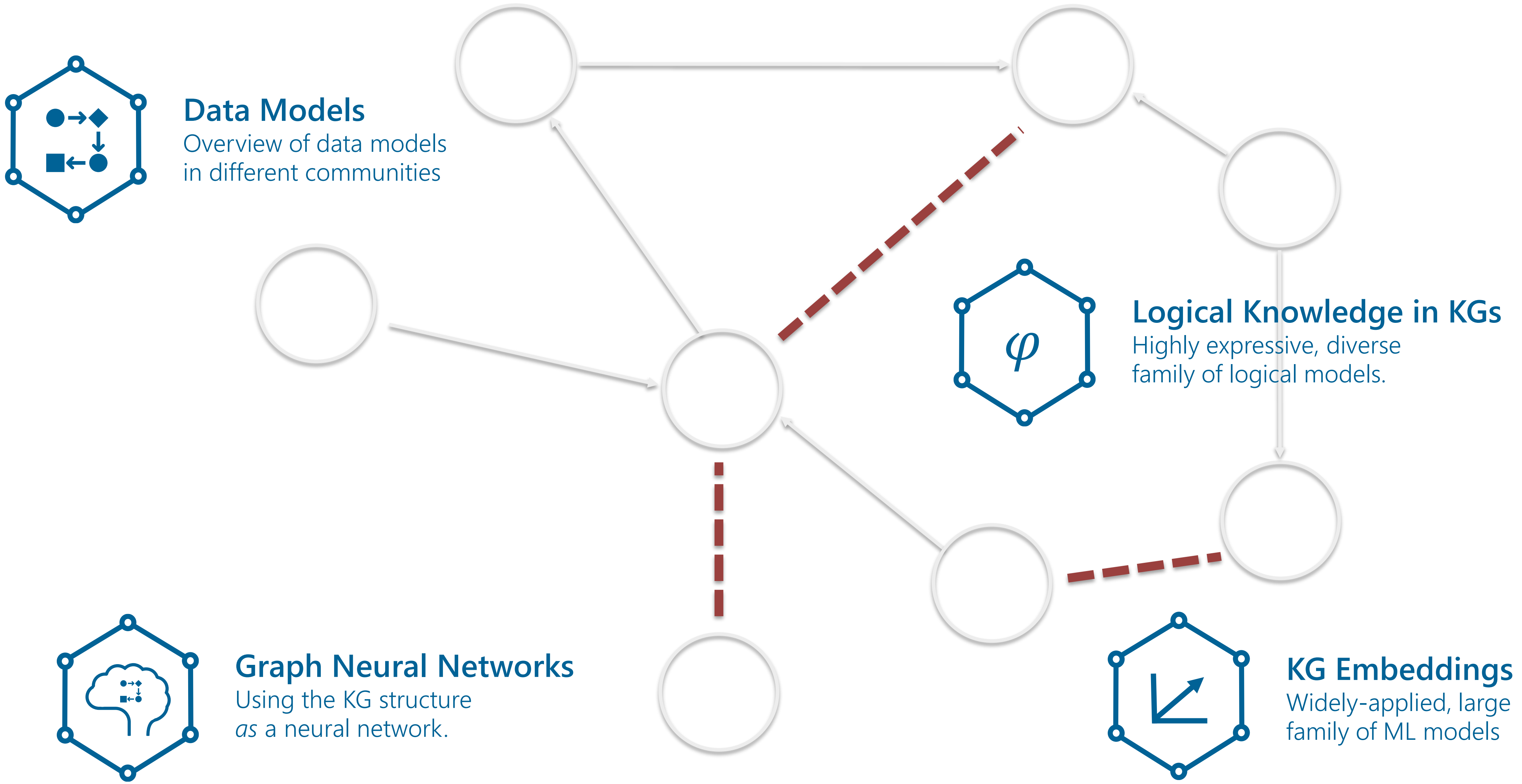
Overview of data models in different communities

Logical Knowledge in KGs

Highly expressive, diverse family of logical models.

KG Embeddings

Widely-applied, large family of ML models



Data Models

Overview of data models in different communities

Logical Knowledge in KGs

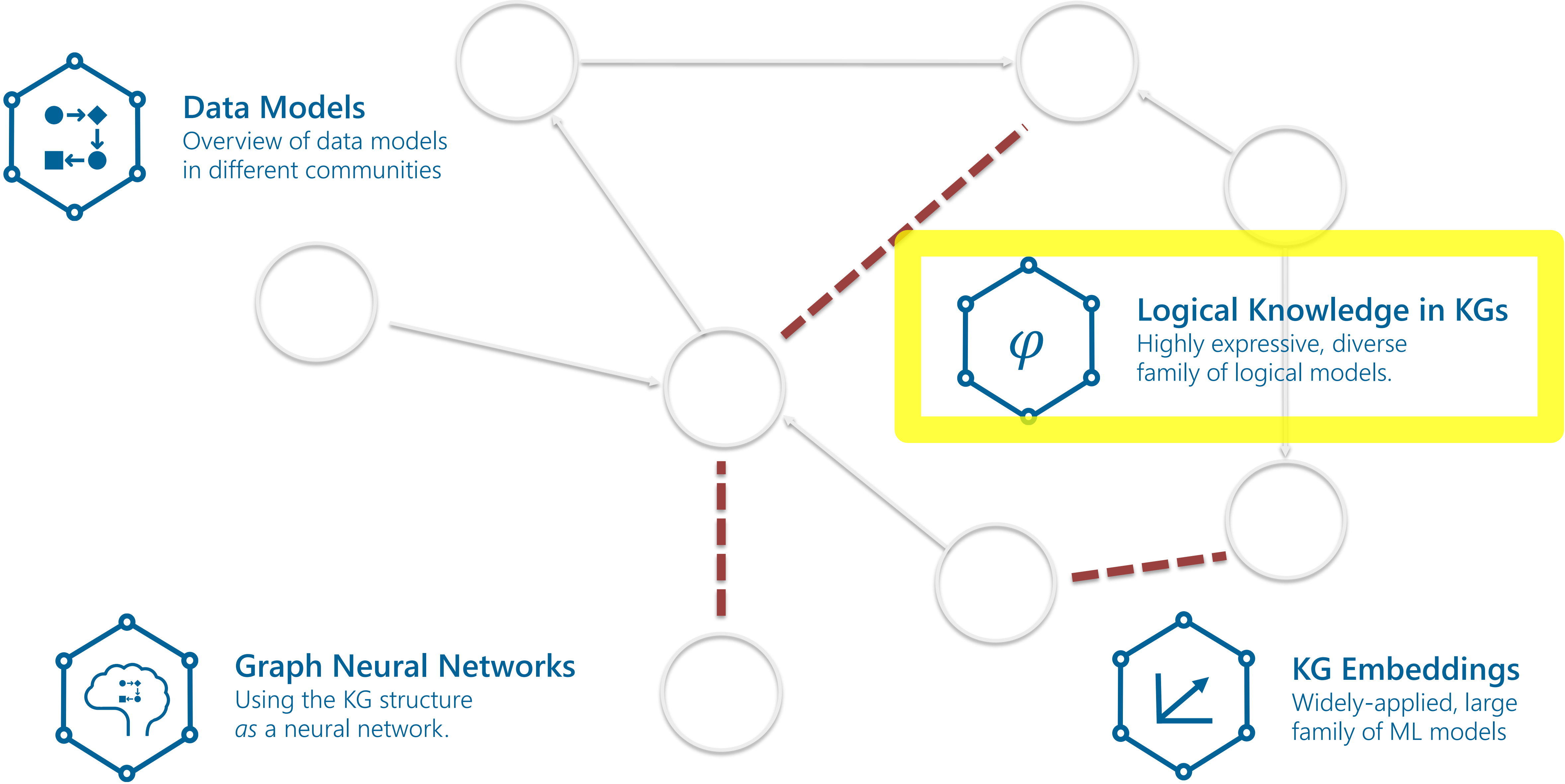
Highly expressive, diverse family of logical models.

Graph Neural Networks

Using the KG structure *as* a neural network.

KG Embeddings

Widely-applied, large family of ML models

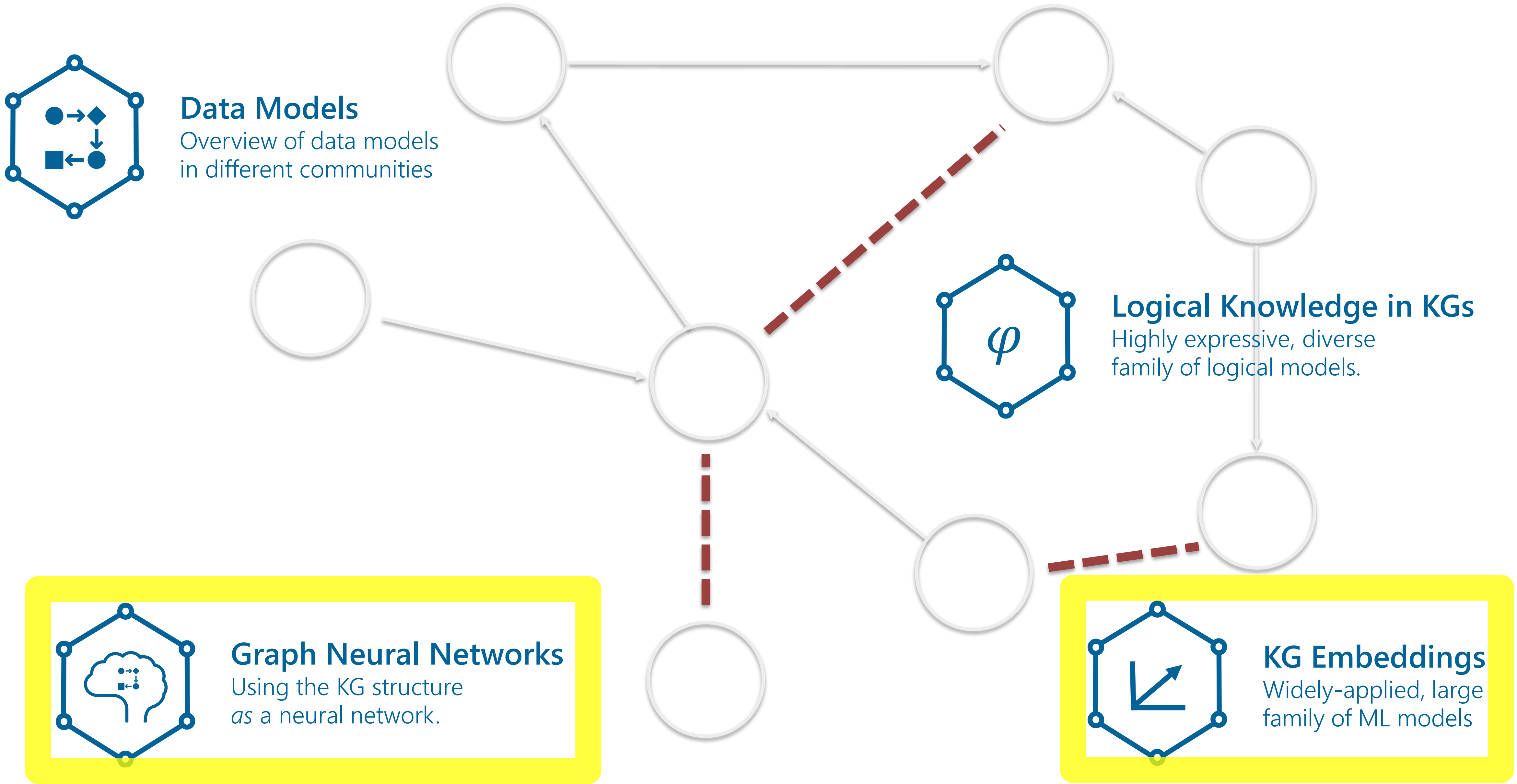


Data Models
Overview of data models
in different communities

Logical Knowledge in KGs
Highly expressive, diverse
family of logical models.

Graph Neural Networks
Using the KG structure
as a neural network.

KG Embeddings
Widely-applied, large
family of ML models

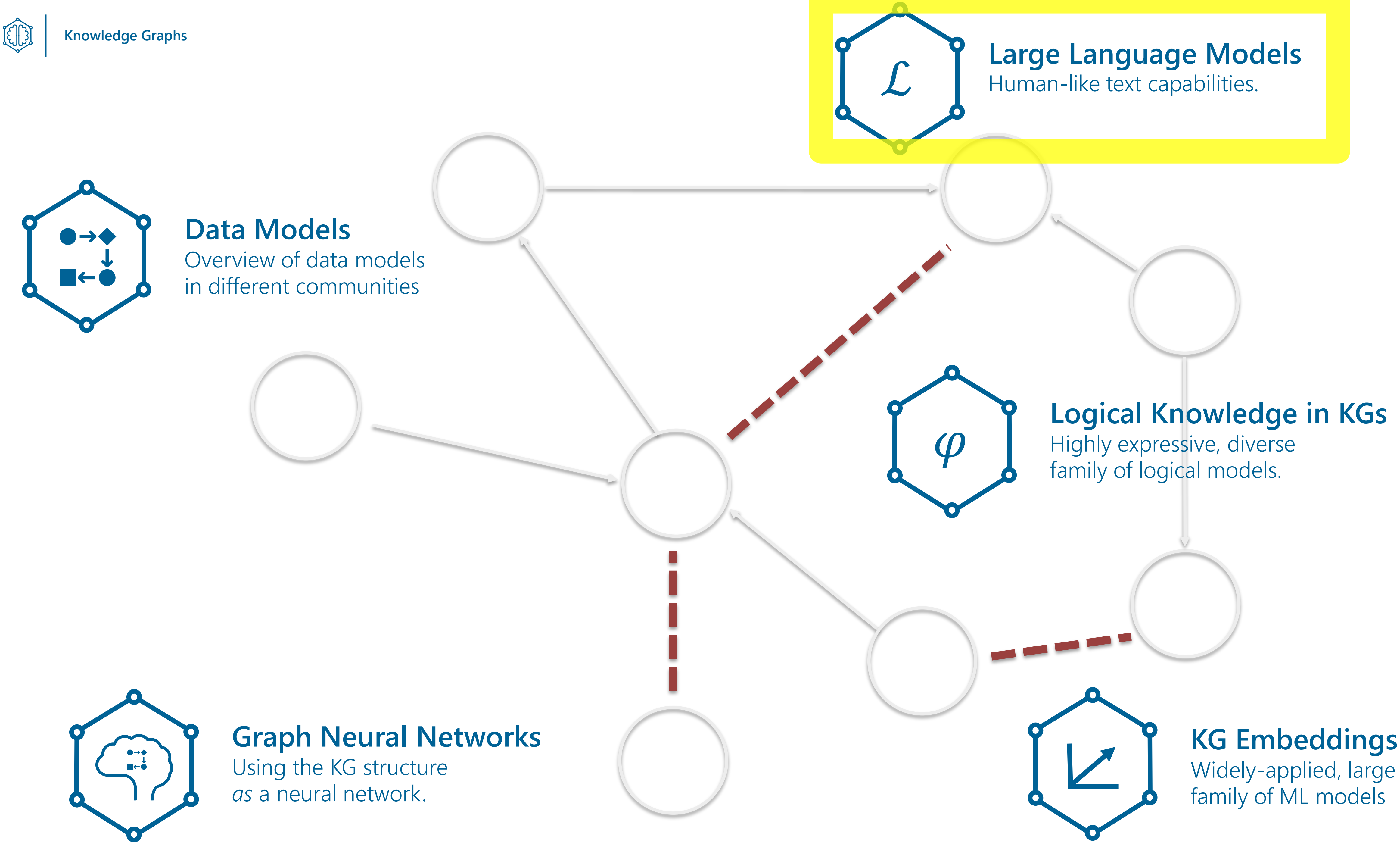


Data Models
Overview of data models
in different communities

Logical Knowledge in KGs
Highly expressive, diverse
family of logical models.

Graph Neural Networks
Using the KG structure
as a neural network.

KG Embeddings
Widely-applied, large
family of ML models



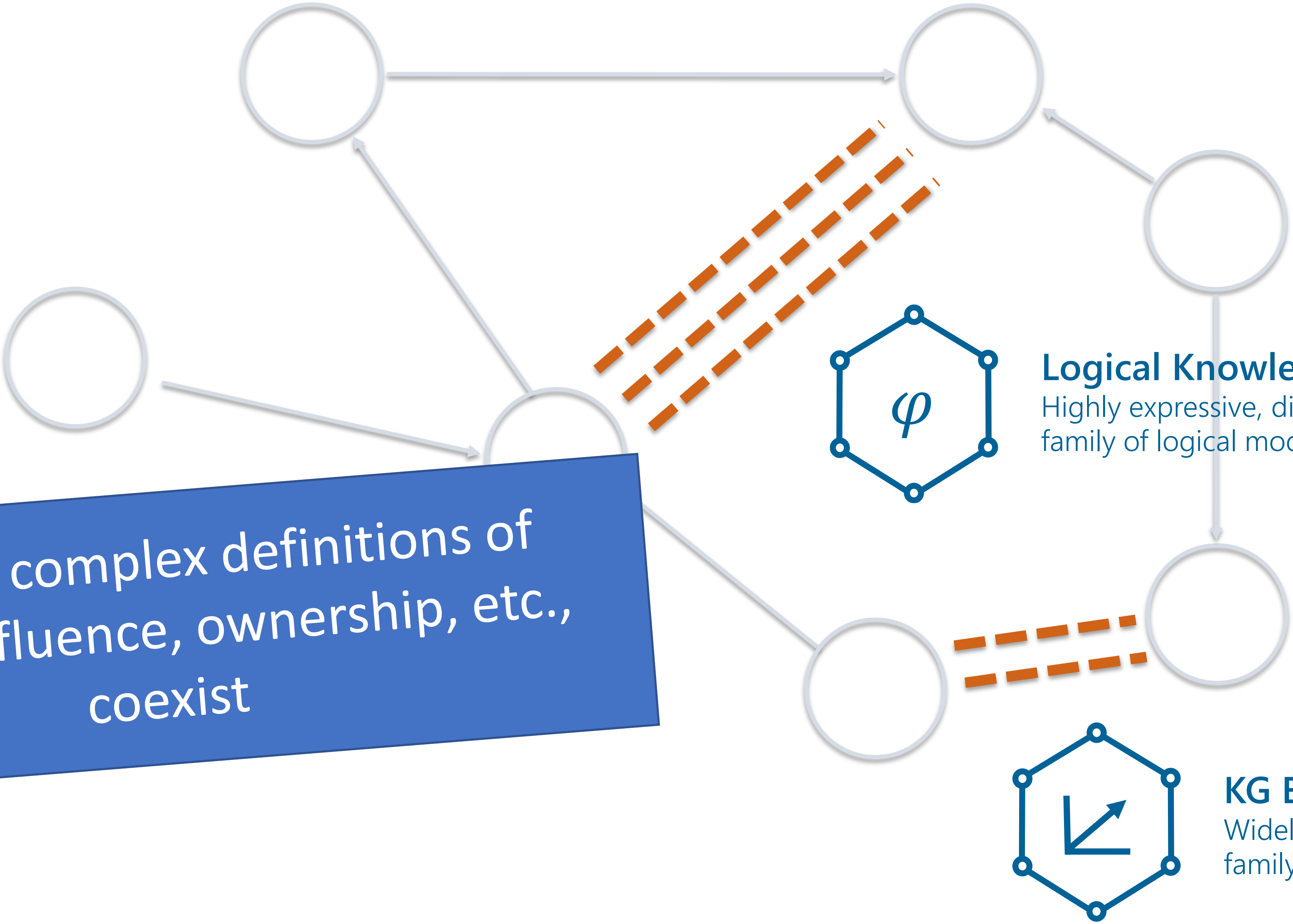
Large Language Models
Human-like text capabilities.

Data Models
Overview of data models
in different communities

Logical Knowledge in KGs
Highly expressive, diverse
family of logical models.

Graph Neural Networks
Using the KG structure
as a neural network.

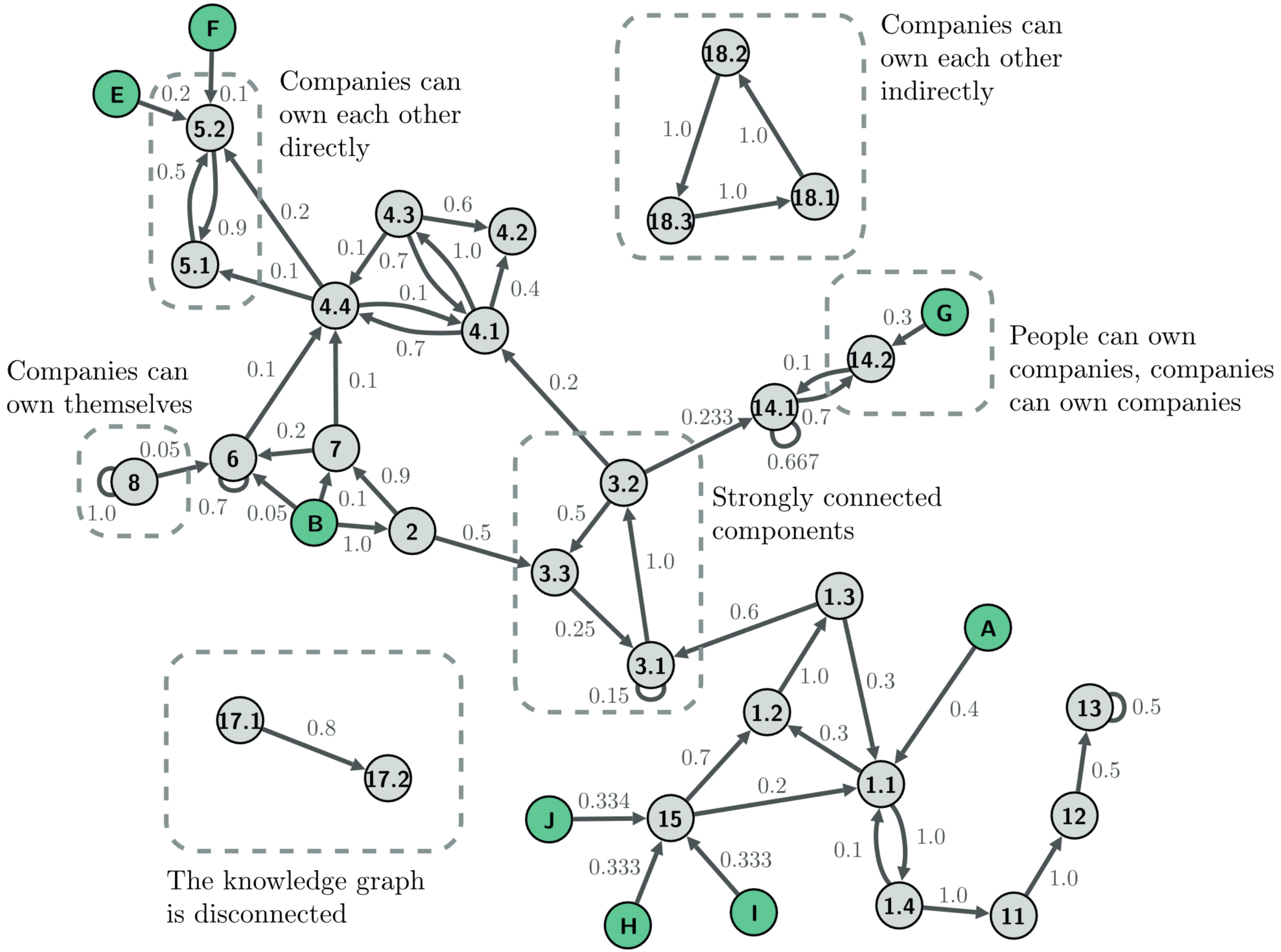
KG Embeddings
Widely-applied, large
family of ML models



Multiple, complex definitions of control, influence, ownership, etc., coexist

Logical Knowledge in KGs
Highly expressive, diverse family of logical models.

KG Embeddings
Widely-applied, large family of ML models



Companies can own each other directly

Companies can own each other indirectly

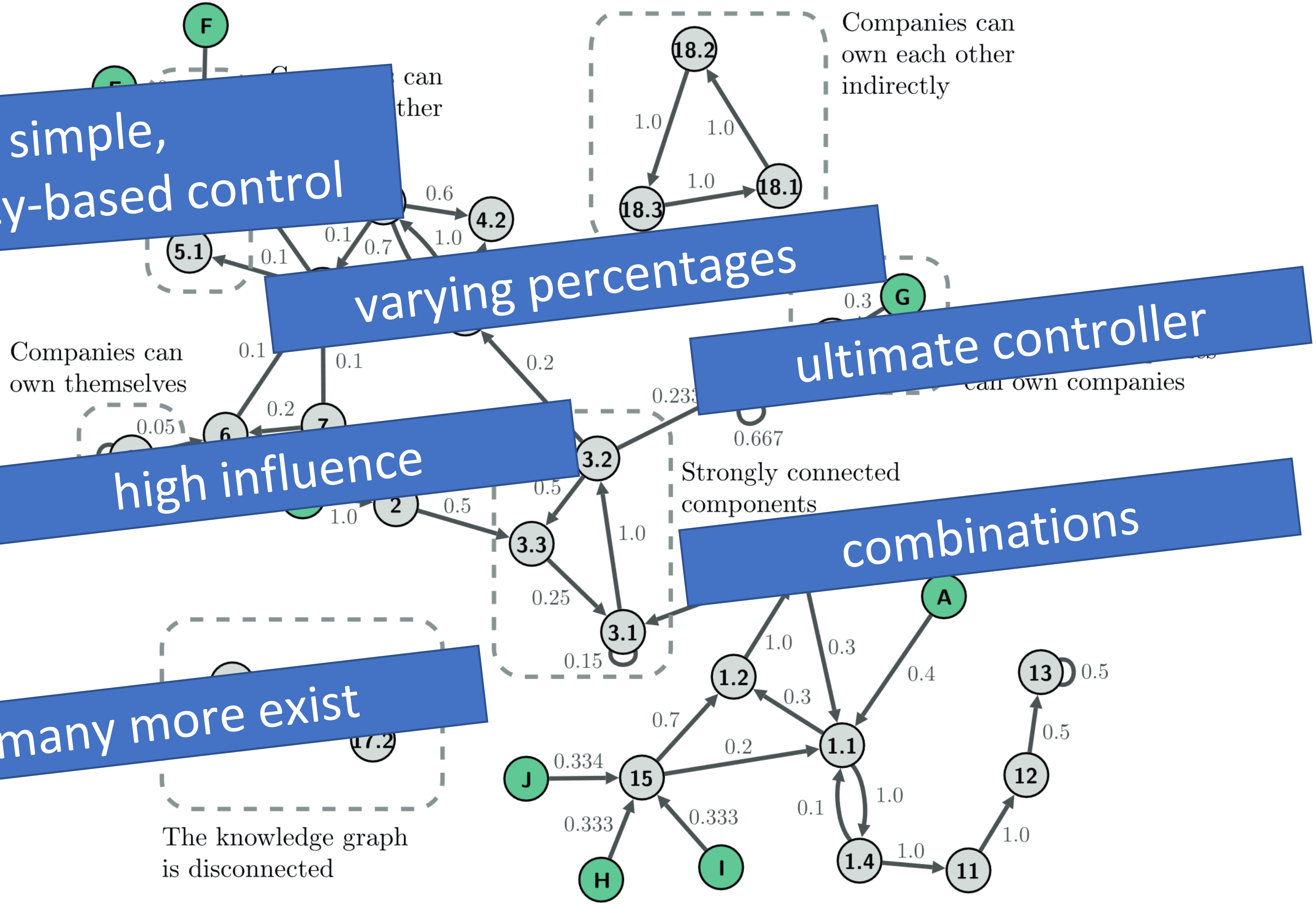
People can own companies, companies can own companies

Companies can own themselves

Strongly connected components

The knowledge graph is disconnected

Source: https://arxiv.org/abs/2004.10119



Companies can own each other indirectly

Companies can own themselves

can own companies

Strongly connected components

The knowledge graph is disconnected

simple, majority-based control

varying percentages

ultimate controller

high influence

combinations

many more exist

Source: https://arxiv.org/abs/2004.10119

x **controls** y if

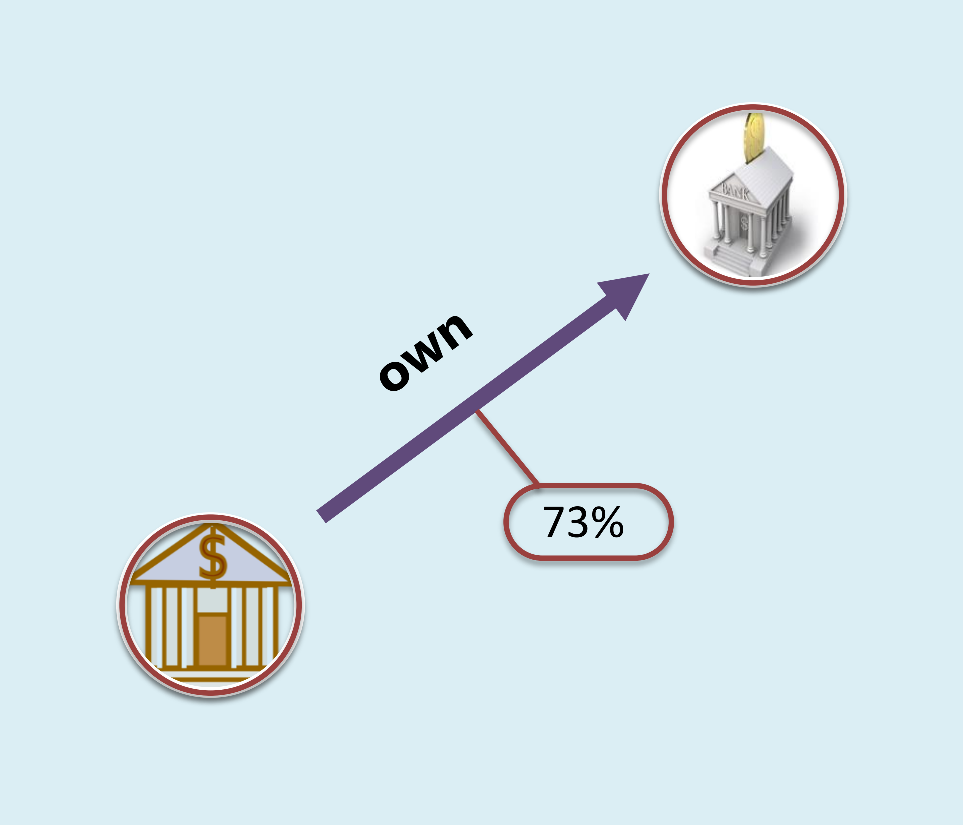
x **directly holds** over 50% of y

First-order Logic

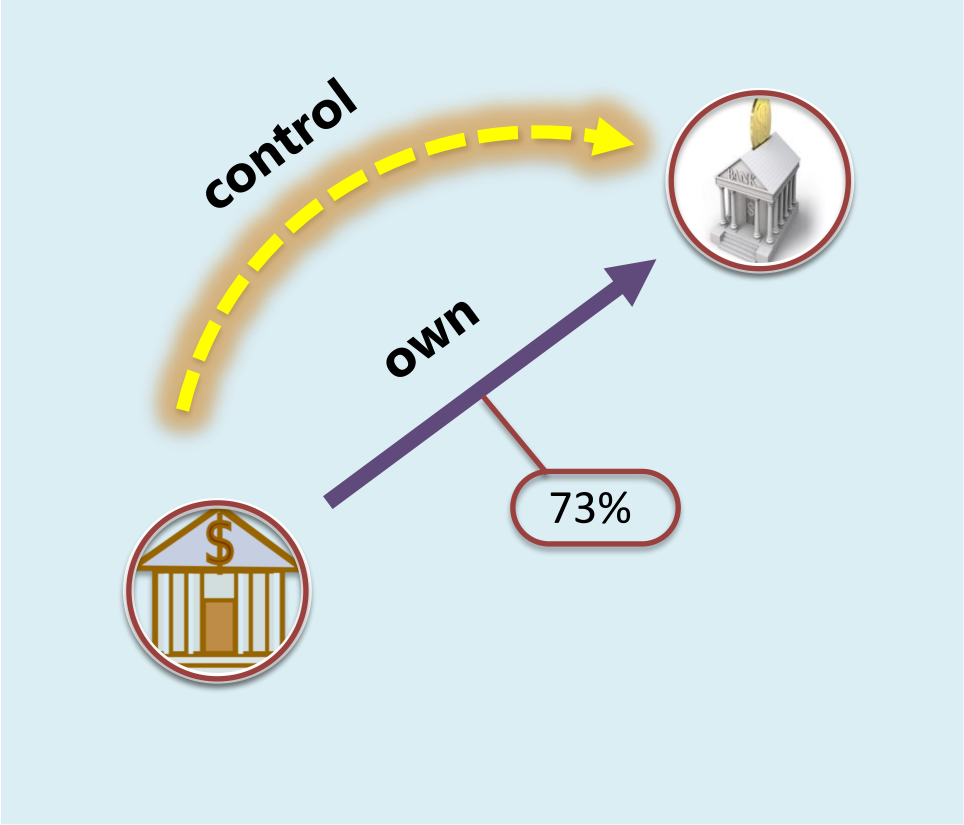
$$\forall x, y\ (own(x, y, w), w > 0.5 \rightarrow control(x, y))$$

x **controls** y if

x **directly holds** over 50% of y



→



x **controls** y if

  x **directly holds** over 50% of y

Datalog

```
control(X,Y) :- own(X,Y,W), W > 0.5.
```

x **controls** y if

x **directly holds** over 50% of y

SQL

```
SELECT x,y INTO control
FROM company
WHERE w > 0.5
```

x **controls** y if

x **directly holds** over 50% of y

Relational Calculus

$$control = \{(x, y) \mid own(x, y, w), w > 0.5\}$$

x **controls** y if
   x **directly holds** over 50% of y

Relational Algebra

$$control = \sigma_{w>0.5}\ own$$

x **controls** y if
x **directly holds** over 50% of y

Cypher (Graph DBs)

```
MATCH  (x:Company) –[o:OWN]-> (y:Company)]
WHERE o.w > 0.5
CREATE (x) –[:CONTROL]-> (y)
```

Cypher

…

SPARQL

SQL

Datalog

First-order Logic

Relational Algebra

Control

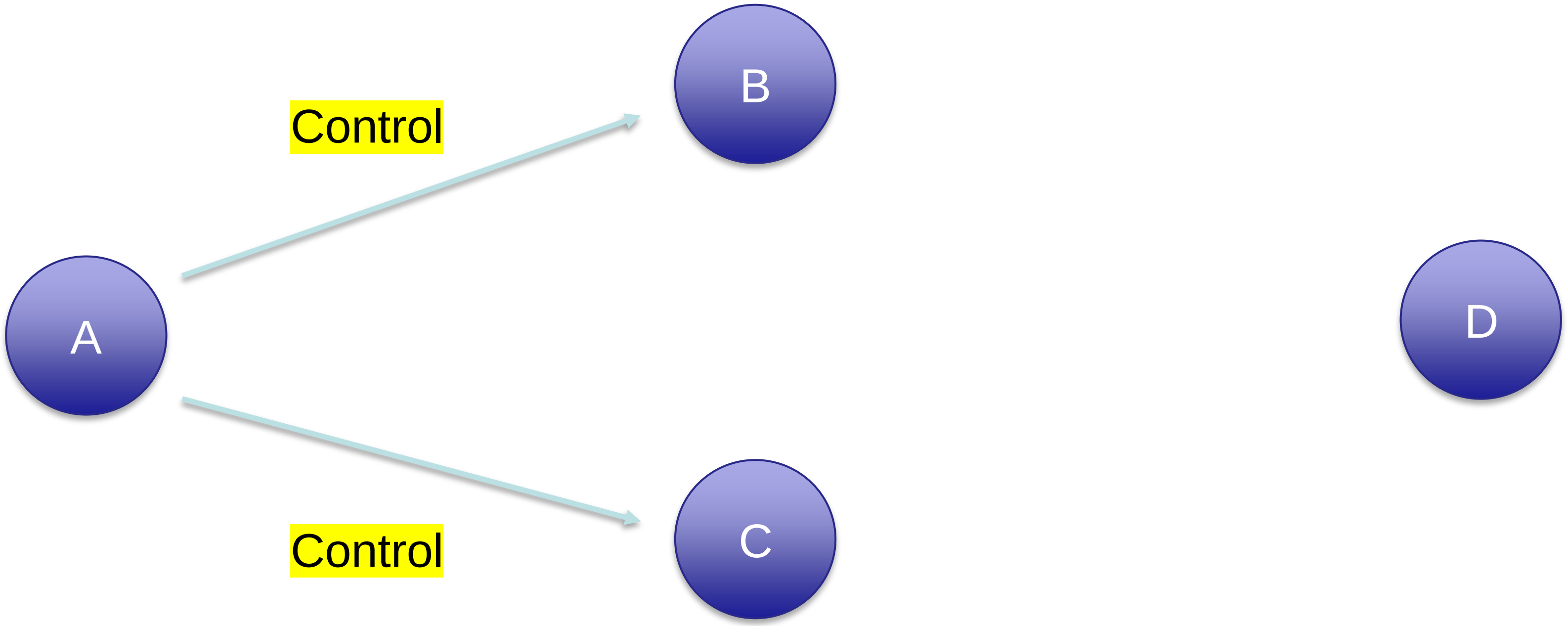
Example: simple, majority-based control

A bank or intermediary x controls another bank or intermediary y if (i) x directly owns more than 50% of y; or (ii) x controls a set of banks or intermediaries that jointly (i.e., summing the shares), and possibly together with x, own more than 50% of y

$$Bank(x) \rightarrow Control(x,x)$$
$$Control(x,y), Own(y,z,w), v = msum(w, \langle y \rangle), v > 0.5 \rightarrow Control(x,z)$$

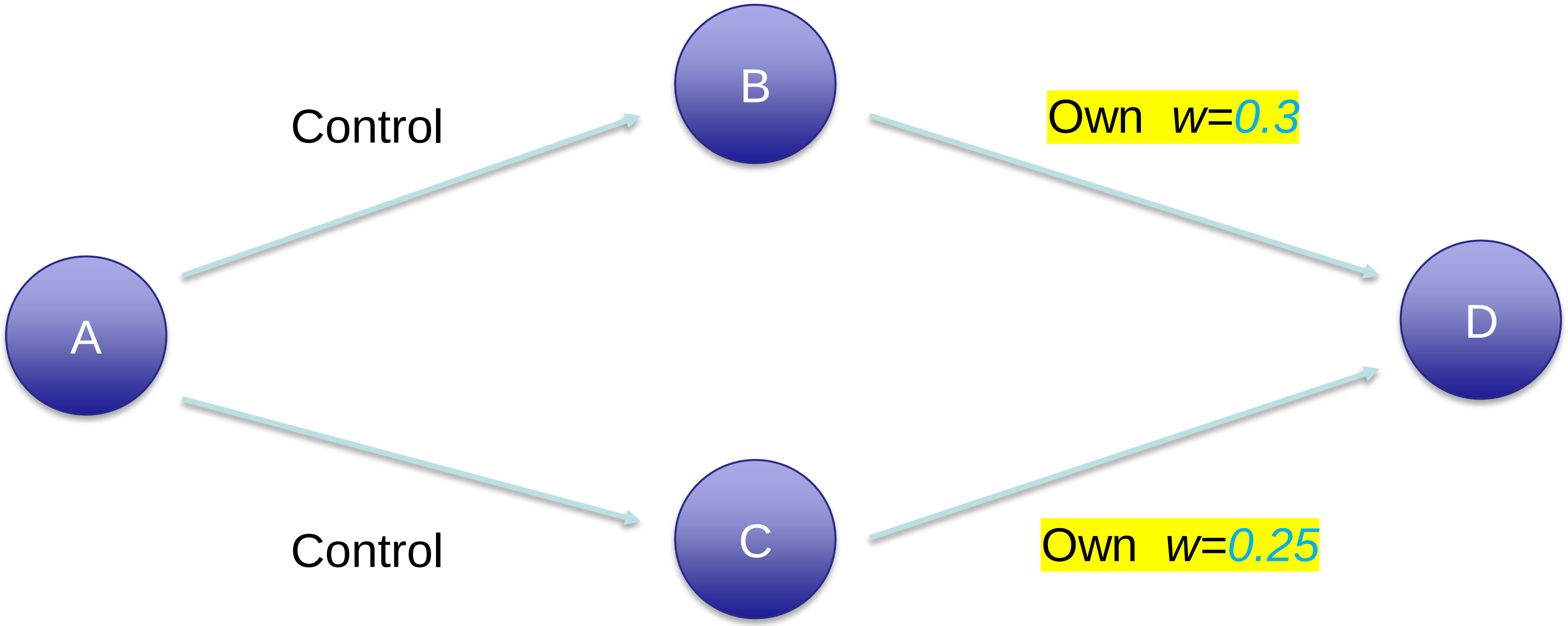
$Bank(x) \rightarrow Control(x,x)$

$Control(x,y), Own(y,z,w), v = msum(w,\langle y\rangle), v > 0.5 \rightarrow Control(x,z)$



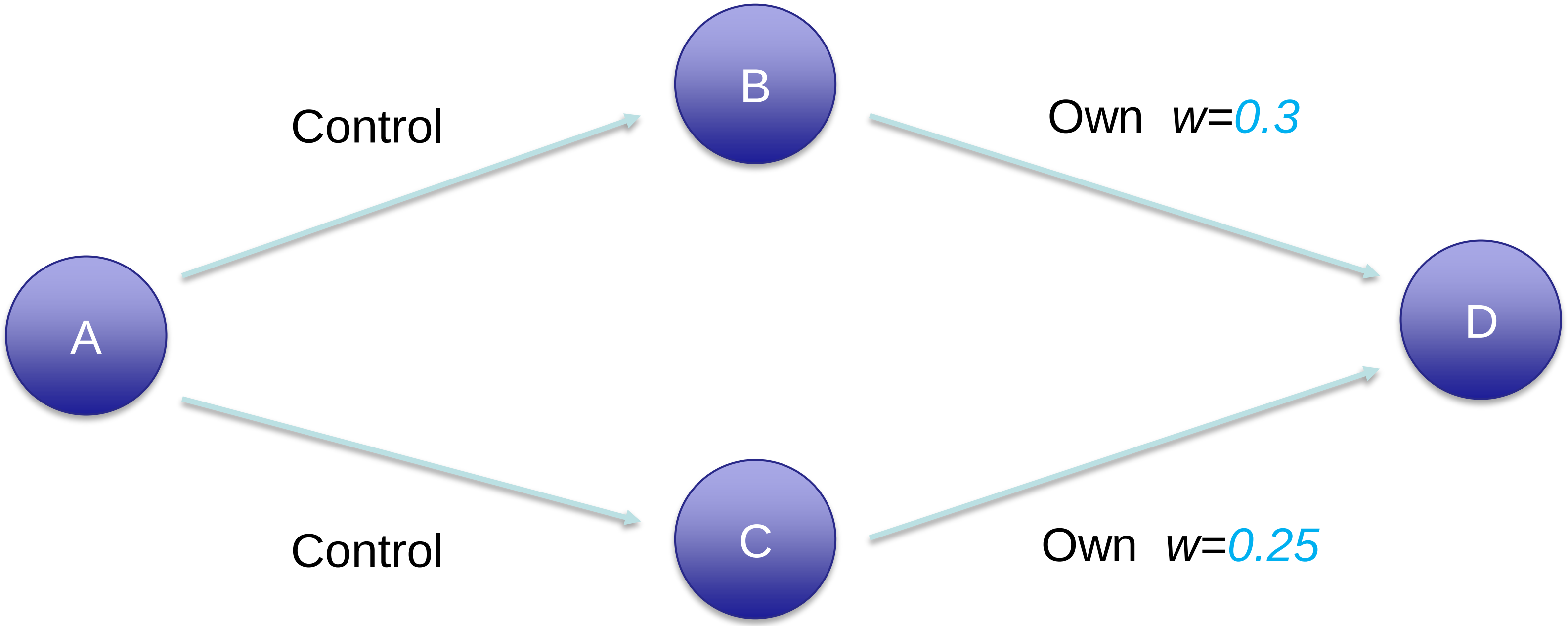
$$Bank(x) \rightarrow Control(x,x)$$

$$Control(x,y), Own(y,z,w), v = msum(w, \langle y \rangle), v > 0.5 \rightarrow Control(x,z)$$



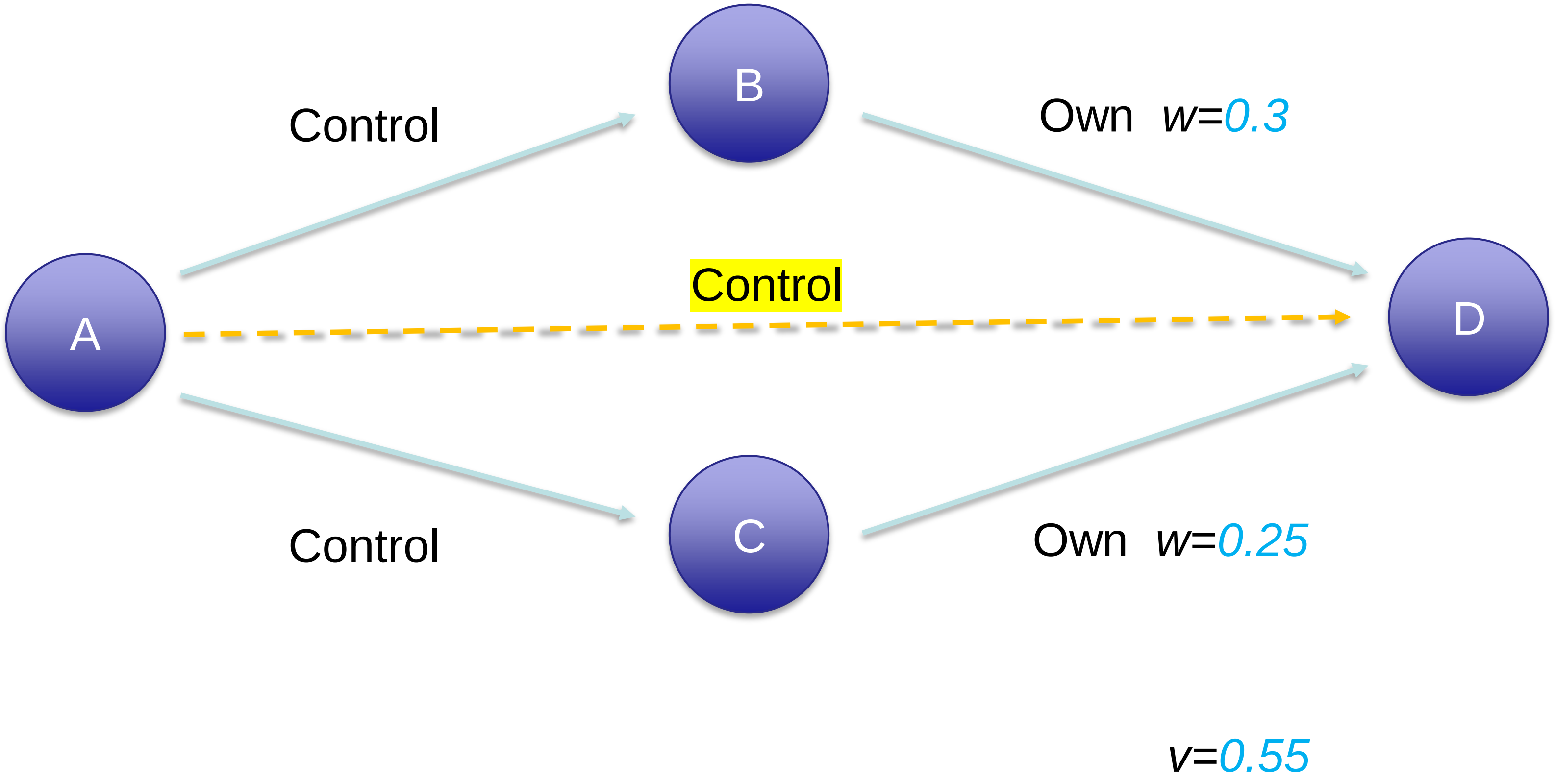
$$Bank(x) \rightarrow Control(x,x)$$

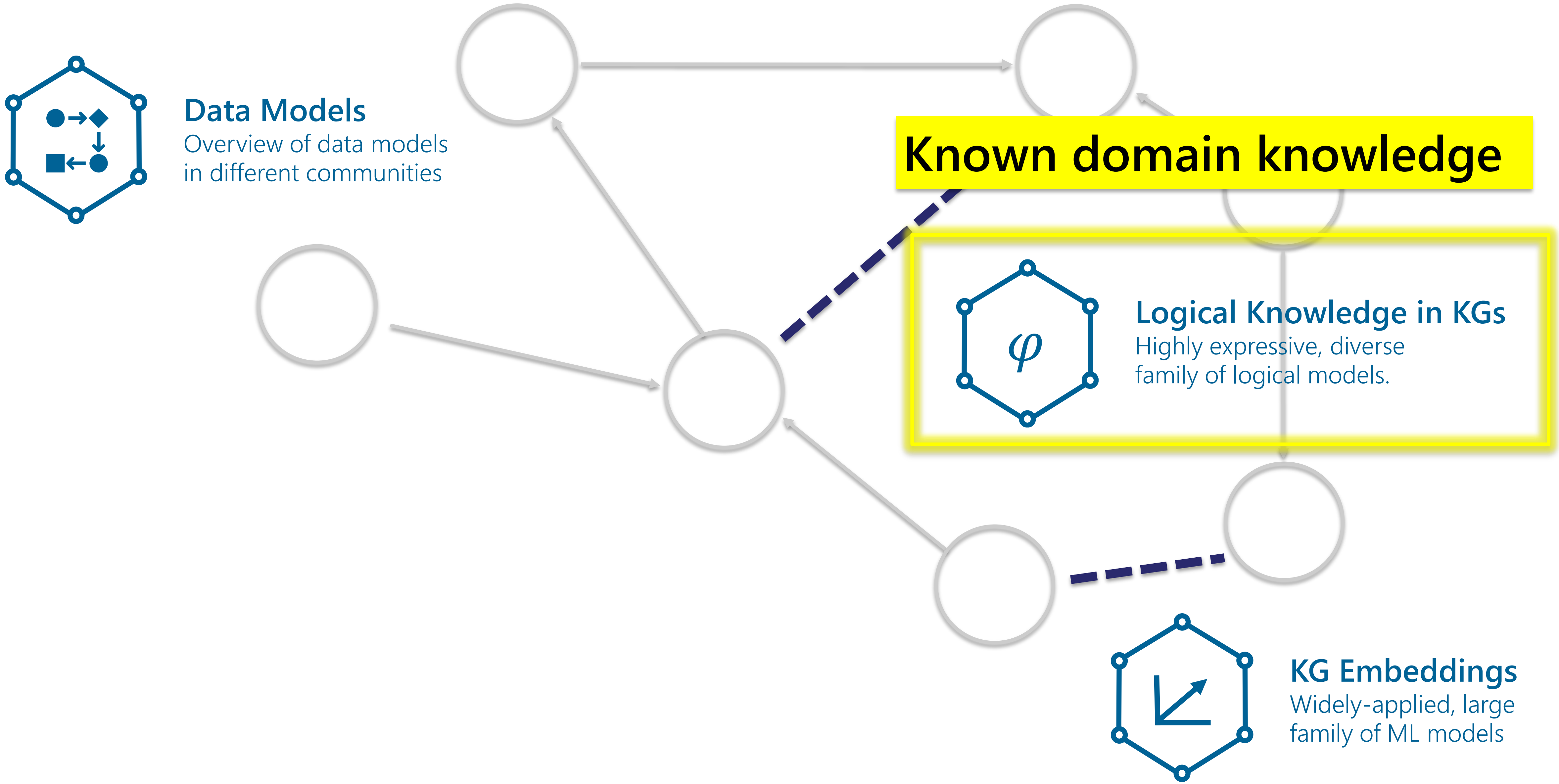
$$Control(x,y), Own(y,z,w), v = msum(w,\langle y\rangle), v > 0.5 \rightarrow Control(x,z)$$



v=0.55

$$Bank(x) \rightarrow Control(x,x)$$
$$Control(x,y), Own(y,z,w), v = msum(w, \langle y \rangle), v > 0.5 \rightarrow Control(x,z)$$



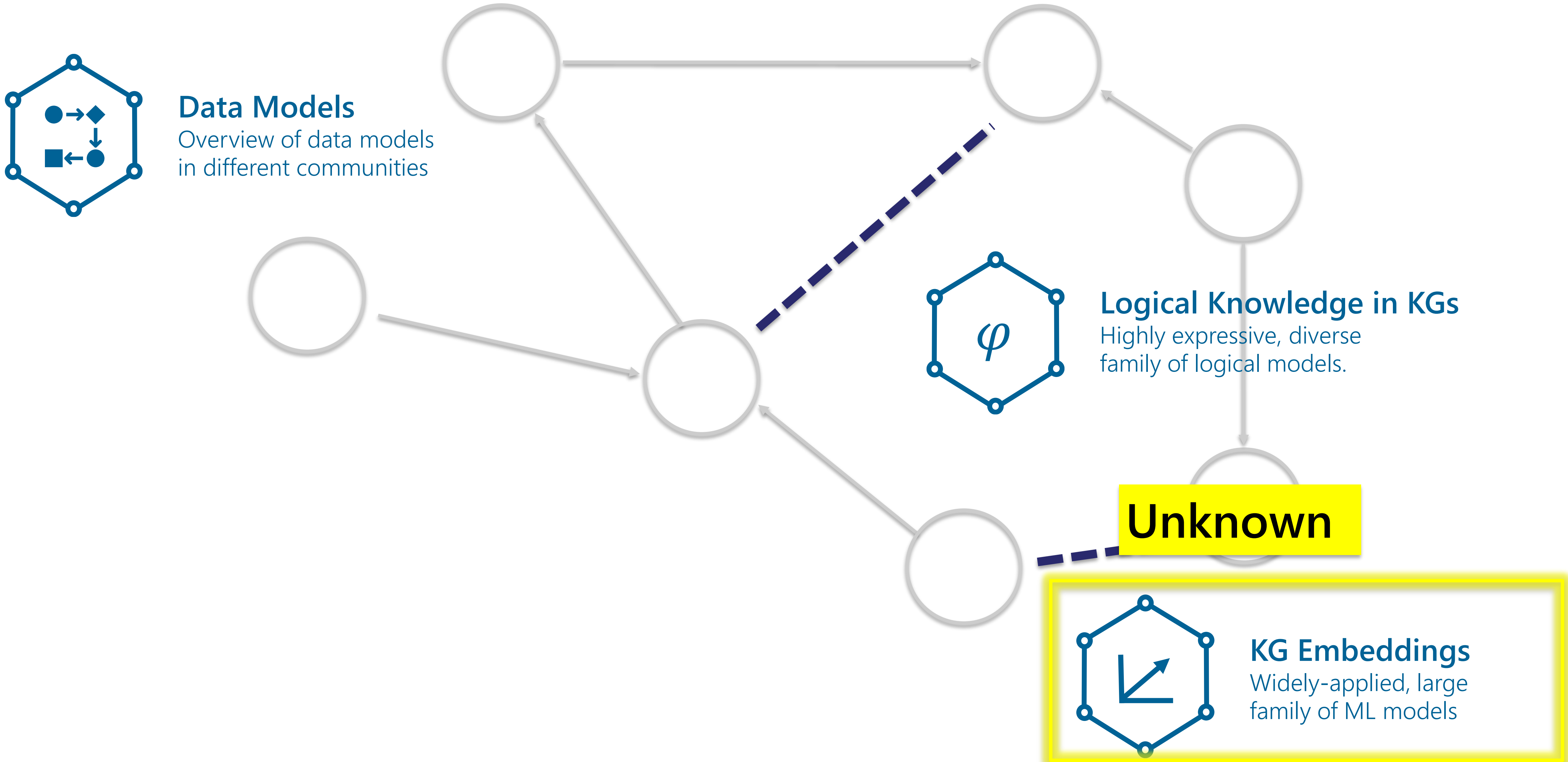


Known domain knowledge

Data Models
Overview of data models
in different communities

Logical Knowledge in KGs
Highly expressive, diverse
family of logical models.

KG Embeddings
Widely-applied, large
family of ML models



Data Models

Overview of data models in different communities

Logical Knowledge in KGs

Highly expressive, diverse family of logical models.

Unknown

KG Embeddings

Widely-applied, large family of ML models

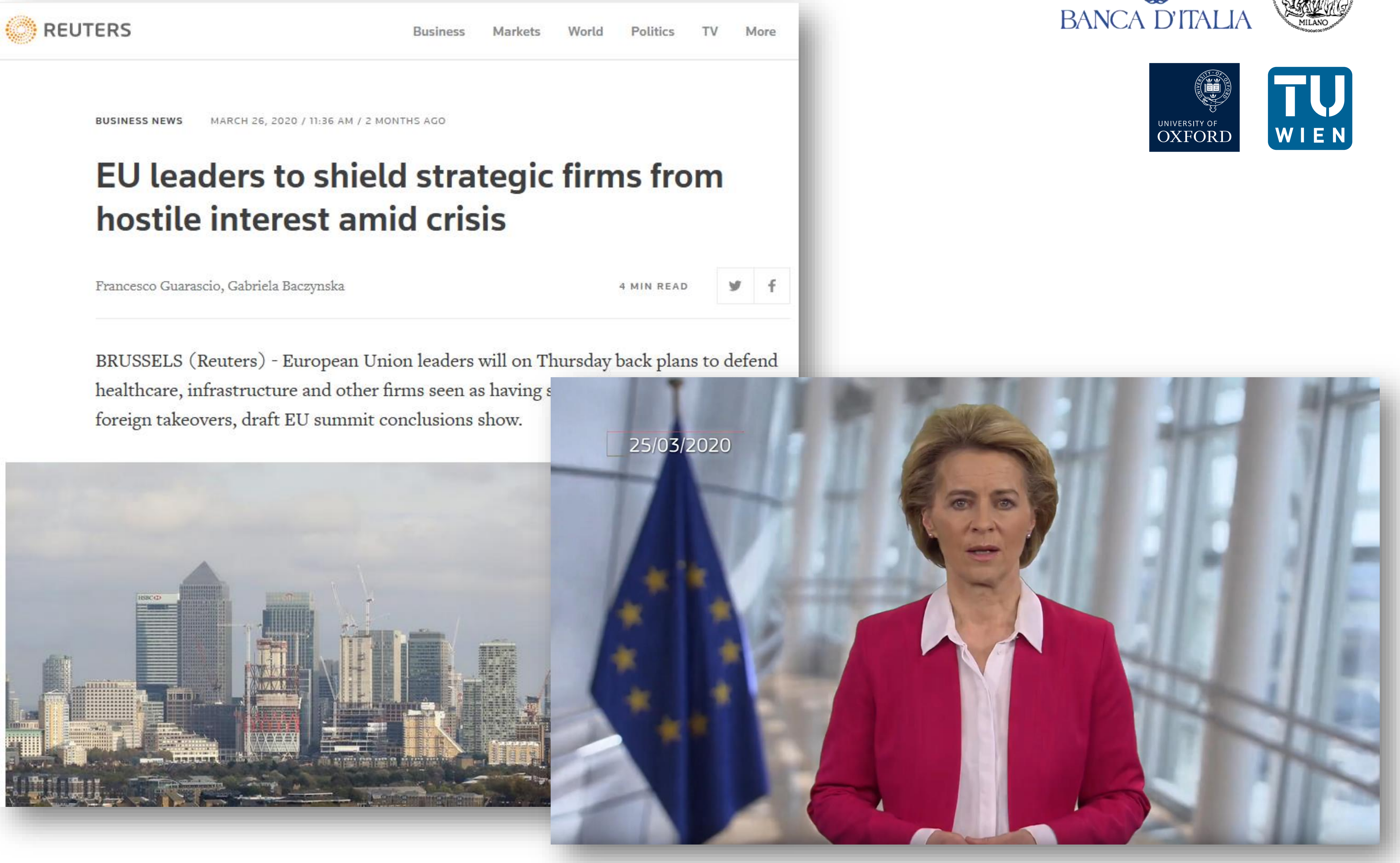
Hostile Takeovers

BANCA D'ITALIA

POLITECNICO MILANO

UNIVERSITY OF OXFORD

TU WIEN



BUSINESS NEWS MARCH 26, 2020 / 11:36 AM / 2 MONTHS AGO

EU leaders to shield strategic firms from hostile interest amid crisis

Francesco Guarascio, Gabriela Baczynska

BRUSSELS (Reuters) - European Union leaders will on Thursday back plans to defend healthcare, infrastructure and other firms seen as having s foreign takeovers, draft EU summit conclusions show.

25/03/2020

Creation

BANCA D'ITALIA

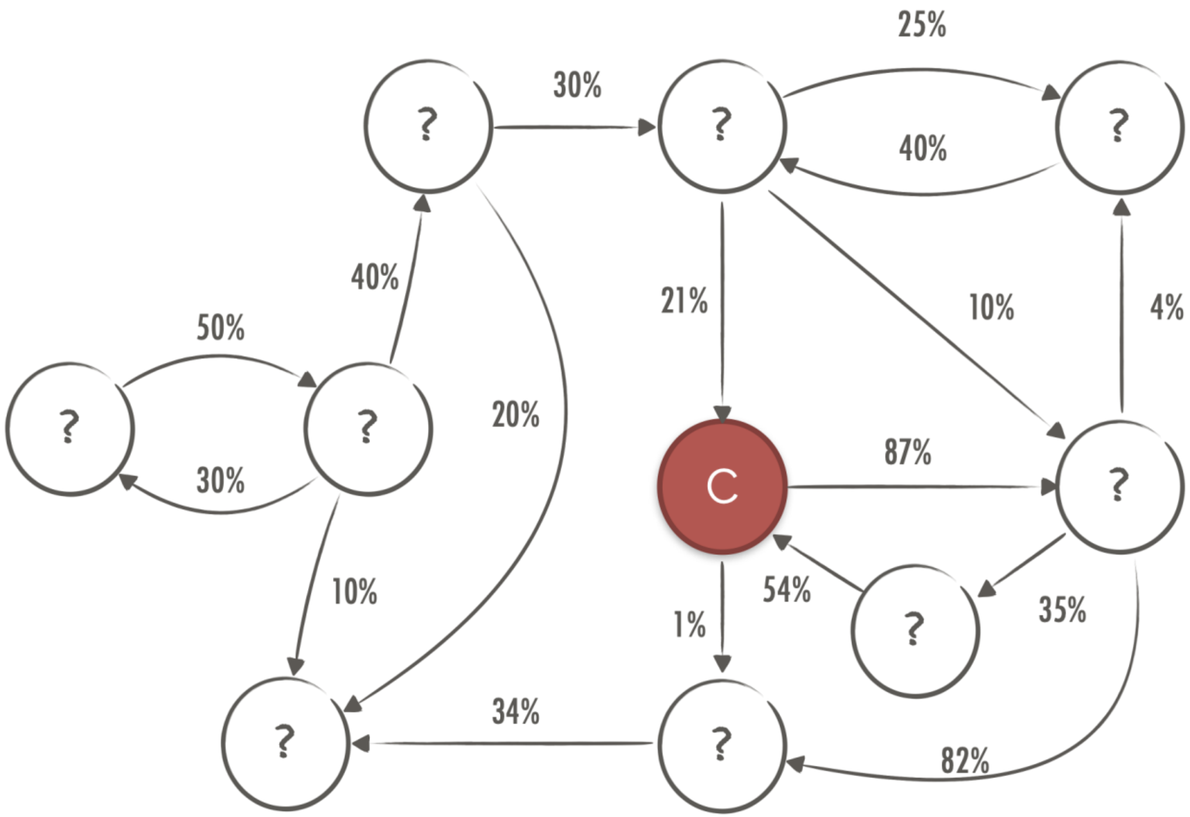
- 10 million individuals
- 30 million ownerships
- 20 million roles (e.g. CEO)
- 200k company events (e.g. M&A)

- 264 million nodes
- 660 million edges
- 1+ billion properties

Plus, derived knowledge:

- company control
- relevant influence
- family links

Company Knowledge Graphs



- Who takes **decisions**?
- Who's the ultimate **beneficial owner**?
- Is there **collusion**?
- How does **risk** propagate?
- What are the **real cash flows**?

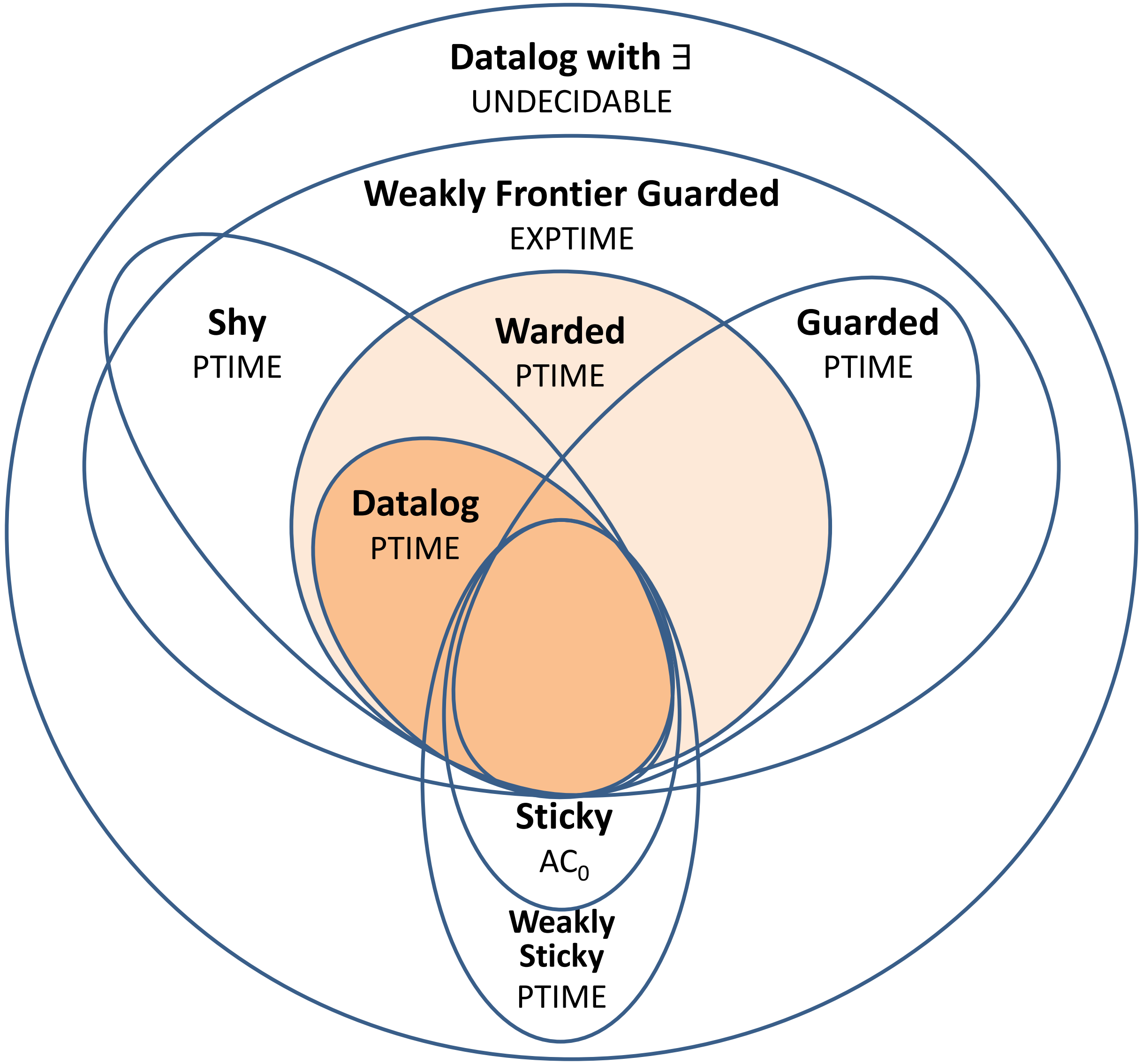
Integrated ownership

Company control

Close links

Logical Knowledge in KGs

Warded and Vadalog

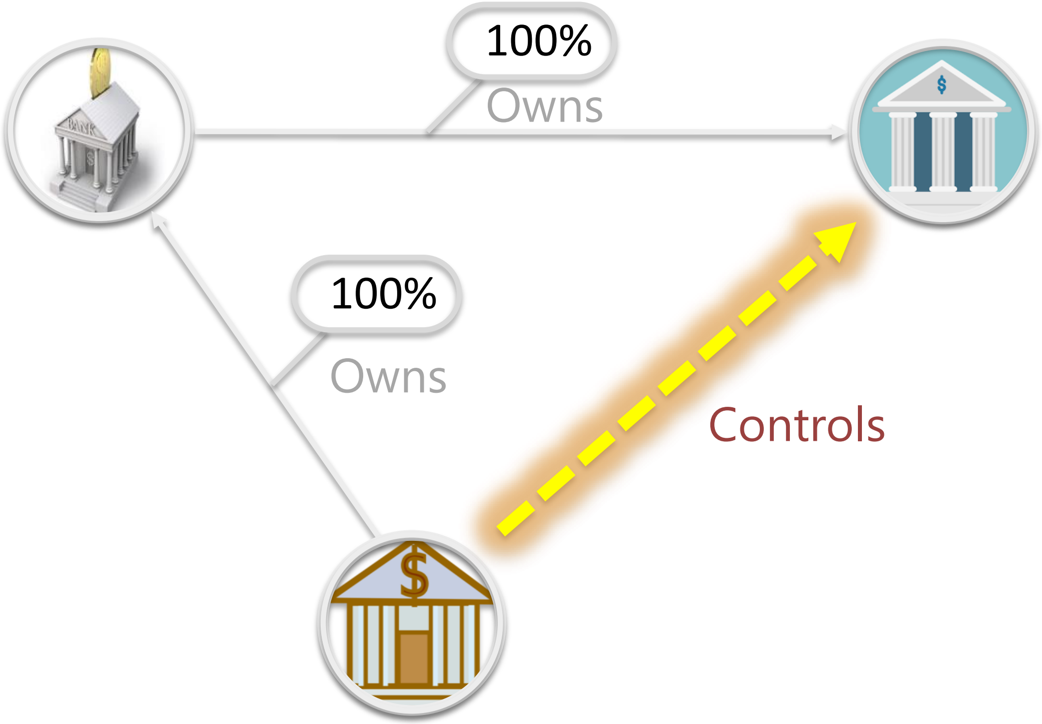

Datalog with ∃
UNDECIDABLE
Weakly Frontier Guarded
EXPTIME
Shy
PTIME
Warded
PTIME
Guarded
PTIME
Datalog
PTIME
Sticky
AC_0
Weakly Sticky
PTIME

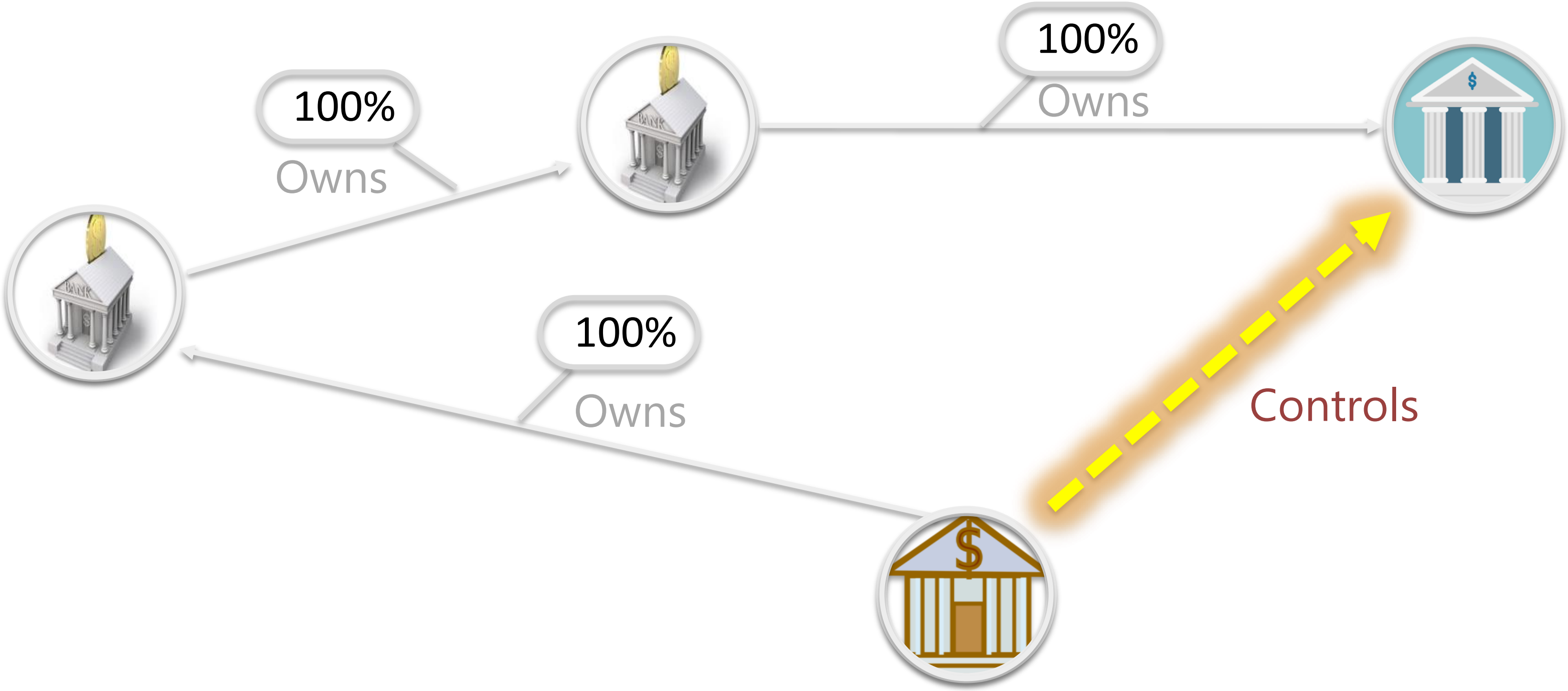
Vadalog Requirements

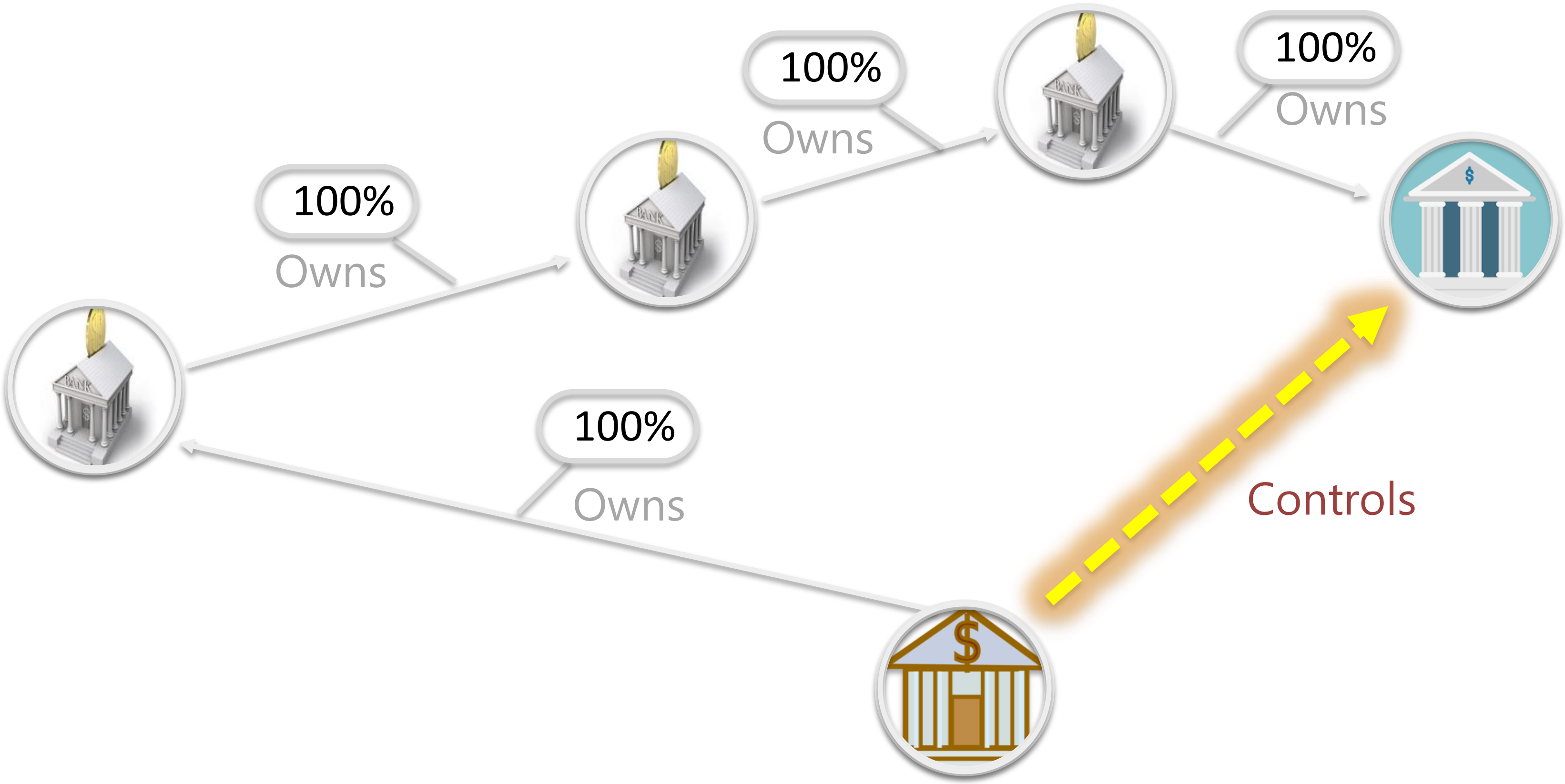
1. **Recursive** Reasoning: *full recursion over graphs*

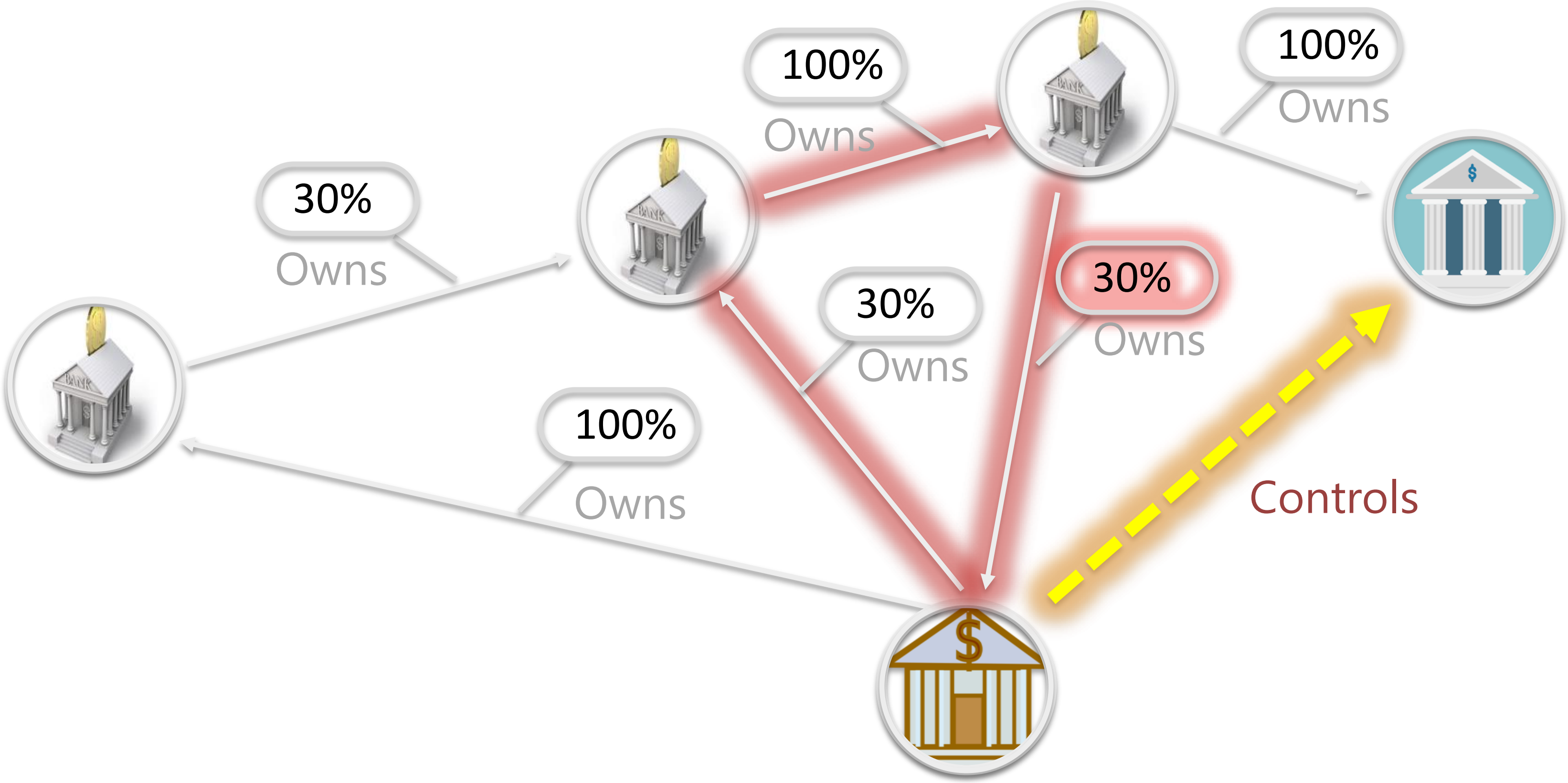
Vadalog Requirements

1. **Recursive** Reasoning: *full recursion over graphs*


100%
Owns
100%
Owns
Controls

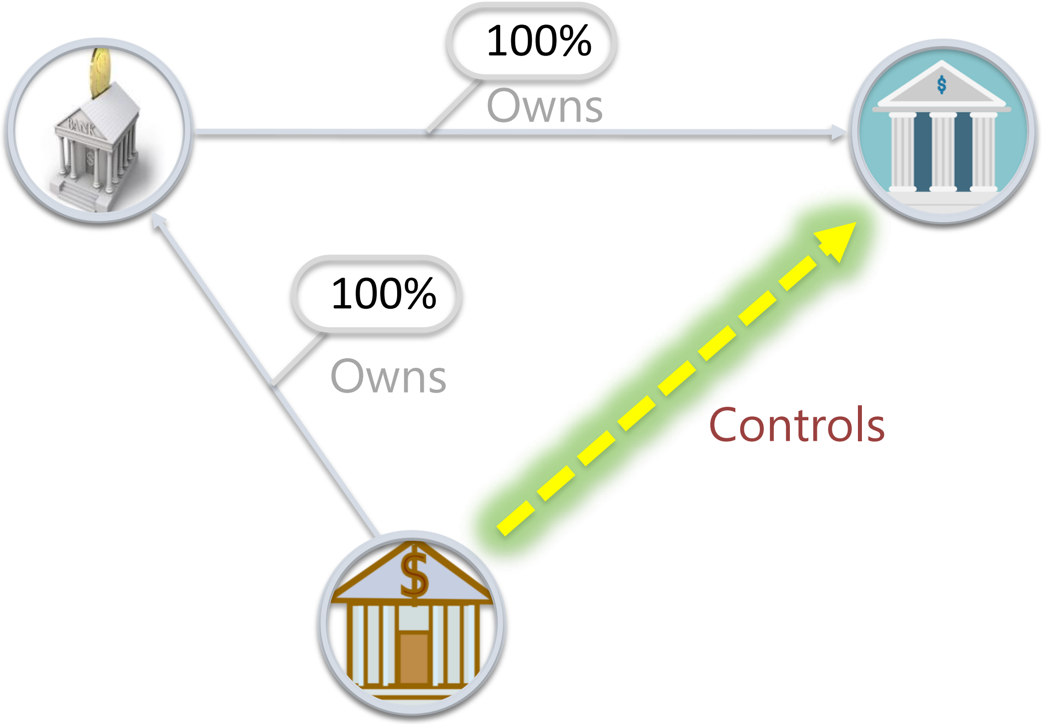

100%
Owns
100%
Owns
100%
Owns
Controls

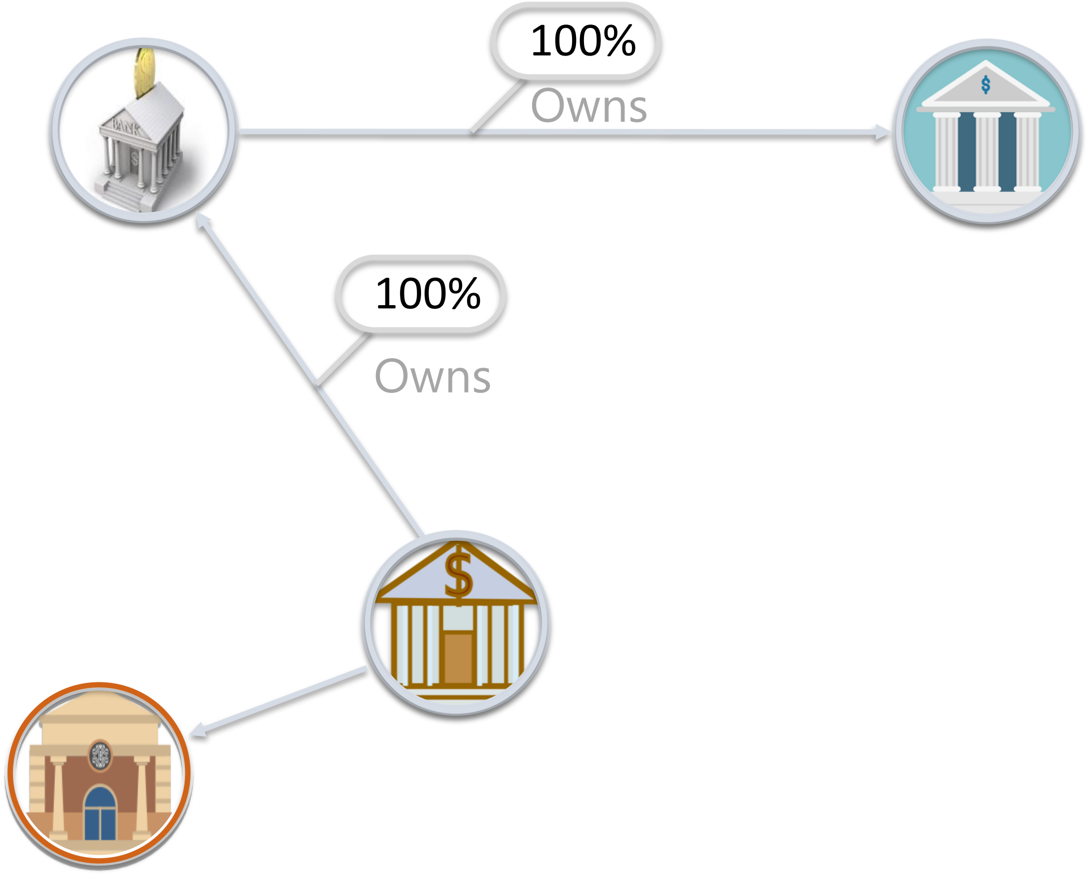

100%
Owns
100%
Owns
100%
Owns
100%
Owns
Controls

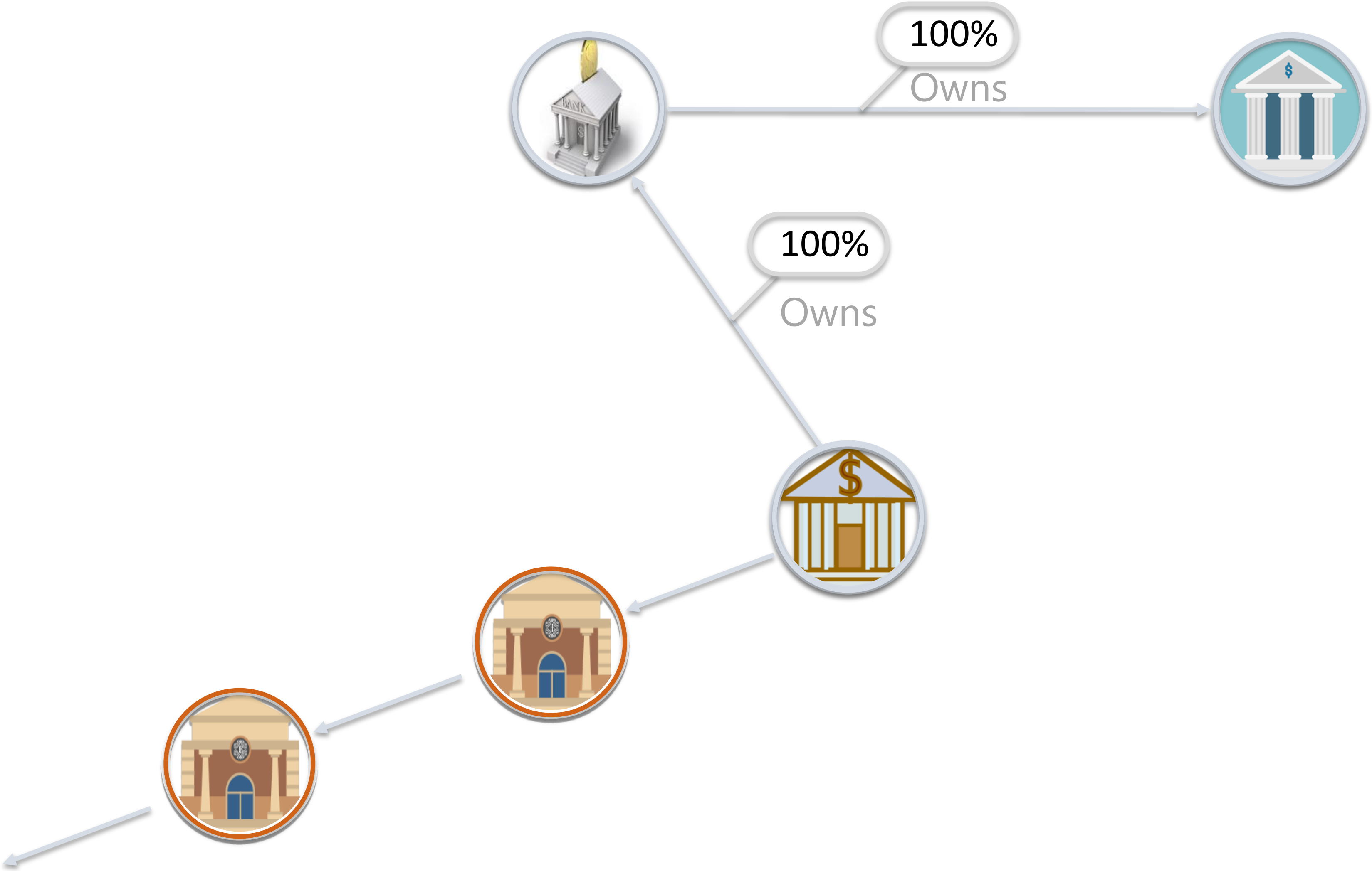

100%
Owns
100%
Owns
30%
Owns
30%
Owns
30%
Owns
100%
Owns
Controls

Vadalog Requirements

1. **Recursive** Reasoning: *full recursion over graphs*
2. **Ontological** Reasoning: *object creation*, …


100%
Owns
100%
Owns
Controls


100%
Owns
100%
Owns


100%
Owns
100%
Owns

Vadalog Requirements

1. **Recursive** Reasoning: *full recursion over graphs*
2. **Ontological** Reasoning: *object creation, …*

Vadalog Requirements

1. **Recursive** Reasoning: *full recursion over graphs*
2. **Ontological** Reasoning: *object creation, …*
3. **Numerical** Reasoning: *numeric computation and aggregation*
4. **Probabilistic** Reasoning: *uncertain information*
5. **Subsymbolic** Reasoning: *low-dimensional spaces*
6. **Temporal** Reasoning: *reasoning over time*
7. **Scalable** Reasoning: *coping with large datasets*

Vadalog

1. **Recursive** Reasoning:

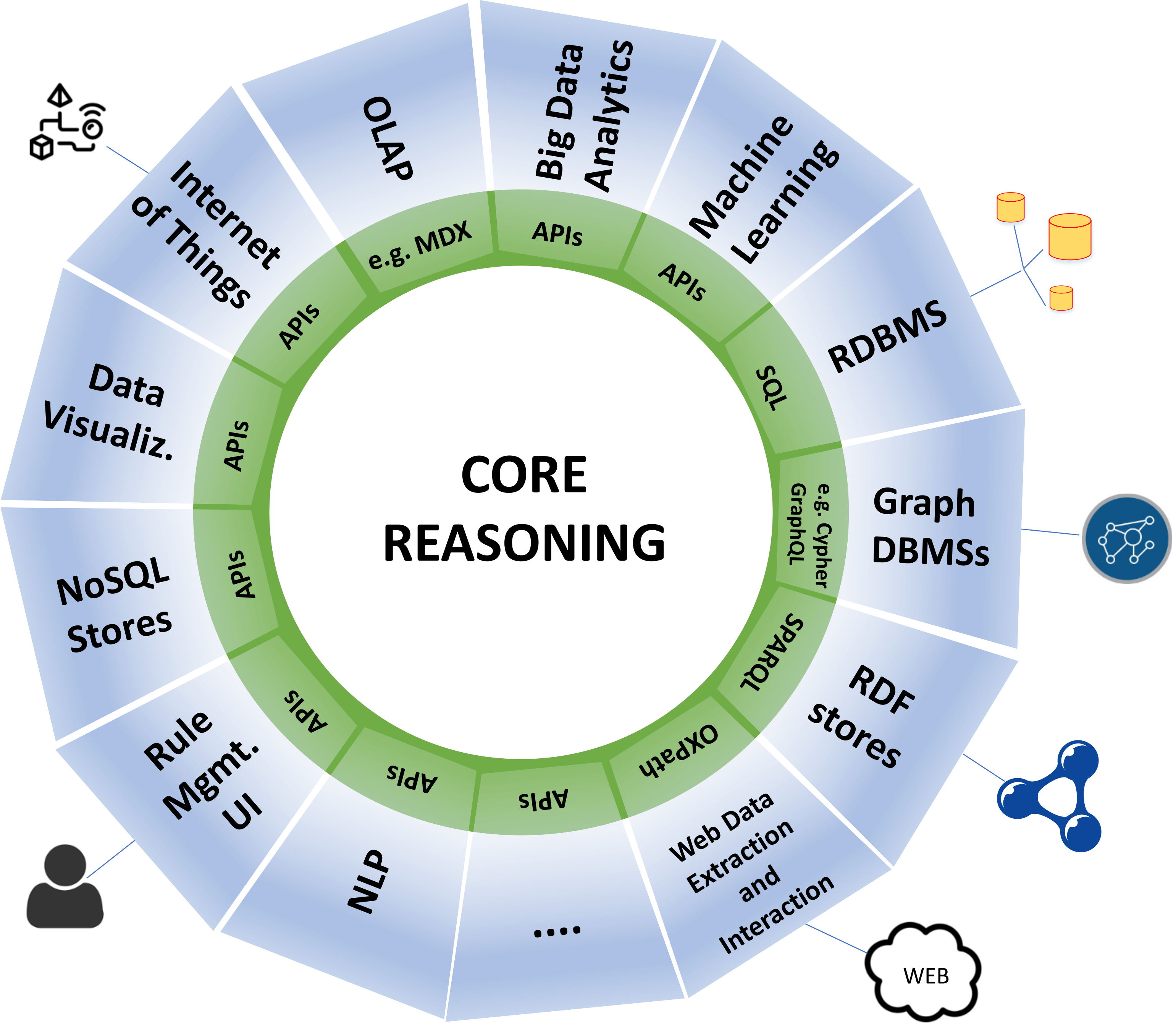
   Full support of recursive **Datalog**

2. **Ontological** Reasoning:

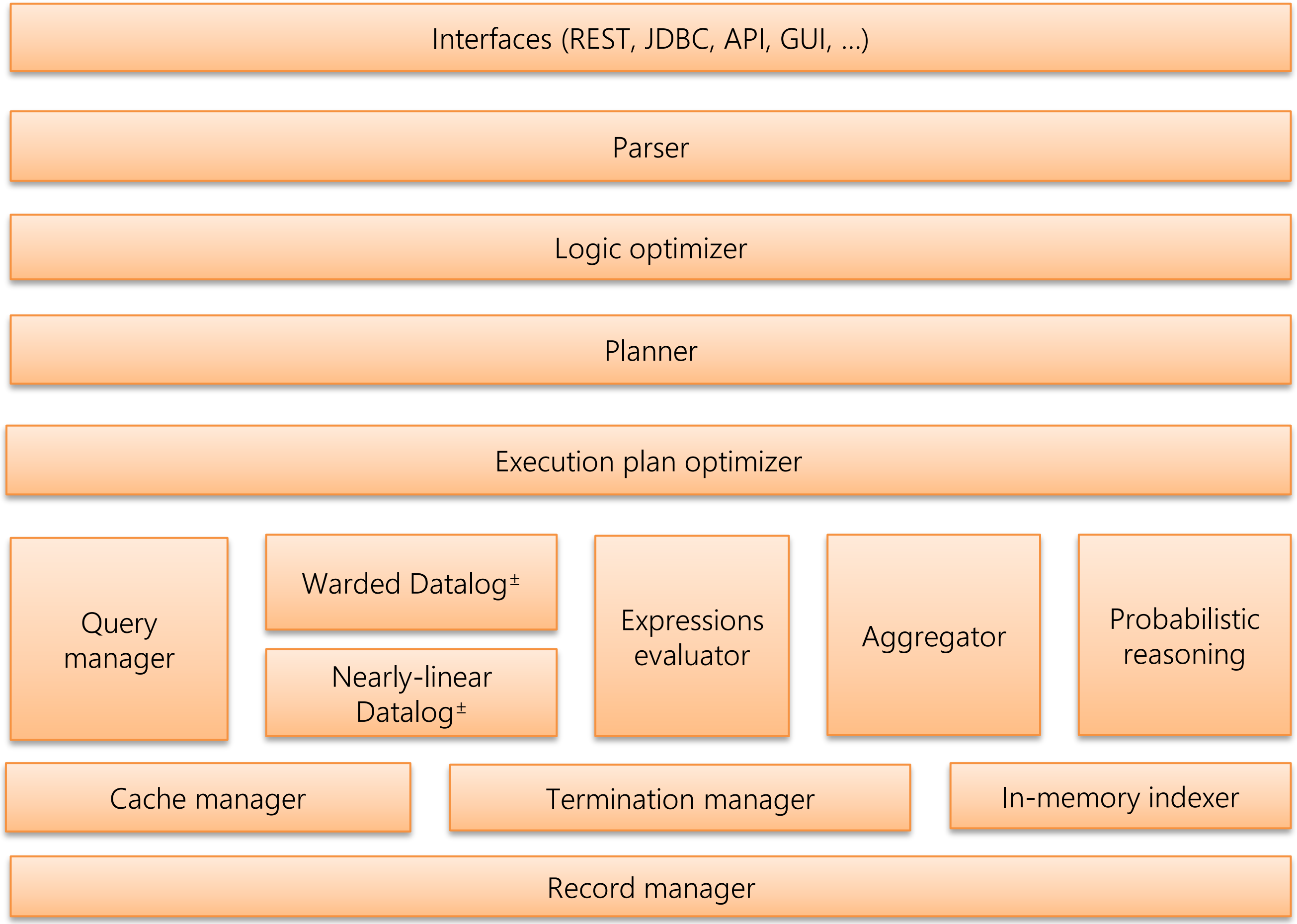
   Expressive power of **SPARQL** and **OWL 2 QL**

3. **Scalable** Reasoning: **polynomial time**,

   sub-fragments that are **fully parallelizable**


CORE
REASONING
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
....
APIs
NLP
APIs
Rule Mgmt. UI
APIs
NoSQL Stores
APIs
Data Visualiz.
APIs
Internet of Things
APIs

CORE REASONING



Interfaces (REST, JDBC, API, GUI, …)

Parser

Logic optimizer

Planner

Execution plan optimizer

Query manager

Warded Datalog$^{\pm}$

Nearly-linear Datalog$^{\pm}$

Expressions evaluator

Aggregator

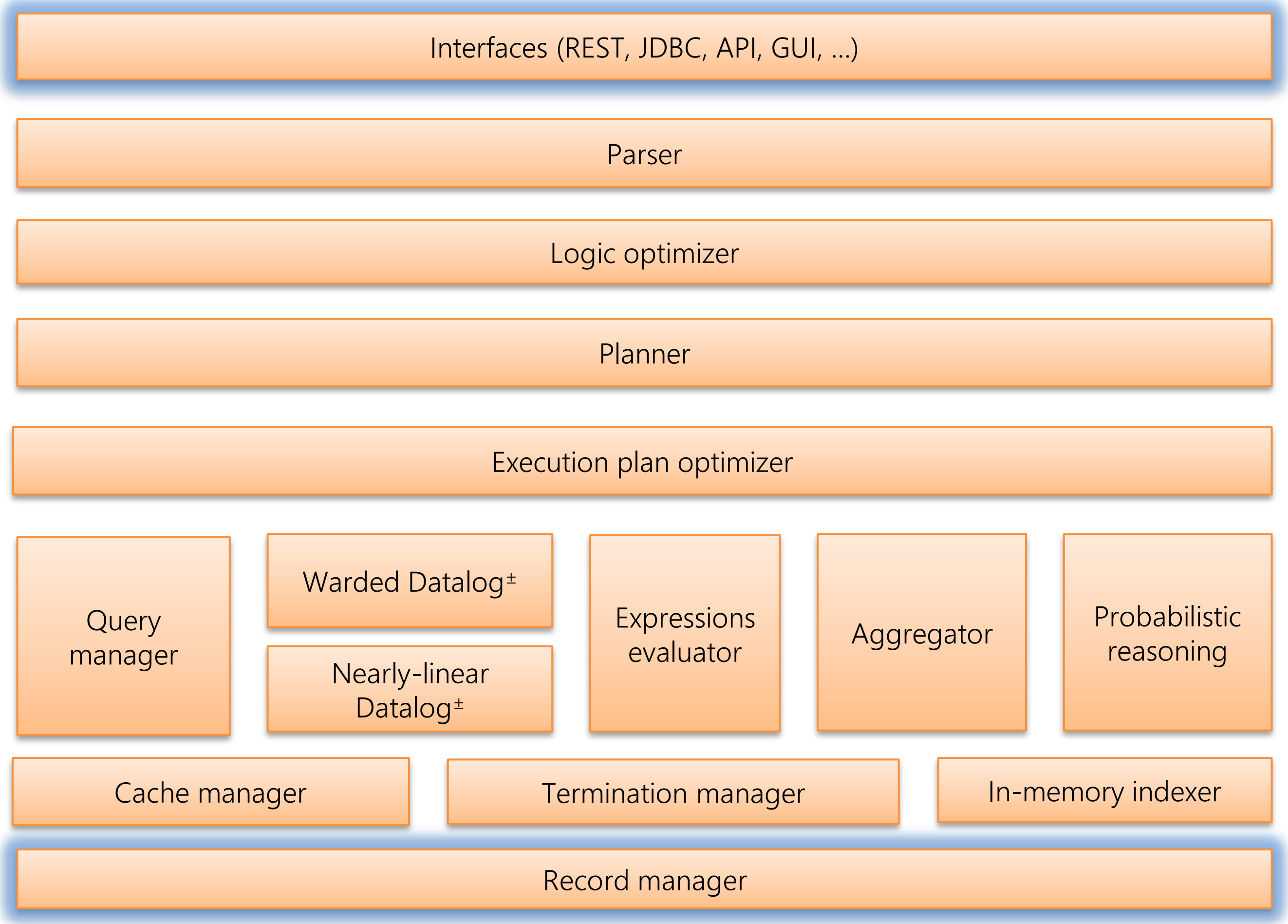
Probabilistic reasoning

Cache manager

Termination manager

In-memory indexer

Record manager



Interfaces (REST, JDBC, API, GUI, ...)

Parser

Logic optimizer

Planner

Execution plan optimizer

Query manager

Warded Datalog$^{\pm}$

Nearly-linear Datalog$^{\pm}$

Expressions evaluator

Aggregator

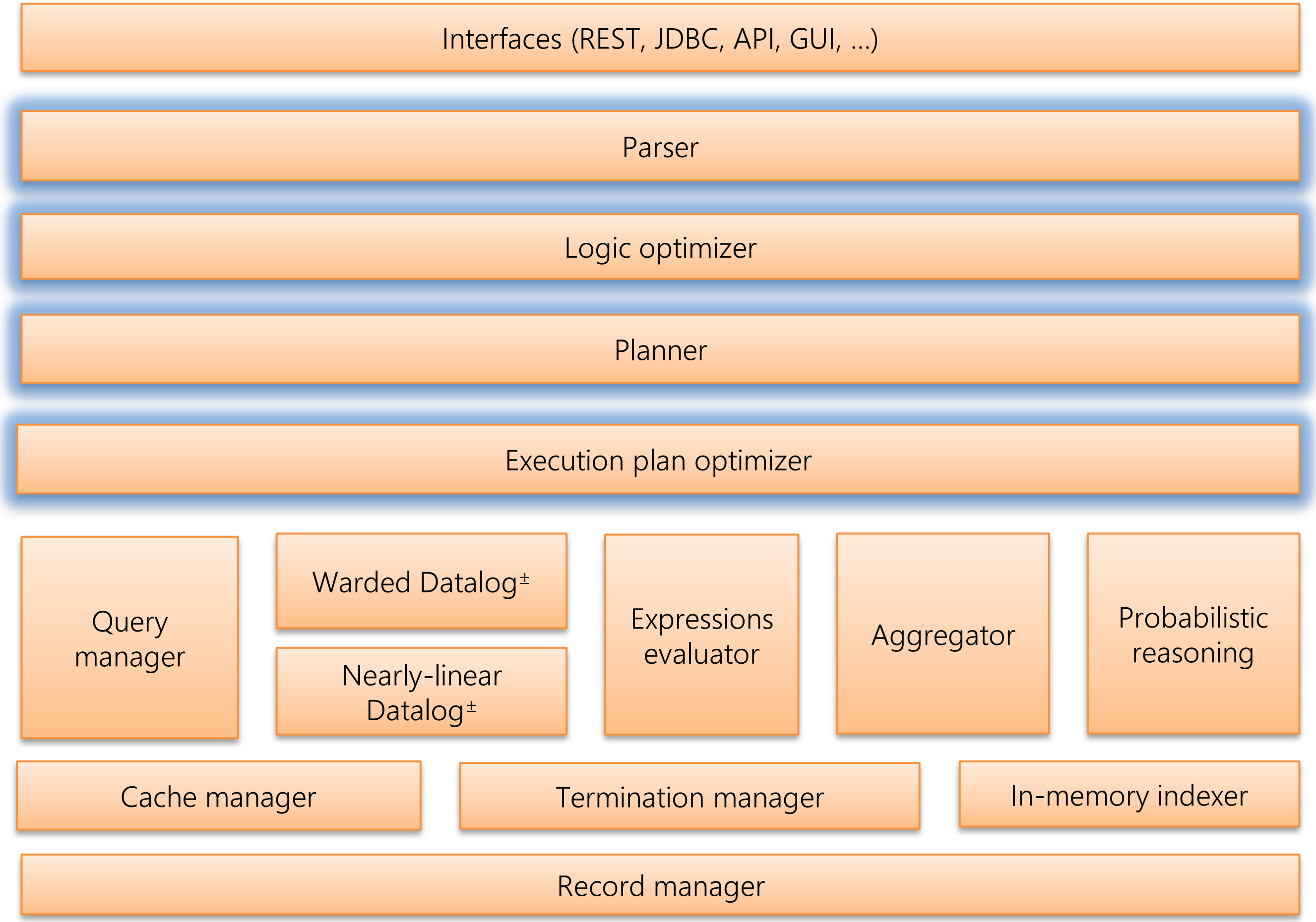
Probabilistic reasoning

Cache manager

Termination manager

In-memory indexer

Record manager



Interfaces (REST, JDBC, API, GUI, …)

Parser

Logic optimizer

Planner

Execution plan optimizer

Query manager

Warded Datalog$^{\pm}$

Nearly-linear Datalog$^{\pm}$

Expressions evaluator

Aggregator

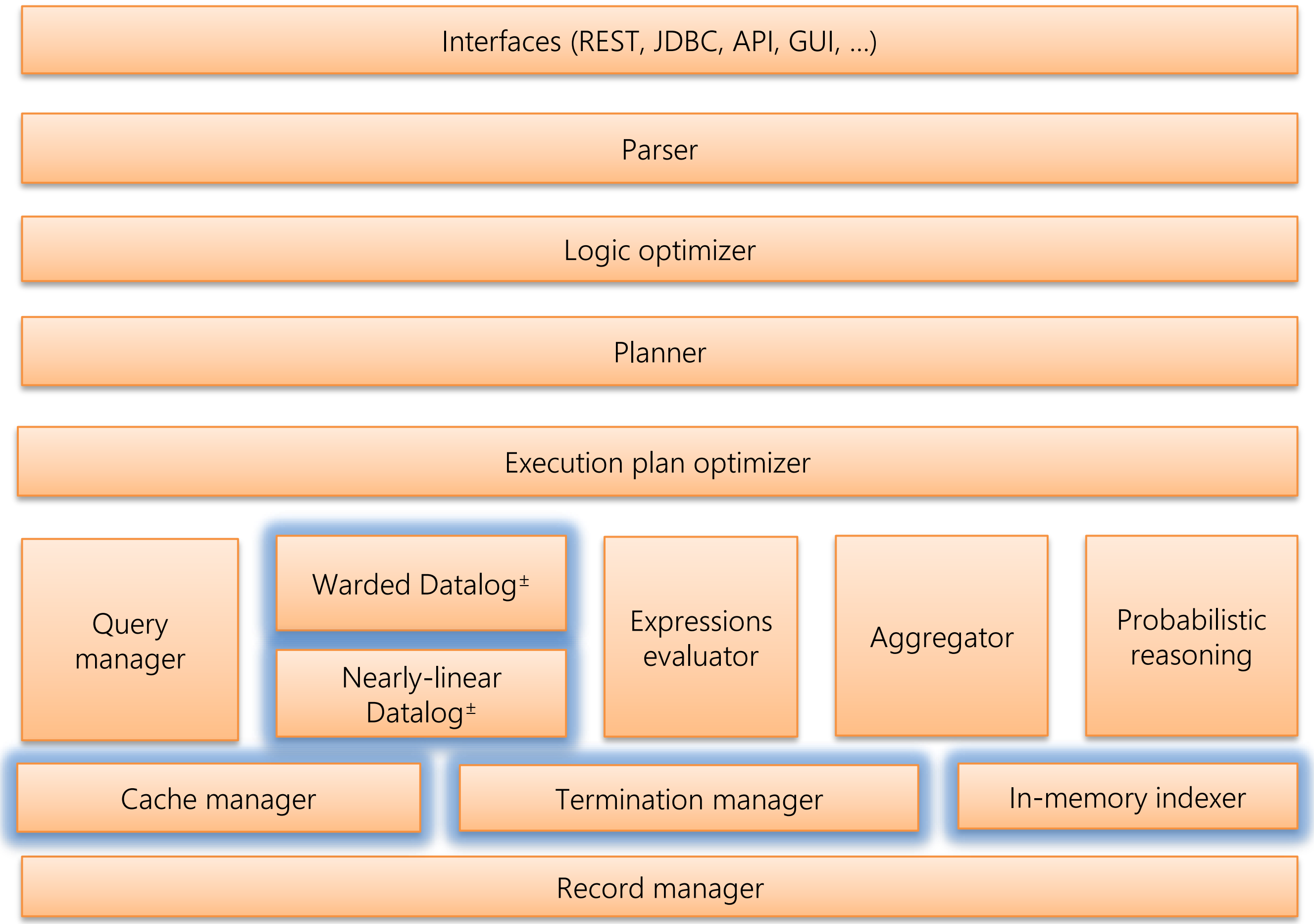
Probabilistic reasoning

Cache manager

Termination manager

In-memory indexer

Record manager



Interfaces (REST, JDBC, API, GUI, …)

Parser

Logic optimizer

Planner

Execution plan optimizer

Query manager

Warded Datalog$^{\pm}$

Nearly-linear Datalog$^{\pm}$

Expressions evaluator

Aggregator

Probabilistic reasoning

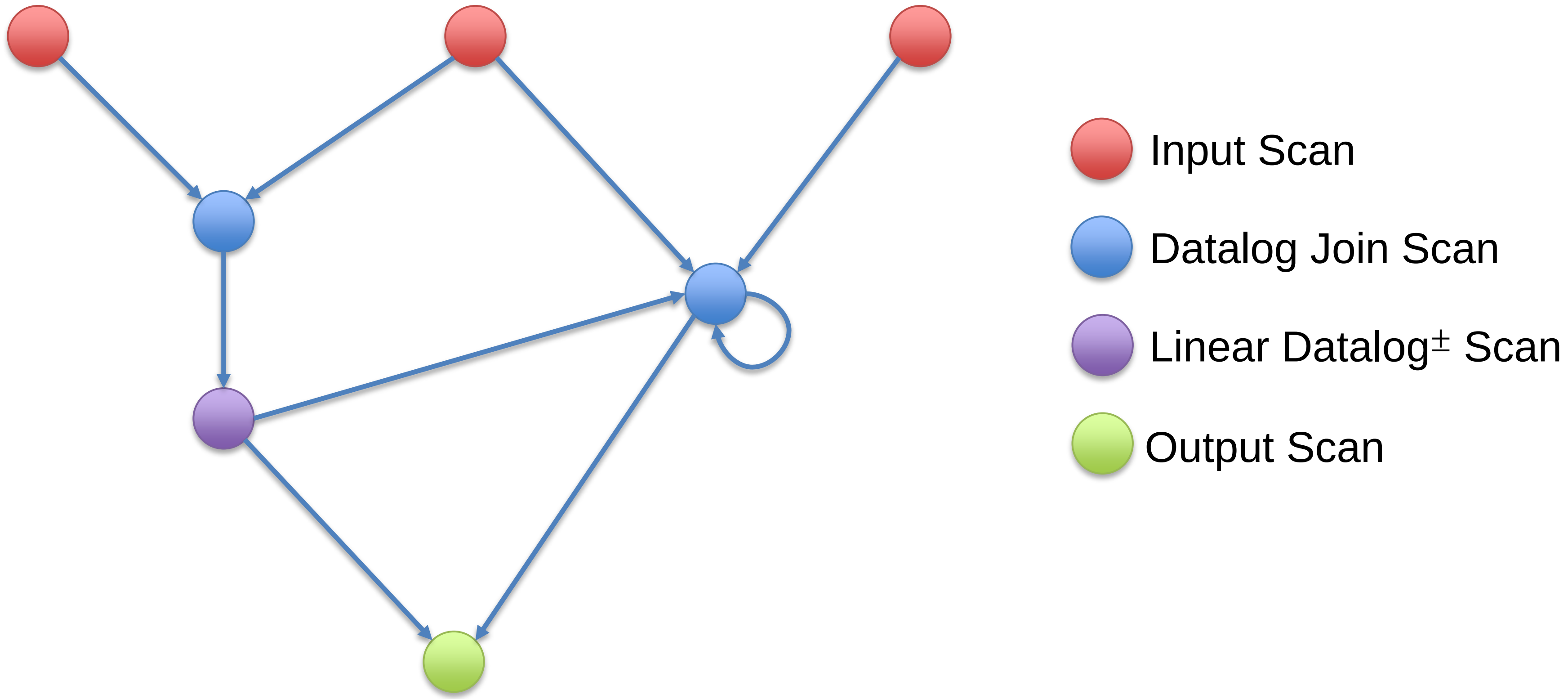
Cache manager

Termination manager

In-memory indexer

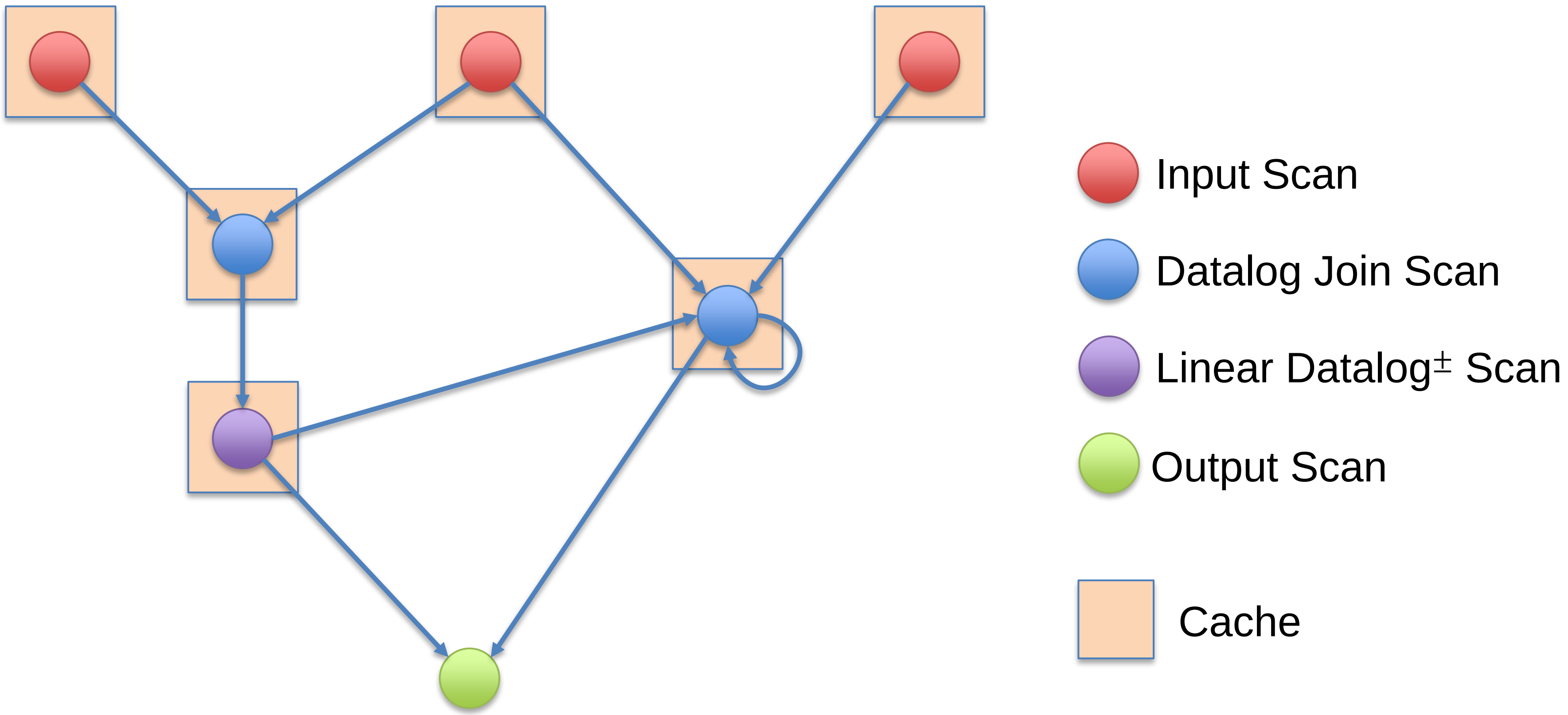
Record manager

In-Memory Stream Processing

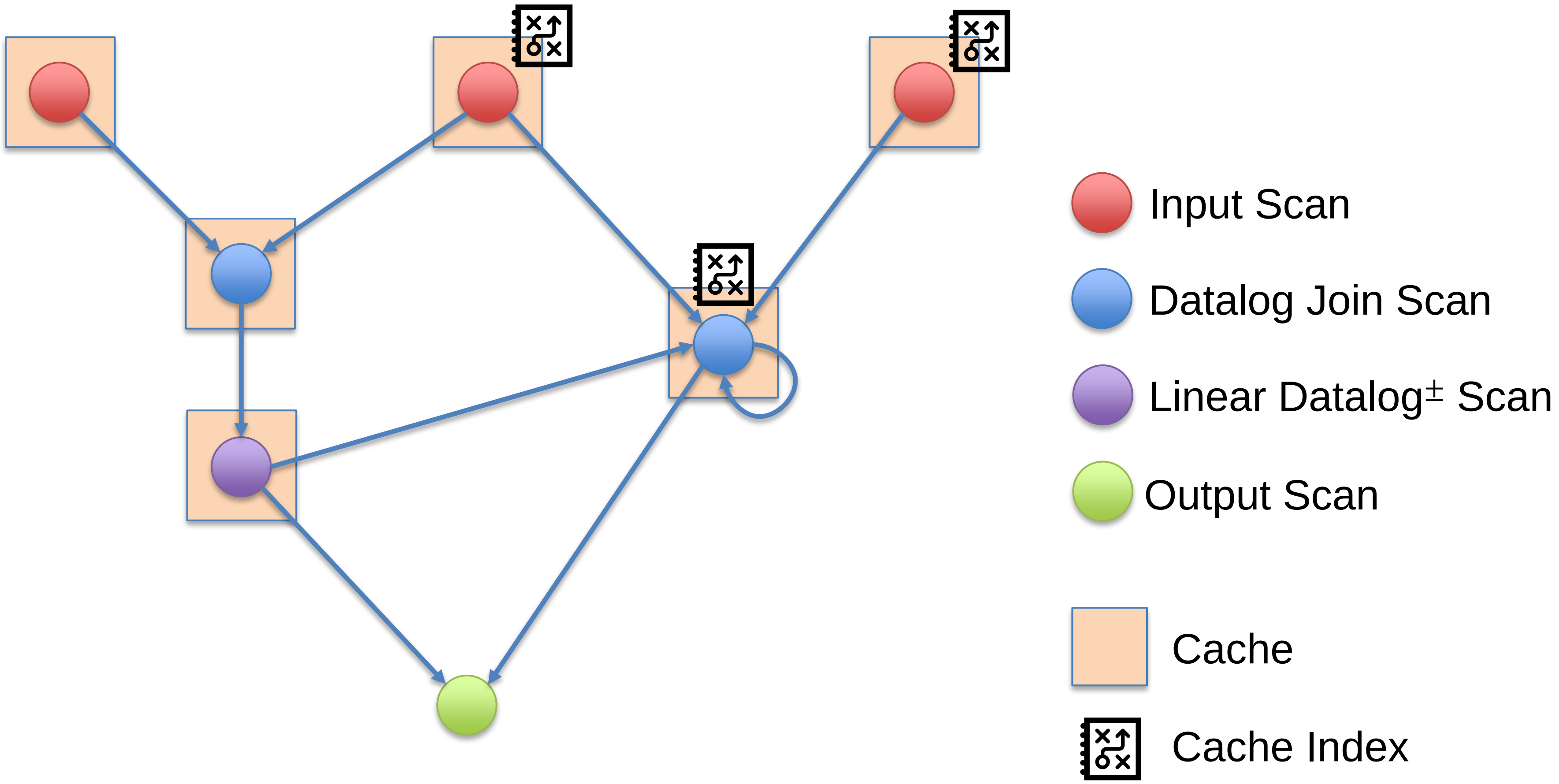


- 🔴 Input Scan
- 🔵 Datalog Join Scan
- 🟣 Linear Datalog$^{\pm}$ Scan
- 🟢 Output Scan

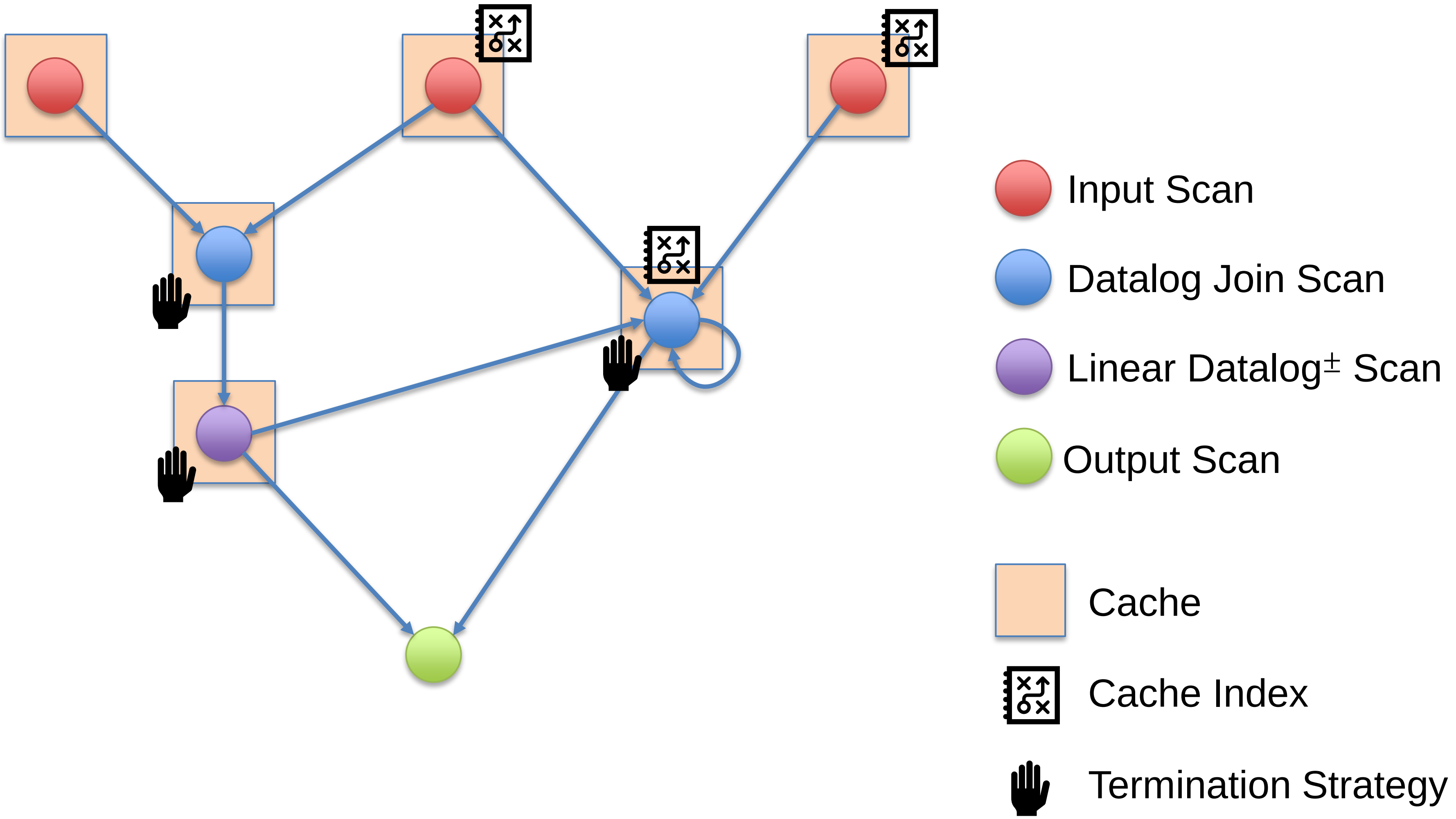
In-Memory Stream Processing

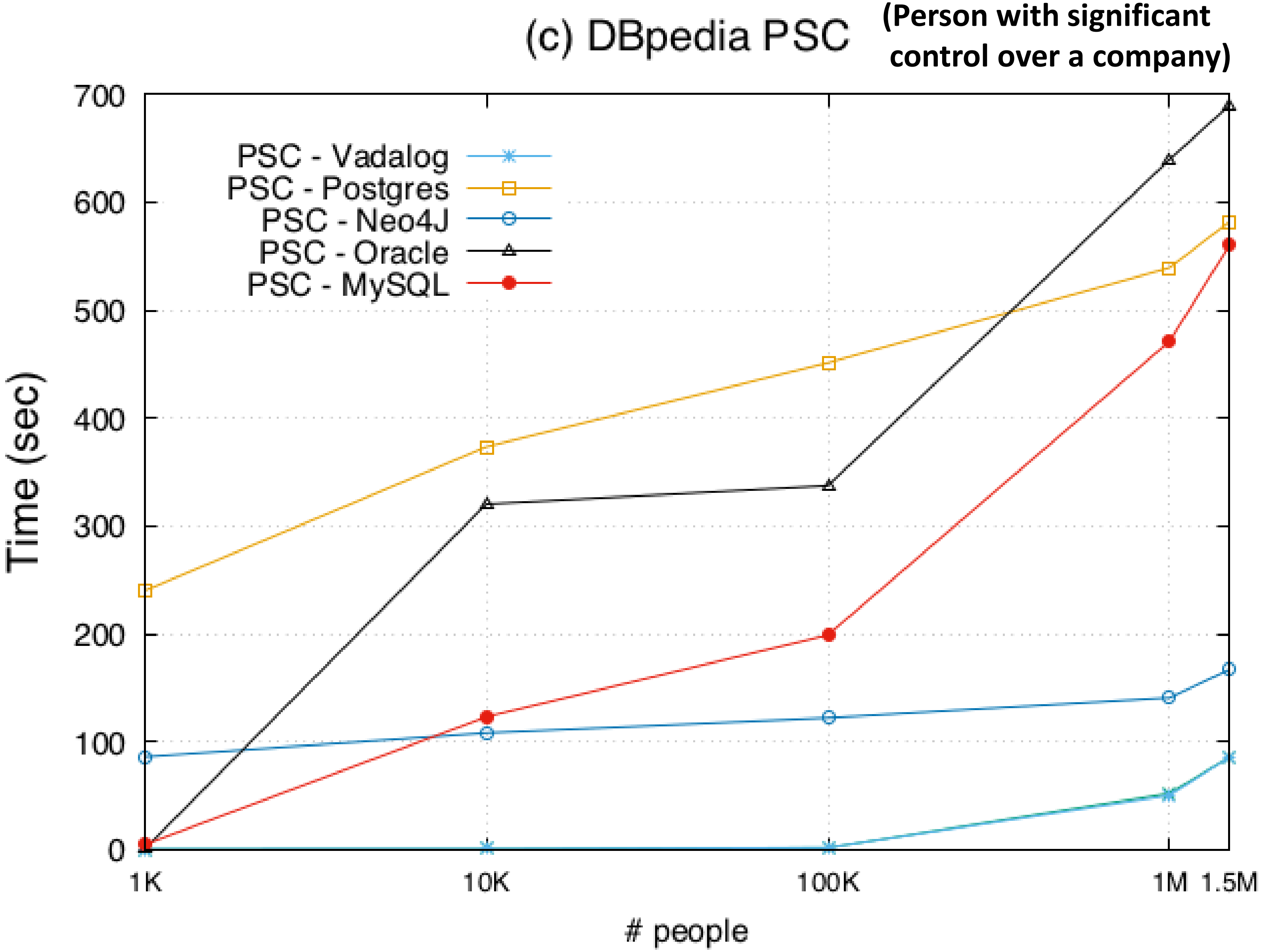


In-Memory Stream Processing

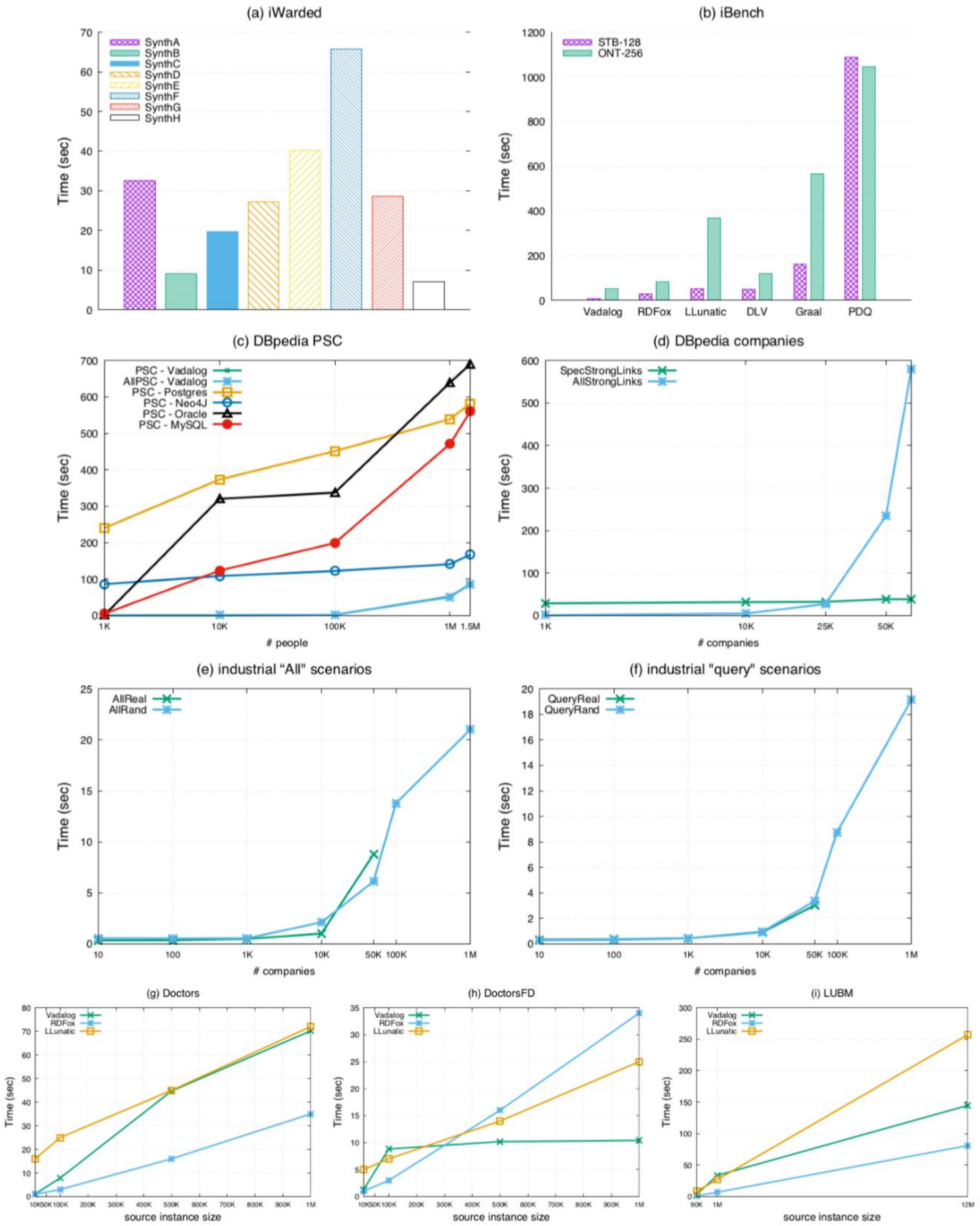


In-Memory Stream Processing





(Person with significant control over a company)



(a) iWarded

(b) iBench

(c) DBpedia PSC

(d) DBpedia companies

(e) industrial “All" scenarios

(f) industrial "query" scenarios

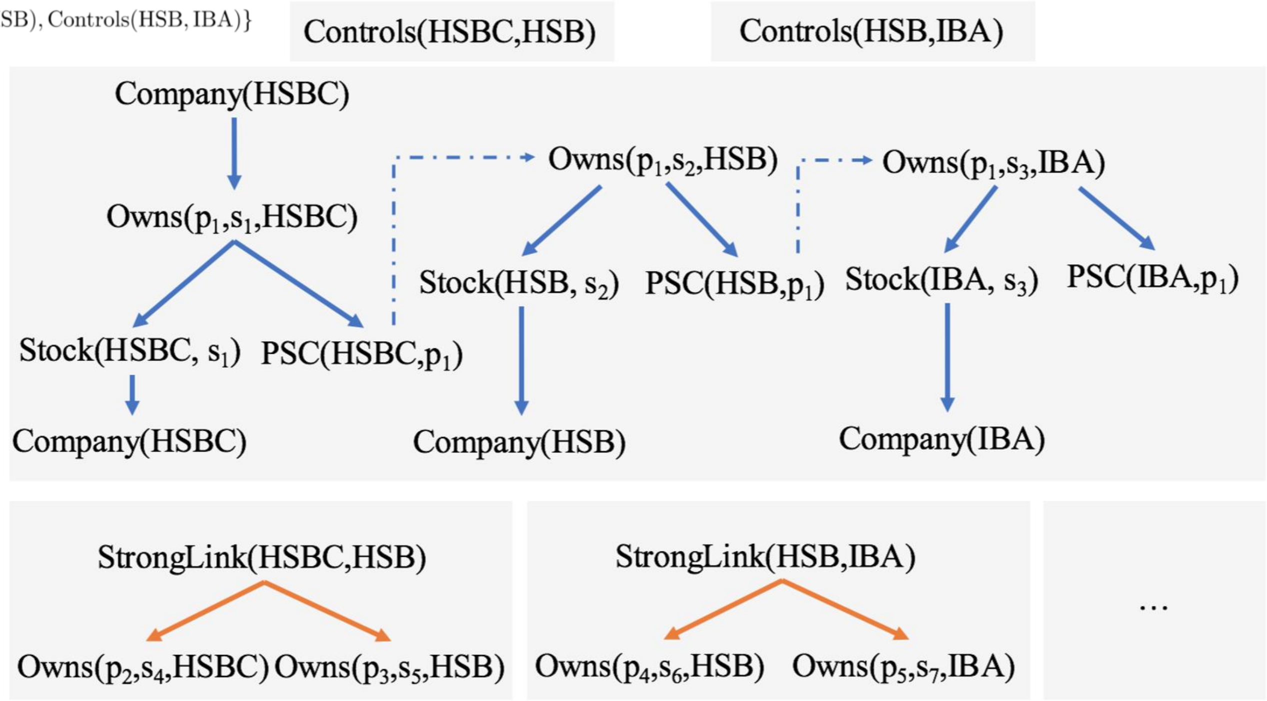
(g) Doctors

(h) DoctorsFD

(i) LUBM

$$
\begin{aligned}
1 &: \mathrm{Company}(x) \rightarrow \exists p \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, x)\\
2 &: \mathrm{Owns}(\hat{p}, \hat{s}, x) \rightarrow \mathrm{Stock}(x, \hat{s})\\
3 &: \mathrm{Owns}(\hat{p}, \hat{s}, x) \rightarrow \mathrm{PSC}(x, \hat{p})\\
4 &: \mathrm{PSC}(x, \hat{p}), \mathrm{Controls}(x, y) \rightarrow \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, y)\\
5 &: \mathrm{PSC}(x, \hat{p}), \mathrm{PSC}(y, \hat{p}) \rightarrow \underline{\mathrm{StrongLink}}(x, y)\\
6 &: \mathrm{StrongLink}(x, y) \rightarrow \exists p \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, x)\\
7 &: \mathrm{StrongLink}(x, y) \rightarrow \exists p \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, y)\\
8 &: \mathrm{Stock}(x, \hat{s}) \rightarrow \mathrm{Company}(x).
\end{aligned}
$$

$$
\begin{aligned}
D = \{&\mathrm{Company}(\mathrm{HSBC}), \mathrm{Company}(\mathrm{HSB}), \mathrm{Company}(\mathrm{IBA}),\\
&\mathrm{Controls}(\mathrm{HSBC}, \mathrm{HSB}), \mathrm{Controls}(\mathrm{HSB}, \mathrm{IBA})\}
\end{aligned}
$$



Controls(HSBC,HSB)

Controls(HSB,IBA)

Company(HSBC)

Owns(p_1,s_1,HSBC)

Stock(HSBC, s_1) PSC(HSBC,p_1)

Company(HSBC)

Owns(p_1,s_2,HSB)

Stock(HSB, s_2) PSC(HSB,p_1)

Company(HSB)

Owns(p_1,s_3,IBA)

Stock(IBA, s_3) PSC(IBA,p_1)

Company(IBA)

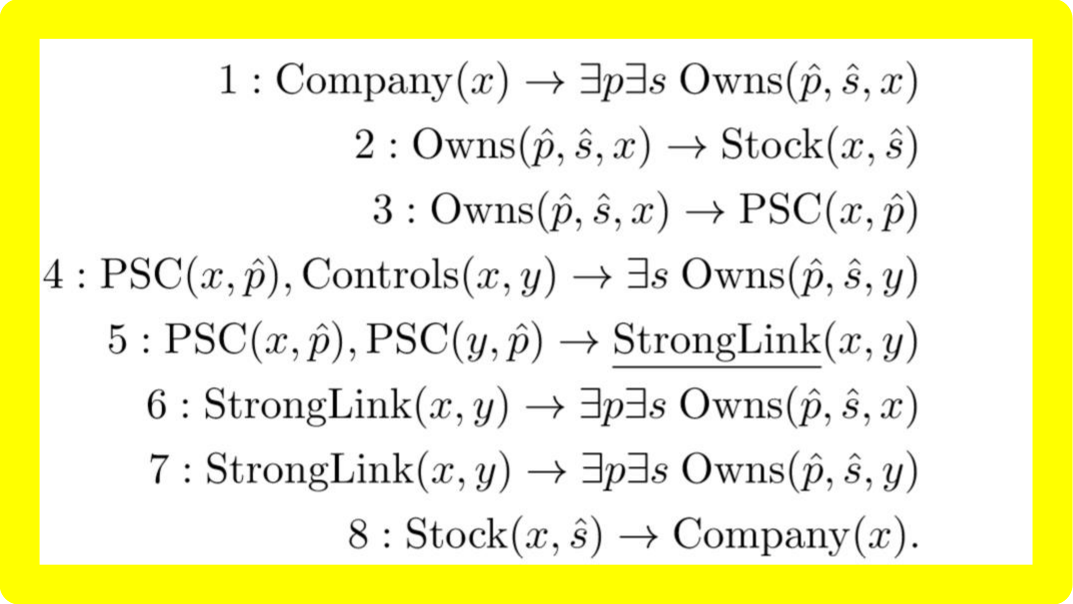
StrongLink(HSBC,HSB)

Owns(p_2,s_4,HSBC) Owns(p_3,s_5,HSB)

StrongLink(HSB,IBA)

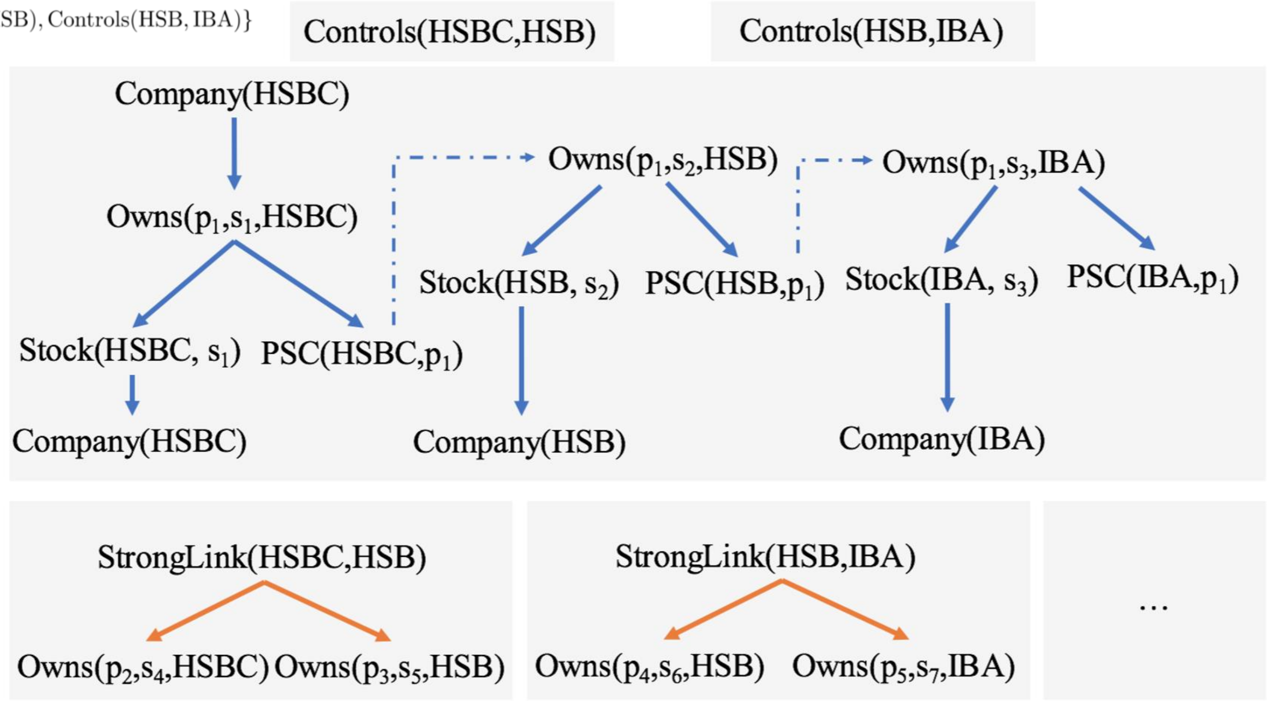
Owns(p_4,s_6,HSB) Owns(p_5,s_7,IBA)

…



$$\begin{aligned}
1 &: \mathrm{Company}(x) \rightarrow \exists p \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, x)\\
2 &: \mathrm{Owns}(\hat{p}, \hat{s}, x) \rightarrow \mathrm{Stock}(x, \hat{s})\\
3 &: \mathrm{Owns}(\hat{p}, \hat{s}, x) \rightarrow \mathrm{PSC}(x, \hat{p})\\
4 &: \mathrm{PSC}(x, \hat{p}), \mathrm{Controls}(x, y) \rightarrow \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, y)\\
5 &: \mathrm{PSC}(x, \hat{p}), \mathrm{PSC}(y, \hat{p}) \rightarrow \underline{\mathrm{StrongLink}}(x, y)\\
6 &: \mathrm{StrongLink}(x, y) \rightarrow \exists p \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, x)\\
7 &: \mathrm{StrongLink}(x, y) \rightarrow \exists p \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, y)\\
8 &: \mathrm{Stock}(x, \hat{s}) \rightarrow \mathrm{Company}(x).
\end{aligned}$$

$$\begin{aligned}
D = \{&\mathrm{Company}(\mathrm{HSBC}), \mathrm{Company}(\mathrm{HSB}), \mathrm{Company}(\mathrm{IBA}),\\
&\mathrm{Controls}(\mathrm{HSBC}, \mathrm{HSB}), \mathrm{Controls}(\mathrm{HSB}, \mathrm{IBA})\}
\end{aligned}$$



Controls(HSBC,HSB)

Controls(HSB,IBA)

Company(HSBC)

Owns(p_1,s_1,HSBC)

Stock(HSBC, s_1)

PSC(HSBC,p_1)

Company(HSBC)

Owns(p_1,s_2,HSB)

Stock(HSB, s_2)

PSC(HSB,p_1)

Company(HSB)

Owns(p_1,s_3,IBA)

Stock(IBA, s_3)

PSC(IBA,p_1)

Company(IBA)

StrongLink(HSBC,HSB)

Owns(p_2,s_4,HSBC)

Owns(p_3,s_5,HSB)

StrongLink(HSB,IBA)

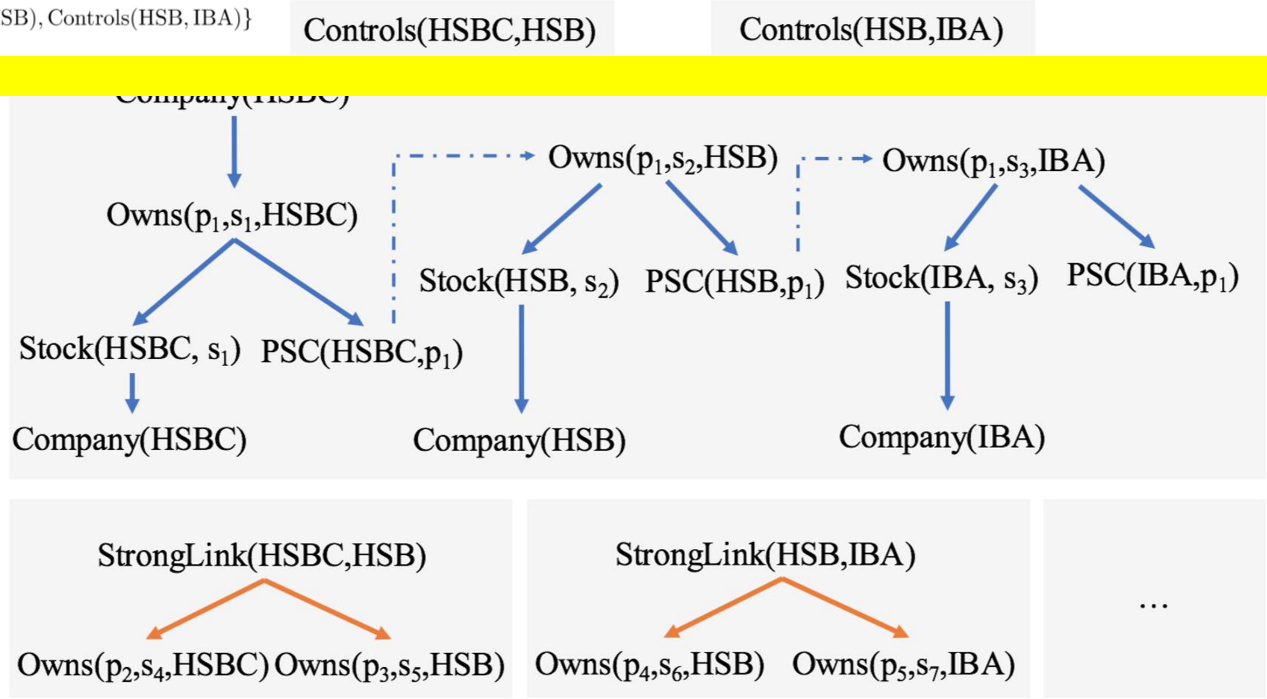
Owns(p_4,s_6,HSB)

Owns(p_5,s_7,IBA)

…

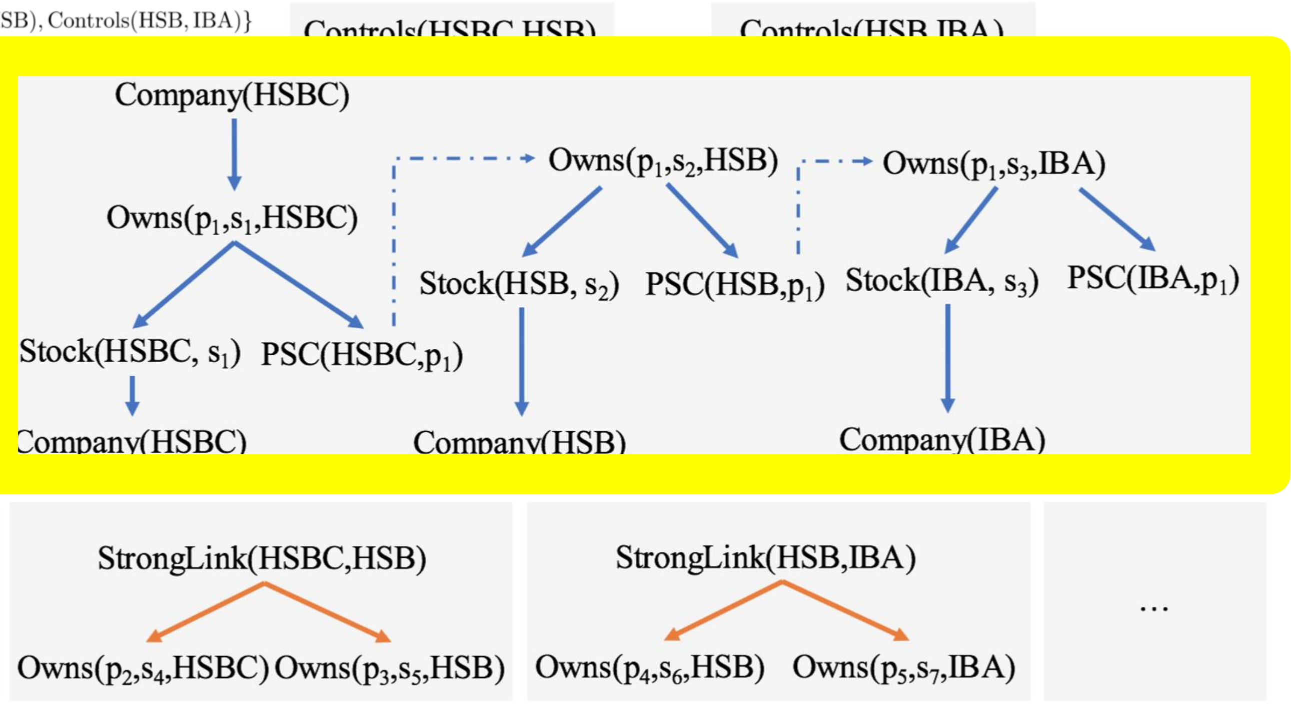
$$
\begin{aligned}
1 &: \mathrm{Company}(x) \rightarrow \exists p \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, x)\\
2 &: \mathrm{Owns}(\hat{p}, \hat{s}, x) \rightarrow \mathrm{Stock}(x, \hat{s})\\
3 &: \mathrm{Owns}(\hat{p}, \hat{s}, x) \rightarrow \mathrm{PSC}(x, \hat{p})\\
4 &: \mathrm{PSC}(x, \hat{p}), \mathrm{Controls}(x, y) \rightarrow \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, y)\\
5 &: \mathrm{PSC}(x, \hat{p}), \mathrm{PSC}(y, \hat{p}) \rightarrow \underline{\mathrm{StrongLink}}(x, y)\\
6 &: \mathrm{StrongLink}(x, y) \rightarrow \exists p \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, x)\\
7 &: \mathrm{StrongLink}(x, y) \rightarrow \exists p \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, y)\\
8 &: \mathrm{Stock}(x, \hat{s}) \rightarrow \mathrm{Company}(x).
\end{aligned}
$$

$= \{\mathrm{Company}(\mathrm{HSBC}), \mathrm{Company}(\mathrm{HSB}), \mathrm{Company}(\mathrm{IBA}),$
$\mathrm{Controls}(\mathrm{HSBC}, \mathrm{HSB}), \mathrm{Controls}(\mathrm{HSB}, \mathrm{IBA})\}$



$$\begin{aligned}
1 &: \mathrm{Company}(x) \rightarrow \exists p \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, x)\\
2 &: \mathrm{Owns}(\hat{p}, \hat{s}, x) \rightarrow \mathrm{Stock}(x, \hat{s})\\
3 &: \mathrm{Owns}(\hat{p}, \hat{s}, x) \rightarrow \mathrm{PSC}(x, \hat{p})\\
4 &: \mathrm{PSC}(x, \hat{p}), \mathrm{Controls}(x, y) \rightarrow \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, y)\\
5 &: \mathrm{PSC}(x, \hat{p}), \mathrm{PSC}(y, \hat{p}) \rightarrow \underline{\mathrm{StrongLink}}(x, y)\\
6 &: \mathrm{StrongLink}(x, y) \rightarrow \exists p \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, x)\\
7 &: \mathrm{StrongLink}(x, y) \rightarrow \exists p \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, y)\\
8 &: \mathrm{Stock}(x, \hat{s}) \rightarrow \mathrm{Company}(x).
\end{aligned}$$

$$\begin{aligned}
D = \{&\mathrm{Company}(\mathrm{HSBC}), \mathrm{Company}(\mathrm{HSB}), \mathrm{Company}(\mathrm{IBA}),\\
&\mathrm{Controls}(\mathrm{HSBC}, \mathrm{HSB}), \mathrm{Controls}(\mathrm{HSB}, \mathrm{IBA})\}
\end{aligned}$$



Controls(HSBC,HSB)

Controls(HSB,IBA)

Company(HSBC)

Owns(p_1,s_1,HSBC)

Stock(HSBC, s_1)

PSC(HSBC,p_1)

Company(HSBC)

Owns(p_1,s_2,HSB)

Stock(HSB, s_2)

PSC(HSB,p_1)

Company(HSB)

Owns(p_1,s_3,IBA)

Stock(IBA, s_3)

PSC(IBA,p_1)

Company(IBA)

StrongLink(HSBC,HSB)

Owns(p_2,s_4,HSBC)

Owns(p_3,s_5,HSB)

StrongLink(HSB,IBA)

Owns(p_4,s_6,HSB)

Owns(p_5,s_7,IBA)

…

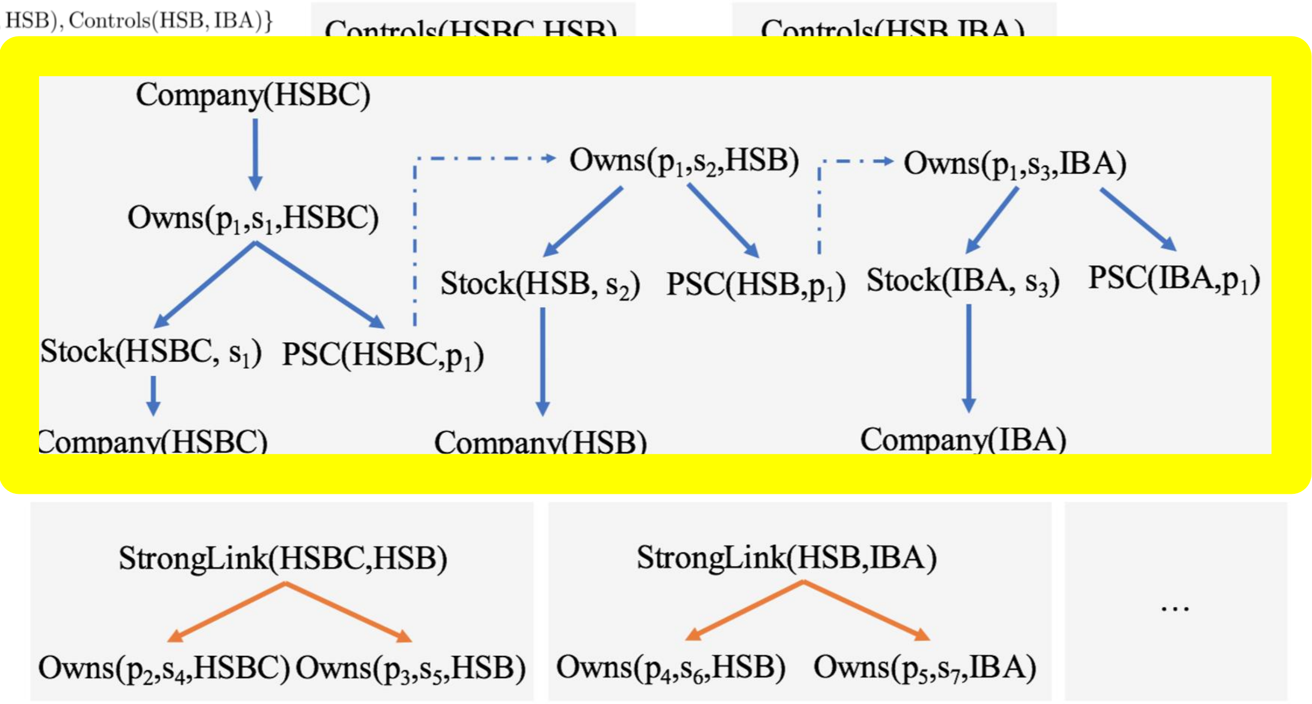
Dangerous

$$\mathrm{Emp}(x) \rightarrow \exists \boldsymbol{z}\, \mathrm{Mgr}(x, \boldsymbol{z}) \qquad \underbrace{\mathrm{Mgr}(x, \boldsymbol{y})}_{\textbf{Ward}} \land \mathrm{Pers}(x) \rightarrow \mathrm{Emp}(\boldsymbol{y})$$

1. all the “**dangerous**” variables should coexist in a single body-atom α, called the **ward**

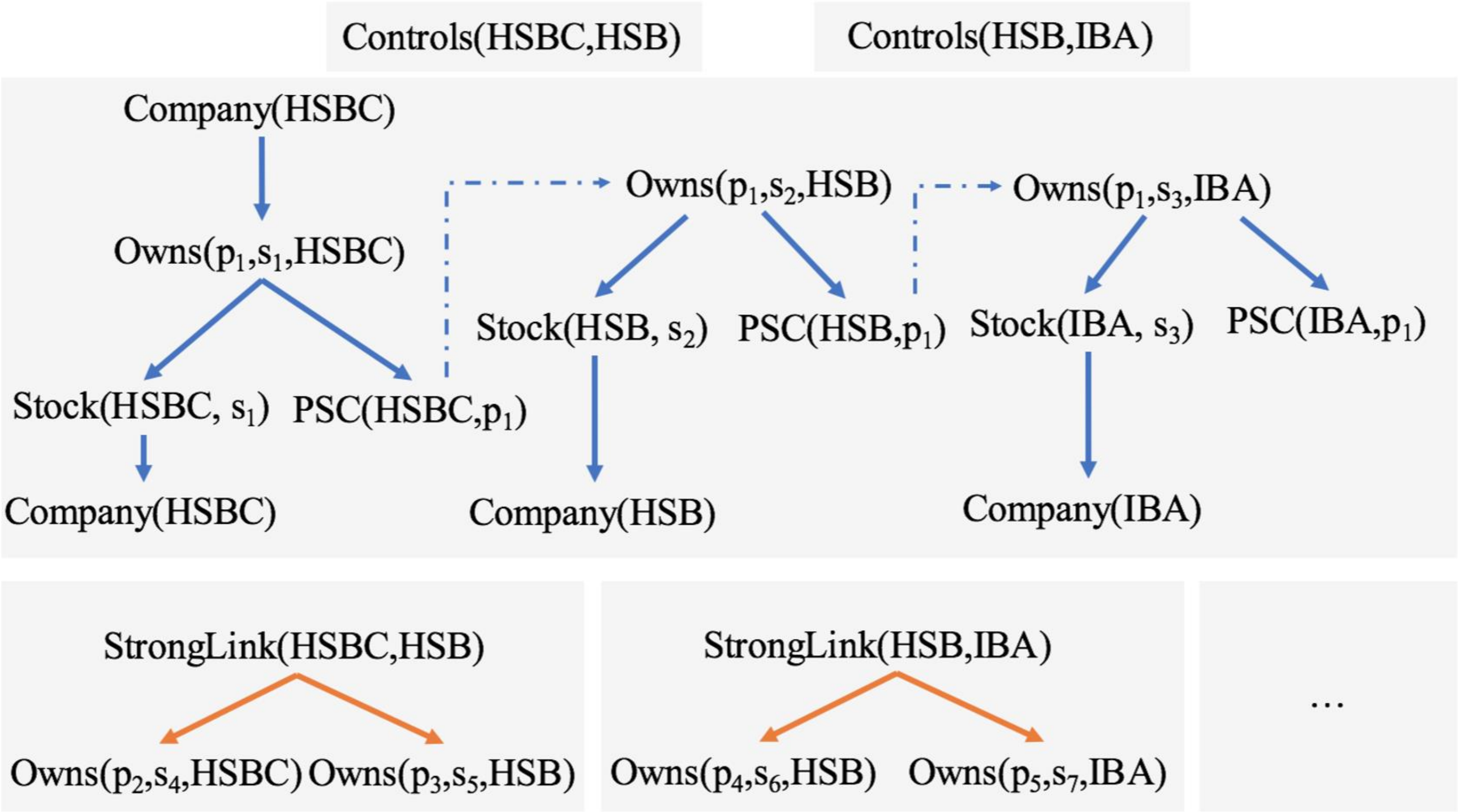
$$\begin{aligned}
1 &: \mathrm{Company}(x) \rightarrow \exists p \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, x)\\
2 &: \mathrm{Owns}(\hat{p}, \hat{s}, x) \rightarrow \mathrm{Stock}(x, \hat{s})\\
3 &: \mathrm{Owns}(\hat{p}, \hat{s}, x) \rightarrow \mathrm{PSC}(x, \hat{p})\\
4 &: \mathrm{PSC}(x, \hat{p}), \mathrm{Controls}(x, y) \rightarrow \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, y)\\
5 &: \mathrm{PSC}(x, \hat{p}), \mathrm{PSC}(y, \hat{p}) \rightarrow \underline{\mathrm{StrongLink}}(x, y)\\
6 &: \mathrm{StrongLink}(x, y) \rightarrow \exists p \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, x)\\
7 &: \mathrm{StrongLink}(x, y) \rightarrow \exists p \exists s\ \mathrm{Owns}(\hat{p}, \hat{s}, y)\\
8 &: \mathrm{Stock}(x, \hat{s}) \rightarrow \mathrm{Company}(x).
\end{aligned}$$

$$D = \{\mathrm{Company}(\mathrm{HSBC}), \mathrm{Company}(\mathrm{HSB}), \mathrm{Company}(\mathrm{IBA}),$$
$$\mathrm{Controls}(\mathrm{HSBC}, \mathrm{HSB}), \mathrm{Controls}(\mathrm{HSB}, \mathrm{IBA})\}$$



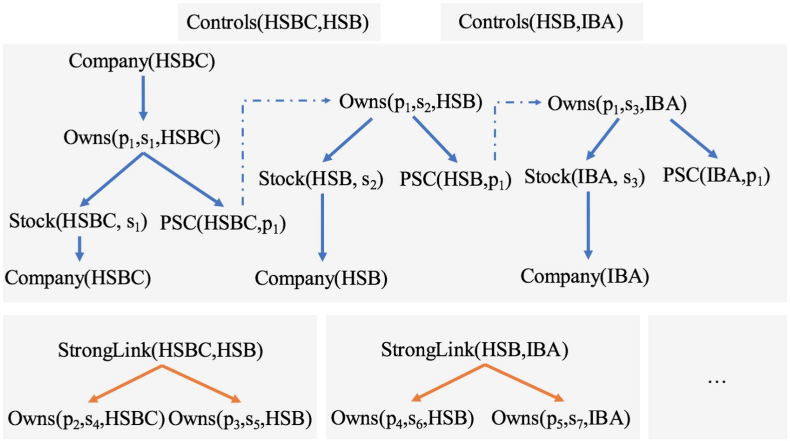
Theorem

Let a and b be two facts in a warded forest. If they are isomorphic, then $subtree(a)$ is isomorphic to $subtree(b)$.



Theorem

Let a and b be two facts in a warded forest. If they are isomorphic, then $subtree(a)$ is isomorphic to $subtree(b)$.

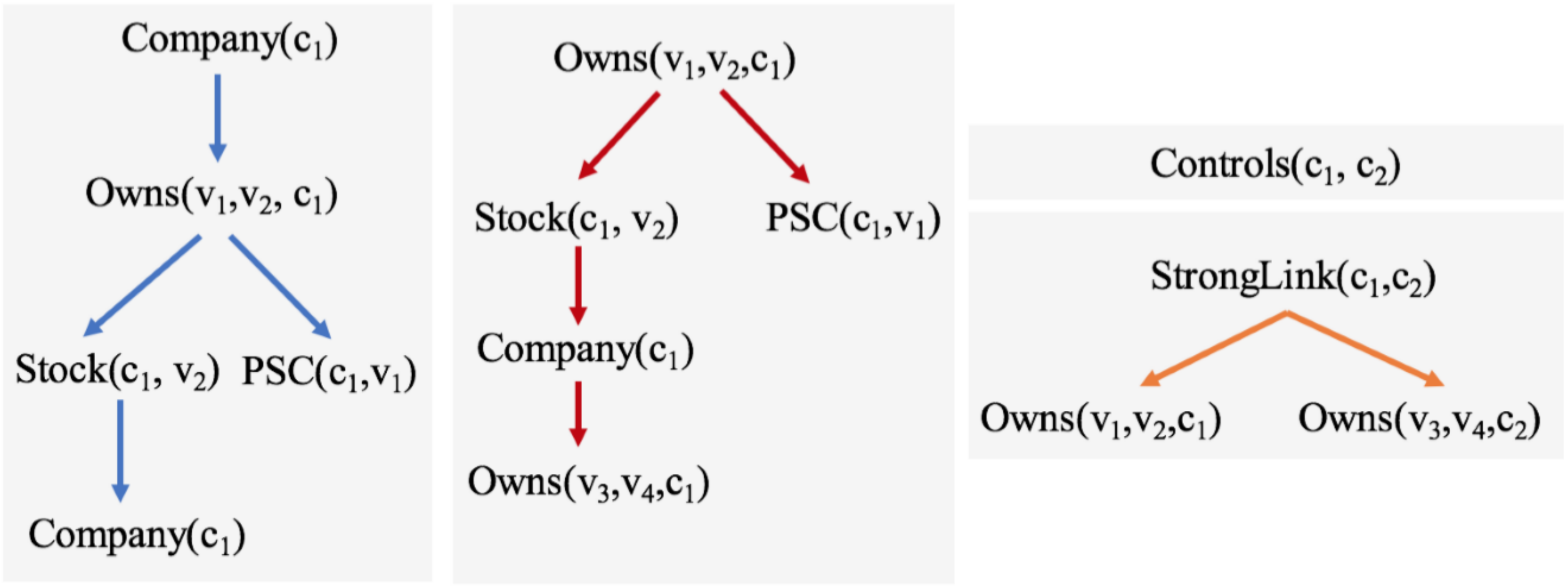


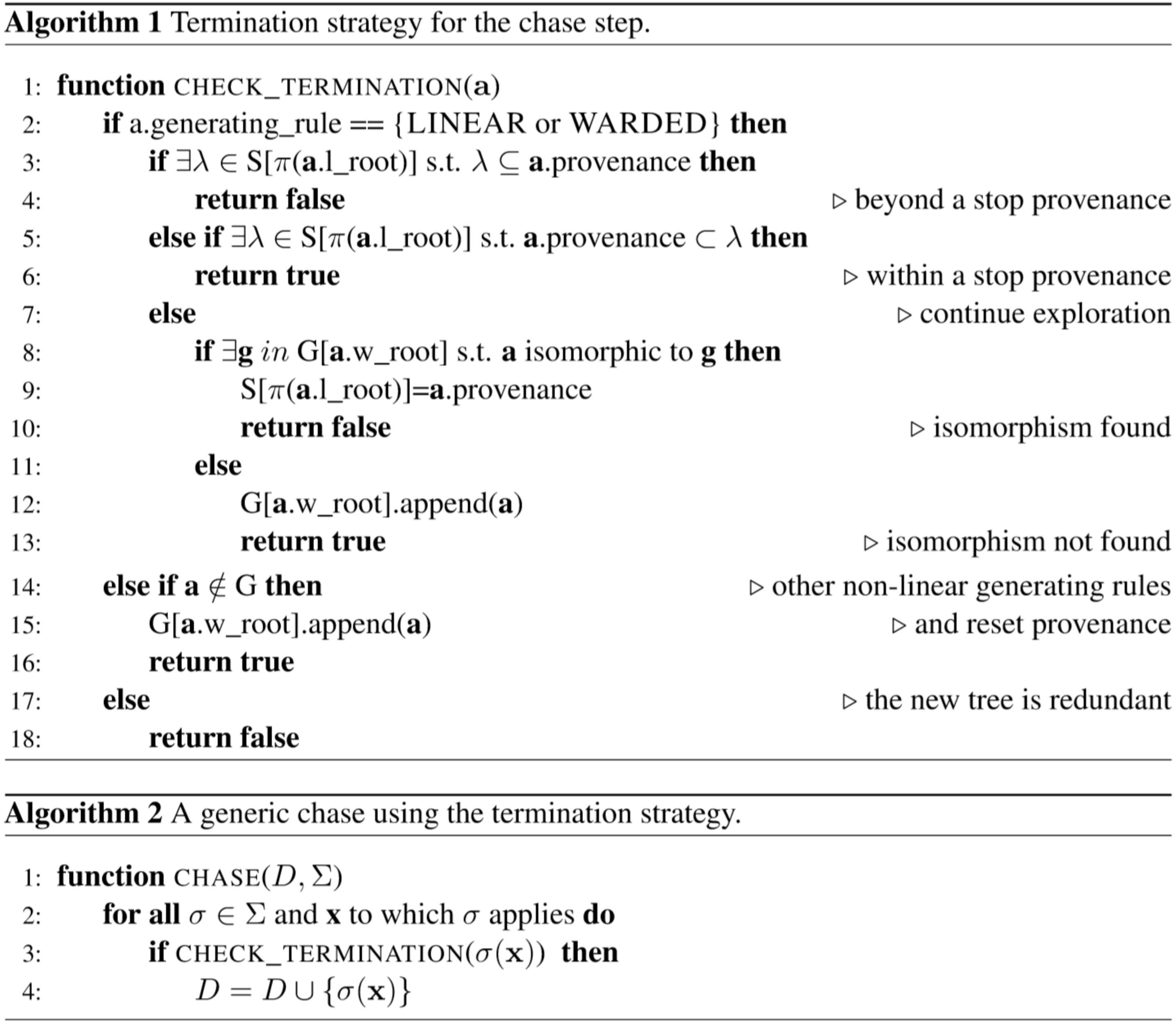
Theorem

Let a and b be two facts in the chase graph of a set of harmless warded rules. If a and b are isomorphic, then $subgraph(a)$ is isomorphic to $subgraph(b)$.

Proposition

Let a and b be two facts in a linear forest. If they are pattern-isomorphic, then $subtree(a)$ is pattern-isomorphic $to\ subtree(b)$.





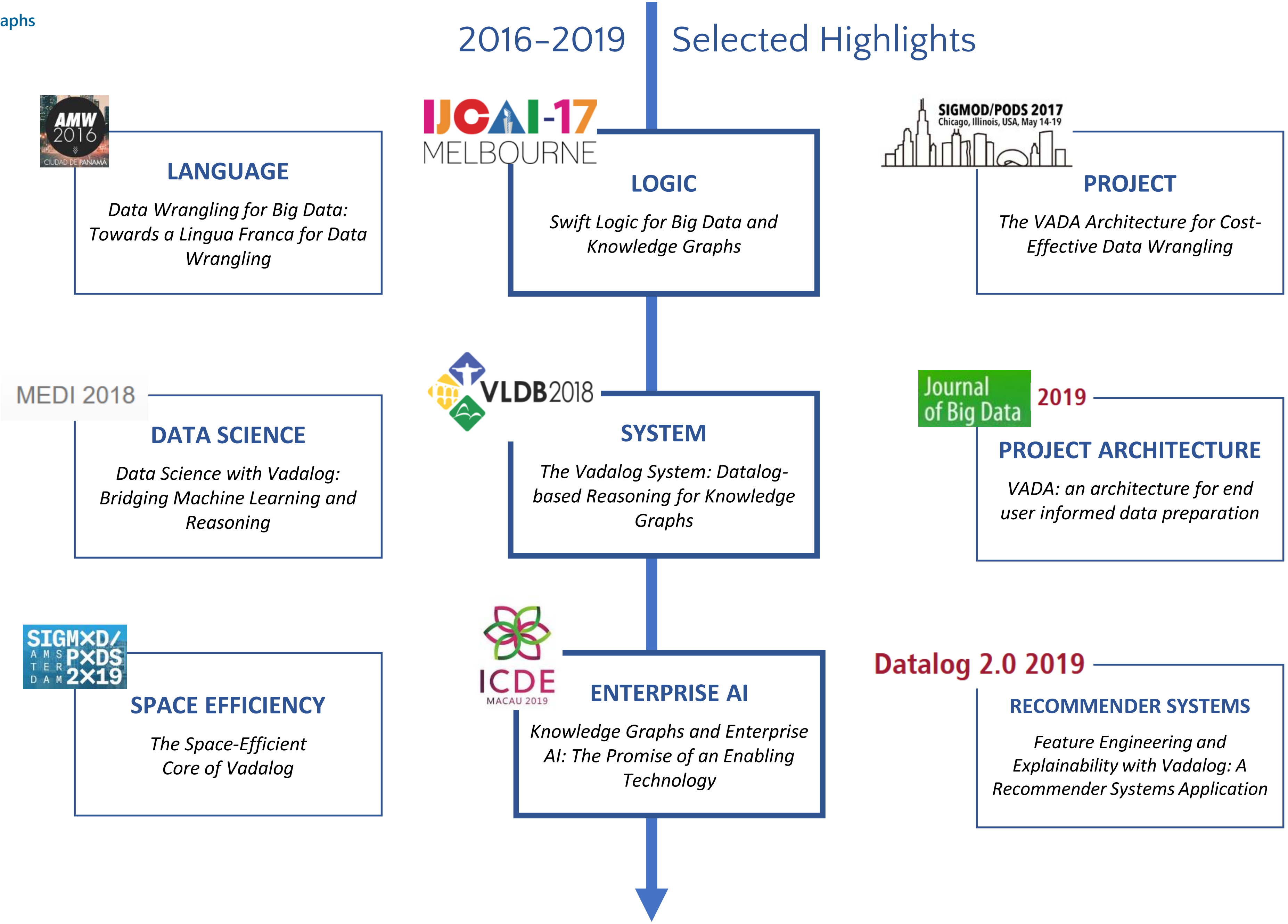
Algorithm 1 Termination strategy for the chase step.

```
function CHECK_TERMINATION($\mathbf{a}$)
    if a.generating_rule == {LINEAR or WARDED} then
        if $\exists\lambda \in$ S[$\pi$($\mathbf{a}$.l_root)] s.t. $\lambda \subseteq$ $\mathbf{a}$.provenance then
            return false                                        ▷ beyond a stop provenance
        else if $\exists\lambda \in$ S[$\pi$($\mathbf{a}$.l_root)] s.t. $\mathbf{a}$.provenance $\subset \lambda$ then
            return true                                         ▷ within a stop provenance
        else                                                    ▷ continue exploration
            if $\exists$g $in$ G[$\mathbf{a}$.w_root] s.t. $\mathbf{a}$ isomorphic to g then
                S[$\pi$($\mathbf{a}$.l_root)]=$\mathbf{a}$.provenance
                return false                                    ▷ isomorphism found
            else
                G[$\mathbf{a}$.w_root].append($\mathbf{a}$)
                return true                                     ▷ isomorphism not found
    else if $\mathbf{a} \notin$ G then                          ▷ other non-linear generating rules
        G[$\mathbf{a}$.w_root].append($\mathbf{a}$)             ▷ and reset provenance
        return true
    else                                                        ▷ the new tree is redundant
        return false
```

Algorithm 2 A generic chase using the termination strategy.

```
function CHASE($D, \Sigma$)
    for all $\sigma \in \Sigma$ and $\mathbf{x}$ to which $\sigma$ applies do
        if CHECK_TERMINATION($\sigma(\mathbf{x})$) then
            $D = D \cup \{\sigma(\mathbf{x})\}$
```

2016–2019 | Selected Highlights



AMW 2016 CIUDAD DE PANAMÁ

LANGUAGE

Data Wrangling for Big Data: Towards a Lingua Franca for Data Wrangling

IJCAI-17 MELBOURNE

LOGIC

Swift Logic for Big Data and Knowledge Graphs

SIGMOD/PODS 2017 Chicago, Illinois, USA, May 14-19

PROJECT

The VADA Architecture for Cost-Effective Data Wrangling

MEDI 2018

DATA SCIENCE

Data Science with Vadalog: Bridging Machine Learning and Reasoning

VLDB2018

SYSTEM

The Vadalog System: Datalog-based Reasoning for Knowledge Graphs

Journal of Big Data 2019

PROJECT ARCHITECTURE

VADA: an architecture for end user informed data preparation

SIGMOD/PODS 2019 AMSTERDAM

SPACE EFFICIENCY

The Space-Efficient Core of Vadalog

ICDE MACAU 2019

ENTERPRISE AI

Knowledge Graphs and Enterprise AI: The Promise of an Enabling Technology

Datalog 2.0 2019

RECOMMENDER SYSTEMS

Feature Engineering and Explainability with Vadalog: A Recommender Systems Application

2020 Selected Highlights

EMERGENCY RESPONSE
(not peer-reviewed)

COVID-19: LOCKDOWN

COVID-19 and Company Knowledge Graphs: Assessing Golden Powers and Economic Impact of Selective Lockdown via AI Reasoning

EDBT ICDT COPENHAGEN 2020

ENTERPRISE AI IN PRACTICE

Weaving Enterprise Knowledge Graphs: The Case of Company Ownership Graphs

Declarative AI 2020

COVID-19: TAKEOVERS

Reasoning on Company Takeovers during the COVID-19 Crisis with Knowledge Graphs

Declarative AI 2020

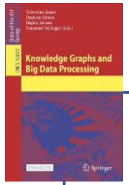
PROBABILISTIC

Reasoning Under Uncertainty in Knowledge Graphs

Declarative AI 2020

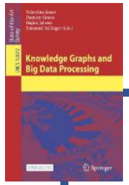
MONEY LAUNDERING

Rule-based Anti-Money Laundering in Financial Intelligence Units: Experience and Vision



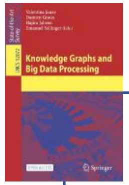
KNOWLEDGE GRAPHS

Knowledge Graphs: The Layered Perspective



BOOK

Knowledge Graphs and Big Data Processing



KG EMBEDDINGS

Reasoning in Knowledge Graphs: An Embeddings Spotlight

2021 Selected Highlights

EDBT ICDT NICOSIA 2021

CONFIDENTIALITY

Financial Data Exchange with Statistical Confidentiality: A Reasoning-based Approach

ICDE 2021 VIRTUAL CONFERENCE GREECE

COMPANY CONTROL

Distributed Company Control in Company Shareholding Graphs

Declarative AI 2021

TEMPORAL

Monotonic Aggregation for Temporal Datalog

Declarative AI 2021

HARMFUL JOINS

Eliminating Harmful Joins in Warded Datalog+/-.

Declarative AI 2021

INDUSTRIAL BLOCKCHAIN

Rule-based Blockchain Knowledge Graphs: Declarative AI for Solving Industrial Blockchain Challenges

ECAI 2020 Santiago de Compostela

JOINS

Traversing Knowledge Graphs with Good Old (and New) Joins

ECAI 2020 Santiago de Compostela

HYBRID AI

Augmenting Logic-based Knowledge Graphs: The Case of Company Graphs

ECAI 2020 Santiago de Compostela

BLOCKCHAIN VISION

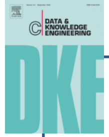
Blockchains as Knowledge Graphs - Blockchains for Knowledge Graphs

2022 Selected Highlights

FGCS

DATA SCIENCE

Data science with Vadalog: Knowledge Graphs with machine learning and reasoning in practice



DATA & KNOWLEDGE ENGINEERING

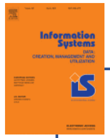
DKE

CS AND ECONOMY

Reasoning on company takeovers: From tactic to strategy

EcoFinKG

DATALOG 2.0



Information Systems

IS

AI ARCHITECTURE

Vadalog: A modern architecture for automated reasoning with large knowledge graphs

EDBT ICDT EDINBURGH 2022

KG DESIGN

Model-Independent Design of Knowledge Graphs - Lessons Learnt From Complex Financial Graphs

ICDE 2022

38th IEEE International Conference on Data Engineering

TEMPORAL

iTemporal: An Extensible Generator of Temporal Benchmarks

KR

SHY AND WARDED

On the Relationship between Shy and Warded Datalog+/-

EDBT ICDT EDINBURGH 2022

RULE LEARNING

Rule Learning over Knowledge Graphs with Genetic Logic Programming

PVLDB Very Large Data Bases

EGDs

Exploiting the Power of Equality-generating Dependencies in Ontological Reasoning

DATA SCIENCE

Data Science with Vadalog: Bridging Machine Learning and Reasoning

HYBRID AI

Augmenting Logic-based Knowledge Graphs: The Case of Company Graphs

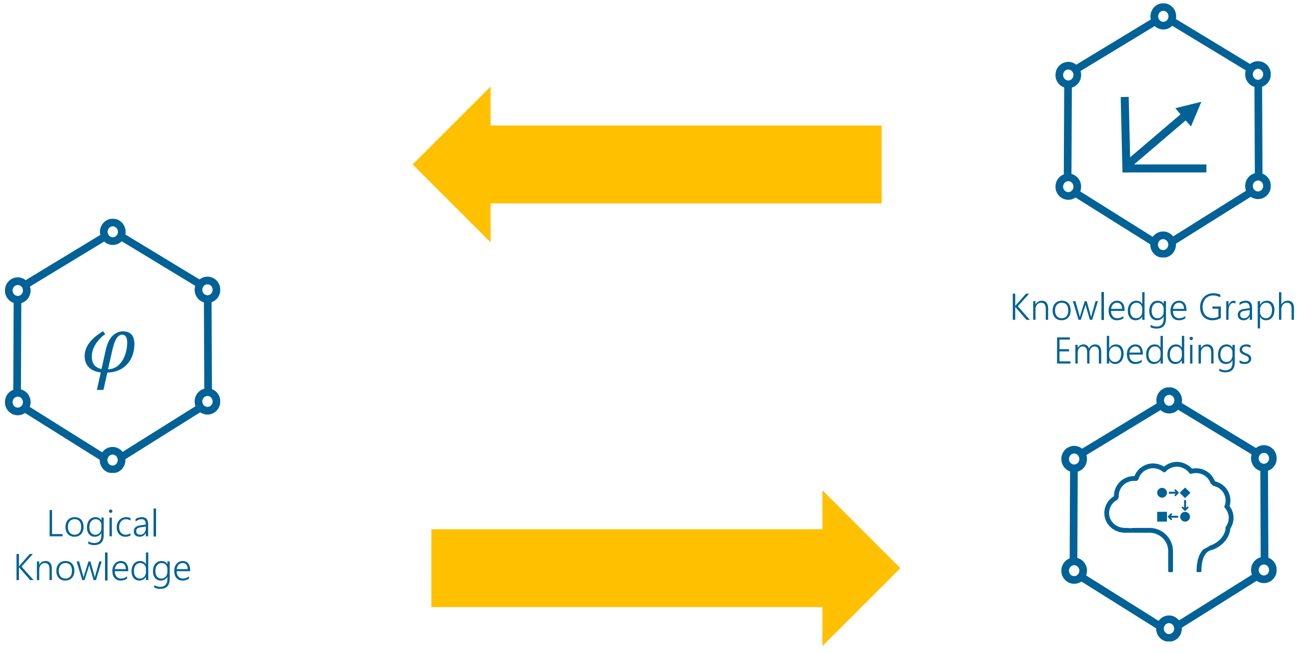
38th IEEE International Conference on Data Engineering

RULE LEARNING

Rule Learning over Knowledge Graphs with Genetic Logic Programming

PATTERN AWARENESS

Pattern-Aware and Noise-Resilient Embedding Models



Knowledge Graph Embeddings

Logical Knowledge

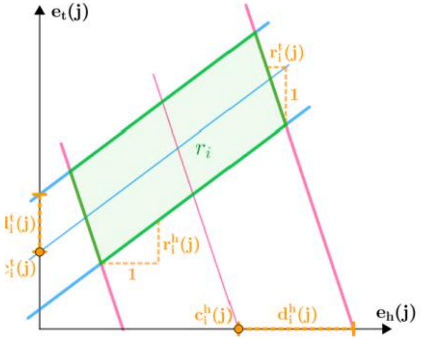
Graph Neural Networks

ICLR 2023

ExpressivE

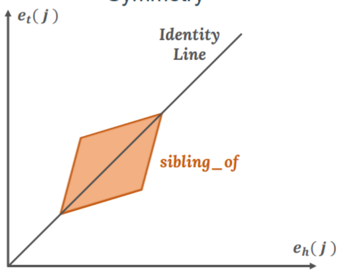
A Spatio-Functional Knowledge Graph Embedding

$$(e_h - c_i^h - r_i^t \odot e_t)^{|.|} \leq d_i^h$$
$$(e_t - c_i^t - r_i^h \odot e_h)^{|.|} \leq d_i^t$$

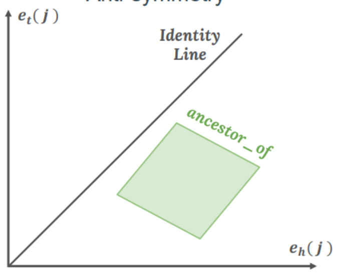


Inference Pattern	ExpressivE	BoxE	RotatE	TransE	DistMult	ComplEx
Symmetry: $r_1(X,Y) \Rightarrow r_1(Y,X)$	✓	✓	✓	✗	✓	✓
Anti-symmetry: $r_1(X,Y) \Rightarrow \neg r_1(Y,X)$	✓	✓	✓	✓	✗	✓
Inversion: $r_1(X,Y) \Leftrightarrow r_2(Y,X)$	✓	✓	✓	✓	✗	✓
Comp. def.: $r_1(X,Y) \wedge r_2(Y,Z) \Leftrightarrow r_3(X,Z)$	✓	✗	✓	✓	✗	✗
Gen. comp.: $r_1(X,Y) \wedge r_2(Y,Z) \Rightarrow r_3(X,Z)$	✓	✗	✗	✗	✗	✗
Hierarchy: $r_1(X,Y) \Rightarrow r_2(X,Y)$	✓	✓	✗	✗	✓	✓
Intersection: $r_1(X,Y) \wedge r_2(X,Y) \Rightarrow r_3(X,Y)$	✓	✓	✓	✓	✗	✗
Mutual exclusion: $r_1(X,Y) \wedge r_2(X,Y) \Rightarrow \perp$	✓	✓	✓	✓	✓	✓

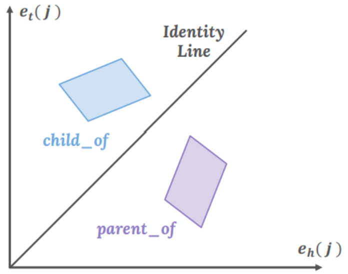
Symmetry



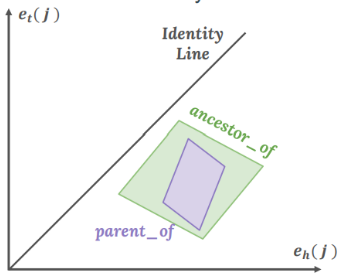
Anti-symmetry



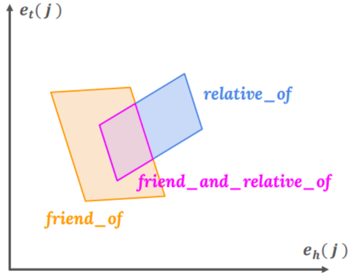
Inversion



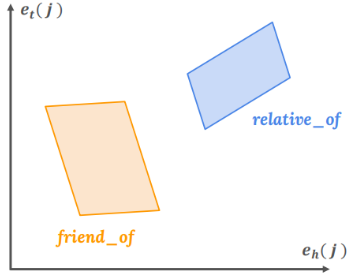
Hierarchy



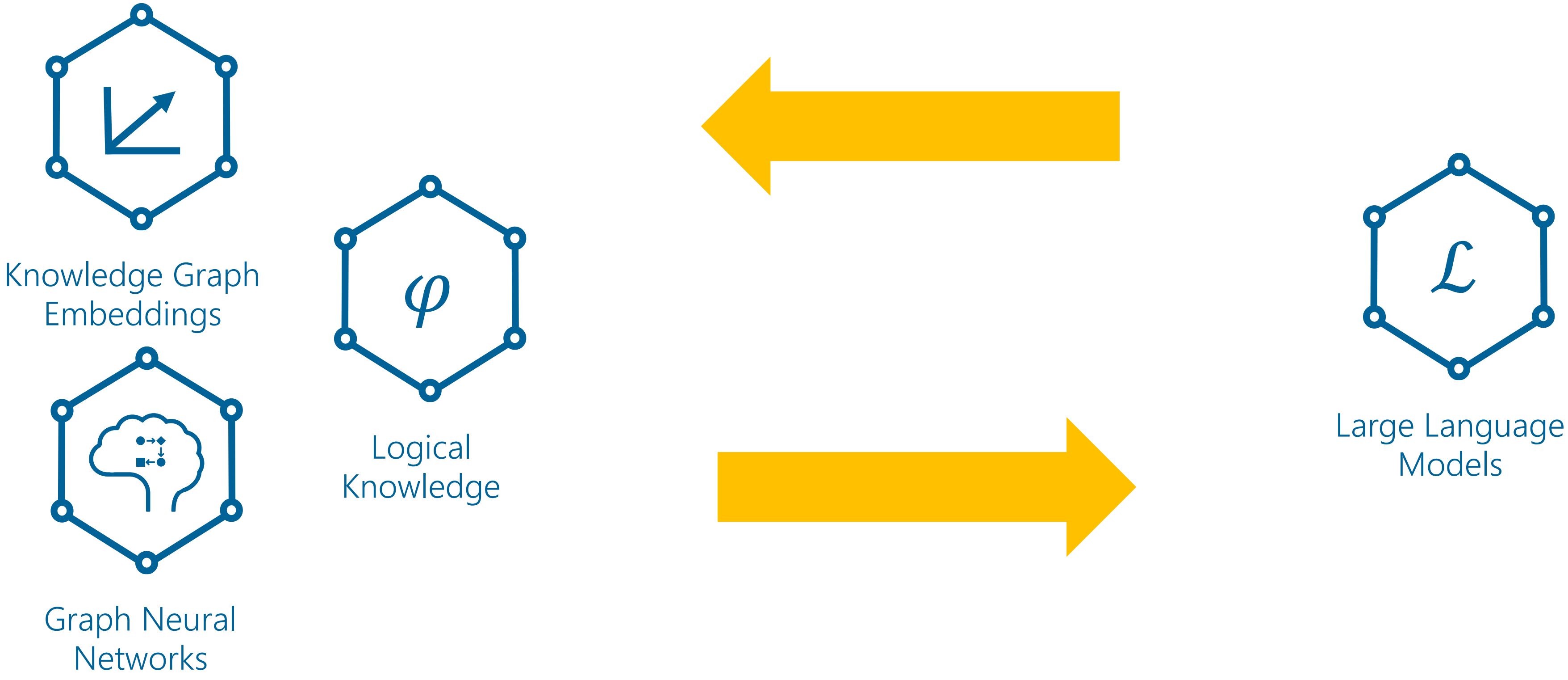
Intersection

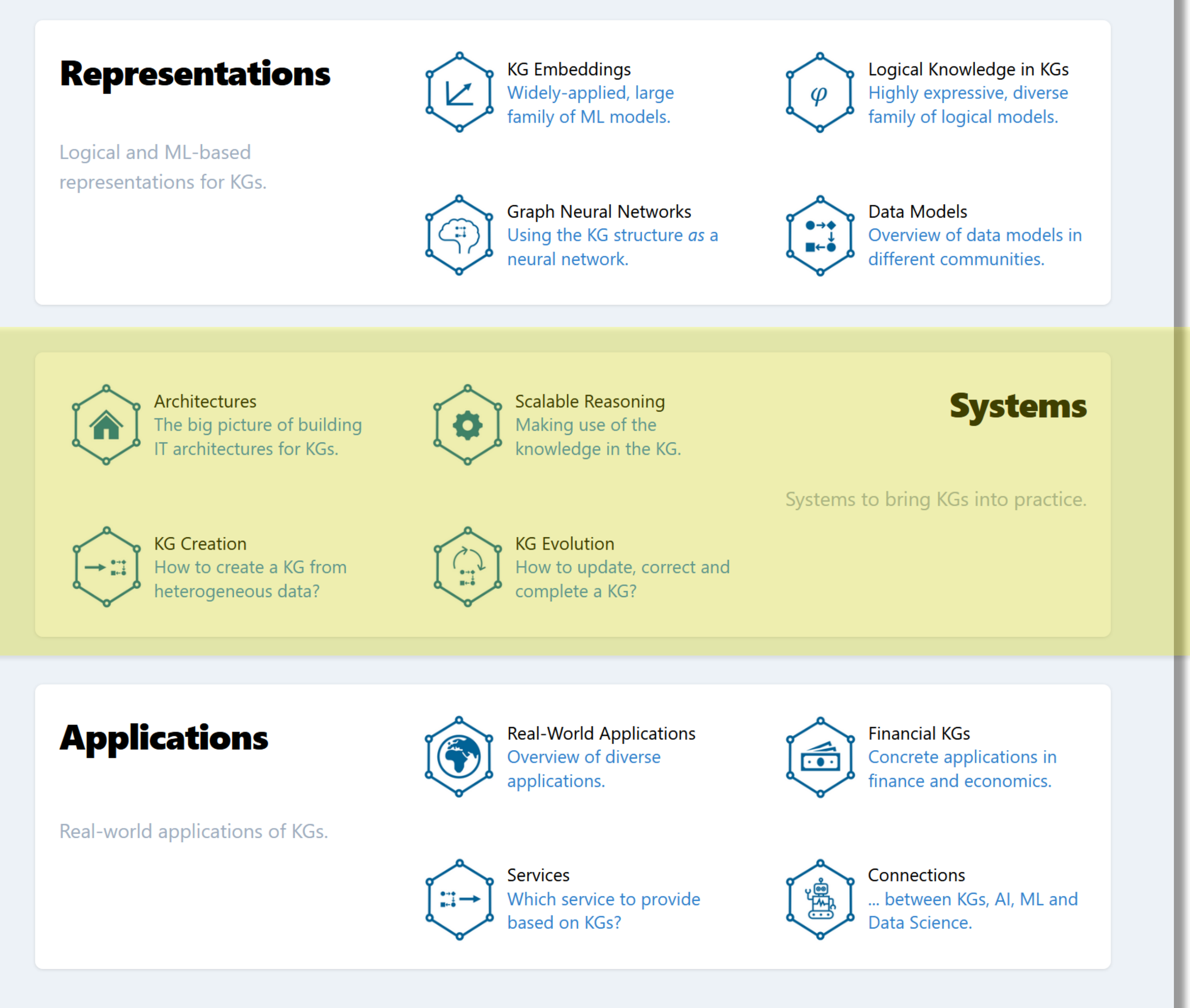


Mutual Exclusion



Aleksandar Pavlovic, Emanuel Sallinger: ExpressivE – A Spatio-Funcational Embedding for Knowledge Graph Completion. ICLR 2023.


Knowledge Graph Embeddings
φ
Logical Knowledge
Graph Neural Networks
$\mathcal{L}$
Large Language Models



Representations

Logical and ML-based representations for KGs.

KG Embeddings
Widely-applied, large family of ML models.

Logical Knowledge in KGs
Highly expressive, diverse family of logical models.

Graph Neural Networks
Using the KG structure *as* a neural network.

Data Models
Overview of data models in different communities.

Systems

Systems to bring KGs into practice.

Architectures
The big picture of building IT architectures for KGs.

Scalable Reasoning
Making use of the knowledge in the KG.

KG Creation
How to create a KG from heterogeneous data?

KG Evolution
How to update, correct and complete a KG?

Applications

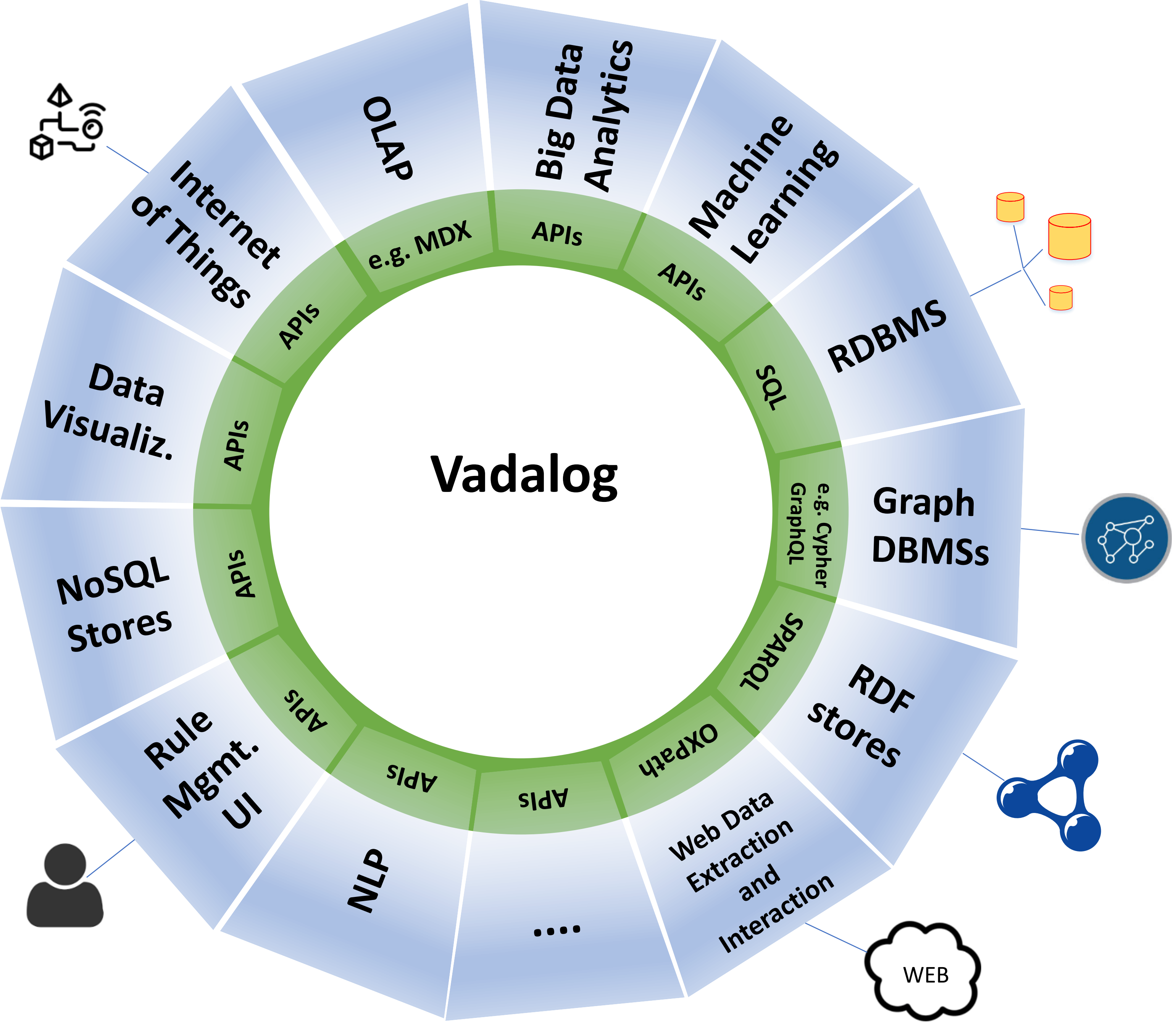
Real-world applications of KGs.

Real-World Applications
Overview of diverse applications.

Financial KGs
Concrete applications in finance and economics.

Services
Which service to provide based on KGs?

Connections
... between KGs, AI, ML and Data Science.


Vadalog
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
....
APIs
NLP
APIs
Rule Mgmt. UI
APIs
NoSQL Stores
APIs
Data Visualiz.
APIs
Internet of Things
APIs

Neurosymbolic Reasoning

Reasoning Tasks

AI-driven Applications

Decision Support

ML-able tasks

Top brain
to setup reasoning tasks

Left brain
Logic Symbolic Reasoning

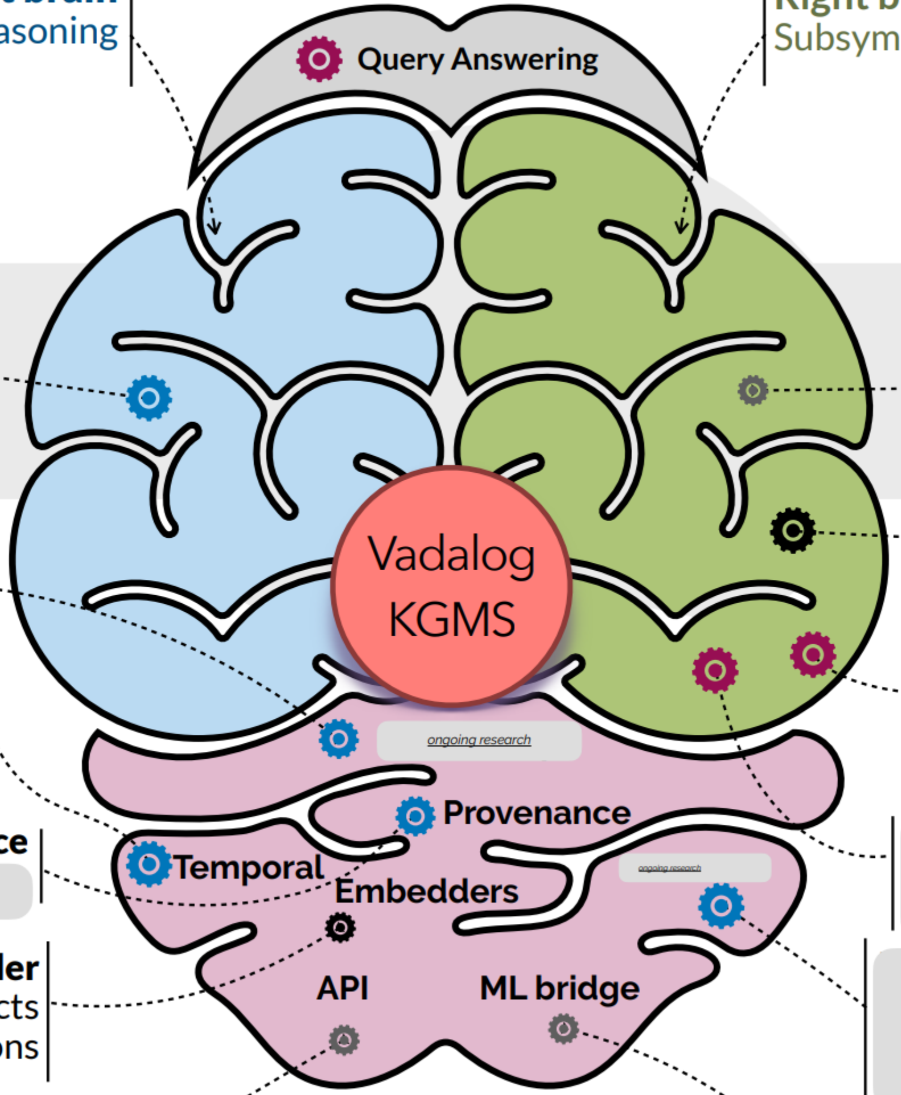
Right brain
Subsymbolic Reasoning

AI Technologies in use

- Logic Reasoning
- Neural Reasoning
- Statistical Reasoning
- Other/custom

Extensional Knowledge

Intensional Knowledge



Logic Reasoner
to build a reasoning graph based on extensional knowledge and formalized domain experience

Subsymbolic Reasoner

ongoing research

ongoing research

the full reasoning graph, the bridge to subsymbolic reasoners

ongoing research

Markov Logic Probabilistic Reasoner
to apply expressive probabilistic reasoning based on a variants of Markov Logic Networks

Temporal

ongoing research

ongoing research

Provenance

ongoing research

ongoing research

Embedder
to associate input facts to vector-based representations

API
to read data from a variety of external sources (RDBMs, graph DBMSs, RDF stores, OLAP stores and DWHs, NoSQL stores, the Web, …)

Bottom brain
Reasoning Modes and Interactions

ML Bridge
to train ML models from reasoning results, or to use ML models to provide extensional knowledge

Hostile **Takeovers** and **Golden Powers**

KG19

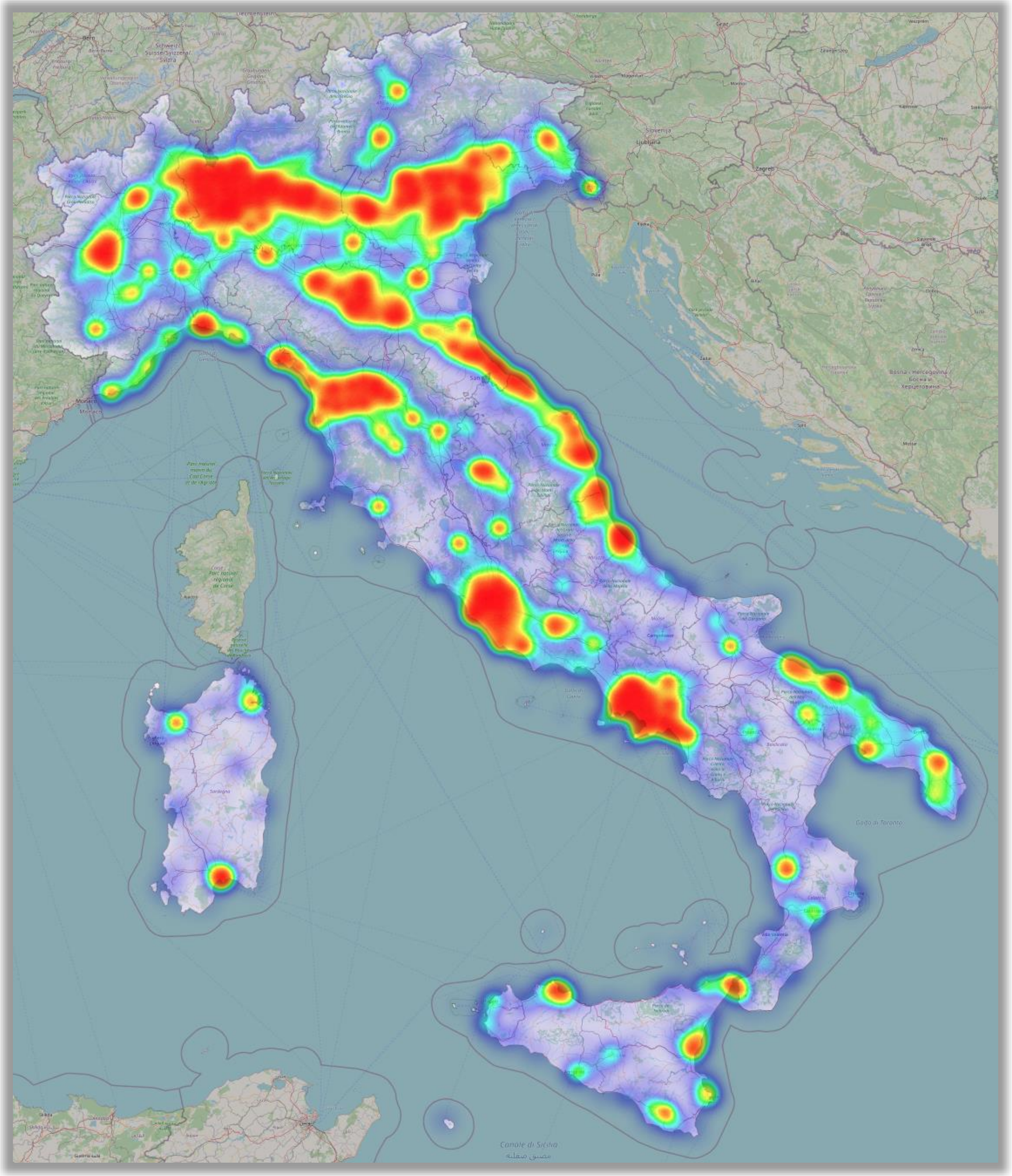
BANCA D'ITALIA UNIVERSITY OF OXFORD TU WIEN POLITECNICO MILANO

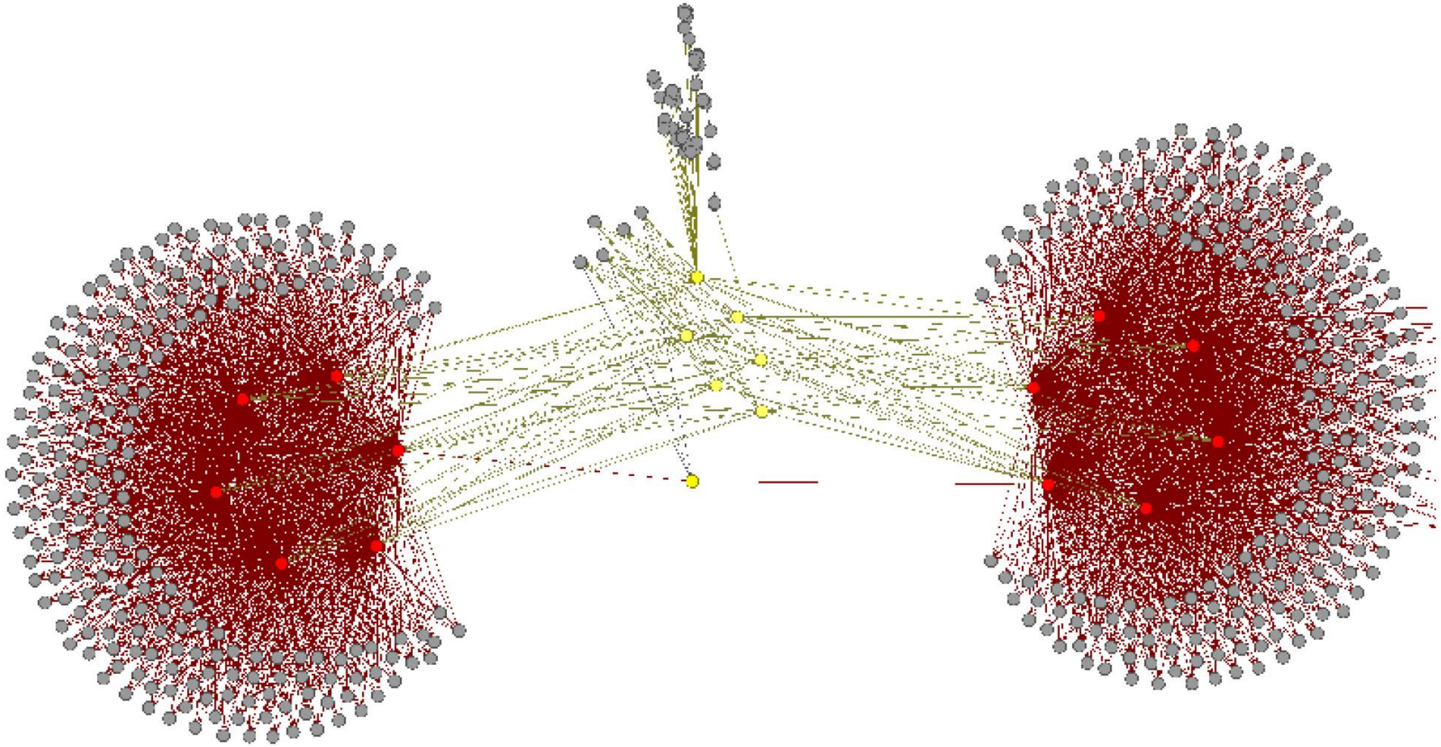
- Applying KG reasoning to support the application of **Golden Powers** to prevent hostile takeover attempts on strategic companies.

- In crises, taking advantage of market turbulence, specific players are inclined to pursuing **takeovers** and affect the **public control** over such companies.

- Many countries have developed **legal frameworks** (e.g., Golden Powers in Italy) to **protect strategic companies** by vetoing specific share acquisition operations.

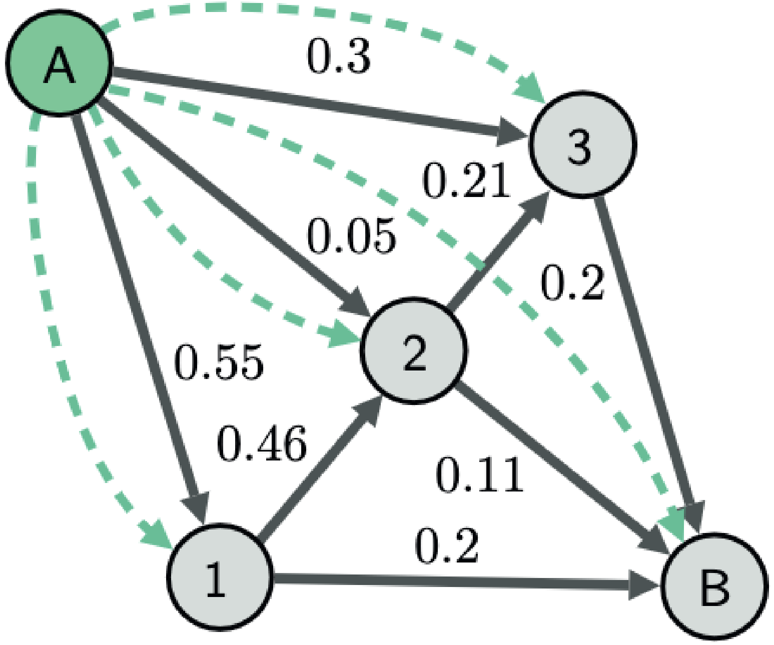
Hostile **Takeovers** and **Golden Powers**





Hostile **Takeovers** and **Golden Powers**

- Company Control

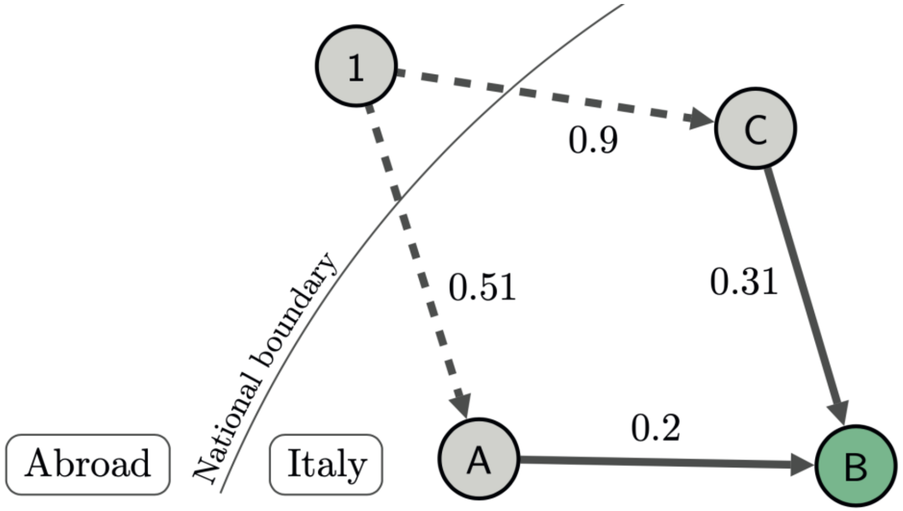


$$Company(x) \rightarrow Control(x, x)$$

$$Control(x, y), Own(y, z, w), v = sum(w), v > 0.5 \rightarrow Control(x, z)$$

Hostile **Takeovers** and **Golden Powers**

- Golden Power Check

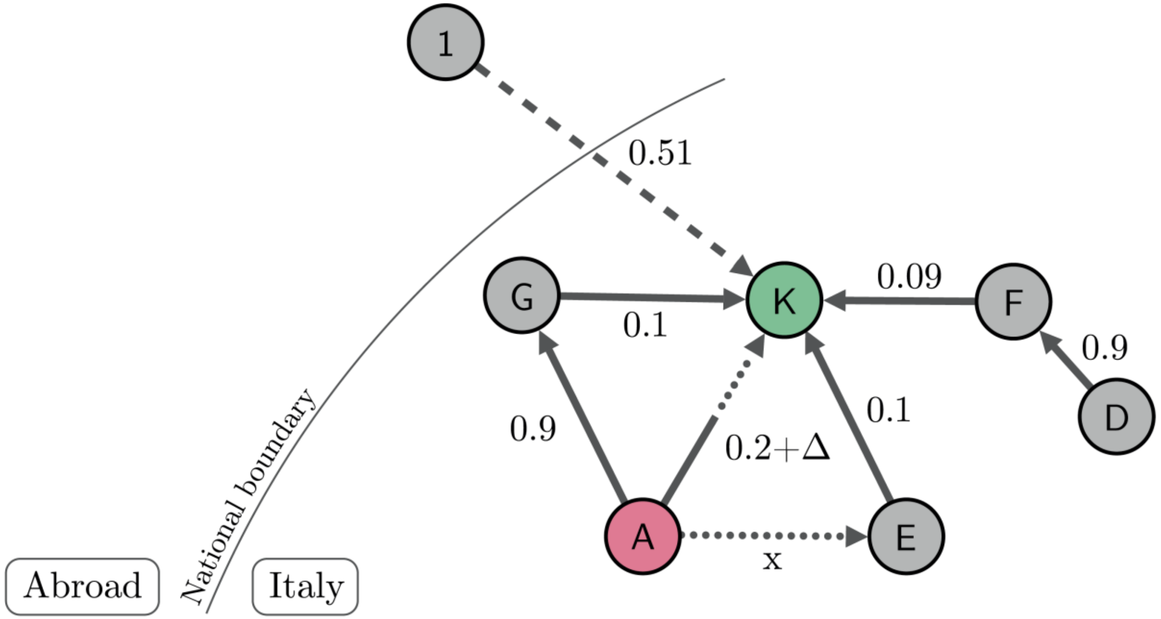


$$Attacker(x), \neg Attacker(y), Tx(x,y,w) \rightarrow Own(x,y,w)$$

$$Attacker(x), Target(y), Control(x,y) \rightarrow GPCheck(x,y)$$

Hostile **Takeovers** and **Golden Powers**

- Golden Power Protection

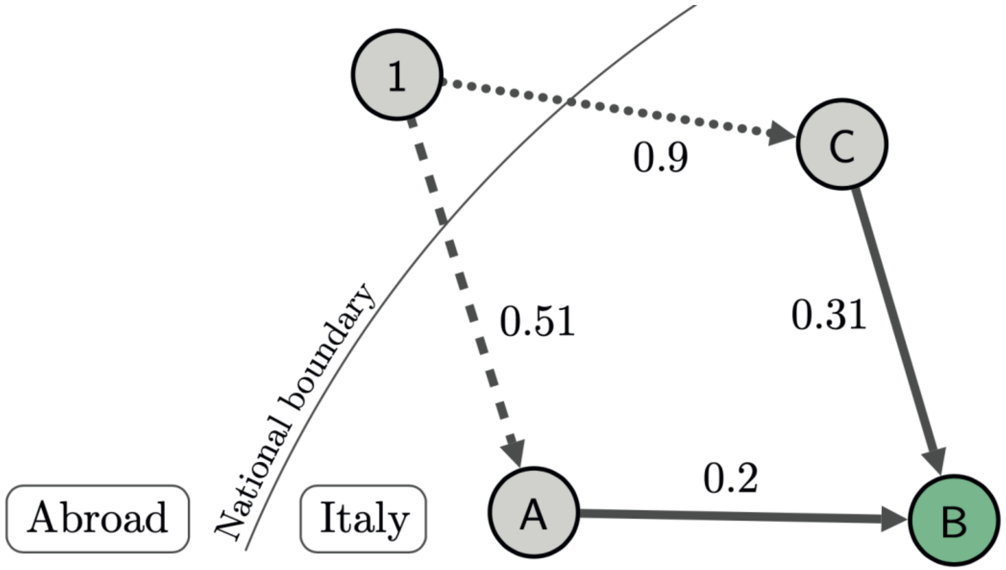


$$Control(x,y), Own(y,z,w), v = sum(w) \rightarrow PControl(x,z,v)$$

$$P(x), T(y), PControl(x,y,v), v < 0.5 \rightarrow Prot(x,y,0.5-v)$$

Hostile **Takeovers** and **Golden Powers**

- Cautious Golden Power Check

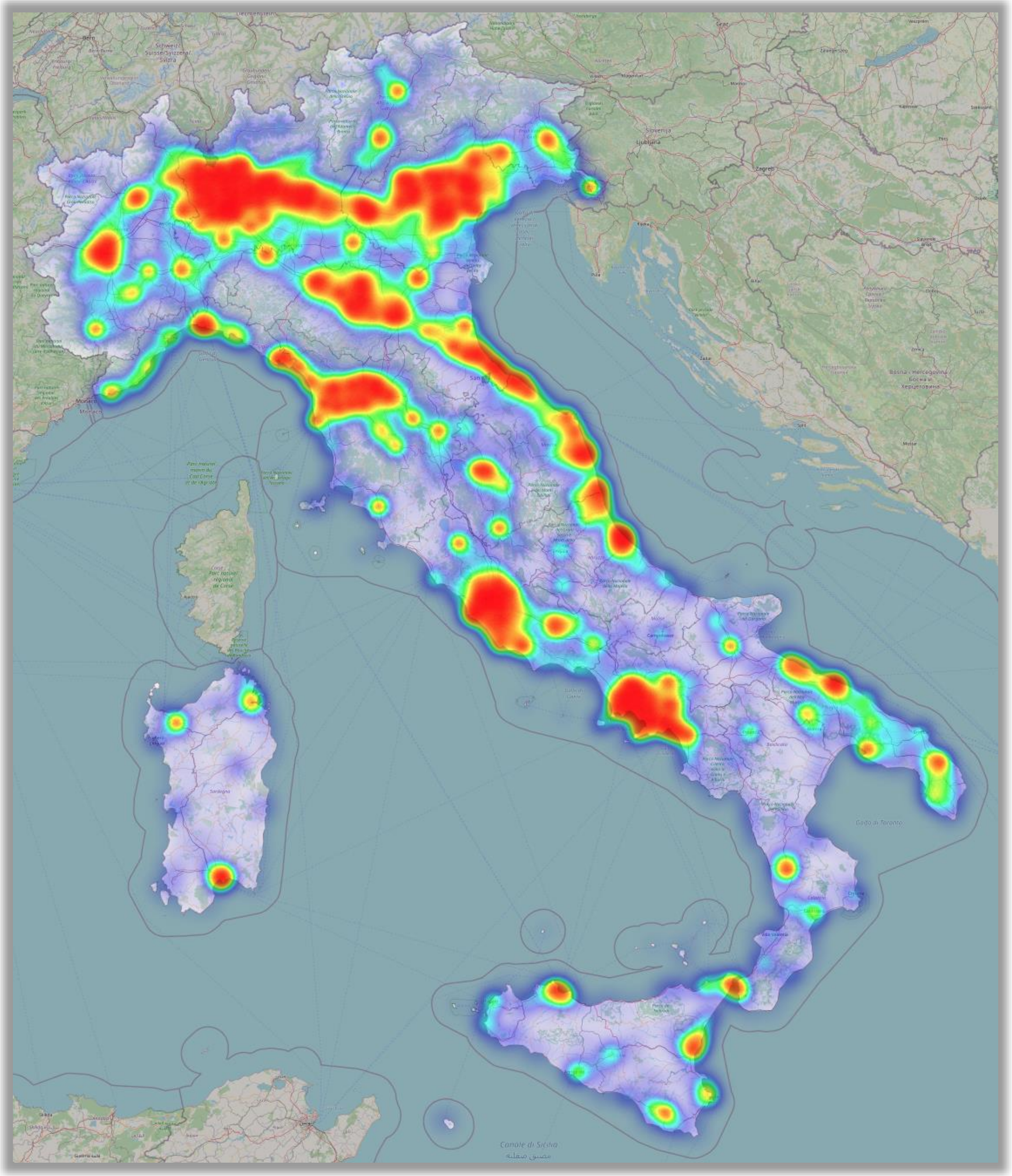


$$\neg \mathit{Attacker}(x), \neg \mathit{Attacker}(y), \mathit{Own}(x,y,v), w = \mathit{sum}(v) \rightarrow \mathit{Assigned}(y,w)$$

$$\mathit{Assigned}(y,w), w < 1 \rightarrow \exists z\, \mathit{Company}(z), \mathit{Attacker}(z), \mathit{Own}(z,y,1-w)$$

$$\mathit{Attacker}(x), \neg \mathit{Attacker}(y), \mathit{Tx}(x,y,w), v = \mathit{sum}(w) \rightarrow \mathit{Own}(x,y,v)$$

$$\mathit{Attacker}(x), T(y), \mathit{Control}(x,y) \rightarrow \mathit{GPCCheck}(x,y)$$



Hostile **Takeovers** and **Golden Powers**

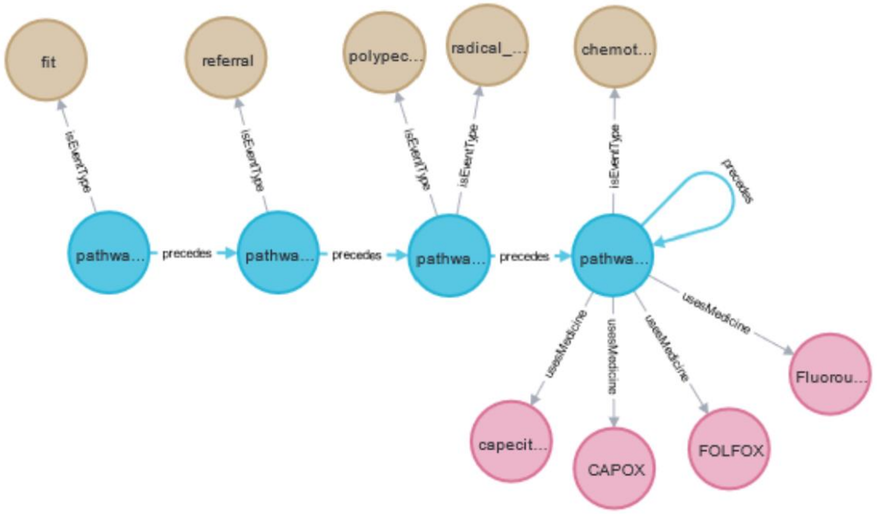
- Applying KG reasoning to support the application of **Golden Powers** to prevent hostile takeover attempts on strategic companies.
- In crises, taking advantage of market turbulence, specific players are inclined to pursuing **takeovers** and affect the **public control** over such companies.
- Many countries have developed **legal frameworks** (e.g., Golden Powers in Italy) to **protect strategic companies** by vetoing specific share acquisition operations.

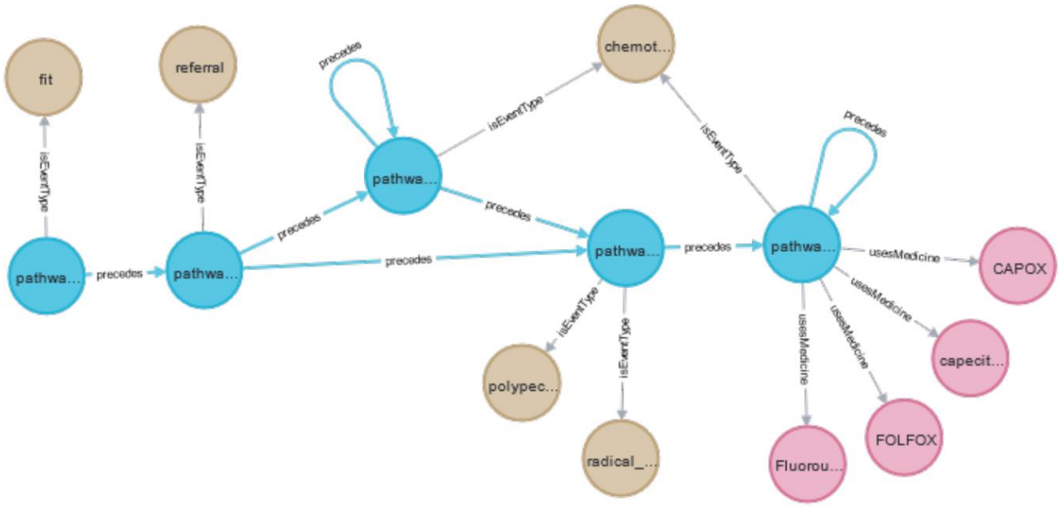
Cancer Pathways

UNIVERSITY OF OXFORD
DEPARTMENT OF COMPUTER SCIENCE

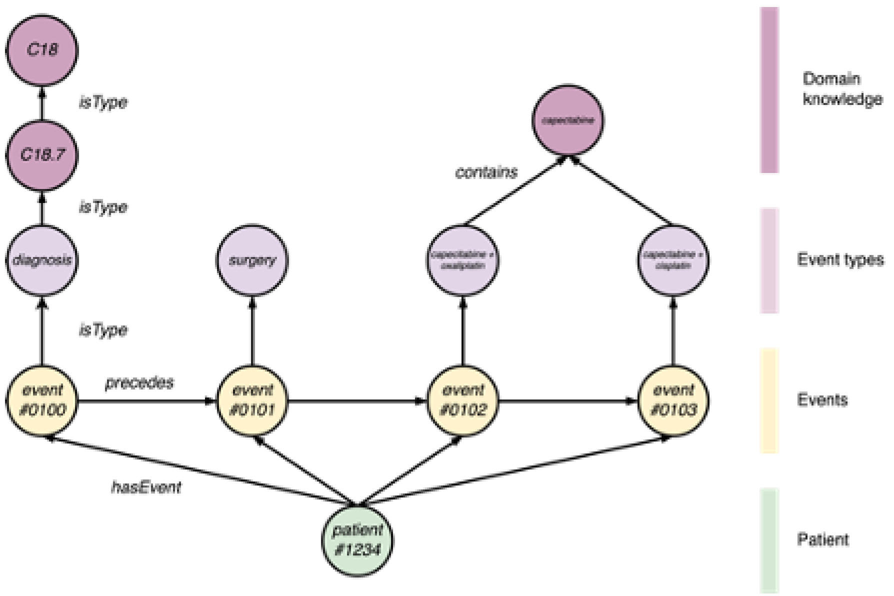
UNIVERSITY OF OXFORD
BIG DATA INSTITUTE
HEALTH DATA SCIENCE

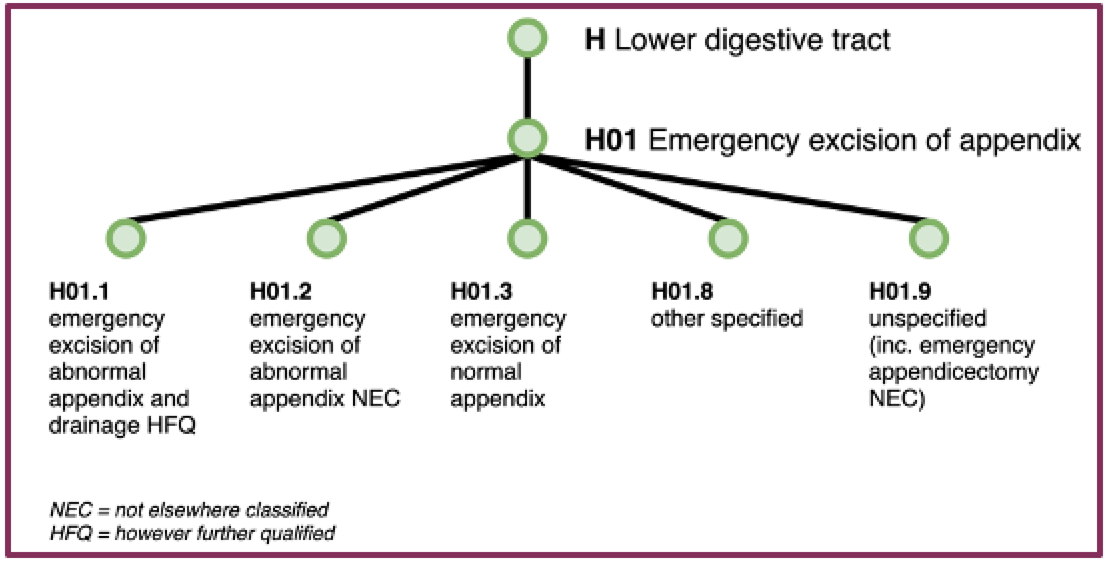
TU WIEN





Cancer Pathways





H Lower digestive tract

H01 Emergency excision of appendix

H01.1 emergency excision of abnormal appendix and drainage HFQ

H01.2 emergency excision of abnormal appendix NEC

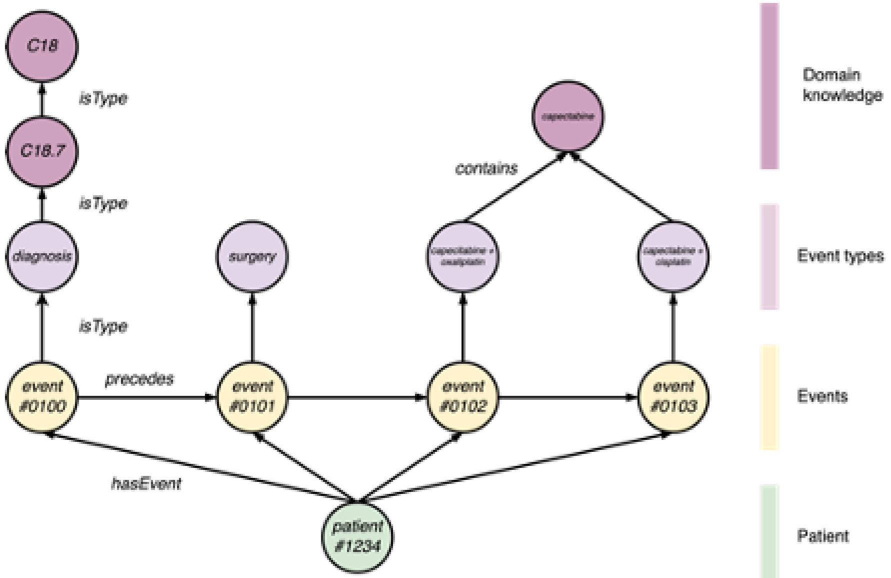
H01.3 emergency excision of normal appendix

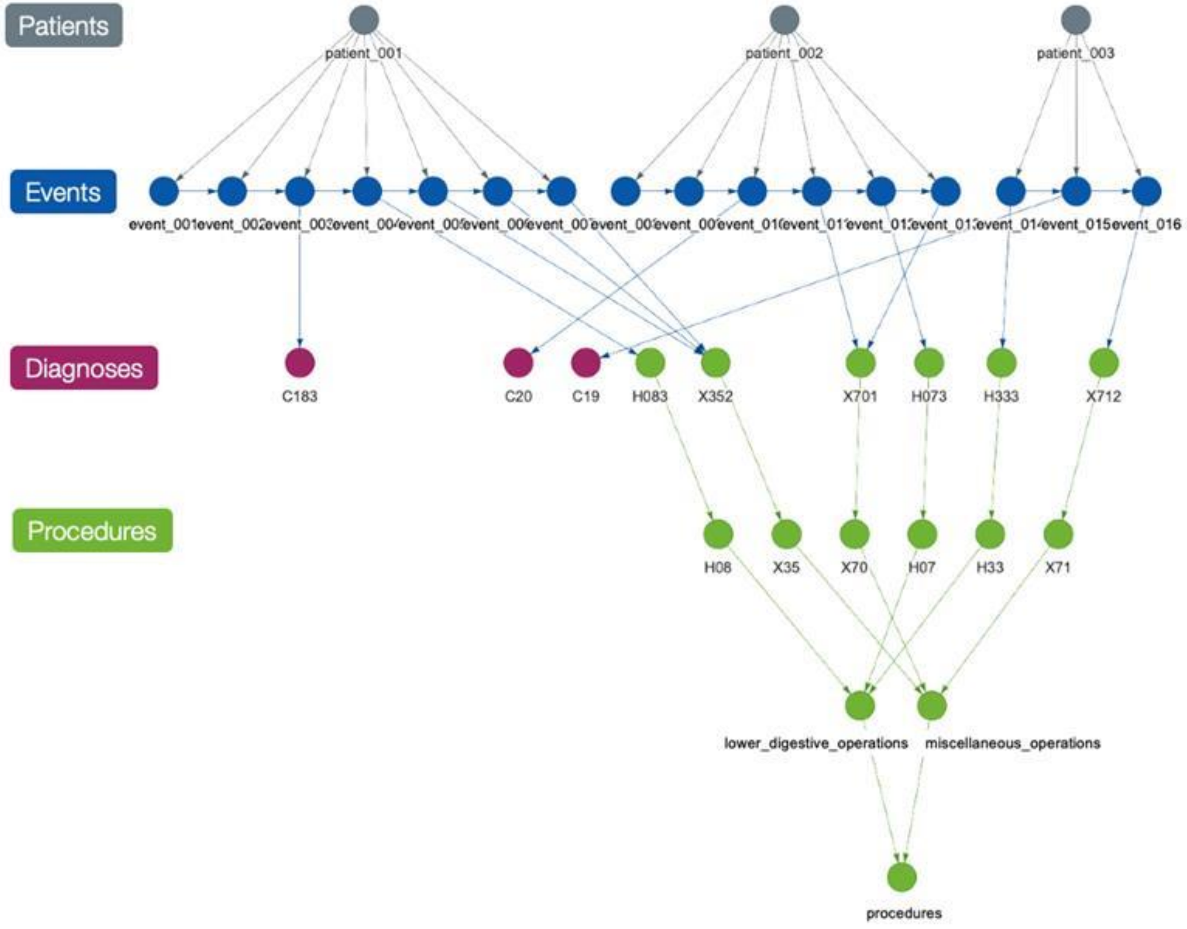
H01.8 other specified

H01.9 unspecified (inc. emergency appendicectomy NEC)

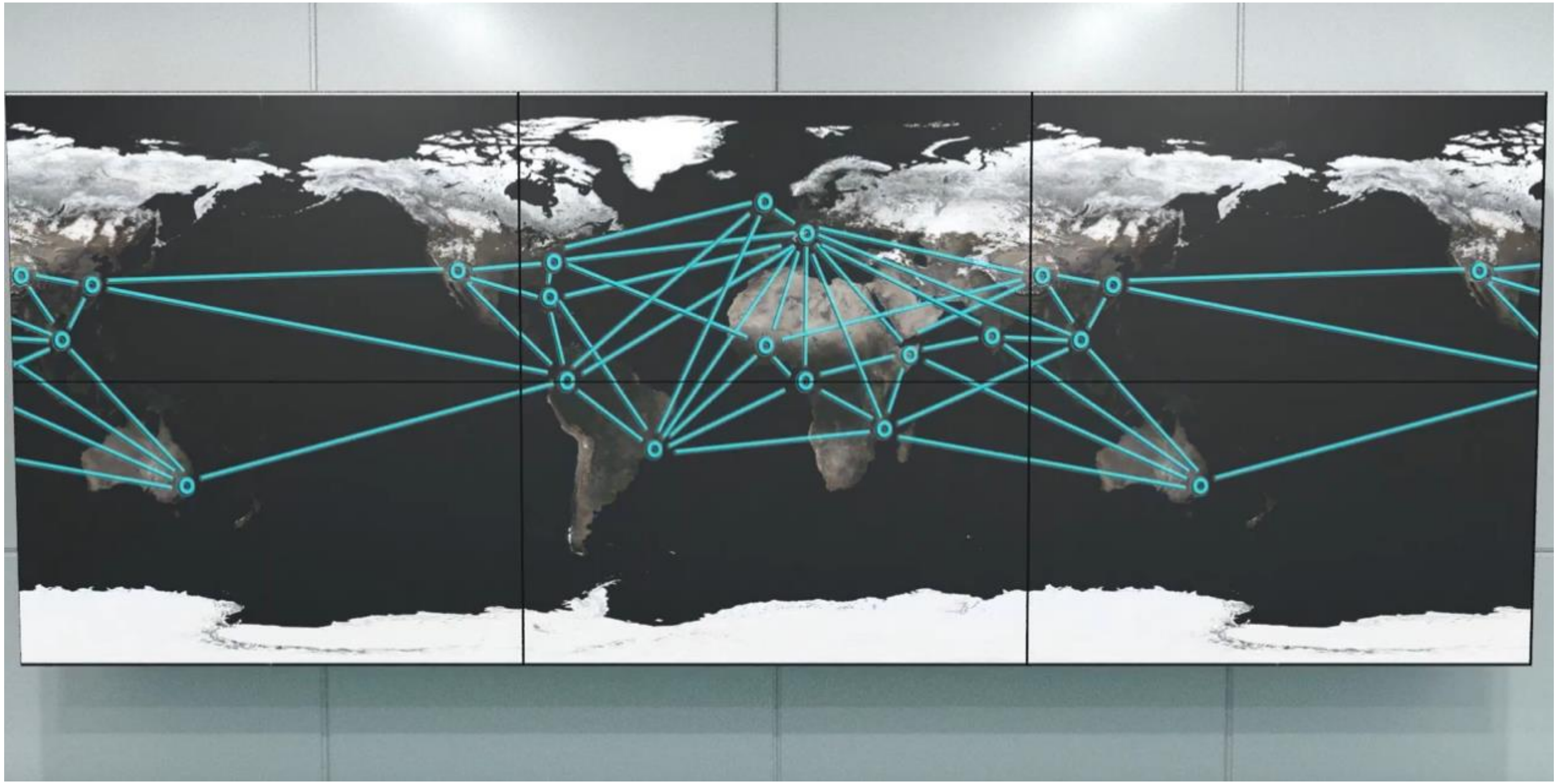
NEC = not elsewhere classified
HFQ = however further qualified

Cancer Pathways

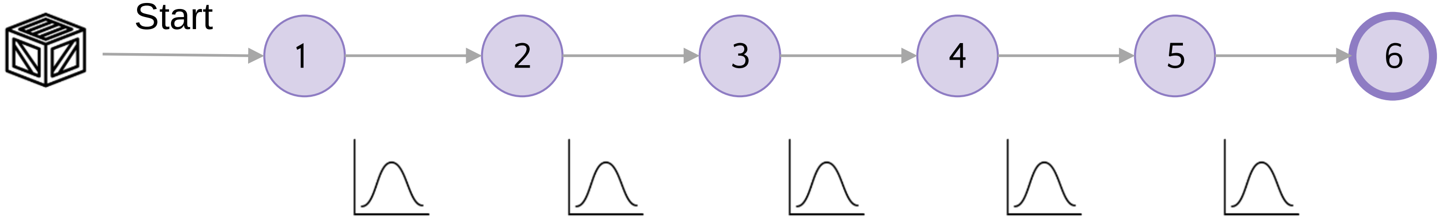




Supply Chains

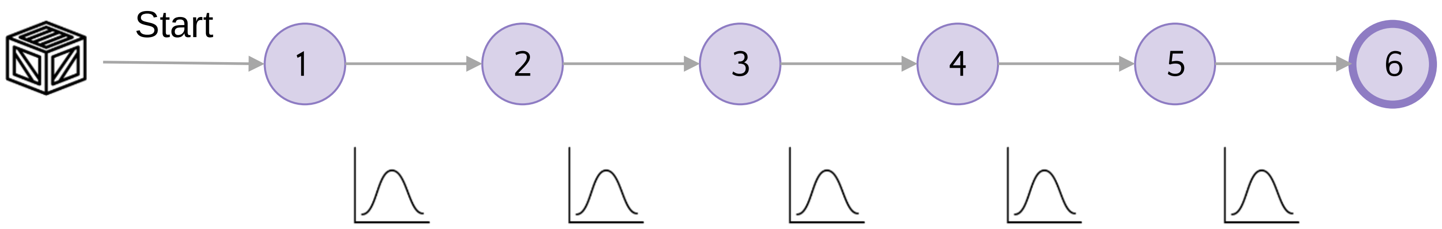


Supply Chains



Historic Data,
Machine Learning

Supply Chains

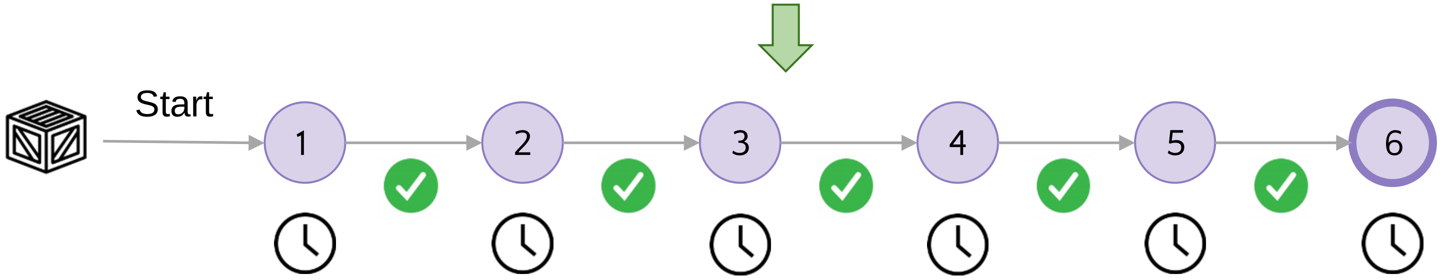


Historic Data,
Machine Learning

→ **Rules:** Express what you want, not how you want it done.

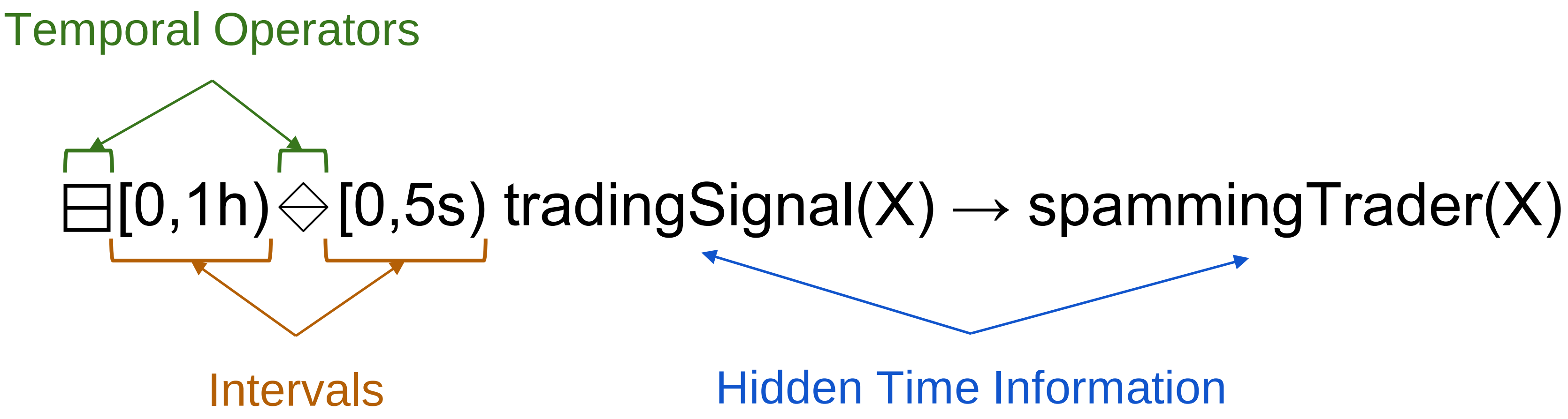
```
path(X, Y, Time) :-
     start(X, T0),
     section(X, Y),
     Time = T0 + stat:mean(X, Y)
```

```
path(X, Z, Time) :-
          path(X, Y, T1),
          section(Y, Z),
          Time = T + stat:mean(Y, Z).
```



Temporal: DatalogMTL

Datalog extended with operators from the Metric Temporal Logic



Temporal Operators

$\boxminus[0,1h) \diamondminus[0,5s)$ tradingSignal(X) → spammingTrader(X)

Intervals

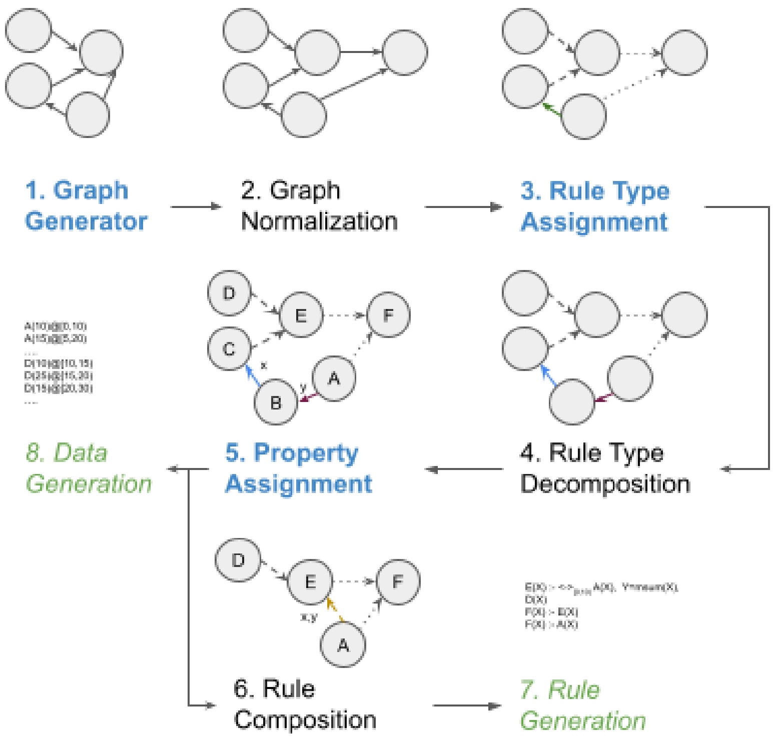
Hidden Time Information

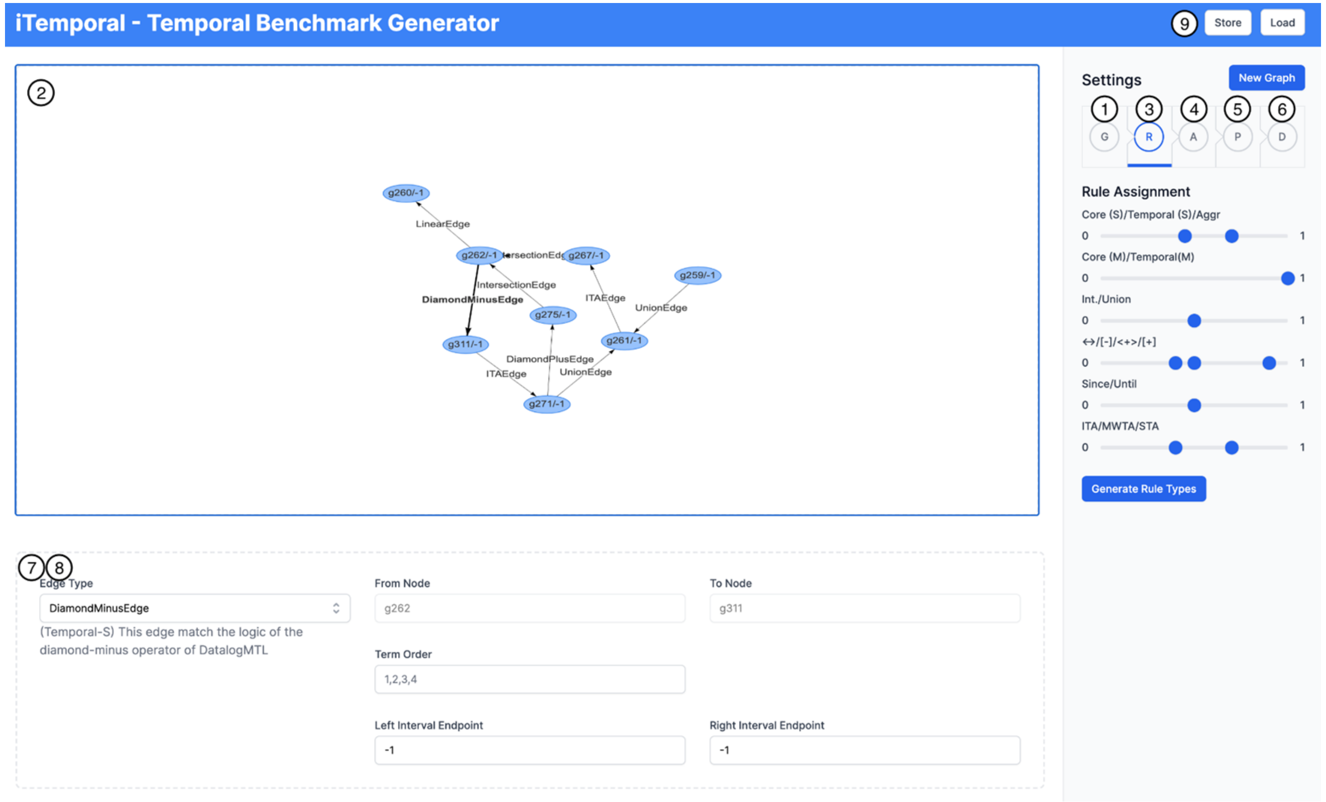
iTemporal

Temporal Benchmark Suite

TEMPORAL

iTemporal: An Extensible Generator of Temporal Benchmarks





iTemporal - Temporal Benchmark Generator

Store | Load

Settings | New Graph

Rule Assignment

Core (S)/Temporal (S)/Aggr 0 1

Core (M)/Temporal(M) 0 1

Int./Union 0 1

↔/[-]/<+>/[+] 0 1

Since/Until 0 1

ITA/MWTA/STA 0 1

Generate Rule Types

Edge Type: DiamondMinusEdge

(Temporal-S) This edge match the logic of the diamond-minus operator of DatalogMTL

From Node: g262

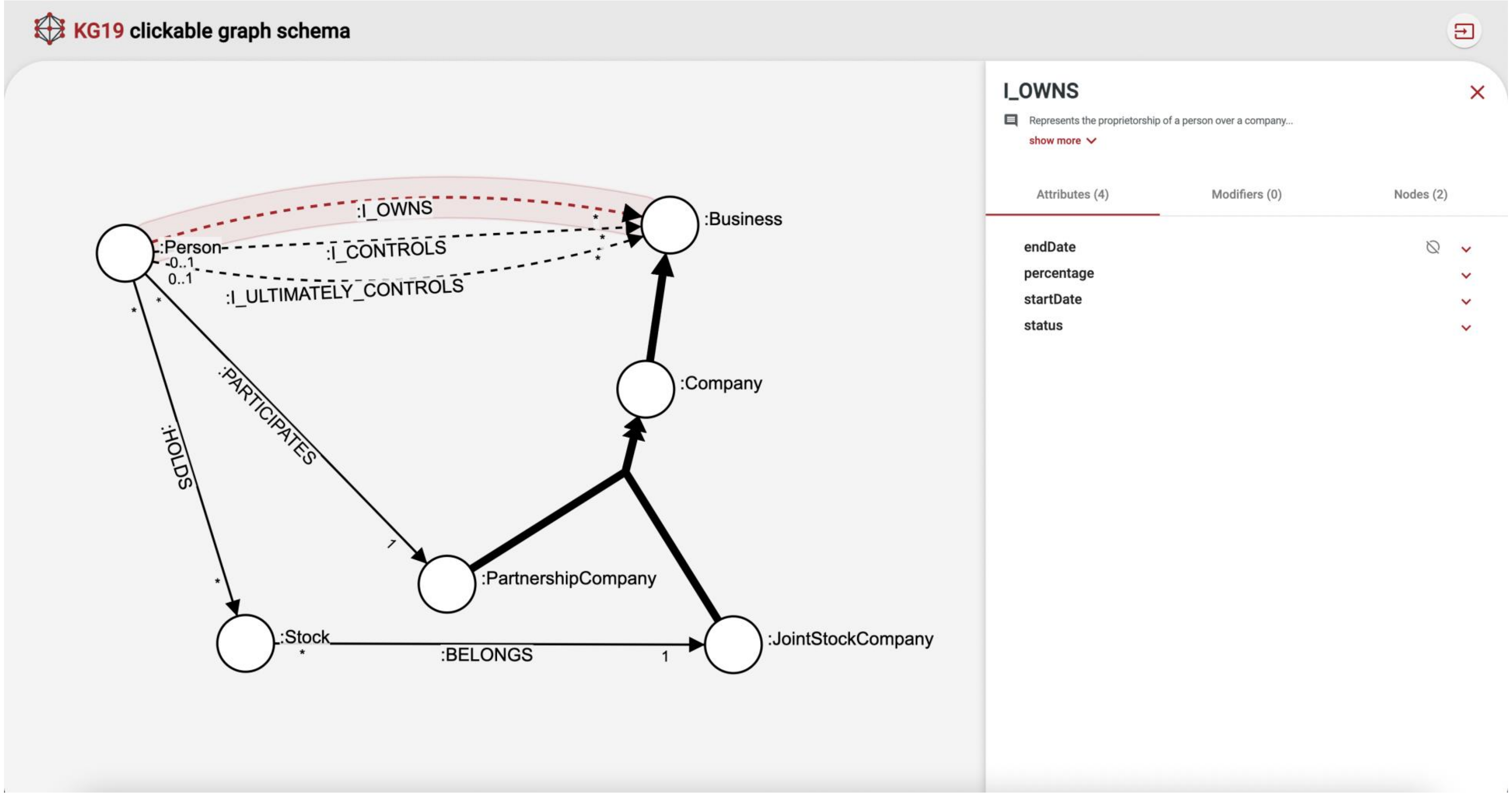
To Node: g311

Term Order: 1,2,3,4

Left Interval Endpoint: -1

Right Interval Endpoint: -1

Graph Schema Design



KG19 clickable graph schema

I_OWNS

Represents the proprietorship of a person over a company...

show more

Attributes (4) | Modifiers (0) | Nodes (2)

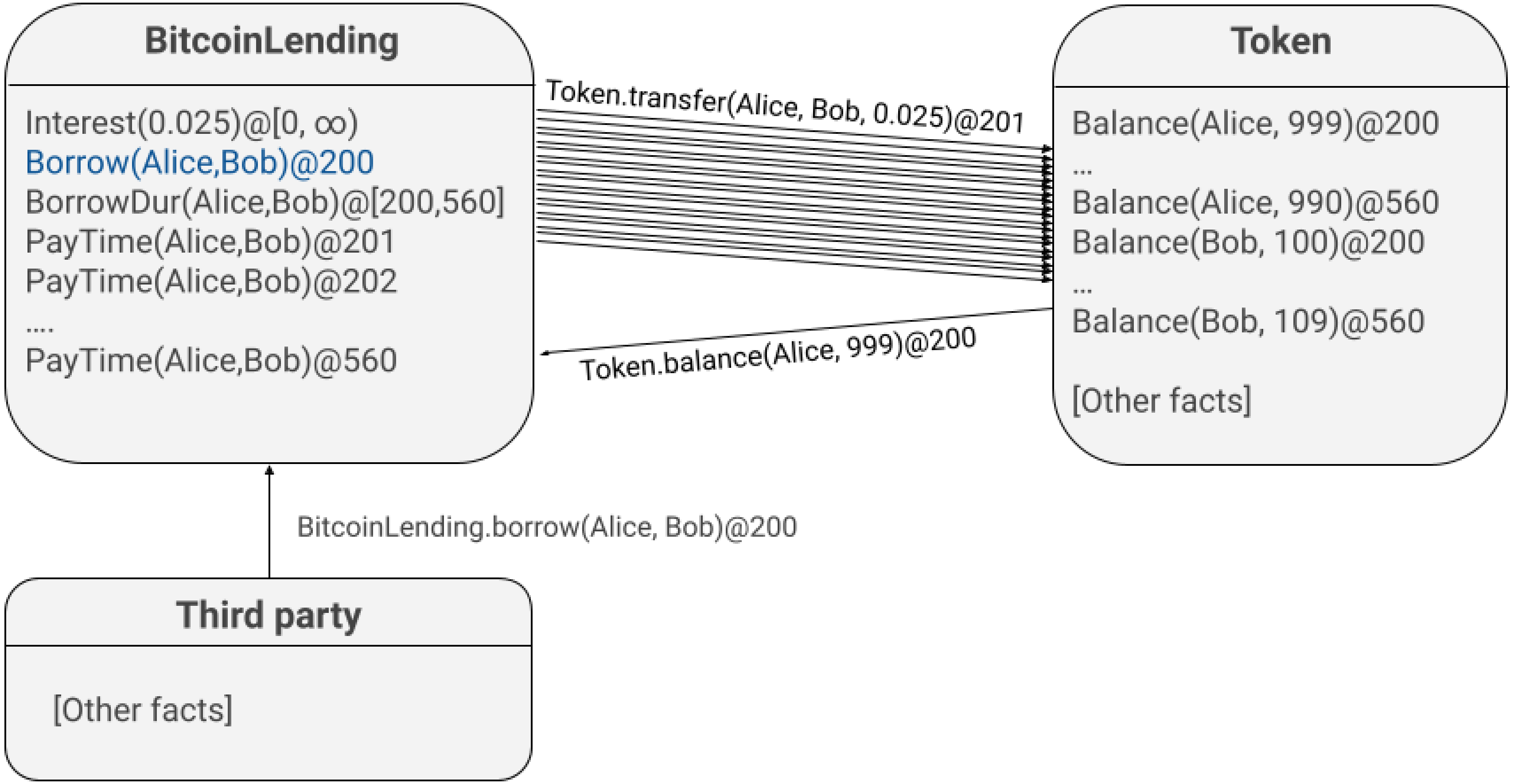
endDate

percentage

startDate

status

Modelling Smart Contracts with Temporal Datalog



BitcoinLending

Interest(0.025)@[0, ∞)
Borrow(Alice,Bob)@200
BorrowDur(Alice,Bob)@[200,560]
PayTime(Alice,Bob)@201
PayTime(Alice,Bob)@202
….
PayTime(Alice,Bob)@560

Token.transfer(Alice, Bob, 0.025)@201

Token.balance(Alice, 999)@200

Token

Balance(Alice, 999)@200
…
Balance(Alice, 990)@560
Balance(Bob, 100)@200
…
Balance(Bob, 109)@560

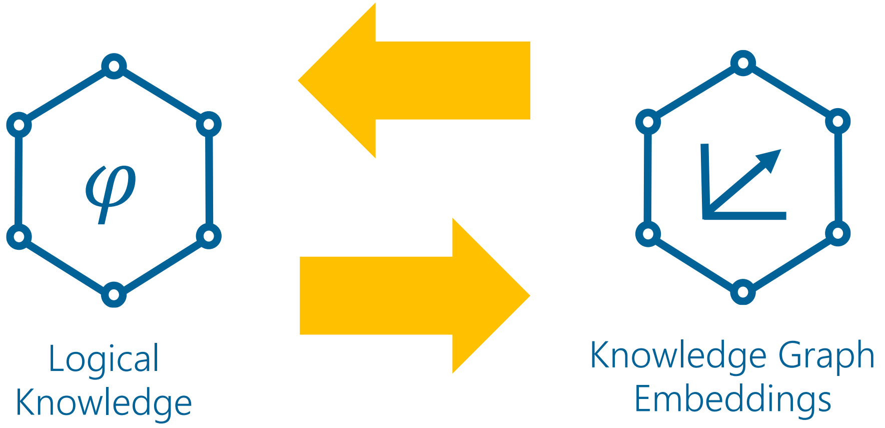
[Other facts]

BitcoinLending.borrow(Alice, Bob)@200

Third party

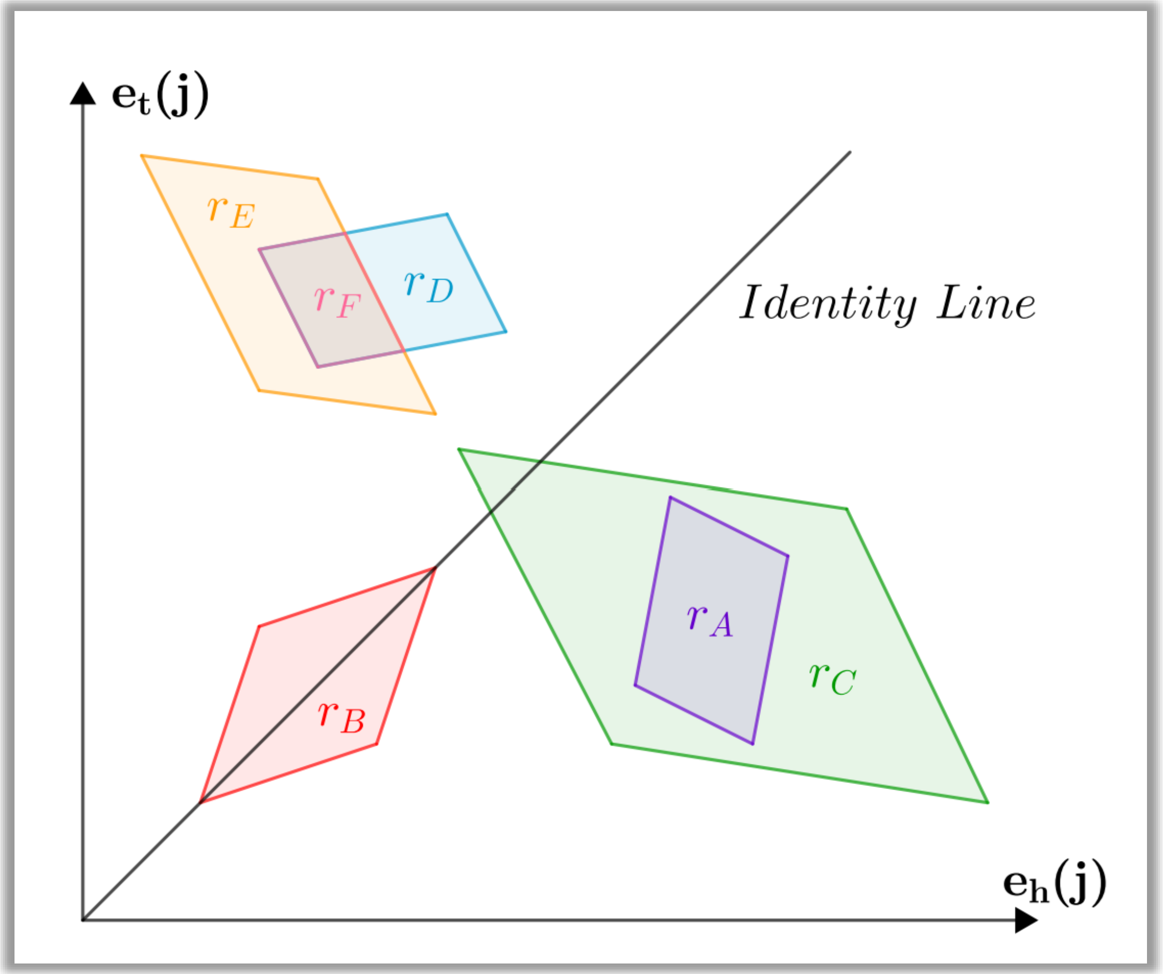
[Other facts]

“ExpressivE”: Expressive Knowledge Graph Embeddings



Logical Knowledge

Knowledge Graph Embeddings

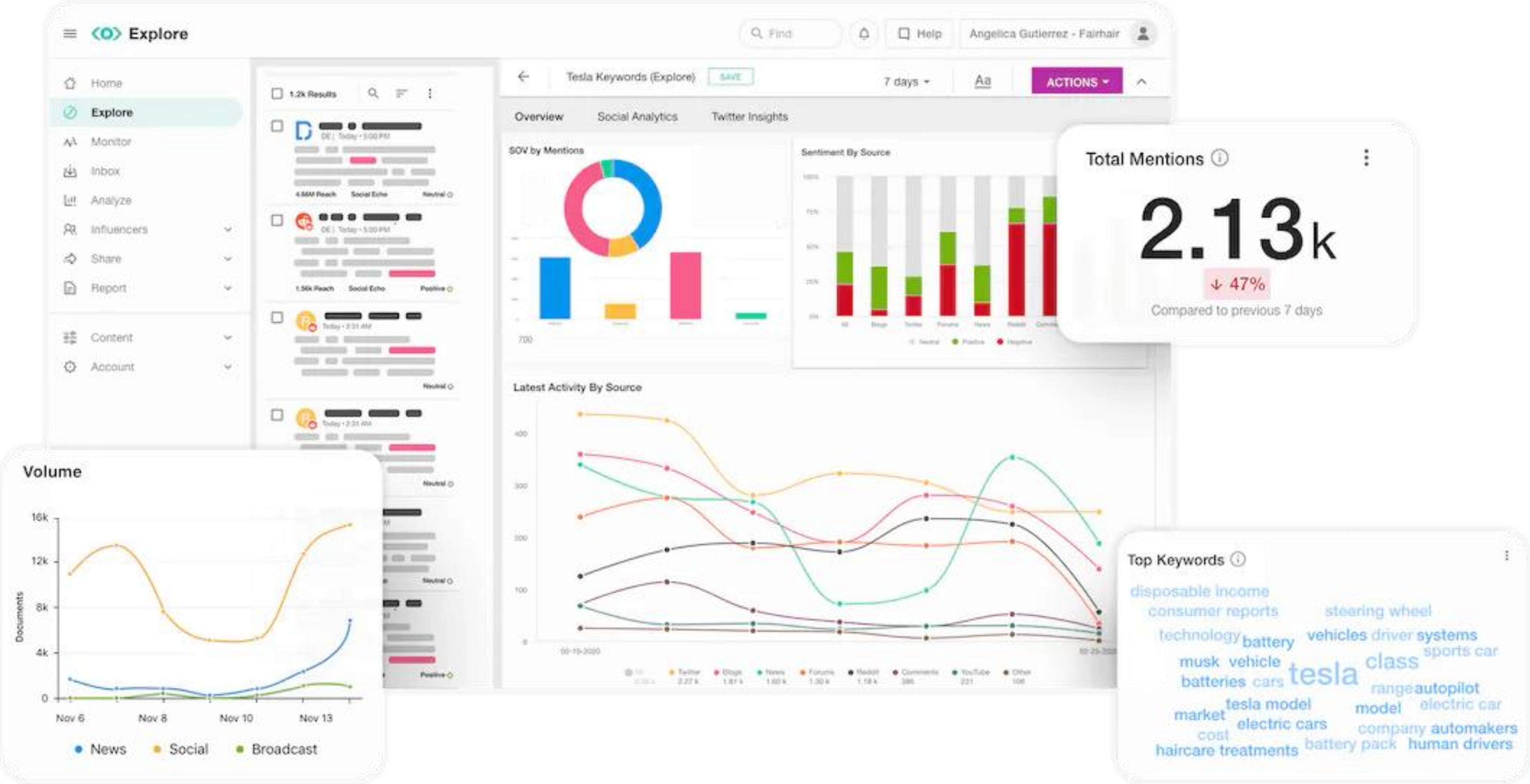


Media Intelligence

Meltwater

UNIVERSITY OF OXFORD

TU WIEN



Knowledge Graphs in Action

Part 2: Theory to Practice

Emanuel Sallinger

TU WIEN

Knowledge Graph Lab

UNIVERSITY OF OXFORD

AI Summer School 2023

AIML Center for Artificial Intelligence and Machine Learning

asai austrian society for artificial intelligence

3 July 2023

Knowledge Graphs in Action

Georg Gottlob

Emanuel Sallinger

AI Summer School 2023

3 July 2023