RDFS & OWL Reasoning for Linked Data

Axel Polleres, Aidan Hogan, Renaud Delbru, Jürgen Umbrich

http://www.polleres.net
http://aidanhogan.com

29 July 2013, 14:00–17:30

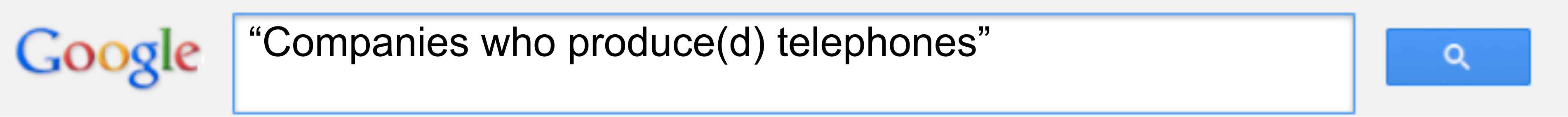
Reasoning on Linked Data:

- Goal/Scope of this lecture:
  - Support *structured queries*, i.e. **conjunctive query answering** (SPARQL) …
  - … combining various **linked data** sources
  - … taking into account the implicit answers inferable by **RDFS+OWL**

- Interesting Questions:
  - What does it buy me (for *practical* examples)?
  - Which reasoning techniques are available in principle?
  - What are (open) challenges?

… along the way we will try introducing the basics of the standards involved (SPARQL, RDFS, OWL, Rules, Query rewriting...)

Some Examples…



- The data is there!

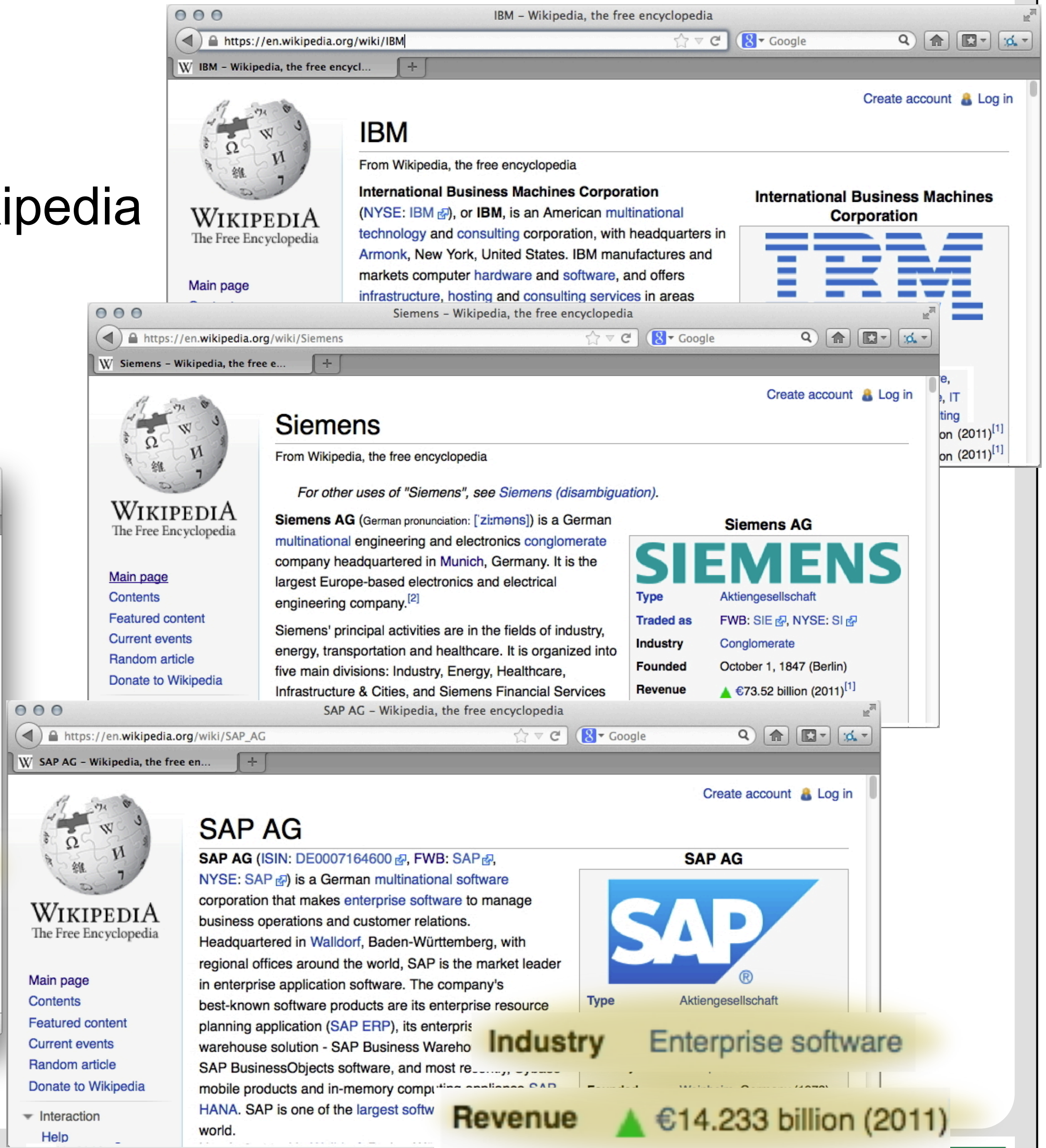
e)

Google “News on NYT about companies with a revenue greater than 10B”

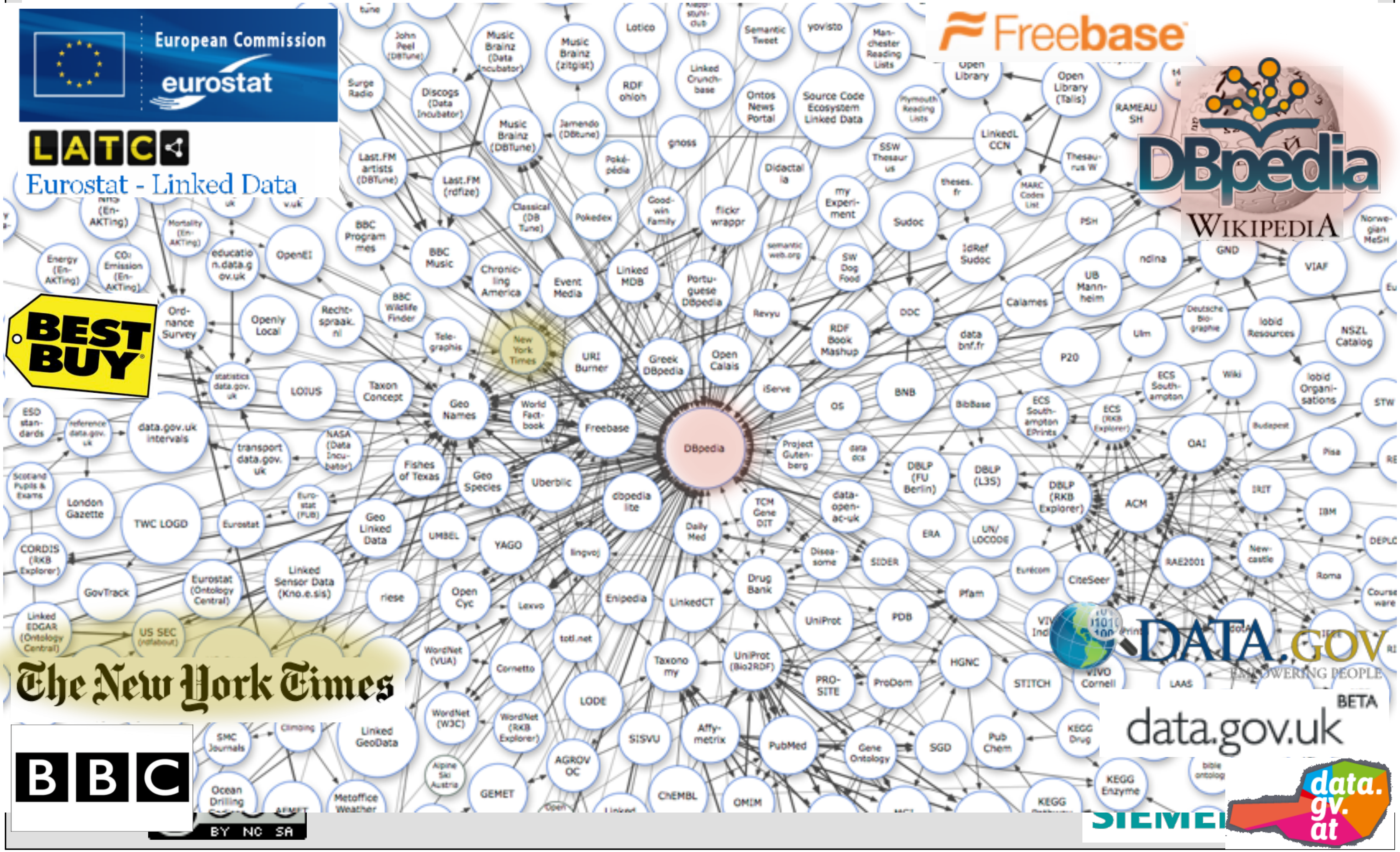
- Again, the data is there!

Wikipedia

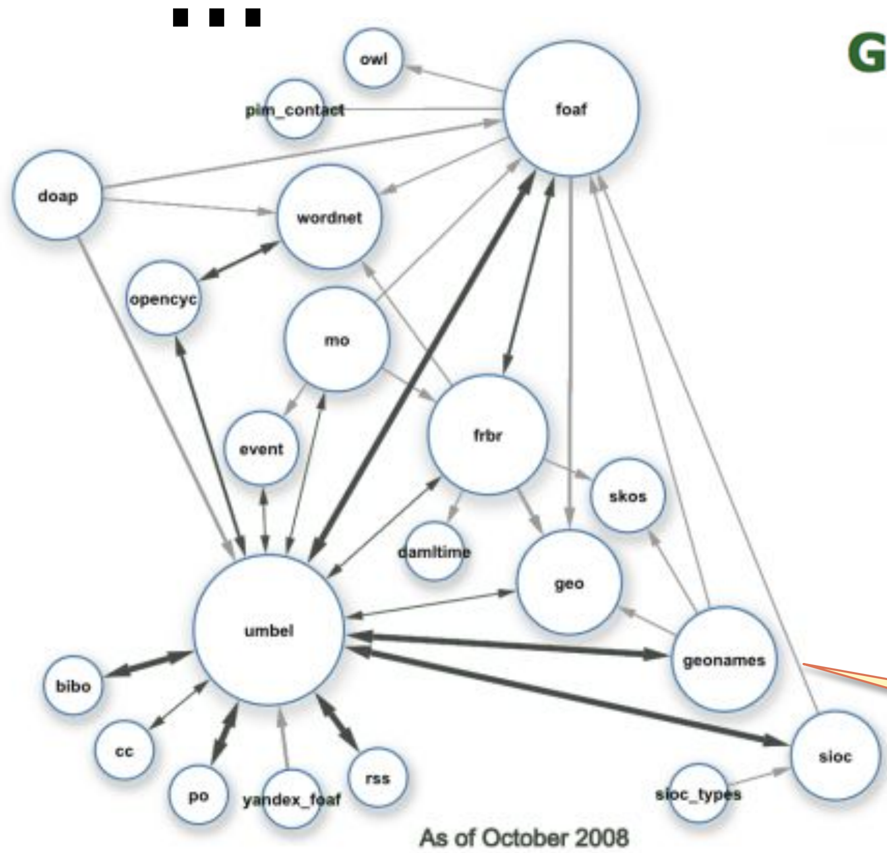
New York Times thesaurus and article search API



The data is there! … as Linked Data even!



Data is described using – also linked - vocabularies



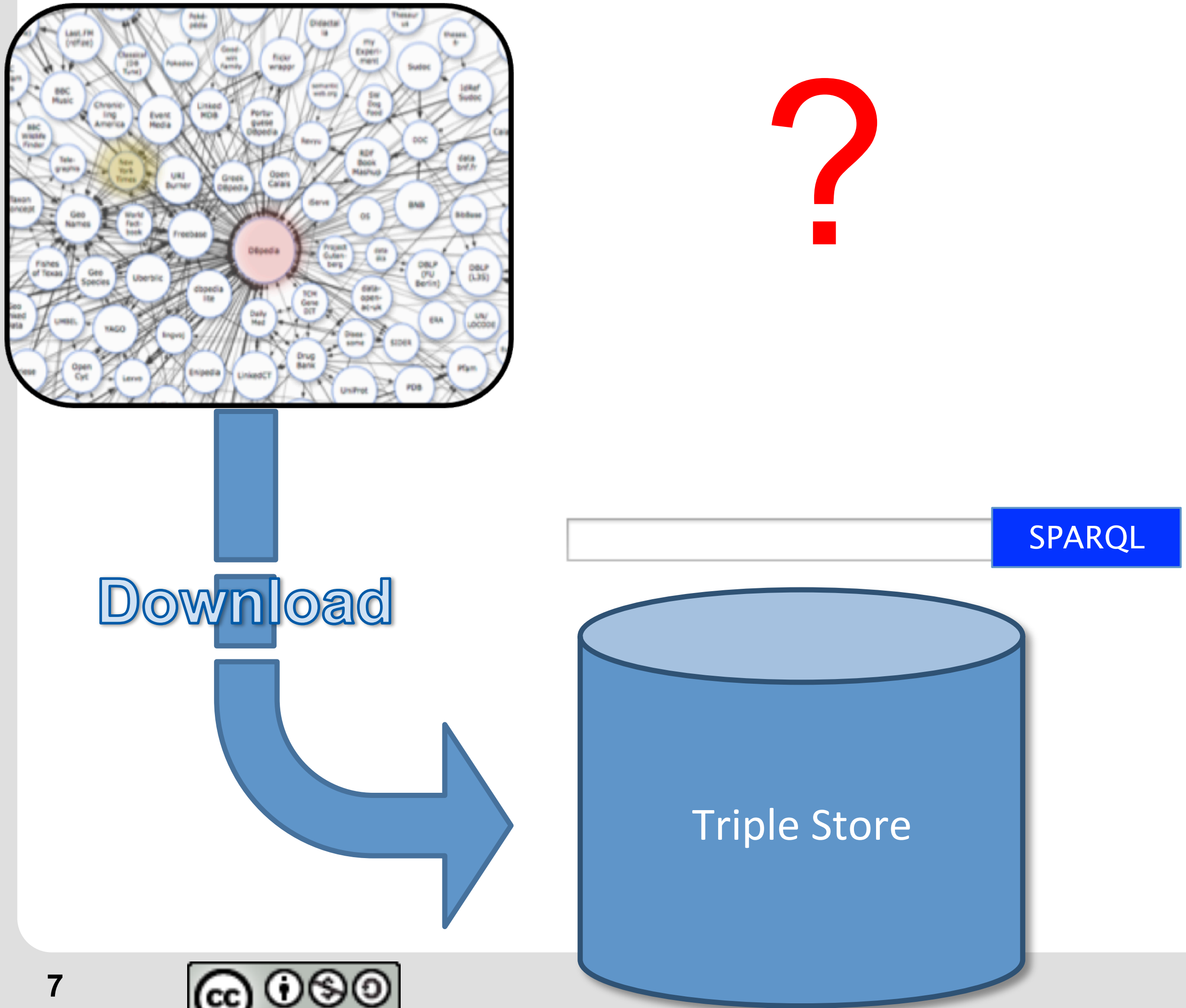
*"Links" here typically described using the **RDFS** and **OWL** (meta-)vocabularies*

Image from http://blog.dbtune.org/public/.081005_lod_constellation_m.jpg ; Giasson, Bergman

Cf. also http://lov.okfn.org/dataset/lov/ "Linked Open Vocabularies", mentioned by Chris

Semantic Search engine:

- Can I just download all that data into a triple store and query it with SPARQL?



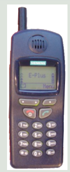
How to formulate queries?

"Companies who produce(d) telephones"

dbo:Organisation
foaf:made
yago:Telephone

?

dbr:Siemens rdf:type dbo:Organisation.
dbr:Siemens foaf:made dbr:Siemens_C25.
dbr:Siemens_C25 a yago:Telephone.



dbr:Siemens_C25

dbr:IBM_Simon

dbr:IBM foaf:made dbr:IBM_Simon.
dbr:IBM rdf:type dbo:Organisation.
dbr:IBM_Simon rdf:type yago:Telephone .

How to formulate queries?

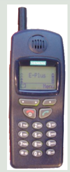
“Companies who produce(d) telephones”

dbo:Organisation
foaf:made
yago:Telephone

?

rdf:type written as “a” in Turtle RDF syntax
http://www.w3.org/TR/turtle/

dbr:Siemens **a** dbo:Organisation.
dbr:Siemens foaf:made dbr:Siemens_C25.
dbr:Siemens_C25 a yago:Telephone.



dbr:Siemens_C25

dbr:IBM_Simon

dbr:IBM foaf:made dbr:IBM_Simon.
dbr:IBM **a** dbo:Organisation.
dbr:IBM_Simon rdf:type yago:Telephone .

SIEMENS DERI

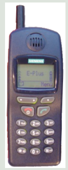
How to formulate the query? SPARQL!

"Companies who produce(d) telephones"

```
SELECT ?C {   ?C a dbo:Organisation.
              ?C foaf:made ?P .
              ?P a yago:Telephone .      }
```

- SPARQL http://www.w3.org/TR/sparql11-query/ :
  - "SQL Look-and-feel" query language for RDF
  - Basic Graph pattern matching
  - Complex patterns on top (UNION, FILTER, BIND, …)

dbr:Siemens a dbo:Organisation.
dbr:Siemens foaf:made dbr:Siemens_C25.
dbr:Siemens_C25 a yago:Telephone.



dbr:IBM foaf:made dbr:IBM_Simon.
dbr:IBM a dbo:Organisation.
dbr:IBM_Simon rdf:type yago:Telephone .

SIEMENS DERI

SPARQL – bare minimum overview:

- Basic Graph pattern matching

SELECT **?C** { **?C** a dbo:Organisation.
?C foaf:made **?P** .
?P a yago:Telephone . }

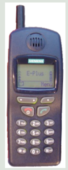
"Basic graph pattern" (BGP)

?C
dbr:Siemens
dbr:IBM

…in principle ≈ Conjunctive Query (CQ) answering over RDF

$$q(C) \leftarrow Organisation(C), made(C,P), Telephone(P)$$

dbr:Siemens a dbo:Organisation.
dbr:Siemens foaf:made **dbr:Siemens_C25**.
dbr:Siemens_C25 a yago:Telephone.



dbr:IBM foaf:made **dbr:IBM_Simon**.
dbr:IBM a dbo:Organisation.
dbr:IBM_Simon rdf:type yago:Telephone .

SIEMENS DERI

SPARQL – bare minimum overview:

- Complex patterns on top (**UNION**, FILTER, BIND, …)

SELECT **?C** { { **?C** foaf:made dbr:Siemens_C25 }

UNION

{ **?C** foaf:made dbr:IBM_Simon } }

?C
dbr:Siemens

U

?C
dbr:IBM

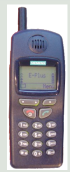
=

?C
dbr:Siemens
dbr:IBM

…in principle ≈ unions of CQs (UCQ)

$$q(C) \leftarrow made(C, siemens\_C25)$$
$$q(C) \leftarrow made(C, ibm\_Simon)$$

dbr:Siemens a dbo:Organisation.
dbr:Siemens foaf:made dbr:Siemens_C25.
dbr:Siemens_C25 a yago:Telephone.



dbr:IBM foaf:made dbr:IBM_Simon.
dbr:IBM a dbo:Organisation.
dbr:IBM_Simon rdf:type yago:Telephone .

SPARQL – bare minimum overview:

- Complex patterns on top (UNION, **FILTER,** BIND, …)

SELECT ?C { ?C a dbo:Organisation.
?C dbo:revenueEUR ?R .
FILTER (?R > 5E10) }

?C	?R
dbr:Siemens	7.829E10
dbr:SAP	1.622E10

?C
dbr:Siemens

… standard operators & functions to define FILTER constraints, cf. http://www.w3.org/TR/sparql11-query/#expressions

dbr:Siemens a dbo:Organisation.
dbr:Siemens dbo:revenueEUR 7.829E10

dbr:SAP a dbo:Organisation.
dbr:SAP dbo:revenueEUR 1.622E10

dbr:IBM a dbo:Organisation.
dbr:IBM dbo:revenueUSD 1.06916E11

SIEMENS DERI

SPARQL – bare minimum overview:

- Complex patterns on top (UNION, FILTER, **BIND**, …)

```
SELECT ?C ?R {    ?C a dbo:Organisation.
                        ?C dbo:revenueUSD ?RU . }
                  BIND ( ?RU / 1.3 AS ?R )   }
```

… compute new values

(same operators & functions as for FILTERs)

?C	?R
dbr:IBM	8.1363E10

dbr:Siemens a dbo:Organisation.
dbr:Siemens dbo:revenueEUR 7.829E10

dbr:SAP a dbo:Organisation.
dbr:SAP dbo:revenueEUR 1.622E10

1.3 **USD** = 1 **EUR**

dbr:IBM a dbo:Organisation.
dbr:IBM dbo:revenueUSD 1.06916E11

SIEMENS DERI

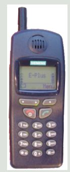
How to formulate the query?

"Companies who produce(d) telephones"

```
SELECT ?C {   ?C a dbo:Organisation.
              ?C foaf:made ?P .
              ?P a yago:Telephone .     }
```

More SPARQL Tutorials, cf. RW2011 (Birte Glimm), RW2012 (Jorge Perez) or also http://www.polleres.net/presentations/

dbr:Siemens a dbo:Organisation.
dbr:Siemens foaf:made dbr:Siemens_C25.
dbr:Siemens_C25 a yago:Telephone.



dbr:IBM foaf:made dbr:IBM_Simon.
dbr:IBM a dbo:Organisation.
dbr:IBM_Simon rdf:type yago:Telephone .

How to formulate the query?

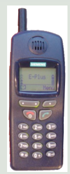
"Companies who produce(d) telephones"

SELECT ?C { ?C a dbo:Organisation.
?C foaf:made ?P .
?P a yago:Telephone . }

?

However, the data rather looks like that:

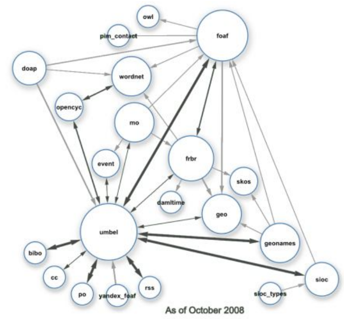
dbr:Siemens_C25 dbo:manufacturer dbr:Siemens .
dbr:Siemens_C25 a yago:SiemensMobilePhones .



dbr:IBM_Simon dbo:manufacturer dbr:IBM .
dbr:IBM_Simon rdf:type yago:IBMMobilePhones .

Make use of ontological inference! How?

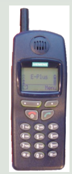
"Companies who produce(d) telephones"



SELECT ?C { ?C a dbo:Organisation.
?C foaf:made ?P .
?P a yago:Telephone . }

dbo:manufacturer rdfs:subPropertyOf foaf:maker.
foaf:maker owl:inverseOf foaf:made.
dbo:manufacturer rdfs:range dbo:Organisation.
yago:SiemensMobilePhones rdfs:subClassOf yago:Telephone .
yago:IBMMobilePhones rdfs:subClassOf yago:Telephone .

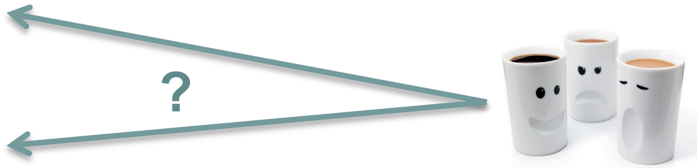
dbr:Siemens_C25 dbo:manufacturer dbr:Siemens .
dbr:Siemens_C25 a yago:SiemensMobilePhones .



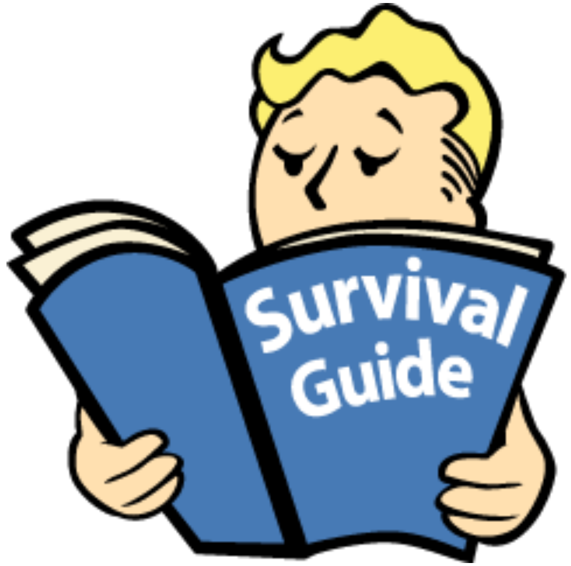
dbr:IBM_Simon dbo:manufacturer dbr:IBM .
dbr:IBM_Simon a yago:IBMMobilePhones .

Lecture Roadmap

- Scope/Motivation
  (Axel)
- **Short Introduction to RDFS+OWL**
  (Aidan)
- RDFS+OWL usage in Linked Data
  (Aidan)
- High-level Reasoning approaches: Query rewriting vs. Materialization
  (Axel)
- Challenges on Reasoning over Linked Data
  (Axel)



- Practical approaches for Reasoning over Linked Data
  - Quarantined & Authoritative Materialization *(Aidan)*
  - Link-Traversal Based Query Execution with Reasoning *(Aidan)*
  - Reasoning with Attribute Equations *(Axel)*
- Wrap-up/Outlook *(all)*



RDFS/OWL SURVIVAL GUIDE

RDFS and OWL!

***RDF S**chema*

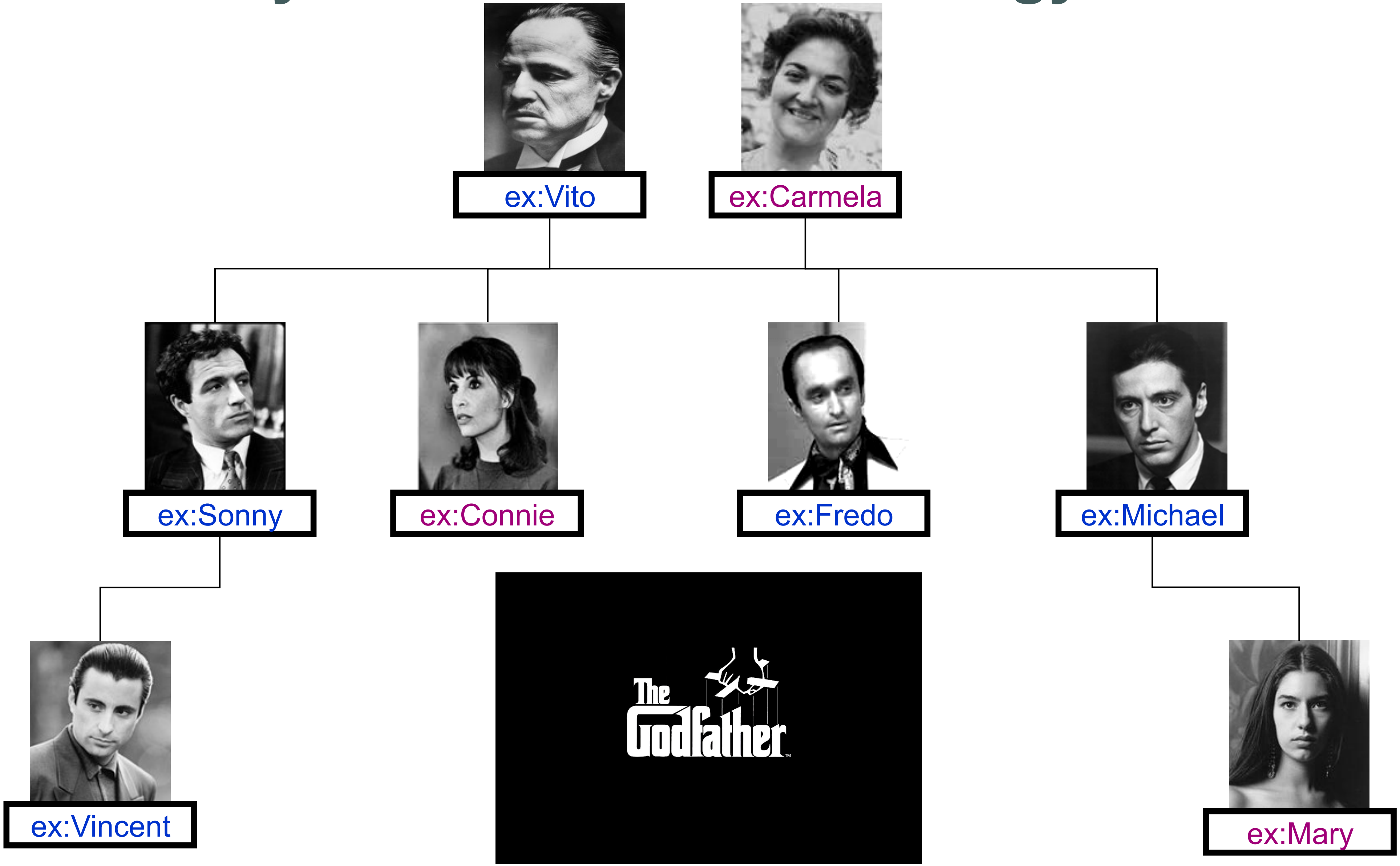
***W**eb **O**ntology **L**anguage*

- W3C Recommendations, 2004
  - OWL 2 since 2009
- Standardised schema/ontology languages
  - Can be serialised in RDF
  - OWL (partly) extends RDFS
    - (Saying “OWL” often covers “RDFS”)

Features walkthrough…

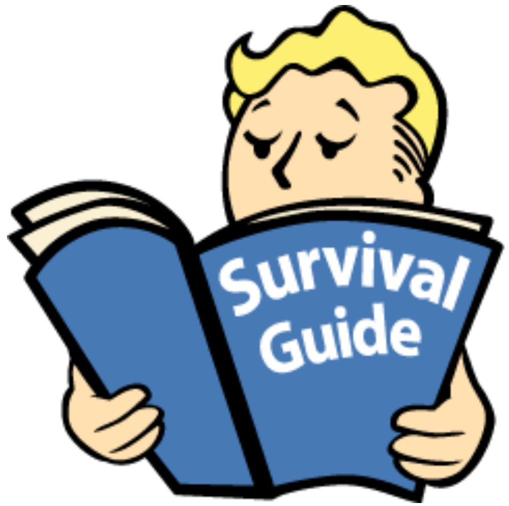
…modelling family relationships in RDFS and OWL…

A Family-Relations OWL Ontology

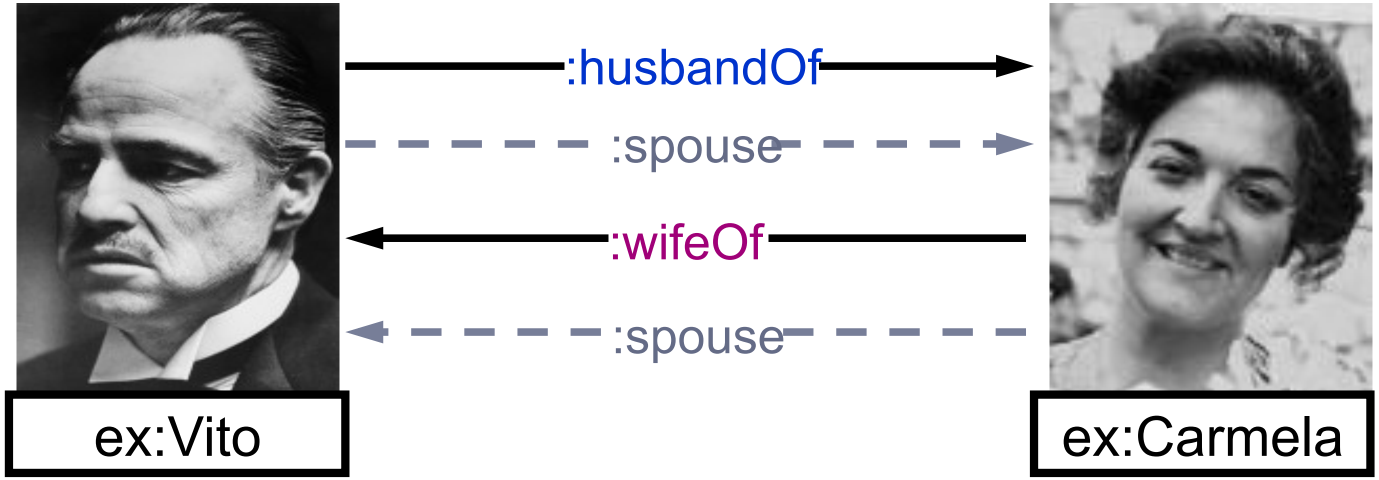


RDFS

1. `rdfs:subPropertyOf`
2. `rdfs:subClassOf`
3. `rdfs:domain`
4. `rdfs:range`



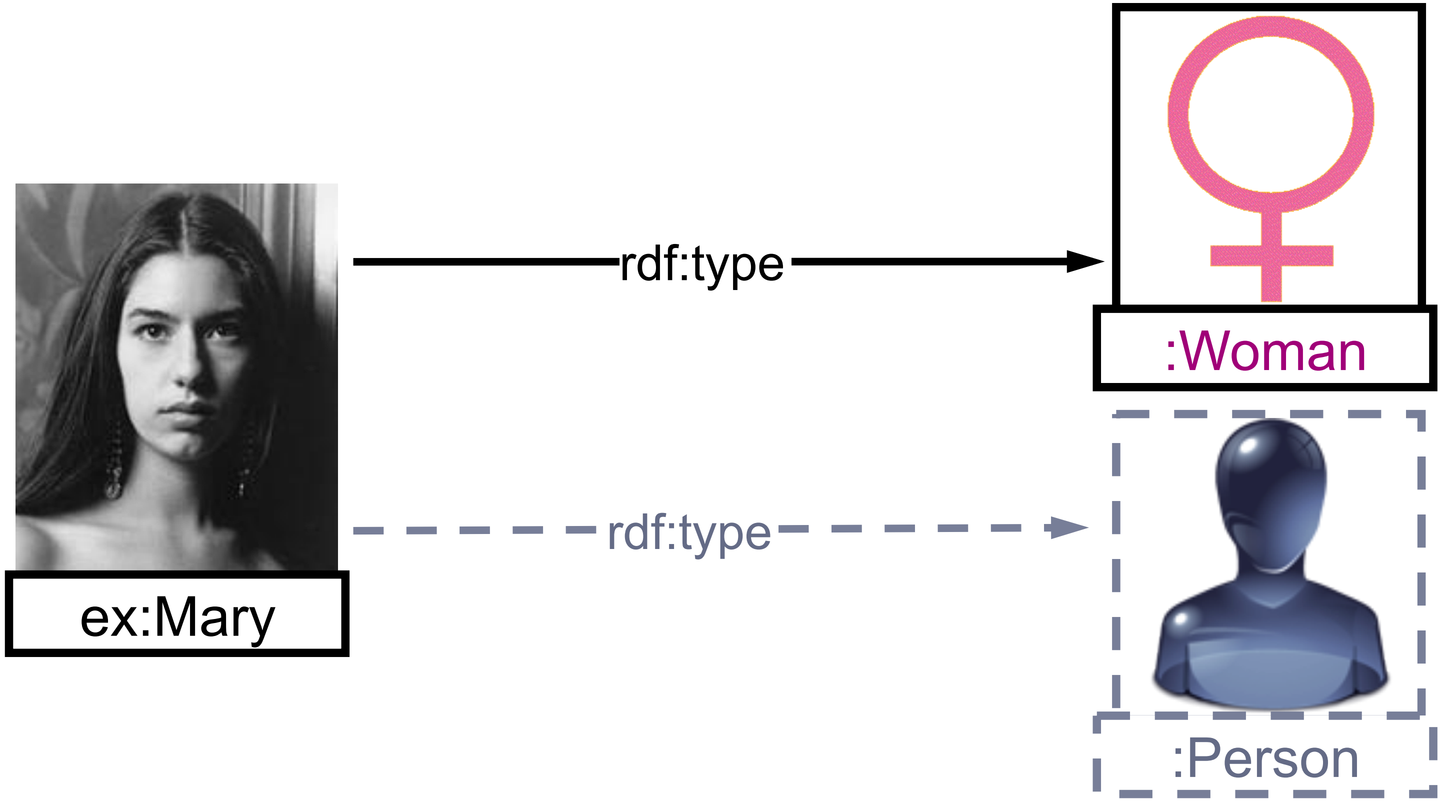
rdfs:subPropertyOf




```
ex:Vito :husbandOf ex:Carmela .
:husbandOf rdfs:subPropertyOf :spouse .
⇒ ex:Vito :spouse ex:Carmela .

ex:Carmela :wifeOf ex:Vito .
:wifeOf rdfs:subPropertyOf :spouse .
⇒ ex:Carmela :spouse ex:Vito .
```

rdfs:subClassOf

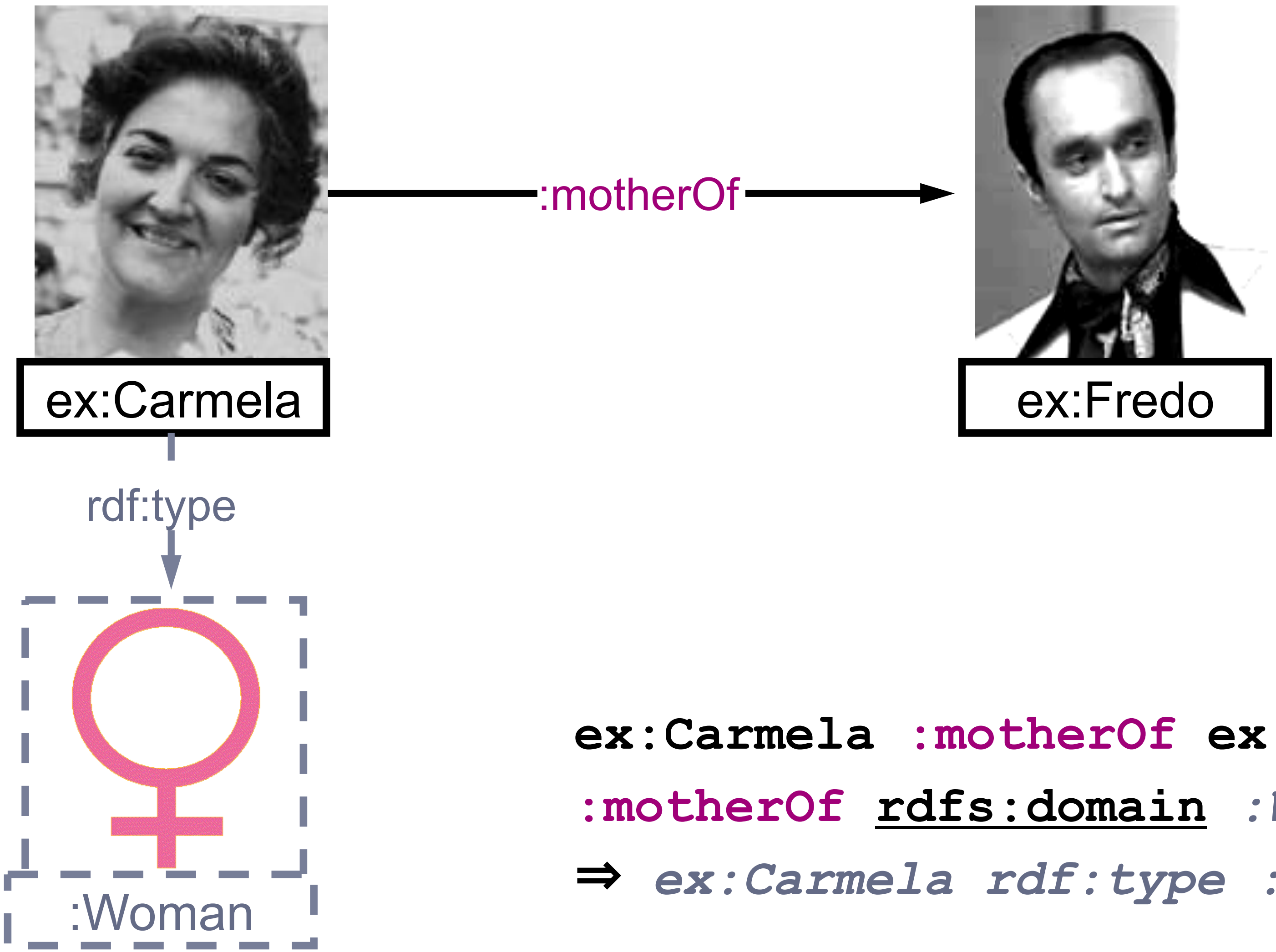


ex:Mary rdf:type :Woman .

:Woman <u>rdfs:subClassOf</u> *:Person* .

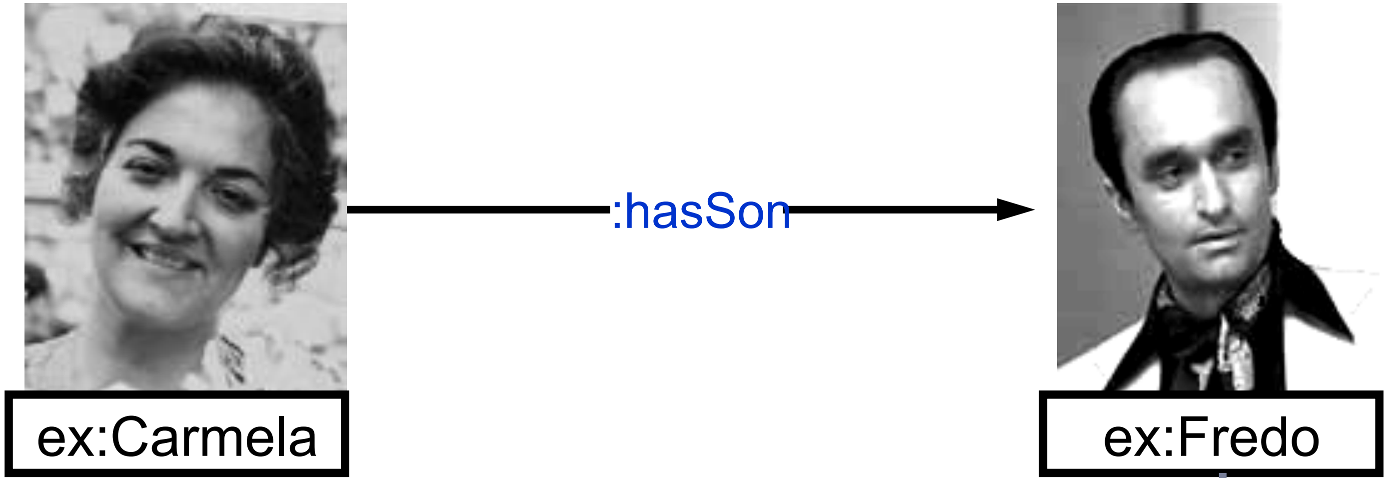
⇒ ***ex:Mary rdf:type :Person .***

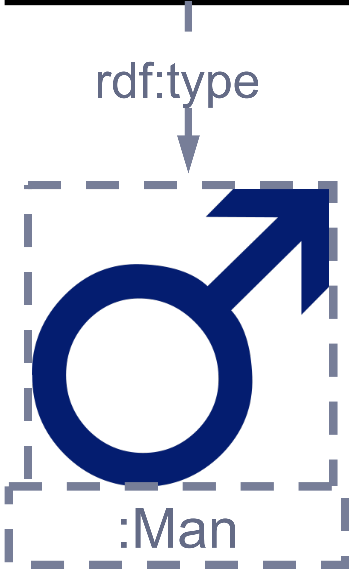
rdfs:domain




```
ex:Carmela :motherOf ex:Fredo .
:motherOf rdfs:domain :Woman .
⇒ ex:Carmela rdf:type :Woman .
```

rdfs:range





ex:Carmela :hasSon ex:Fredo .

:hasSon rdfs:range *:Man* **.**

⇒ *ex:Fredo rdf:type :Man .*

Recap RDFS

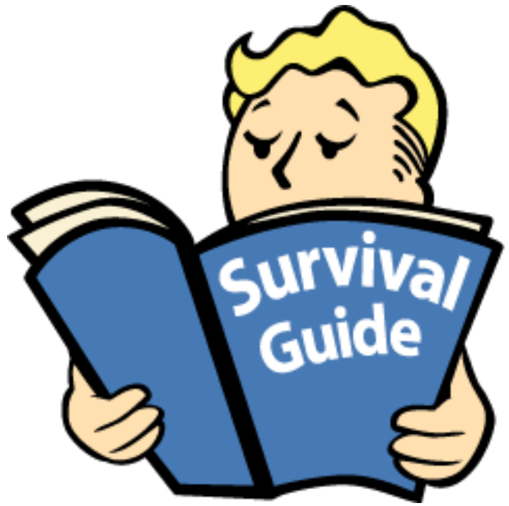
- What would be the **rdfs:domain** of the property **:fatherOf**?
- What would be the **rdfs:range** of the property **:hasSister**?

RDFS

1. `rdfs:subPropertyOf`
2. `rdfs:subClassOf`
3. `rdfs:domain`
4. `rdfs:range`

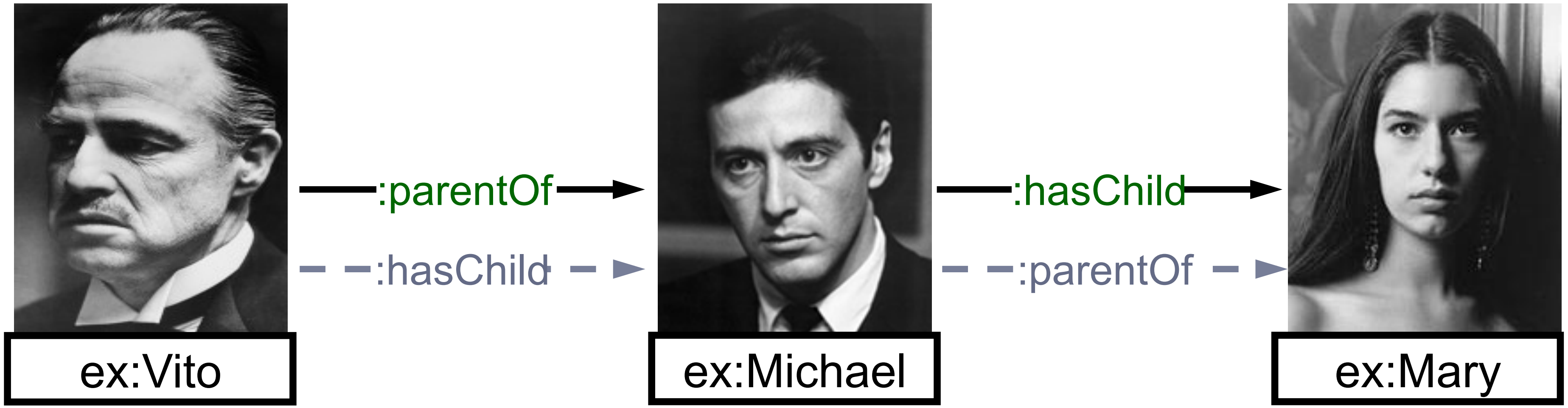
OWL

1. **Property Axioms**
   a. `owl:equivalentProperty`
   b. `owl:inverseOf`
   c. `owl:SymmetricProperty`
   d. `owl:TransitiveProperty`
   e. `owl:propertyChainAxiom`



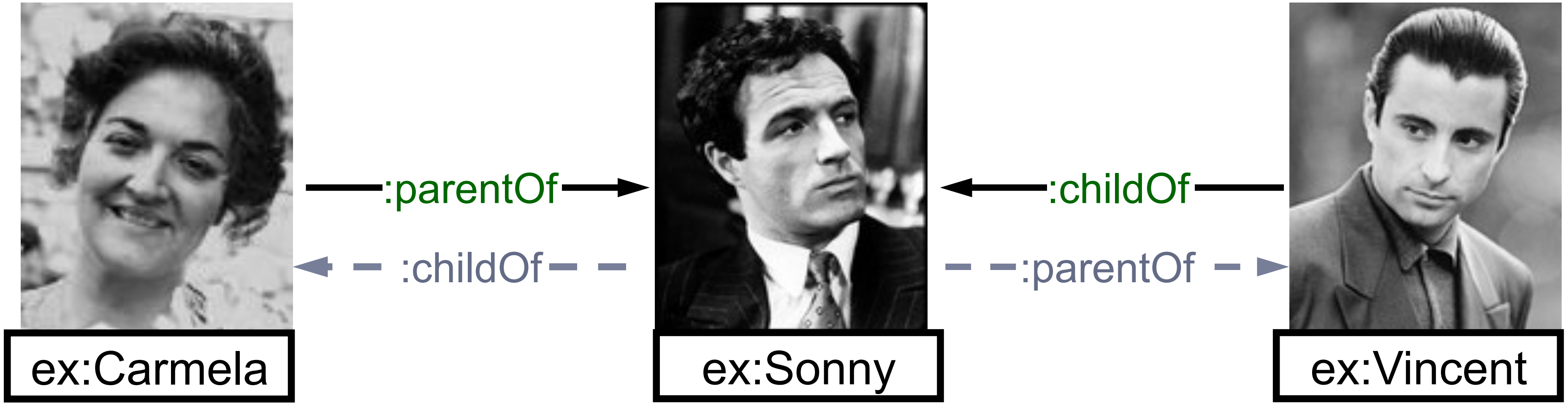
SIEMENS DERI

owl:equivalentProperty



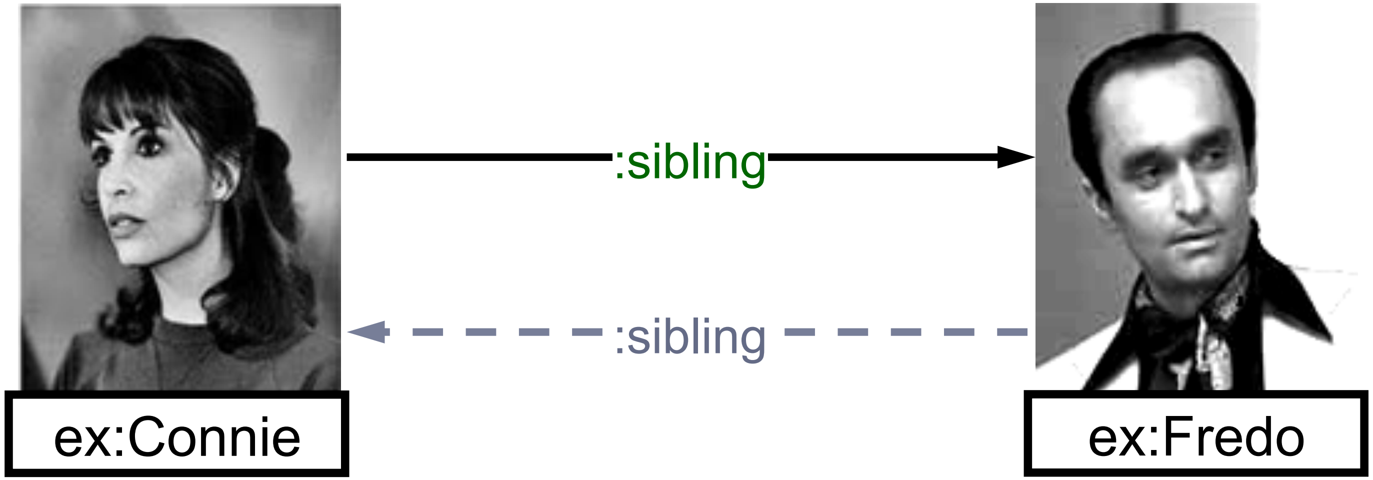

```
ex:Vito :parentOf ex:Michael .
ex:Michael :hasChild ex:Mary .
:parentOf owl:equivalentProperty :hasChild .
⇒ ex:Vito :hasChild ex:Vincent .
⇒ ex:Michael :parentOf ex:Mary .
```

owl:inverseOf



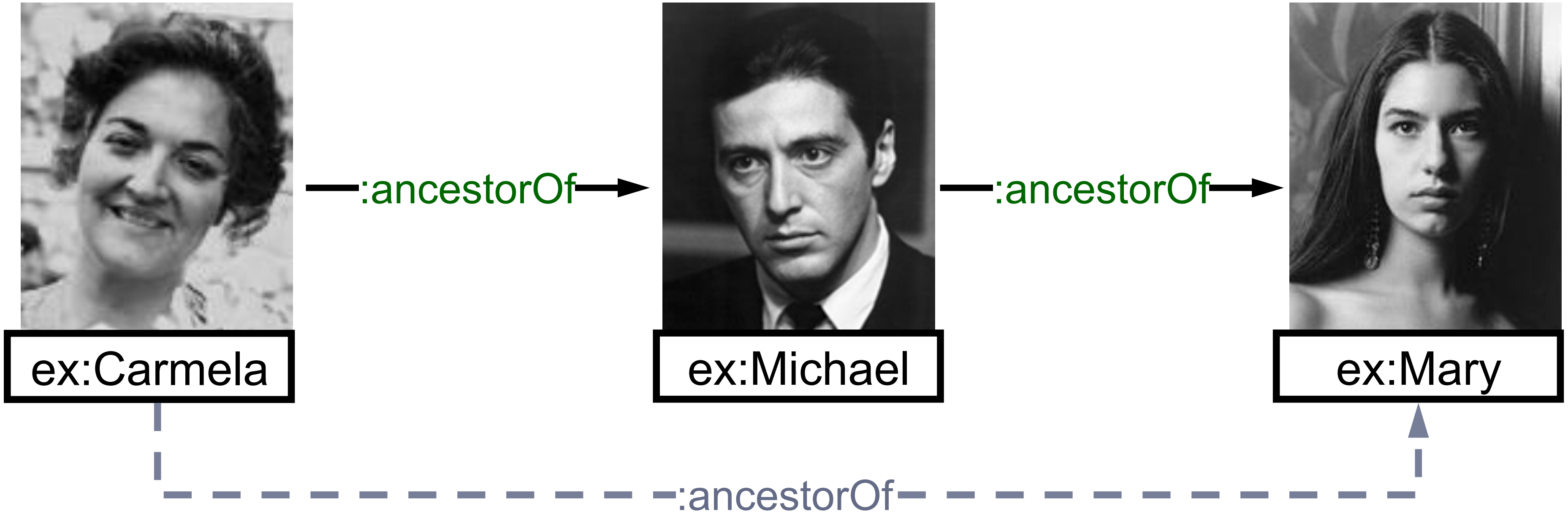

```
ex:Carmela :parentOf ex:Sonny .
ex:Vincent :childOf ex:Sonny .
:parentOf owl:inverseOf :childOf .
⇒ ex:Sonny :parentOf ex:Vincent .
⇒ ex:Sonny :childOf ex:Carmela .
```

owl:SymmetricProperty




```
ex:Connie :sibling ex:Fredo .
:sibling rdf:type owl:SymmetricProperty .
⇒ ex:Fredo :sibling ex:Connie .
```

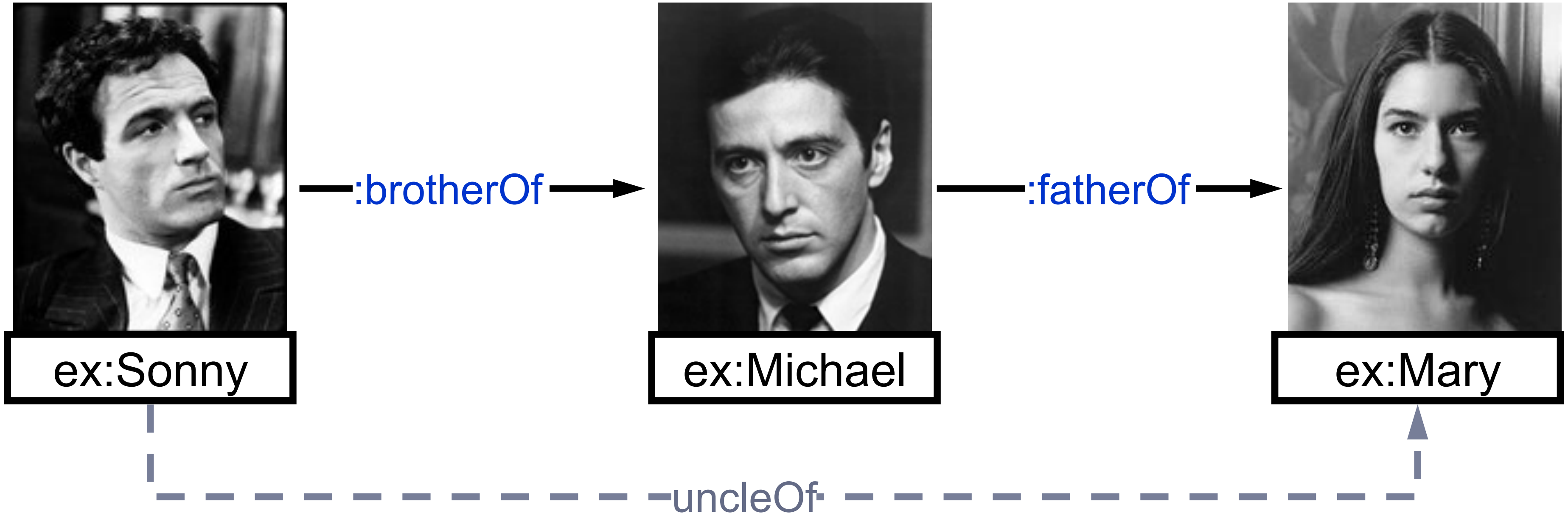
owl:TransitiveProperty




```
ex:Carmela :ancestorOf ex:Michael .
ex:Michael :ancestorOf ex:Mary .
:ancestorOf rdf:type owl:TransitiveProperty .
⇒ ex:Carmela :ancestorOf ex:Mary .
```


SIEMENS DERI

owl:propertyChainAxiom




```
ex:Sonny :brotherOf ex:Michael .
ex:Michael :fatherOf ex:Mary .
:uncleOf owl:propertyChainAxiom (:brotherOf :fatherOf) .
⇒ ex:Sonny :uncleOf ex:Mary .
```

Recap OWL property axioms

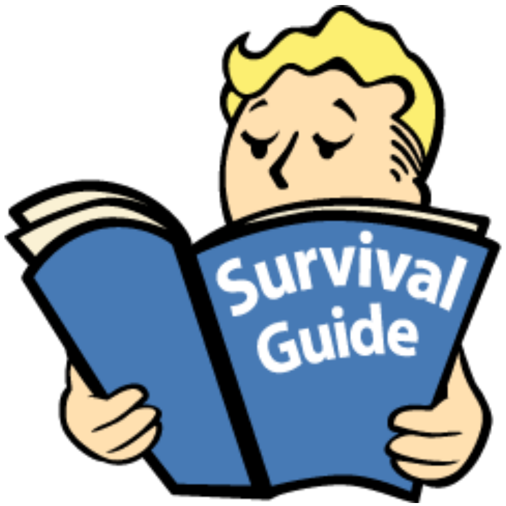
- What would be the **owl:inverseOf** of the property **:childOf**?
- Name an **owl:SymmetricProperty** to do with family relations?
- Name an **owl:TransitiveProperty** to do with family relations?
- Give a **owl:propertyChainAxiom** for :**hasNiece?**

RDFS

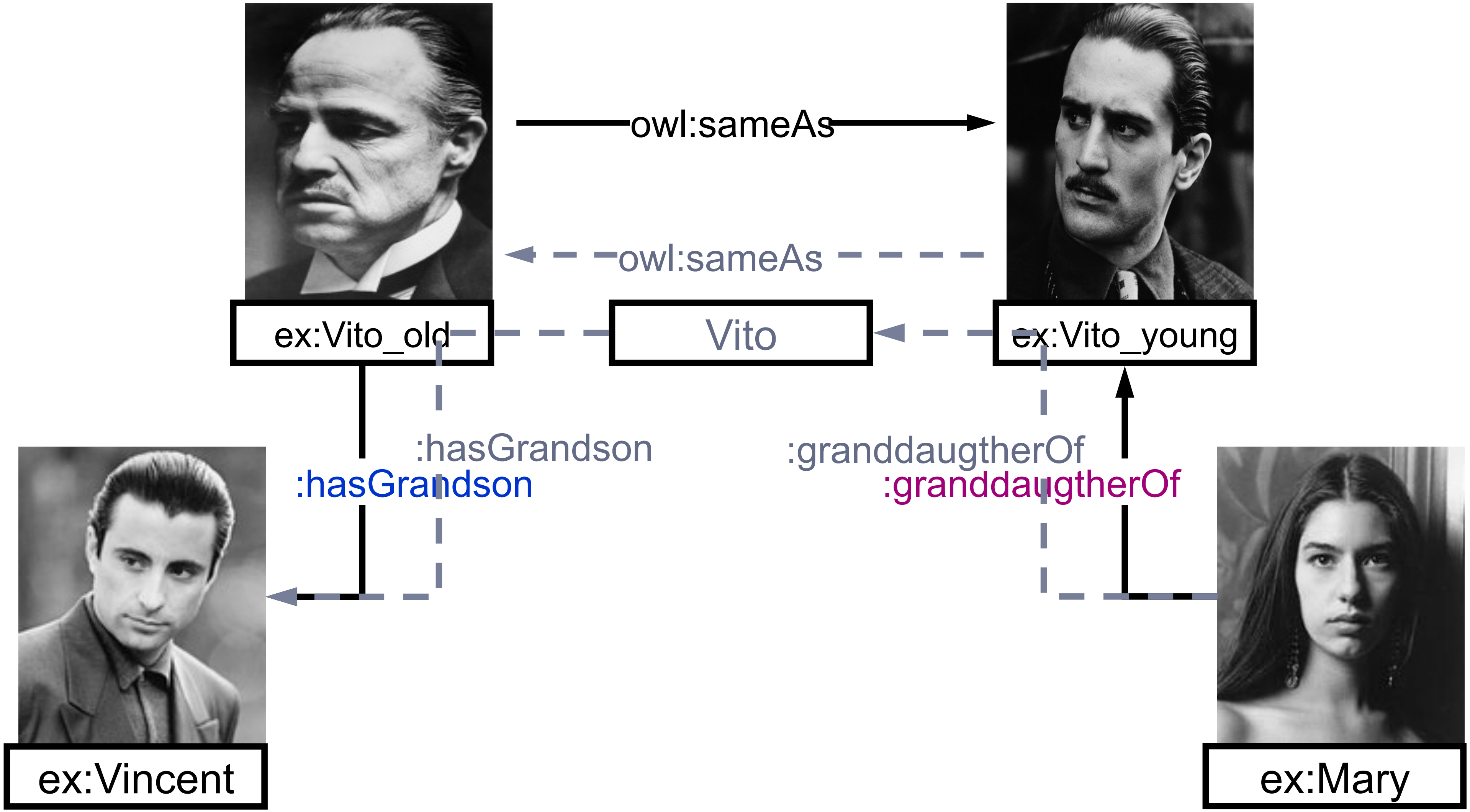
1. rdfs:subPropertyOf
2. rdfs:subClassOf
3. rdfs:domain
4. rdfs:range

OWL

1. Property Axioms
2. **Equality**
   a. **owl:sameAs**
   b. **owl:FunctionalProperty**
   c. **owl:InverseFunctionalProperty**

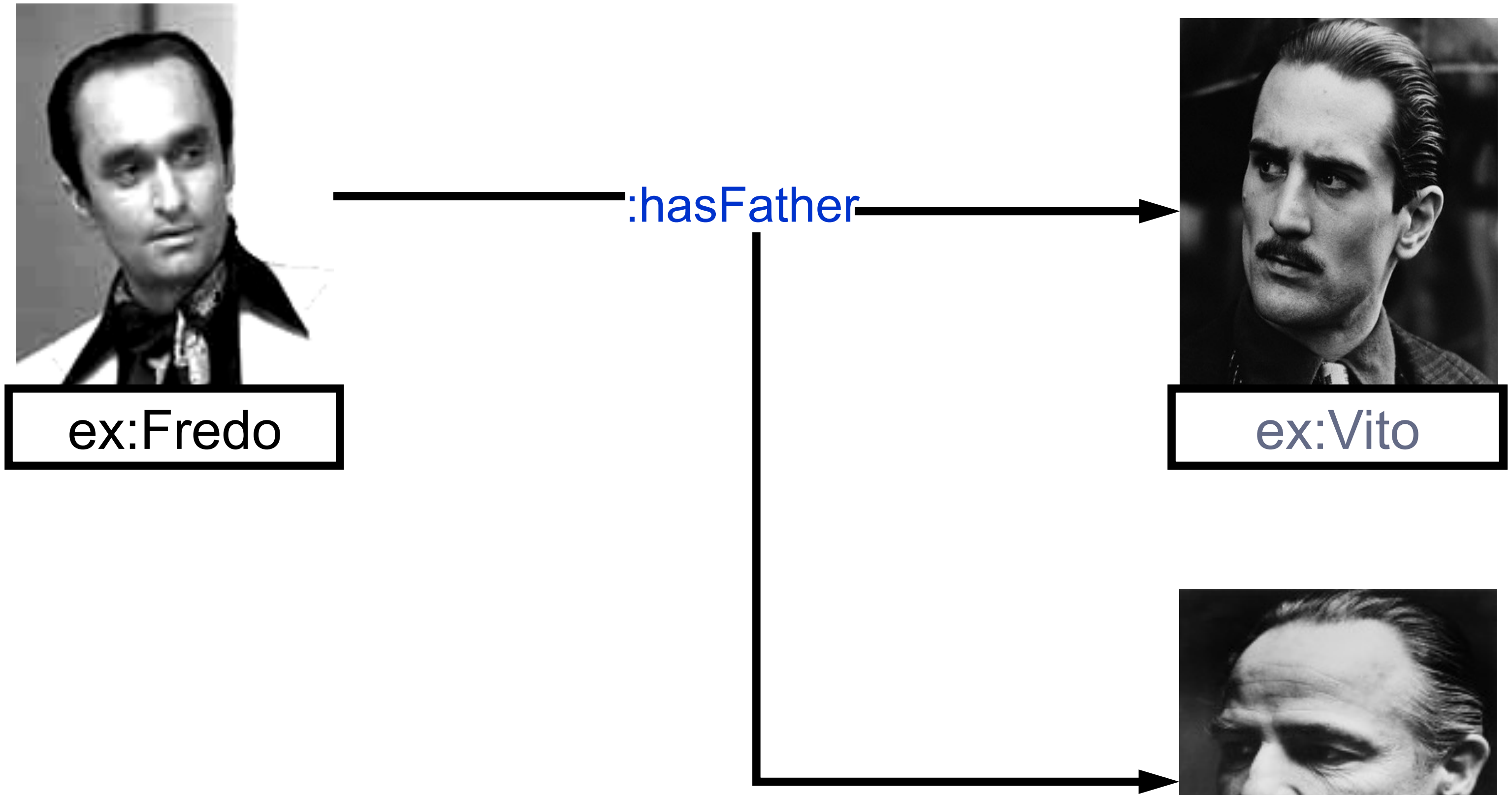


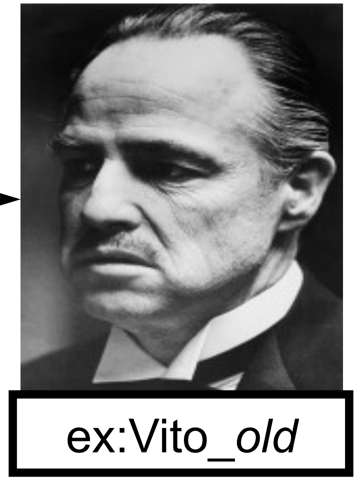
owl:sameAs



ex:Vito_old owl:sameAs ex:Vito_young .

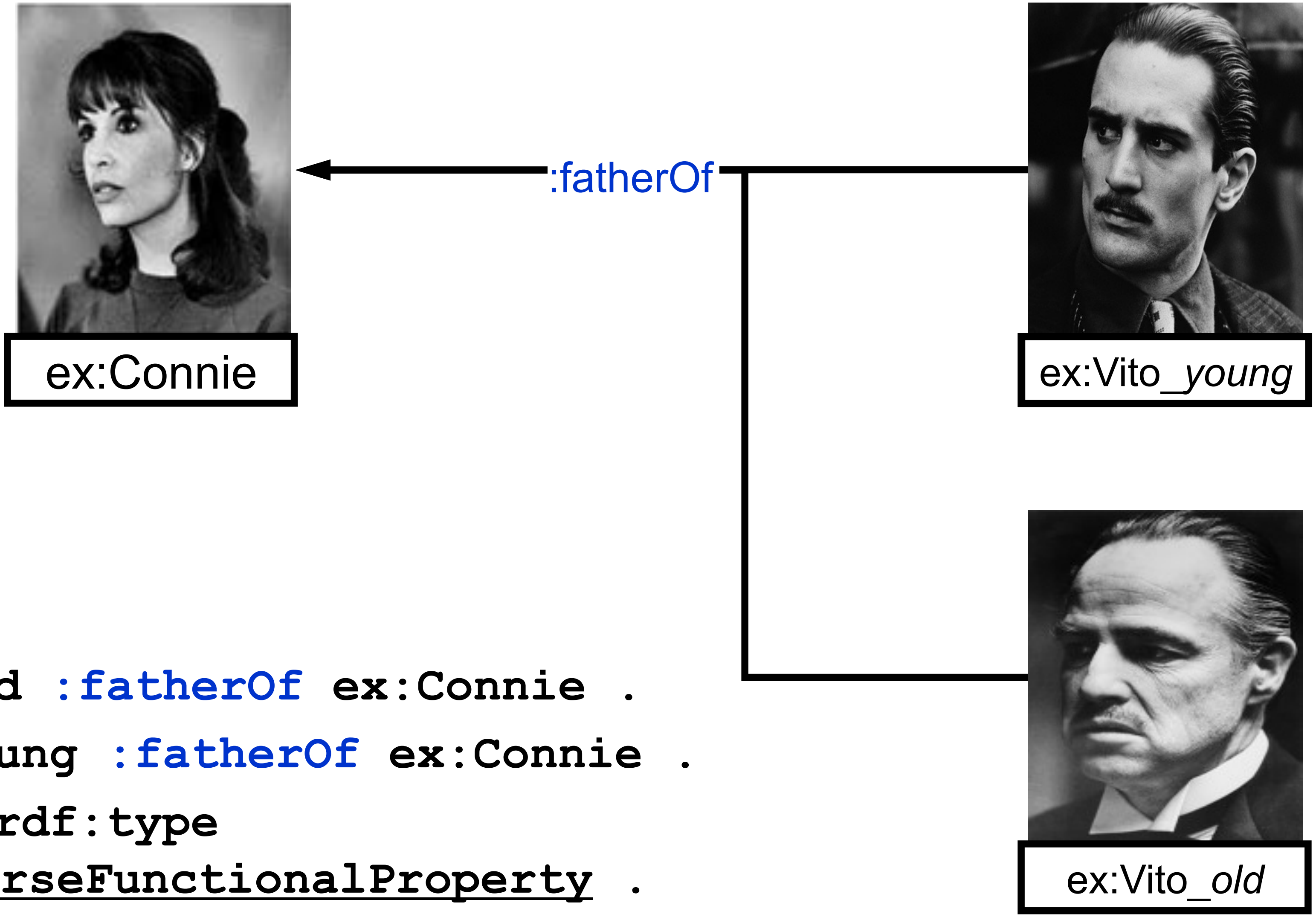
owl:FunctionalProperty






```
ex:Fredo :hasFather ex:Vito_old .
ex:Fredo :hasFather ex:Vito_young .
:hasFather rdf:type owl:FunctionalProperty .
⇒ ex:Vito_old owl:sameAs ex:Vito_young .
```

owl:InverseFunctionalProperty



ex:Vito_old :fatherOf ex:Connie .

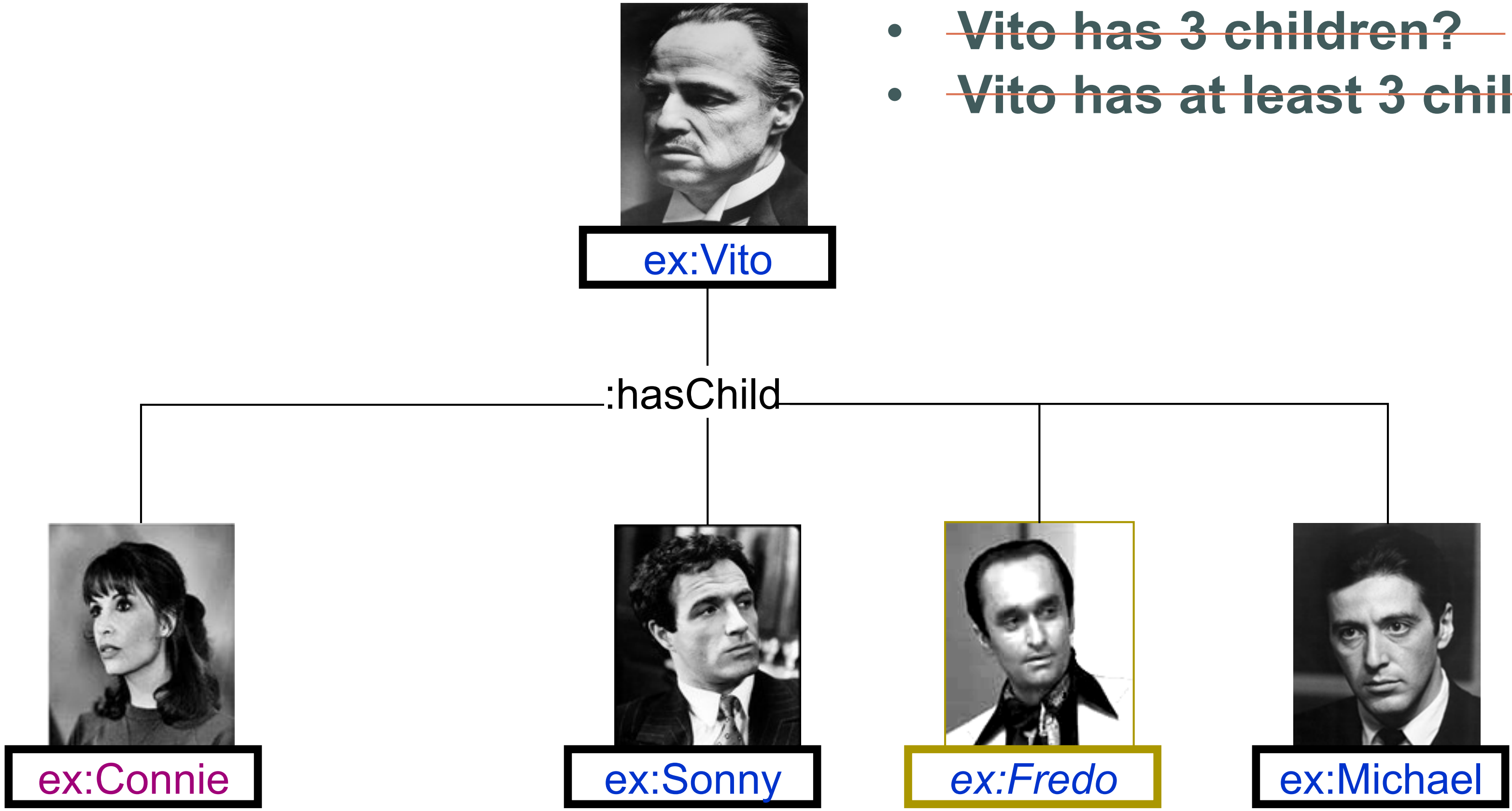
ex:Vito_young :fatherOf ex:Connie .

:fatherOf rdf:type
owl:InverseFunctionalProperty .

⇒ ***ex:Vito_old owl:sameAs ex:Vito_young .***

… Open World Assumption (OWA)

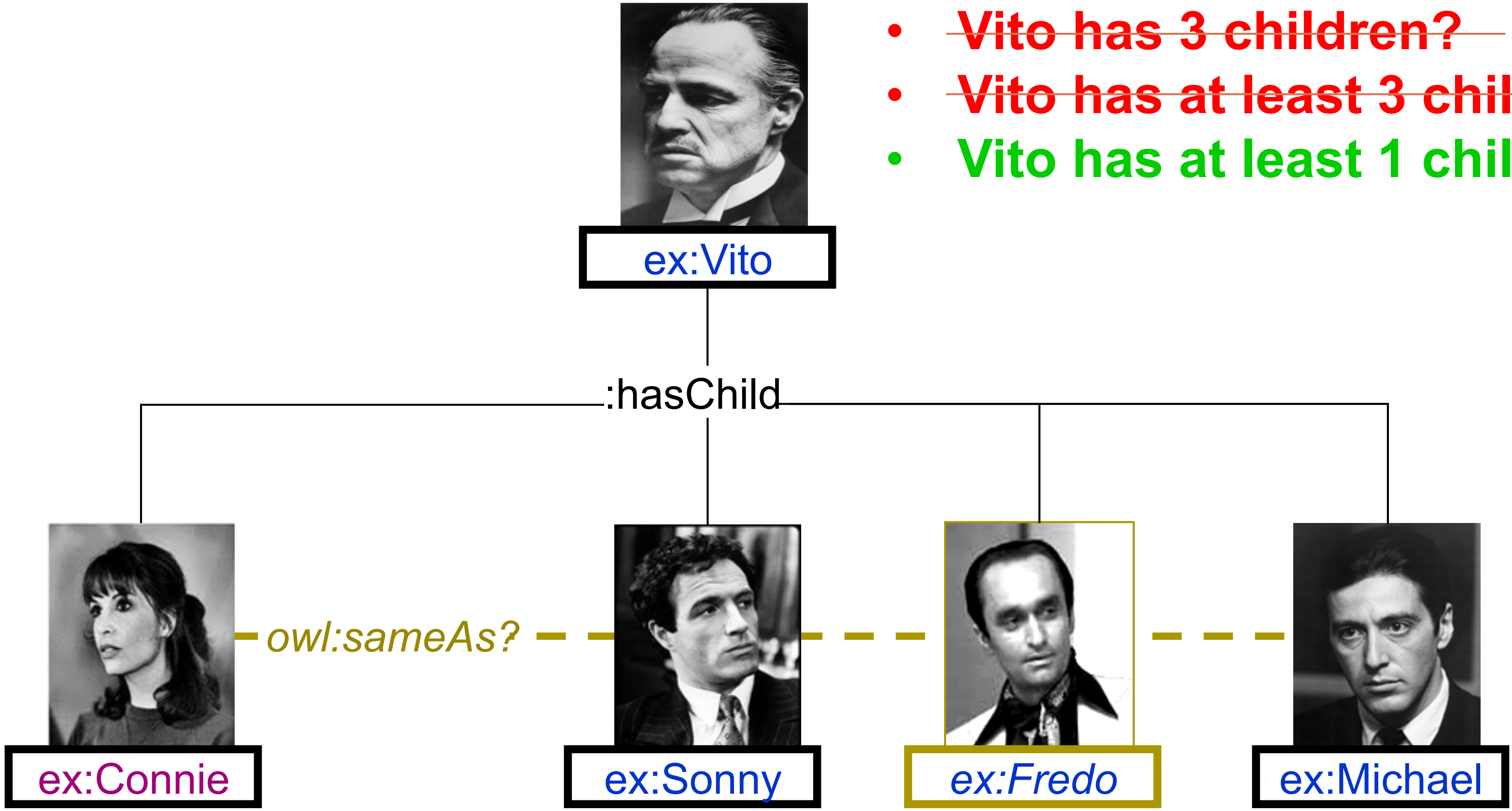
- ~~Vito has 3 children?~~
- ~~Vito has at least 3 children?~~




```
ex:Vito :hasChild ex:Connie, ex:Sonny, ex:Michael .
ex:Vito :hasChild ex:Fredo .
... ?
```

… No Unique Name Assumption (UNA)

- ~~Vito has 3 children?~~
- ~~Vito has at least 3 children?~~
- Vito has at least 1 child!




```
ex:Vito :hasChild ex:Connie, ex:Sonny, ex:Michael .
ex:Vito :hasChild ex:Fredo .
... ?
```

Recap OWL equality axioms

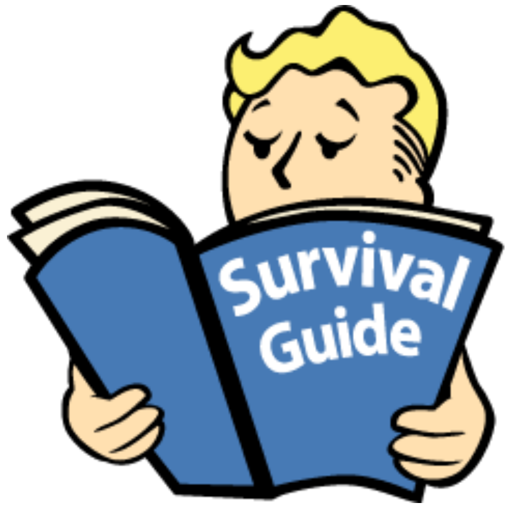
- Name an **owl:FunctionalProperty** to do with family relations?
- Name a similar **owl:InverseFunctionalProperty**?

RDFS

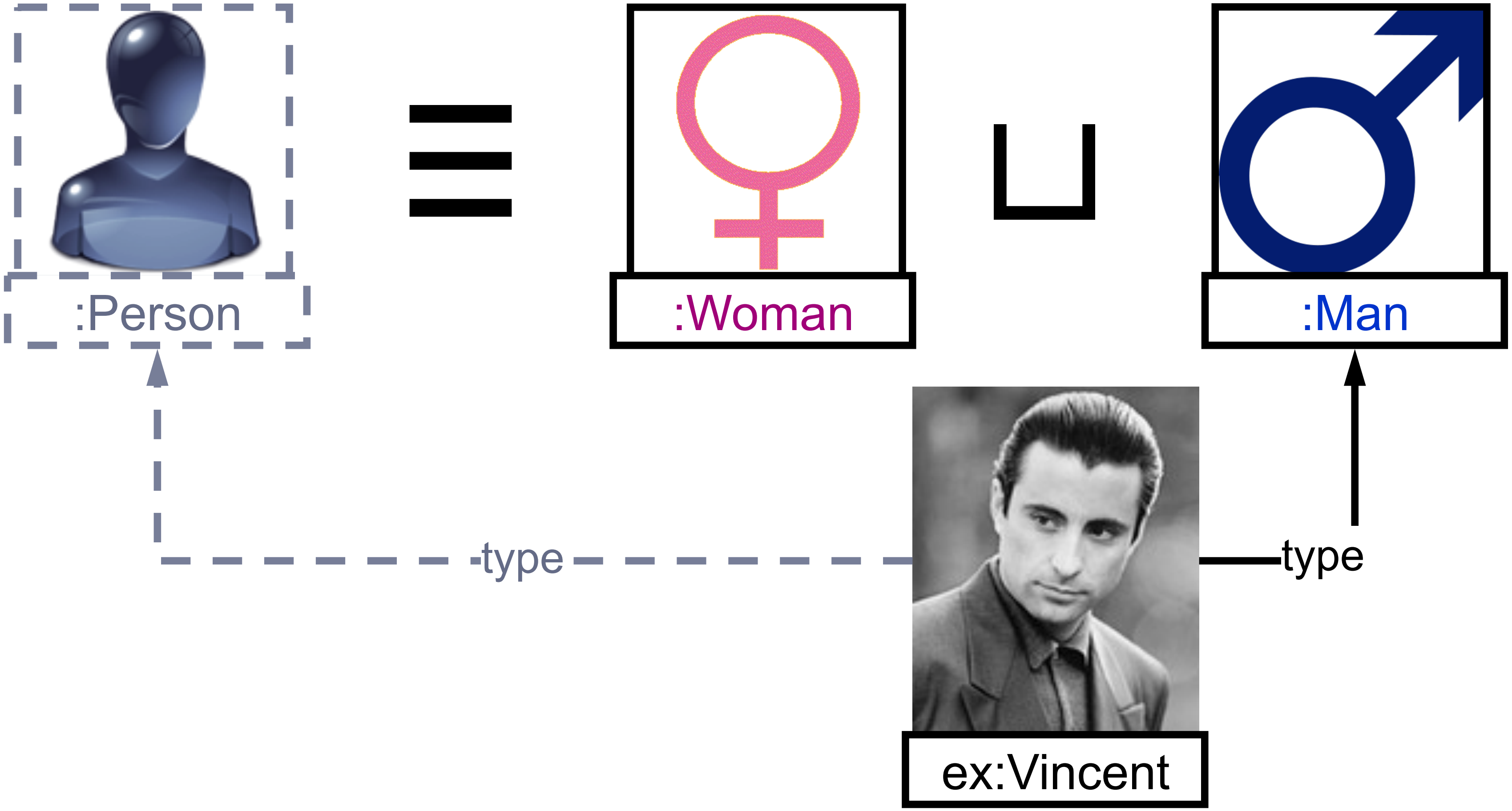
1. rdfs:subPropertyOf
2. rdfs:subClassOf
3. rdfs:domain
4. rdfs:range

OWL

1. Property Axioms
2. Equality
3. **Class Axioms**
   a. owl:unionOf
   b. owl:intersectionOf
   c. owl:oneOf
   d. owl:allValuesFrom
   e. owl:someValuesFrom



owl:unionOf

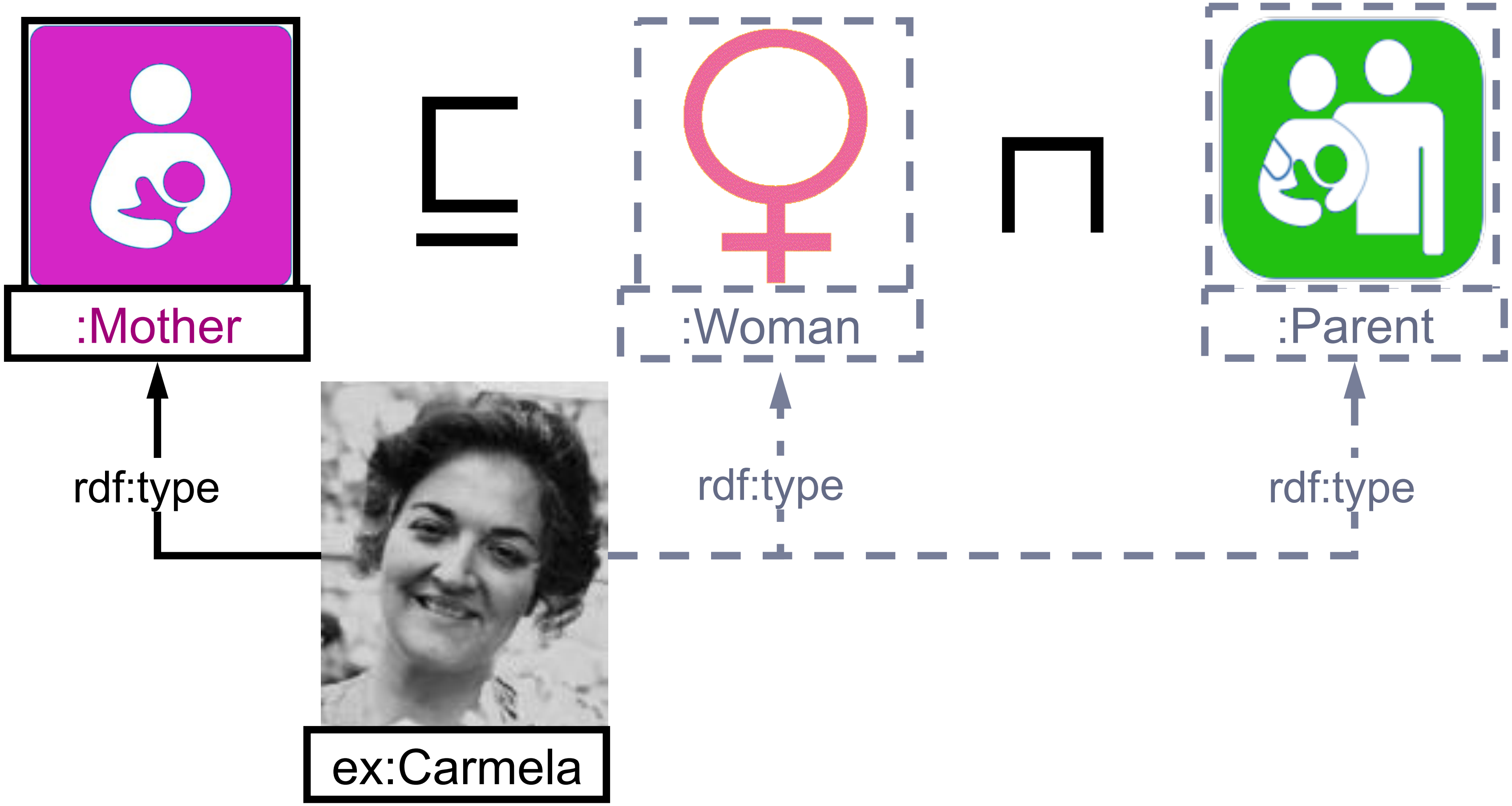


ex:Vincent rdf:type :Man .

:Person owl:equivalentClass [owl:unionOf (:Woman :Man)]

⇒ ***ex:Vincent rdf:type :Person .***

owl:intersectionOf (I)

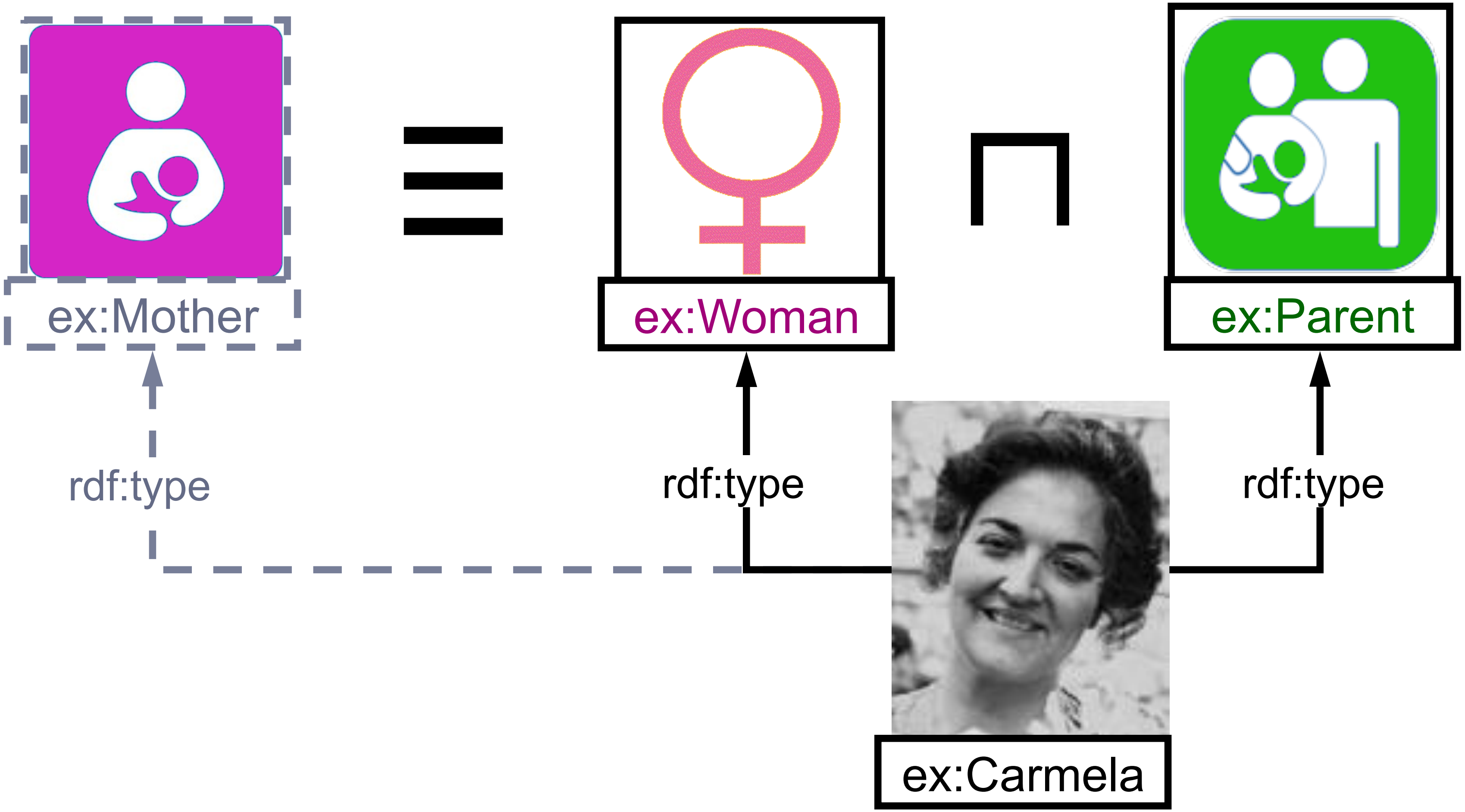


ex:Carmela rdf:type :Mother .

:Mother rdfs:subClassOf [owl:intersectionOf (*:Woman :Parent*)]

⇒ ***ex:Carmela rdf:type :Woman , :Parent .***

owl:intersectionOf (II)

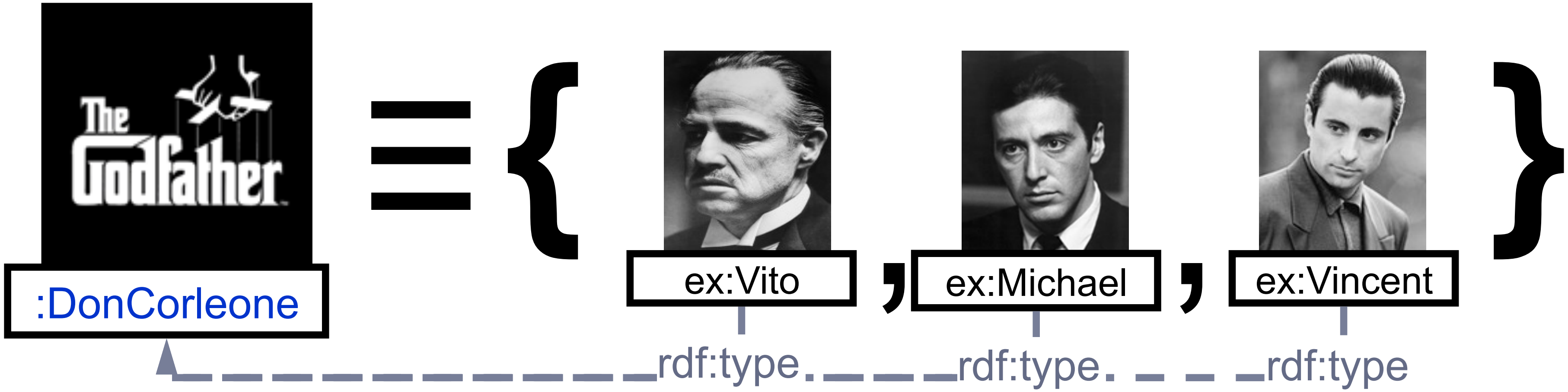


ex:Carmela rdf:type :Woman , :Parent .

:Mother owl:equivalentClass [owl:intersectionOf (:Woman :Parent)]

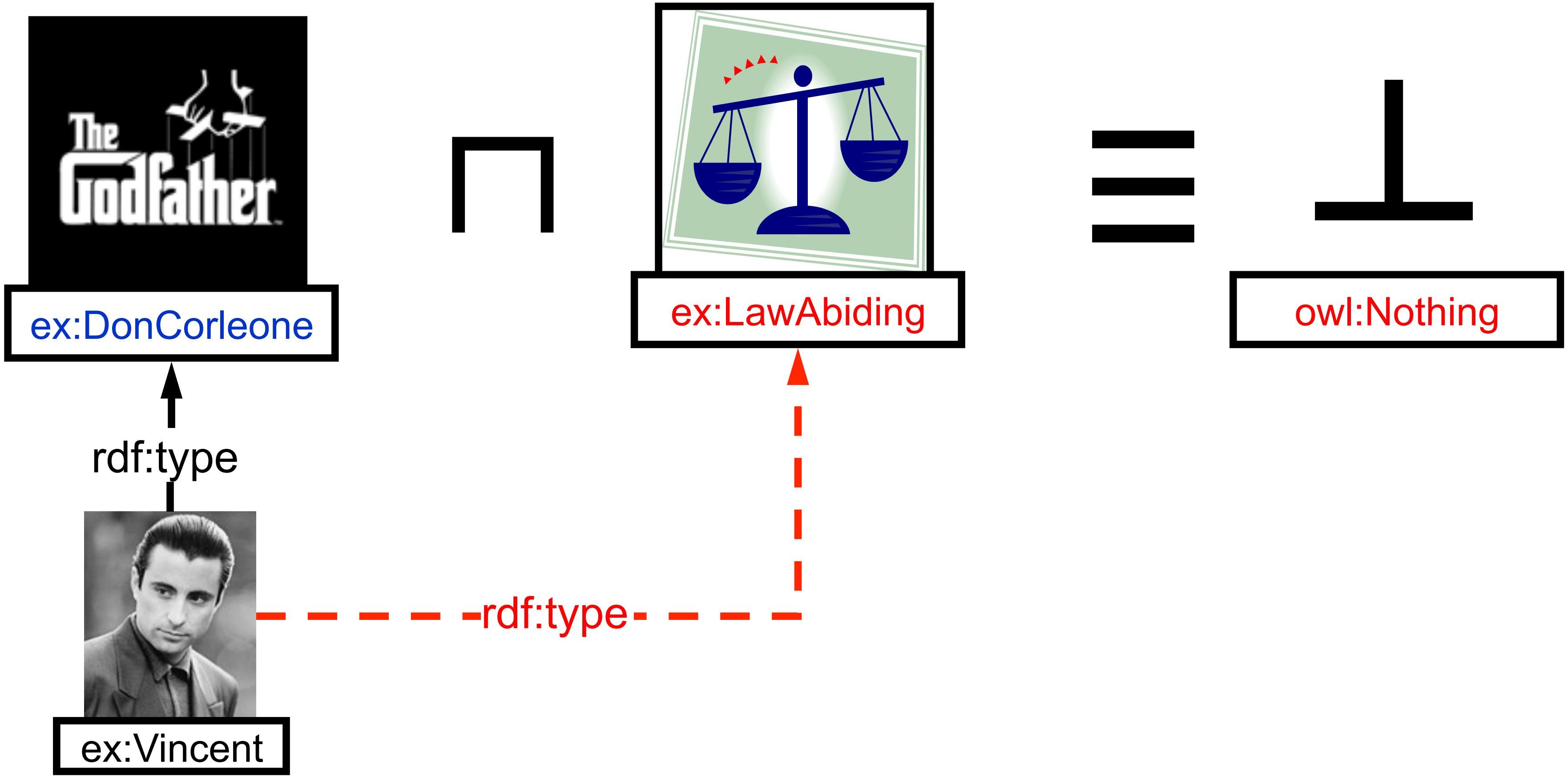
⇒ ***ex:Carmela rdf:type :Mother .***

owl:oneOf



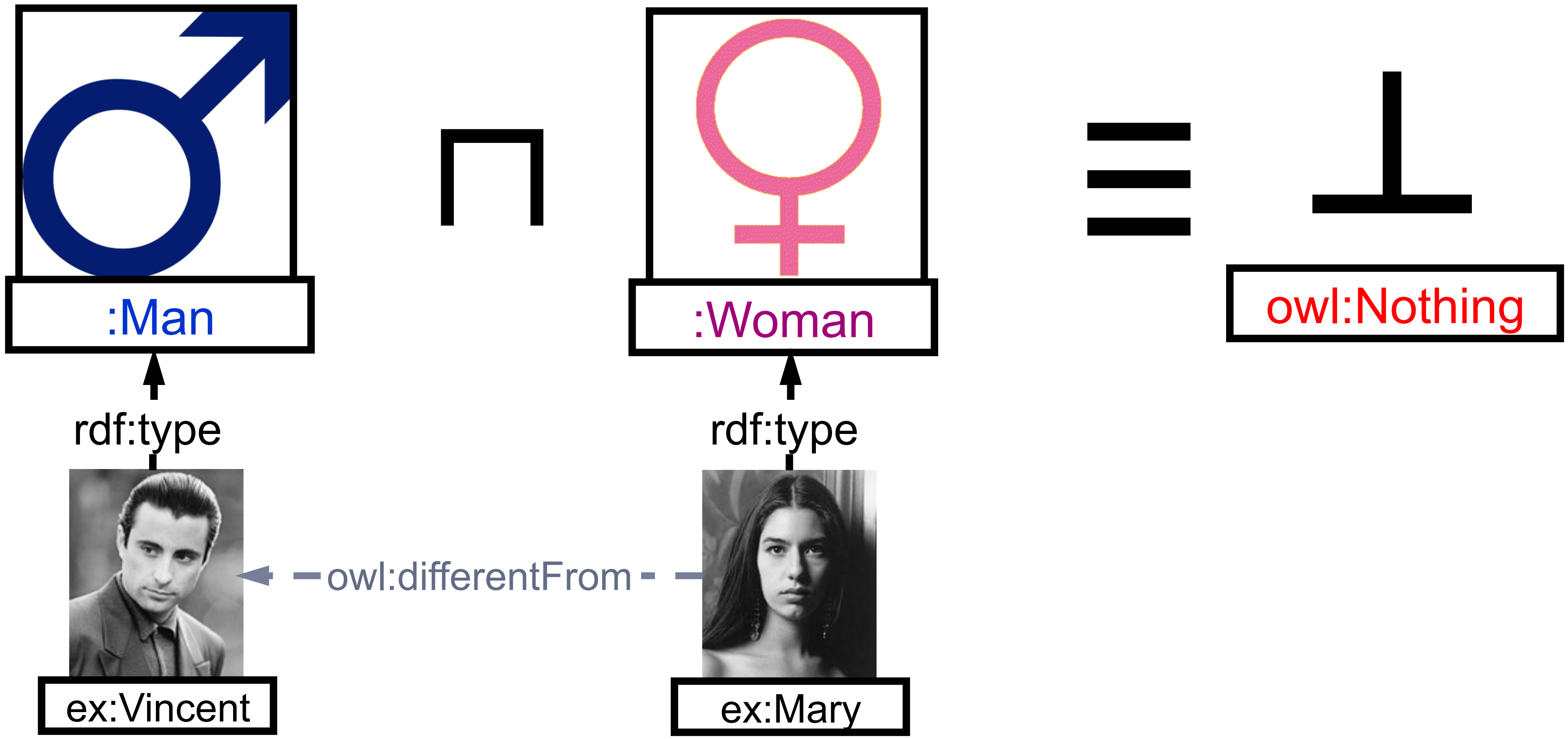

```
:DonCorleone owl:equivalentClass
   [ owl:oneOf (ex:Vito ex:Michael ex:Vincent) ]
⇒ ex:Vito rdf:type :DonCorleone .
⇒ ex:Michael rdf:type :DonCorleone .
⇒ ex:Vincent rdf:type :DonCorleone .
```

owl:disjointWith (I)




```
ex:Michael rdf:type ex:DonCorleone .
ex:DonCorleone owl:disjointWith ex:LawAbiding .
ex:Michael rdf:type ex:LawAbiding .
```

owl:disjointWith (II)



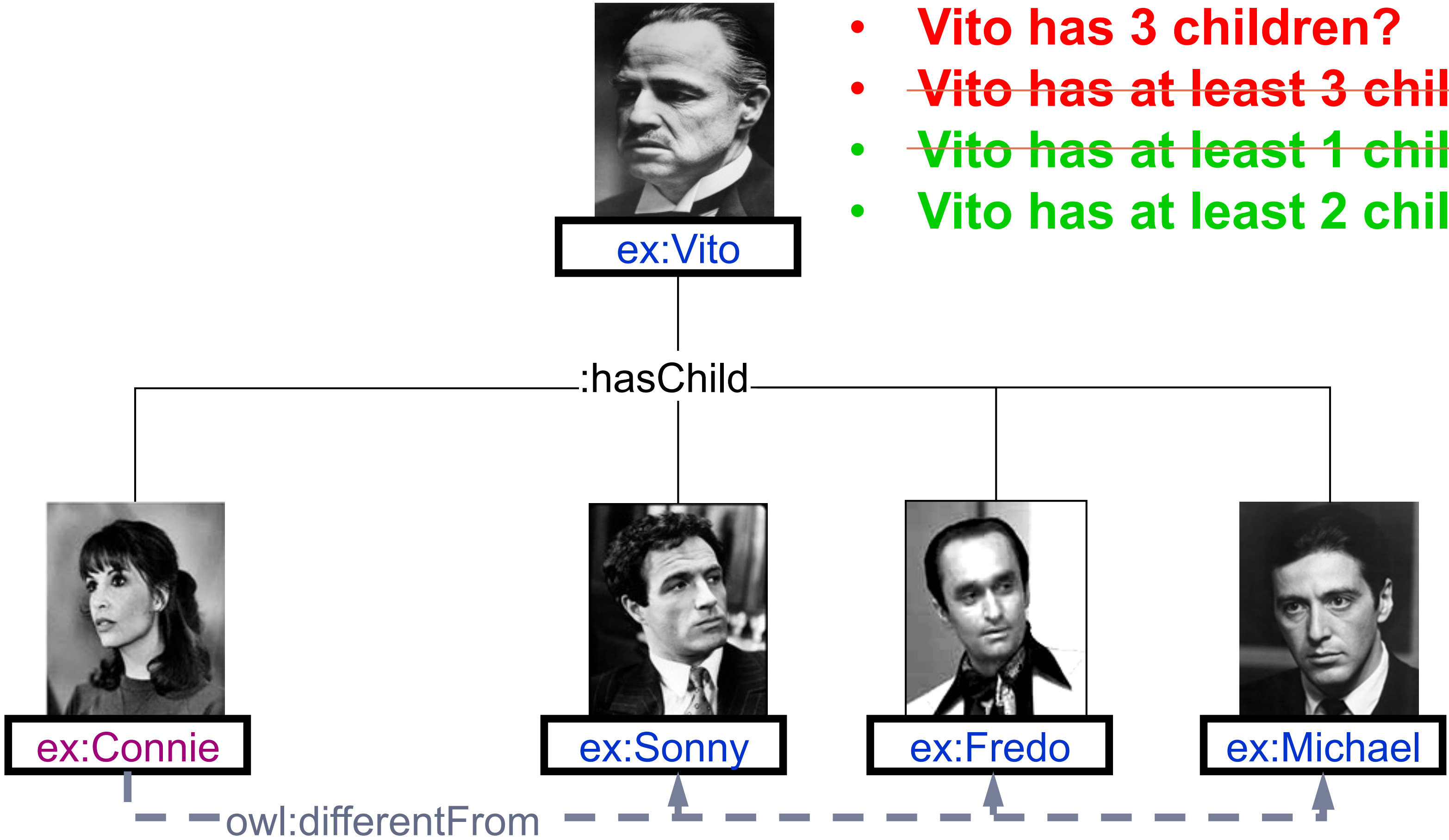
ex:Michael rdf:type :Man . ex:Mary rdf:type :Woman .

:Man owl:disjointWith :Woman .

⇒ ***ex:Mary owl:differentFrom ex:Michael .***

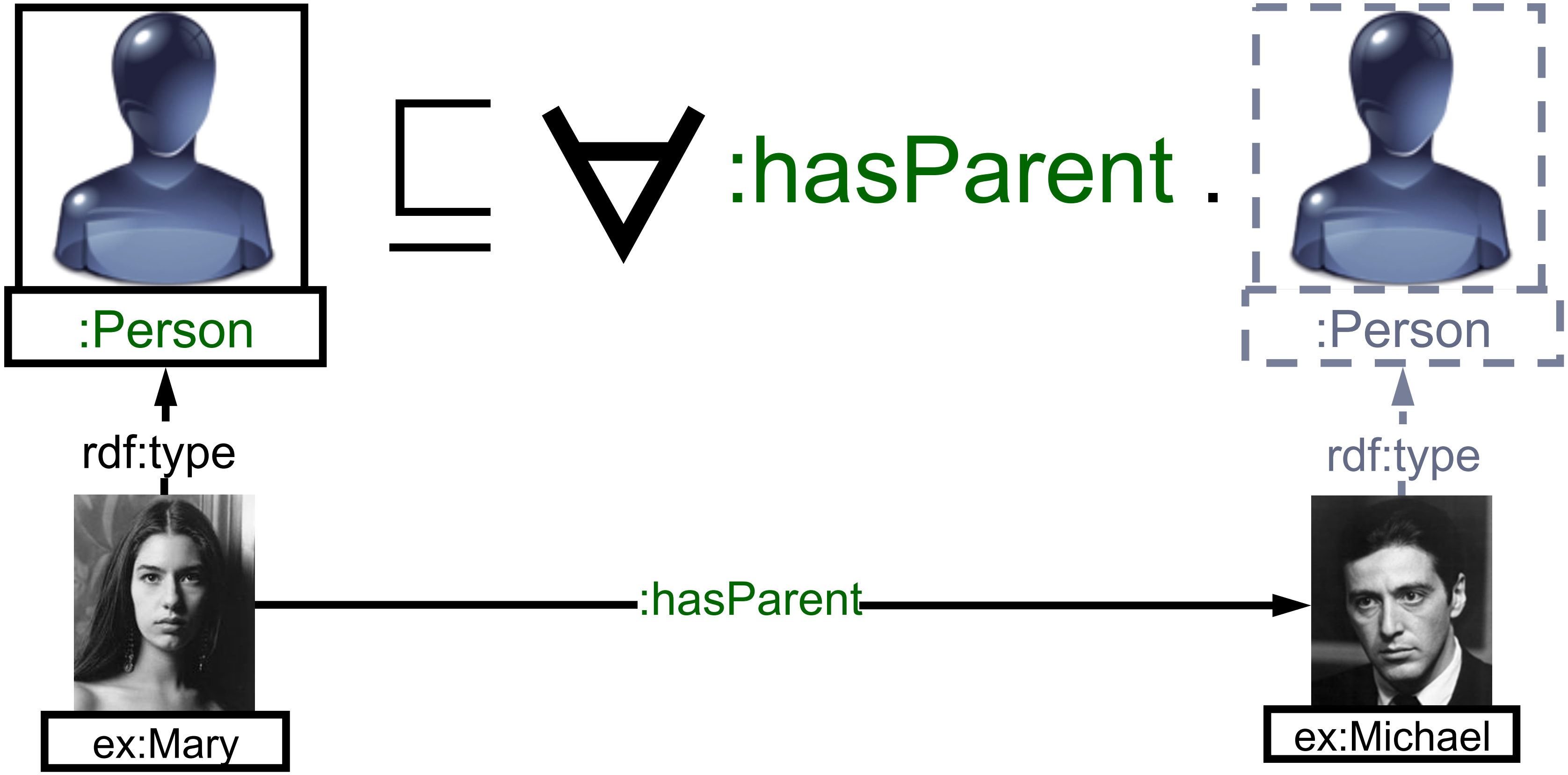
… No Unique Name Assumption (UNA)

- Vito has 3 children?
- ~~Vito has at least 3 children?~~
- ~~Vito has at least 1 child!~~
- Vito has at least 2 children!



`:Man owl:disjointWith :Woman .`

owl:allValuesFrom



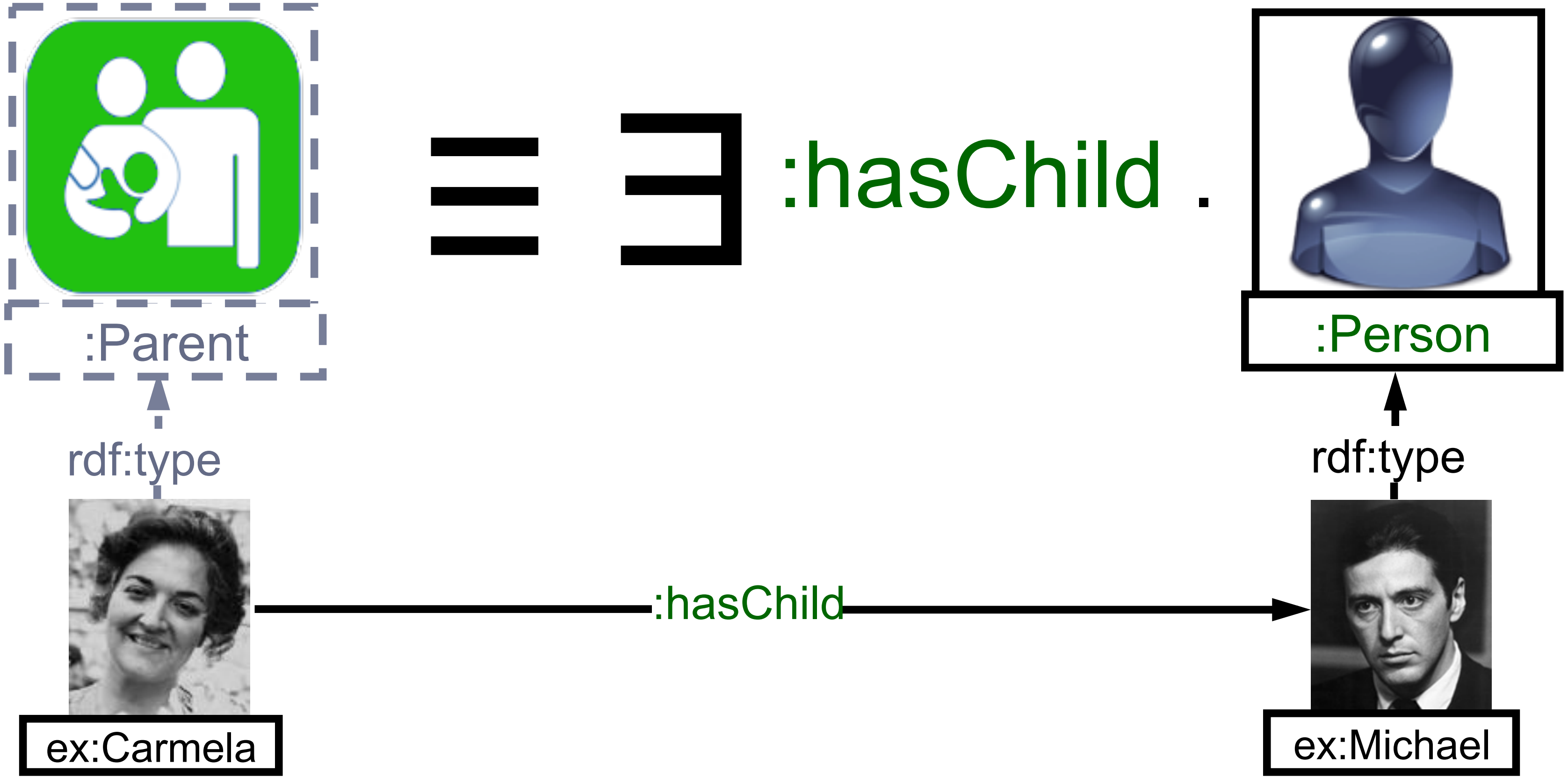
ex:Mary rdf:type :Person ; hasParent ex:Michael .

:Person rdfs:subClassOf

[owl:allValuesFrom :Person ; owl:onProperty :hasParent]

⇒ *ex:Michael rdf:type :Person .*

owl:someValuesFrom (I)



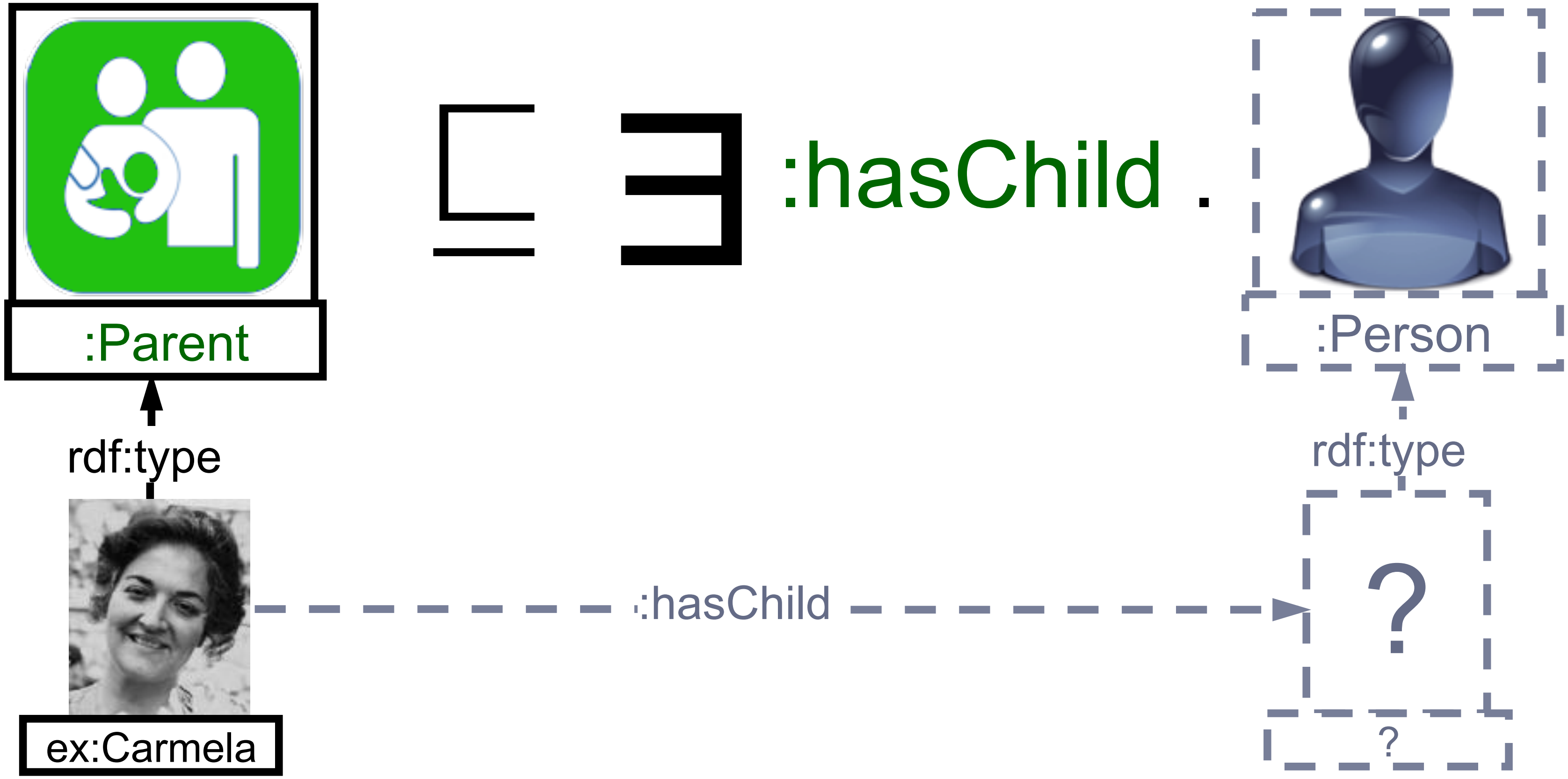
ex:Carmela :hasChild ex:Michael . ex:Michael rdf:type :Person .

:Parent owl:equivalentClass

[owl:someValuesFrom :Person ; owl:onProperty :hasChild]

⇒ ***ex:Carmela rdf:type :Parent .***

owl:someValuesFrom (II)



ex:Carmela rdf:type :Parent .

:Parent rdfs:subClassOf

[owl:someValuesFrom :Person ; owl:onProperty :hasChild]

⇒ ***ex:Carmela :hasChild _:someone . _:someone rdf:type :Person .***

Recap OWL class axioms

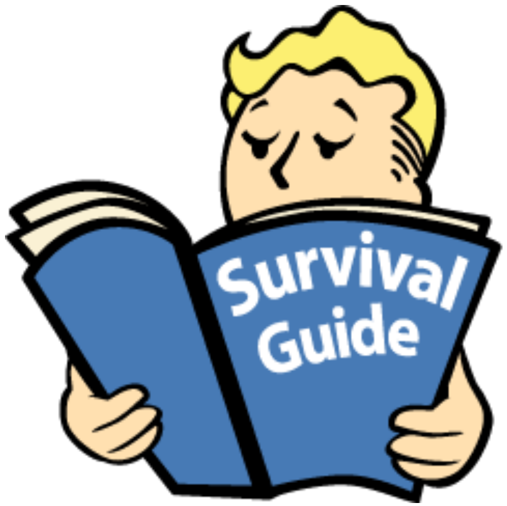
- A class `:Parent` might be the **owl:unionOf** what classes?
- A class `:OnlySon` might be the **owl:intersectionOf** what classes?
- What OWL feature allows to define enumerations?
- An example of **owl:allValuesFrom** for family relations?
- An example of **owl:someValuesFrom** for the class **:Uncle**?

RDFS

1. `rdfs:subPropertyOf`
2. `rdfs:subClassOf`
3. `rdfs:domain`
4. `rdfs:range`

OWL

1. Property Axioms
2. Equality
3. Class Axioms



Lots not covered!

1. `owl:hasKey`
2. `owl:hasValue`
3. `owl:cardinality`(s)
4. `owl:qualifiedCardinality`(s)
5. `owl:AssymetricProperty`
6. `owl:IrreflexiveProperty`
7. `owl:propertyDisjointWith`

…

…

…

Direct/RDF-Based Semantics …

OWL Profiles …

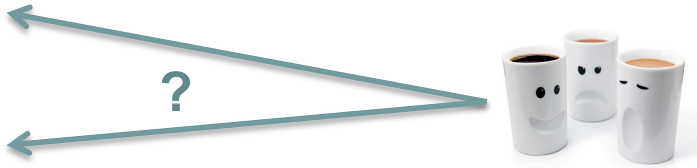
Datatypes …

Reasoning Tasks …

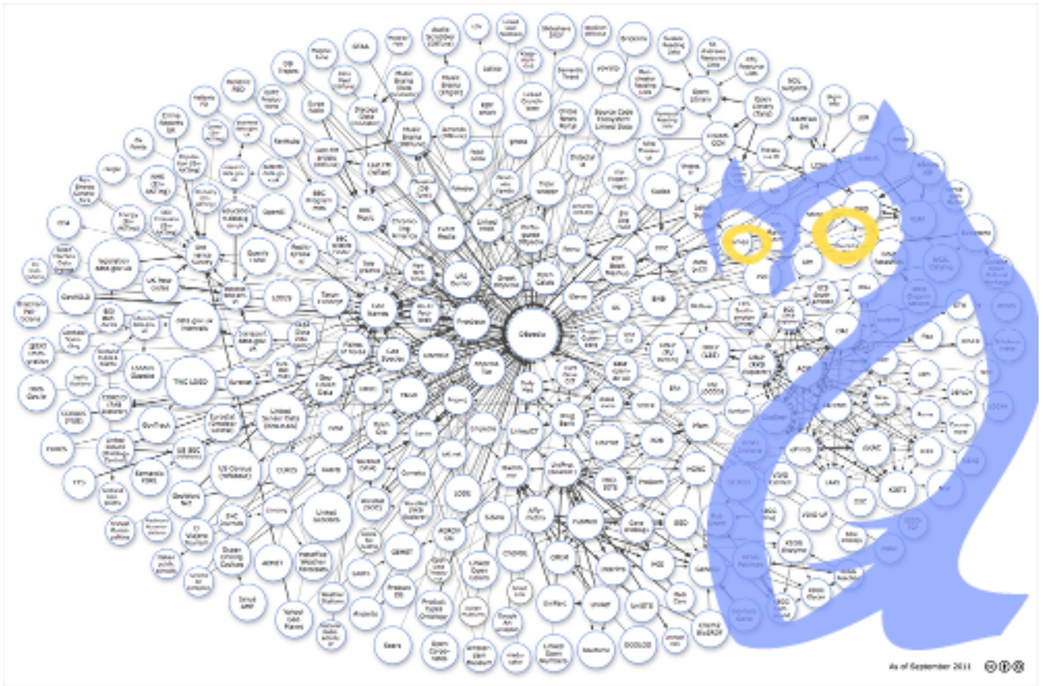
SIEMENS DERI

Lecture Roadmap

- Scope/Motivation
  (Axel)
- Short Introduction to RDFS+OWL
  (Aidan)
- **RDFS+OWL usage in Linked Data**
  (Aidan)
- High-level Reasoning approaches: Query rewriting vs. Materialization
  (Axel)
- Challenges on Reasoning over Linked Data
  (Axel)



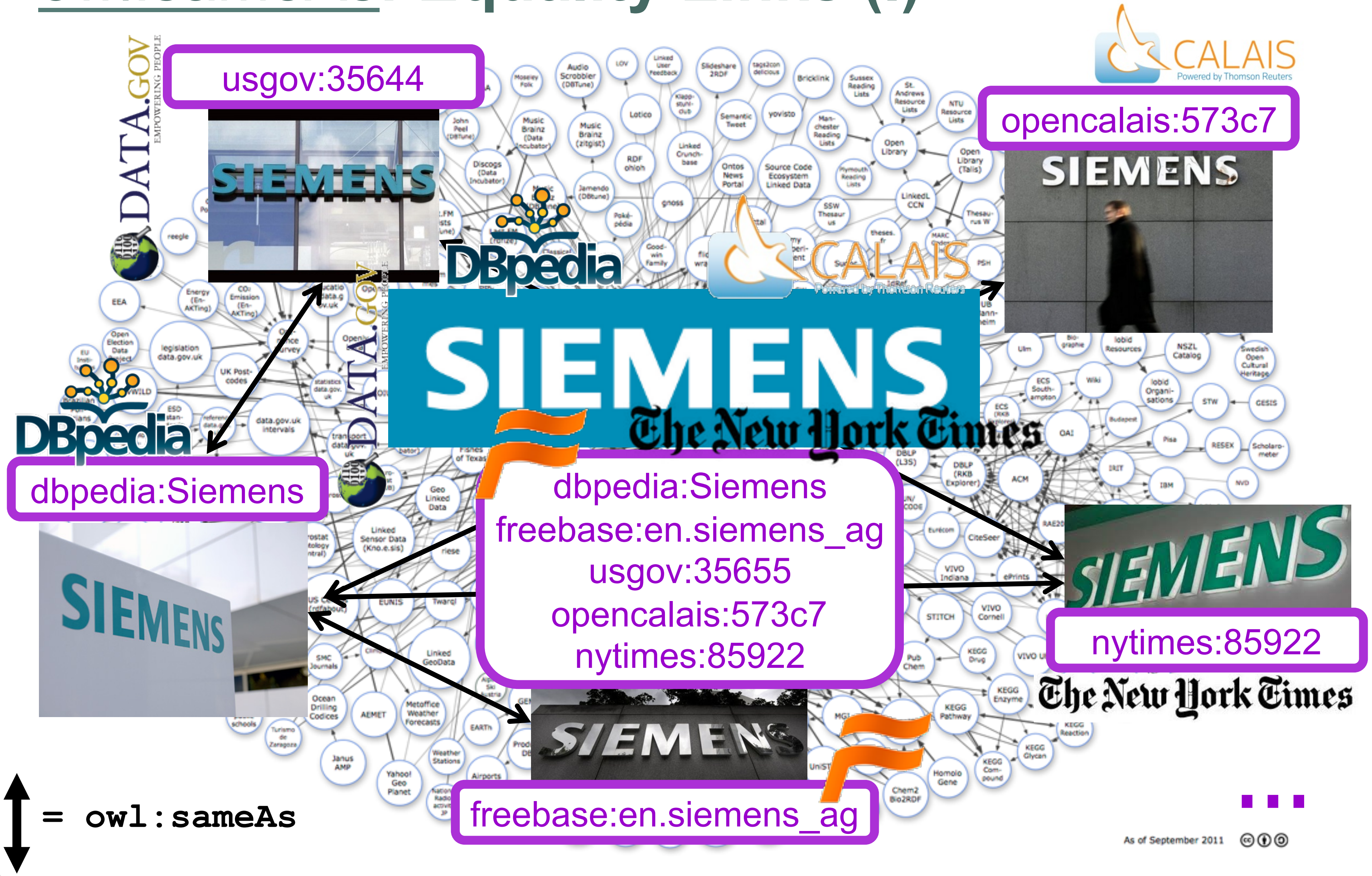
- Practical approaches for Reasoning over Linked Data
  - Quarantined & Authoritative Materialization *(Aidan)*
  - Link-Traversal Based Query Execution with Reasoning *(Aidan)*
  - Reasoning with Attribute Equations *(Axel)*
- Wrap-up/Outlook *(all)*



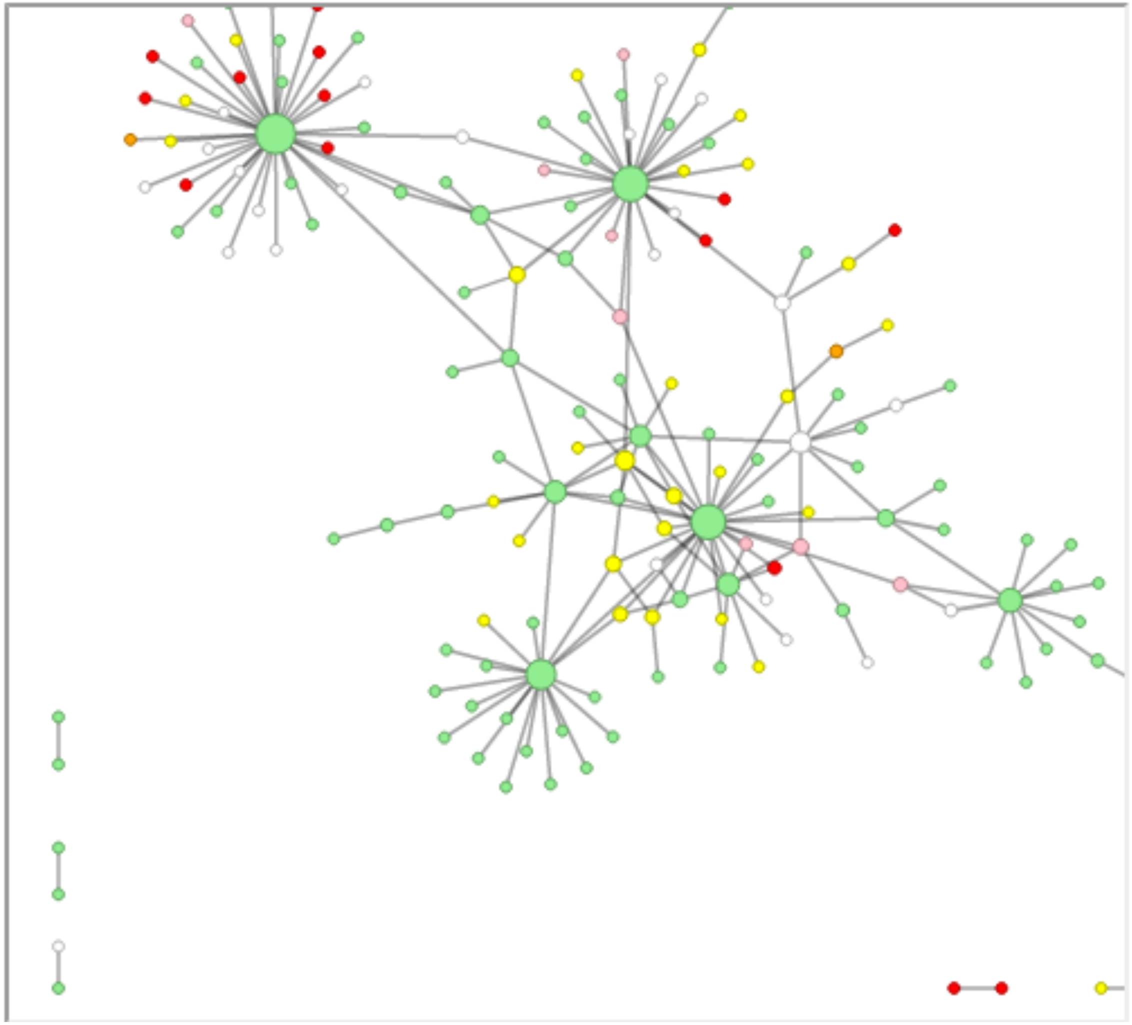
(I) Equality Links

HOW ARE RDFS AND OWL USED IN LINKED DATA?

owl:sameAs: Equality Links (I)



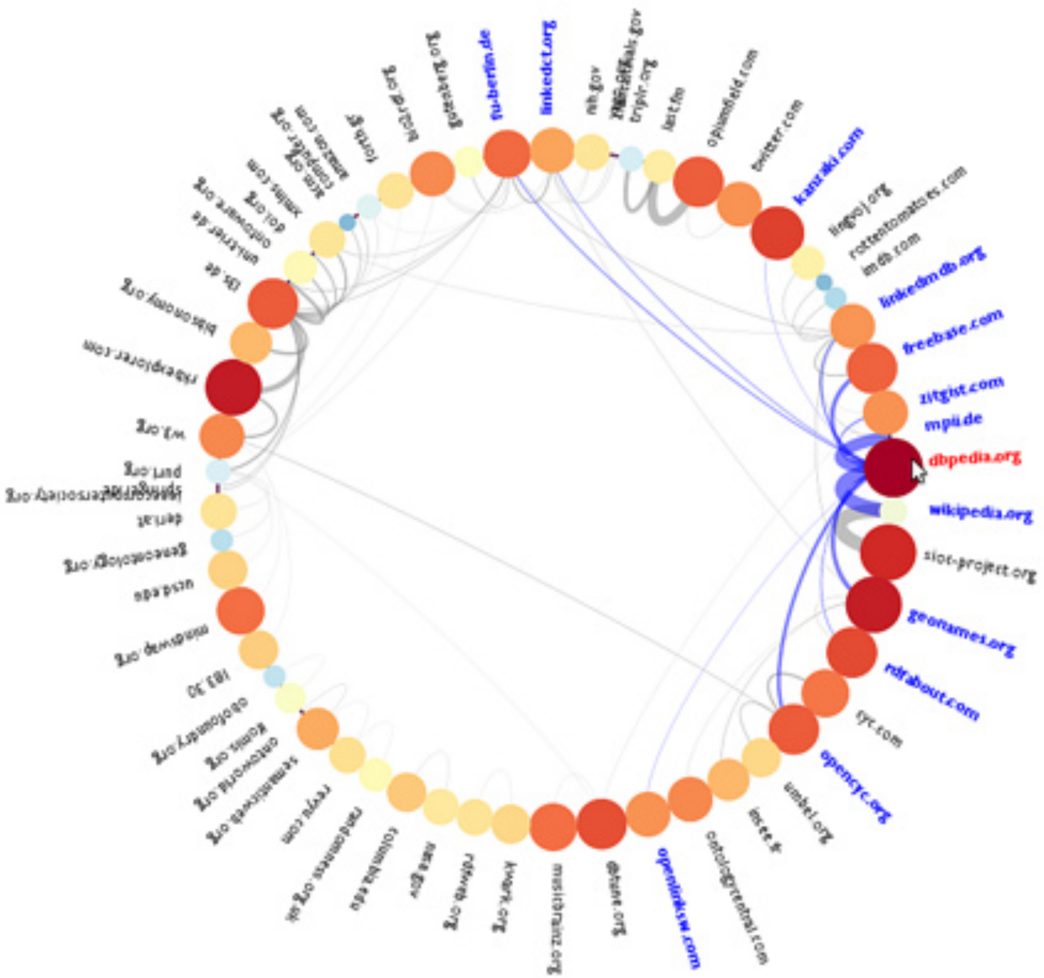
owl:sameAs: Equality Links (II)



Interactive http://inkdroid.org/empirical-cloud/ ; E. Summers

SIEMENS DERI

[owl:sameAs](#): Equality Links (III)



Interactive http://gromgull.net/2010/01/swball/swball.svg ; G.A. Grimnes

SIEMENS DERI

More Details …

When owl:sameAs isn't the Same: An Analysis of Identity in Linked Data

Harry Halpin, Patrick J. Hayes, James P. McCusker, Deborah L. McGuinness, and Henry S. Thompson
{hhalpin,ht}@inf.ed.ac.uk, phayes@ihmc.us, {mccusj,dlm}@cs.rpi.edu

School of Informatics
University of Edinburgh
10 Crichton St.
EH8 9LW Edinburgh, UK
and
Institute for Human and Machine Cognition
40 South Alcaniz St.
Pensacola, FL 32502 USA
and
Tetherless World Constellation
Department of Computer Science
Rensselaer Polytechnic Institute
110 8th Street, Troy, NY 12180 USA

Abstract. In Linked Data, the use of *owl:sameAs* is ubiquitous in interlinking data-sets. There is however, ongoing discussion about its use, and potential misuse, particularly with regards to interactions with inference. In fact, *owl:sameAs* can be viewed as encoding only one point on a scale of similarity, one that is often too strong for many of its current uses. We describe how referentially opaque contexts that do not allow inference exist, and then outline some varieties of referentially-opaque alternatives to *owl:sameAs*. Finally, we report on an empirical experiment over randomly selected *owl:sameAs* statements from the Web of data. This theoretical apparatus and experiment shed light upon how *owl:sameAs* is being used (and misused) on the Web of data.

Keywords: *linked data, identity, coreference*

Harry Halpin, Patrick J. Hayes, James P. McCusker, Deborah L. McGuinness, Henry S. Thompson. "WHEN OWL:SAMEAS ISN'T THE SAME: AN ANALYSIS OF IDENTITY IN LINKED DATA". In the Proceedings of the 9th International Semantic Web Conference (1) 2010: 305-320

More Details …

Scalable and Distributed Methods for Entity Matching, Consolidation and Disambiguation over Linked Data Corpora

Aidan Hogan [a], Antoine Zimmermann [b], Jürgen Umbrich [a], Axel Polleres [c], Stefan Decker [a],

[a] *Digital Enterprise Research Institute, National University of Ireland, Galway*
[b] *INSA-Lyon, LIRIS, UMR5205, F-69621, France*
[c] *Siemens AG Österreich, Siemensstrasse 90, 1210 Vienna, Austria*

Abstract

With respect to large-scale, static, Linked Data corpora, in this paper we discuss scalable and distributed methods for entity consolidation (aka. smushing, entity resolution, object consolidation, etc.) to locate and process names that signify the same entity. We investigate (i) a baseline approach, which uses explicit `owl:sameAs` relations to perform consolidation; (ii) extended entity consolidation which additionally uses a subset of OWL 2 RL/RDF rules to derive novel `owl:sameAs` relations through the semantics of inverse-functional properties, functional-properties and (max-)cardinality restrictions with value one; (iii) deriving weighted concurrence measures between entities in the corpus based on shared inlinks/outlinks and attribute values using statistical analyses; (iv) disambiguating (initially) consolidated entities based on inconsistency detection using OWL 2 RL/RDF rules. Our methods are based upon distributed sorts and scans of the corpus, where we deliberately avoid the requirement for indexing all data. Throughout, we offer evaluation over a diverse Linked Data corpus consisting of 1.118 billion quadruples derived from a domain-agnostic, open crawl of 3.985 million RDF/XML Web documents, demonstrating the feasibility of our methods at that scale, and giving insights into the quality of the results for real-world data.

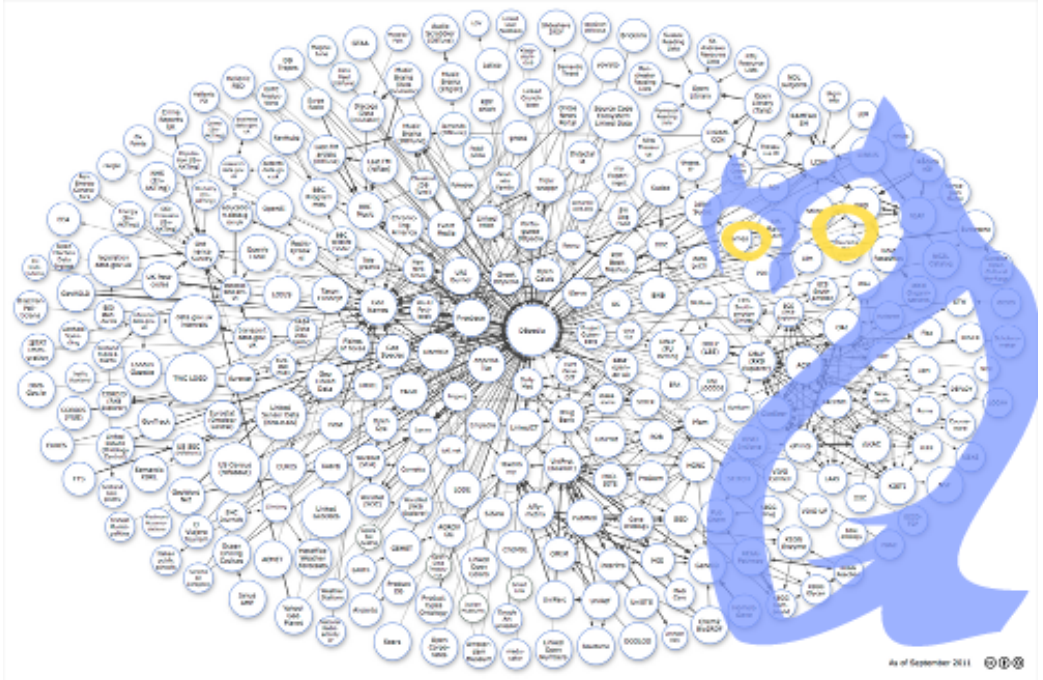
Key words: entity consolidation, web data, linked data, rdf

1. Introduction

Over a decade since the dawn of the Semantic Web,

The Linked Open Data project has advocated the goal of providing dereferencable machine readable data in a common format (RDF), with emphasis on the re-use

Aidan Hogan, Antoine Zimmermann, Jürgen Umbrich, Axel Polleres and Stefan Decker. "SCALABLE AND DISTRIBUTED METHODS FOR ENTITY MATCHING, CONSOLIDATION AND DISAMBIGUATION OVER LINKED DATA CORPORA ". In the Journal of Web Semantics 10: pp. 76–110, 2012



(II) Linked Vocabularies

HOW ARE RDFS AND OWL USED IN LINKED DATA?

Linked Vocabularies (in RDFS/OWL)

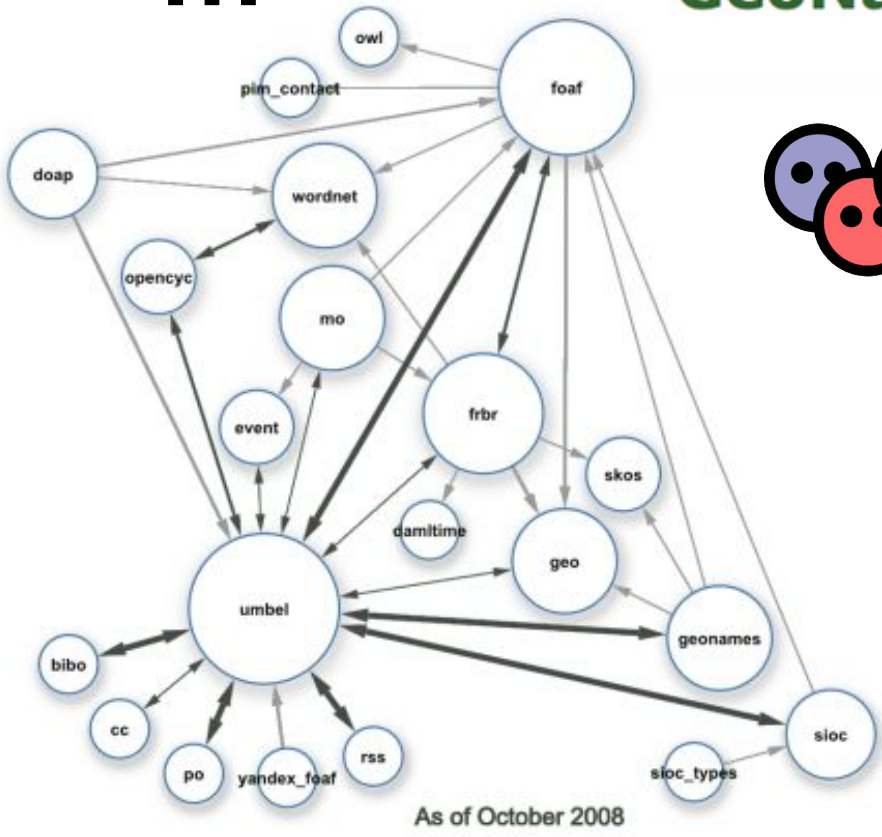
SIOC

DOAP

DBpedia

music ontology

...



GeoNames

GoodRelations
The Web Ontology for E-Commerce

Image from http://blog.dbtune.org/public/.081005_lod_constellation_m.jpg: Giasson, Bergman

Vocabulary Re-use / Extension

"*The Web of Data takes a two-fold approach to dealing with heterogeneous data representation.*

"*On the one hand side, it tries to avoid heterogeneity by advocating the **reuse of terms from widely deployed vocabularies**.* […] *a set of vocabularies for describing common things like people, places or projects has emerged in the Linked Data community.*

"*On the other hand,* […] *a Linked Data application which discovers some data* […] *using a previously unknown vocabulary should be able to find all meta-information that it requires to translate the data into a representation that it understands and can process.* **[Vocabularies provide]** ***RDFS and OWL definition of terms***, **[each]** ***vocabulary term links to its own definition***, […] ***mappings*** **[are provided]** ***between terms from different vocabularies***."

Heath & Bizer. *Linked Data: Evolving the Web into a Global Data Space.* Morgan & Claypool. 2011.

Prominent Vocabularies (I)

- Dublin Core (DC)
  - Meta-data for documents

Properties in the */terms/* namespace	abstract, accessRights, accrualMethod, accrualPeriodicity, accrualPolicy, alternative, audience, available, bibliographicCitation, conformsTo, contributor, coverage, created, creator, date, dateAccepted, dateCopyrighted, dateSubmitted, description, educationLevel, extent, format, hasFormat, hasPart, hasVersion, identifier, instructionalMethod, isFormatOf, isPartOf, isReferencedBy, isReplacedBy, isRequiredBy, issued, isVersionOf, language, license, mediator, medium, modified, provenance, publisher, references, relation, replaces, requires, rights, rightsHolder, source, spatial, subject, tableOfContents, temporal, title, type, valid
Properties in the legacy */elements/1.1/* namespace	contributor, coverage, creator, date, description, format, identifier, language, publisher, relation, rights, source, subject, title, type
Vocabulary Encoding Schemes	DCMIType, DDC, IMT, LCC, LCSH, MESH, NLM, TGN, UDC
Syntax Encoding Schemes	Box, ISO3166, ISO639-2, ISO639-3, Period, Point, RFC1766, RFC3066, RFC4646, RFC5646, URI, W3CDTF
Classes	Agent, AgentClass, BibliographicResource, FileFormat, Frequency, Jurisdiction, LicenseDocument, LinguisticSystem, Location, LocationPeriodOrJurisdiction, MediaType, MediaTypeOrExtent, MethodOfAccrual, MethodOfInstruction, PeriodOfTime, PhysicalMedium, PhysicalResource, Policy, ProvenanceStatement, RightsStatement, SizeOrDuration, Standard

Table from http://dublincore.org/documents/dcmi-terms/

Prominent Vocabularies (II)

- Friend Of A Friend (FOAF)
  - Personal Information

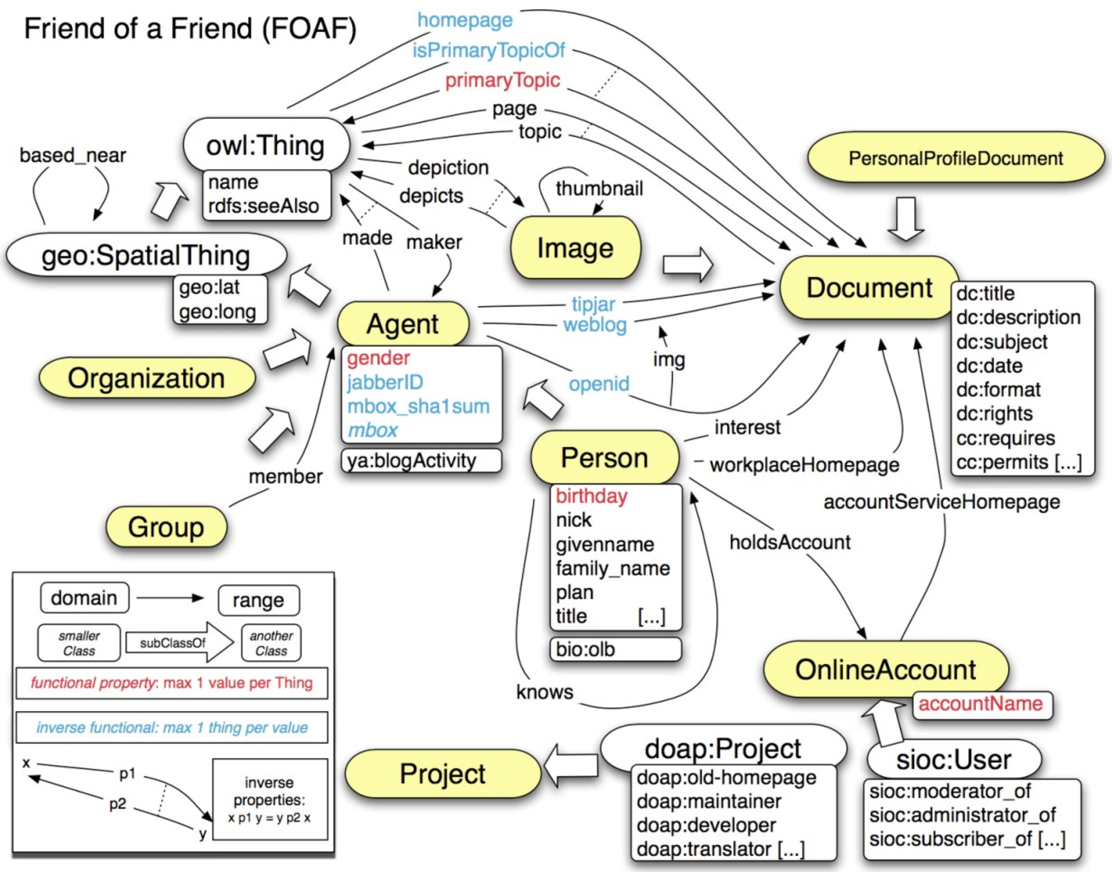


Image from http://www.deri.ie/fileadmin/images/blog/ : **Breslin**

Prominent Vocabularies (III)

- Semantically Interlinked Online Communities (SIOC)
  - Online communities and presence

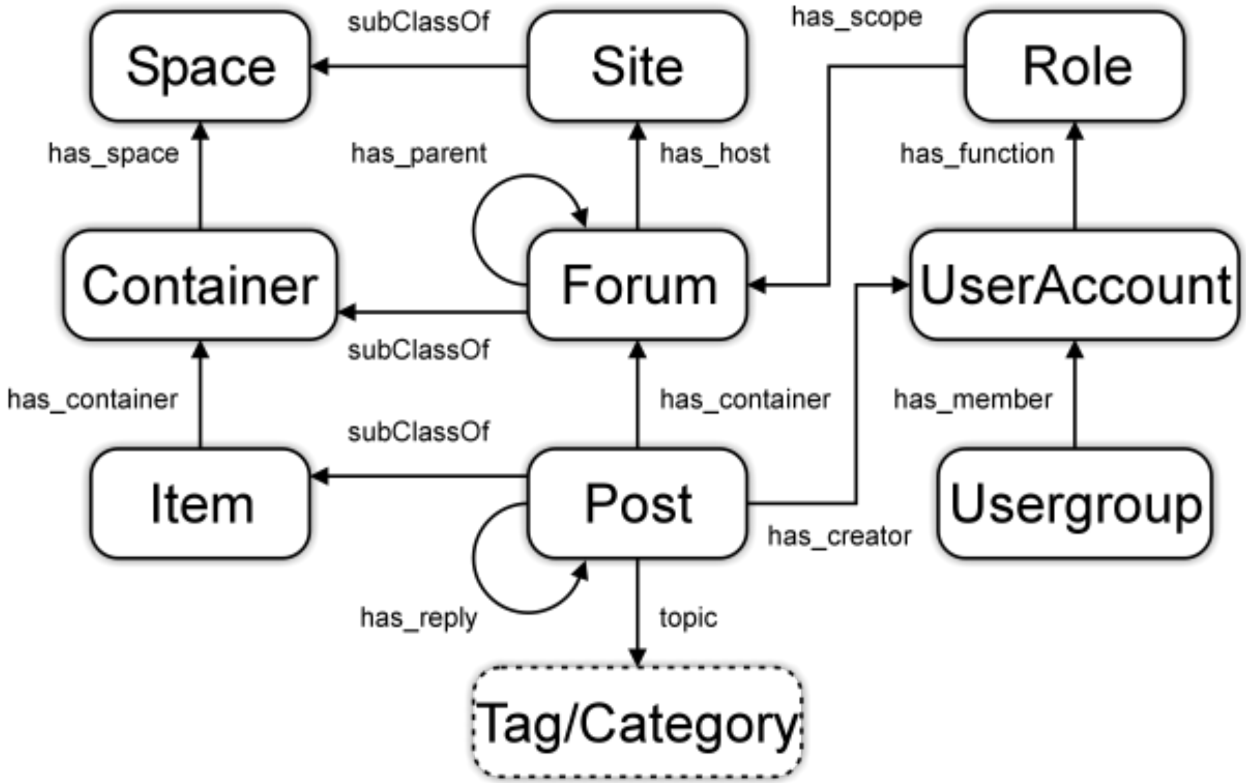


Image from http://rdfs.org/sioc/spec/ : **Bojārs, Breslin et al.**

SIEMENS DERI

Prominent Vocabularies (IV)

- Simple Knowledge Organization System
  - Meta-vocabulary for generic taxonomies

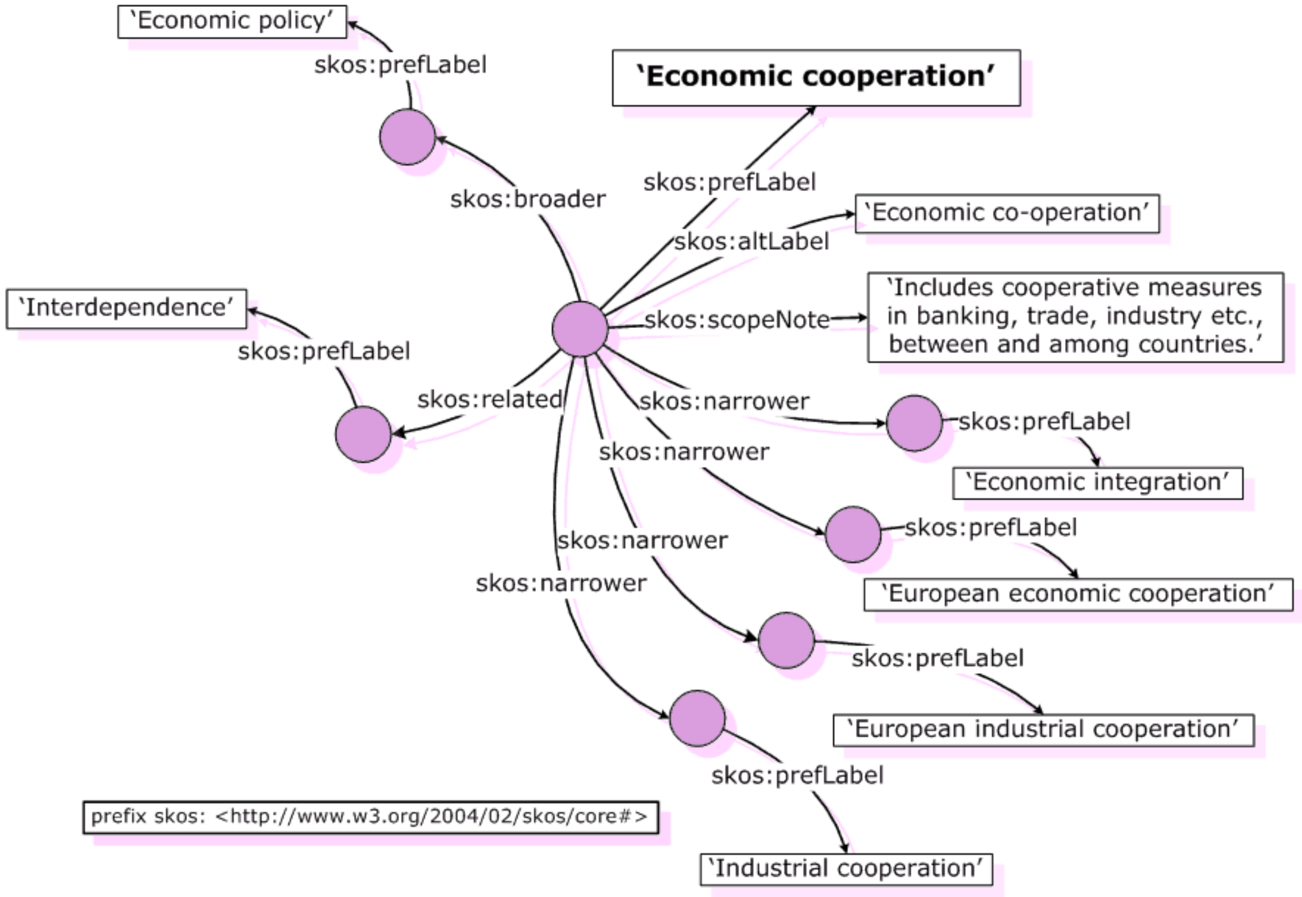


Image from http://www.w3.org/TR/swbp-skos-core-guide : Miles, Brickley

W3C Recommendation

SIEMENS DERI

Prominent Vocabularies: Work Together

- Ontologies combine to cover multiple areas

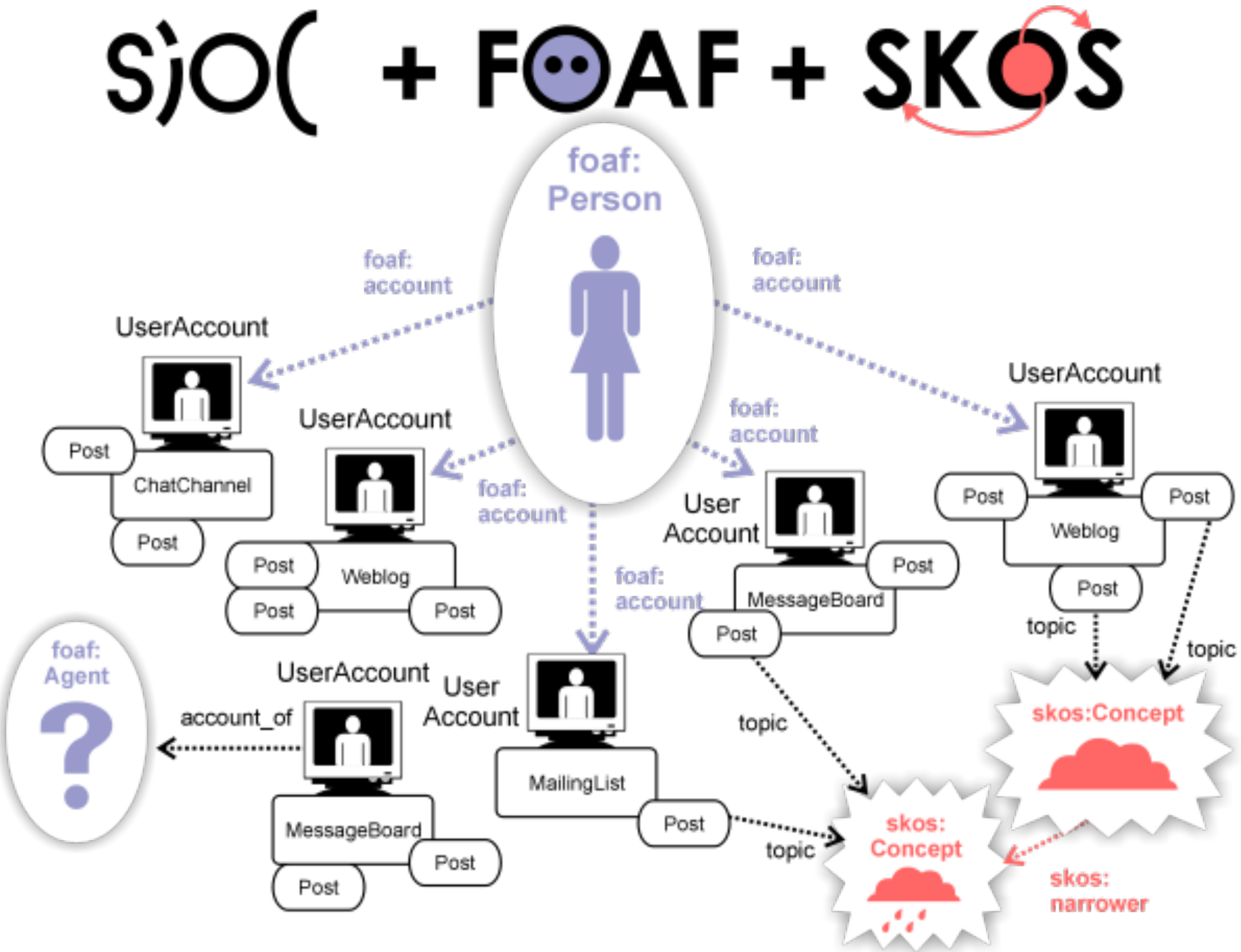


Image from http://sioc-project.org/node/158; **Breslin**

SIEMENS DERI

Prominent Vocabularies (V)

DOAP

- Description Of A Project (DOAP)
  - Describing contributors to (software) projects

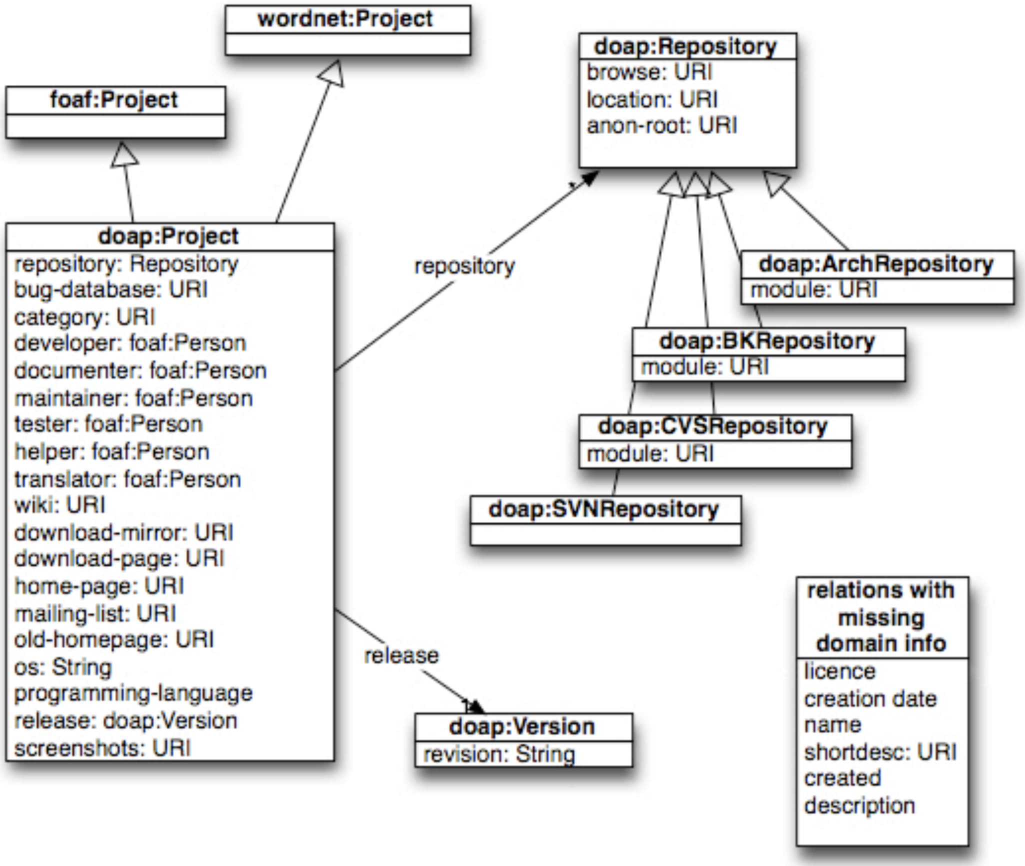


Image from http://code.google.com/p/baetle/wiki/DoapOntology ; **Breslin**

SIEMENS DERI

Prominent Vocabularies (VI)

- Music Ontology (MO)
  - Artists, bands, songs, records, albums, performances

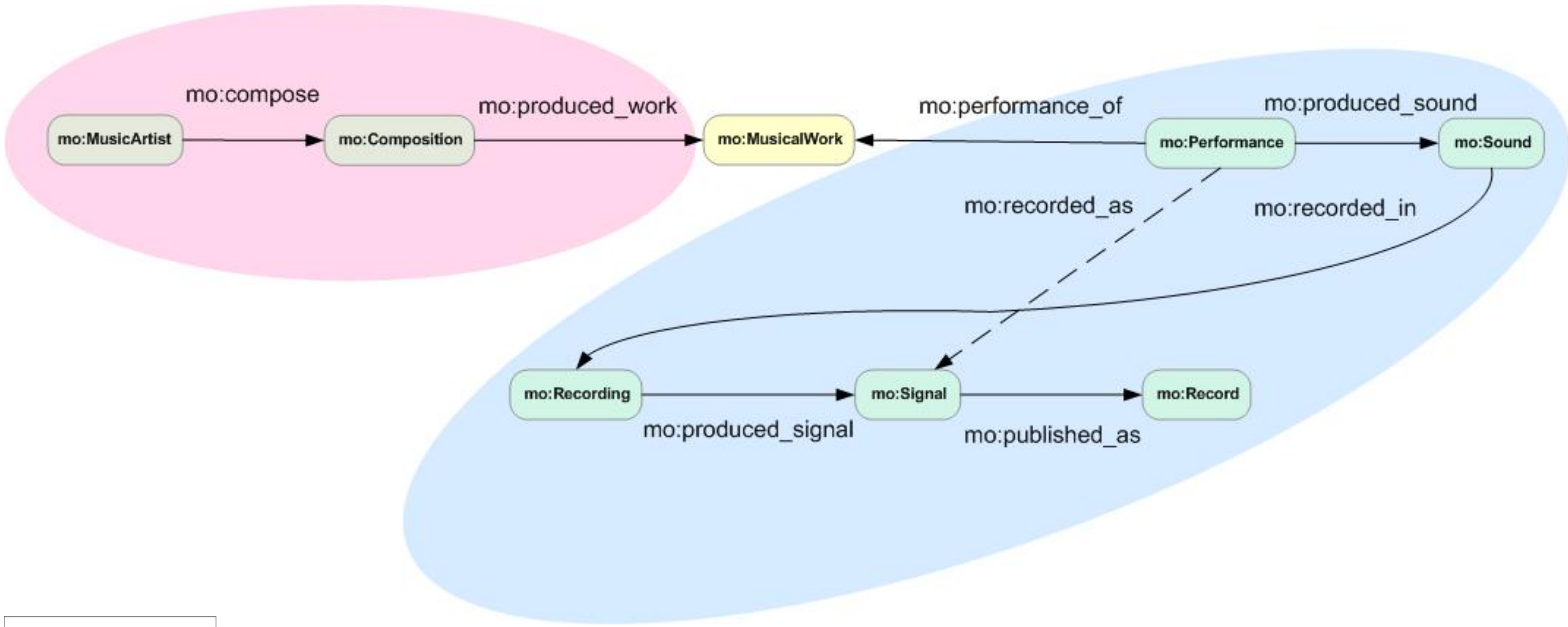


Image from http://musicontology.com/; **Raimond, Giasson**

Prominent Vocabularies (VII)

- GoodRelations (GR)
  - Products, services, stores, prices, etc.

GoodRelations
The Web Ontology for E-Commerce

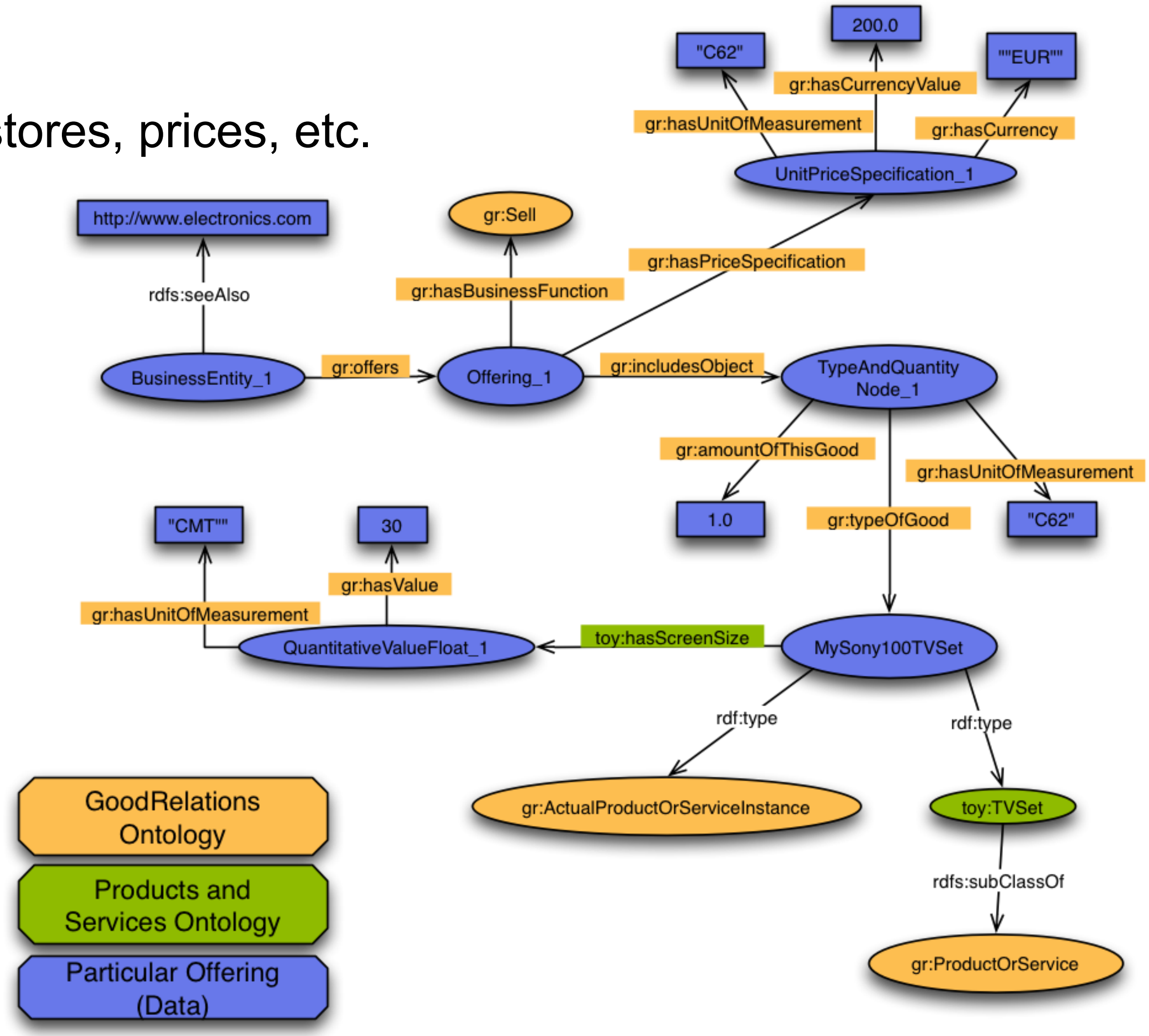


Image from http://www.heppnetz.de/projects/goodrelations/primer/; **Hepp**

SIEMENS DERI

Prominent Vocabularies (VII)

- GoodRelations (GR)
  - Products, services, stores, prices, etc.

GoodRelations
The Web Ontology for E-Commerce

Shopping results for **One-Touch Gold BBQ (22.5-in.): Blue by Weber**

Weber 75001 **Blue One-Touch 22.5" Blue** Stainless Steel Charcoal **Gold** Gr +1
$159.70 - Build.com
Weber One Touch 751001 **22.5" One-Touch Gold** Charcoal Grill with ... +1
$129.00 - AJ Madison
Weber One Touch 758001 **22.5" One-Touch Gold** Charcoal Grill with ... +1
$149.00 - AJ Madison

Google™ BEST BUY® O'REILLY® Y!™ Kmart overstock.com® Sears

Prominent Vocabularies (VIII)

- DBpedia Ontology
  - Hundreds of classes and properties
  - Captures “info-box” information

```
{{Infobox Town AT |
  name = Innsbruck |
  image_coa =  InnsbruckWappen.png |
  image_map = Karte-tirol-I.png |
  state = [[Tyrol]] |
  regbzk = [[Statutory city]] |
  population = 117,342 |
  population_as_of = 2006 |
  pop_dens = 1,119 |
  area = 104.91 |
  elevation = 574 |
  lat_deg = 47 |
  lat_min = 16 |
  lat_hem = N |
  lon_deg = 11 |
  lon_min = 23 |
  lon_hem = E |
  postal_code = 6010-6080 |
  area_code = 0512 |
  licence = I |
  mayor = Hilde Zach |
  website = [http://innsbruck.at] |
}}
```

Innsbruck

Country	Austria
State	Tyrol
Administrative region	Statutory city
Population	117,342 (2006)
Area	104.91 km²
Population density	1,119 /km²
Elevation	574 m
Coordinates	47°16′ N 11°23′ E
Postal code	6010-6080
Area code	0512
Licence plate code	I
Mayor	Hilde Zach
Website	www.innsbruck.at

About: Innsbruck

An Entity of Type : city, from Named Graph : http://dbpedia.org, within Data Space : dbpedia.org

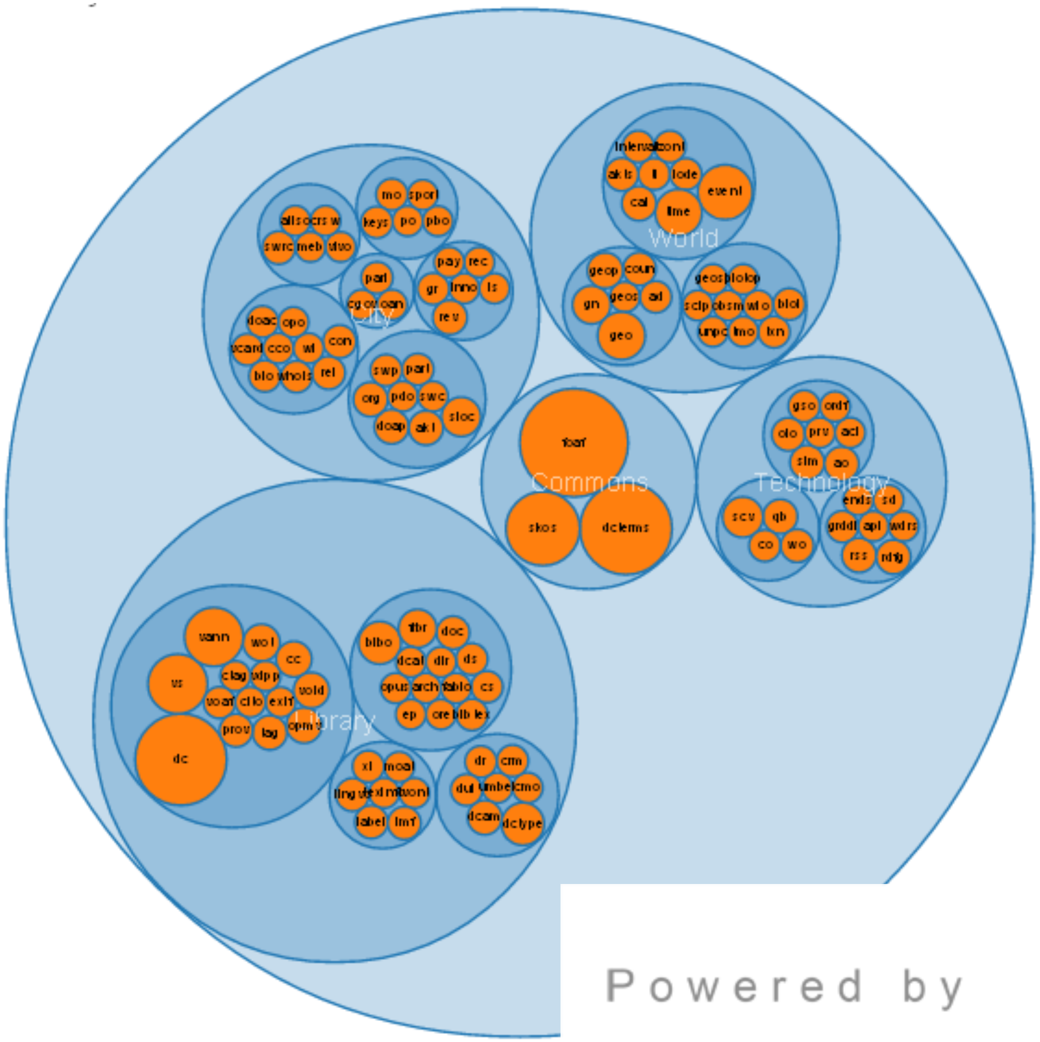
Innsbruck is the capital city of the federal state of Tyrol in western Austria. It is located in the Inn Valley at the junction with the Wipptal, which provides access to the Brenner Pass, some 30 kilometers (19 mi) south of Innsbruck.

Property	Value
dbpedia-owl:PopulatedPlace/populationDensity	▪ 1119.0
dbpedia-owl:abstract	▪ Innsbruck ist die Landeshauptstadt des Bundeslandes Tirol Transit-Strecke Brenner (Auto- und Eisenbahn) nach Südti (Brücke über den Inn). Innsbruck ist mit 118.082 (Stand 1. und Salzburg die fünftgrößte Stadt Österreichs, im Ballung dazu kommen ca. 30.000 Studenten und andere Nebenwol Nächtigungen von Städtetouristen. ▪ Innsbruck is the capital city of the federal state of Tyrol in v the junction with the Wipptal, which provides access to the

See http://wiki.dbpedia.org/

SIEMENS DERI

Linked Open Vocabularies (LOV)

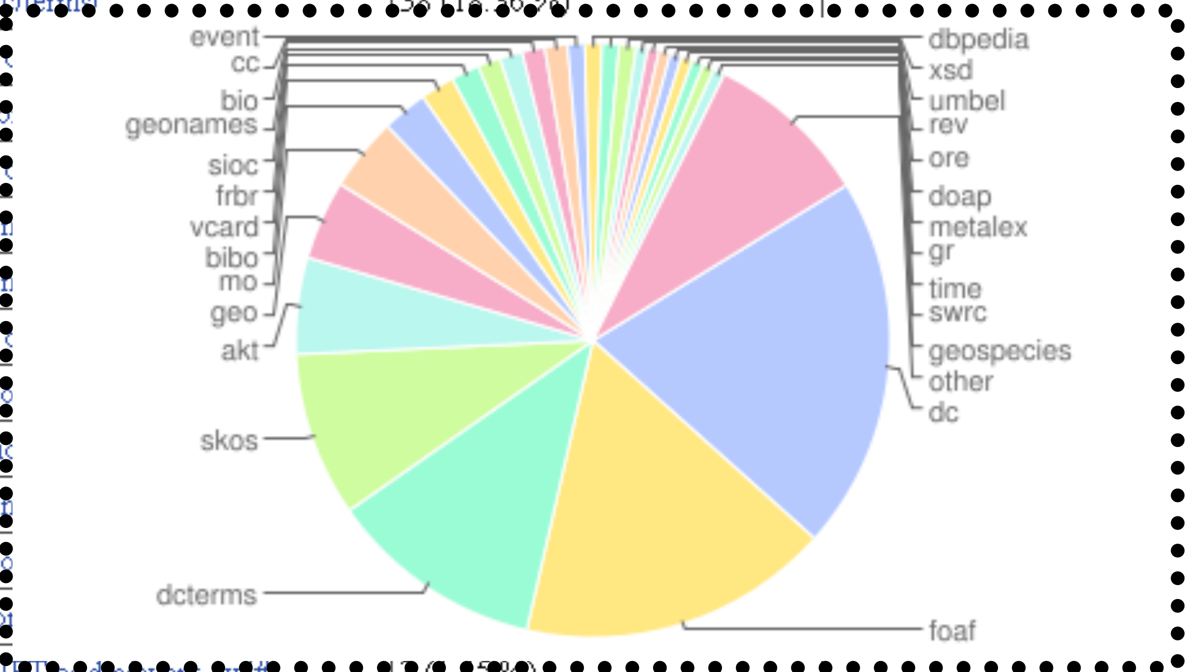


Powered by MONDECA

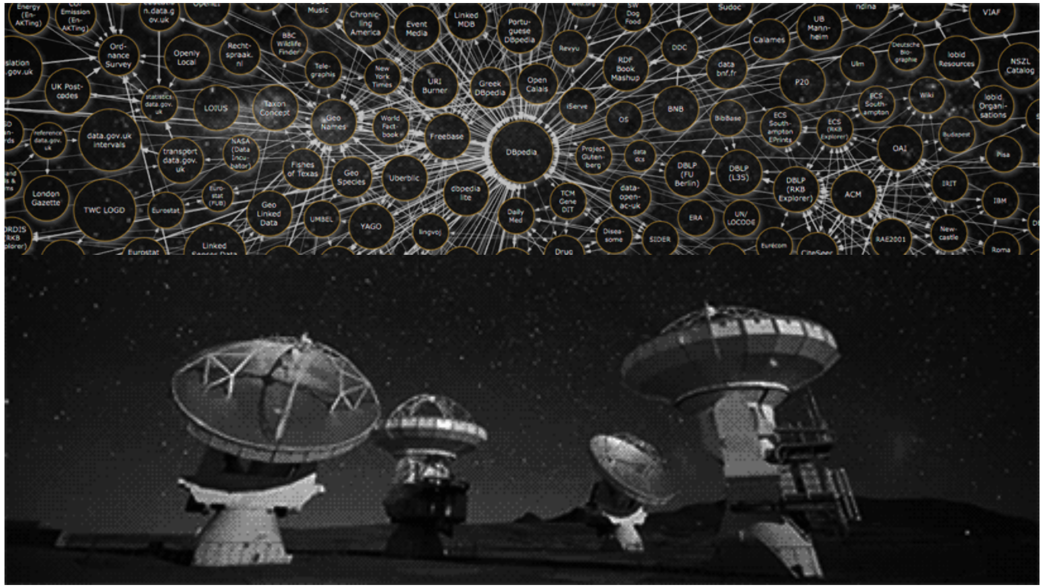
Interactive http://labs.mondeca.com/dataset/lov/; **Vatant, Vandenbussche**

Vocabulary Re-use

Vocabulary prefix	Vocabulary link	Number of usages in data sets
dc	http://purl.org/dc/elements/1.1/	66 (31.88 %)
foaf	http://xmlns.com/foaf/0.1/	55 (26.57 %)
dcterms	http://purl.org/dc/terms/	38 (18.36 %)
skos	http://www.w3.	
akt	http://www.akt	
geo	http://www.w3.	
mo	http://purl.org/o	
bibo	http://purl.org/o	
vcard	http://www.w3.	
frbr	http://purl.org/v	
sioc	http://rdfs.org/si	
geonames	http://www.geo	
bio	http://purl.org/v	
cc	http://creativeco	
event	http://purl.org/NET/c4dm/event.owl#	3 (1.45 %)
dbpedia	http://dbpedia.org/resource/	3 (1.45 %)
xsd	http://www.w3.org/2001/XMLSchema#	3 (1.45 %)
umbel	http://umbel.org/umbel#	3 (1.45 %)



Info from http://www4.wiwiss.fu-berlin.de/lodcloud/state/ : Bizer, Jentzsch, Cyganiak



A Survey

OWL LANGUAGE FEATURES USED IN LINKED DATA

So what OWL is used out there?

- Looked at Billion Triple Challenge 2011 (BTC 2011) Dataset
  - 2.1 billion quadruples, crawled from…
  - 7.4 million RDF/XML documents, covering…
  - 791 (pay-level) domains

- Counted OWL features used in the dataset:
  - Per use
  - Per document
  - Per domain
  - Can be skewed by data

- Ranked OWL features using PageRank:
  - Rank documents based on dereferenceable links
  - For each OWL feature, sum the rank of documents using it
  - **Approximates probability of encountering an OWL feature during a random walk of the data**

RDFS/OWL Features Ranked by Use

1	`rdf:Property`	5.74E-1
2	`rdfs:range`	4.67E-1
3	`rdfs:domain`	4.62E-1
4	`rdfs:subClassOf`	4.60E-1
5	`rdfs:Class`	4.45E-1
6	`rdfs:subPropertyOf`	2.35E-1
7	`owl:Class`	1.74E-1
8	`owl:ObjectProperty`	1.68E-1
9	`rdfs:Datatype`	1.68E-1
10	`owl:DatatypeProperty`	1.65E-1
11	`owl:AnnotationProperty`	1.60E-1
12	`owl:FunctionalProperty`	9.18E-2
13	`owl:equivalentProperty`	8.54E-2
14	`owl:inverseOf`	7.91E-2
15	`owl:disjointWith`	7.65E-2

RDFS/OWL Features Ranked by Use

…

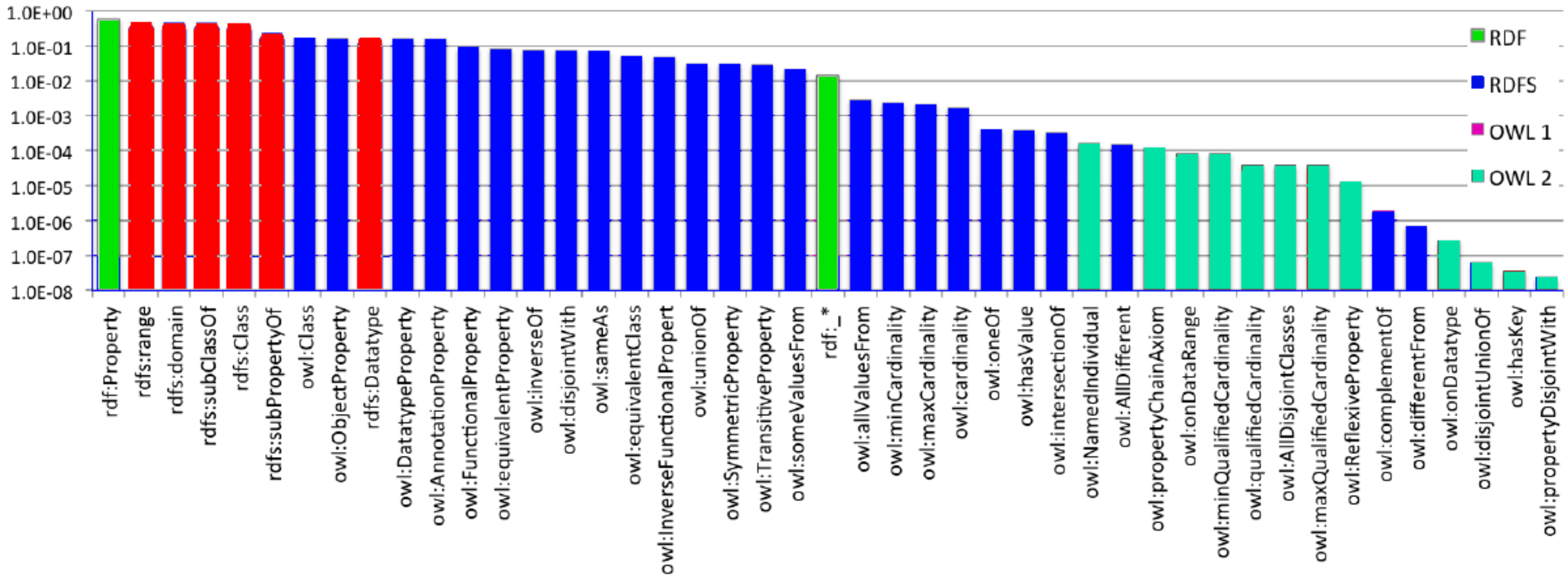
16	owl:sameAs	7.29E-2
17	owl:equivalentClass	5.24E-2
18	owl:InverseFunctionalProperty	4.79E-2
19	owl:unionOf	3.15E-2
20	owl:SymmetricProperty	3.13E-2
21	owl:TransitiveProperty	2.98E-2
22	owl:someValuesFrom	2.13E-2
23	rdf:_*	1.42E-2
24	owl:allValuesFrom	2.98E-3
25	owl:minCardinality	2.43E-3
26	owl:maxCardinality	2.14E-3
27	owl:cardinality	1.75E-3
28	owl:oneOf	4.13E-4
29	owl:hasValue	3.91E-4
30	owl:intersectionOf	3.37E-4

Observations?

- RDFS features amongst the most prominently used
- OWL 2 features not yet used prominently

RDF | **RDFS** | **OWL** | **OWL 2**

***x*-axis is log-scale!**



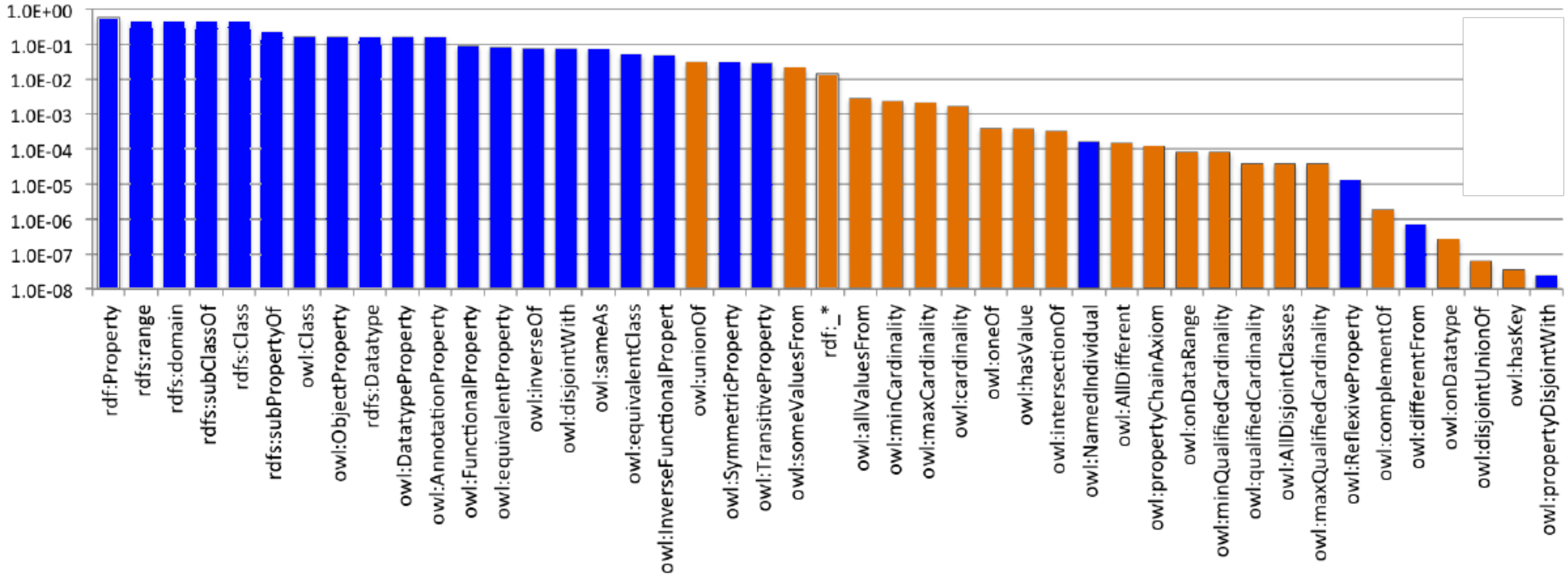
SIEMENS DERI

Observations?

- (OWL) Features expressible with a single RDF triple are most prominent
  - Roughly speaking, features not *requiring* blank nodes
    - e.g., sub-class/-property, inverse-of, equivalent property/class, sameas, domain/range, disjoint with, etc.
  - Not those requiring lists or *n*-ary predicate in RDF mapping
    - e.g., union, intersection, cardinalities, all-disjoint, some/all/has-value restrictions, hasKey, pCAs, etc.

Single Triple (No BNodes) | Multi-Triple (Needs BNodes)

***x*-axis is log-scale!**



More Details …

OWL: Yet to arrive on the Web of Data?

Birte Glimm
Ulm University, Institute of Artificial Intelligence, 89069 Ulm, Germany

Aidan Hogan
Digital Enterprise Research Institute, National University of Ireland Galway, Ireland

Markus Krötzsch
University of Oxford, Department of Computer Science, OX1 3QD Oxford, United Kingdom

Axel Polleres
Siemens AG Österreich, Siemensstrasse 90, 1210 Vienna, Austria

ABSTRACT

Seven years on from OWL becoming a W3C recommendation, and two years on from the more recent OWL 2 W3C recommendation, OWL has still experienced only patchy uptake on the Web. Although certain OWL features (like owl:sameAs) are very popular, other features of OWL are largely neglected by publishers in the Linked Data world. This may suggest that despite the promise of easy implementations and the proposal of tractable profiles suggested in OWL's second version, there is still no "right" standard fragment for the Linked Data community. In this paper, we (1) analyse uptake of OWL on the Web of Data, (2) gain insights into the OWL fragment that is actually used/usable on the Web, where we arrive at the conclusion that this fragment is likely to be a simplified profile based on OWL RL, (3) propose and discuss such a new fragment, which we call OWL LD (for Linked Data).

1. INTRODUCTION

Under the initial impetus of the Linking Open Data project – and guided by the Linked Data principles [3] and associated best-

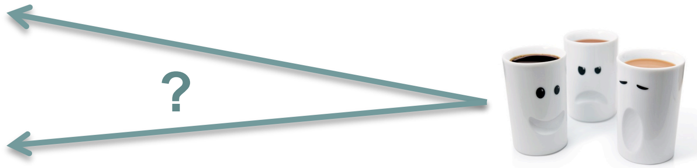
lines recommend use of RDFS [18, § 4.4.2] for defining and interlinking vocabularies. Regarding OWL, guidelines explicitly recommend use of owl:equivalentClass, owl:equivalentProperty, owl:InverseFunctionalProperty & owl:inverseOf [18, § 4.4.2]. However, other OWL features are not concretely mentioned.

In terms of standards, RDFS and OWL 1 pre-date the Linked Data movement and are not directly tailored towards Linked Data requirements. Although the informative entailment rules for supporting RDFS inferences are *relatively* straightforward, things like the infinitely many entailed axiomatic triples reduce their practicality [27]. In OWL 1 the situation is more complex: OWL 1 Full further extends the RDFS semantics to the extent that reasoning becomes undecidable. In OWL 1 DL and OWL 1 Lite, where the semantics are based on Description Logics, typical reasoning tasks remain decidable, but are of exponential or harder worst-case complexity. OWL 2 addresses the complexity issue by defining *profiles* [6]: fragments for which at least some reasoning tasks are tractable. Reasoning with inconsistent data is, however, still problematic in any OWL fragment. Further, each profile is a syntactic subset of OWL DL such that RDF data must adhere to certain non-

Birte Glimm, Aidan Hogan, Markus Krötzsch and Axel Polleres. "OWL: YET TO ARRIVE ON THE WEB OF DATA". In the Proceedings of the 4th Linked Data on the Web WWW2012 Workshop (LDOW 2012), Lyon, France, 16 April, 2012.

Lecture Roadmap

- Scope/Motivation
  (Axel)
- Short Introduction to RDFS+OWL
  (Aidan)
- RDFS+OWL usage in Linked Data
  (Aidan)
- **High-level Reasoning approaches: Query rewriting vs. Materialization**
  (Axel)
- Challenges on Reasoning over Linked Data
  (Axel)



- Practical approaches for Reasoning over Linked Data
  - Quarantined & Authoritative Materialization *(Aidan)*
  - Link-Traversal Based Query Execution with Reasoning *(Aidan)*
  - Reasoning with Attribute Equations *(Axel)*
- Wrap-up/Outlook *(all)*

2 overall approaches for Reasoning with RDF(S) & OWL for query answering as per W3C specs:

reasoning by query rewriting

- OWL2 QL:
  http://www.w3.org/TR/owl2-profiles/#OWL_2_QL

reasoning by rule-based materialization

- RDFS inference rules
  http://www.w3.org/TR/rdf-mt/#rules
- OWL 2 RL:
  http://www.w3.org/TR/owl2-profiles/#OWL_2_RL

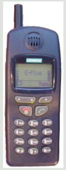
Back to our example:

"Companies who produce(d) telephones"

```
SELECT ?C {   ?C a dbo:Organisation.
              ?C foaf:made ?P .
              ?P a yago:Telephone .       }
```

dbo:manufacturer rdfs:subPropertyOf foaf:maker.
foaf:maker owl:inverseOf foaf:made.
dbo:manufacturer rdfs:range dbo:Organisation.
yago:SiemensMobilePhones rdfs:subClassOf yago:Telephone .
yago:IBMMobilePhones rdfs:subClassOf yago:Telephone .

dbr:Siemens_C25 dbo:manufacturer dbr:Siemens .
dbr:Siemens_C25 a yago:SiemensMobilePhones .



dbr:IBM_Simon dbo:manufacturer dbr:IBM .
dbr:IBM_Simon a yago:IBMMobilePhones .

SIEMENS DERI

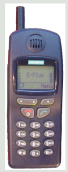
Approach 1: Query rewriting

"Companies who produce(d) telephones"

```
SELECT ?C { ?C a dbo:Organisation.
            ?C foaf:made ?P .
            ?P a yago:Telephone . }
```

dbo:manufacturer rdfs:subPropertyOf foaf:maker.
foaf:maker owl:inverseOf foaf:made.
dbo:manufacturer rdfs:range dbo:Organisation.
yago:SiemensMobilePhones rdfs:subClassOf yago:Telephone .
yago:IBMMobilePhones rdfs:subClassOf yago:Telephone .

dbr:Siemens_C25 dbo:manufacturer dbr:Siemens .
dbr:Siemens_C25 rdf:type yago:SiemensMobilePhones .



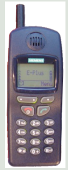
Approach 1: Query rewriting - informally

"Companies who produce(d) telephones"

```
SELECT ?C {  {{ ?C a dbo:Organisation. } UNION  { [] dbo:manufacturer ?C . }}
             ?C foaf:made ?P .
             ?P a yago:Telephone . }
```

dbo:manufacturer rdfs:subPropertyOf foaf:maker.
foaf:maker owl:inverseOf foaf:made.
dbo:manufacturer rdfs:range dbo:Organisation.
yago:SiemensMobilePhones rdfs:subClassOf   yago:Telephone .
yago:IBMMobilePhones rdfs:subClassOf   yago:Telephone .

dbr:Siemens_C25 dbo:manufacturer dbr:Siemens .
dbr:Siemens_C25 rdf:type yago:SiemensMobilePhones .



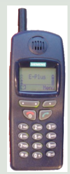
Approach 1: Query rewriting - informally

"Companies who produce(d) telephones"

SELECT ?C { {{ ?C a dbo:Organisation. } UNION { [] dbo:manufacturer ?C . }}
{{ ?C foaf:made ?P . } UNION { ?P foaf:maker ?C . } UNION { ?P dbo:manufacturer ?C . }}
?P a yago:Telephone . }

dbo:manufacturer rdfs:subPropertyOf foaf:maker.
foaf:maker owl:inverseOf foaf:made.
dbo:manufacturer rdfs:range dbo:Organisation.
yago:SiemensMobilePhones rdfs:subClassOf yago:Telephone .
yago:IBMMobilePhones rdfs:subClassOf yago:Telephone .

dbr:Siemens_C25 dbo:manufacturer dbr:Siemens .
dbr:Siemens_C25 rdf:type yago:SiemensMobilePhones .



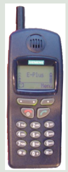
Approach 1: Query rewriting - informally

"Companies who produce(d) telephones"

SELECT ?C { {{ ?C a dbo:Organisation. } UNION { [] dbo:manufacturer ?C . }}
{{ ?C foaf:made ?P . } UNION { ?P foaf:maker ?C . } UNION { ?P dbo:manufacturer ?C . }}
{{?P a yago:Telephone . } UNION {?P a yago:SiemensMobilePhones . } UNION {?P a yago:IBMMobilePhones . }} }

dbo:manufacturer rdfs:subPropertyOf foaf:maker.
foaf:maker owl:inverseOf foaf:made.
dbo:manufacturer rdfs:range dbo:Organisation.
yago:SiemensMobilePhones rdfs:subClassOf yago:Telephone .
yago:IBMMobilePhones rdfs:subClassOf yago:Telephone .

dbr:Siemens_C25 dbo:manufacturer dbr:Siemens .
dbr:Siemens_C25 rdf:type yago:SiemensMobilePhones .



SIEMENS DERI

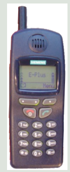
Approach 1: Query rewriting - informally

"Companies who produce(d) telephones"

SELECT ?C { {{ ?C a dbo:Organisation. } UNION { [] dbo:manufacturer ?C . }}
{{ ?C foaf:made ?P . } UNION { ?P foaf:maker ?C . } UNION { ?P dbo:manufacturer ?C . }}
{{?P a yago:Telephone . } UNION {?P a yago:SiemensMobilePhones . } UNION {?P a yago:IBMMobilePhones . }} }

dbo:manufacturer rdfs:subPropertyOf foaf:maker.
foaf:maker owl:inverseOf foaf:made.
dbo:manufacturer rdfs:range dbo:Organisation.
yago:SiemensMobilePhones rdfs:subClassOf yago:Telephone .
yago:IBMMobilePhones rdfs:subClassOf yago:Telephone .

dbr:Siemens_C25 dbo:manufacturer dbr:Siemens .
dbr:Siemens_C25 a yago:SiemensMobilePhones .



?C
dbr:Siemens
dbr:IBM

Approach 1: Query rewriting - formally

"Companies who produce(d) telephones"

SELECT ?C { ?C a dbo:Organisation.
?C foaf:made ?P .
?P a yago:Telephone . }

$q(C) \leftarrow Organisation(C), made(C,P), Telephone(P)$
$q(C) \leftarrow manufacturer(\_,C), made(C,P), Telephone(P)$
$q(C) \leftarrow Organisation(C), maker(P,C), Telephone(P)$
$\ldots$

dbo:manufacturer rdfs:subPropertyOf foaf:maker.
foaf:maker owl:inverseOf foaf:made.
dbo:manufacturer rdfs:range dbo:Organisation.
yago:SiemensMobilePhones rdfs:subClassOf yago:Telephone .
yago:IBMMobilePhones rdfs:subClassOf yago:Telephone .

$maker(X,Y) \leftarrow manufacturer(X,Y)$
$made(X,Y) \leftarrow maker(Y,X)$
$Organisation(X) \leftarrow manufacturer(\_,X)$
$Telephone(X) \leftarrow SiemensMobile(X)$
$Telephone(X) \leftarrow IBMMobile(X)$

Rewrite SPARQL BGPs to CQs

Algorithm *PerfectRef*$(Q, \mathcal{T}_P)$
Input: union of conjunctive queries Q, set of *DL-Lite*$_\mathcal{A}$ PIs $\mathcal{T}_P$
Output: union of conjunctive queries PR
$PR := Q$;
repeat
  $PR' := PR$;
  for each $q \in PR'$ **do**
    for each g in q **do**
      for each PI I in $\mathcal{T}_P$ **do**
        if I is applicable to g **then** $PR := PR \cup \{ ApplyPI(q, g, I) \}$;
    for each g_1, g_2 in q **do**
      if g_1 and g_2 unify **then** $PR := PR \cup \{\tau(Reduce(q, g_1, g_2))\}$;
until $PR' = PR$;
return PR

Apply a query rewriting algorithm for DL-Lite…

… e.g. [Calvanese,2007]

Translate resulting UCQ back to SPARQL UNION query.

Approach 1: Query rewriting pros/cons

Pro:

- Efficient for small hierarchies, lots of research from OBDA community (see lecture XYZ)
- Can handle Complex queries with non-distinguished variables and complex ontological statements for , e.g.

```
:Company rdfs:subClassOf [a owl:Restriction;
                          owl:onProperty hasEmployee;
                          owl:someValuesFrom owl:Thing . ]
```

(not compatible with SPARQL1.1 Entailment semantics, though)

Con:

- Not compatible with "arbitrary" RDF, e.g.

```
my:subClassOf rdfs:subPropertyof rdfs:subClassOf
```

- Works only for restricted ontologies, i.e. DL-Lite (e.g. no owl:sameAs, no owl:TransitiveProperty)
- Works only for restricted SPARQL queries, e.g. no variables in predicate positions

```
SELECT { dbr:SAP ?P ?O  UNION ?S ?P dbr:SAP }
```

not allowed

Approach 2: (rule-based) materialization - informally

"Companies who produce(d) telephones"

SELECT ?C { ?C a dbo:Organisation. ?C foaf:made ?P . ?P a yago:Telephone . }

dbo:manufacturer rdfs:subPropertyOf foaf:maker.
foaf:maker owl:inverseOf foaf:made.
dbo:manufacturer rdfs:range dbo:Organisation.
yago:SiemensMobilePhones rdfs:subClassOf yago:Telephone .
yago:IBMMobilePhones rdfs:subClassOf yago:Telephone .

e.g. http://www.w3.org/TR/rdf-mt/#rules

rdfs3	aaa rdfs:range xxx . uuu aaa vvv .	vvv rdf:type xxx .
rdfs7	aaa rdfs:subPropertyOf bbb . uuu aaa yyy .	uuu bbb yyy .
rdfs9	uuu rdfs:subClassOf xxx . vvv rdf:type uuu .	vvv rdf:type xxx .

dbr:IBM_Simon dbo:manufacturer dbr:IBM .
dbr:IBM_Simon rdf:type yago:IBMMobilePhones .
dbr:IBM a dbo:Organisation.
dbr:IBM_Simon foaf:maker dbr:IBM .
dbr:IBM_Simon rdf:type yago:Telephone .

Approach 2: (rule-based) materialization - informally

"Companies who produce(d) telephones"

SELECT ?C { ?C a dbo:Organisation. ?C foaf:made ?P . ?P a yago:Telephone . }

dbo:manufacturer rdfs:subPropertyOf foaf:maker.
foaf:maker owl:inverseOf foaf:made.
dbo:manufacturer rdfs:range dbo:Organisation.
yago:SiemensMobilePhones rdfs:subClassOf yago:Telephone .
yago:IBMMobilePhones rdfs:subClassOf yago:Telephone .

e.g. http://www.w3.org/TR/owl2-profiles/#Reasoning_in_OWL_2_RL_and_RDF_Graphs_using_Rules

prp-rng	$T(O, \text{rdf:type}, C)$	$\leftarrow$	$T(P, \text{rdfs:range}, C), T(S, P, O)$
prp-spo1	$T(S, P_2, O)$	$\leftarrow$	$T(P_1, \text{rdfs:subPropertyOf}, P_2), T(S, P_1, O)$
cax-sco	$T(S, \text{a}, C_2)$	$\leftarrow$	$T(C_1, \text{rdfs:subClassOf}, C_2), T(S, a, C_1)$

dbr:IBM_Simon dbo:manufacturer dbr:IBM .
dbr:IBM_Simon rdf:type yago:IBMMobilePhones .
dbr:IBM a dbo:Organisation.
dbr:IBM_Simon foaf:maker dbr:IBM .
dbr:IBM_Simon rdf:type yago:Telephone .

Approach 2: (rule-based) materialization - informally

"Companies who produce(d) telephones"

SELECT ?C { ?C a dbo:Organisation. ?C foaf:made ?P . ?P a yago:Telephone . }

dbo:manufacturer rdfs:subPropertyOf foaf:maker.
foaf:maker owl:inverseOf foaf:made.
dbo:manufacturer rdfs:range dbo:Organisation.
yago:SiemensMobilePhones rdfs:subClassOf yago:Telephone .
yago:IBMMobilePhones rdfs:subClassOf yago:Telephone .

e.g. http://www.w3.org/TR/owl2-profiles/#Reasoning_in_OWL_2_RL_and_RDF_Graphs_using_Rules

$$\text{prp-rng} \quad (O, \text{rdf:type}, C) \leftarrow (P, \text{rdfs:range}, C), (S, P, O)$$
$$\text{prp-spo1} \quad (S, P_2, O) \leftarrow (P_1, \text{rdfs:subPropertyOf}, P_2), (S, P_1, O)$$
$$\text{cax-sco} \quad (S, \text{a}, C_2) \leftarrow (C_1, \text{rdfs:subClassOf}, C_2), (S, a, C_1)$$
$$\text{prp-inv1} \quad (O, P_2, S) \leftarrow (P_1, \text{owl:inverseOf}, P_2), (S, P_1, O)$$

dbr:IBM_Simon dbo:manufacturer dbr:IBM .
dbr:IBM_Simon rdf:type yago:IBMMobilePhones .
dbr:IBM a dbo:Organisation.
dbr:IBM_Simon foaf:maker dbr:IBM .
dbr:IBM_Simon rdf:type yago:Telephone .
dbr:IBM foaf:made dbr:IBM_Simon .

SIEMENS DERI

Approach 2: (rule-based) materialization - informally

"Companies who produce(d) telephones"

SELECT ?C { ?C a dbo:Organisation. ?C foaf:made ?P . ?P a yago:Telephone . }

dbo:manufacturer rdfs:subPropertyOf foaf:maker.
foaf:maker owl:inverseOf foaf:made.
dbo:manufacturer rdfs:range dbo:Organisation.
yago:SiemensMobilePhones rdfs:subClassOf yago:Telephone .
yago:IBMMobilePhones rdfs:subClassOf yago:Telephone .

e.g. http://www.w3.org/TR/owl2-profiles/#Reasoning_in_OWL_2_RL_and_RDF_Graphs_using_Rules

prp-rng	$(O, \text{rdf:type}, C)$	$\leftarrow$	$(P, \text{rdfs:range}, C), (S, P, O)$
prp-spo1	(S, P_2, O)	$\leftarrow$	$(P_1, \text{rdfs:subPropertyOf}, P_2), (S, P_1, O)$
cax-sco	(S, a, C_2)	$\leftarrow$	$(C_1, \text{rdfs:subClassOf}, C_2), (S, a, C_1)$
prp-inv1	(O, P_2, S)	$\leftarrow$	$(P_1, \text{owl:inverseOf}, P_2), (S, P_1, O)$

?C
dbr:Siemens
dbr:IBM

dbr:IBM_Simon dbo:manufacturer dbr:IBM .
dbr:IBM_Simon rdf:type yago:IBMMobilePhones .
dbr:IBM a dbo:Organisation.
dbr:IBM_Simon foaf:maker dbr:IBM .
dbr:IBM_Simon rdf:type yago:Telephone .
dbr:IBM foaf:made dbr:IBM_Simon .

Approach 2: (rule-based) materialization - informally

Alternative for more efficient execution: ***compile RDFS/OWL axioms into a specific ruleset***
(more efficient materialization for small ontologies ("Tbox") and big amounts of data "Abox")

dbo:manufacturer rdfs:subPropertyOf foaf:maker.
foaf:maker owl:inverseOf foaf:made.
dbo:manufacturer rdfs:range dbo:Organisation.
yago:SiemensMobilePhones rdfs:subClassOf yago:Telephone .
yago:IBMMobilePhones rdfs:subClassOf yago:Telephone .

e.g. in "OWL 2 RL in RIF"
http://www.w3.org/TR/rif-owl-rl/#templateRules_algorithm

Works very effectively for rules with only on Tbox-pattern and one Abox pattern, since these rules can be run "per statement" over the whole Abox.

Approach 2: (rule-based) materialization - informally

Alternative for more efficient execution: ***compile RDFS/OWL axioms into a specific ruleset*** *(more efficient materialization for small ontologies ("Tbox") and big amounts of data "Abox")*

dbo:manufacturer rdfs:subPropertyOf foaf:maker.
foaf:maker owl:inverseOf foaf:made.
dbo:manufacturer rdfs:range dbo:Organisation.
yago:SiemensMobilePhones rdfs:subClassOf yago:Telephone .
yago:IBMMobilePhones rdfs:subClassOf yago:Telephone .

prp-rng	$(O, \text{rdf:type}, C)$	$\leftarrow$	$(P, \text{rdfs:range}, C), (S, P, O)$
prp-spo1	(S, P_2, O)	$\leftarrow$	$(P_1, \text{rdfs:subPropertyOf}, P_2), (S, P_1, O)$
cax-sco	(S, a, C_2)	$\leftarrow$	$(C_1, \text{rdfs:subClassOf}, C_2), (S, a, C_1)$
prp-inv1	(O, P_2, S)	$\leftarrow$	$(P_1, \text{owl:inverseOf}, P_2), (S, P_1, O)$

$$(O, \text{rdf:type}, \text{dbo:Organisation}) \leftarrow (S, \text{dbo:manufactuer}, O)$$
$$(S, \text{foaf:maker}, O) \leftarrow (S, \text{dbo:manufactuer}, O)$$
$$(S, \text{rdf:type}, \text{yago:Telephone}) \leftarrow (S, \text{rdf:type}, \text{yago:SiemensMobilePhones})$$
$$(S, \text{rdf:type}, \text{yago:Telephone}) \leftarrow (S, \text{rdf:type}, \text{yago:IBMMobilePhones})$$
$$(O, \text{foaf:maker}, S) \leftarrow (S, \text{foaf:made}, O)$$
$$(O, \text{foaf:made}, S) \leftarrow (S, \text{foaf:maker}, O)$$

Works very effectively for rules with only on Tbox-pattern and one Abox pattern, since these rules can be run "per statement" over the whole Abox.

Approach 2: rule-based materialization pros/cons

Pro:

- Compatible with arbitrary RDF: OWL2 RL rules extend RDFS Entailment rules straightforwardly.
- Materialization computable with off-the-shelf **Datalog** engines (simple fixpoint computation)
- Covers features prominently used in LD, such as e.g. or owl:transitiveProperty …

$$\text{prp-trp} \quad (X, P, Z) \quad \leftarrow \quad (P, \text{type}, \text{owl:TransitiveProperty}), (X, P, Y), (Y, P, Z)$$

  … owl:sameAs

$$\text{eq-rep-s} \quad (S', P, O) \quad \leftarrow \quad (S, \text{owl:sameAs}, S'), (S, P, O)$$
$$\text{eq-rep-p} \quad (S, P', O) \quad \leftarrow \quad (P, \text{owl:sameAs}, P'), (S, P, O)$$
$$\text{eq-rep-o} \quad (S, P, O') \quad \leftarrow \quad (O, \text{owl:sameAs}, O'), (S, P, O)$$

  … plus rules for transitivity and reflexivity of owl:sameas…

  … but these rule fall outside the efficient compilation technique mentioned in the prev. slide.
- Covers most features of QL as well

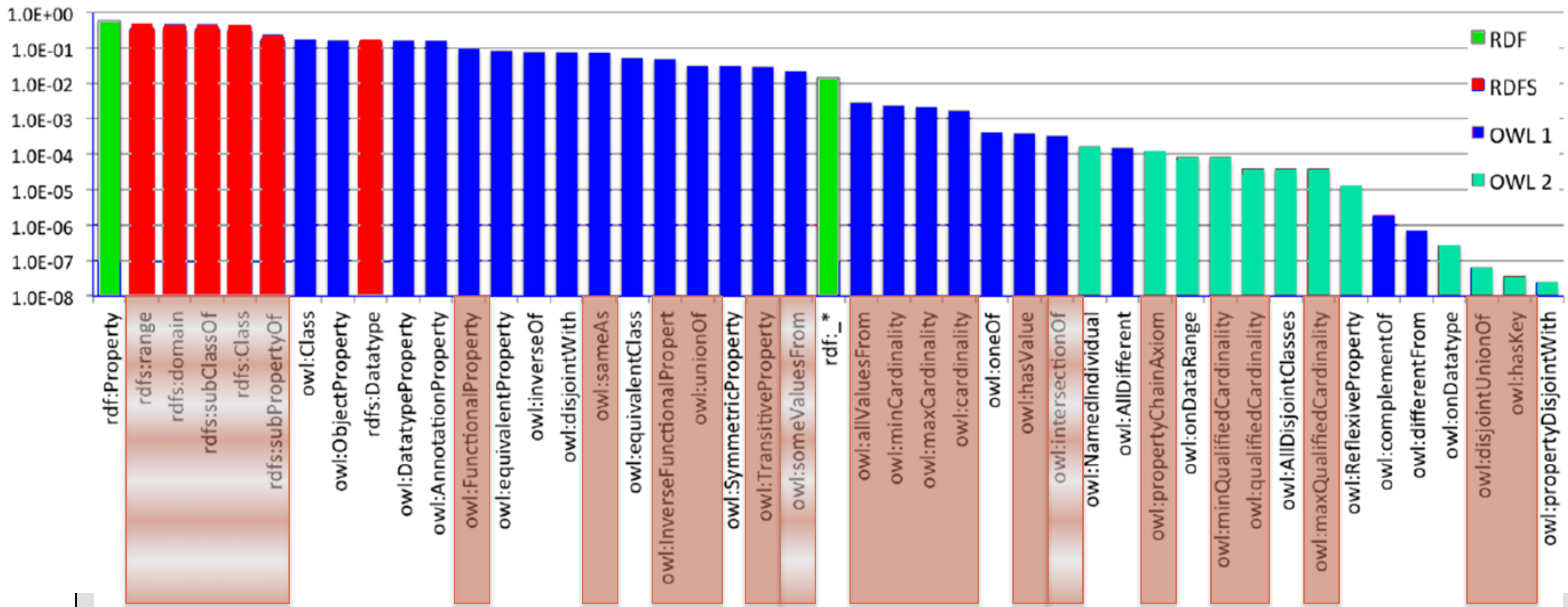
Con:

- Materialization might still be prohibitively expensive in practical settings… see following slides

SIEMENS DERI

What's needed for Linked Data

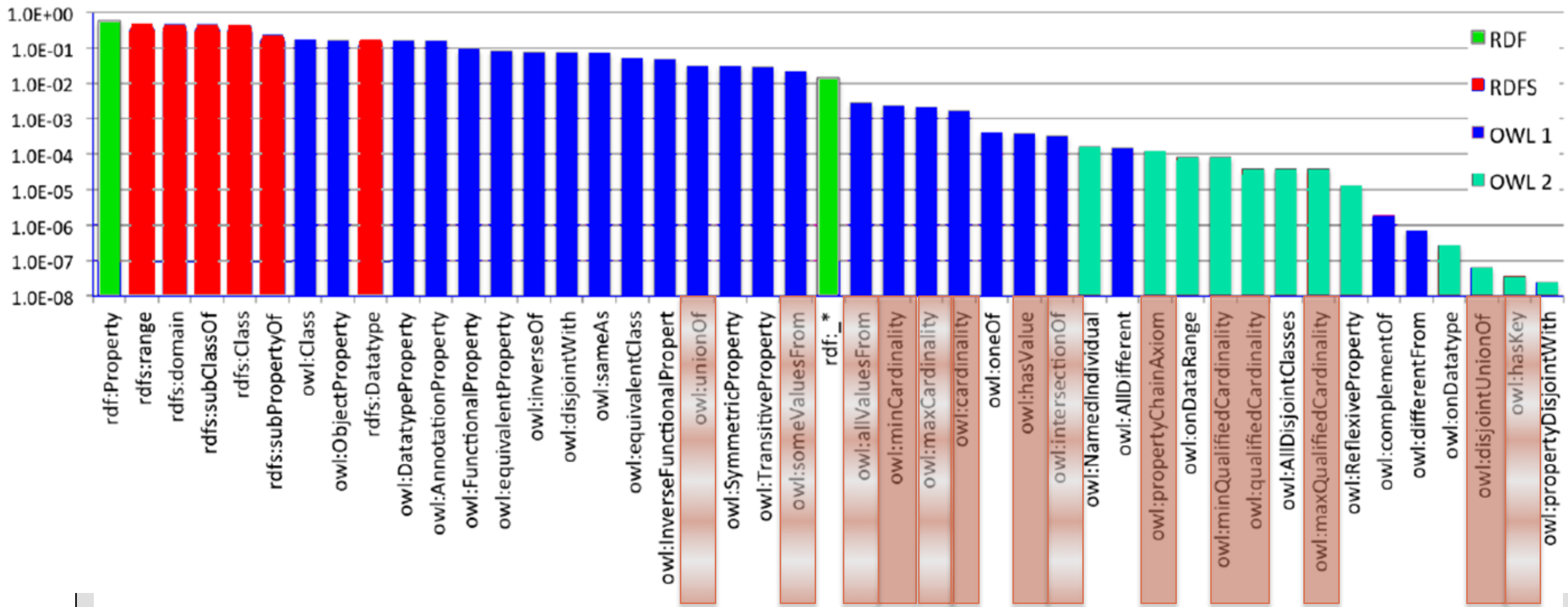
- **OWL 2 QL** … not/partially supported in QL
- OWL 2 RL



SIEMENS DERI

What’s needed for Linked Data

- OWL 2 QL
- **OWL 2 RL** [legend box] / [striped legend box] **… not/partially supported in RL**



SIEMENS DERI

Lecture Roadmap

- Scope/Motivation
  (Axel)
- Short Introduction to RDFS+OWL
  (Aidan)
- RDFS+OWL usage in Linked Data
  (Aidan)
- High-level Reasoning approaches: Query rewriting vs. Materialization
  (Axel)
- **Challenges on Reasoning over Linked Data**
  (Axel)
- Practical approaches for Reasoning over Linked Data
  - Quarantined & Authoritative Materialization *(Aidan)*
  - Link-Traversal Based Query Execution with Reasoning *(Aidan)*
  - Reasoning with Attribute Equations *(Axel)*
- Wrap-up/Outlook *(all)*

?

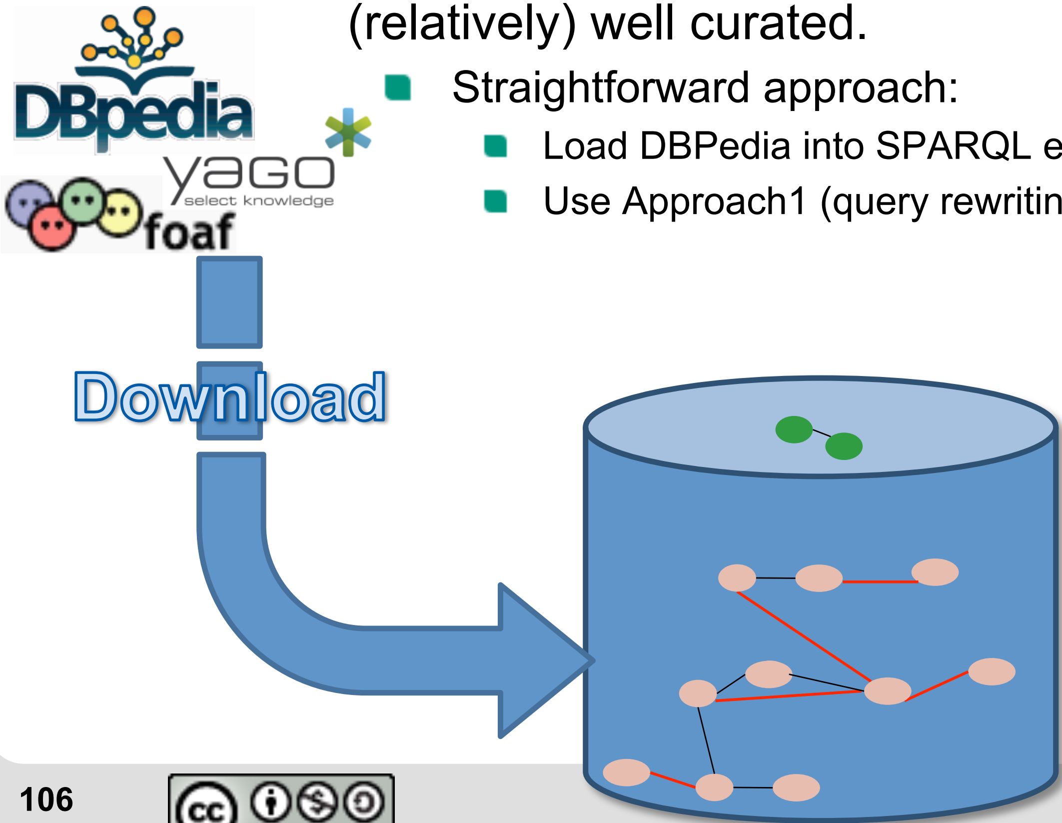
SIEMENS DERI

Examples:

Google “Companies who produce(d) telephones”

```
SELECT ?C { ?C a dbo:Organisation. ?C
foaf:made ?P . ?P a yago:Telephone . }
```

- Now that was relatively easy…
- … all data available on one source (DBpedia) which is (relatively) well curated.
  - Straightforward approach:
    - Load DBPedia into SPARQL engine
    - Use Approach1 (query rewriting) or Approach2 (materialization)



SIEMENS DERI

What about the other Examples?

Google "**Latest** News on NYT about companies with a revenue greater than 10B **EUR**"

- What if…

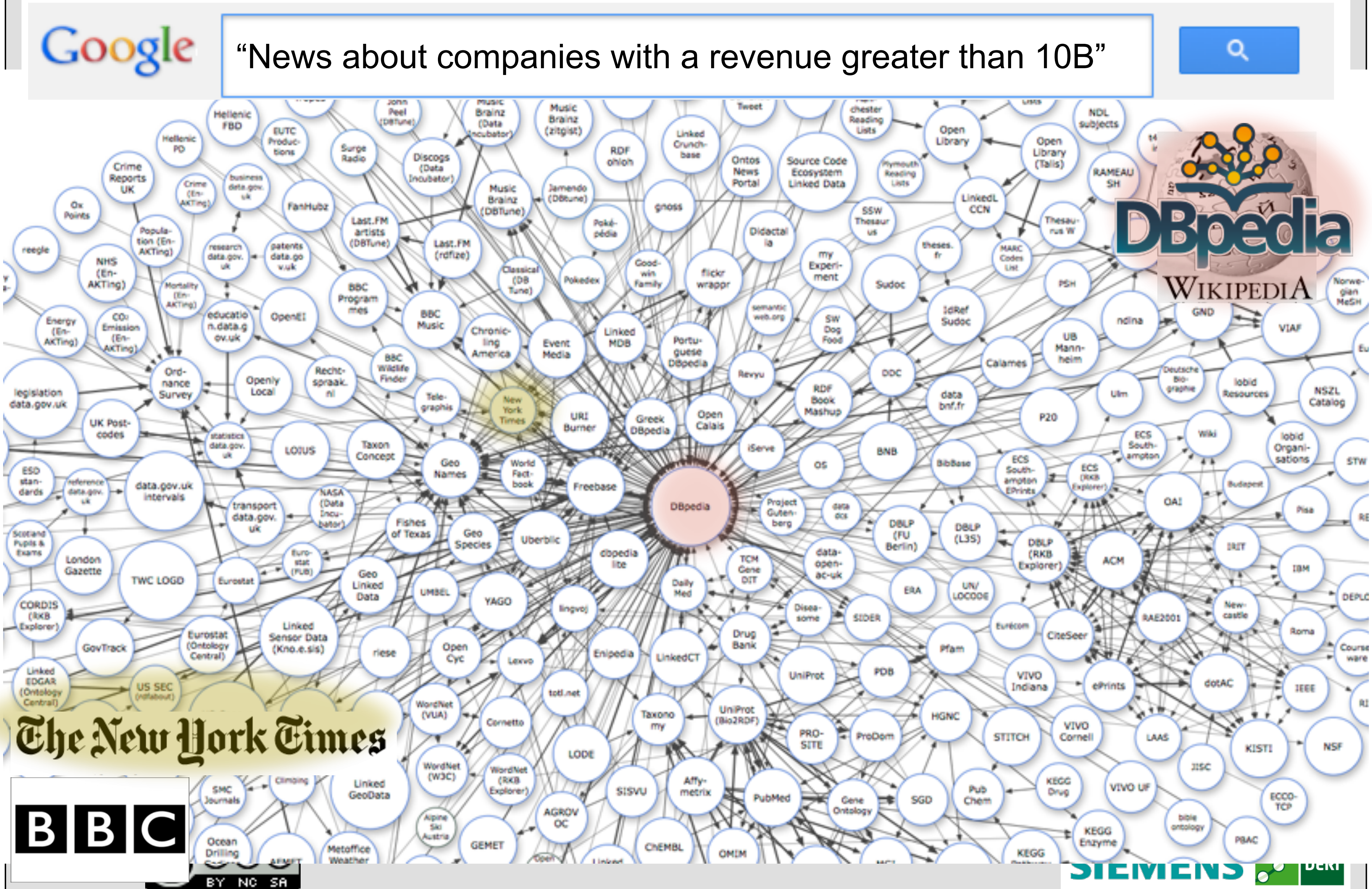
  … there are several datasets from the Linked Data Cloud involved?

  - Which datasets do I need?
  - Which ontologies should I use?

  … datasets change?

  … if implicit data is not expressible in OWL (e.g. 1 **USD** = 0,75 **EUR**) ?

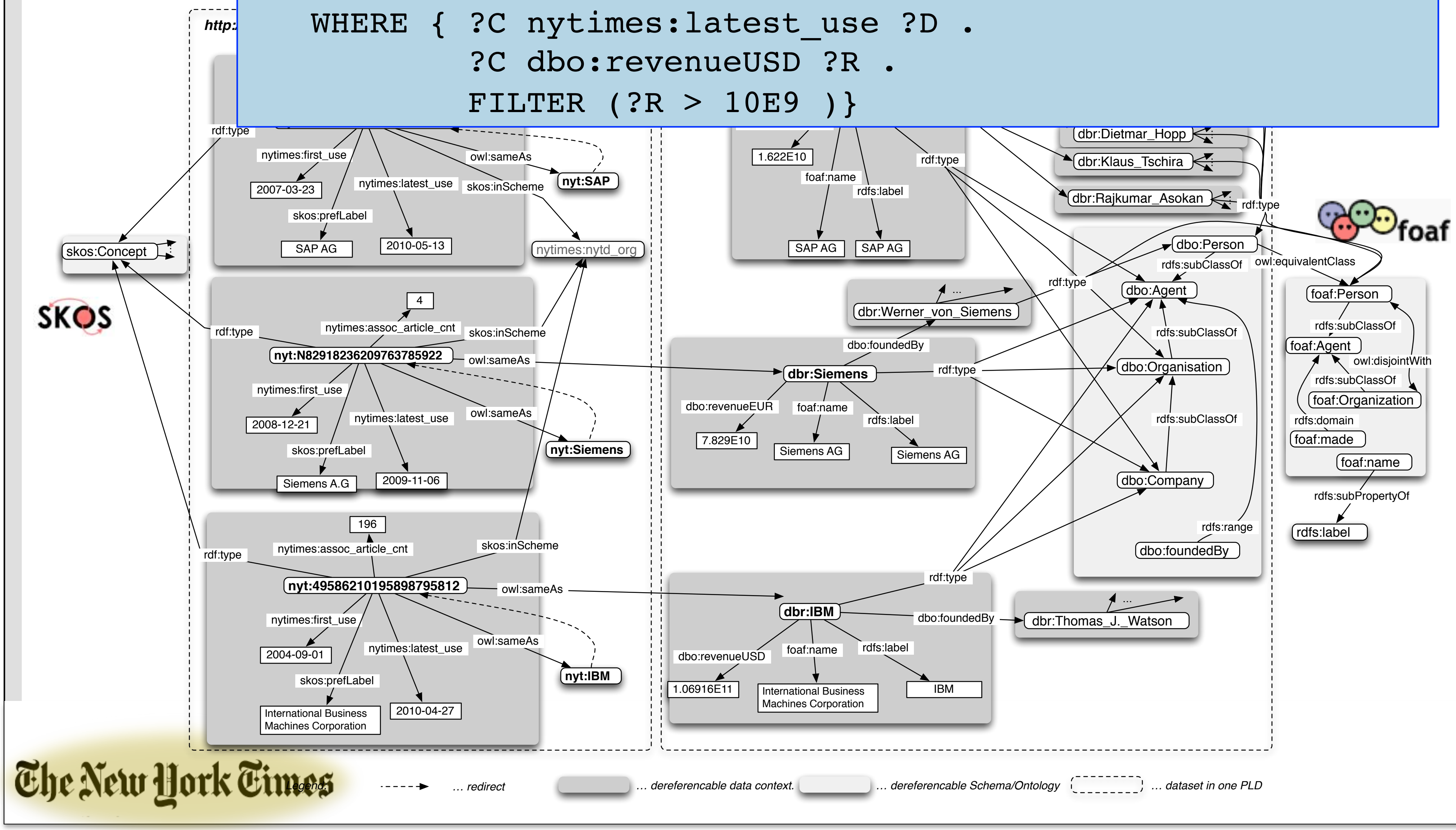
Several datasets and involved…



Several datasets & ontologies and involved…

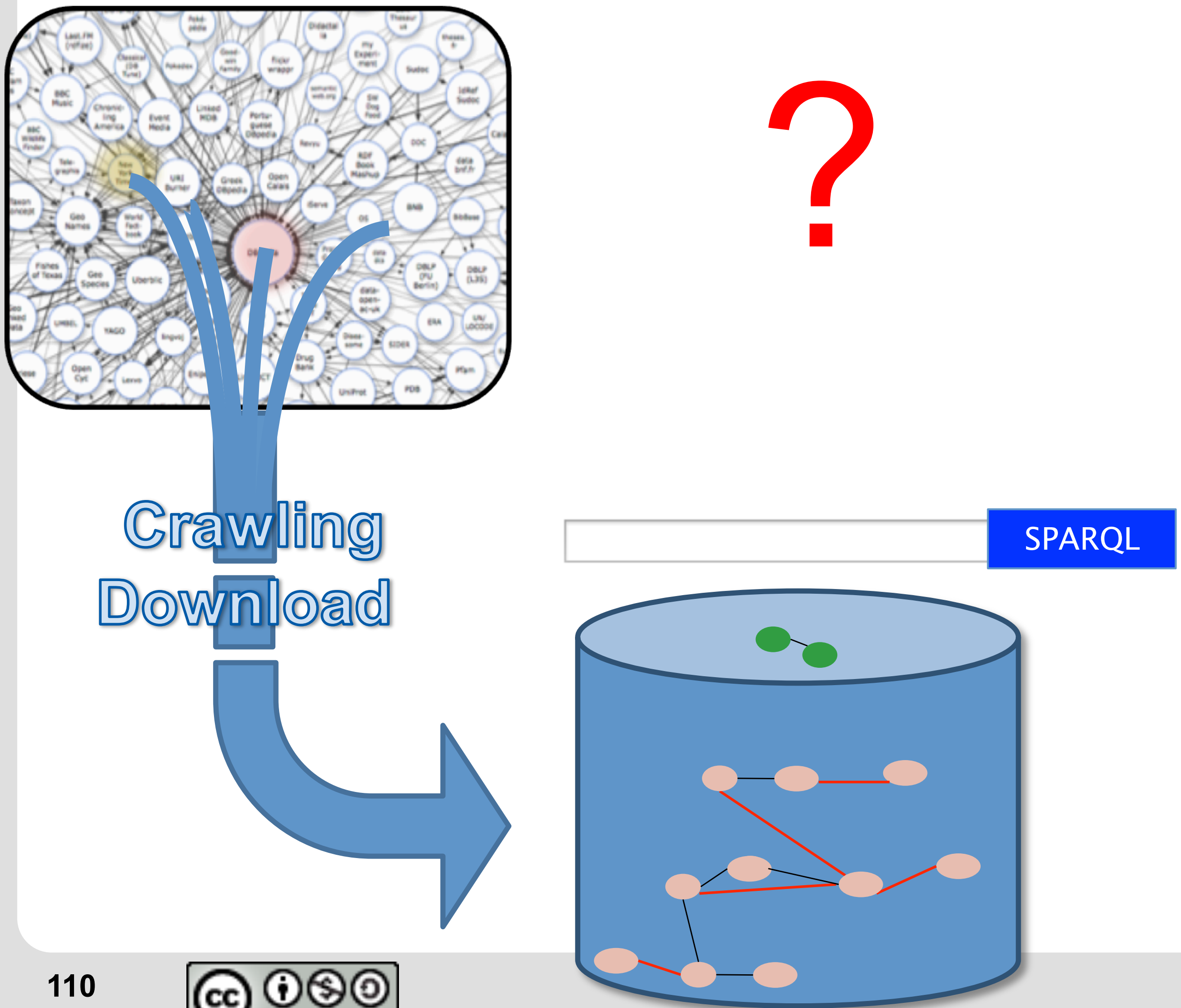
"News on NYT about companies with a revenue greater than 10B"

```
SELECT ?C ?D
WHERE { ?C nytimes:latest_use ?D .
        ?C dbo:revenueUSD ?R .
        FILTER (?R > 10E9 )}
```



Naïve extension:

- Could I use the same approach for arbitrary queries over linked data?
- Some challenges ahead…



Challenges:

“Some of the challenges for the Semantic Web include vastness, vagueness, uncertainty, inconsistency, and deceit. Automated reasoning systems will have to deal with all of these issues in order to deliver on the promise of the Semantic Web.” (from: https://en.wikipedia.org/wiki/Semantic_Web)

WIKIPEDIA

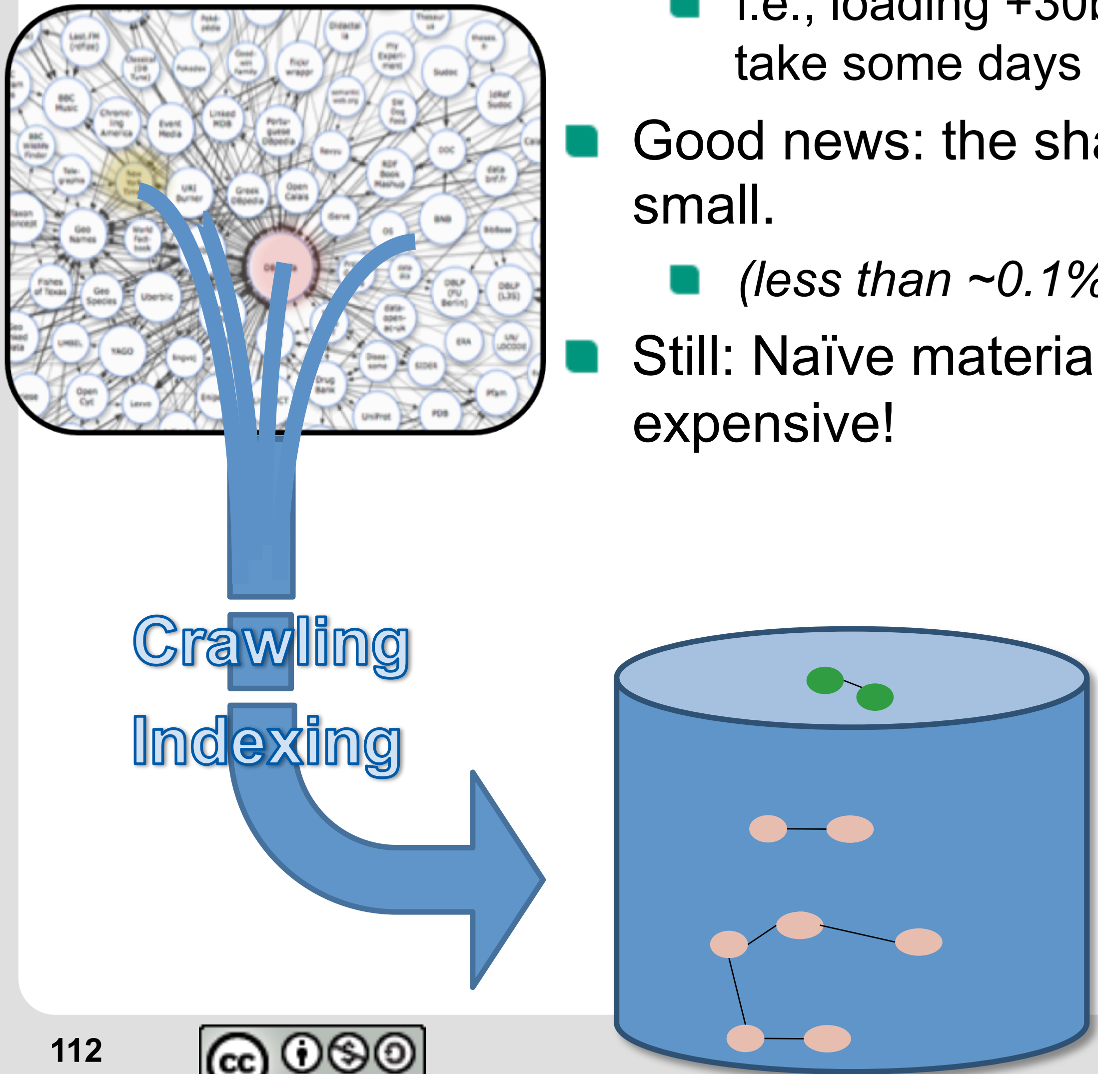
More specifically for RDFS+OWL Reasoning over Linked Data:

- **C1** Linked Data is **huge**
  - LOD Cloud: +30b triples/+300 Datasets cf. http://datahub.io/group/lodcloud)
  - WebDataCommons Crawl: +7,3b triples from +2m domains)
- **C2** Linked Data is not “pure” OWL (DL)
- **C3** Linked Data is **NOT consistent**
- **C4** Linked Data is evolving
- **C5** Linked Data Reasoning needs more than RDFS+OWL

→ *Reasoning for Linked Data should be designed in a way to overcome these challenges (rather than stumbling over them)*

C1 Linked Data is huge 1/2

- Not only crawling time, but also indexing time needs to be considered…
  - i.e., loading +30b triples into a normal triple can take some days
- Good news: the share of Ontologies is relatively small.
  - *(less than ~0.1% [IJSWIS'2009, Hogan 2010])*
- Still: Naïve materialization can be prohibitively expensive!



C1 Linked Data is huge 2/2

- Examples of "expensive" OWL features
  e.g. ow:sameAs, owl:TransitiveProperty

Example 1: my: ontology "hijacking" foaf

```
foaf:knows rdfs:subProperty my:foafClosure .
my:foafClosure a owl:SymmetricProperty .
my:foafClosure a owl:TransitiveProperty .
```

... Computes the symmetric transitive closure of foaf:knows

… imagine this applied to a FOAF export of facebook (+1b users)
→ $O(n^2)$

Example 2: my: ontology redefining the rdfs: & owl: vocabularies:

```
rdf:type rdfs:subPropertyOf owl:sameAs .
rdf:type rdfs:range rdfs:Resource.
rdf:type rdfs:domain rdfs:Resource.
```

... Also called "non-standard use" of RDFS/OWL

… *essentially equates **any** resources, similar example*:
http://polleres.net/nasty.rdf → $O(n^3)$

C2 Linked Data is not "pure" OWL

- http://www.w3.org/TR/owl2-mapping-to-rdf/ defines which OWL DL ontologies are mappable to RDF…
- …but vice versa not all RDF Graphs using owl: and rdfs: vocabularies are mappable to OWL DL ontologies…

Example 1: foaf itself is not OWL DL

```
foaf:mbox_sha1sum a
      owl:DatatypeProperty ,
      owl:InverseFunctionalProperty .
```

… there is only inverseFunctional ***ObjectProperties*** *in OWL DL*

Example 2: *as before: "non-standard use" is not compatible with OWL DL*

```
rdf:type rdfs:subPropertyOf owl:sameAs .
rdf:type rdfs:range rdfs:Resource.
rdf:type rdfs:domain rdfs:Resource.
```

- *Good News: OWL2 RL is tolerant/robust against this!*

C3 Linked Data is NOT consistent 1/2

- Logical inconsistencies may happen "accidentially"

DBpedia

dbr:Siemens a dbo:Organisation.

foaf

foaf:Person owl:disjointWith foaf:Organisation.

My homepage:

:dbr:Siemens a foaf:Person .

SIEMENS DERI

C3 Linked Data is NOT consistent 1/2

- Logical inconsistencies may happen "accidentially"

DBpedia

```
dbr:Siemens a dbo:Organisation.
```

foaf

```
foaf:Person owl:disjointWith foaf:Organisation.
foaf:knows rdfs:range foaf:Person.
```

My homepage … a more innocuous looking example "I know Siemens"…

```
:me foaf:knows dbr:Siemens.
```

… such inconcistent misuses happen in practice!

C3 Linked Data is NOT consistent 2/2

- OWL 2 RL is again a good choice to start with:

DBpedia

dbr:Siemens a dbo:Organisation.

foaf

foaf:Person owl:disjointWith foaf:Organisation.

My homepage:

:dbr:Siemens a foaf:Person .

- OWL2 RL has a set of optional inconsistency detection rules, e.g. …

cax-dw' $Q(Src1, \text{:conflicts}, Src_2, Ont) \leftarrow Q(C_1, \text{owl:disjointWith}, C_2, Ont),$
$Q(S, \text{rdf:type}, C_1, Src_1), Q(S, rdf:type, C_2, Src_2)$

Idea: these rules could be either dropped, or extended to pinpoint to local inconsistencies … more on that later.

O-bombs …

BONUS MATERIAL:

SIEMENS DERI

Noisy Data: Omnipotent Being

Web data is noisy.

Proof:

08445a31a78661b5c746feff39a9db6e4e2cc5cf

- sha1-sum of **`'mailto:'`**
- common value for **`foaf:mbox_sha1sum`**
  - An inverse-functional (uniquely identifying) property!!!
  - Any person who shares the same value will be considered the same

Q.E.D.

Noisy Data: Redefining everything

More proof (courtesy of http://www.eiao.net/rdf/1.0)

```
rdf:type rdf:type owl:Property .
rdf:type rdfs:label “type”@en .
rdf:type rdfs:comment “Type of resource” .
rdf:type rdfs:domain eiao:testRun .
rdf:type rdfs:domain eiao:pageSurvey .
rdf:type rdfs:domain eiao:siteSurvey .
rdf:type rdfs:domain eiao:scenario .
rdf:type rdfs:domain eiao:rangeLocation .
rdf:type rdfs:domain eiao:startPointer .
rdf:type rdfs:domain eiao:endPointer .
rdf:type rdfs:domain eiao:header .
rdf:type rdfs:domain eiao:runs .
```

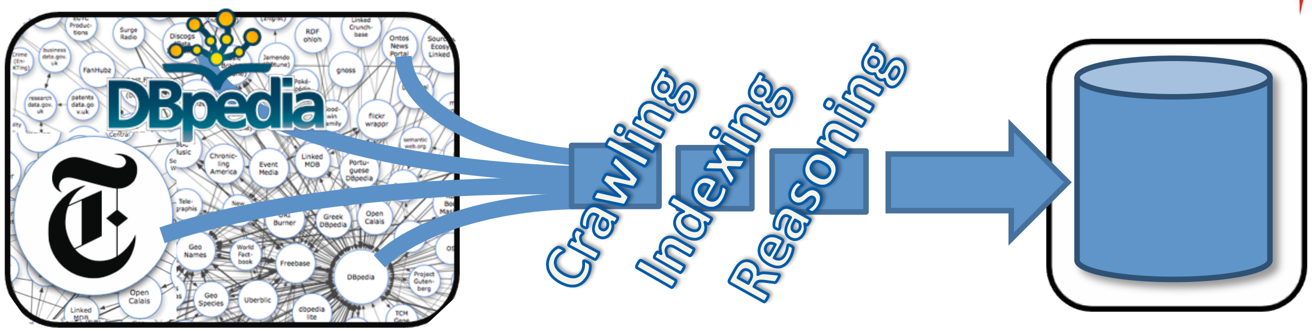
C4 Linked Data is evolving

- Crawling + Reasoning not really suitable for evolving linked data:

Google

"**Latest** News on NYT about companies with a revenue greater than 10B"

```
SELECT * WHERE
{ ?C rdf:type NYT:Org .
  ?C dbpedia:revenue ?R .
  ?C NYT:latestArticle ?A .
  FILTER( ?R > 1E10 )   }
```



Can't be sure that crawling keeps up with changes!

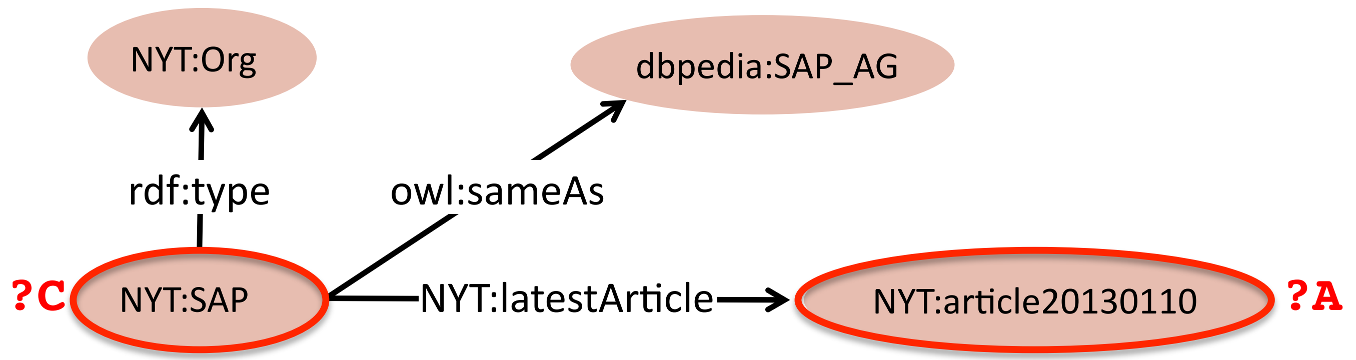
SIEMENS DERI

C4 Linked Data is evolving

- Basic idea (originally proposed by Hartig & Bizer): "Link-based-query-processing (LBQE)"
  - Start with URIs in a SPARQL query
  - Interleave crawl + query processing

```
SELECT * WHERE
{ ?C rdf:type NYT:Org .
  ?C dbpedia:revenue ?R .
  ?C NYT:latestArticle ?A.
  FILTER (?R > 1E10)   }
```

HTTP GET NYT:Org → HTTP GET NYT:SAP → HTTP GET NYT:article20130128_1



Stop. No new query relevant results found

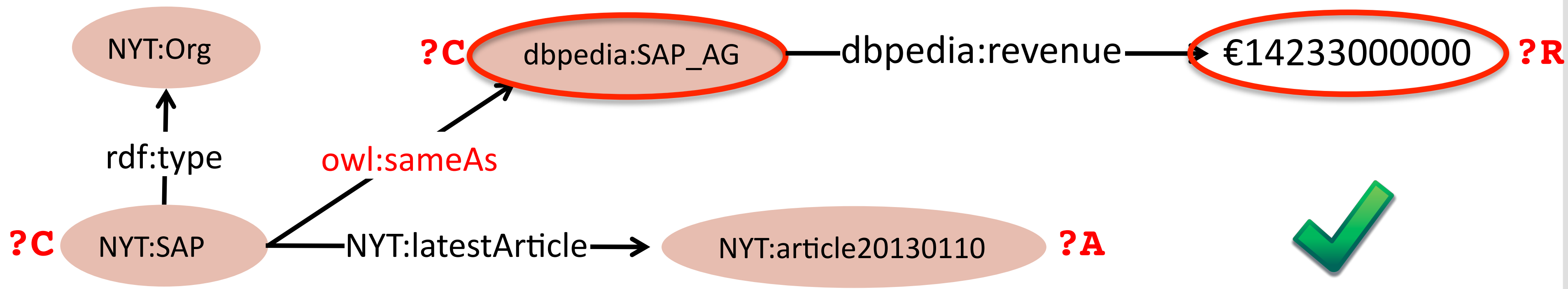
C4 Linked Data is evolving

- Basic idea (originally proposed by Hartig & Bizer): "Link-based-query-processing (LBQE)"
  - Start with URIs in a SPARQL query
  - Interleave crawl + query processing
    + OWL Reasoning

How?… **more on that later**

```
SELECT * WHERE
{ ?C rdf:type NYT:Org .
  ?C dbpedia:revenue ?R .
  ?C NYT:latestArticle ?A.
  FILTER (?R > 1E10)   }
```

HTTP GET NYT:Org → HTTP GET NYT:SAP → HTTP GET dbpedia:SAP



C5: Linked data needs more than RDFS+OWL

Google: "Latest News on NYT about companies with a revenue greater than 10B **EUR**"

```
SELECT * WHERE
{ ?C rdf:type NYT:Org .
  ?C dbpedia:revenueEUR ?R .
  ?C NYT:latestArticle ?A .
  FILTER( ?R > 1E10 )  }
```

- *There is implicit knowledge not expressible in OWL, e.g. in the form of so called **"attribute equations"***

dbo:revenueUSD = dbo:revenueEUR / 1.3 .
dbo:profitEUR = "dbo:revenueEUR – dbo:totalExpensesEUR".

dbr:Siemens a dbo:Organisation.
dbr:Siemens dbo:revenueEUR 7.829E10

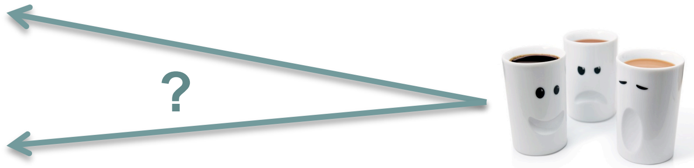
dbr:SAP a dbo:Organisation.
dbr:SAP dbo:revenueEUR 1.622E10

1.3 **USD** = 1**EUR**

dbr:IBM a dbo:Organisation.
dbr:IBM dbo:revenueUSD 1.06916E11

Lecture Roadmap

- Scope/Motivation
  (Axel)
- Short Introduction to RDFS+OWL
  (Aidan)
- RDFS+OWL usage in Linked Data
  (Aidan)
- High-level Reasoning approaches: Query rewriting vs. Materialization
  (Axel)
- Challenges on Reasoning over Linked Data
  (Axel)



- **Practical approaches for Reasoning over Linked Data**
  - Quarantined & Authoritative Materialization *(Aidan)*
  - Link-Traversal Based Query Execution with Reasoning *(Aidan)*
  - Reasoning with Attribute Equations *(Axel)*
- Wrap-up/Outlook *(all)*

Context-Dependant Reasoning

MATERIALISATION (I)

Sindice.com

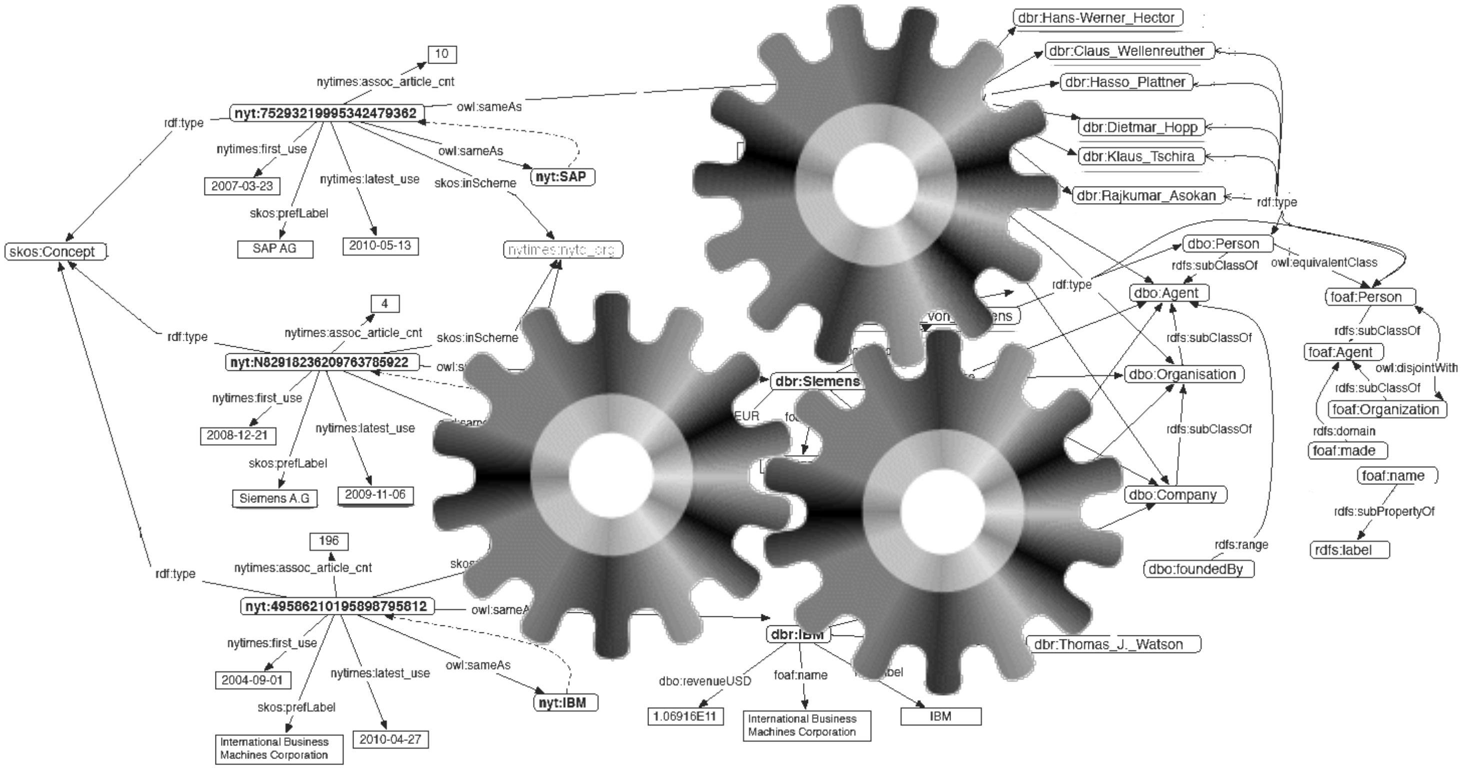
SIEMENS DERI

Reasoning over Web Data

- Challenges
  - C1: Linked Data is **huge**
  - C2: Linked Data is not "pure" OWL (DL)
  - C3: Linked Data is not consistent
    - Linked Data is Web data
  - C4: Linked Data is Evolving
  - C5: Linked Data needs more than RDFS+OWL

Context-Dependant Reasoning in Sindice

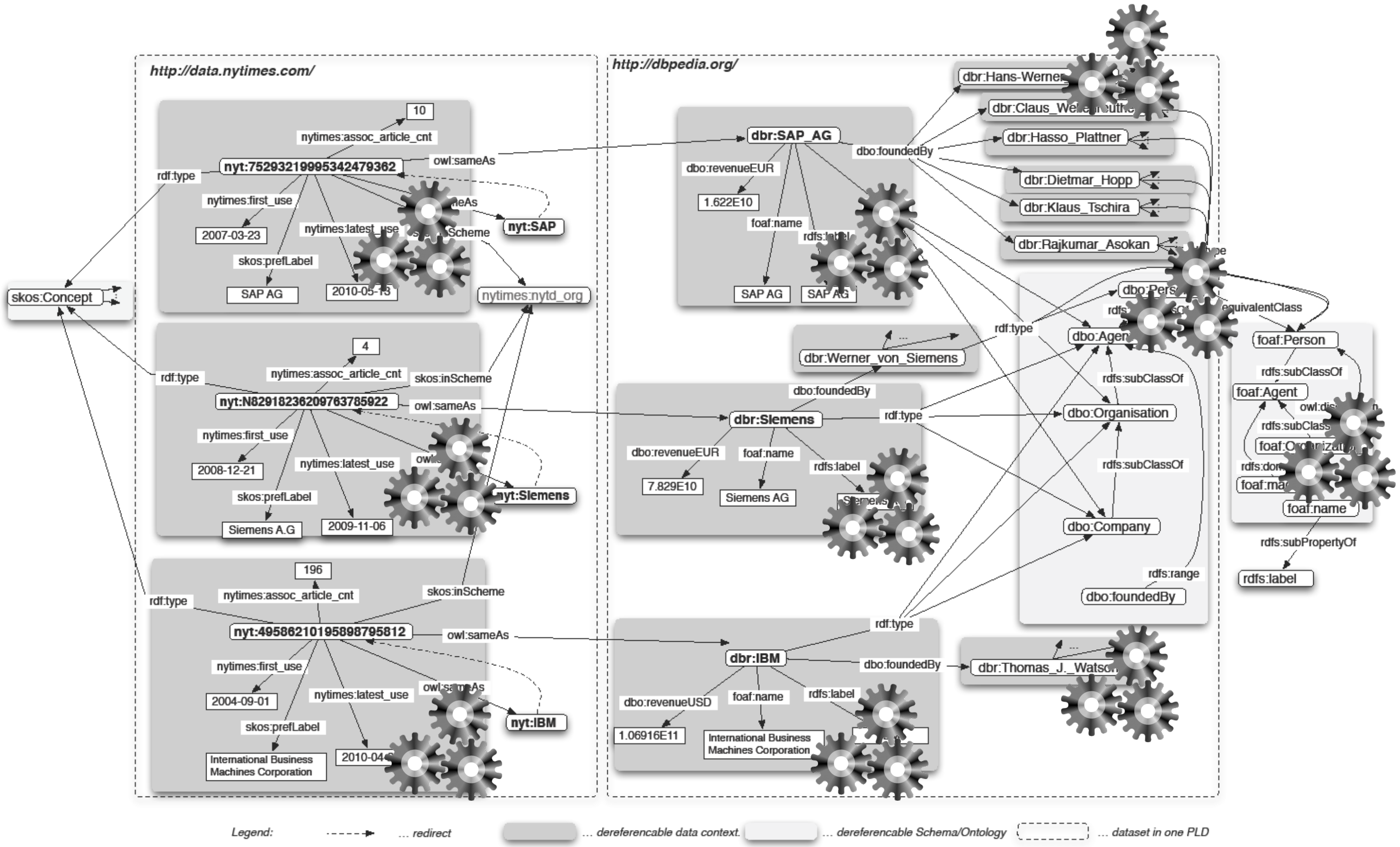
- C1 (Scale): Divide and conquer!
  - Break reasoning up into many small contexts



SIEMENS DERI

Context-Dependant Reasoning in Sindice

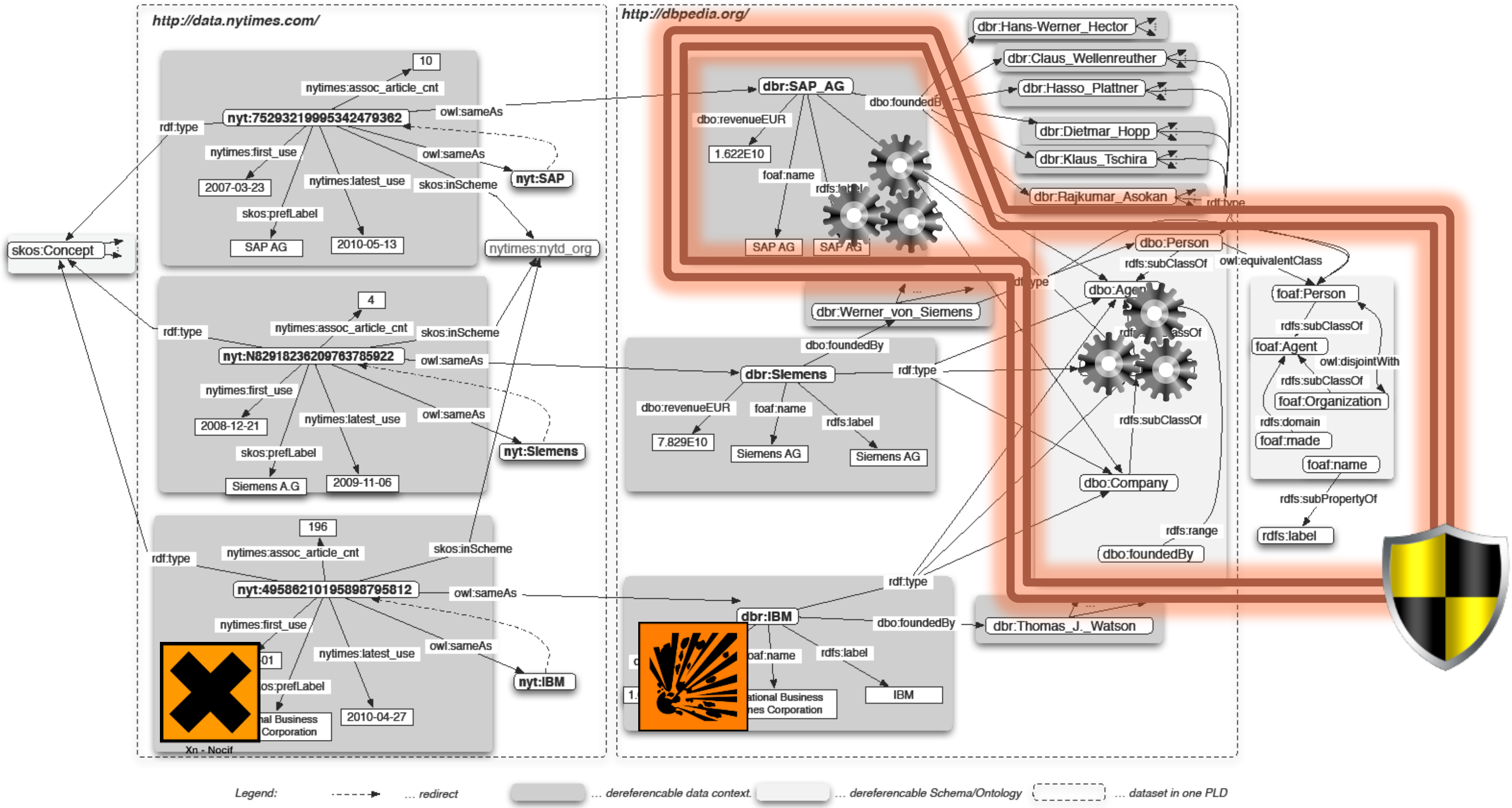
- C1 (Scale): Divide and conquer!
  - Break reasoning up into many small contexts



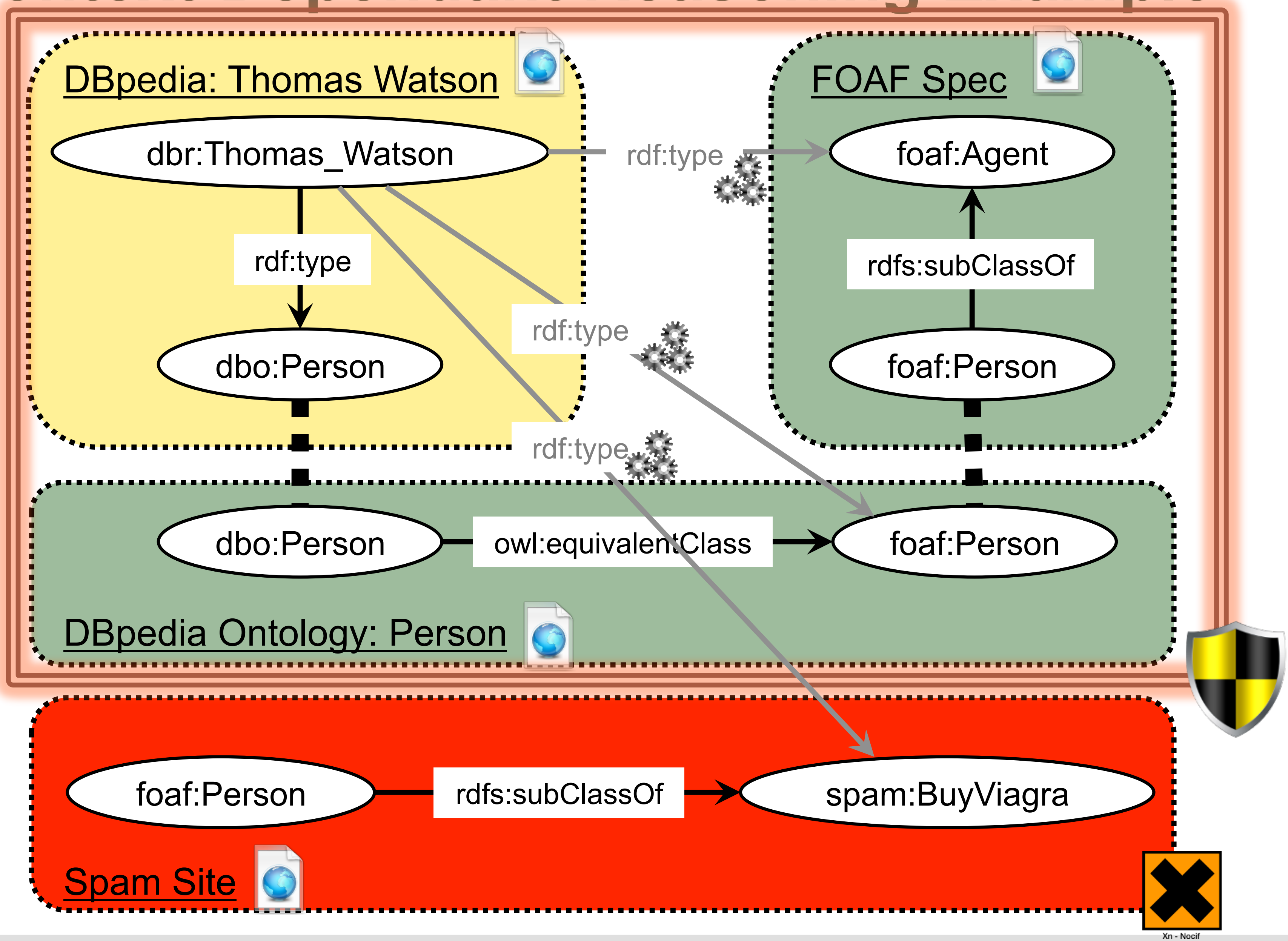
SIEMENS DERI

Context-Dependant Reasoning in Sindice

- C3 (Web Data): Keep contexts closed!
  - “Quarantine” reasoning within connected contexts



Context-Dependant Reasoning Example

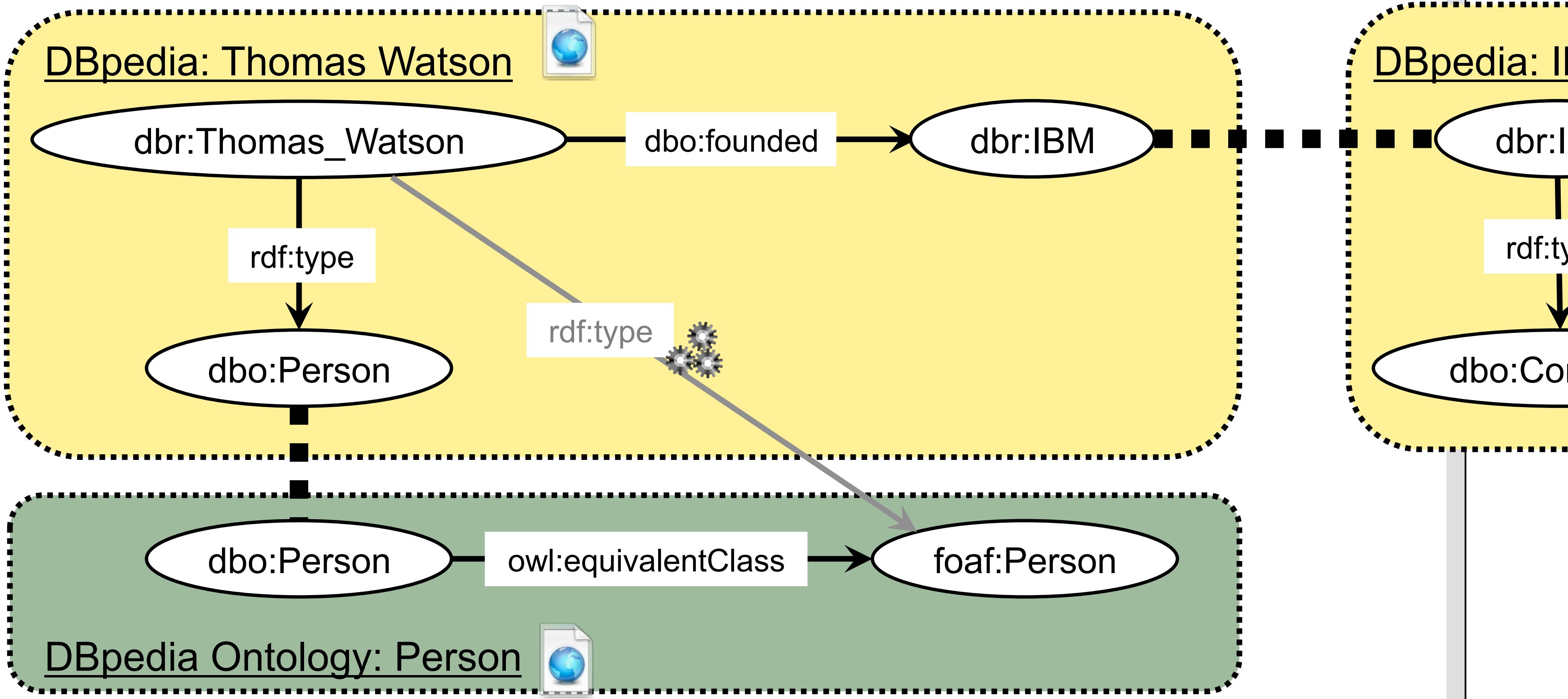


SIEMENS DERI

Core Intuition …

By using a URI to name a thing or property or class (e.g., dbo:Person), you are providing a link that implicitly validates the contents of the document that URI links to (the DBpedia page describing dbo:Person) and the recursive links from that document (e.g., the FOAF spec for foaf:Person).

Building Context through Links

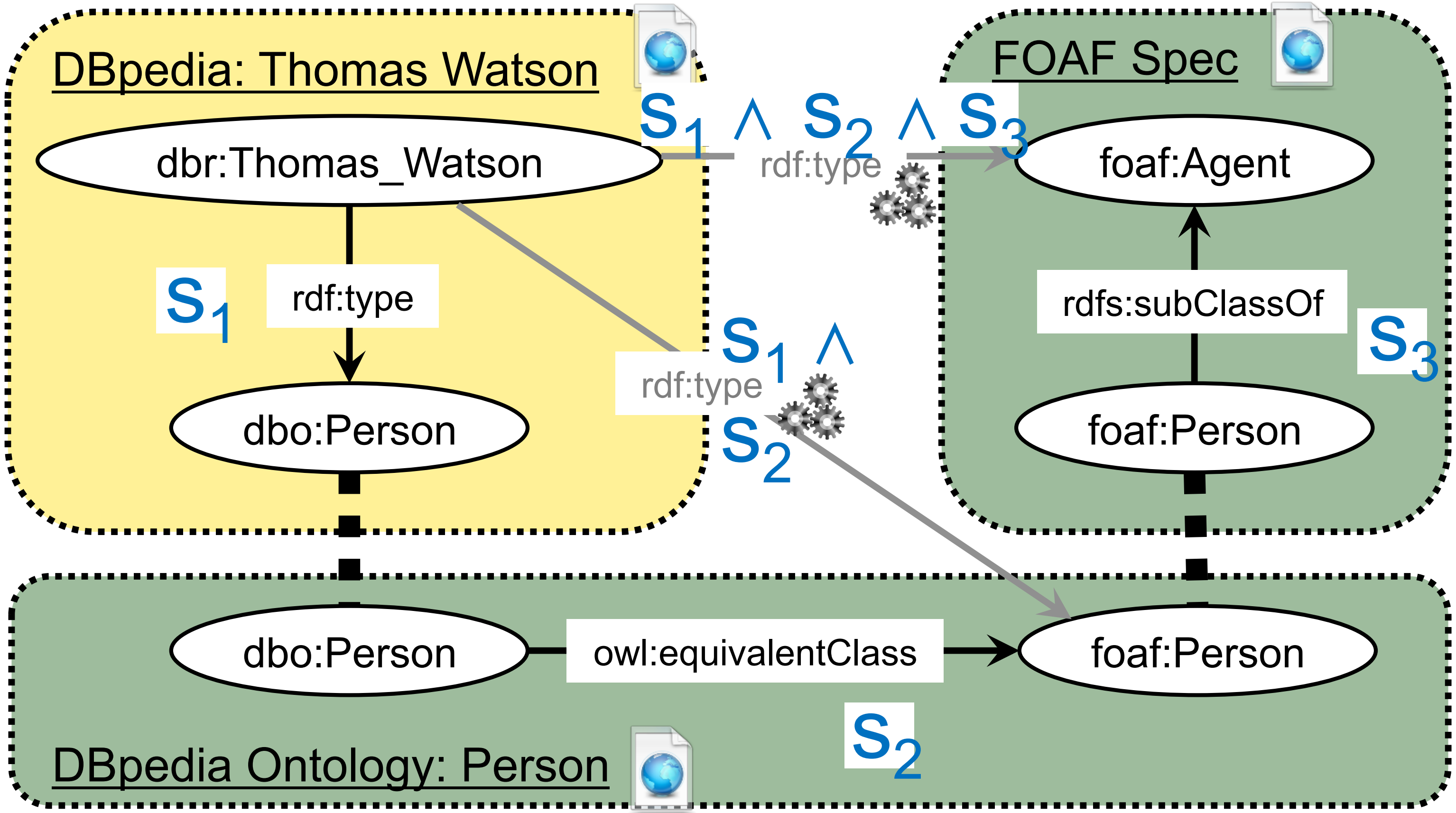


- From data: dereference properties and values of rdf:type
- From vocabularies: follow all OWL/RDFS links
- From all: follow owl:imports links
  - (Erm: not in example. Imagining it is left as exercise for the audience.)

Customisable!
(e.g., add owl:sameAs)

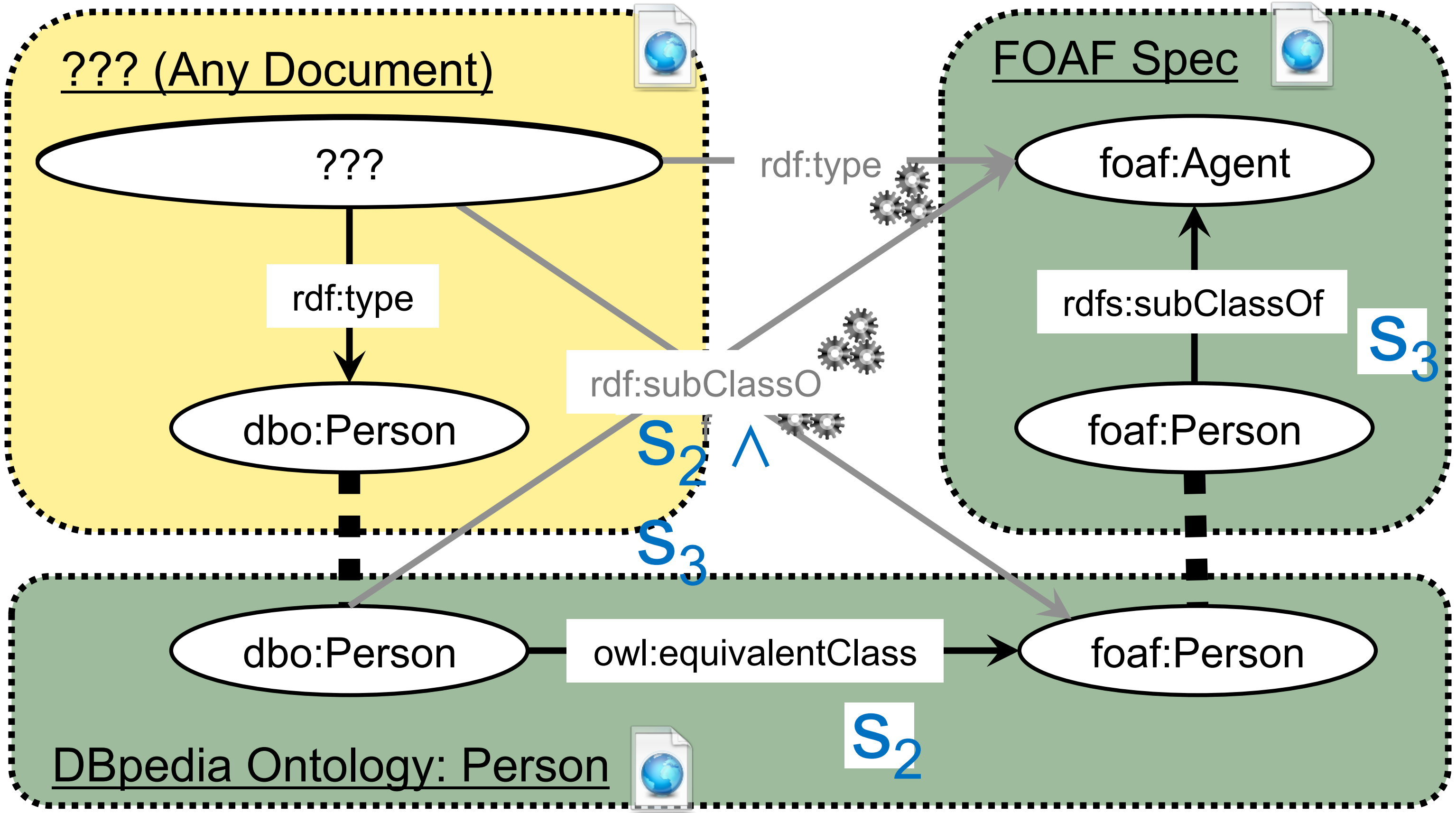
SIEMENS DERI

Context for Provenance



- Contexts created as conjunction of sources
- Inferences assigned provenence based on context

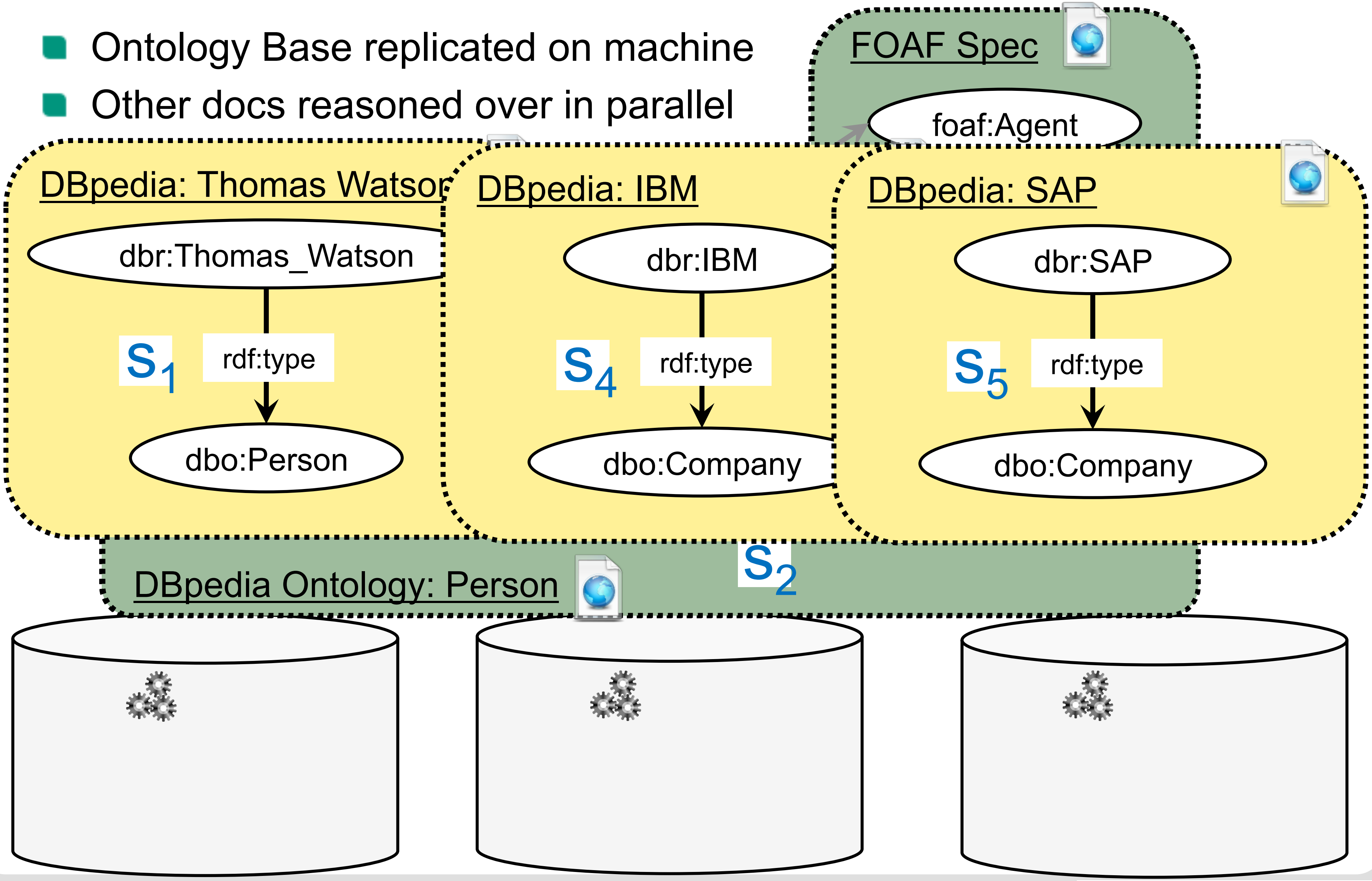
Context Re-use



- Context links for vocabularies not dependant on “data” documents
- Re-use/cache common contextual links in an “Ontology Base”
  - e.g., the link from DBpedia Ontology Person → FOAF Specification
- Pre-materialise inferences from S_2, S_3 and $S_2 \wedge S_3$

Parallelisation

- Ontology Base replicated on machine
- Other docs reasoned over in parallel

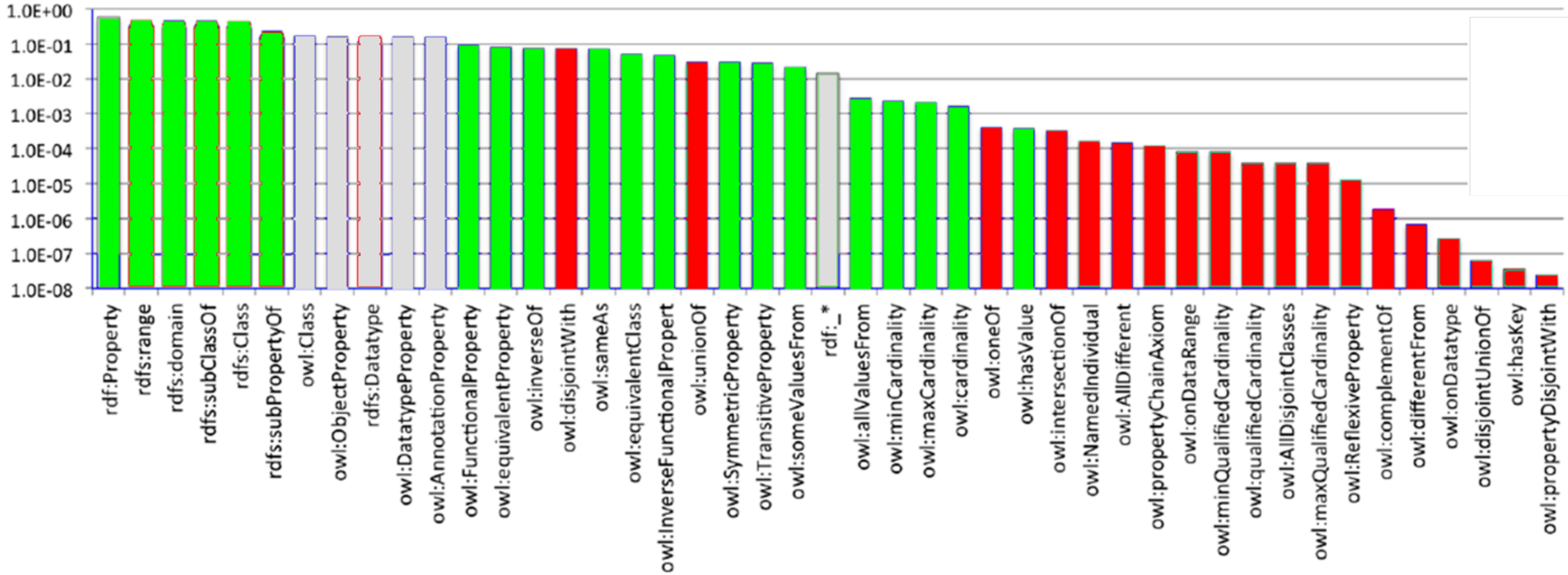


Sindice Reasoning Profile

- pD* (aka. OWL Horst)
- Rule-based (Datalog-style) inferencing
- Support for (incomplete) OWL 1 Full

pD* Supports (partially) | pD* Does Not Support (at all) | No inference required

***x*-axis is log-scale!**



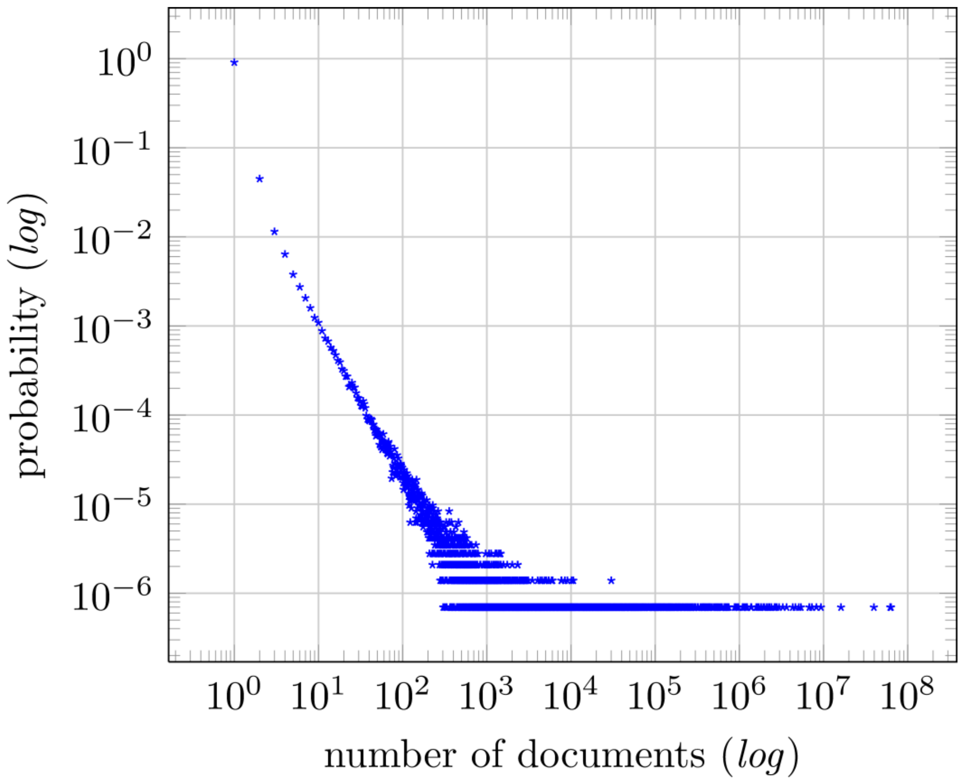
SIEMENS DERI

Context-Dependent Ontology Base

		T-Box Size		A-Box Size	
Dataset	Ontology	Explicit	Implicit	Explicit	Implicit
Geonames	15	1820	4005	6	14
DBPedia	8	657	1556	7	16
Sindice	14	2085	4601	69	170

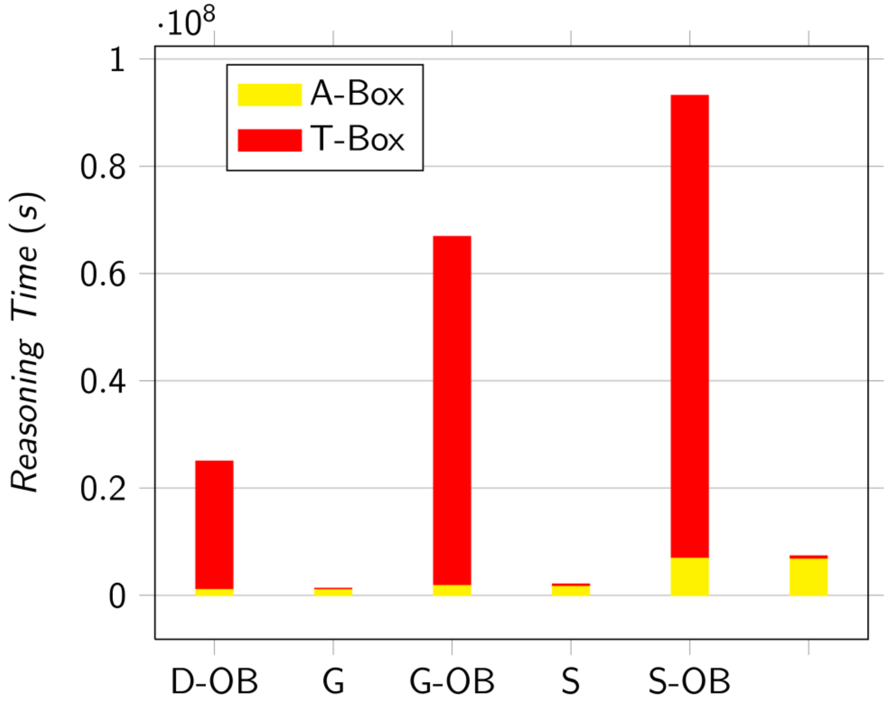
Table: Statistics over 100.000 random entity descriptions

Caching Ontology Closure



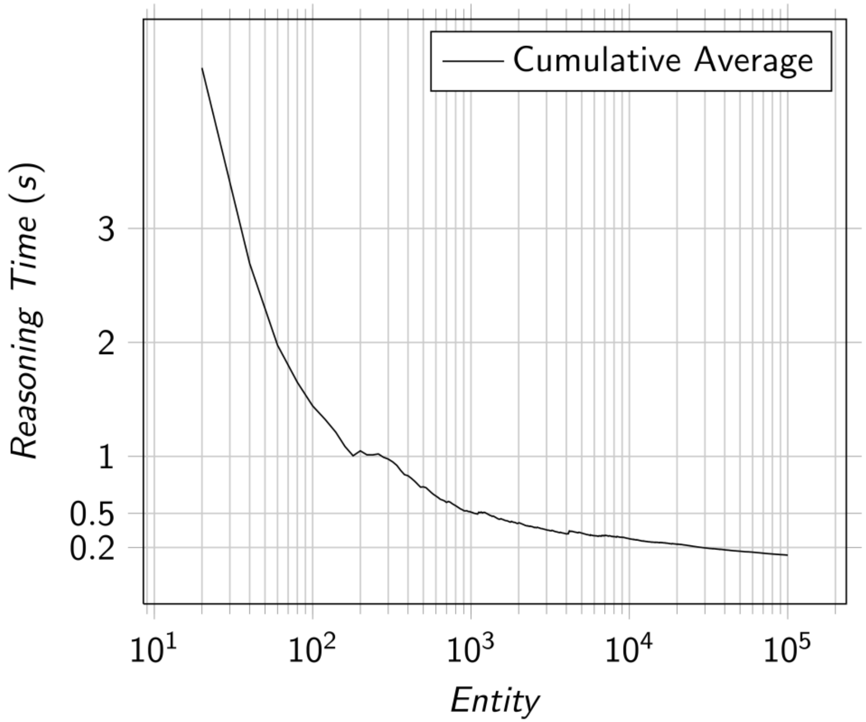
- Reuse of ontology closures across entities follows a power-law(-like) distribution
- Cache most recently used ontology closures

Optimisation with Ontology Base



- Computation of the T-Box closure: most time consuming operation
- It becomes negligible when the ontology base is activated

Optimisation with Ontology Base



- At the beginning, the reasoning time is very high
- After 10,000 entities, reach steady state of performance

Distributed Performance

DBPedia	Geonames	Sindice
83.3	50	13.9

Table: Average number of entities per second processed by one computing node

- One node is in charge of computing the deductive closure of one context
- Distributed model scales linearly with the number of nodes
- Context is generally small enough to be kept in memory

Linked Data Reasoning In Action: Sindice

- Context reasoning mechanism
  - Avoid deduction of undesirable assertions
  - Efficient distributed computing model
- Methodology deployed in production since 2008
  - More than 370 million documents
  - More than 40 billion triples (20 billion triples inferred)

More Details …

Context-Dependent OWL Reasoning in Sindice - Experiences and Lessons Learnt *

Renaud Delbru[1], Giovanni Tummarello[1], and Axel Polleres[1,2]

[1] Digital Enterprise Research Institute
National University of Ireland, Galway
{renaud.delbru,giovanni.tummarello,axel.polleres}@deri.org
[2] Siemens AG Österreich
Siemensstrasse 90, 1210, Vienna, Austria

Abstract. The Sindice Semantic Web index provides search capabilities over 260 million documents. Reasoning over web data enables to make explicit what would otherwise be implicit knowledge: it adds value to the information and enables Sindice to ultimately be more competitive in terms of precision and recall. However, due to the scale and heterogeneity of web data, a reasoning engine for the Sindice system must (1) scale out through parallelisation over a cluster of machines; and (2) cope with unexpected data usage. In this paper, we report our experiences and lessons learned in building a large scale reasoning engine for Sindice. The reasoning approach has been deployed, used and improved since 2008 within Sindice and has enabled Sindice to reason over billions of triples.

1 Introduction

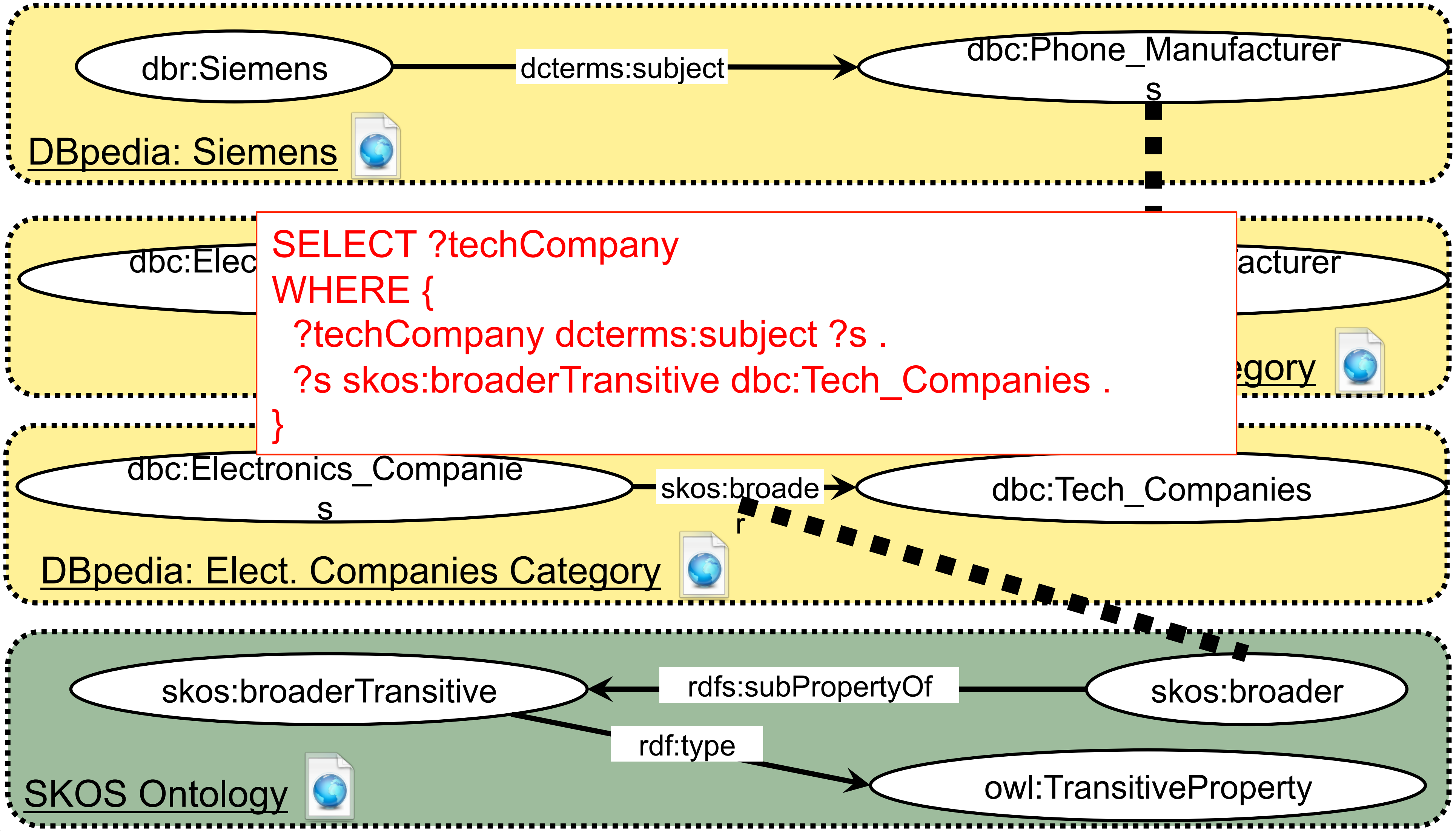
Reasoning over semantic entity description enables to make explicit what would otherwise be implicit knowledge: it adds value to the information and enables a web data search engine such as Sindice to ultimately be more competitive in terms of precision and recall [16]. The drawback is that inference can be computationally expensive, and therefore drastically slow down the process of indexing large amounts of information. Therefore, large scale reasoning through parallelisation is one requirement of Sindice.

A common strategy for reasoning with web data is to put several entity descriptions together and to compute the deductive closure across all the entity descriptions. How-

Renaud Delbru, Giovanni Tummerello and Axel Polleres. "CONTEXT-DEPENDENT OWL REASONING IN SINDICE – EXPERIENCES AND LESSONS LEARNT ". In the Proceedings of the 5th International Conference on Web Reasoning and Rule Systems (RR). 2011.

SIEMENS DERI

Misses inferences across documents …




```
SELECT ?techCompany
WHERE {
  ?techCompany dcterms:subject ?s .
  ?s skos:broaderTransitive dbc:Tech_Companies .
}
```


SIEMENS DERI

Authoritative Reasoning

MATERIALISATION (II)

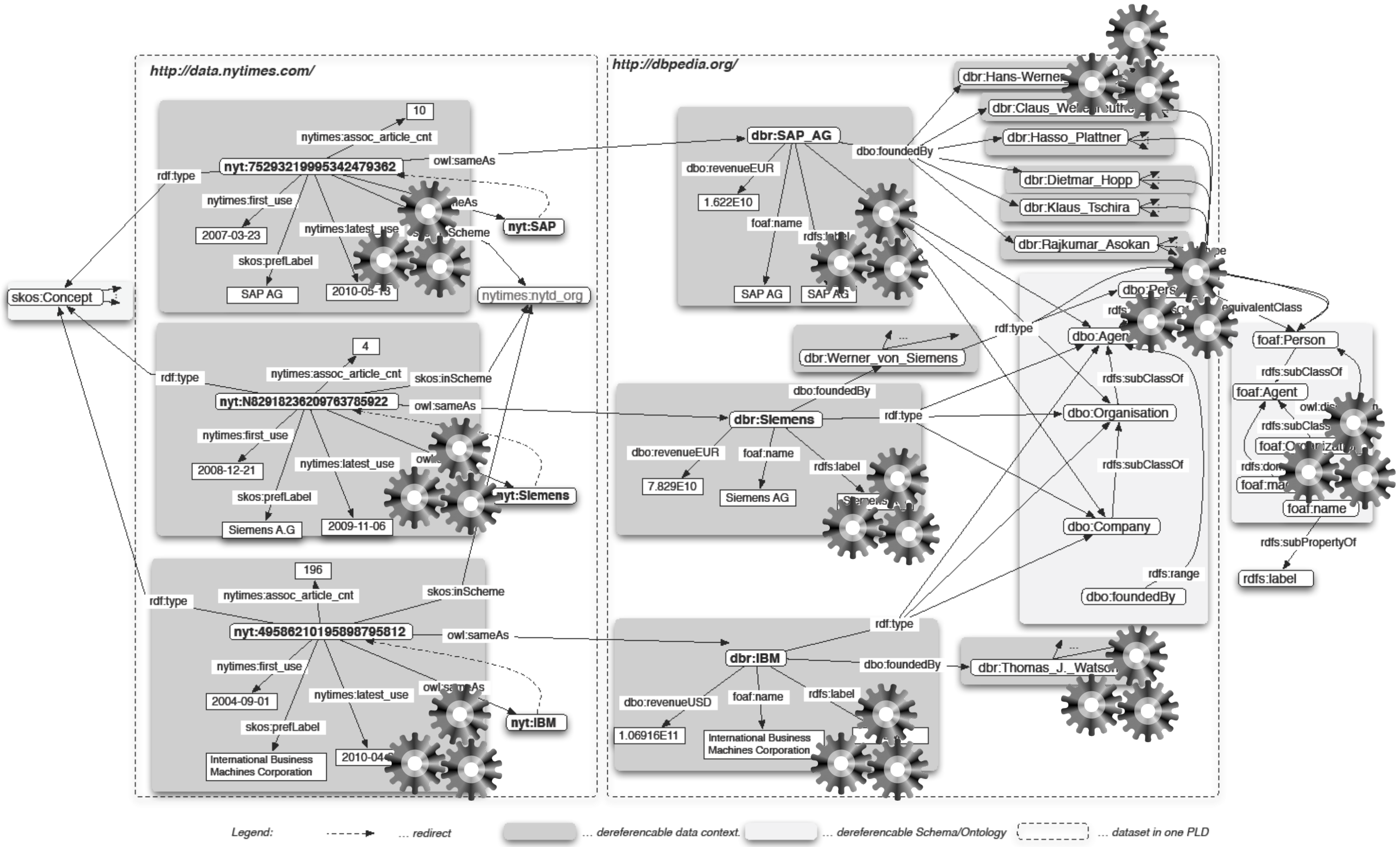
Reasoning over Web Data

- Challenges
  - C1: Linked Data is **huge**
  - C2: Linked Data is not “pure” OWL (DL)
  - C3: Linked Data is not consistent
    - Linked Data is Web data
  - C4: Linked Data is Evolving
  - C5: Linked Data needs more than RDFS+OWL

SIEMENS DERI

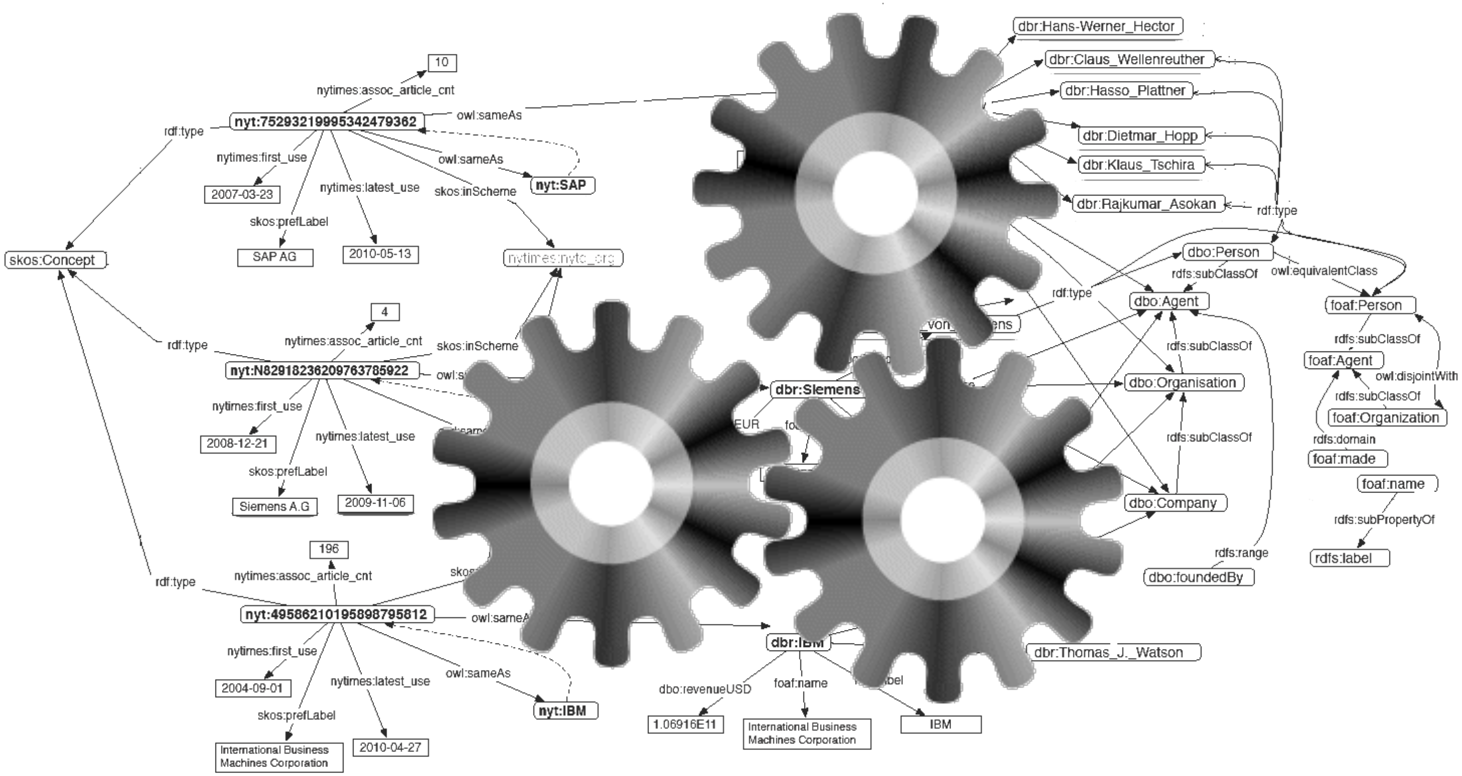
Authoritative Reasoning: Global Axioms

- Instead of breaking up reasoning into small contexts …



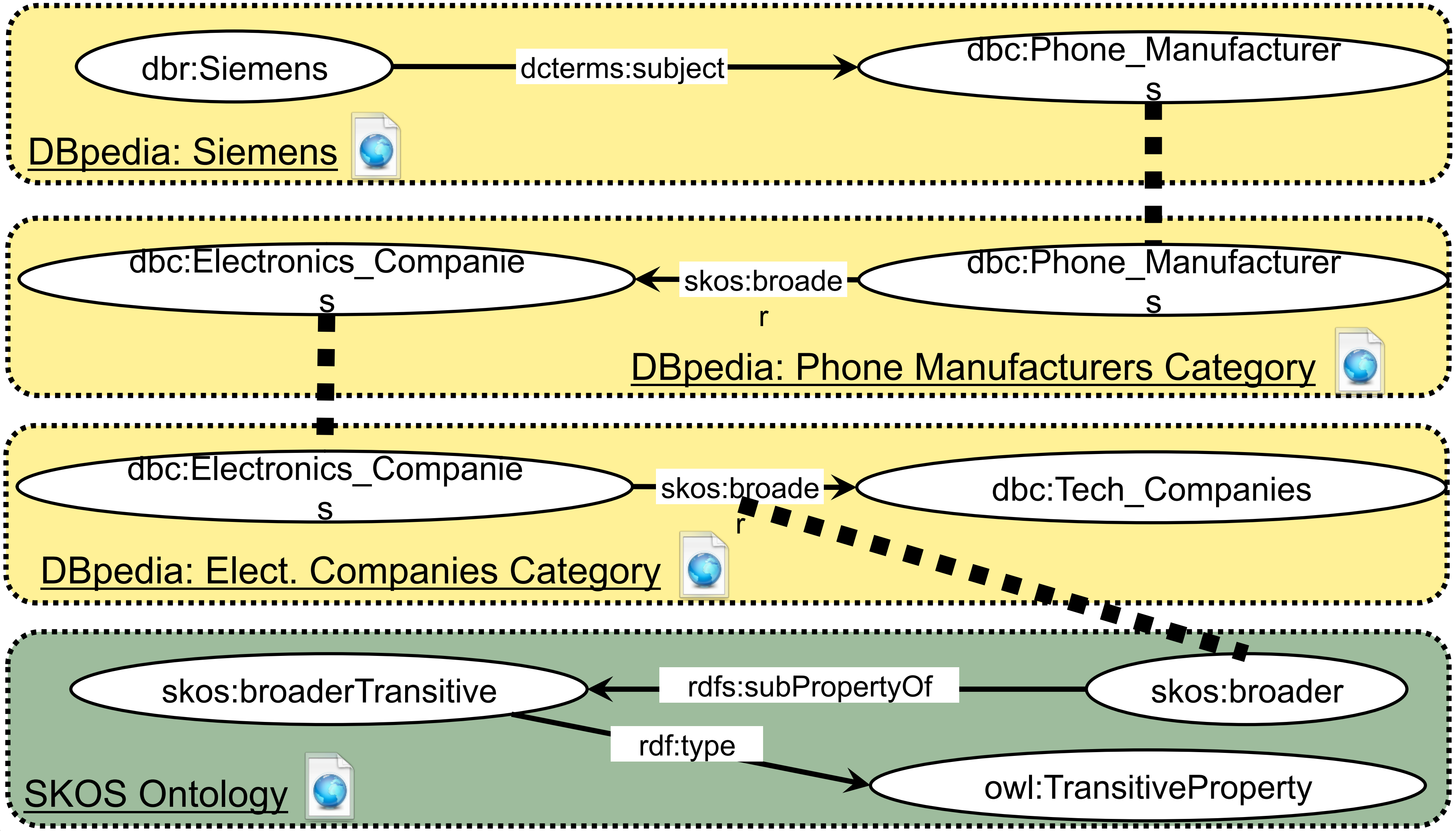
Authoritative Reasoning: Global Axioms

- Return to reasoning over one big graph …
  - But only for "trusted" data!



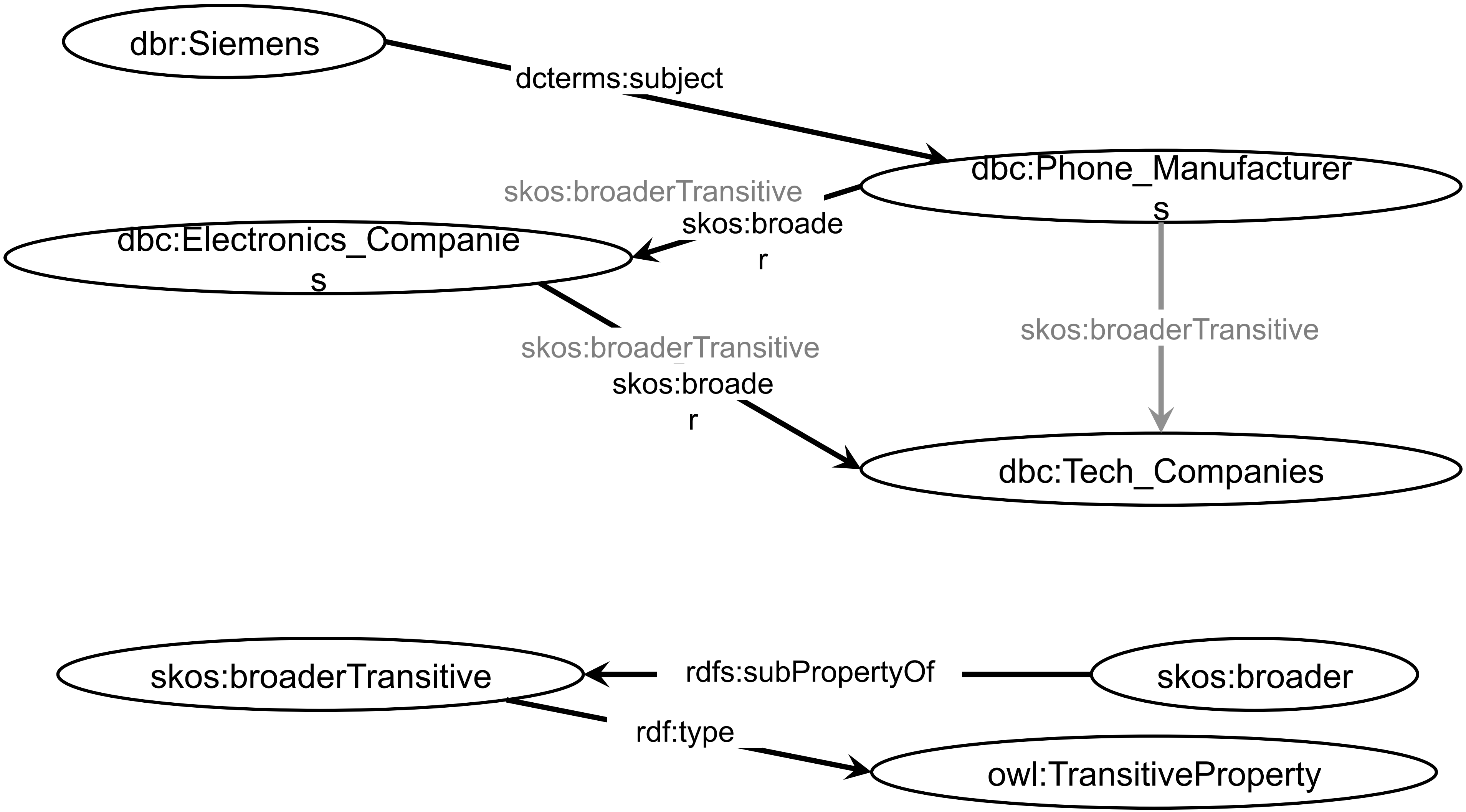
SIEMENS DERI

Inferences across documents …

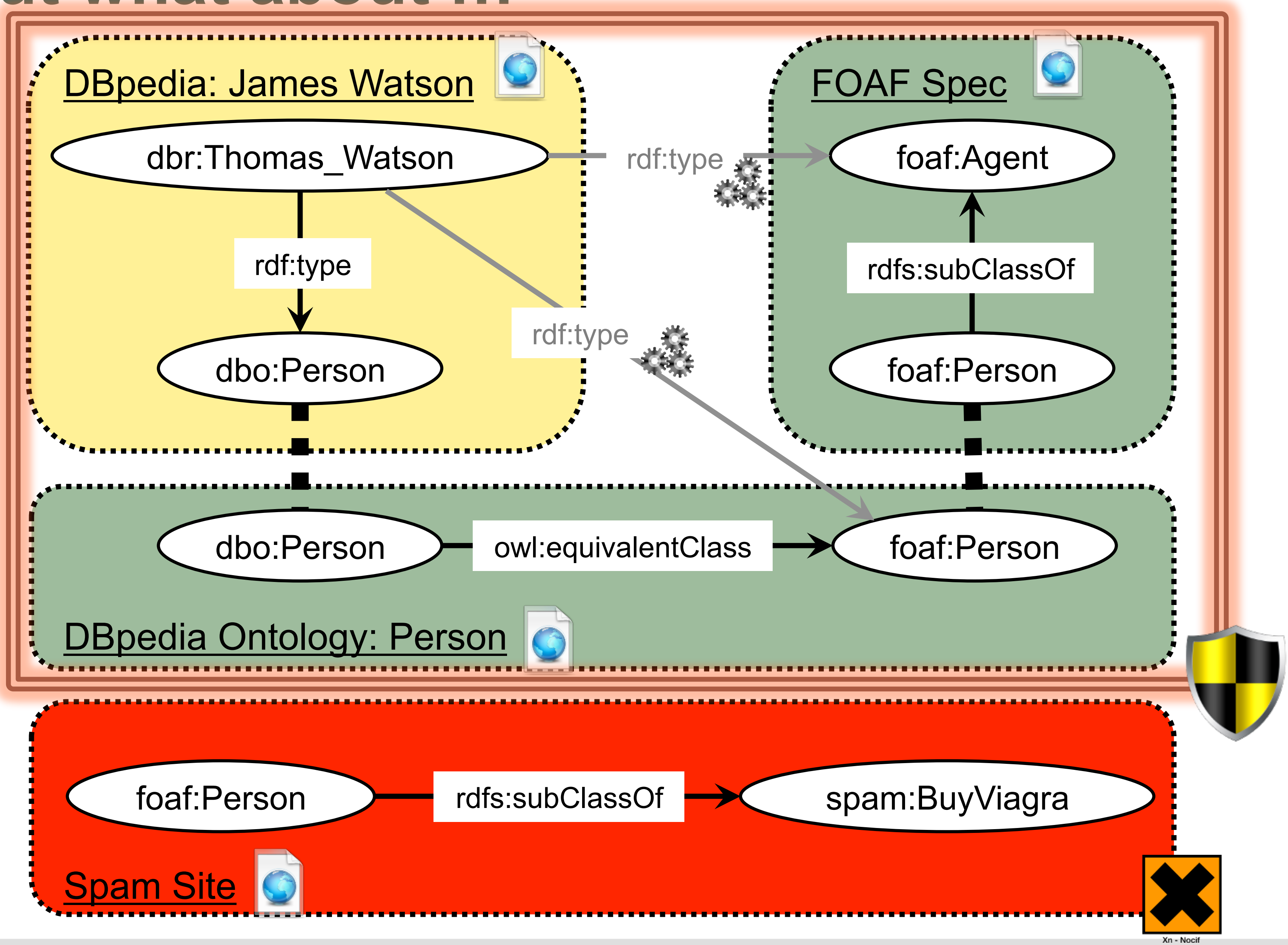


SIEMENS DERI

Inferences across documents …

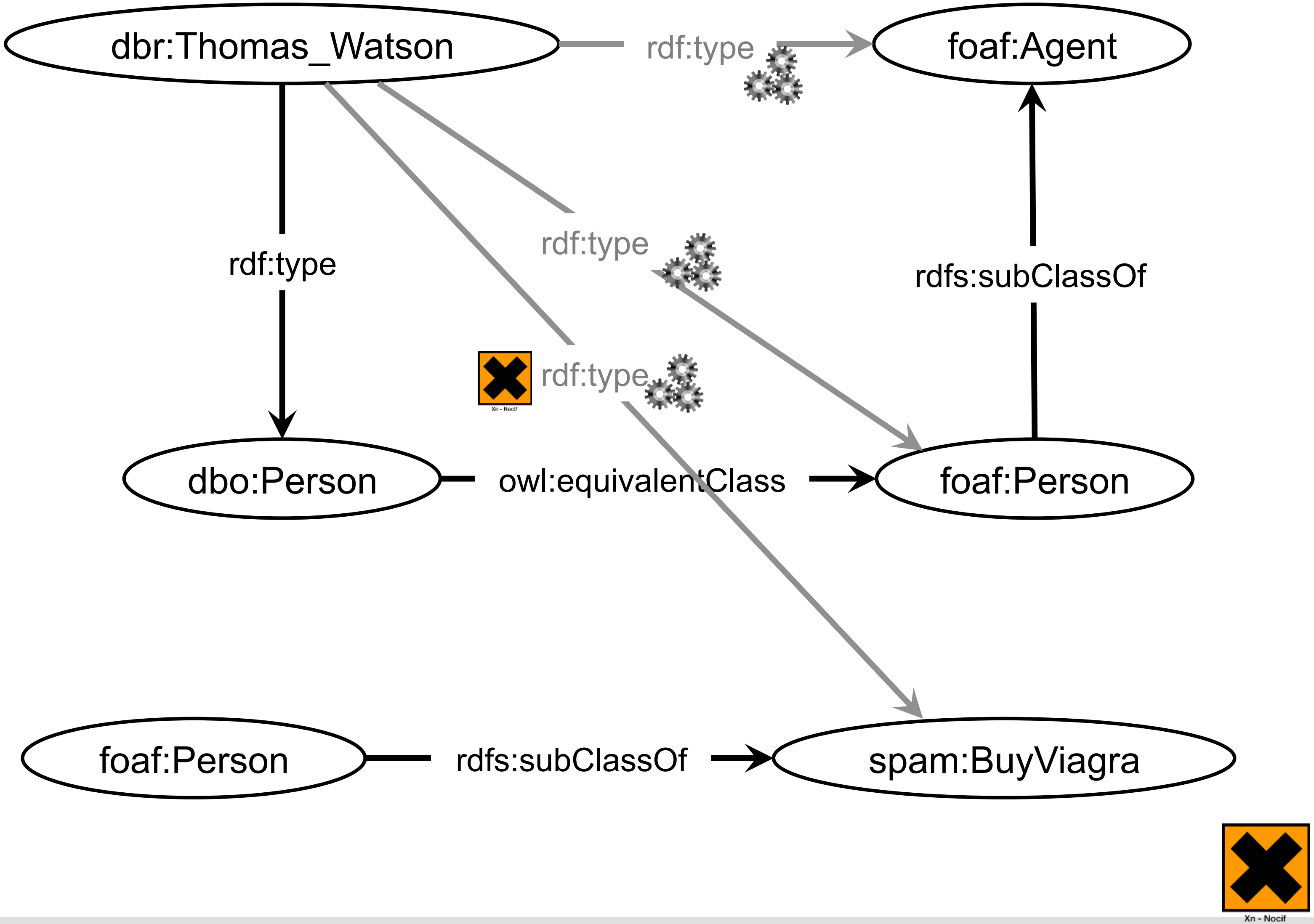


But what about …



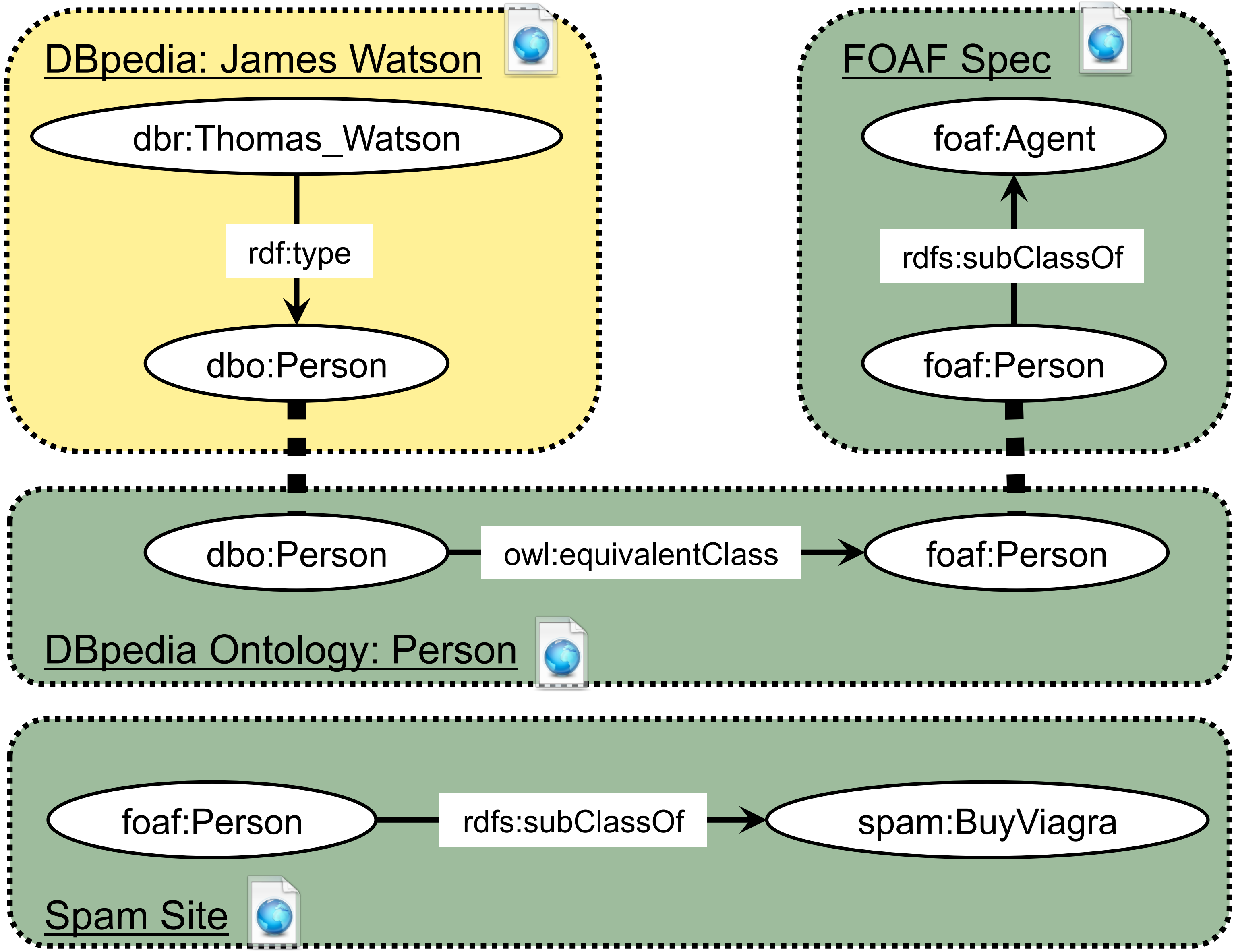
SIEMENS DERI

But what about …

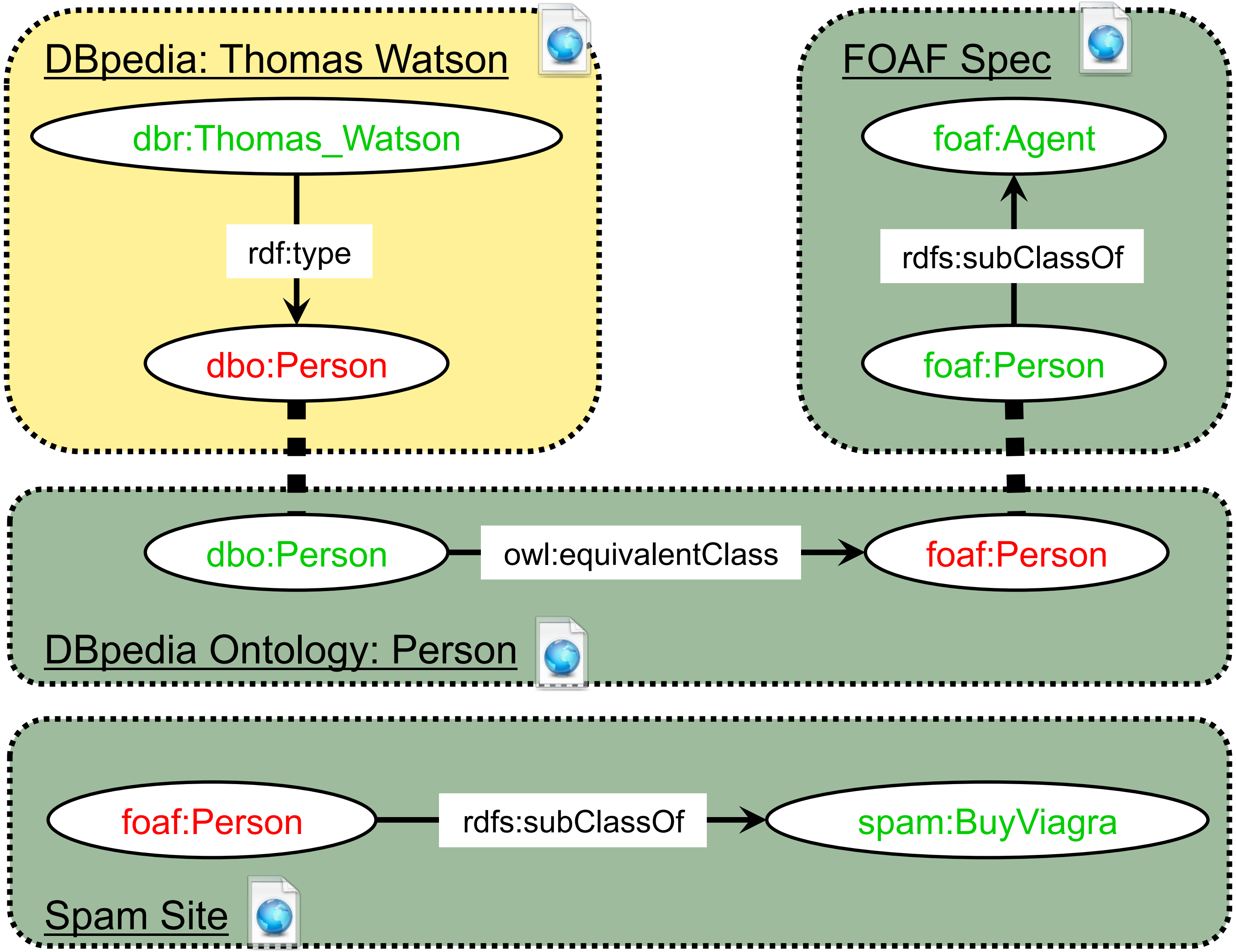


SIEMENS DERI

But what about …

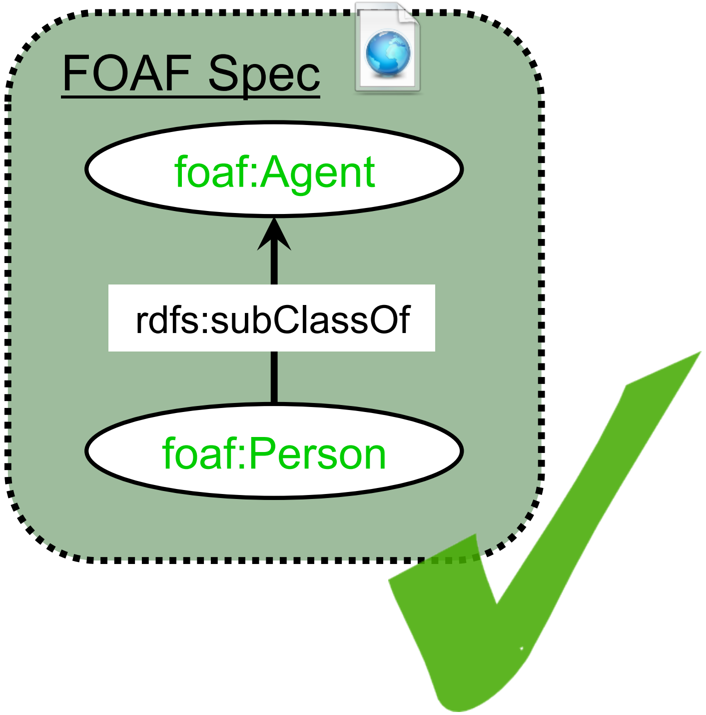


But what about …

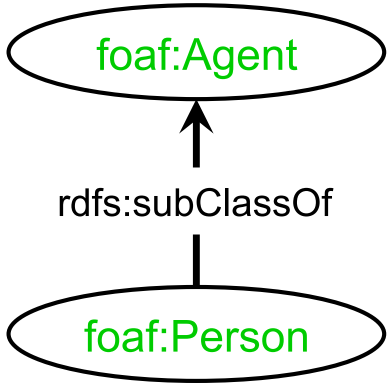


Which axioms to trust …

- The FOAF Spec states:
  "All instances of foaf:Person are instances of foaf:Agent"

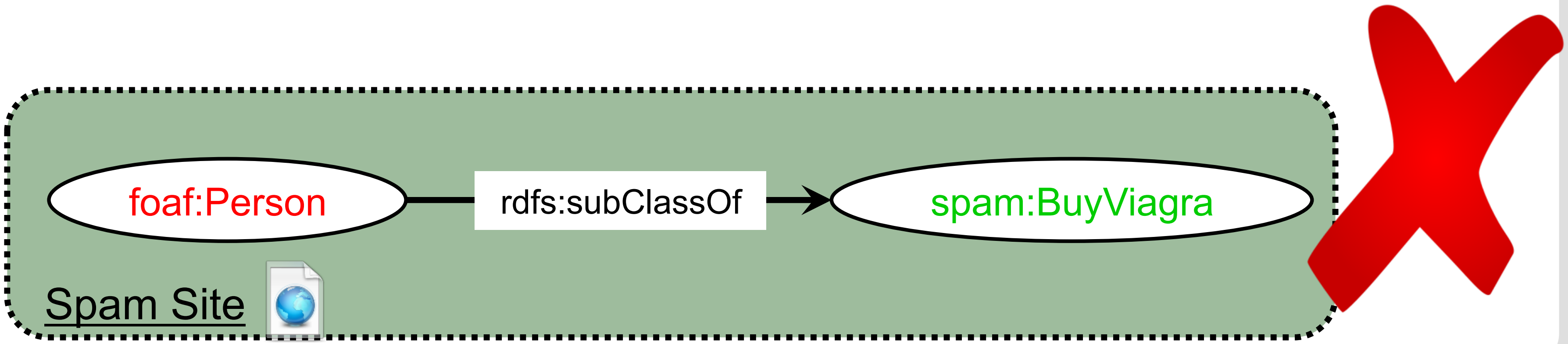


Which axioms to trust …



- The Spam Site states:

  "All instances of foaf:Person are instances of foaf:Agent"



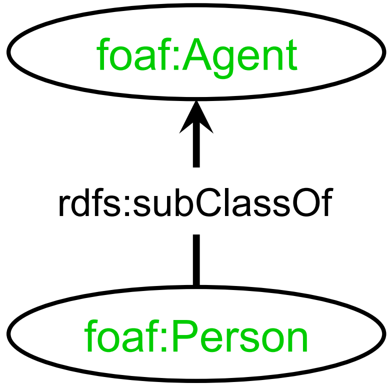
Which axioms to trust …

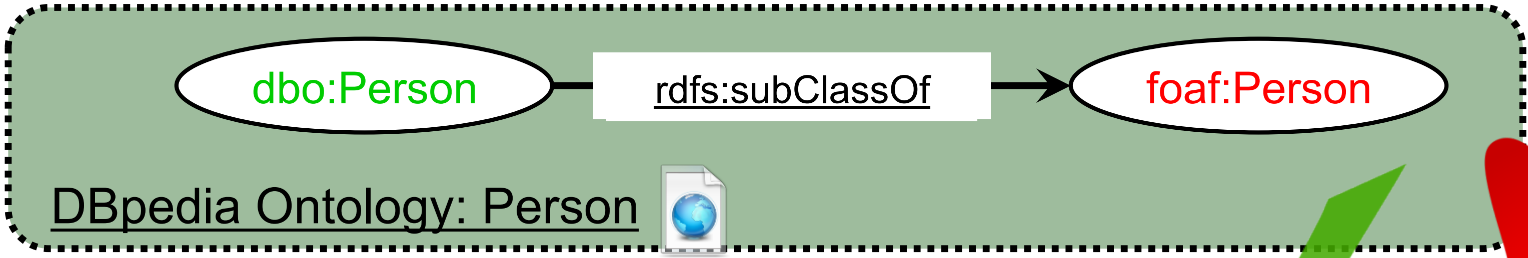
- The DBpedia page says

  "All instances of dbo:Person are instances of foaf:Person"

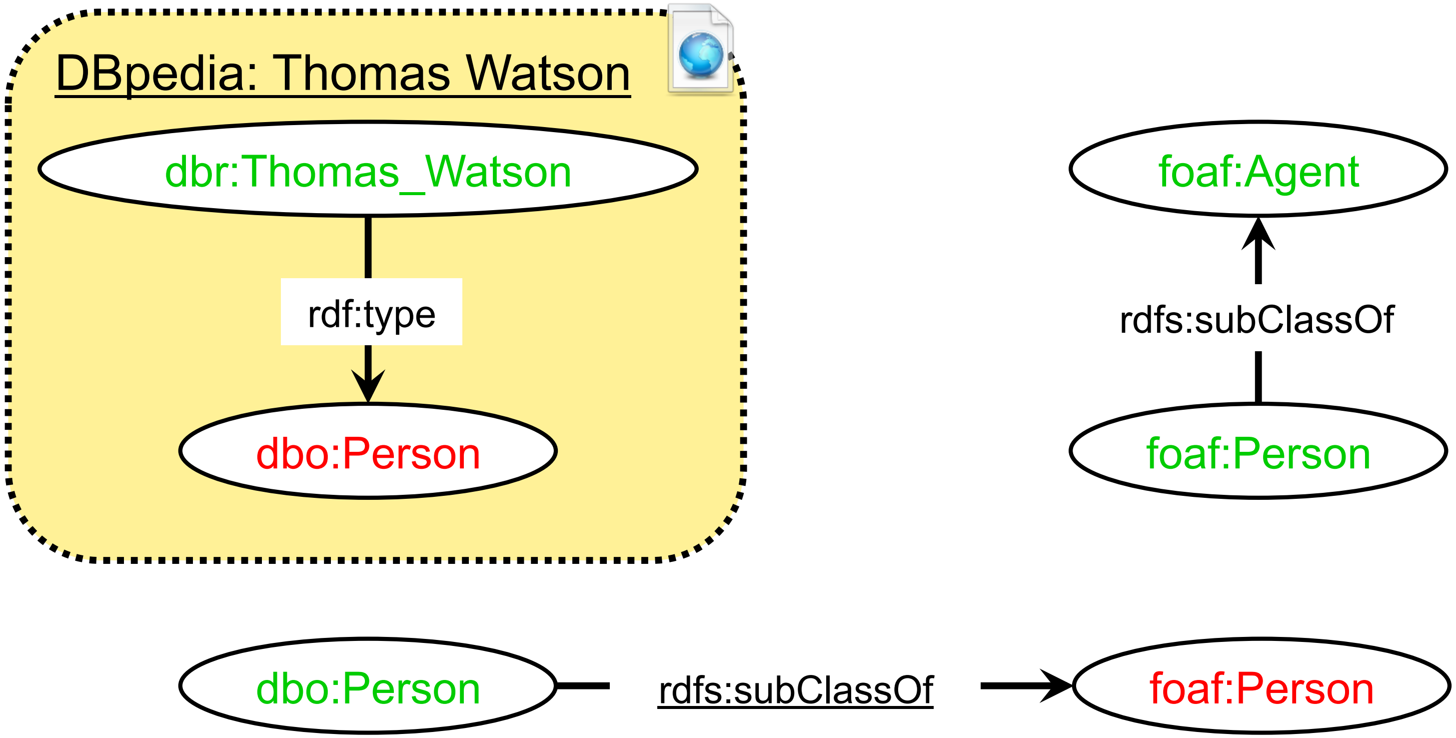
  and

  "All instances of foaf:Person are instances of dbo:Person"





Which axioms to trust …

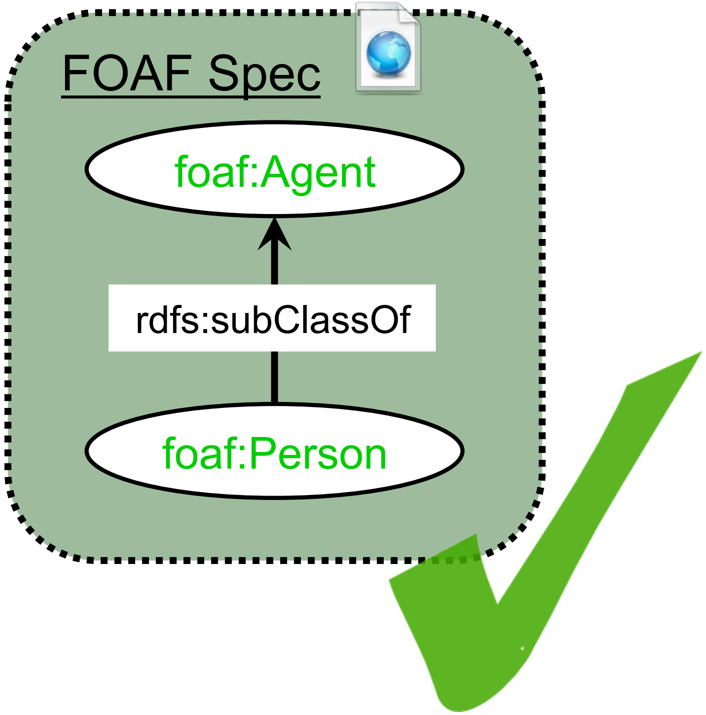


- The DBpedia page says
  "dbr:Thomas_Watson is an instance of dbo:Person"

Authoritative reasoning …

- The FOAF Spec states:

  "All instances of foaf:Person are instances of foaf:Agent"

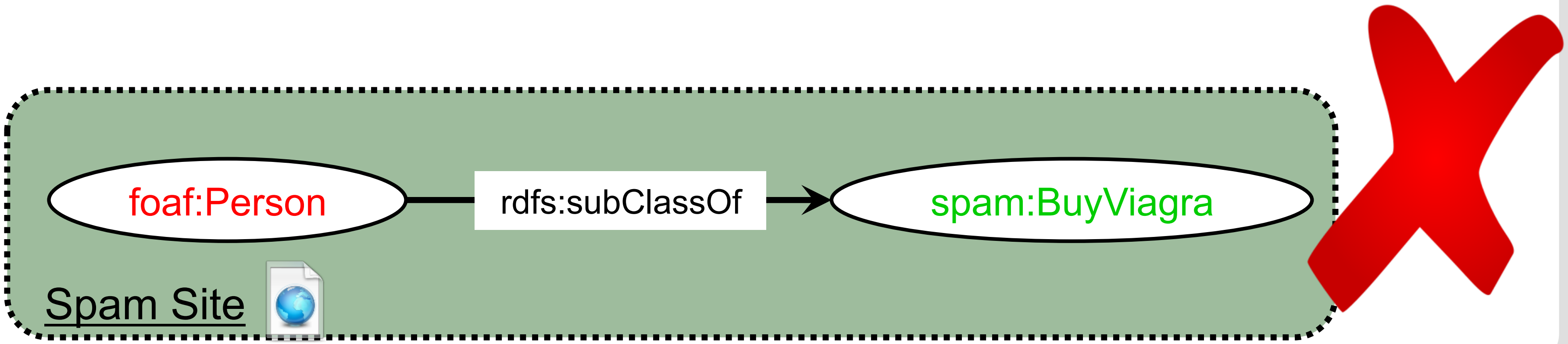


foaf:Person(?x) → foaf:Agent(?x)

Authoritative reasoning …

- The Spam Site states:
  "All instances of foaf:Person are instances of foaf:Agent"

foaf:Person(?x) → spam:BuyViagra(?x)



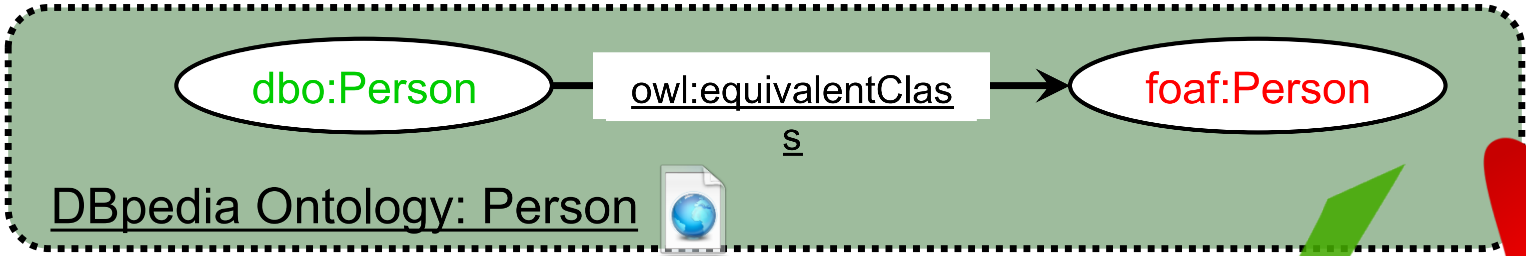
SIEMENS DERI

Authoritative reasoning …

- The DBpedia page says
  “All instances of dbo:Person are instances of foaf:Person”

  and

  “All instances of foaf:Person are instances of dbo:Person”

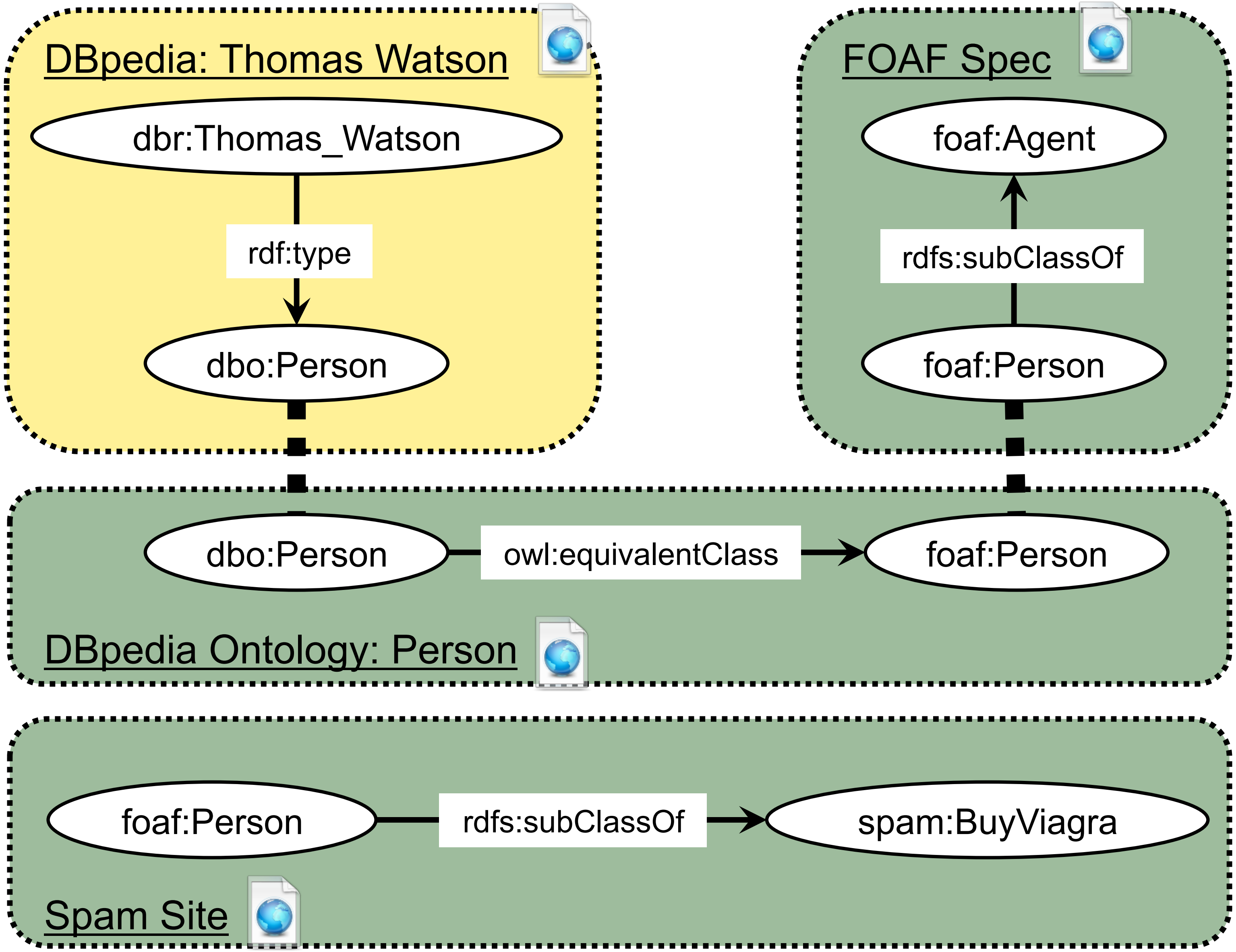


dbo:Person(?x) → foaf:Person(?x) ✓

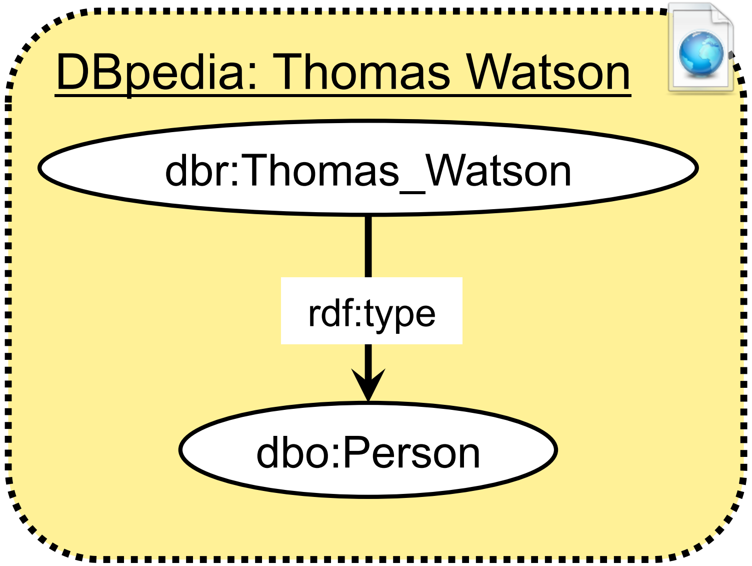
foaf:Person(?x) → dbo:Person(?x) ✗

SIEMENS DERI

Authoritative reasoning …



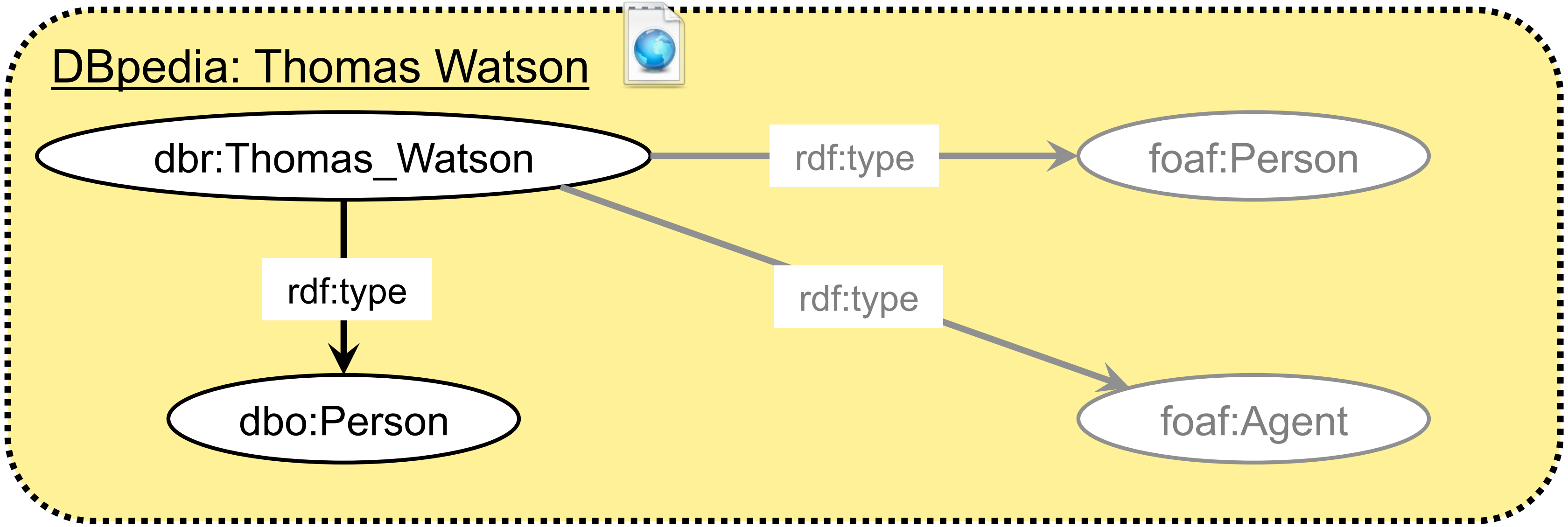
Apply Rules …



dbo:Person(?x) → foaf:Person(?x)

foaf:Person(?x) → foaf:Agent(?x)

Apply Rules to Data …



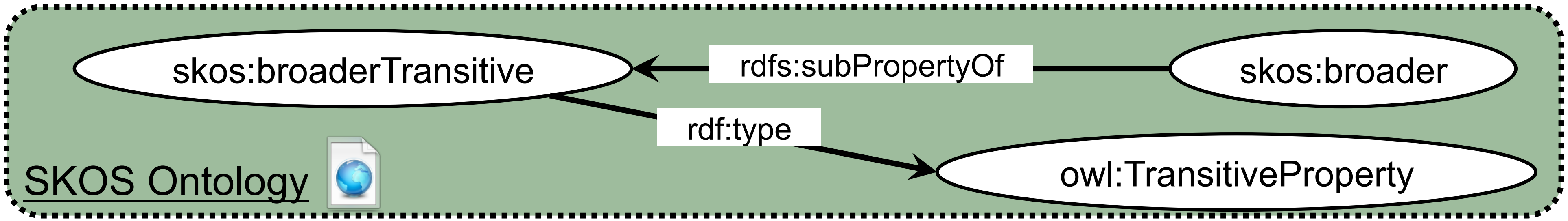
dbo:Person(?x) → foaf:Person(?x)

foaf:Person(?x) → foaf:Agent(?x)

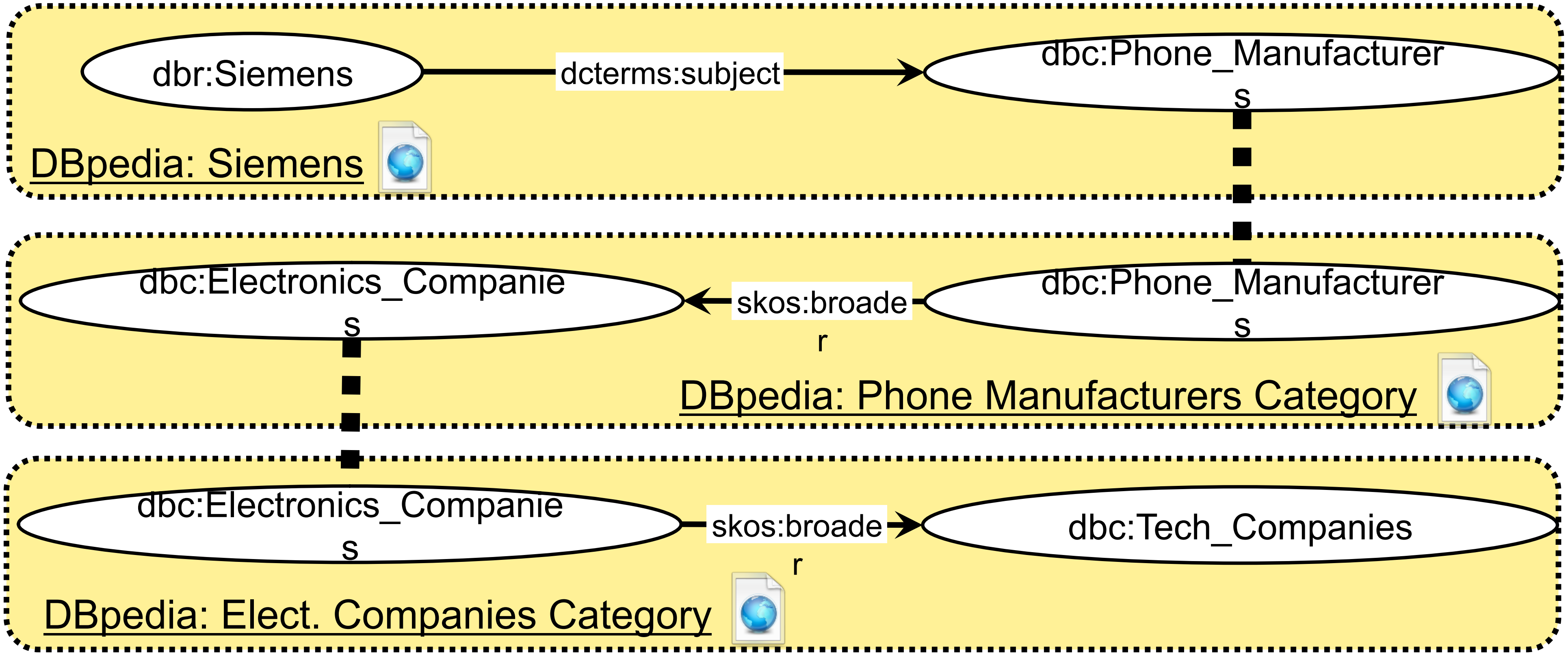
Inferences across documents …

skos:broader(?x,?y) → skos:broaderTransitive(?x,y)

skos:broaderTransitive(?x,?y), skos:broaderTransitive(?y,?z) → skos:broaderTransitive(?x,?z)



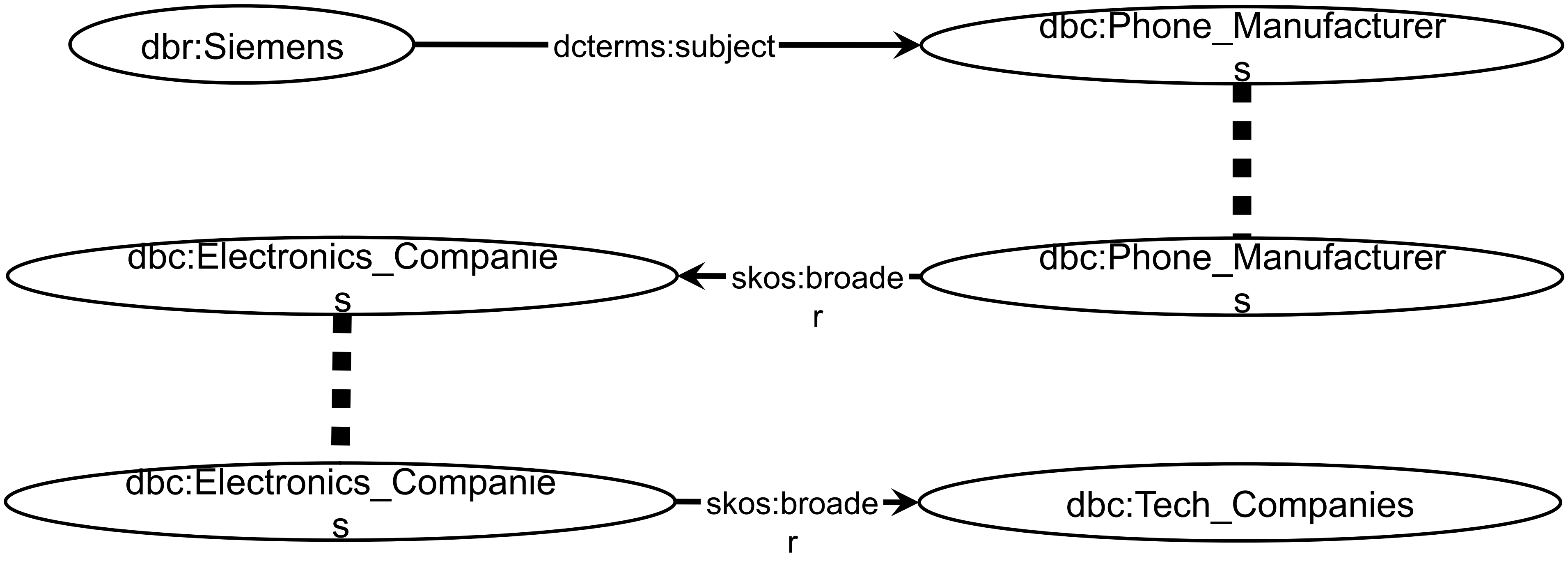
Inferences across documents …



skos:broader(?x,?y) → skos:broaderTransitive(?x,?y)

skos:broaderTransitive(?x,?y), skos:broaderTransitive(?y,?z) → skos:broaderTransitive(?x,?z)

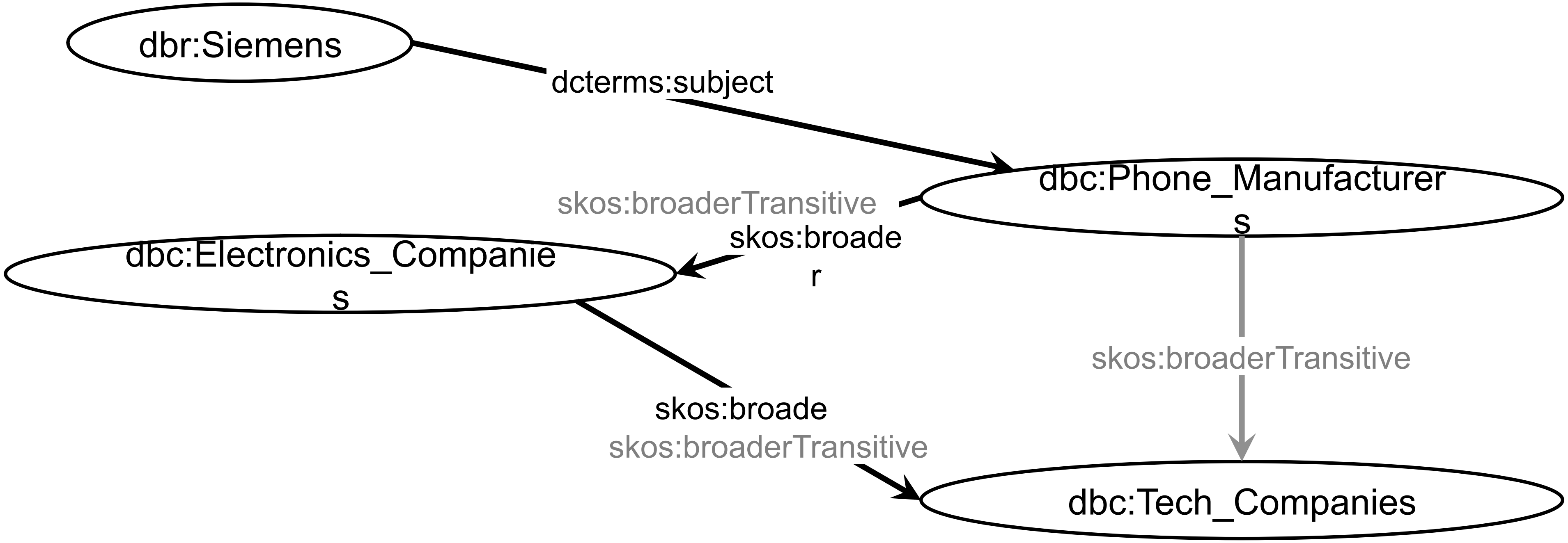
Inferences across documents …



skos:broader(?x,?y) → skos:broaderTransitive(?x,?y)

skos:broaderTransitive(?x,?y), skos:broaderTransitive(?y,?z) → skos:broaderTransitive(?x,?z)

Inferences across documents …



skos:broader(?x,?y) → skos:broaderTransitive(?x,?y)

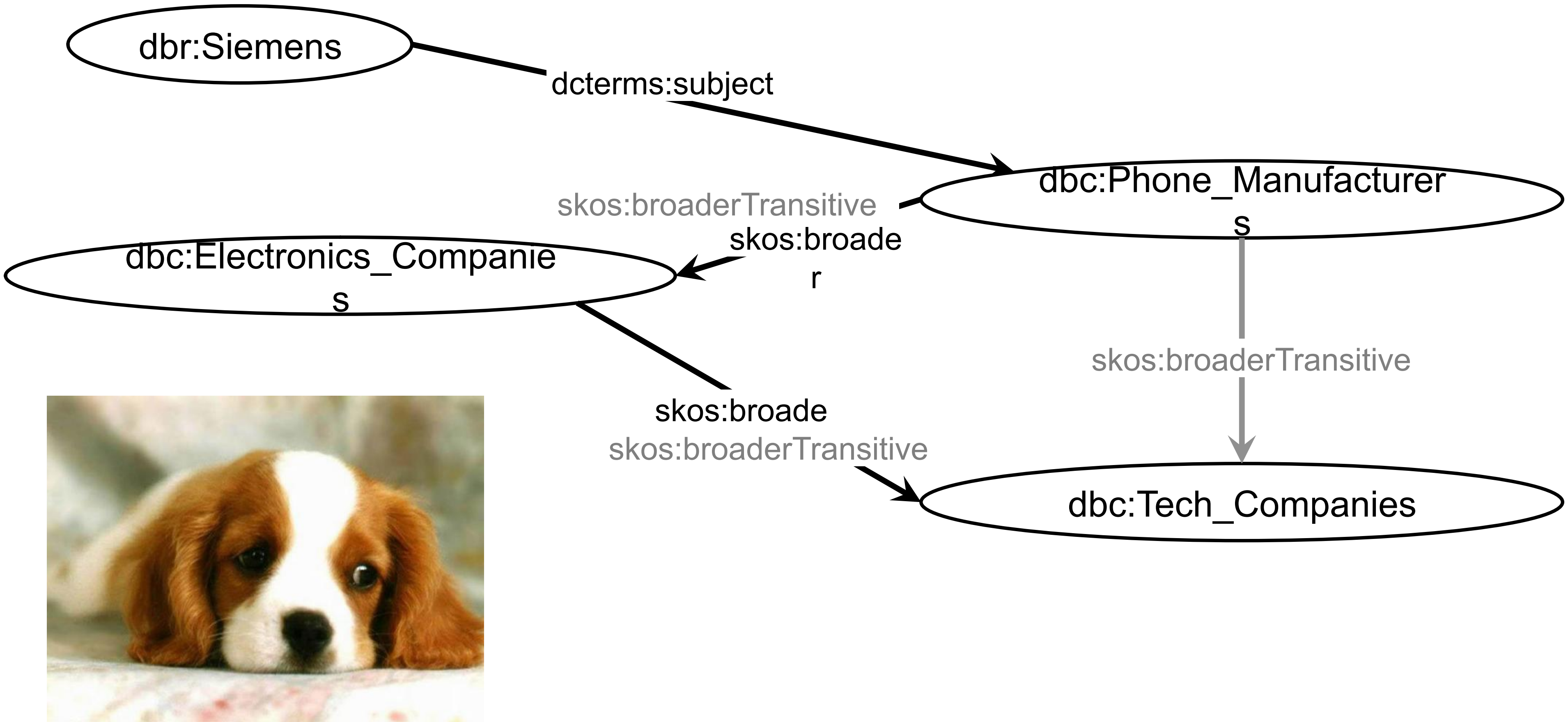
skos:broaderTransitive(?x,?y), skos:broaderTransitive(?y,?z) → skos:broaderTransitive(?x,?z)

SIEMENS DERI

Reasoning over Web Data

- Challenges
  - C1: Linked Data is **huge**
  - C2: Linked Data is not “pure” OWL (DL)
  - C3: Linked Data is not consistent
    - Linked Data is Web data
  - C4: Linked Data is Evolving
  - C5: Linked Data needs more than RDFS+OWL

How can we do transitive reasoning like this on a Web scale graph?



skos:broader(?x,?y) → skos:broaderTransitive(?x,?y)

skos:broaderTransitive(?x,?y), skos:broaderTransitive(?y,?z) → skos:broaderTransitive(?x,?z)

How can we do transitive reasoning like this on a Web scale graph? (You can't reliably)

- Transitivity is quadratic in output: $O(n^2)$
  - Though that's tractable, it's definitely not tractable
- Transitivity cannot be effectively parallelised
  - Throwing more machines at the problem won't help
- You can do some transitivity, maybe all transitivity, but in practice, you cannot guarantee these features over a Web-scale database

… and the problem is not just transitivity

Take the Easy Way Out …

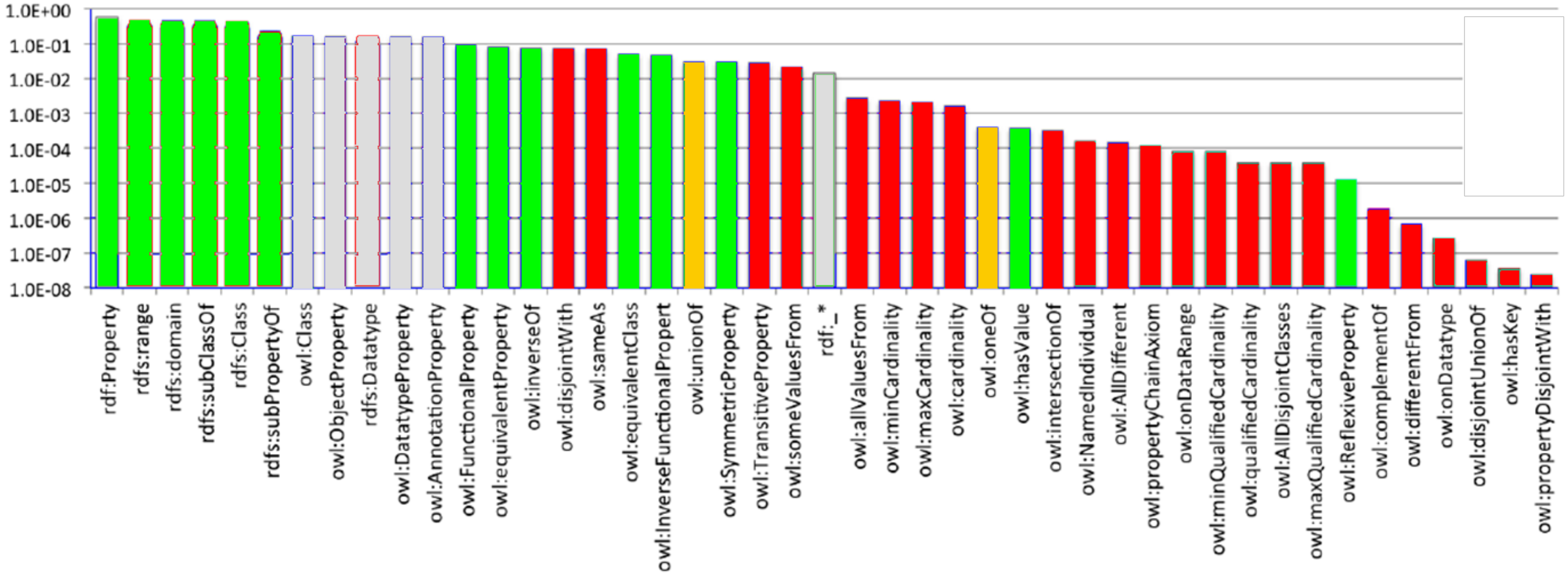
- Only use rules with one condition (1 body atom):
  - skos:broader(?x,?y) → skos:broaderTransitive(?x)
  - dbo:Person(?x) → foaf:Person(?x)
  - foaf:Person(?x) → foaf:Agent(?x)
  - "Linear rules"
- Leave out the rules with more than one condition (>1 body atom)
  - ~~skos:broaderTransitive(?x,?y), skos:broaderTransitive(?y,?z) → skos:broaderTransitive(?x,?z)~~
- Remaining rules relatively easy to run … parallelisable … scalable …
- But is this a gross simplification?

A Gross Simplification?

- RDFS/OWL features that can be "covered" by linear rules
- RDFS covered in its entirety

Covered | **Partial** | **Not Covered** | **No inference required**

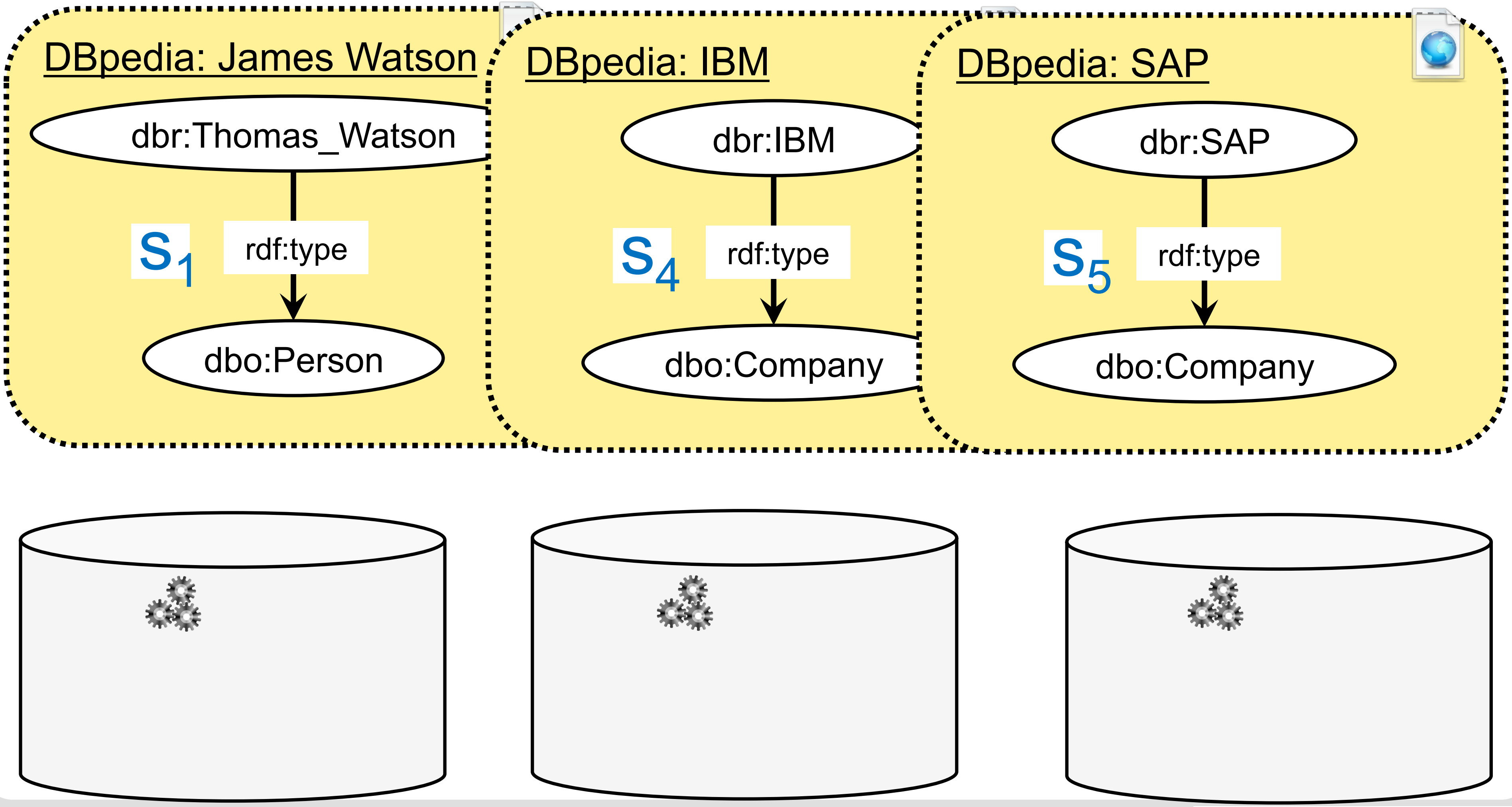
***x*-axis is log-scale!**



SIEMENS DERI

Parallelisation

- Rule Base replicated
- Data reasoned over in parallel (no joins; embarrassingly parallel)



Application for SWSE

- Apply reasoning over 1.1 billion RDF triples from 4.4 million Web documents (an open-domain unguided Web crawl)
  - Guaranteed to be a lot of crap in there!
  - May 2010: Dataset still up on http://swse.deri.org/ (I hope :P)

Example Noise …

Five authoritative rules for (164 million) members of foaf:Person

- foaf:Person(?x) → foaf:Agent(?x)
- foaf:Person(?x) → wgs84:SpatialThing(?x)
- foaf:Person(?x) → contact:Person(?x)
- foaf:Person(?x) → dcterms:Agent(?x)
- foaf:Person(?x) → contact:SocialEntity(?x)

Twenty-five non-authoritative rules dropped:

- foaf:Person(?x) → po:Person(?x)
- foaf:Person(?x) → aifb:Kategorie-3AAIFB(?x)
- foaf:Person(?x) → b2r2008:Controlled-vocabularies(?x)
- …

Final Rule Base contains
301 thousand rules

Materialisation …

- Use the 301 thousand rules to apply materialisation
  - Over 1.1 billion triples

- But how to handle so many rules?
  - They're simple (one condition) but there's a lot of them …

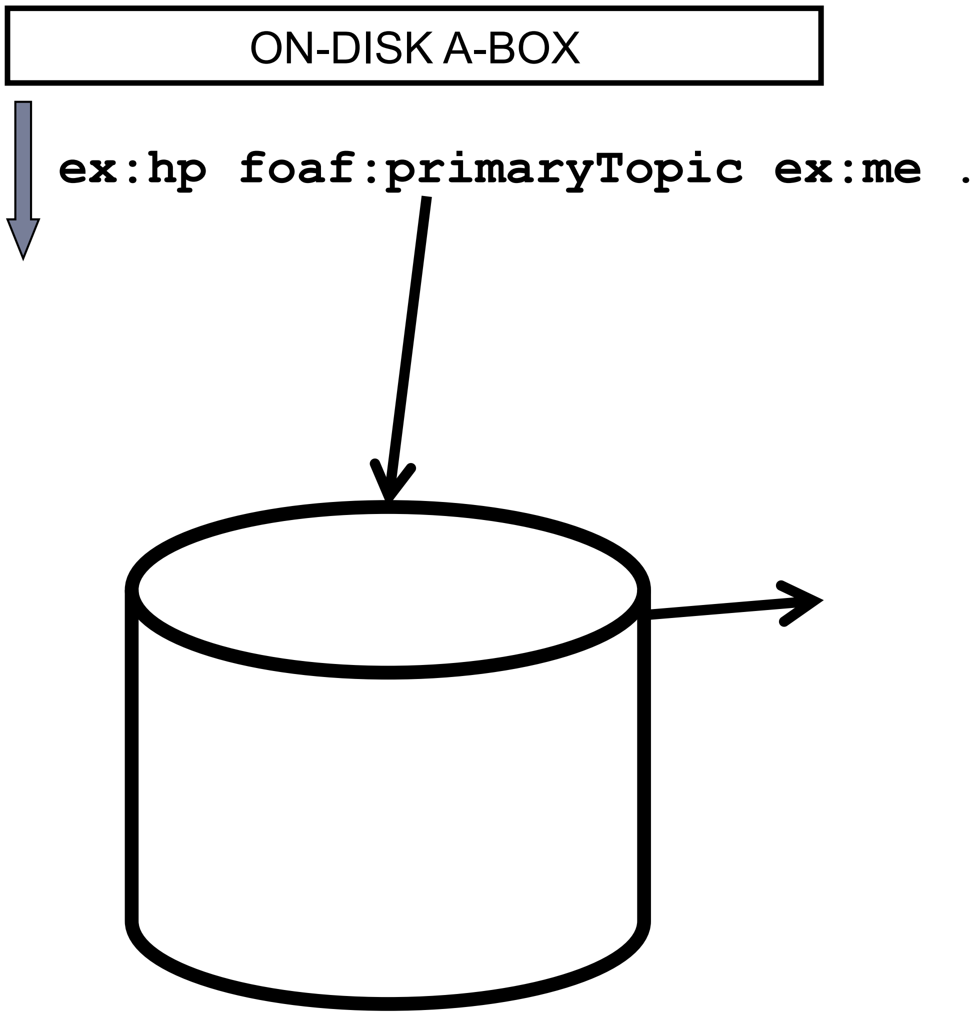
Naïve: Run all rules against all triples

- Scan all of the data
- Try each rule against each input triple
  - If there's an inference, apply recursion

- 301k rules applied to 2.1 billion triples (input+inferred)
  - = 750 trillion rule applications

Projected to take 19 years

Optimisation: **In-memory Rule index …**



foaf:primaryTopic(?x,?y) .
→ foaf:isPrimaryTopicOf(?y,?x) .

foaf:topic(?x,?y) .
→ foaf:page(?y,?x) .

foaf:primaryTopic(?x,?y)
→ foaf:isPrimaryTopicOf(?y,?x)

foaf:homepage(?x,?y)
→ foaf:isPrimaryTopicOf(?x,?y)

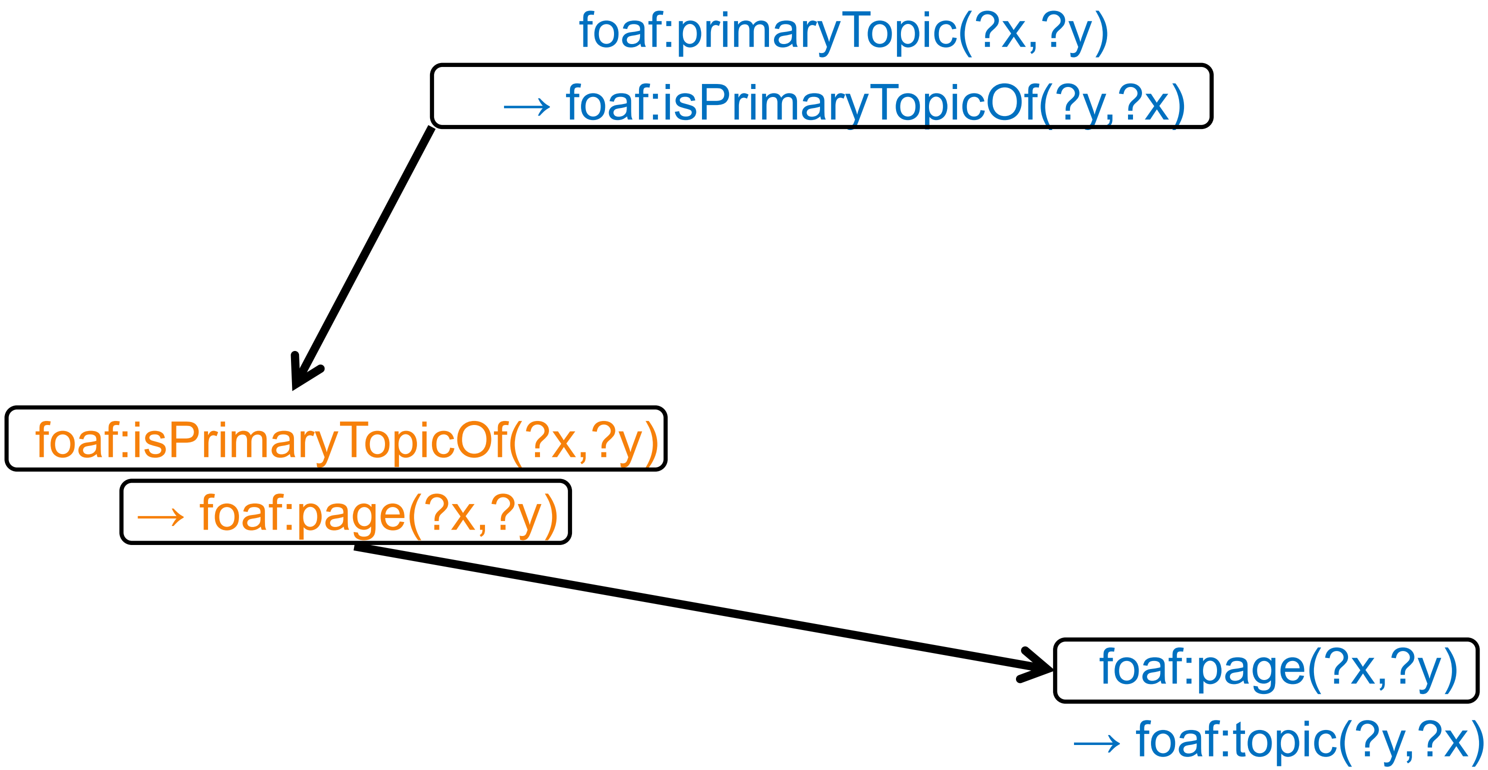
foaf:isPrimaryTopicOf(?x,?y)
→ foaf:page(?x,?y)

foaf:primaryTopic(?x,?y)
→ foaf:topic(?x,?y)

foaf:primaryTopic(?x,?y)
→ foaf:topic(?x,?y)

OWL 2 RL/RDF: prp-inv1, prp-spo1

Optimisation: **Linked Rule index …**



Optimisation: Linked Rule index …

22.1 hours

~~**19 years**~~

SIEMENS DERI

Optimisation: **Merge Rules …**

foaf:primaryTopic(?x,?y)
→ foaf:isPrimaryTopicOf(?y,?x)

+

foaf:primaryTopic(?x,?y)
→ foaf:topic(?x,?y)

↓

foaf:primaryTopic(?x,?y)
→ foaf:isPrimaryTopicOf(?y,?x)
foaf:topic(?x,?y)

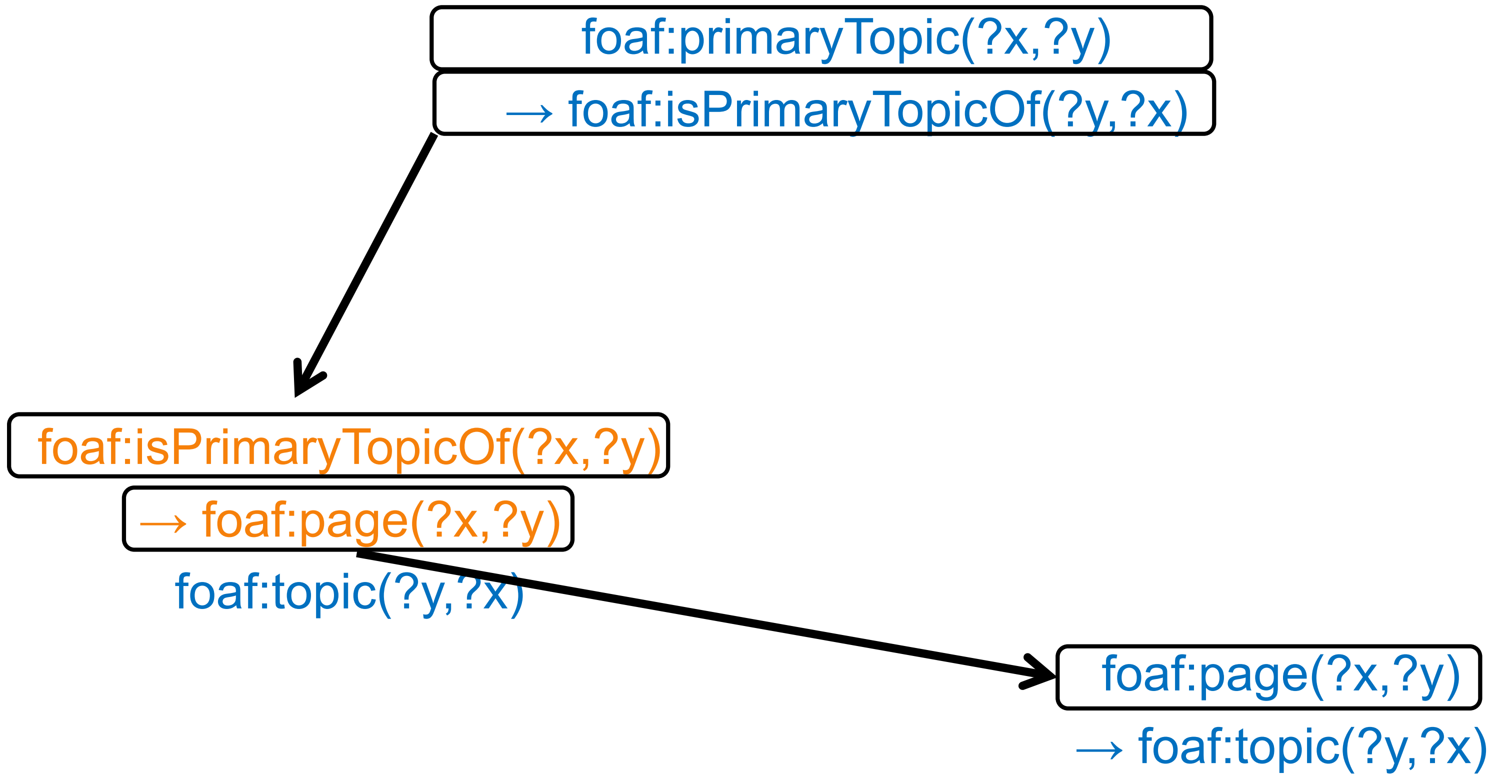
Optimisation: **Merge Rules …**

17.7 hours

~~**22.1 hours**~~

~~**19 years**~~

Failed Optimisation: **Close Rules …**



foaf:primaryTopic(?x,?y)
→ foaf:isPrimaryTopicOf(?y,?x)
foaf:page(?y,?x)
foaf:topic(?x,?y)

SIEMENS DERI

Failed Optimisation: **Close Rules …**

19.5 hours

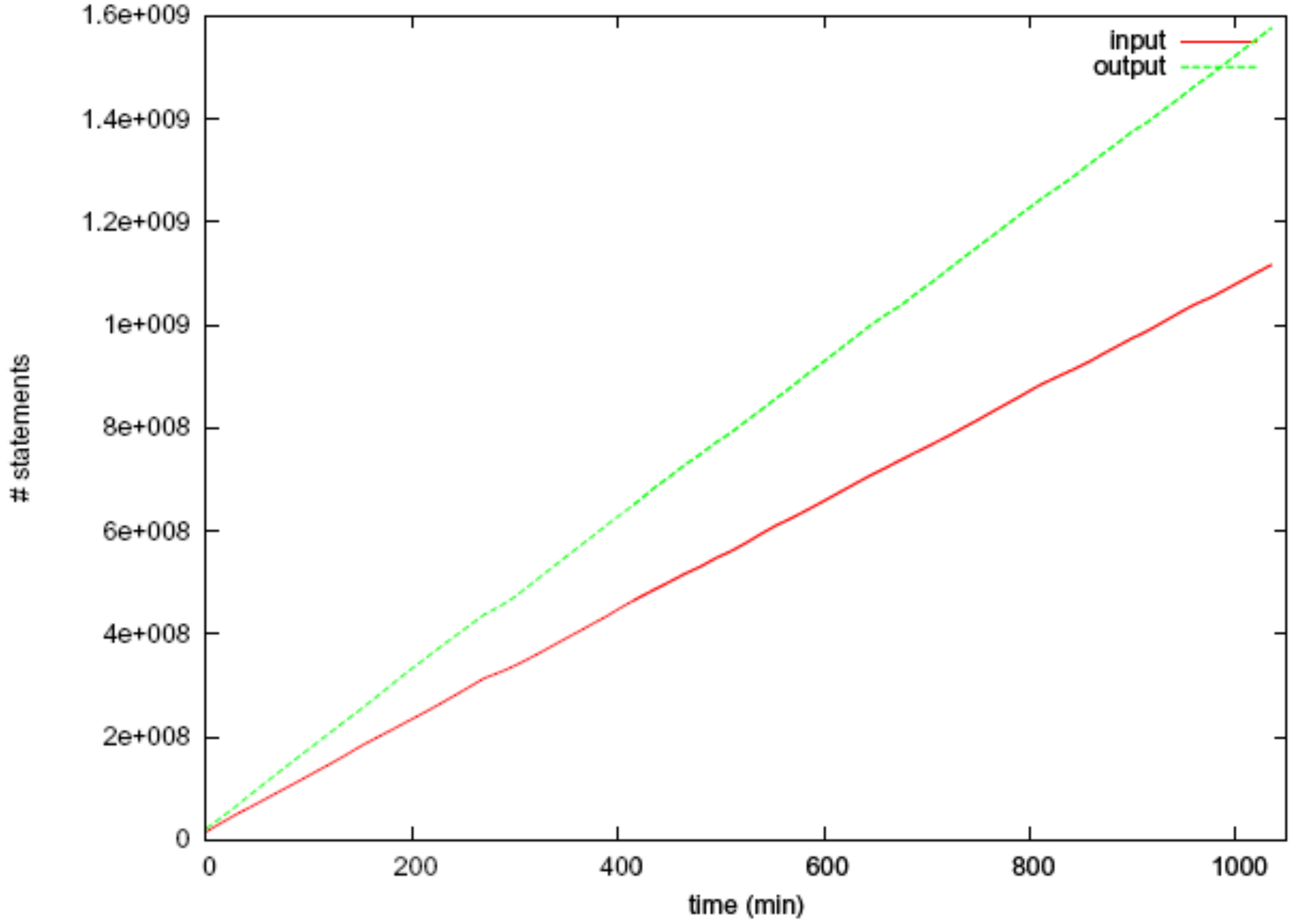
17.7 hours

~~22.1 hours~~

~~19 years~~

SIEMENS DERI

Reasoning Performance: **One Machine …**



SIEMENS DERI

Distributed Computation …

- Eight machines, 4GB main memory, 2.2 GHz

Machines	*Extract T-Box*	*Build T-Box*	*Reason A-Box*	*Total*
1	492	8.9	1062	1565
2	240	10.2	465	719
4	131	10.4	239	383
8	67	9.8	121	201

minutes

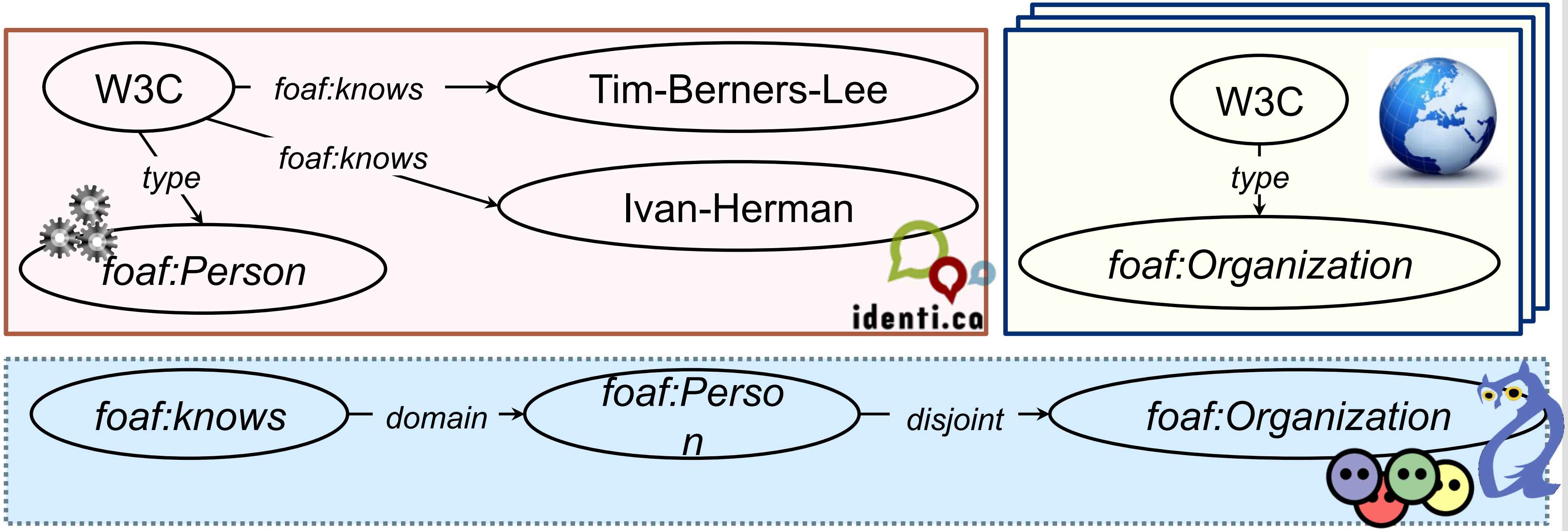
- Fastest:

8 machines: Total 3.35 hours

Reasoning over Web Data

- Challenges
  - C1: Linked Data is **huge**
  - C2: Linked Data is not “pure” OWL (DL)
  - <u>C3: Linked Data is not consistent</u>
    - Linked Data is Web data
  - C4: Linked Data is Evolving
  - C5: Linked Data needs more than RDFS+OWL

Dealing with Inconsistency

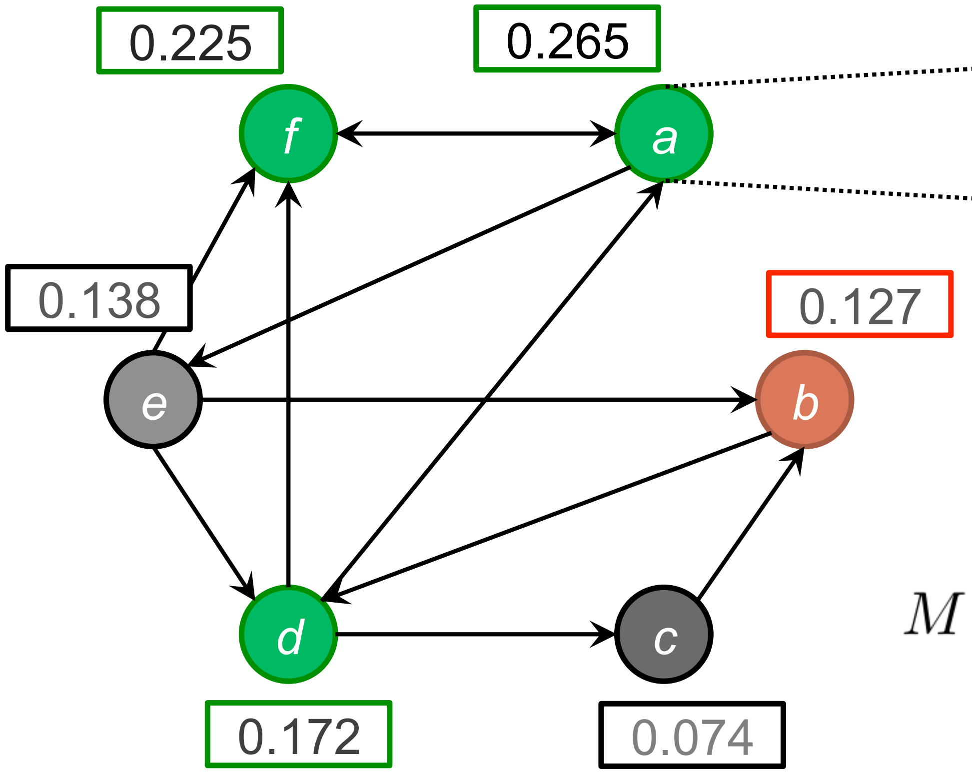


- foaf:Organization(W3C) [W3C is an Organization]
- foaf:knows(W3C,…) [W3C knows various people]
- $\top \sqsubseteq \forall$ foaf:knows⁻.foaf:Person [People know People]
- ∴ foaf:Person(W3C) [W3C is a Person]
- foaf:Person ⊓ foaf:Organization ≡ ⊥ [Person and Organisation are disjoint]
  - (Something can't be both a Person and an Organisation at the same time)

∴ W3C is inconsistent!

What Would Google Do?

PageRank: Compute prominence of documents



$\{f_{a,1} \ldots f_{a,n}\}$

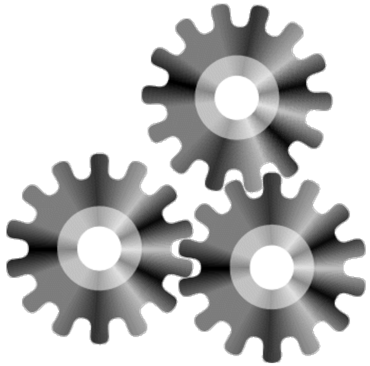
$$M = \begin{array}{c|cccccc} & a & b & c & d & e & f \\ \hline a & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ b & 0 & 0 & 0 & 1 & 0 & 0 \\ c & 0 & 1 & 0 & 0 & 0 & 0 \\ d & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} \\ e & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ f & 1 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Compute probability of being at each document after a “long walk”

SIEMENS DERI

Annotated Logic Program Framework

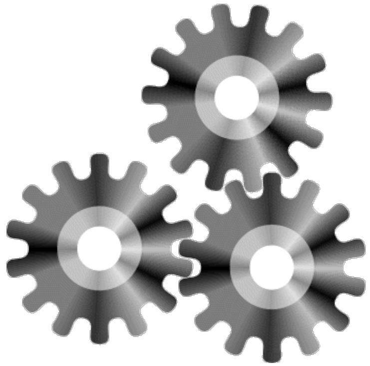
- foaf:Organization(W3C) [0.662]
- foaf:knows(W3C,Tim-Berners-Lee) [0.127]
- foaf:knows(W3C,Ivan-Herman) [0.107]
- $\top \sqsubseteq \forall$ foaf:knows^{-}.foaf:Person [0.8]
  - foaf:knows(?x,?y) → foaf:Person(?x) [0.8]



- ∴ foaf:Person(W3C) [max(min(0.127,0.8),min(0.107,0.8))]
- ∴ foaf:Person(W3C) [0.127]

Dealing with Inconsistency

- foaf:Organization(W3C) [0.662]
- foaf:Person(W3C) [0.127]
- foaf:Person $\sqcap$ foaf:Organization $\equiv \bot$ [0.8]



∴ W3C is inconsistent!

(*Minimal repair based on annotated ranks*)

What if a fact is involved in multiple inconsistencies?

- One "strong" fact vs. multiple weak facts
- Minimal Hitting Set problem
- We implement a greedy strategy: "cheapest first" repair

Inconsistencies Found

~294k ill-typed datatypes

~7k members of disjoint classes

Class 1	Class 2	Violations
foaf:Agent	foaf:Document	3,842
foaf:Document	foaf:Person	2,918
sioc:Container	sioc:Item	128
foaf:Person	foaf:Project	100
ecs:Group	ecs:Individual	38
skos:Concept	skos:Collection	36
foaf:Document	foaf:Project	26
foaf:Organization	foaf:Person	7
sioc:Community	sioc:Item	3
ecs:Fax	ecs:Telephone	3

More Details …

SAOR: Template Rule Optimisations for Distributed Reasoning over 1 Billion Linked Data Triples*

Aidan Hogan[1], Jeff Z. Pan[2], Axel Polleres[1], and Stefan Decker[1]

[1] Digital Enterprise Research Institute, National University of Ireland, Galway
{firstname.lastname}@deri.org
[2] Dpt. of Computing Science, University of Aberdeen
jeff.z.pan@abdn.ac.uk

Abstract. In this paper, we discuss optimisations of rule-based materialisation approaches for reasoning over large static RDF datasets. We generalise and reformalise what we call the "partial-indexing" approach to scalable rule-based materialisation: the approach is based on a separation of terminological data, which has been shown in previous and related works to enable highly scalable and distributable reasoning for specific rulesets; in so doing, we provide some completeness propositions with respect to semi-naïve evaluation. We then show how related work on template rules – T-Box-specific dynamic rulesets created by binding the terminological patterns in the static ruleset – can be incorporated and optimised for the partial-indexing approach. We evaluate our methods using LUBM(10) for RDFS, pD* (OWL Horst) and OWL 2 RL, and thereafter demonstrate pragmatic distributed reasoning over 1.12 billion Linked Data statements for a subset of OWL 2 RL/RDF rules we argue to be suitable for Web reasoning.

1 Introduction

More and more structured data is being published on the Web in conformance with the

Aidan Hogan, Jeff Z. Pan, Axel Polleres and Stefan Decker. "SAOR: TEMPLATE RULE OPTIMISATIONS FOR DISTRIBUTED REASONING OVER 1 BILLION LINKED DATA TRIPLES ". In the Proceedings of the 9th International Semantic Web Conference (ISWC2010), Shanghai, China, November 2010.

More Details …

OWL reasoning with WebPIE: calculating the closure of 100 billion triples

Jacopo Urbani, Spyros Kotoulas, Jason Maassen, Frank van Harmelen, and Henri Bal

Department of Computer Science, Vrije Universiteit Amsterdam,
{j.urbani,kot,j.maassen,frank.van.harmelen,he.bal}@few.vu.nl

Abstract. In previous work we have shown that the MapReduce framework for distributed computation can be deployed for highly scalable inference over RDF graphs under the RDF Schema semantics. Unfortunately, several key optimizations that enabled the scalable RDFS inference do not generalize to the richer OWL semantics. In this paper we analyze these problems, and we propose solutions to overcome them. Our solutions allow distributed computation of the closure of an RDF graph under the OWL Horst semantics.
We demonstrate the WebPIE inference engine, built on top of the Hadoop platform and deployed on a compute cluster of 64 machines. We have evaluated our approach using some real-world datasets (UniProt and LDSR, about 0.9-1.5 billion triples) and a synthetic benchmark (LUBM, up to 100 billion triples). Results show that our implementation is scalable and vastly outperforms current systems when comparing supported language expressivity, maximum data size and inference speed.

Jacopo Urbani, Spyros Kotoulas, Jason Maassen, Frank van Harmelen, Henri E. Bal. "OWL Reasoning with WebPIE: Calculating the Closure of 100 Billion Triples". In the Proceedings of the 7th Extended Semantic Web Conference (ESWC2010), Heraklion, Greece, June 2010.

More Details …

Robust and Scalable Linked Data Reasoning Incorporating Provenance and Trust Annotations

Piero A. Bonatti [a], Aidan Hogan [b], Axel Polleres [b], Luigi Sauro [a]

[a] *Università di Napoli "Federico II", Napoli, Italy*
[b] *Digital Enterprise Research Institute, National University of Ireland, Galway*

Abstract

In this paper, we leverage annotated logic programs for tracking indicators of provenance and trust during reasoning, specifically focussing on the use-case of applying a scalable subset of OWL 2 RL/RDF rules over static corpora of arbitrary Linked Data (Web data). Our annotations encode three facets of information: (i) *blacklist*: a (possibly manually generated) boolean annotation which indicates that the referent data are known to be harmful and should be ignored during reasoning; (ii) *ranking*: a numeric value derived by a PageRank-inspired technique—adapted for Linked Data—which determines the centrality of certain data artefacts (such as RDF documents and statements); (iii) *authority*: a boolean value which uses Linked Data principles to *conservatively* determine whether or not some terminological information can be trusted. We formalise a logical framework which annotates inferences with the *strength* of derivation along these dimensions of trust and provenance; we formally demonstrate some desirable properties of the deployment of annotated logic programming in our setting, which guarantees (i) a unique minimal model (least fixpoint); (ii) monotonicity; (iii) finitariness; and (iv) finally decidability. In so doing, we also give some formal results which reveal strategies for scalable and efficient implementation of various reasoning tasks one might consider. Thereafter, we discuss scalable and distributed implementation strategies for applying our ranking and reasoning methods over a cluster of commodity hardware; throughout, we provide evaluation of our methods over 1 billion Linked Data quadruples crawled from approximately 4 million individual Web documents, empirically demonstrating the scalability of our approach, and how our annotation values help ensure a more robust form of reasoning. We finally sketch, discuss and evaluate a use-case for a simple repair of inconsistencies detectable within OWL 2 RL/RDF constraint rules using ranking annotations to detect and defeat the "marginal view", and in so doing, infer an empirical "consistency threshold" for the Web of Data in our setting.

Key words: annotated programs; linked data; web reasoning; scalable reasoning; distributed reasoning; authoritative reasoning; owl 2 rl; provenance; pagerank; inconsistency; repair

Piero A. Bonatti, Aidan Hogan, Axel Polleres and Luigi Sauro. "ROBUST AND SCALABLE LINKED DATA REASONING INCORPORATING PROVENANCE AND TRUST ANNOTATIONS". In the Journal of Web Semantics 9(2): pp. 165–201, 2011.

SIEMENS DERI

More Details …

Scalable and Distributed Methods for Entity Matching, Consolidation and Disambiguation over Linked Data Corpora

Aidan Hogan [a], Antoine Zimmermann [b], Jürgen Umbrich [a], Axel Polleres [c], Stefan Decker [a],

[a]*Digital Enterprise Research Institute, National University of Ireland, Galway*
[b]*INSA-Lyon, LIRIS, UMR5205, F-69621, France*
[c]*Siemens AG Österreich, Siemensstrasse 90, 1210 Vienna, Austria*

Abstract

With respect to large-scale, static, Linked Data corpora, in this paper we discuss scalable and distributed methods for entity consolidation (aka. smushing, entity resolution, object consolidation, etc.) to locate and process names that signify the same entity. We investigate (i) a baseline approach, which uses explicit `owl:sameAs` relations to perform consolidation; (ii) extended entity consolidation which additionally uses a subset of OWL 2 RL/RDF rules to derive novel `owl:sameAs` relations through the semantics of inverse-functional properties, functional-properties and (max-)cardinality restrictions with value one; (iii) deriving weighted concurrence measures between entities in the corpus based on shared inlinks/outlinks and attribute values using statistical analyses; (iv) disambiguating (initially) consolidated entities based on inconsistency detection using OWL 2 RL/RDF rules. Our methods are based upon distributed sorts and scans of the corpus, where we deliberately avoid the requirement for indexing all data. Throughout, we offer evaluation over a diverse Linked Data corpus consisting of 1.118 billion quadruples derived from a domain-agnostic, open crawl of 3.985 million RDF/XML Web documents, demonstrating the feasibility of our methods at that scale, and giving insights into the quality of the results for real-world data.

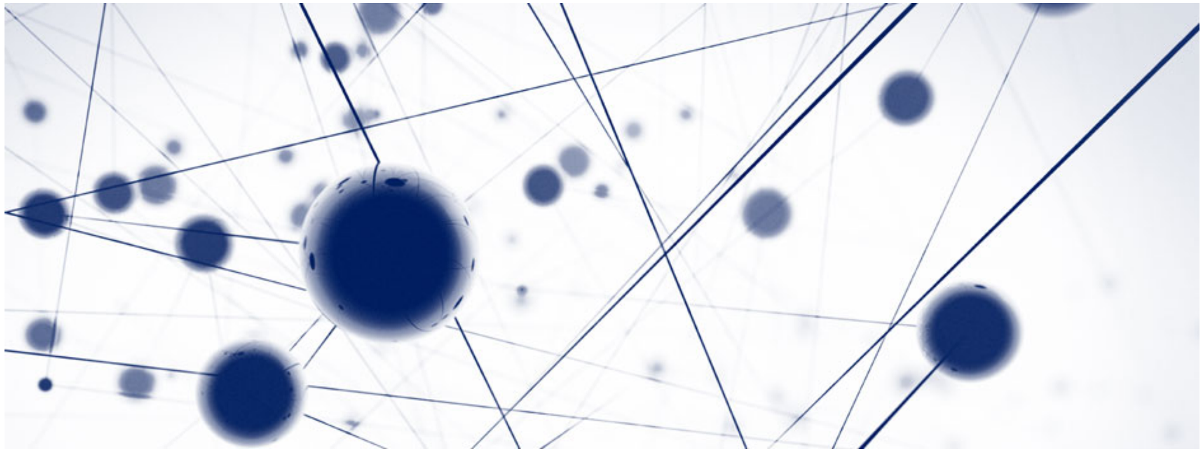
Key words: entity consolidation, web data, linked data, rdf

1. Introduction

Over a decade since the dawn of the Semantic Web,

The Linked Open Data project has advocated the goal of providing dereferencable machine readable data in a common format (RDF), with emphasis on the re-use

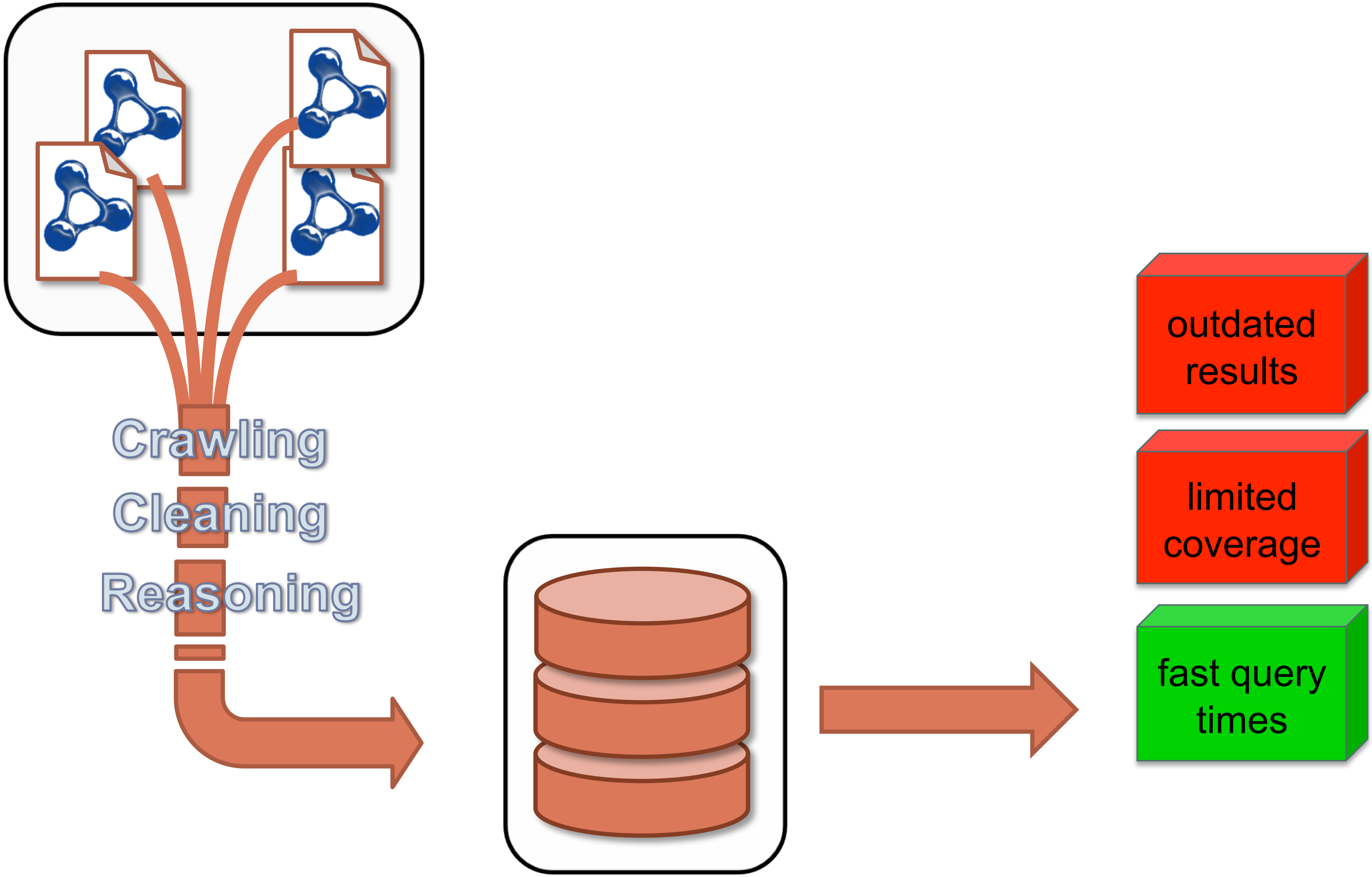
Aidan Hogan, Antoine Zimmermann, Jürgen Umbrich, Axel Polleres and Stefan Decker. "SCALABLE AND DISTRIBUTED METHODS FOR ENTITY MATCHING, CONSOLIDATION AND DISAMBIGUATION OVER LINKED DATA CORPORA ". In the Journal of Web Semantics 10: pp. 76–110, 2012



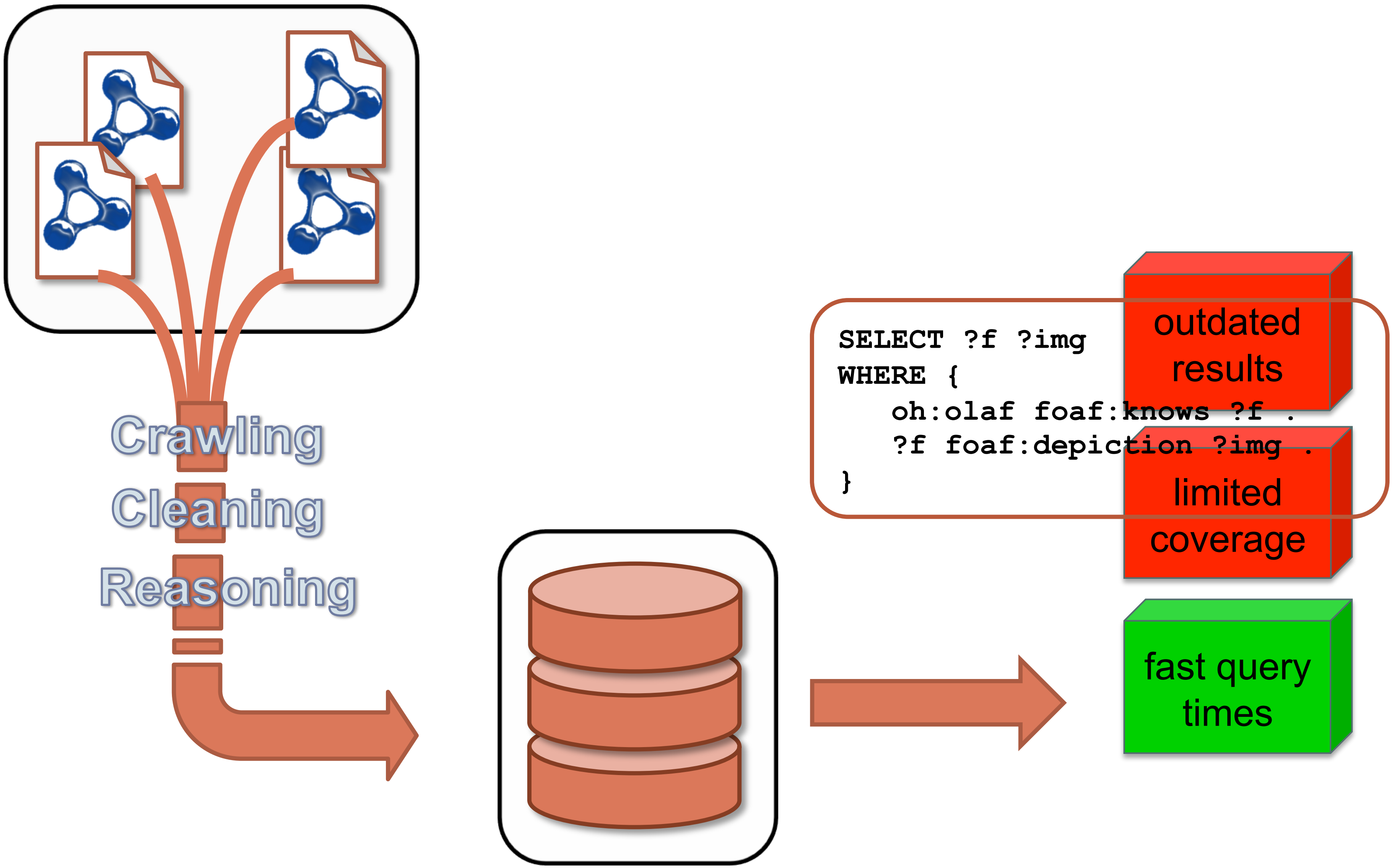
Look mommy, no warehouse

LIVE REASONING

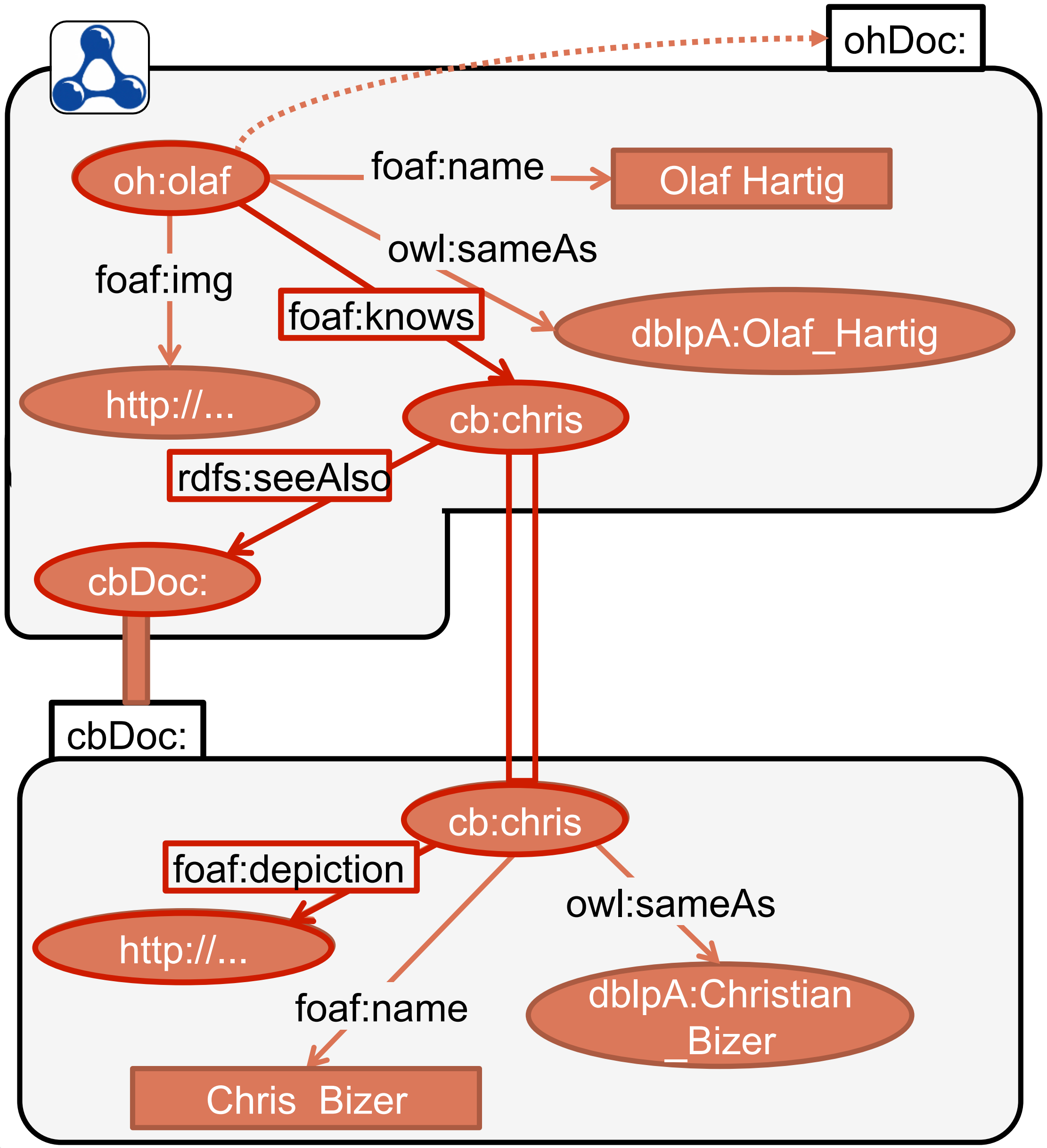
Problems with Warehousing



Problems with Warehousing (what if …)



LTBQE: Link Traversal Based Query Execution



Use Linked Data principles:

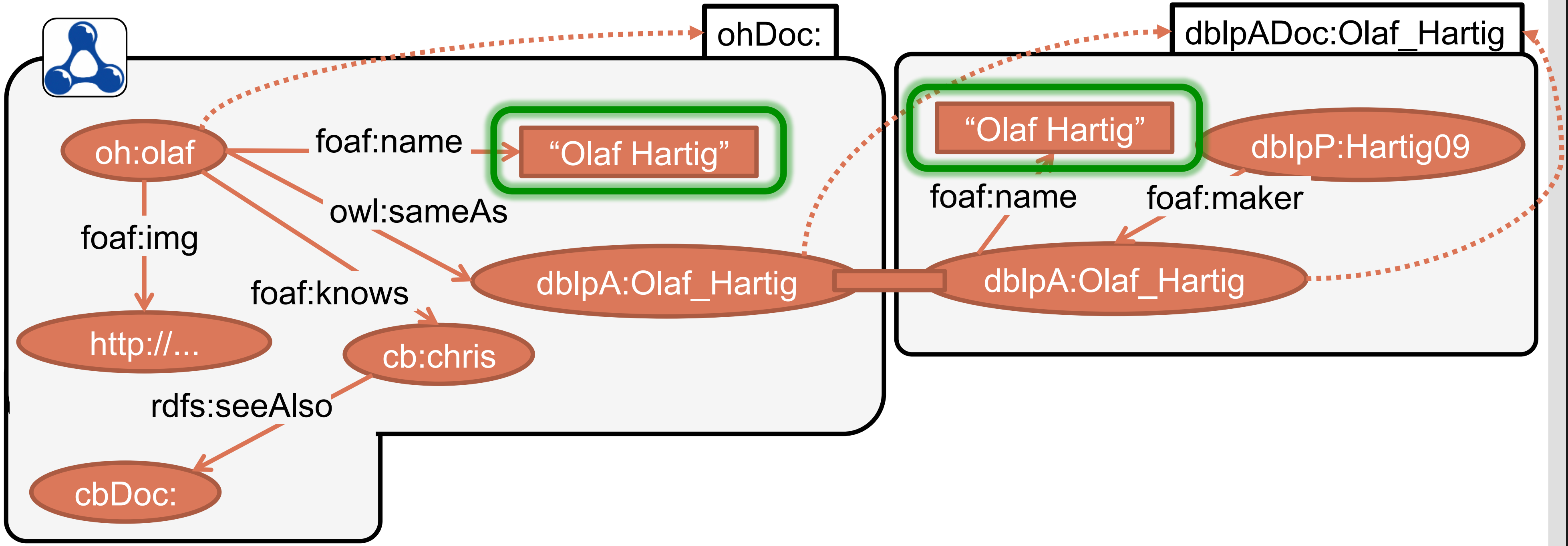
- ❑ dereferencing URIs
- ❑ following links

```
SELECT ?f ?img
WHERE {
    oh:olaf foaf:knows ?f .
    ?f foaf:depiction ?img .
}
```

?f	?img
cb:chris	http://..

SIEMENS DERI

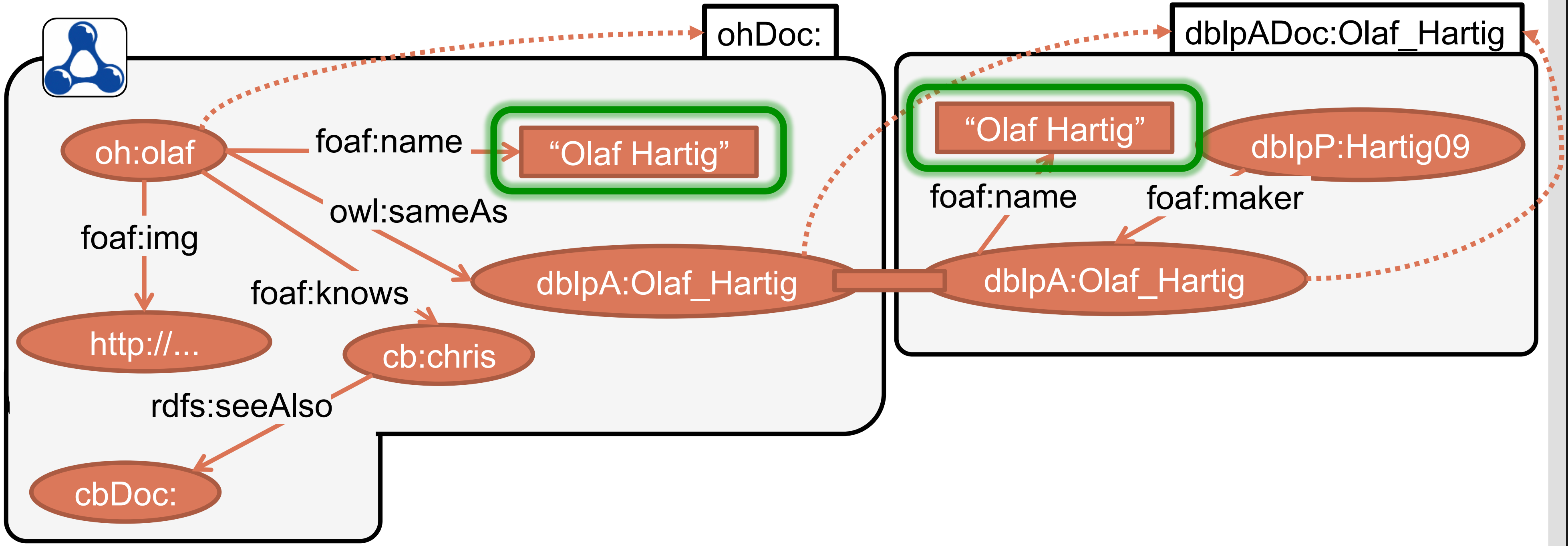
Limitations of LTBQE …



No URI to Deref.

```
SELECT  ?p2
WHERE {
   ?person foaf:name ?name .
}
```

Limitations of LTBQE …

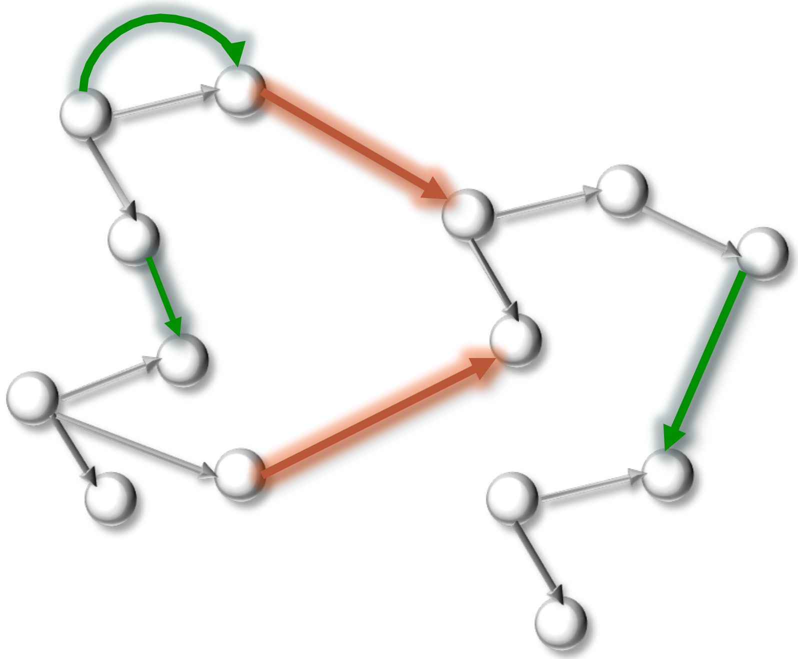


Join over Literal

```
SELECT  ?p2
WHERE {
    oh:olaf foaf:name ?name .
    ?p2 foaf:name ?name    .
}
```

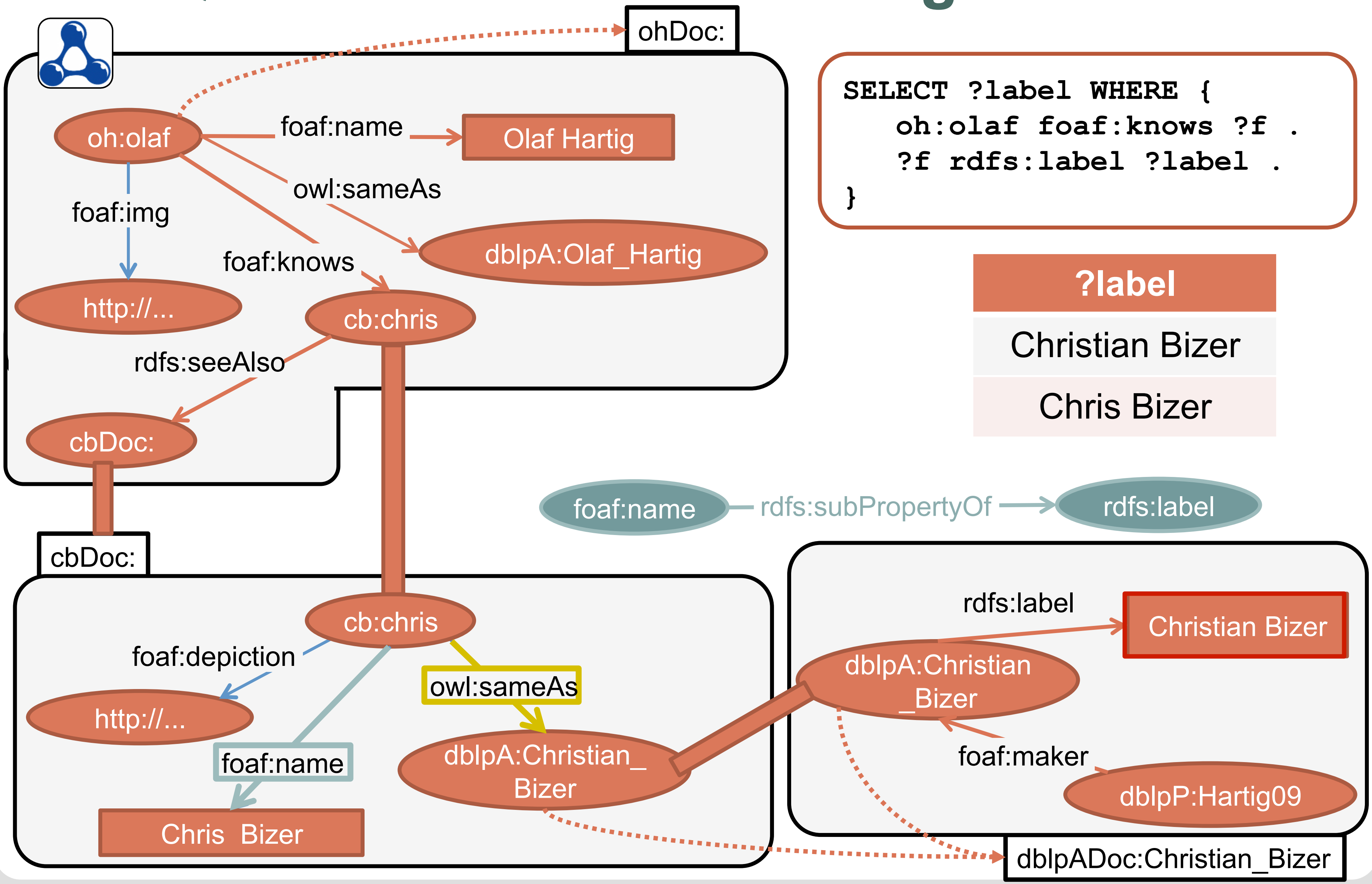
LTBQE Practical Issues …

- Query time is influenced by
  - Source selection
  - Number of sequential HTTP lookups
- Result Recall is influenced by
  - Dereferenceability
  - Execution Order
  - Query features
  - Connectivity
- Reasoning can find new connections:
  - **`owl:sameAs` reasoning**
  - **Lightweight RDFS reasoning**



SIEMENS DERI

LTBQE: Add "Live" Reasoning




```
SELECT ?label WHERE {
   oh:olaf foaf:knows ?f .
   ?f rdfs:label ?label .
}
```

?label
Christian Bizer
Chris Bizer

LTBQE Analysis: Benefits of Reasoning

position	%URIs	data increase
<u> rdfs:seeAlso ?o	2%	1.006x
<u> owl:sameAs ?o	16%	2.5x
RDFS reasoning*	81%	1.78 x

*rdfs:subClassOf, rdfs:subPropertyOf, rdfs:domain, rdfs:range
Using static schema data

SIEMENS DERI

Query Benchmark

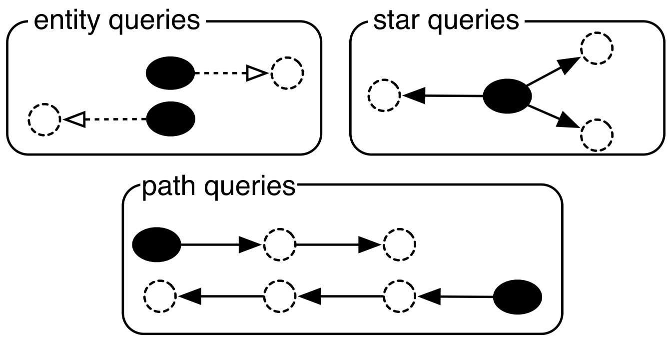
How does reasoning perform in practice?

Existing benchmarks target either a single domain or provide only a few queries.

BTC 2011

QWalk:
Random walk based query generation.

1100 queries
100 each for
11 shapes



Run them live.

Query Throughput: Avg. Results/Second

worst → best

	LTBQE	Base	seeAlso	sameAs	RDFS	Comb
entity-s	1.00	1.68	1.67	2.15	1.29	1.53
entity-o	3.97	6.48	6.16	5.70	5.37	4.33
entity-so	2.02	2.82	2.66	3.71	3.73	4.80
star-3-0	0.11	0.16	0.15	0.15	0.24	0.20
star-2-1	0.58	1.12	1.00	1.04	2.14	1.75
star-1-2	0.17	1.60	1.35	1.60	70.97	58.85
star-0-3	0.18	0.35	0.33	0.94	0.24	0.68
s-path-2	0.44	0.72	0.68	0.70	0.83	0.78
s-path-3	1.76	2.45	2.56	2.46	2.43	2.10
o-path-2	1.38	8.39	7.76	10.55	6.36	6.89
o-path-3	0.95	5.70	5.84	6.08	5.04	4.68

Overall average query time of ~12 seconds.

More Details …

SPARQL for a Web of Linked Data: Semantics and Computability

Olaf Hartig

Humboldt-Universität zu Berlin
hartig@informatik.hu-berlin.de

Abstract. The World Wide Web currently evolves into a Web of Linked Data where content providers publish and link data as they have done with hypertext for the last 20 years. While the declarative query language SPARQL is the de facto for querying a-priory defined sets of data from the Web, no language exists for querying the Web of Linked Data itself. However, it seems natural to ask whether SPARQL is also suitable for such a purpose.
In this paper we formally investigate the applicability of SPARQL as a query language for Linked Data on the Web. In particular, we study two query models: 1) a *full-Web semantics* where the scope of a query is the complete set of Linked Data on the Web and 2) a family of *reachability-based semantics* which restrict the scope to data that is reachable by traversing certain data links. For both models we discuss properties such as monotonicity and computability as well as the implications of querying a Web that is infinitely large due to data generating servers.

1 Introduction

The emergence of vast amounts of RDF data on the WWW has spawned research on storing and querying large collections of such data efficiently. The prevalent query language in this context is SPARQL [16] which defines queries as functions over an RDF dataset, that is, a fixed, a-priory defined collection of sets of RDF triples. This definition naturally fits the use case of querying a repository of RDF data copied from the Web.

However, most RDF data on the Web is published following the Linked Data principles [5], contributing to the emerging Web of Linked Data [6]. This practice allows for query approaches that access the most recent version of remote data on demand. More importantly, query execution systems may automatically discover new data by … not only up-to-date but may also include initially unknown data. These features are the … of querying the Web itself, instead of a predefined, bounded collection of data.

Olaf Hartig. “SPARQL FOR A WEB OF LINKED DATA: SEMANTICS AND COMPUTABILITY". In the Proceedings of the 9th Extended Semantic Web Conference (ESWC). 2012.

SIEMENS DERI

More Details …

On Completeness Classes for Query Evaluation on Linked Data

Andreas Harth and **Sebastian Speiser**
Institute AIFB
Karlsruhe Institute of Technology (KIT)
76128 Karlsruhe, Germany

Abstract

The advent of the Web of Data kindled interest in link-traversal (or lookup-based) query processing methods, with which queries are answered via dereferencing a potentially large number of small, interlinked sources. While several algorithms for query evaluation have been proposed, there exists no notion of completeness for results of so-evaluated queries. In this paper, we motivate the need for clearly-defined completeness classes and present several notions of completeness for queries over Linked Data, based on the idea of authoritativeness of sources, and show the relation between the different completeness classes.

1 Introduction

A tenet in work on query evaluation and reasoning on the Semantic Web is the open world assumption (OWA): given the size and decentralised nature of the web, it is impossible to achieve complete results. Thus, an answer to a query or reasoning task is therefore always a subset of all possible answers. To what degree that subset is complete is left open.

In this paper we define more fine-grained completeness classes for query answers. We do so in the context of Linked Data, a set of principles detailing how to publish …

- systems can implement operations that rely on checking for the absence of results, such as negation-as-failure; and
- certain statements can be restricted to trustworthy sources.

Our specific contributions are:

- We extend and generalise the idea of authoritative sources from (Hogan, Harth, and Polleres 2009); based on authority, we define the notion of completeness for sources.
- We define three completeness classes for triple patterns and conjunctive queries: one that considers the entire web, one that considers documents in the surrounding of sources derived from the query and one that considers documents based on the query execution.
- We show how the completeness classes related to each other.

Please note that our results apply to both web and intranet environments, as long as data providers follow Linked Data principles. Our results also apply to Dataspaces (Franklin, Halevy, and Maier 2005) without central registries.

The remainder of the paper is organised as follows: Section 2 provides an example. Section 3 introduces necessary … the relation between the completeness classes. Section 8 presents related work, and Section 9 concludes.

Andreas Harth, Sebastian Speiser. "ON COMPLETENESS CLASSES FOR QUERY EVALUATION ON LINKED DATA.". In the Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI 2012), Toronto, Ontario, Canada, 22–26 July, 2012.

SIEMENS DERI

More Details …

Improving the Recall of Live Linked Data Querying through Reasoning

Jürgen Umbrich[1], Aidan Hogan[1], Axel Polleres[2], Stefan Decker[1]

[1] Digital Enterprise Research Institute, National University of Ireland, Galway
[2] Siemens AG Österreich, Siemensstrasse 90, 1210 Vienna, Austria
Emails: {*firstname.lastname*}@deri.org, axel.polleres@siemens.com

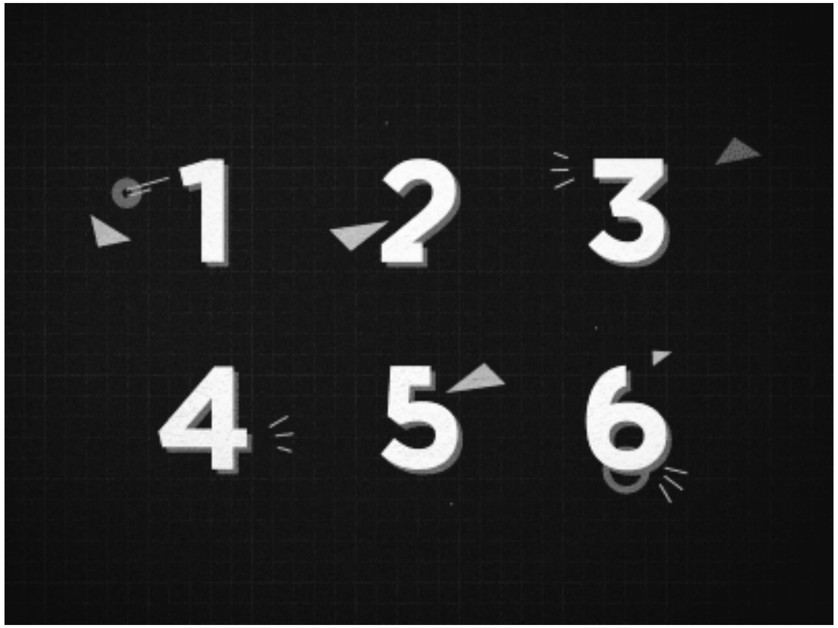
Abstract. Linked Data principles allow for processing SPARQL queries on-the-fly by dereferencing URIs. Link-traversal query approaches for Linked Data have the benefit of up-to-date results and decentralised execution, but operate only on explicit data from dereferenced documents, affecting recall. In this paper, we show how inferable knowledge—specifically that found through `owl:sameAs` and RDFS reasoning—can improve recall in this setting. We first analyse a corpus featuring 7 million Linked Data sources and 2.1 billion quadruples: we (1) measure expected recall by only considering dereferenceable information, (2) measure the improvement in recall given by considering `rdfs:seeAlso` links as previous proposals did. We further propose and measure the impact of additionally considering (3) `owl:sameAs` links, and (4) applying lightweight RDFS reasoning for finding more results, relying on static schema information. We evaluate different configurations for live queries covering different shapes and domains, generated from random walks over our corpus.

1 Introduction

Recently, a rich lode of RDF data has been published on the Web as *Linked Data* by governments, academia, industry, communities and individuals alike [15].

(through HTTP) to the physical location of structured information about them, information published as Linked Data can be viewed as forming a scale-free,

Jürgen Umbrich, Aidan Hogan, Axel Polleres and Stefan Decker. "IMPROVING THE RECALL OF LIVE LINKED DATA QUERYING THROUGH REASONING.". In the Proceedings of the 6th International Conference on Web Reasoning and Rule Systems (RR 2012), Vienna, Austria, 10–12 September, 2012.



Without using fingers

NUMERIC REASONING …

C5: Linked data needs more than RDFS+OWL

Google

"Latest News on NYT about companies with a revenue greater than 10B **EUR**"

```
SELECT * WHERE
{ ?C rdf:type NYT:Org .
  ?C dbpedia:revenueEUR ?R .
  ?C NYT:latestArticle ?A .
  FILTER( ?R > 1E10 )   }
```

- *There is implicit knowledge not expressible in OWL, e.g. in the form of **attribute equations***

*dbo:revenueUSD = dbo:revenueEUR * 1.3 .*
dbo:profitEUR = "dbo:revenueEUR – dbo:totalExpensesEUR".

Questions:

- *Can such equational knowledge co-exist with OWL?*
- *Can rule-based materialization and/or query rewriting be used to exploit it?*

1.3 **USD** = 1 **EUR**

dbr:IBM a dbo:Organisation.
dbr:IBM dbo:revenueUSD 1.06916E11

SIEMENS DERI

Extending RDFS by attribute equations:

RDFS with Attribute Equations via SPARQL Rewriting

Stefan Bischof[1,2] and Axel Polleres[1]

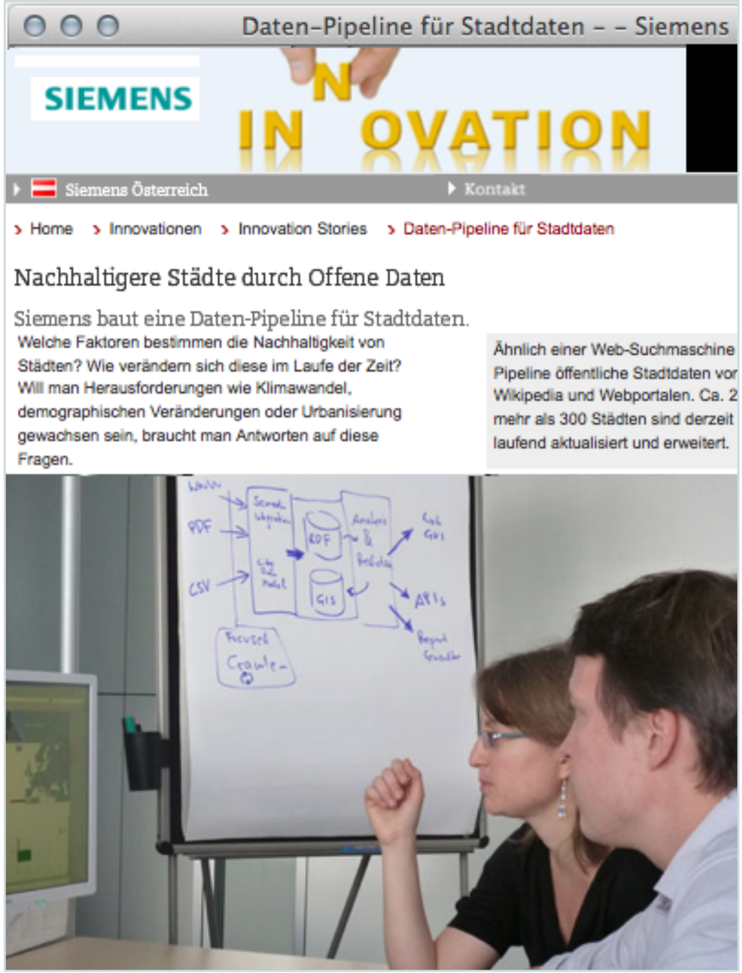
[1] Siemens AG Österreich, Siemensstraße 90, 1210 Vienna, Austria
[2] Vienna University of Technology, Favoritenstraße 9, 1040 Vienna, Austria

Abstract. In addition to taxonomic knowledge about concepts and properties typically expressible in languages such as RDFS and OWL, implicit information in an RDF graph may be likewise determined by arithmetic equations. The main use case here is exploiting knowledge about functional dependencies among numerical attributes expressible by means of such equations. While some of this knowledge can be encoded in rule extensions to ontology languages, we provide an arguably more flexible framework that treats attribute equations as first class citizens in the ontology language. The combination of ontological reasoning and attribute equations is realized by extending query rewriting techniques already successfully applied for ontology languages such as (the DL-Lite-fragment of) RDFS or OWL, respectively. We deploy this technique for rewriting SPARQL queries and discuss the feasibility of alternative implementations, such as rule-based approaches.

1 Introduction

A wide range of literature has discussed completion of data represented in RDF with ... omic reasoning within a

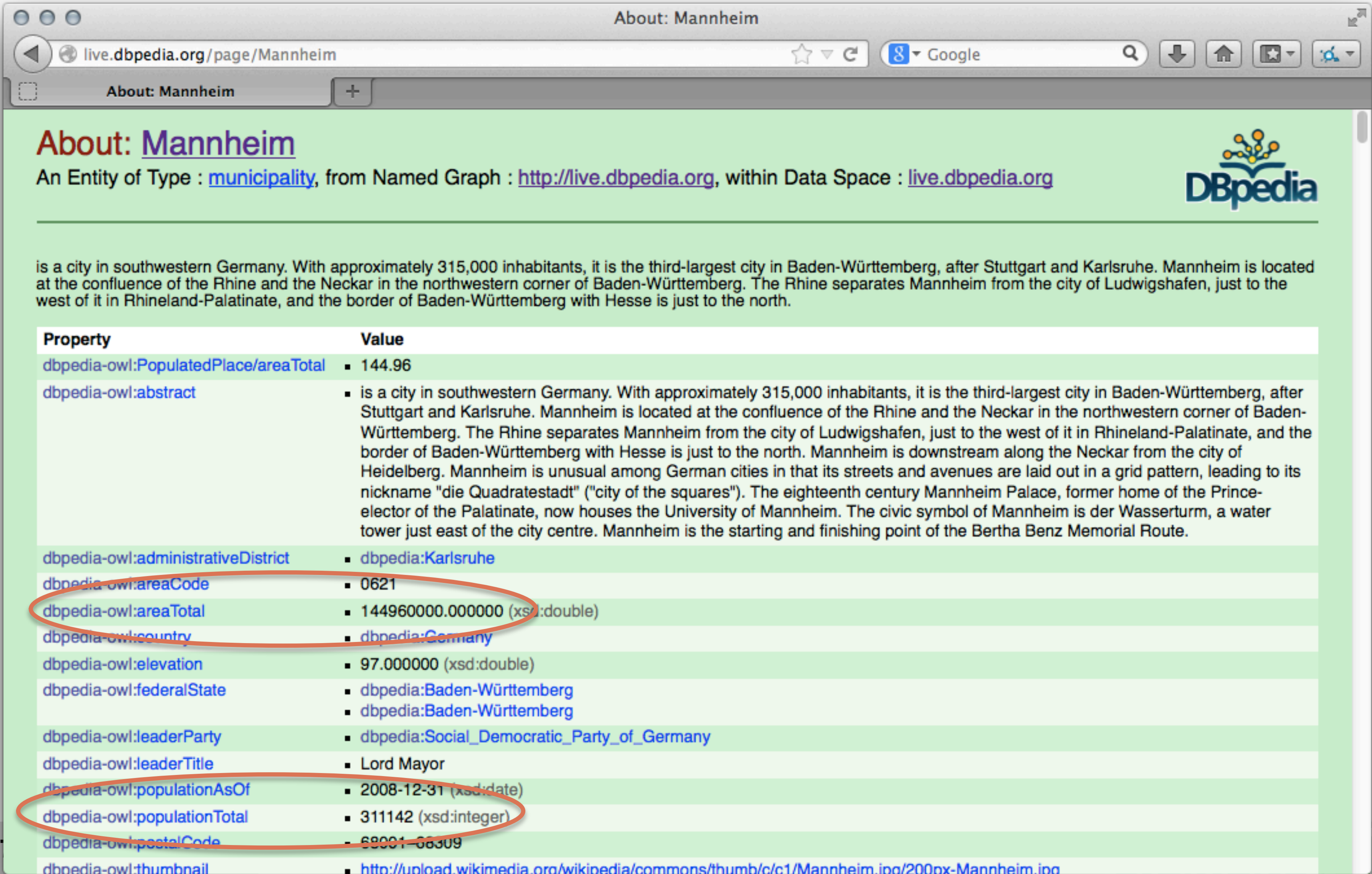
Stefan Bischof, Axel Polleres. ESWC2013



SIEMENS DERI

- Lot's of numeric data out there in linked data that's "convertible" by such equations… e.g. http://live.dbpedia.org/Mannheim

  has ***dbo:areaTotal dbo:populationTotal***, but misses ***dbo:populationDensity***



About: Mannheim

live.dbpedia.org/page/Mannheim

About: Mannheim

An Entity of Type : municipality, from Named Graph : http://live.dbpedia.org, within Data Space : live.dbpedia.org

is a city in southwestern Germany. With approximately 315,000 inhabitants, it is the third-largest city in Baden-Württemberg, after Stuttgart and Karlsruhe. Mannheim is located at the confluence of the Rhine and the Neckar in the northwestern corner of Baden-Württemberg. The Rhine separates Mannheim from the city of Ludwigshafen, just to the west of it in Rhineland-Palatinate, and the border of Baden-Württemberg with Hesse is just to the north.

Property	Value
dbpedia-owl:PopulatedPlace/areaTotal	144.96
dbpedia-owl:abstract	is a city in southwestern Germany. With approximately 315,000 inhabitants, it is the third-largest city in Baden-Württemberg, after Stuttgart and Karlsruhe. Mannheim is located at the confluence of the Rhine and the Neckar in the northwestern corner of Baden-Württemberg. The Rhine separates Mannheim from the city of Ludwigshafen, just to the west of it in Rhineland-Palatinate, and the border of Baden-Württemberg with Hesse is just to the north. Mannheim is downstream along the Neckar from the city of Heidelberg. Mannheim is unusual among German cities in that its streets and avenues are laid out in a grid pattern, leading to its nickname "die Quadratestadt" ("city of the squares"). The eighteenth century Mannheim Palace, former home of the Prince-elector of the Palatinate, now houses the University of Mannheim. The civic symbol of Mannheim is der Wasserturm, a water tower just east of the city centre. Mannheim is the starting and finishing point of the Bertha Benz Memorial Route.
dbpedia-owl:administrativeDistrict	dbpedia:Karlsruhe
dbpedia-owl:areaCode	0621
dbpedia-owl:areaTotal	144960000.000000 (xsd:double)
dbpedia-owl:country	dbpedia:Germany
dbpedia-owl:elevation	97.000000 (xsd:double)
dbpedia-owl:federalState	dbpedia:Baden-Württemberg dbpedia:Baden-Württemberg
dbpedia-owl:leaderParty	dbpedia:Social_Democratic_Party_of_Germany
dbpedia-owl:leaderTitle	Lord Mayor
dbpedia-owl:populationAsOf	2008-12-31 (xsd:date)
dbpedia-owl:populationTotal	311142 (xsd:integer)
dbpedia-owl:postalCode	68001–68309
dbpedia-owl:thumbnail	http://upload.wikimedia.org/wikipedia/commons/thumb/c/c1/Mannheim.jpg/200px-Mannheim.jpg

Can equational knowledge co-exist with OWL?

- *Can equational knowledge co-exist with OWL?*
  - *We need a syntax & define a formal semantics*

- *Syntax:*

*dbo:revenueUSD = dbo:revenueEUR * 1.3.*
dbo:profitEUR = "dbo:revenueEUR – dbo:totalExpensesEUR".
dbo:populationDensity = "dbo:populationTotal / dbo:areaTotal".

dbo:revenueUSD **:defineByEquation** "dbo:revenueEUR * 1.3" .
dbo:revenueEUR **:defineByEquation** "dbo:revenueEUR - dbo:totalExpensesEUR" .
dbo:populationDensity **:defineByEquation** "dbo:areaTotal / dbo:populationTotal" .

- Semantics:
  - Requirements:
    - "Fit" with common model-theoretic semantics for OWL and RDFS
    - Treat equivalent equations equivalently:

*dbo:revenueUSD = dbo:revenueEUR * 1.3.*

dbo:revenueEUR = dbo:revenuUSD / 1.3 .

Can equational knowledge co-exist with OWL?

For those with a DL background (others stay tuned until tomorrow's lecture)

- An Interpretation $\mathcal{I}$ interpret datatype properties U as binary relations between domain elements and Data-Values *(for simple equations rational numbers are sufficient)*:

$$U^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \mathbb{Q}$$

dbo:population rdfs:subPropertyOf dbo:populationTotal .

- Interpretations of inclusion axioms are as usual, e.g.
  - A sub-property axiom **sp**

dbr:Mannheim dbo:population 311142 .

U_1 **rdfs:subPropertyOf** U_2 $\quad U_1 \sqsubseteq U_2$

is satisfied in $\mathcal{I}$ *if* $U_1^{\mathcal{I}} \subseteq U_2^{\mathcal{I}}$

dbr:Mannheim dbo:populationTotal 311142.

dbo:populationDensity :definedByEquation "dbo:populationTotal / dbo:areaTotal" .

- **NEW:** A property equation axiom ***e***

U_0 **:defineByEquation** "$f(U1,\ldots U_n)$" .

dbr:Mannheim dbo:populationTotal 311142 .
dbr:Mannheim dbo:areaTotal 144.96 .

is satisfied in $\mathcal{I}$

$$\text{if } \forall x, y_1, \ldots, y_n (\bigwedge_{i=1}^{n} (x, y_i) \in U_i^{\mathcal{I}}) \wedge \text{defined}(f(U_1/y_1, \ldots, U_n/y_n))$$

dbr:Mannheim dbo:populationDensity 2146.39 .

$$\Rightarrow (x, \text{eval}(f(U_1/y_1, \ldots, U_n/y_n)) \in U_0^{\mathcal{I}}$$

- An interpretation $\mathcal{I}$ is a model it satisfies
  - all inclusion axioms
  - ***all variants of*** all equation axioms

Can equational knowledge co-exist with OWL?

- An Interpretation $\mathcal{I}$ interpret datatype properties U as binary relations between domain elements and Data-Values (for our simple equations rational numbers are sufficient): $U^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \mathbb{Q}$
- Interpretations of inclusion axioms are as usual, e.g.
  - A sub-property axiom **sp**

  U_1 **rdfs:subPropertyOf** U_2 $\quad U_1 \sqsubseteq U_2$

  is satisfied in $\mathcal{I}$ *if* $U_1^{\mathcal{I}} \subseteq U_2^{\mathcal{I}}$

dbo:populationDensity :definedByEquation “dbo:populationTotal / dbo:areaTotal” .

- **NEW:** A property equation axiom **e**

  U_0 **:defineByEquation** “*f(U1,…U$_n$)*” .

  is satisfied in $\mathcal{I}$

  $$\text{if } \forall x, y_1, \ldots, y_n (\bigwedge_{i=1}^{n} (x, y_i) \in U_i^{\mathcal{I}}) \wedge \text{defined}(f(U_1/y_1, \ldots, U_n/y_n))$$

dbr:Mannheim dbo:populationTotal 311142 .
dbr:Mannheim dbo:areaTotal 0.

dbo:populationTotal :definedByEquation “dbo:populationDensity * dbo:areaTotal”.
dbo:areaTotal :definedByEquation “dbo:populationTotal / dbo:populationDensity” .

- An interpretation $\mathcal{I}$ is a model it satisfies
  - all inclusion axioms
  - ***all variants of*** all equation axioms

Can materialization and/or query rewriting be used?

- Rule-based Materialization:

$(S, \text{popDensity}, PD) \leftarrow (S, \text{population}, P), (S, \text{area}, A),\ PD := P/A,\ A \neq 0.$

$(S, \text{area}, PD) \leftarrow (S, \text{population}, P), (S, \text{popDensity}, PD),\ A := P/PD, PD \neq 0.$

$(S, \text{population}, P) \leftarrow (S, \text{area}, A), (S, \text{popDensity}, PD),\ P := A * PD.$

dbr:Mannheim dbo:population **2**.
dbr:Mannheim dbo:area **3**.

dbr:Mannheim dbo:popDensity 0.66666666.

dbr:Mannheim dbo:area 3.00000000003.

dbr:Mannheim dbo:population 1.99999998002.

… potentially infinite values by rounding errors.

Similarly, for ambiguous values (assume 2 population values for Mannheim)

Can materialization and/or query rewriting be used?

- Rewriting? Again consider clausal form of all variants of equations:

$$(S, \text{popDensity}, PD) \leftarrow (S, \text{population}, P), (S, \text{area}, A), \; PD := P/A$$
$$(S, \text{area}, PD) \leftarrow (S, \text{population}, P), (S, \text{popDensity}, PD), \; A := P/PD$$
$$(S, \text{population}, P) \leftarrow (S, \text{area}, A), (S, \text{popDensity}, PD), \; P := A * PD$$

dbr:Mannheim dbo:population 311142 .
dbr:Mannheim dbo:area 144.96 .

Finally, the resulting UCQs with assignments can be rewritten back to SPARQL using BIND

```
SELECT ?PD WHERE { :Mannheim dbo:popDensity ?PD}
```

$$q(PD) \leftarrow (S, \text{popDensity}, PD)$$
$$q(PD) \leftarrow (S, \text{population}, P), (S, \text{area}, A), PD := P/A$$
$$q(PD) \leftarrow (S, \text{popDensity}, PD'), (S, \text{area}, A'), (S, \text{area}, A), PD := P/A, P := PD' * A'$$

.. infinite expansion even if only 1 equation is considered.

Solution: "blocking" recursive expansion of the same equation for the same value.

```
SELECT ?PD WHERE { {:Mannheim dbo:popDensity ?PD }
                   UNION
                   { :Mannheim dbo:population ?P ; dbo:area ?A .
                     BIND (?P/?A AS ?PD )}
                 }
```

Algorithm:

- “Down-stripped-to-RDFS” version of PerfectRef [Calvanese, 2007] which handles equations by keeping “adornments” of attributes during rewriting:

Algorithm 1: Rewriting algorithm $\mathsf{PerfectRef_E}$

Input: Conjunctive query q, TBox $\mathcal{T}$
Output: Union (set) of conjunctive queries

$P := \{q\}$
repeat
    $P' := P$
    foreach $q \in P'$ **do**
        foreach *g in q* **do** // expansion
            foreach *inclusion axiom I in $\mathcal{T}$* **do**
                if *I is applicable to g* **then**
                    $P := P \cup \{q[g/\,\mathrm{gr}(g, I)]\}$
            foreach *equation axiom E in $\mathcal{T}$* **do**
                if *$g = U^{\mathrm{adn}(g)}(x, y)$ is an (adorned) attribute atom and* $\mathrm{vars}(E) \cap \mathrm{adn}(g) = \emptyset$ **then**
                    $P := P \cup \{q[g/\,\mathrm{expand}(g, E)]\}$
until $P' = P$
return P

Another example:

- "Companies and their revenues in EUR" (from above)

- Original query:

```
SELECT * WHERE
{ ?C rdf:type NYT:Org .
  ?C dbpedia:revenueEUR ?R . }
```

- Equations:

*dbo:revenueUSD = dbo:revenueEUR *1.3.*
dbo:profitEUR = "dbo:revenueEUR – dbo:totalExpensesEUR".

- Rewritten query:

```
SELECT * WHERE
{ ?C rdf:type NYT:Org .
  { {?C dbpedia:revenueEUR ?R .}
    UNION {?C dbo:revenueUSD ?RU . BIND (?RU / 1.3 AS ?R ) }
    UNION {?C dbo:totalExpensesEUR ?E ; dbo:profitEUR ?P. BIND (?E + ?P AS ?R ) }} }
```

Can materialization and/or query rewriting be used?

- Back to rule-based Materialization:

$$(S, \text{popDensity}, PD) \leftarrow \ (S, \text{population}, P), (S, \text{area}, A),\ PD := P/A,\ A \neq 0.$$
$$(S, \text{area}, PD) \ \leftarrow \ (S, \text{population}, P), (S, \text{popDensity}, PD),\ A := P/PD, PD \neq 0.$$
$$(S, \text{population}, P) \ \leftarrow \ (S, \text{area}, A), (S, \text{popDensity}, PD),\ P := A * PD.$$

dbr:Mannheim dbo:population 2.
dbr:Mannheim dbo:area 3.

Similar blocking possible in some rule systems, e.g. Jena Rules:

```
[ (?C :area ?A) (?C :population ?P)
  notEqual(?A, 0) quotient(?P, ?A, ?PD)
  noValue(?C, :populationDensity)  -> (?C :populationDensity ?D)]

[ (?C :populationDensity ?PD) (?city :population ?P)
  notEqual(?PD, 0) quotient(?P, ?PD, ?A)
  noValue(?C, :area)  -> (?city :area ?A)]

[ (?C :area ?A) (?C :populationDensity ?P)  product(?A, ?PD, ?P)
  noValue(?city, :population)  -> (?city :population ?P)]
```

Side remark: Experiments in our ESWC2013 paper favor rewriting approach.

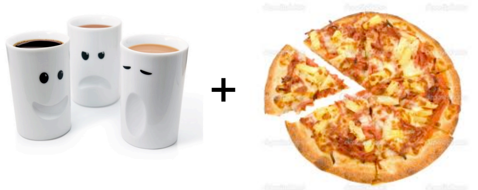
Extending RDFS by attribute equations:

- Only a first step…
- … but seems useful for LOD use cases, particularly as more and more numerical open data gets published (statistical reports, EU, national Open Government initiatives)
- in general infinite results (both for rule-based as well as for rewriting)…
- … but “blocking” helps

Lecture Roadmap

- Scope/Motivation
  (Axel)
- Short Introduction to RDFS+OWL
  (Aidan)
- RDFS+OWL usage in Linked Data
  (Aidan)
- High-level Reasoning approaches: Query rewriting vs. Materialization
  (Axel)
- Challenges on Reasoning over Linked Data
  (Axel)
- Practical approaches for Reasoning over Linked Data
  - Quarantined & Authoritative Materialization *(Aidan)*
  - Link-Traversal Based Query Execution with Reasoning *(Aidan)*
  - Reasoning with Property Equations by Rewriting *(Axel)*

- **Wrap-up/Outlook** ***(all)***



SIEMENS DERI

Wrap-up/Take-home messages:

- RDFS/OWL Reasoning over (open-domain) Linked Data is **difficult**
  - It's the Web: lots of data
  - It's the Web: lots of mess

- … but **not impossible**!
- Requires some compromises:
  - You don't want complete reasoning over (unbounded) Linked Data
  - What should you (automatically) trust?
    - Consider source of data: examine dereferencing
    - Use (Webby) heuristics: examine the link structure

- RDFS/OWL is **useful** for Linked Data! *But perhaps not ideal/enough*:
  - Complexity of OWL standard scares off publishers
  - Numeric/datatype/unit reasoning required
  - Generic rules (RIF/SPIN/N3/Jena) needed

Outlook - Some entry points for further research:

- Combinations of the challenges we outlined raise new issues when addressed at once?
  - E.g. Dynamicity of terminological knowledge + attribute equations e.g. currency conversion
- More usage of inductive methods/statistical reasoning? How can they be combined with deductive methods?
- More beyond OWL? i.e. extensions of attribute equations, other means to describe "mappings" to link vocabularies (SPARQL CONSTRUCT, R2R, SPIN rules, etc.) … and how to use that together with OWL
- Sweet-spot between query rewriting and materialization…
- Efficient/parallelizable techniques from the DB & Datalog communities, e.g.

  Foto N. Afrati, Jeffrey D. Ullman: Transitive closure and recursive Datalog implemented on clusters. EDBT 2012: 132-143
- Techniques for dealing with messy data

On the Exploration of the Query Rewriting Space with Existential Rules

Mélanie König, Michel Leclère, Marie-Laure Mugnier, and Michaël Thomazo

University Montpellier 2, France

Abstract. We address the issue of Ontology-Based Data Access, with ontologies represented in the framework of existential rules, also known as Datalog+/-. A well-known approach involves rewriting the query using ontological knowledge. We focus here on the basic rewriting technique which consists of rewriting a conjunctive query (CQ) into a union of CQs. We assume that the set of rules is a finite unification set, i.e., for any CQ, there exists a finite sound and complete …

but they lead to different explorations of the rewriting space. Finally, an experimental comparison of these operators within an implementation of the generic

Mélanie König, Michel Leclère, Marie-Laure Mugnier, Michaël Thomazo. On the Exploration of the Query Rewriting Space with Existential Rules. RR2013