# Ontologies and the Semantic Web: an introduction

Michele Pasin
Kings College, London
October 2010

More info available at http://staff.cch.kcl.ac.uk/~mpasin/

# Outline

- what are ontologies?
  - *A theoretical perspective: ontology as a knowledge representation*

- how do we build them?
  - *A design perspective: key notions, principles, potential pitfalls*

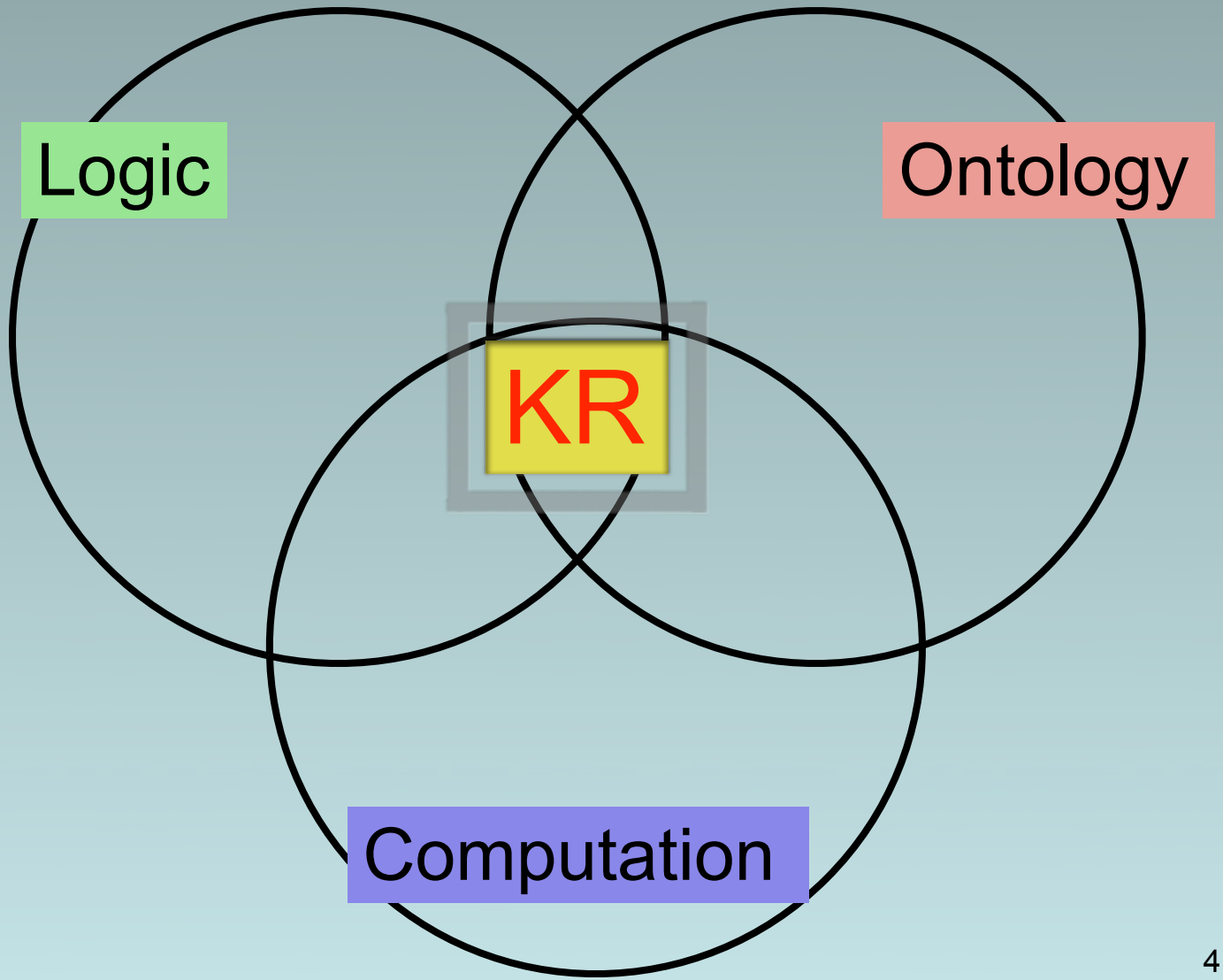
- how can we use them?
  - *A pragmatic perspective: data integration and the semantic web*

# What is an ontology? A plethora of definitions..

1. *Ontology as a philosophical discipline*
2. *Ontology as a an informal conceptual system*
3. *Ontology as a formal semantic account*
4. *Ontology as a specification of a conceptualization*
5. *Ontology as a representation of a conceptual system via a logical theory*
   *5.1 characterized by specific formal properties*
   *5.2 characterized only by its specific purposes*
6. *Ontology as the vocabulary used by a logical theory*
7. *Ontology as a (meta-level) specification of a logical theory"* (from Guarino & Giaretta, 1995).

Doug. Ontologies: State of the Art, Business Potential, and Grand Challenges. Ontology Management: Semantic Web, Semantic Web Services, and Business Applications (2007) pp. 1-20

# John Sowa: three components to a knowledge representation

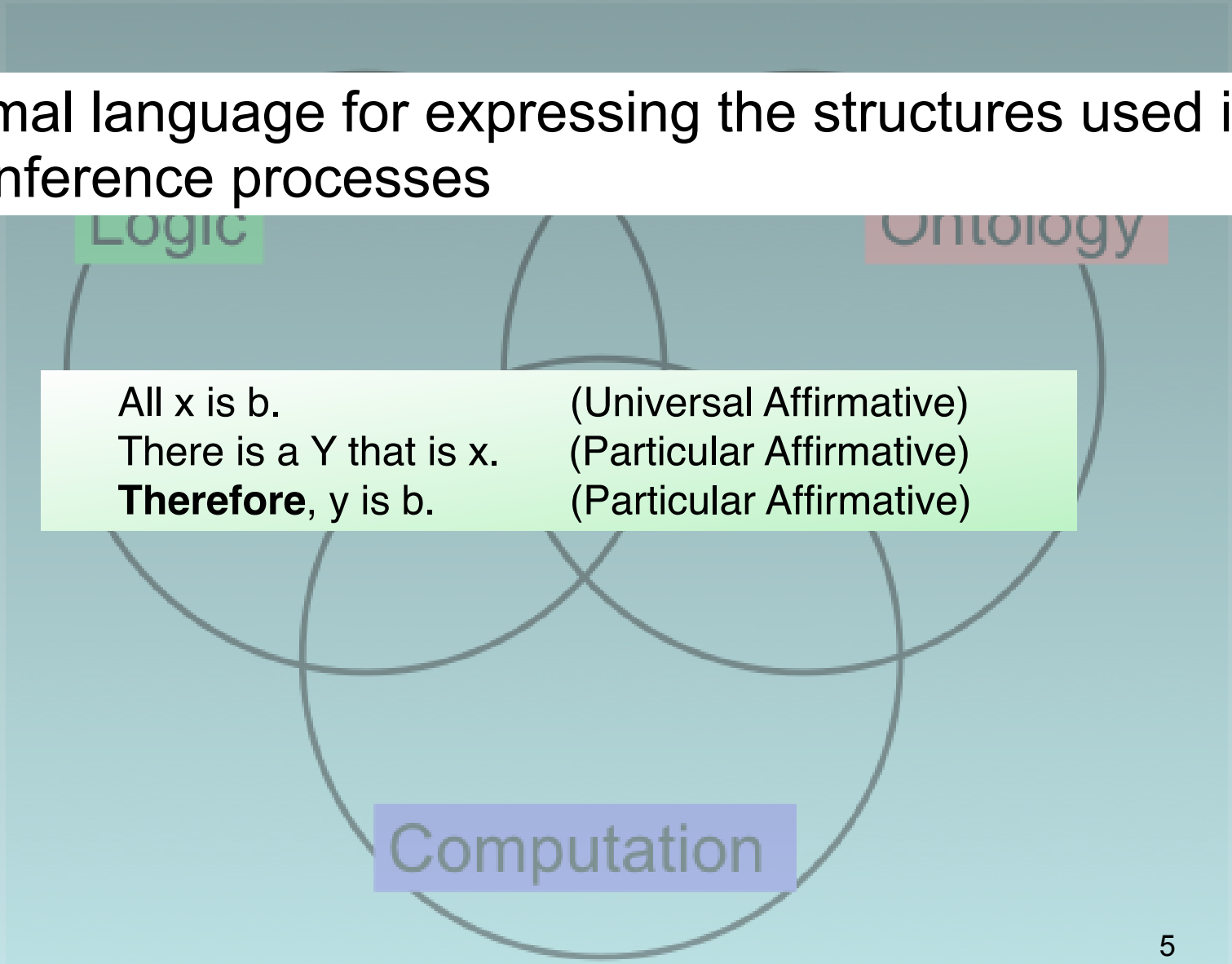





Sowa. Knowledge Representation: Logical, Philosophical and Computational Foundations. Course Technology (1999)

# (I) Logic

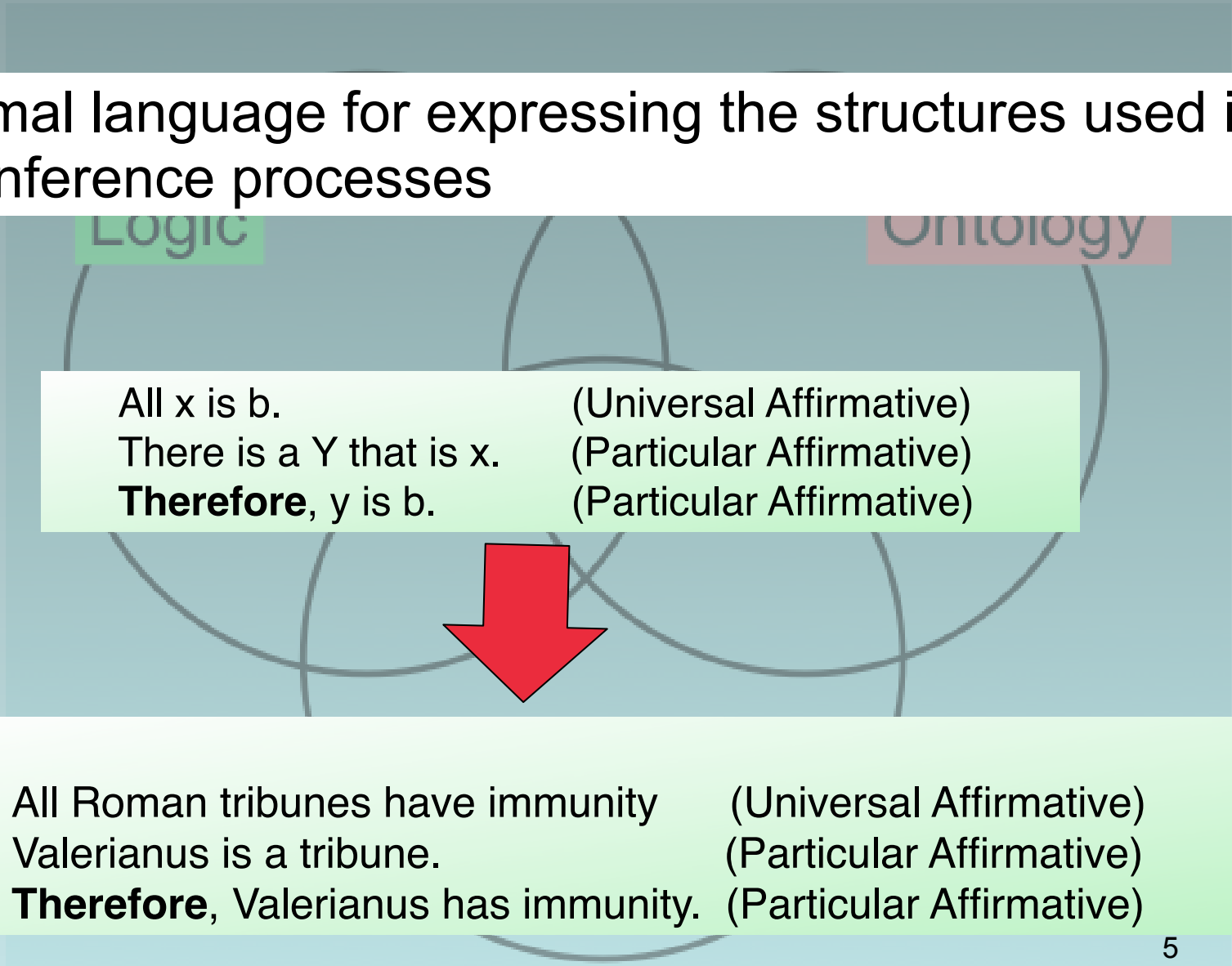
- formal language for expressing the structures used in our inference processes




| | |
|---|---|
| All x is b. | (Universal Affirmative) |
| There is a Y that is x. | (Particular Affirmative) |
| **Therefore**, y is b. | (Particular Affirmative) |

# (I) Logic

- formal language for expressing the structures used in our inference processes

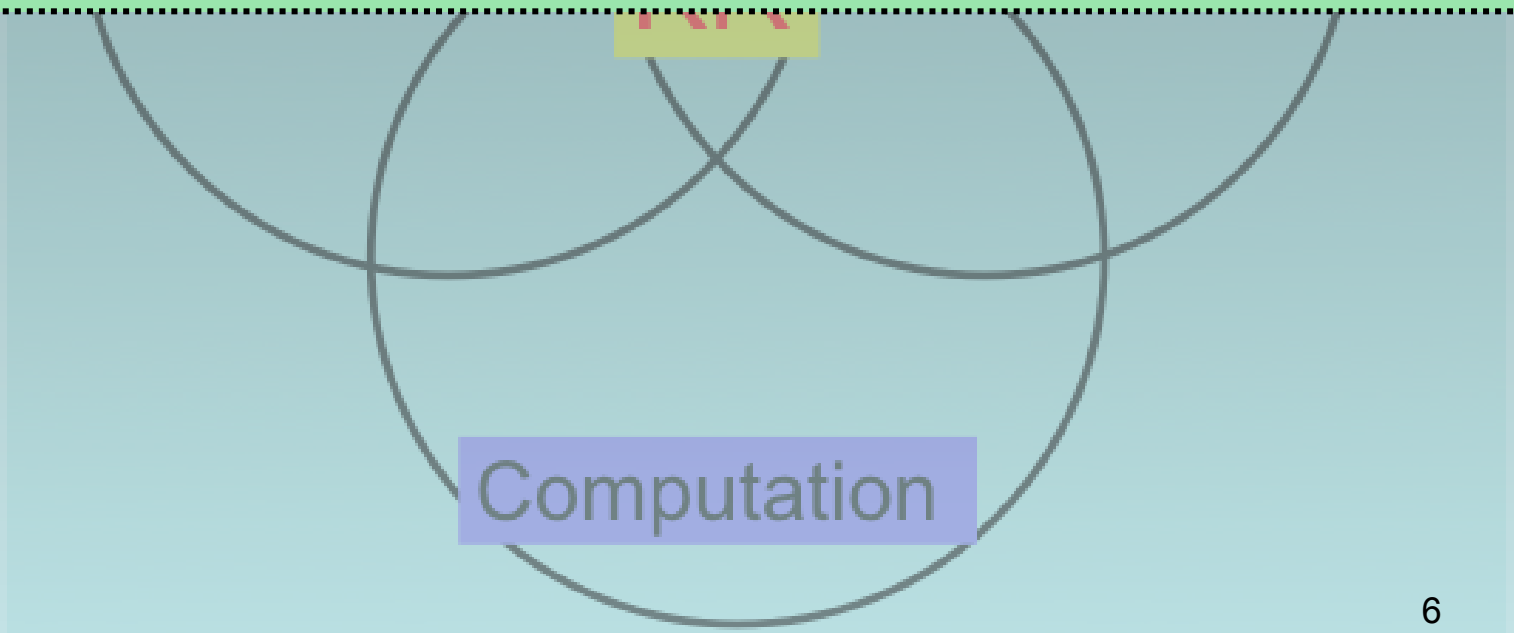



All x is b. (Universal Affirmative)
There is a Y that is x. (Particular Affirmative)
**Therefore**, y is b. (Particular Affirmative)

All Roman tribunes have immunity (Universal Affirmative)
Valerianus is a tribune. (Particular Affirmative)
**Therefore**, Valerianus has immunity. (Particular Affirmative)

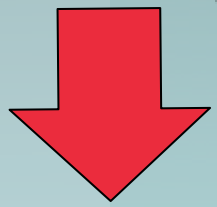
# (II) Ontology

**Tribune** (from the Latin: tribunus; Byzantine Greek form τριβούνος) was a title shared by **10 elected officials** in the **Roman Republic**. Tribunes had the power to convene the **Plebeian Council** and to act as its **president**, which also gave them the **right to** propose **legislation** before it. They were sacrosanct, in the sense that any **assault** on their person **was prohibited**. They had the power to **veto** actions taken by **magistrates**, and specifically to **intervene legally** on behalf of **plebeians**. The tribune could also summon the **Senate** and **lay proposals** before it. [....]



# (II) Ontology

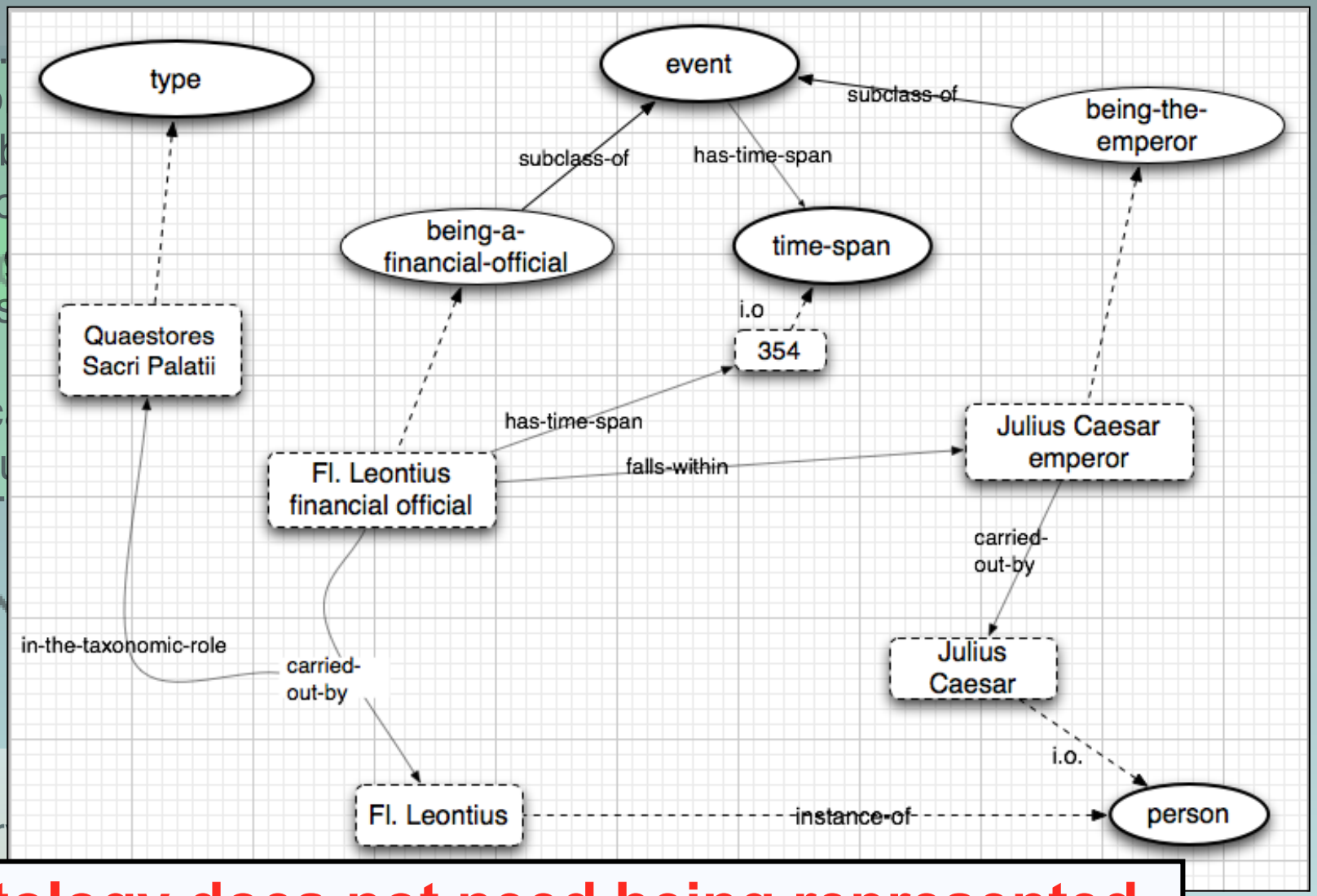
**Tribune** (from the Latin: tribunus; Byzantine Greek form τριβούνος) was a title shared by **10 elected officials** in the **Roman Republic**. Tribunes had the power to convene the **Plebeian Council** and to act as its **president**, which also gave them the **right to** propose **legislation** before it. They were sacrosanct, in the sense that any **assault** on their person **was prohibited**. They had the power to **veto** actions taken by **magistrates**, and specifically to **intervene legally** on behalf of **plebeians**. The tribune could also summon the **Senate** and **lay proposals** before it. [....]



For every x, if (x **isTribune**) ==> exists y such that (y **isCity**) and (y **hasName** Rome) and (**lives_in** x, y)

# (II) Ontology

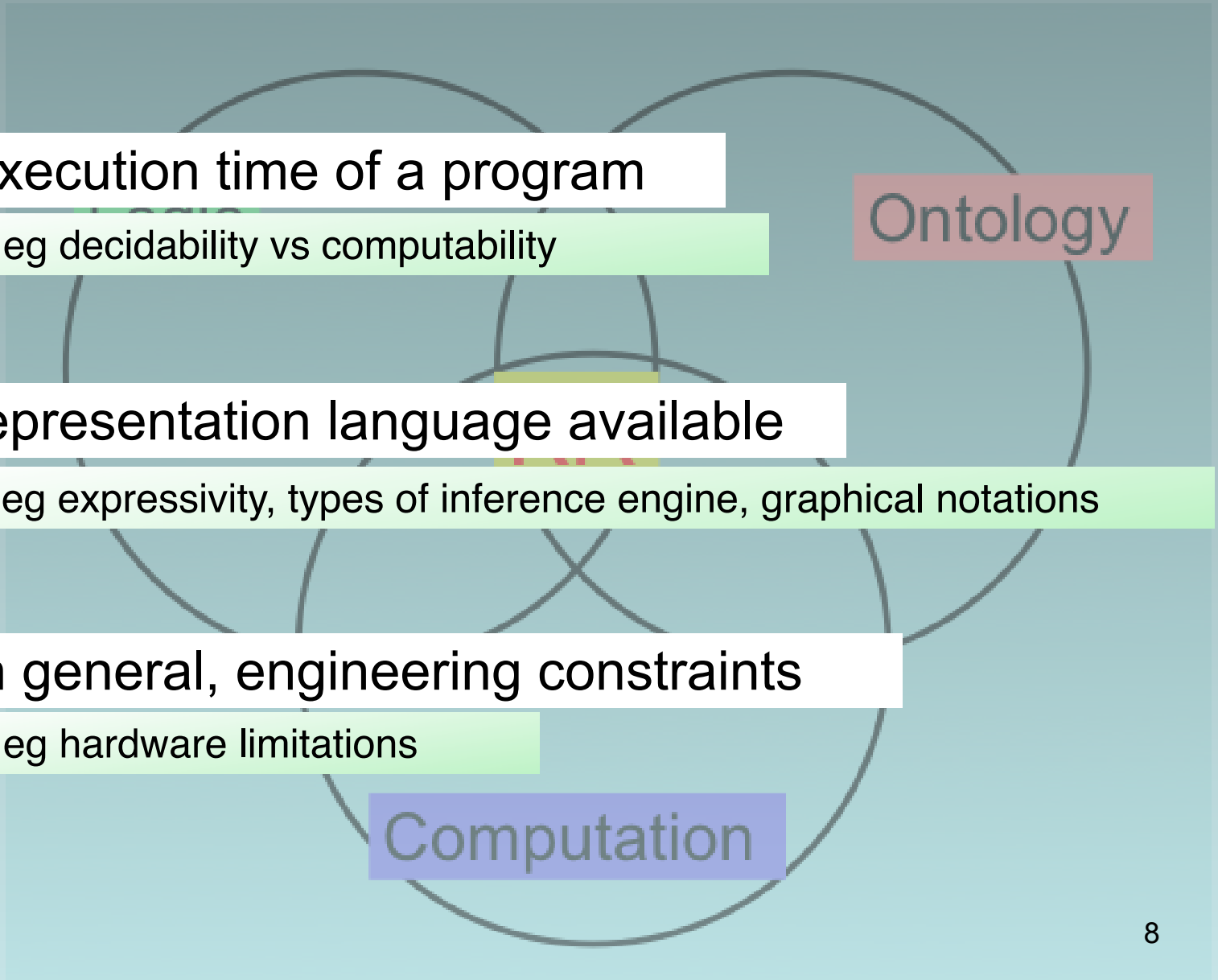
**Tribune** (fro
title shared b
the power to
which also
were sacros
**prohibited**.
and specific
could also su




For ever

**- an ontology does not need being represented through the formal language of logic!**

# (III) Computation




- execution time of a program

  eg decidability vs computability

- representation language available

  eg expressivity, types of inference engine, graphical notations

- in general, engineering constraints

  eg hardware limitations

## John Sowa:

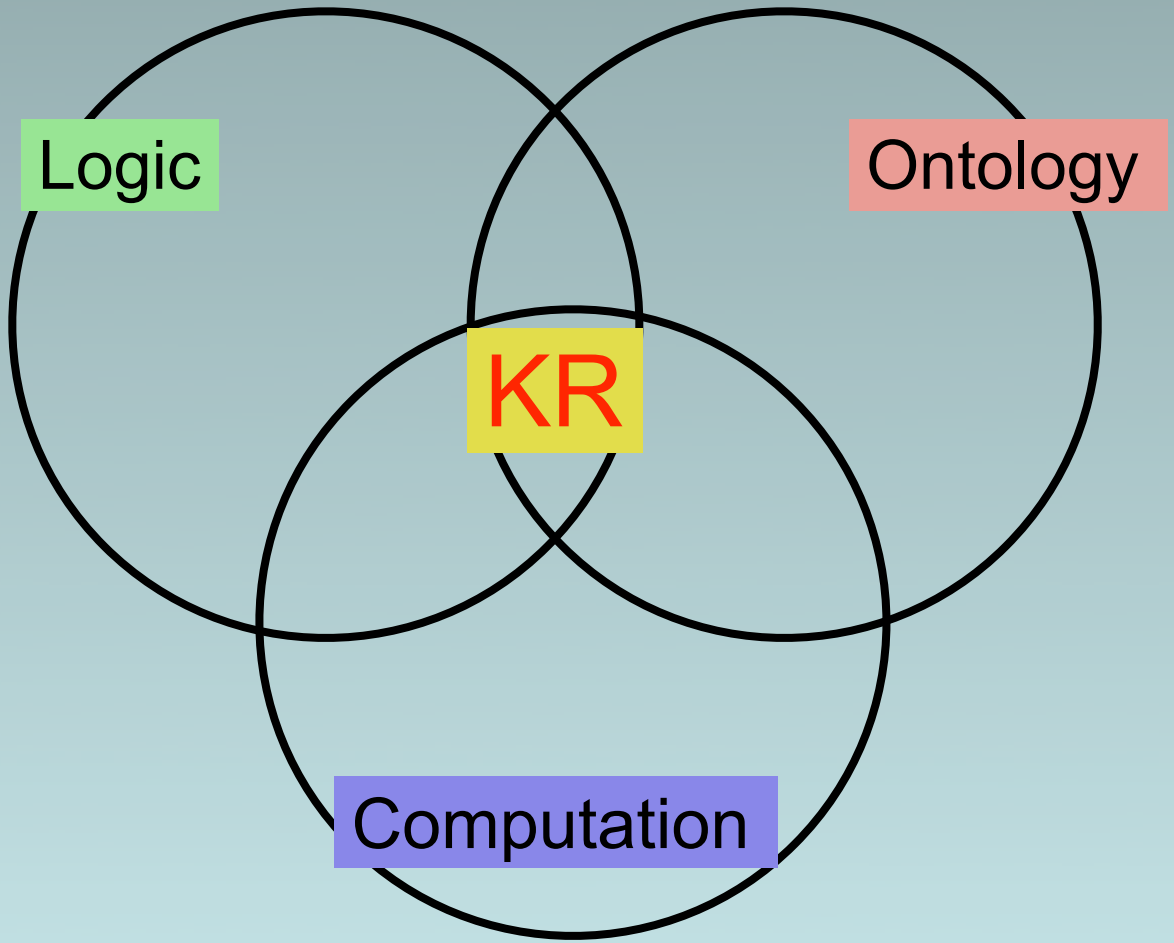
"**Without logic**, a knowledge representation is vague, with no criteria for determining whether statements are redundant or contradictory.

**Without ontology**, the terms and symbols are ill-defined, confused and confusing.

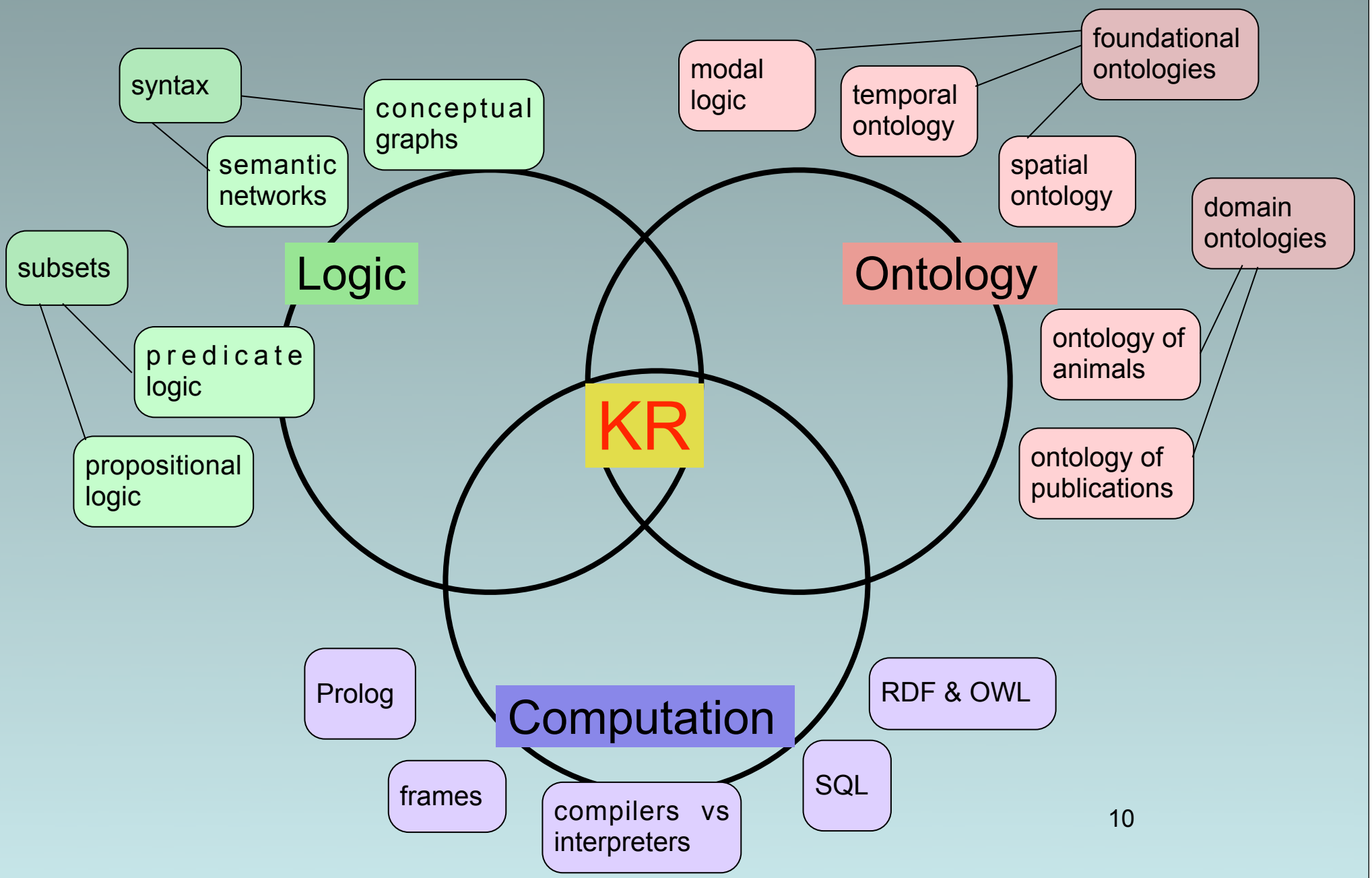
And **without computable models**, the logic and ontology cannot be implemented in computer programs.

**Knowledge representation** is the application of logic and ontology to the task of constructing computable models for some domain."

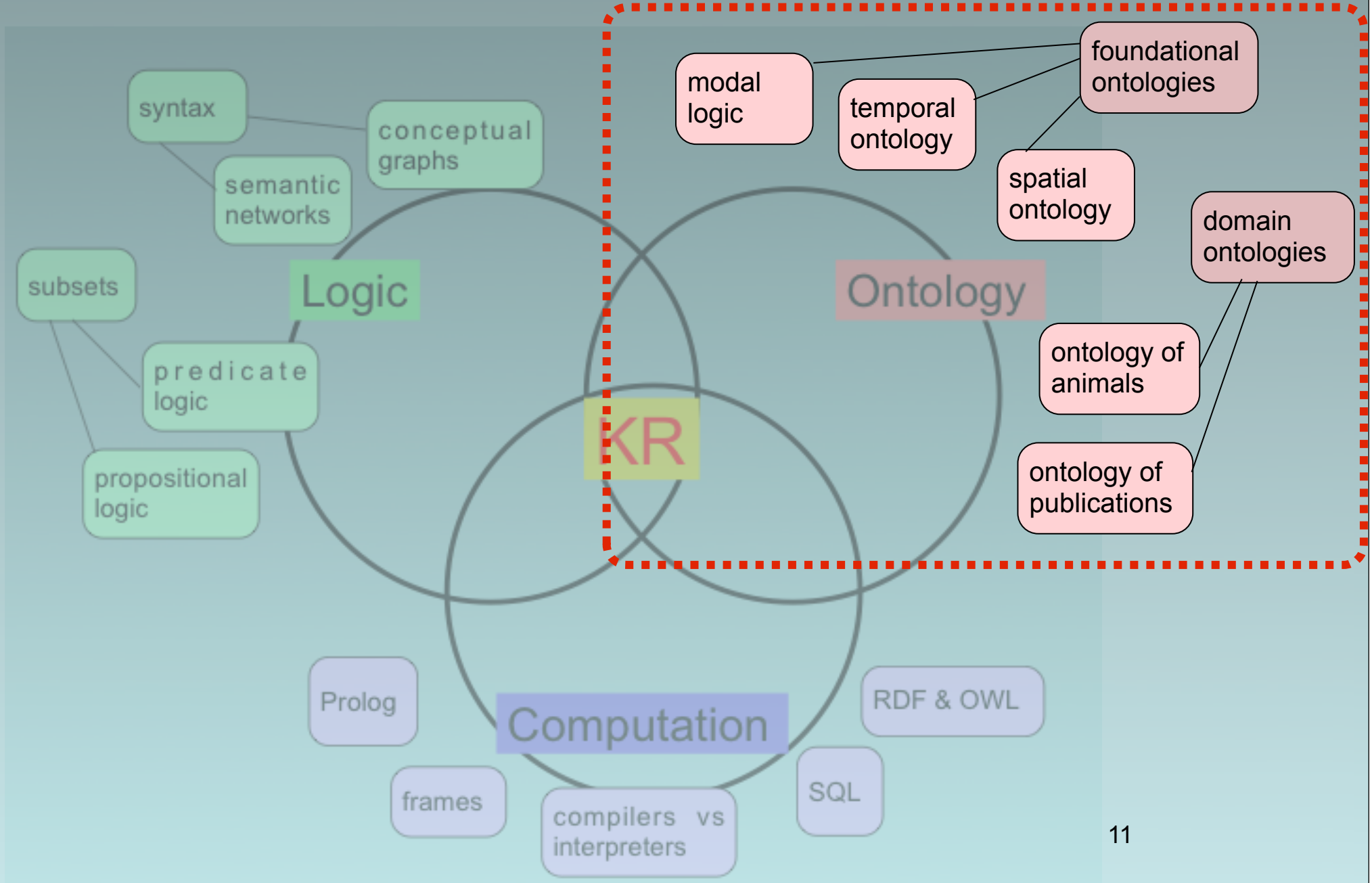
# Possible research directions:



# Possible research directions:

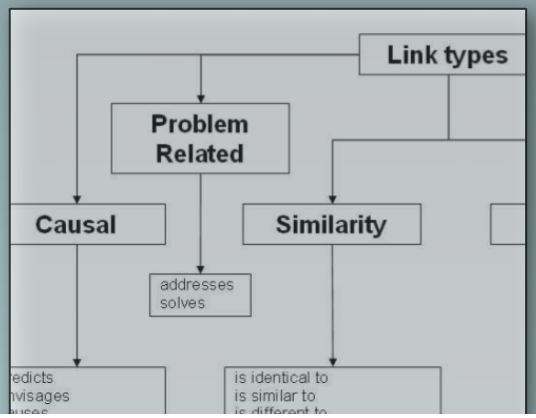


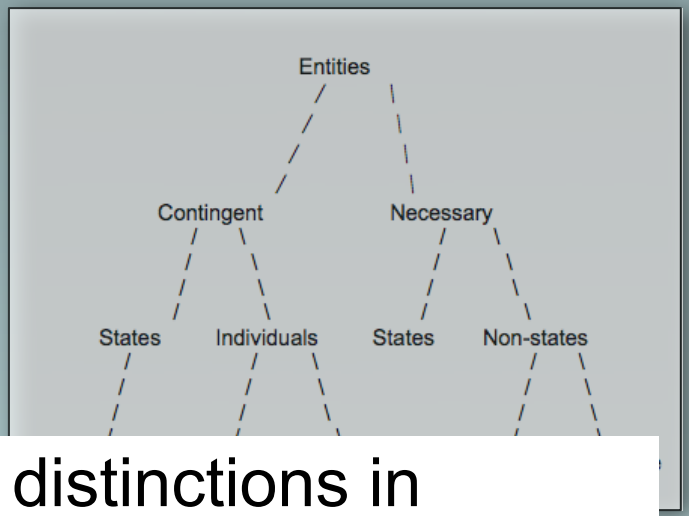
# Possible research directions:



# ontology building 101

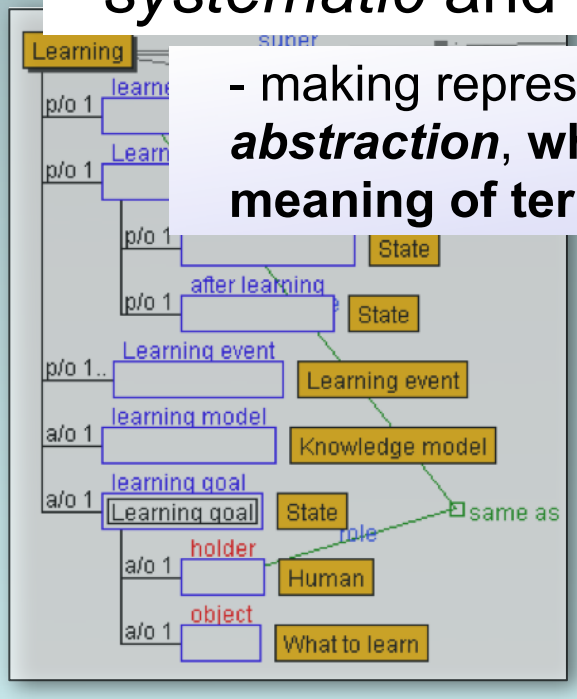
# Reusing philosophical methods&notions

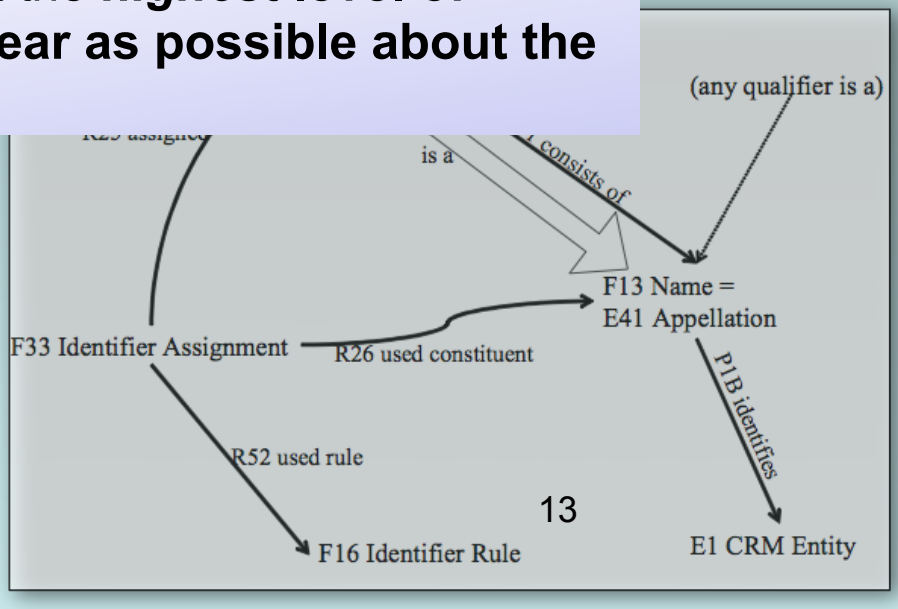



- a *theory* of how to make ontological distinctions in *systematic* and *coherent* manner

- making representational choices at the ***highest*** **level of *abstraction*, while still being as clear as possible about the meaning of terms**



# Universals vs Instances

The central distinction

***universal* vs. *instance***

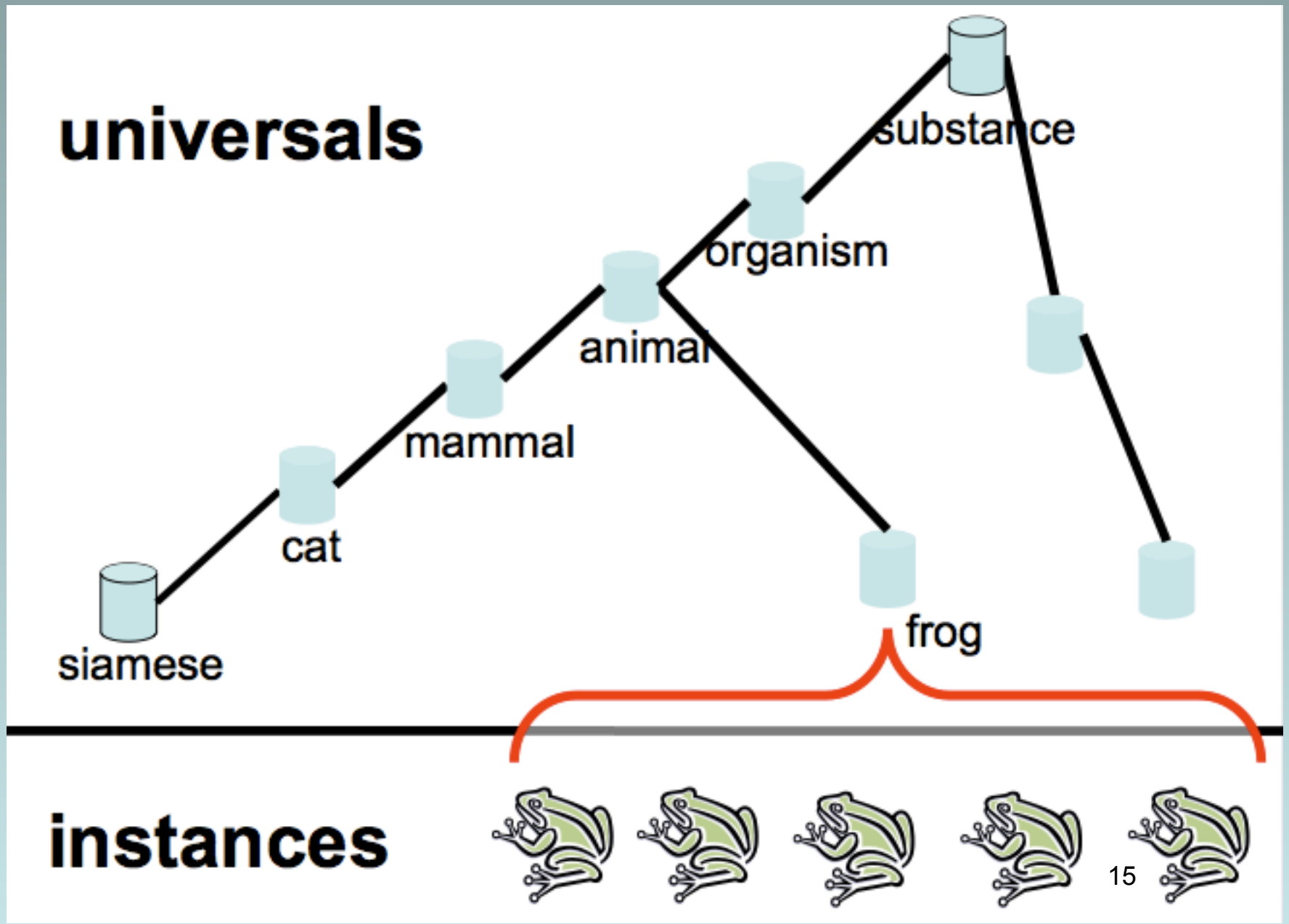
(*catalog* vs. inventory)

(*science text* vs. diary)

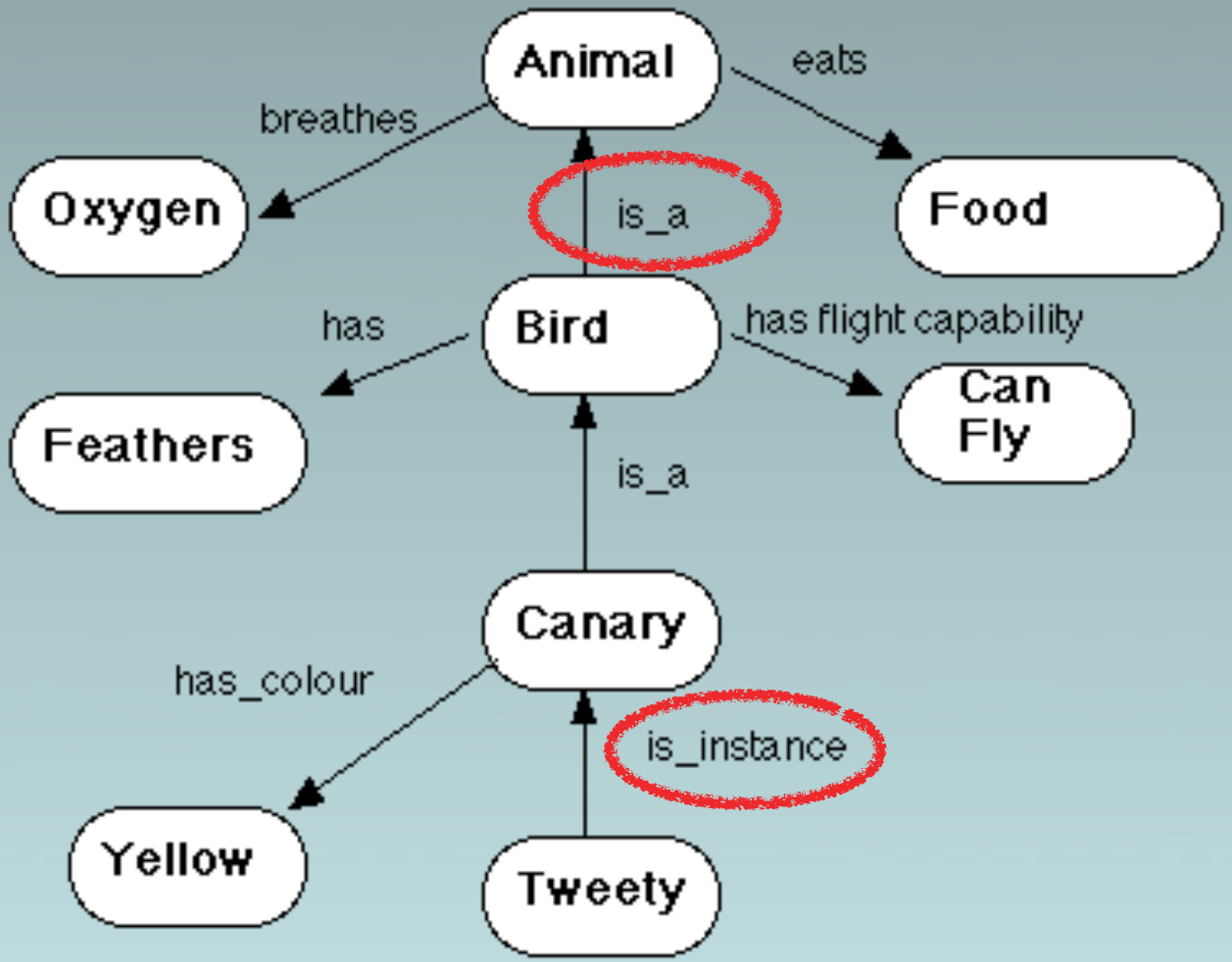
(*human being* vs. Arnold Schwarzenegger)

Adapted from Smith. Introduction to Basic Formal Ontology (video). A two-day tutorial given in conjunction with the International Conference on Biomedical Ontology, Buffalo, NY (2009)
http://www.bioontology.org/wiki/index.php/From_BFO_to_IAO

# Universals vs Instances



# Ontological relations: instance of, subclass of



# A few ontology design principles:

- **concepts** rather than **terms**

  - people are easily trapped by the endless terminological discussion departing from the underlying conceptual structure of the target domain

- **role** concepts vs **basic** concepts

  - Clear and consistent differentiation between basic concepts (man, rice, oil, etc.) and role concepts(teacher, food, fuel, etc.).

- determine an **essential** property for each concept and instance

  - Proper use of is-a relation should inherit the “Essential” property of its super classes (= identity criteria checking)



cf. Guarino and Welty. Evaluating ontological decisions with OntoClean. Commun. ACM (2002) vol. 45 (2) pp. 61-65

# A few ontology design principles:

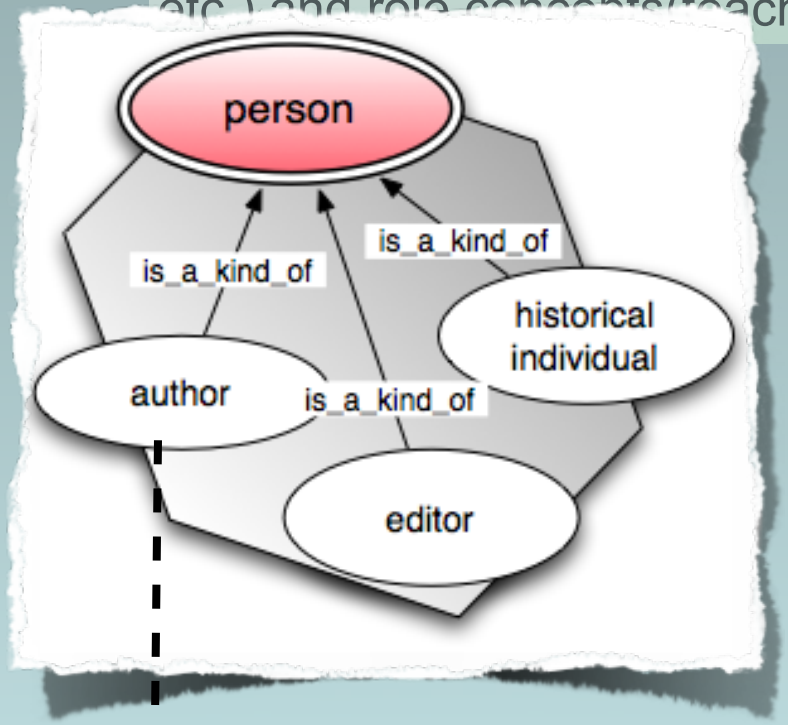
- **role** concepts vs **basic** concepts

  - Clear and consistent differentiation between basic concepts (man, rice, oil, etc.) and role concepts(teacher, food, fuel, etc.).

# A few ontology design principles:

- **role** concepts vs **basic** concepts

  - Clear and consistent differentiation between basic concepts (man, rice, oil, etc.) and role concepts(teacher, food, fuel, etc.).

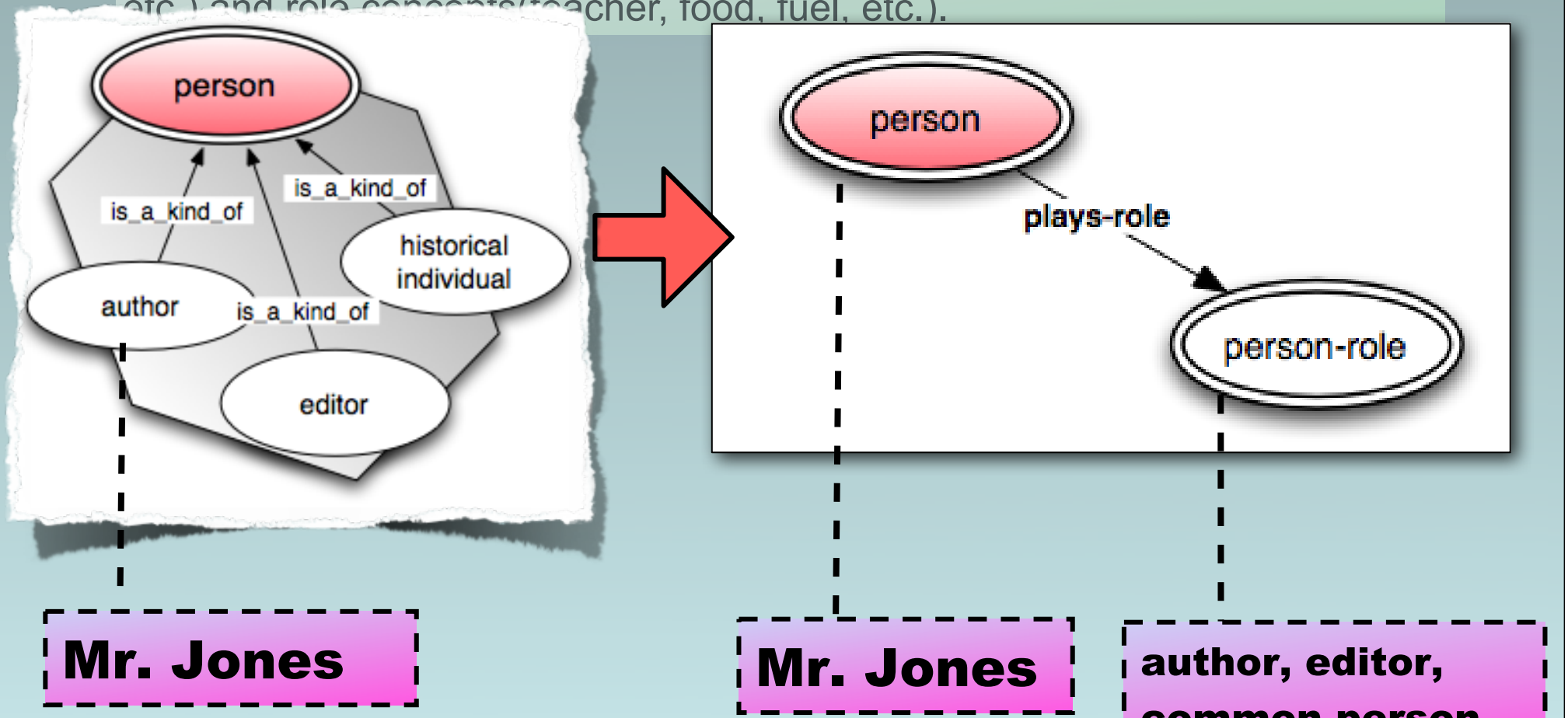



**Mr. Jones**

# A few ontology design principles:

- **role** concepts vs **basic** concepts

  - Clear and consistent differentiation between basic concepts (man, rice, oil, etc.) and role concepts(teacher, food, fuel, etc.).



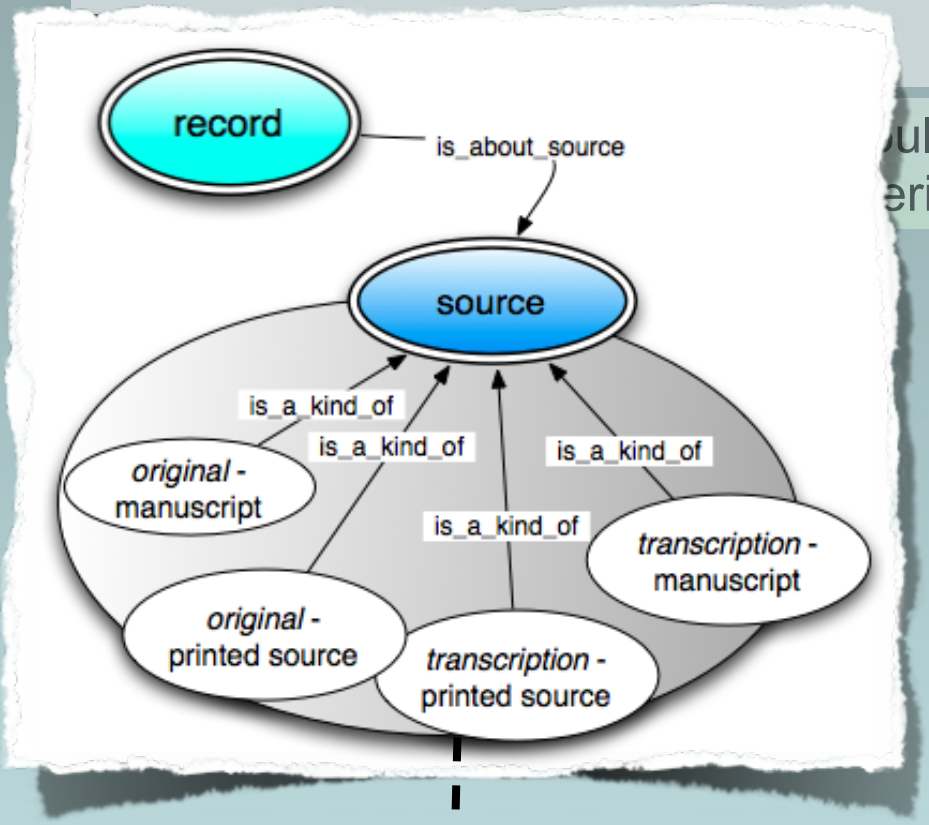
# A few ontology design principles:

- determine an **essential** property for each concept and instance

  - Proper use of is-a relation should inherit the “Essential” property of its super classes (= identity criteria checking)

# A few ontology design principles:

- determine an **essential** property for each concept and

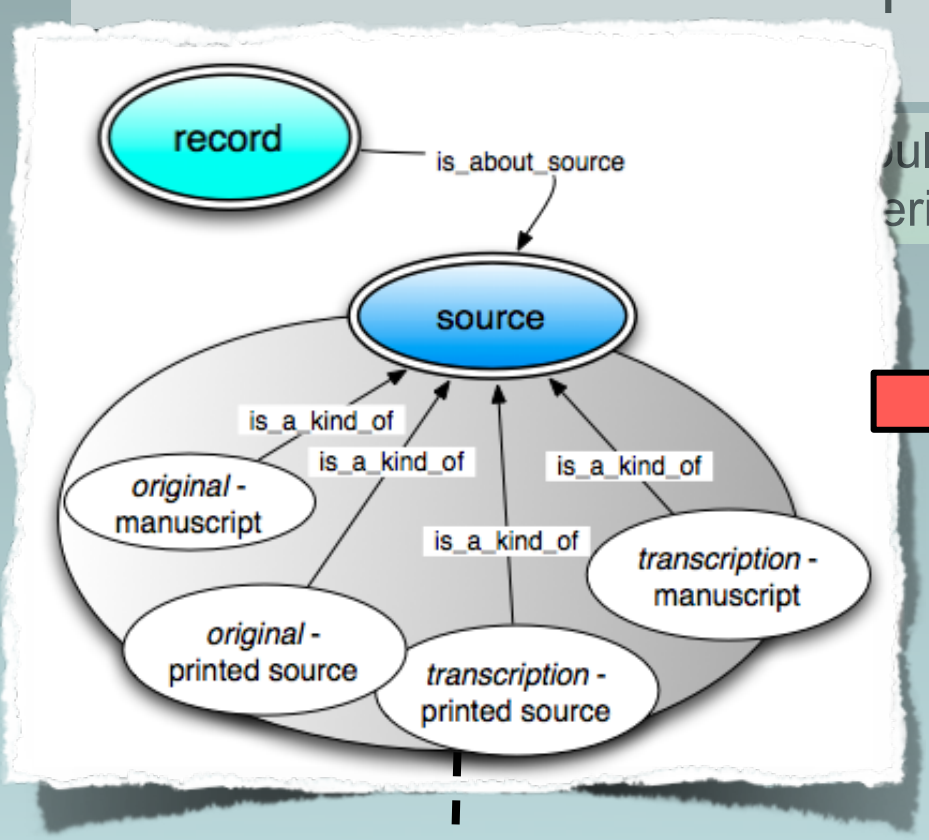


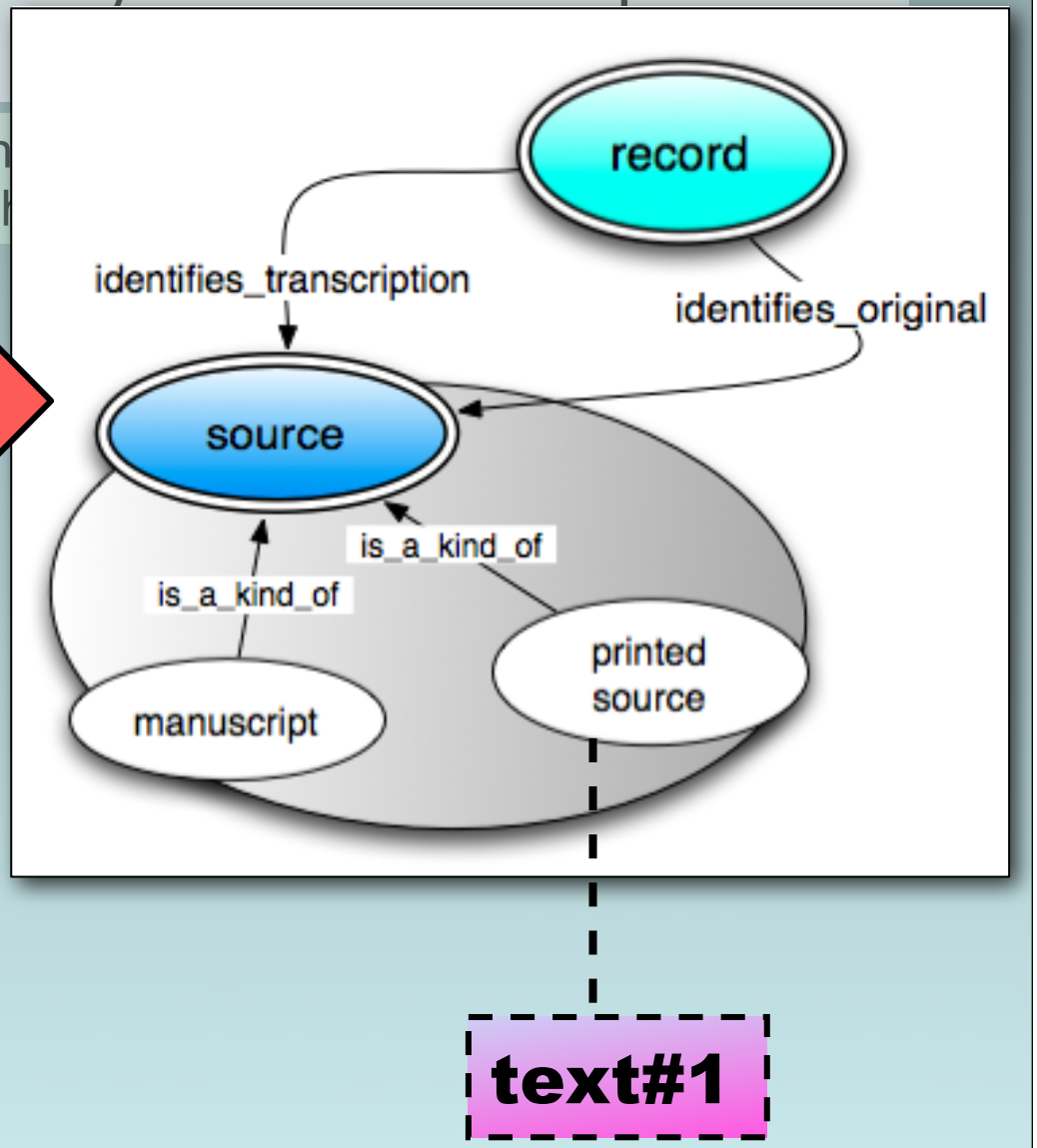

uld inherit the “Essential” property of
eria checking)

**text#1**

# A few ontology design principles:

- determine an **essential** property for each concept and



**potential pitfalls...**

# Pitfall [1]: Ontologies and data models

- main difference with data models is not the content, but the purpose

  - Clarity: context dependent vs context independent design

  - Extendibility: application oriented vs design for future reuse

  - Minimal Encoding Bias - avoid representational choice for benefit of implementation

- a conceptual model written in an ontology language is not necessarily an ontology!

  - you cannot see the difference by looking at the syntax

# Pitfall [2]: Ontologies and knowledge bases

- the same languages (OWL, RDF-S, WSML, etc.) and the same tools and infrastructure can be used both for creating ontologies and for creating knowledge bases

- not every OWL file is an ontology, since OWL files can also be used for representing a knowledge base (eg info about the concept of ‘city’, and the individual ‘Innsbruck’)

- Ontologies are the vocabulary and the formal specification of the vocabulary only, which can be used for expressing a knowledge base

- one initial motivation for ontologies was achieving interoperability between multiple knowledge bases!

# Pitfall [3]: ontologies and XML Schemas

- XML schemas define a single representation syntax for a particular problem domain but not the semantics of domain elements

e.g. sequence and hierarchical ordering of fields in a valid document instance, but do not specify the semantics of this ordering..

- They do not aim at carving out re-usable, context-independent categories of things

e.g. whether a data element “student” refers to the human being or the role of being as student.

- There is no standardized inference layer

To employ XML to generate new data, you need knowledge embedded in some procedural code somewhere, rather than explicitly stated, as in OWL.

# a closer look at existing ontologies

# Upper vs Domain ontologies

- depends on the type of ‘predicates’ our (logical) theory is investigating..

- domain **independent**: *part-whole, temporal relations, concrete-abstract, universal-particular, qualities*

- domain **dependent**: *car makers, car materials, fuel consumption, etc.*

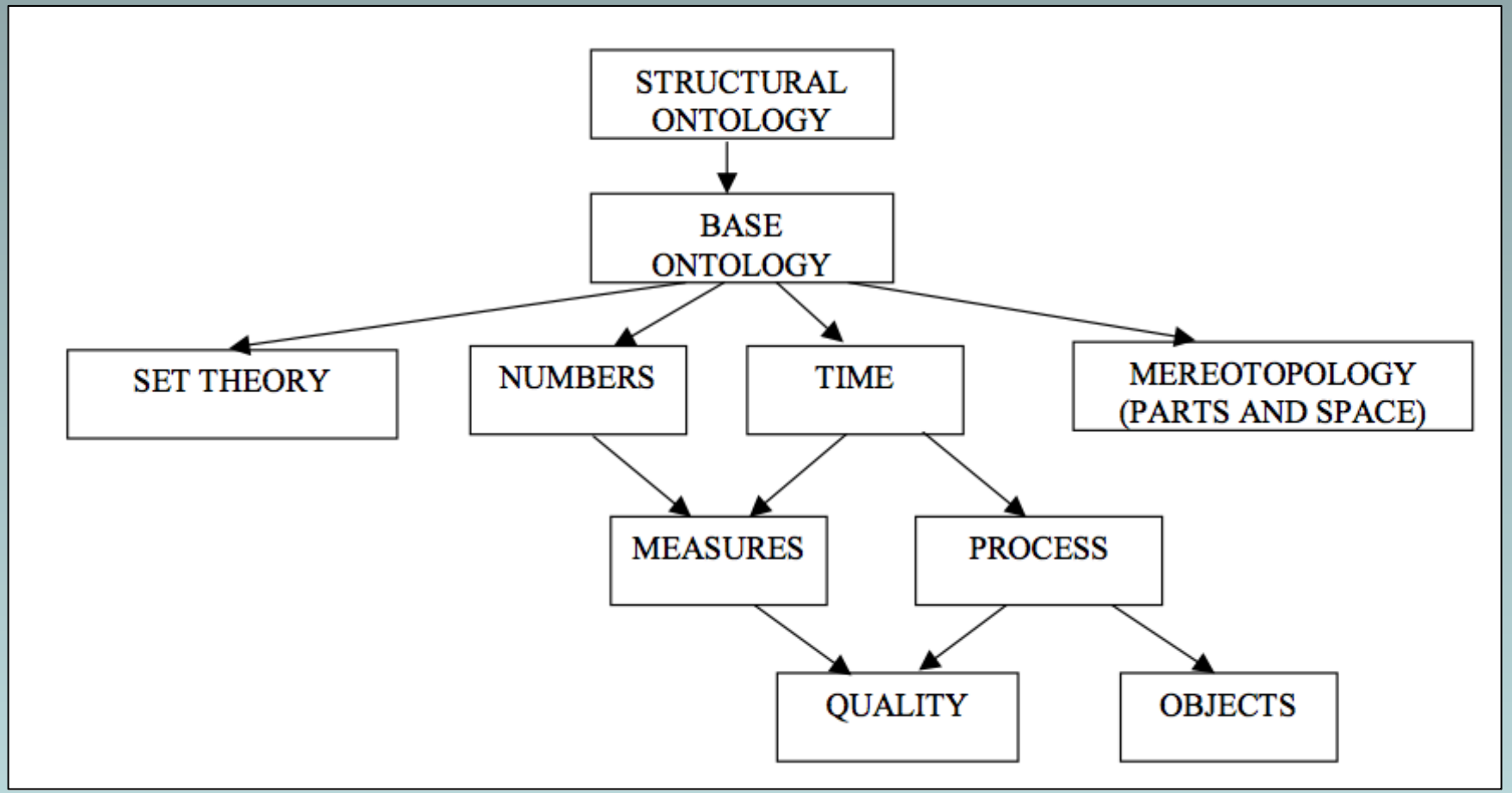
- task ontologies: a problem solving process like diagnosis, monitoring, scheduling, design, and so on

- in the Semantic Web, top level ontologies are supposed to bridge the various possible domain ones

- a top level ontology is very general and abstract

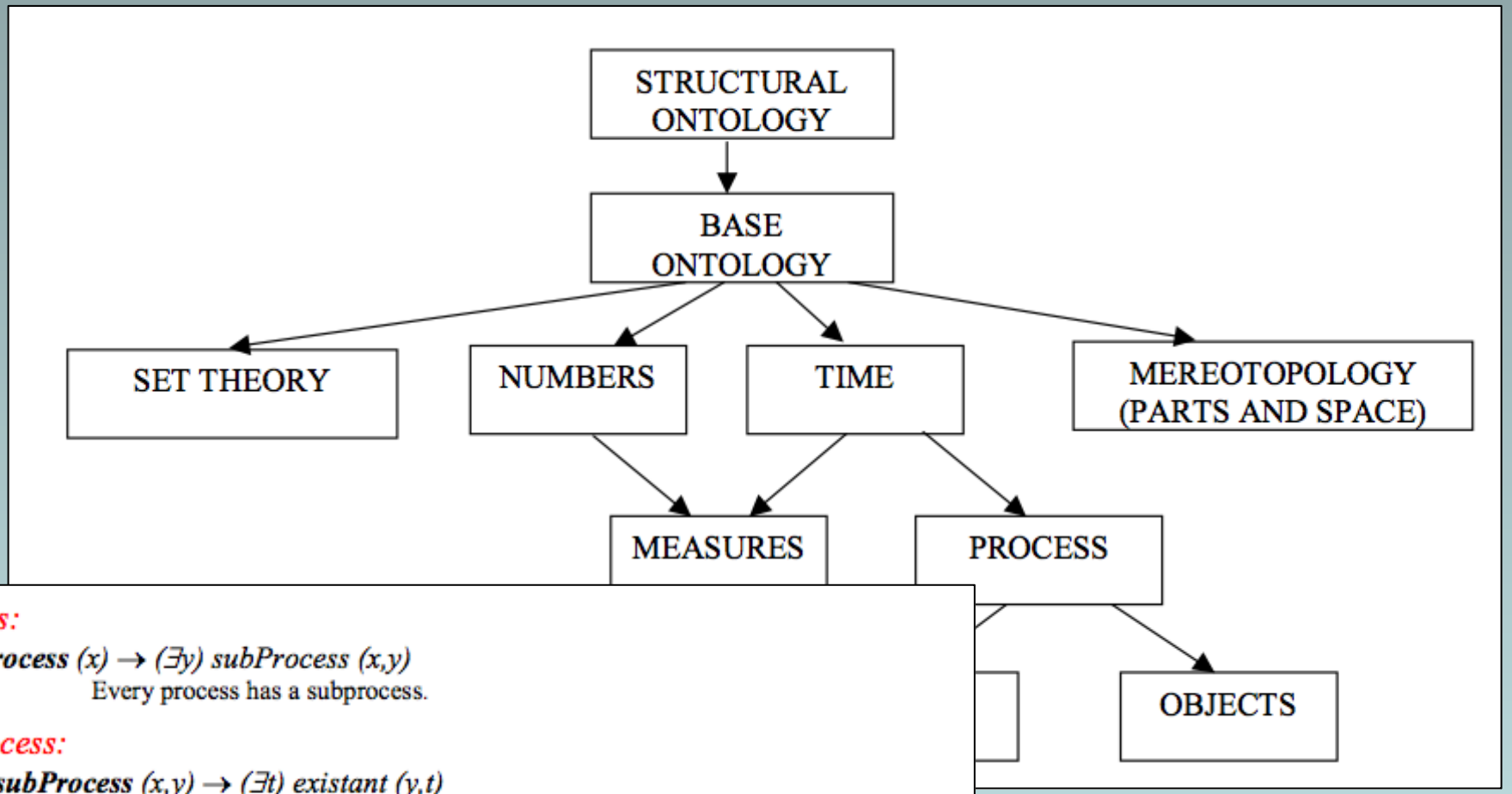
- e.g. DOLCE, SUMO, CYC, BFO

# Upper ontologies: ex. top level of SUMO




http://www.ontologyportal.org/

# Upper ontologies: ex. top level of SUMO




*Process:*

*(∀x)* ***Process*** *(x) → (∃y) subProcess (x,y)*

Every process has a subprocess.

*subProcess:*

*(∀x,y)* ***subProcess*** *(x,y) → (∃t) existant (y,t)*

Every subprocess exists in a temporal instant

*(∀x,y)* ***subProcess*** *(x,y) → [(∀ z) located (y,z) → located (x,z)]*

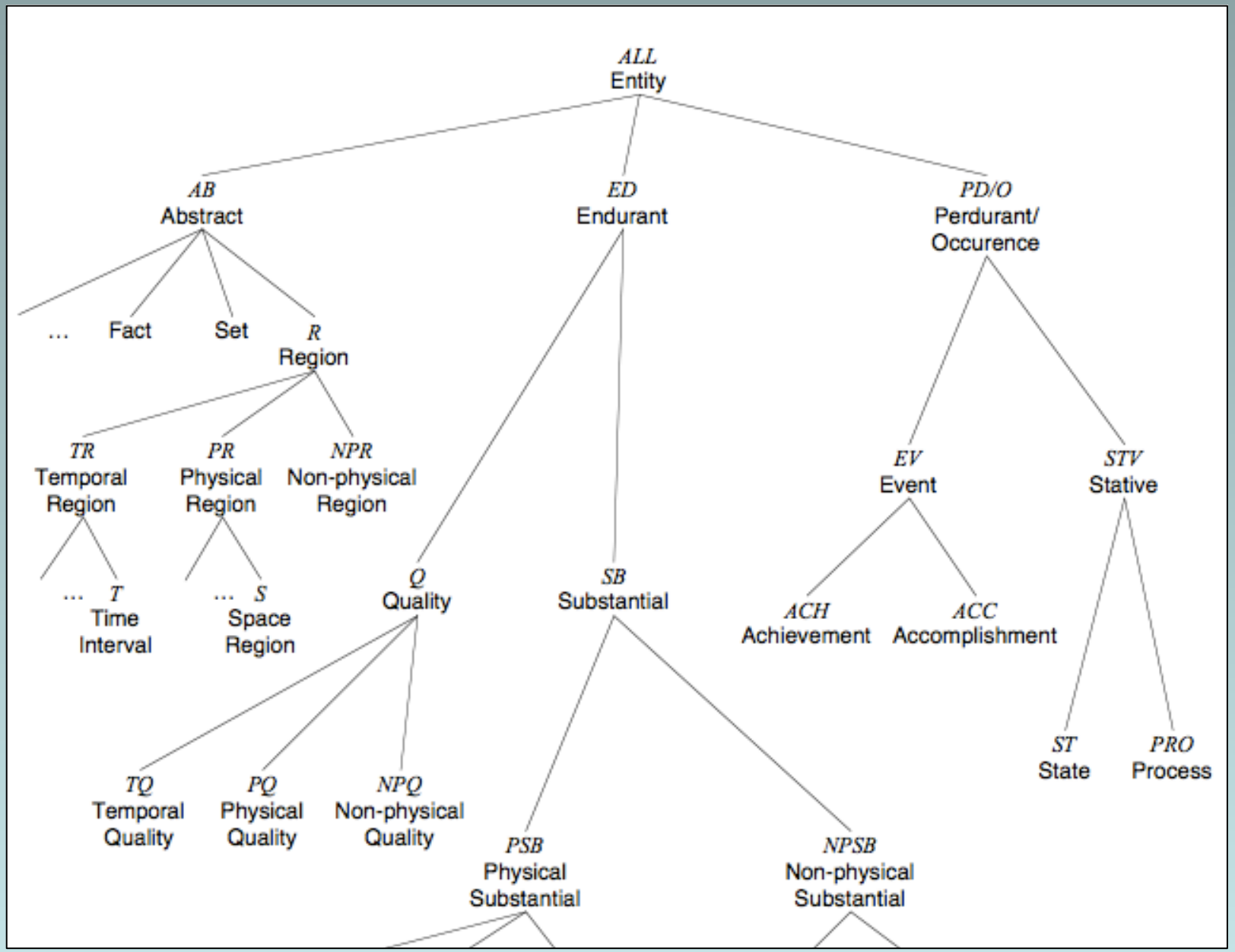
A subprocess is located as the process

*(∀x,y)* ***subProcess*** *(x,y) → WhenFn(x)= WhenFn(y) ∨ during (WhenFn(x), WhenFn(y))*

Temporal inclusion of subprocesses

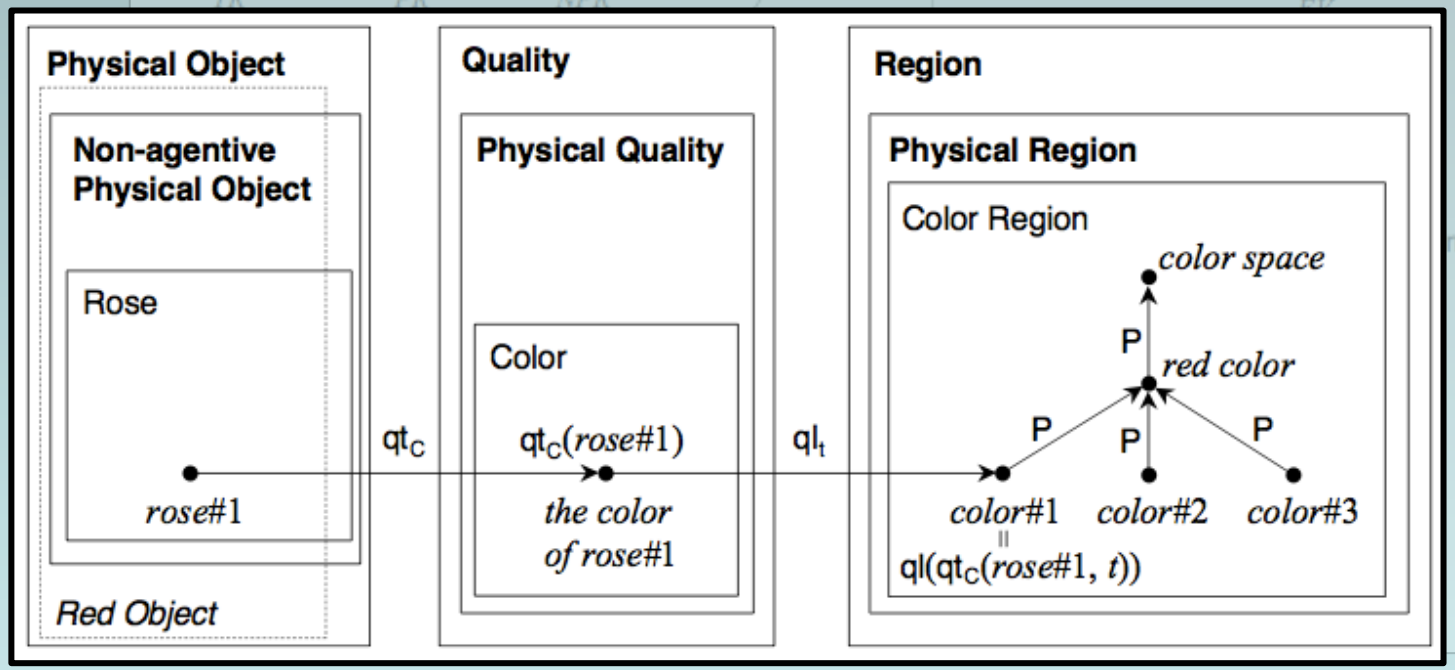
http://www.ontologyportal.org/

# Upper ontologies: ex. top level of DOLCE




http://www.loa-cnr.it/DOLCE.html

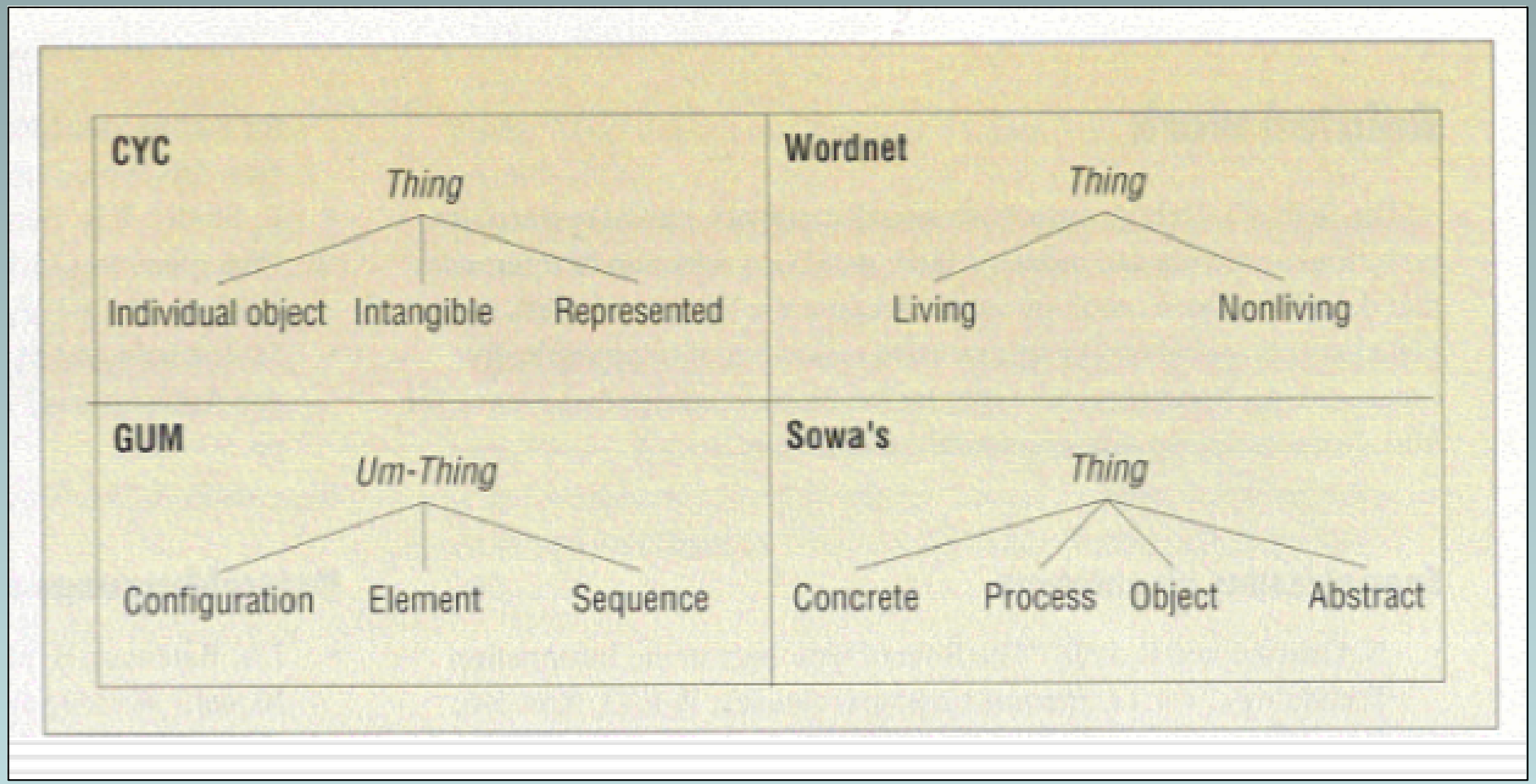
# Upper ontologies: ex. top level of DOLCE



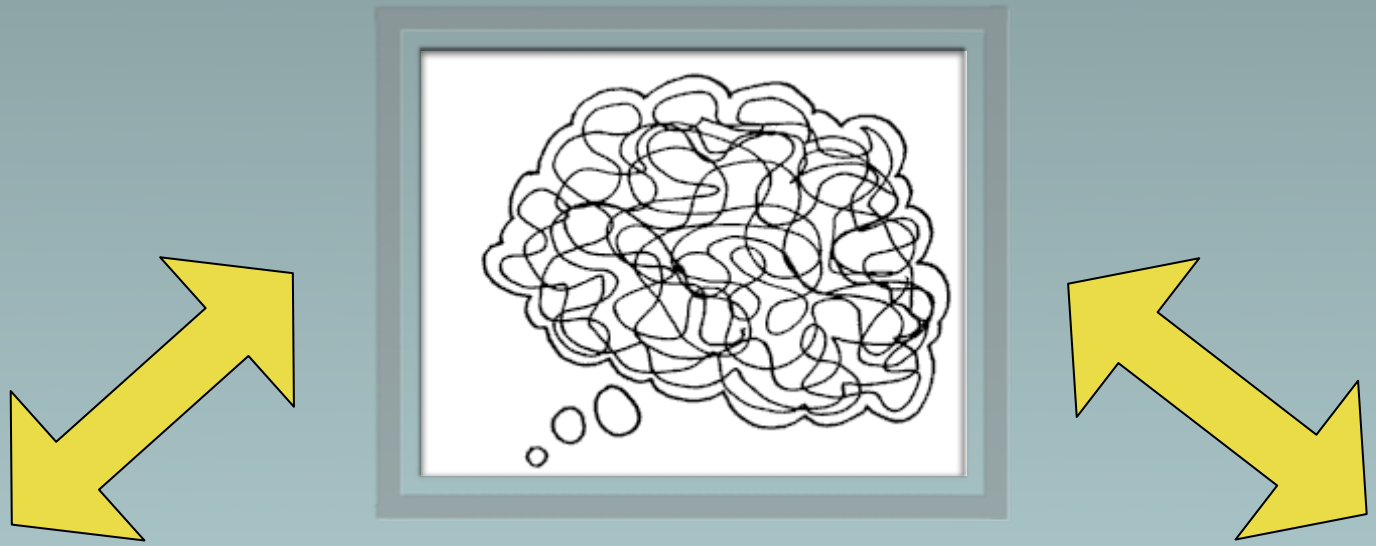



http://www.loa-cnr.it/DOLCE.html

# Upper ontologies: not only one proposal!



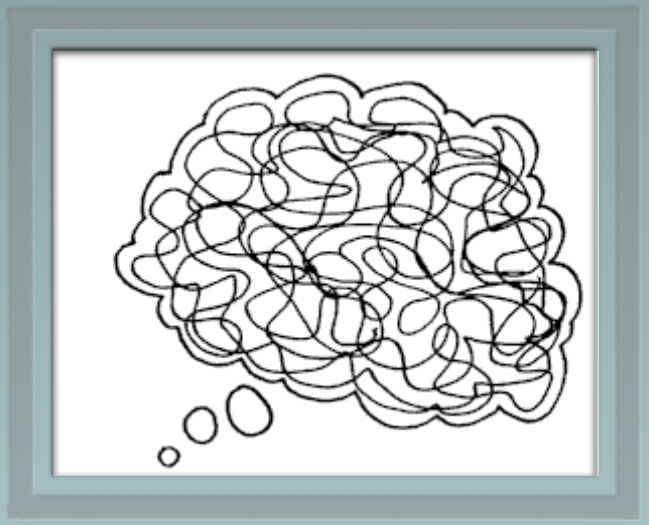
# ‘Realist’ vs ‘Conceptualist’ approaches:

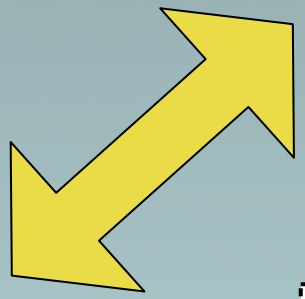


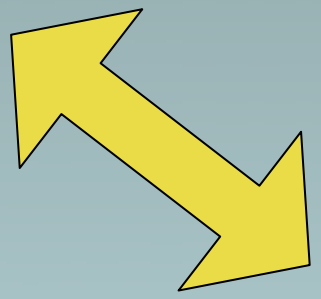




# ‘Realist’ vs ‘Conceptualist’ approaches:









eg **DOLCE**: reality is socially constructed; ontologies should thus have a ‘cognitive bias’

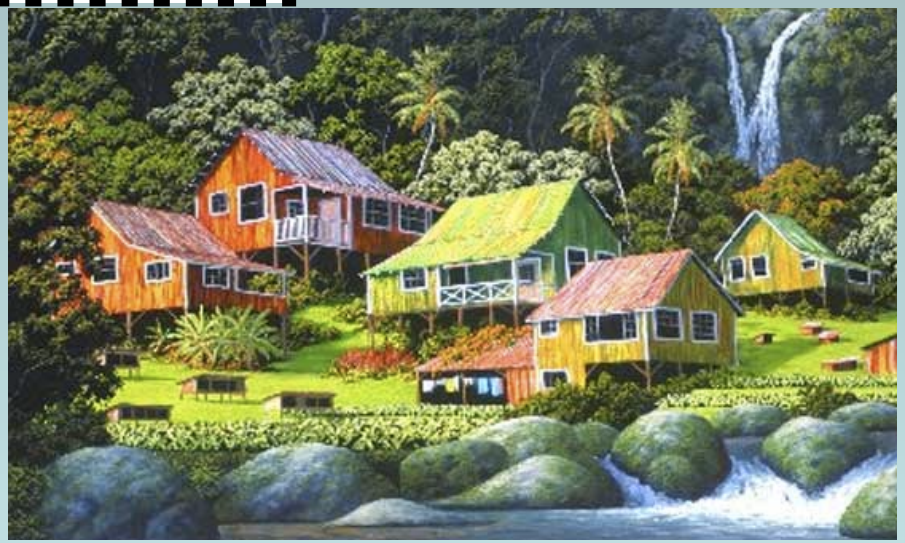


# ‘Realist’ vs ‘Conceptualist’ approaches:

eg **BFO**:

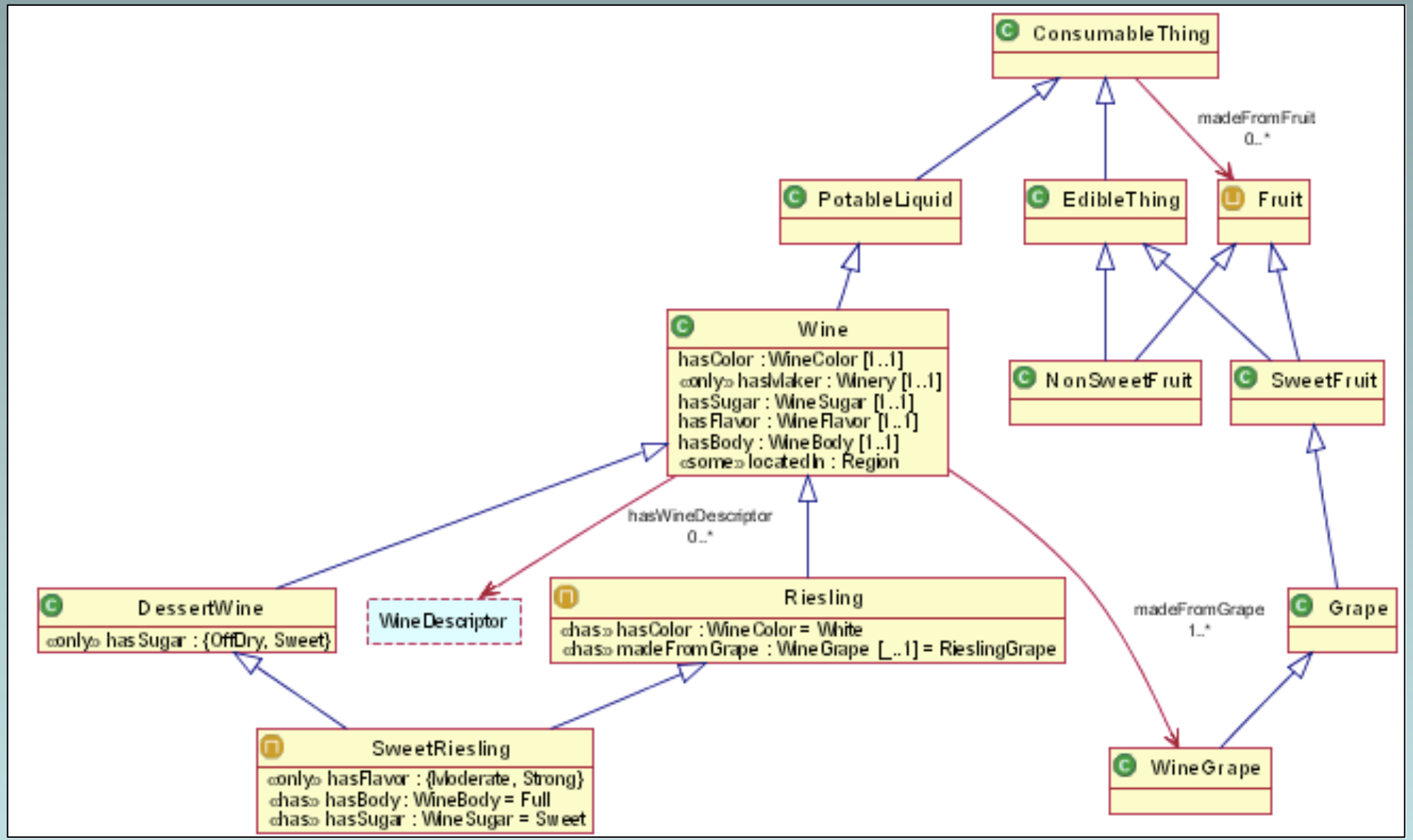
ontologies must mirror the *true* reality, that is what is (incrementally) discovered with the latest scientific experiments





**...domain ontologies...**

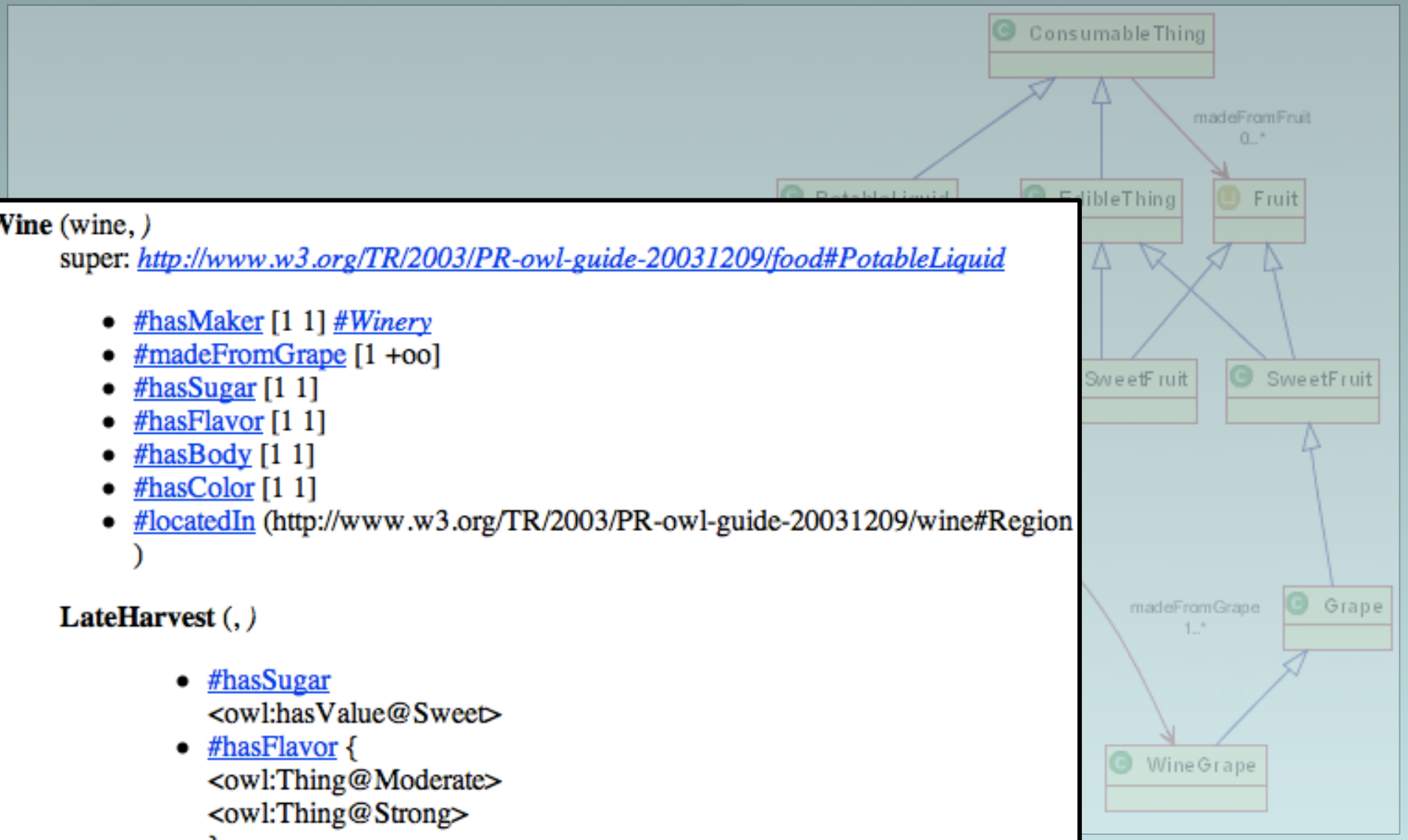
# Domain ontologies: e.g. Wine ontology






http://www.ksl.stanford.edu/projects/wine/explanation.html

# Domain ontologies: e.g. Wine ontology




**Wine** (wine, )
super: *http://www.w3.org/TR/2003/PR-owl-guide-20031209/food#PotableLiquid*

- #hasMaker [1 1] *#Winery*
- #madeFromGrape [1 +oo]
- #hasSugar [1 1]
- #hasFlavor [1 1]
- #hasBody [1 1]
- #hasColor [1 1]
- #locatedIn (http://www.w3.org/TR/2003/PR-owl-guide-20031209/wine#Region )

**LateHarvest** (, )

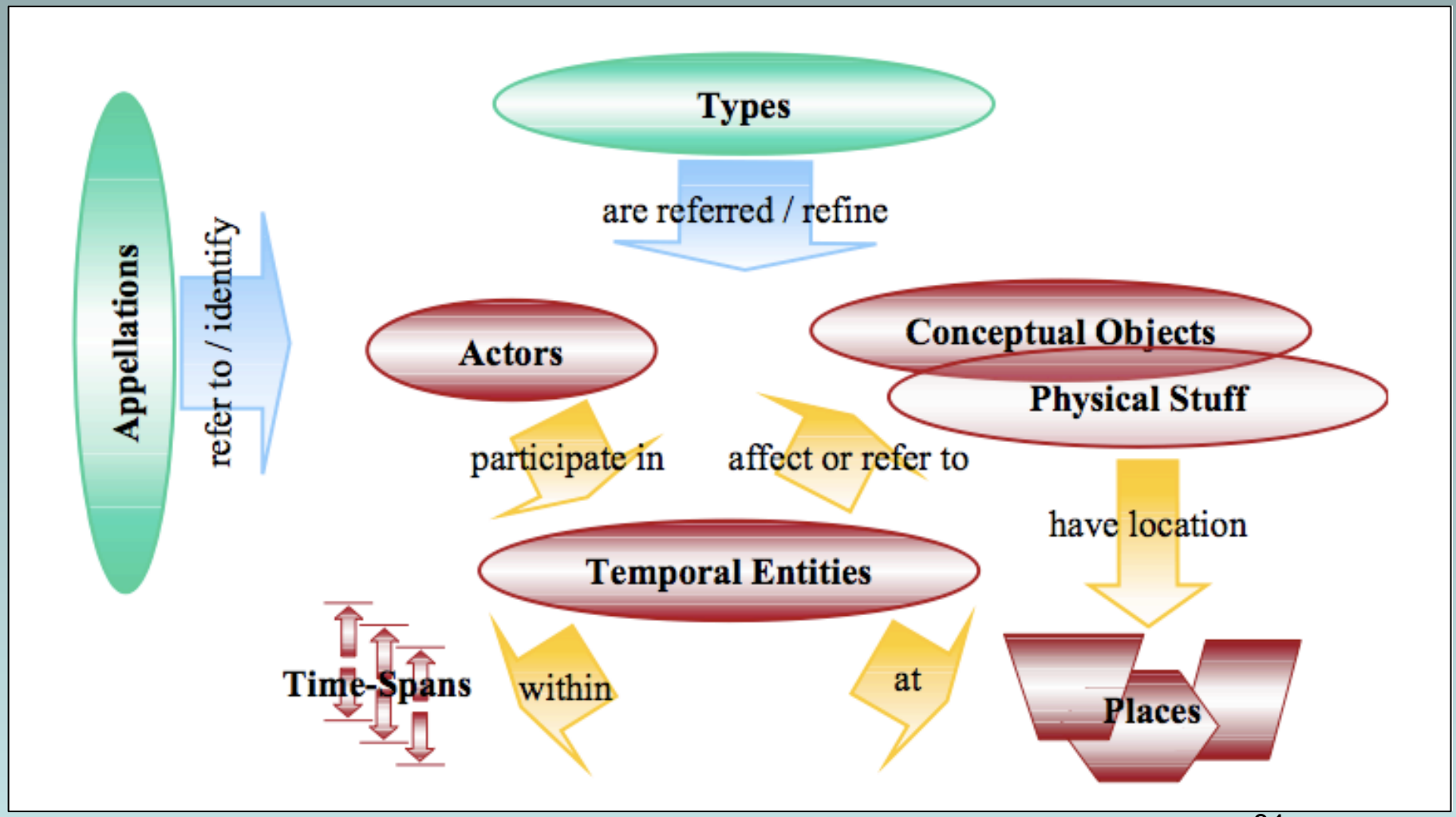
- #hasSugar
  <owl:hasValue@Sweet>
- #hasFlavor {
  <owl:Thing@Moderate>
  <owl:Thing@Strong>
  }



http://www.ksl.stanford.edu/projects/wine/explanation.html

# Domain ontologies: e.g. CIDOC CRM

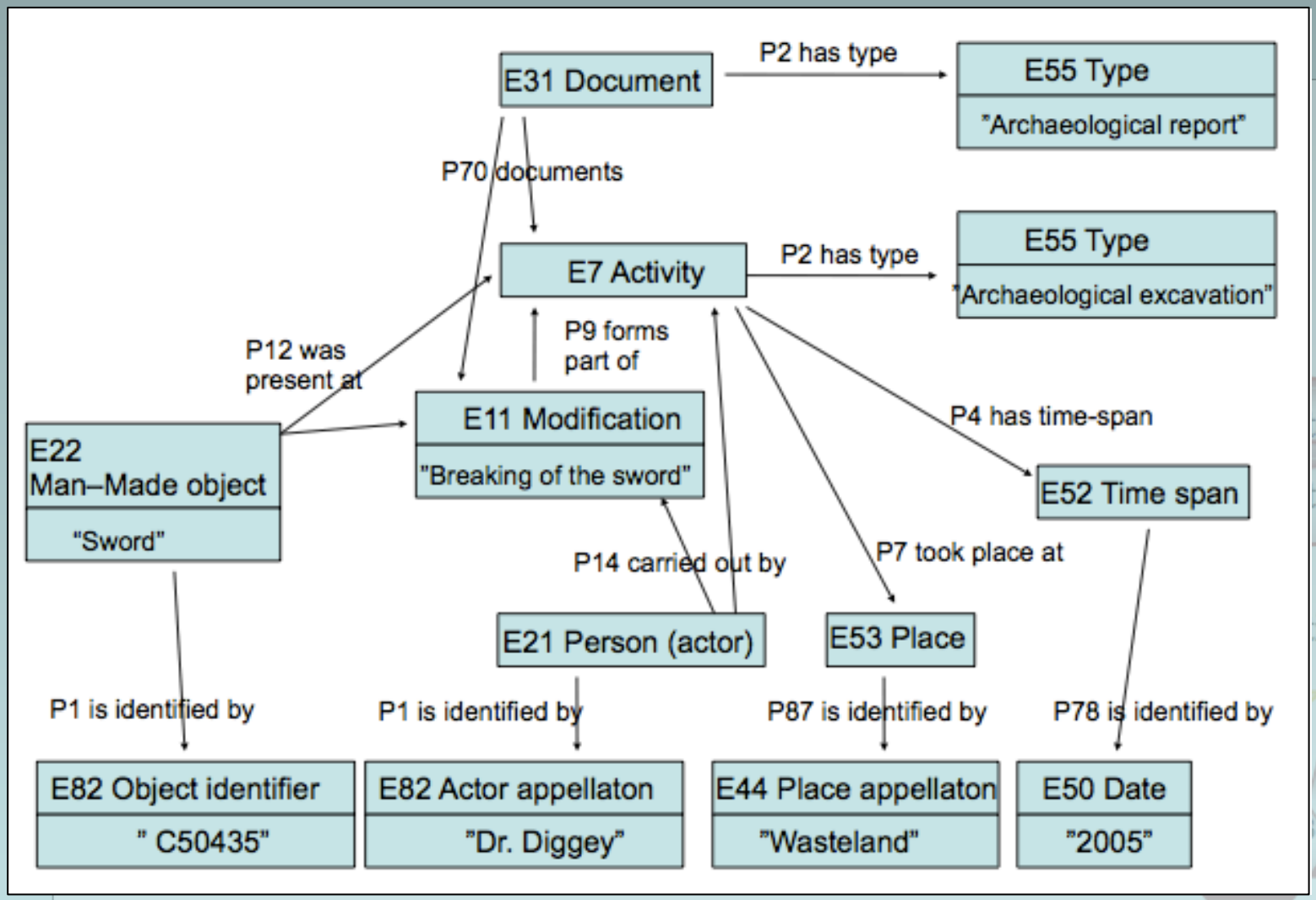
1996 ICOM initiative, 2006 **ISO standard** (version 4.2)






http://www.cidoc-crm.org/

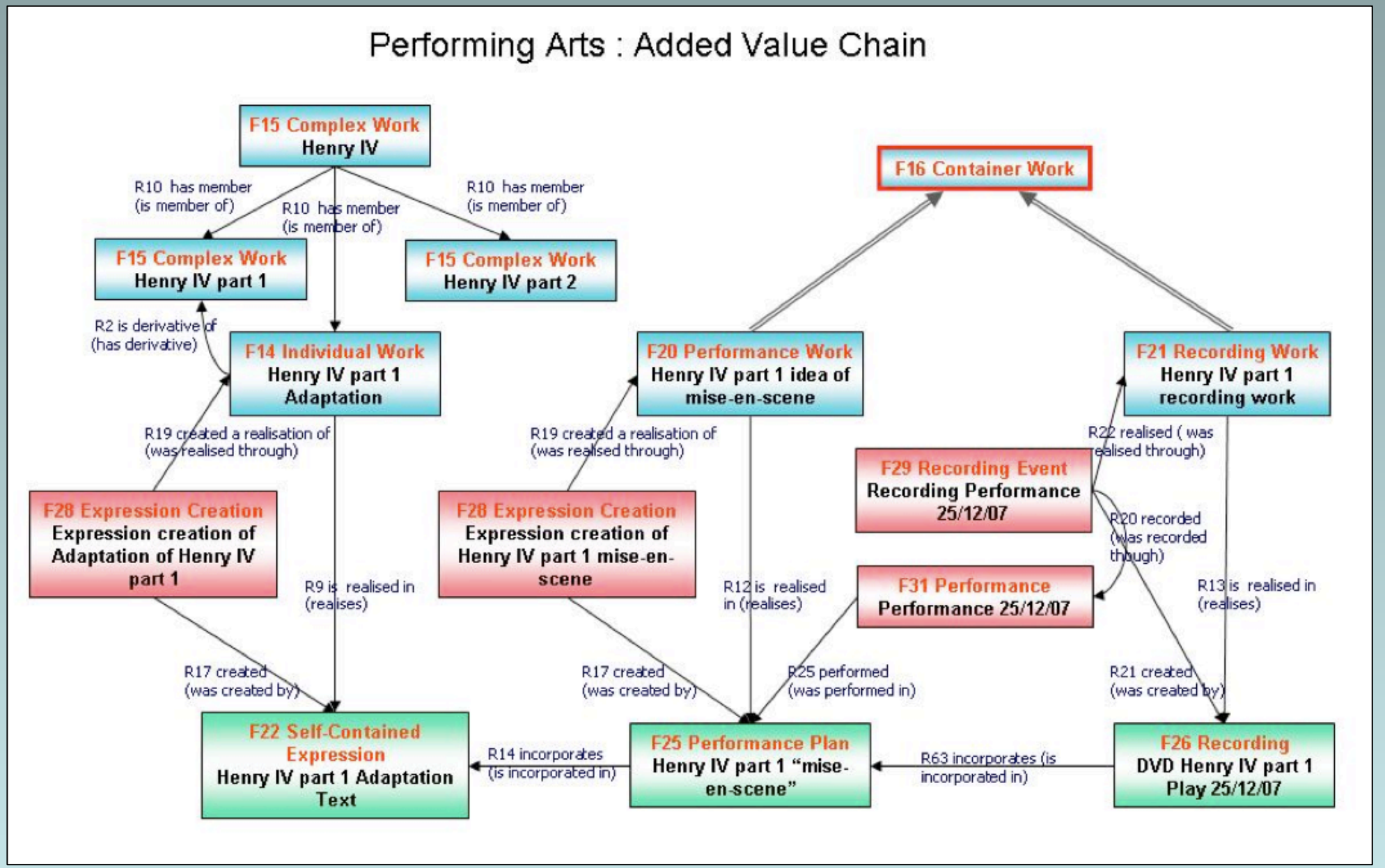
# Domain ontologies: e.g. CIDOC CRM





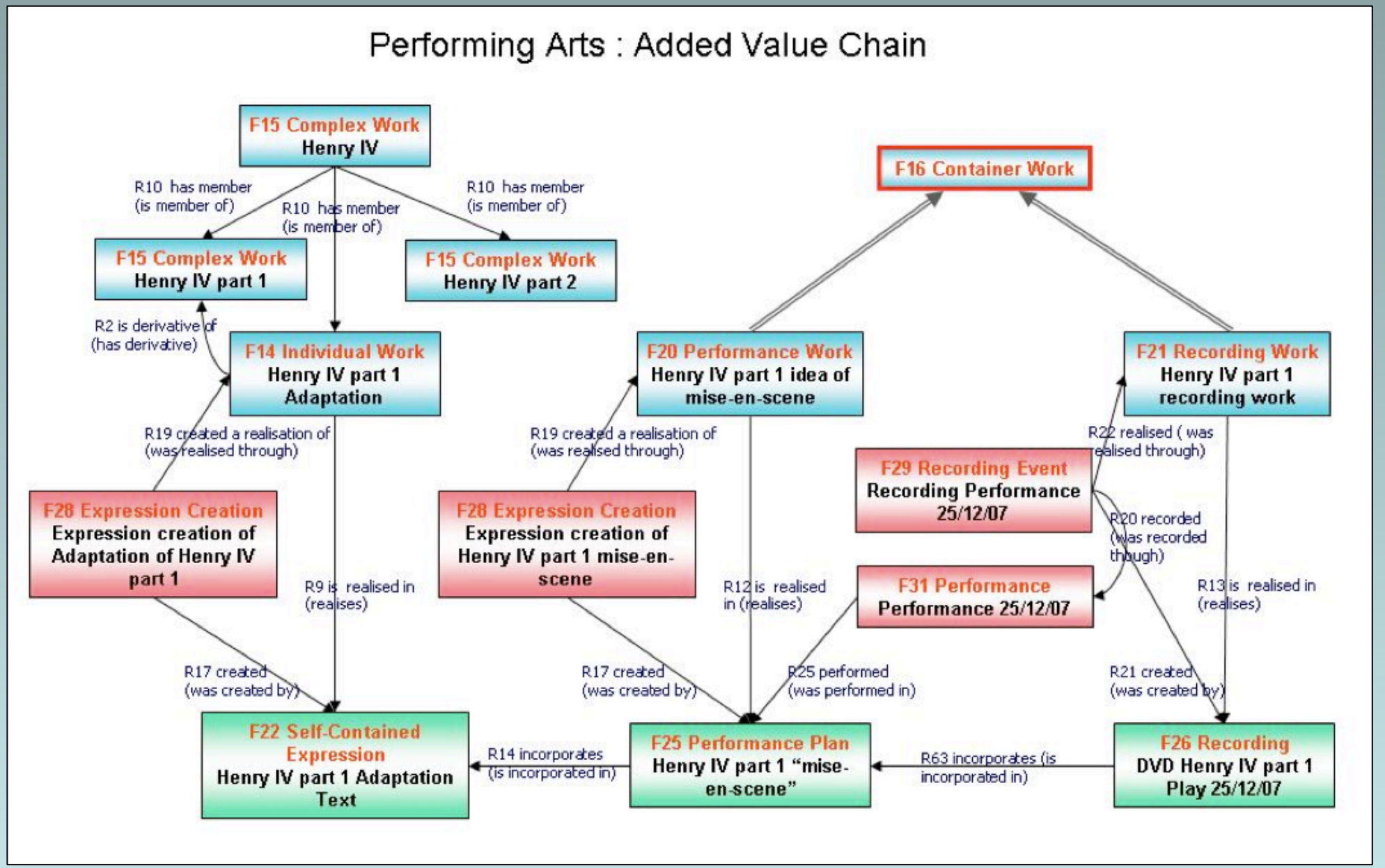

http://www.cidoc-crm.org/

# Domain ontologies: e.g. FRBR-OO (CIDOC)




http://www.cidoc-crm.org/frbr_inro.html

# Domain ontologies: e.g. FRBR-OO (CIDOC)




http://www.cidoc-crm.org/frbr_inro.html

**what is it good for?**

# What is an ontology (as KR) good for?

- to enable **data exchange** among programs
- to simplify unification (or **translation**) of disparate representations

- to embody the representation of a **theory**
- as a **reference** to guide new formalizations
- to facilitate **communication** among people
- to **label** the data you are collecting

- to **find** or **browse** data
- to **reason** with data when you find it
- to build a knowledge model that will **stand the test of time**

# What is an ontology (as KR) good for?

- to enable **data exchange** among programs
- to simplify unification (or **translation**) of disparate representations

- to embody the representation of a **theory**
- as a **reference** to guide new formalizations
- to facilitate **communication** among people
- to **label** the data you are collecting

- to **find** or **browse** data
- to **reason** with data when you find it
- to build a knowledge model that will **stand the test of time**

# What is an ontology (as KR) good for?

- to enable **data exchange** among programs
- to simplify unification (or **translation**) of disparate representations

- to embody the representation of a **theory**
- as a **reference** to guide new formalizations
- to facilitate **communication** among people
- to **label** the data you are collecting

- to **find** or **browse** data
- to **reason** with data when you find it
- to build a knowledge model that will **stand the test of time**

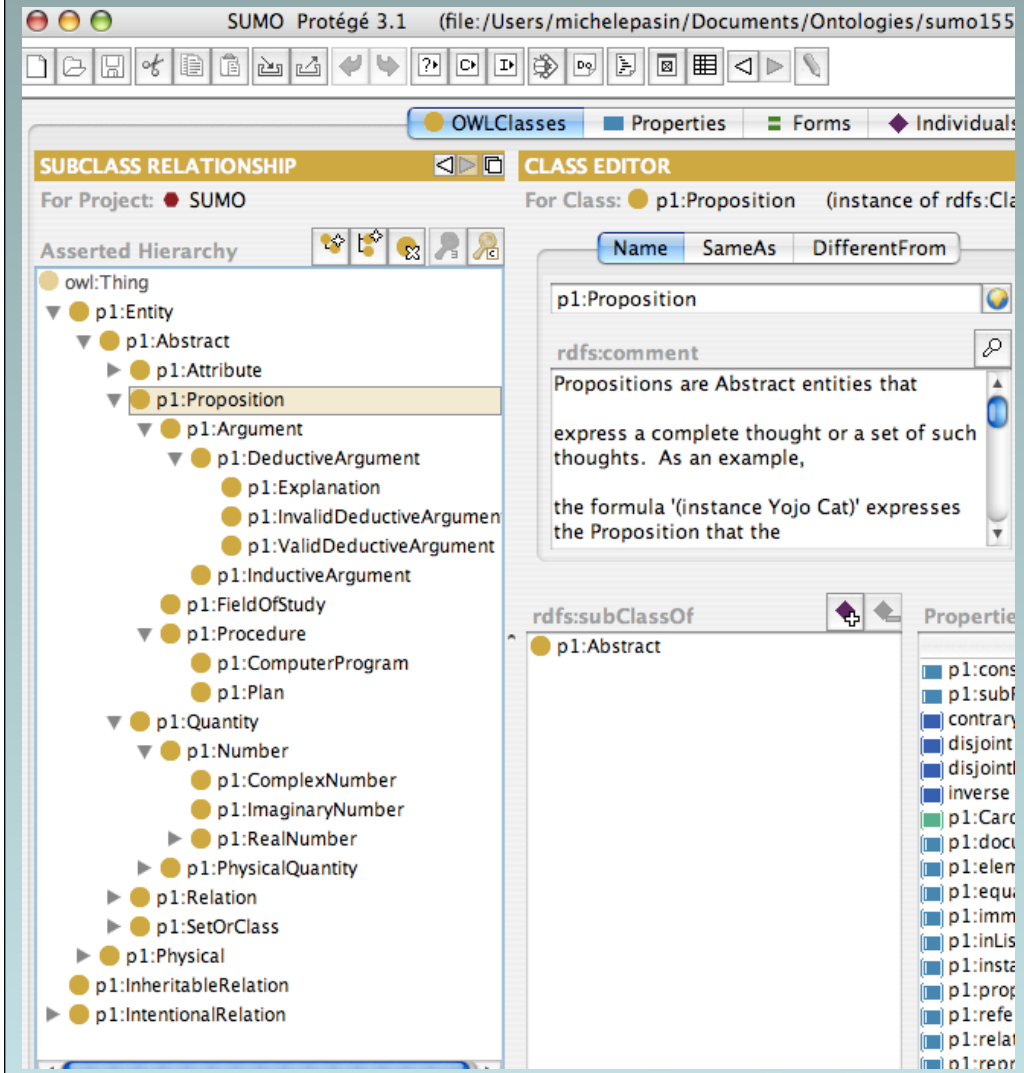
# What is an ontology (as KR) good for?

- to enable **data exchange** among programs
- to simplify unification (or **translation**) of disparate representations

- to embody the representation of a **theory**
- as a **reference** to guide new formalizations
- to facilitate **communication** among people
- to **label** the data you are collecting

- to **find** or **browse** data
- to **reason** with data when you find it
- to build a knowledge model that will **stand the test of time**

# Principle #1: ontology as a program




1. An ontology is an explicit, formal specification of a **theory**

2. An ontology is a **model** that can be manipulated by a computer

3. An ontology can be *run* as we run computer **programs**

# Principle #2: ontology as a contract

## A Pragmatic Perspective

- Ontologies are not about truth or beauty.
- They are agreements, made in a social context, to accomplish some objectives.
- It's important to understand those objectives, and be guided by them.



*Gruber. It Is What It Does: The Pragmatics of Ontology. Invited presentation to the meeting of the CIDOC Conceptual Reference Model committee (2003)*

**software applications**

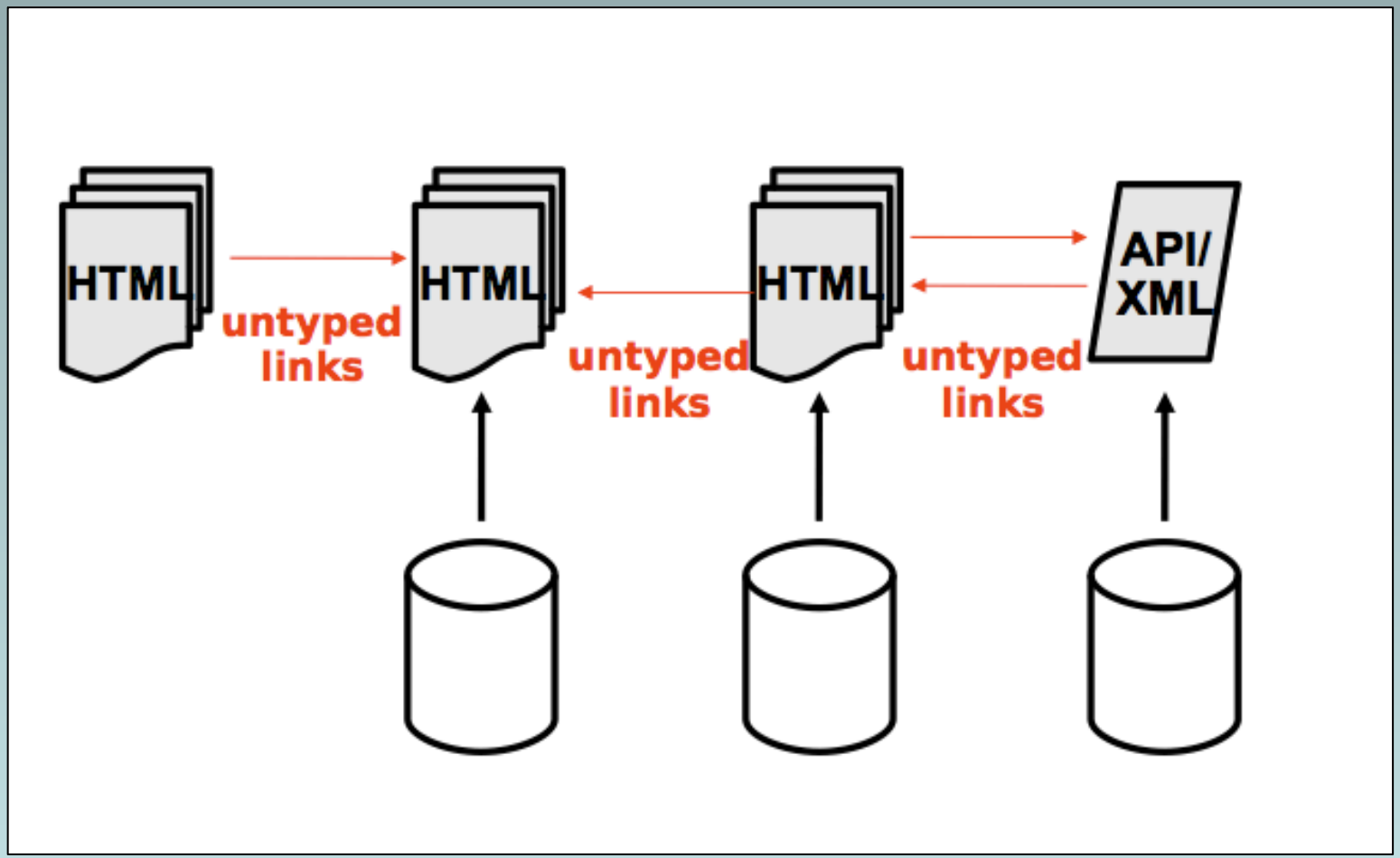
**research communities**

## Common ‘things’ we (as DHs) mention in our contracts:

- information objects
  - key characteristics of entities that can carry information, that can be seen as (or part of) a *representation*
- physical features of information objects
  - e.g., materials, conditions, preservation ...
- abstract features of information objects
  - e.g., the *contents* of an information object, the Hamlet as a work
  - e.g., the *linguistic* features of an information object (latin, english, etc.)
  - e.g., aspects of the *discourse* used to communicate the contents of an information object (e.g., proem, dispositive word, bound, curse etc.). These aspects will vary with different projects!

# Ontologies in the Semantic Web / Web of Data

*from here.....*

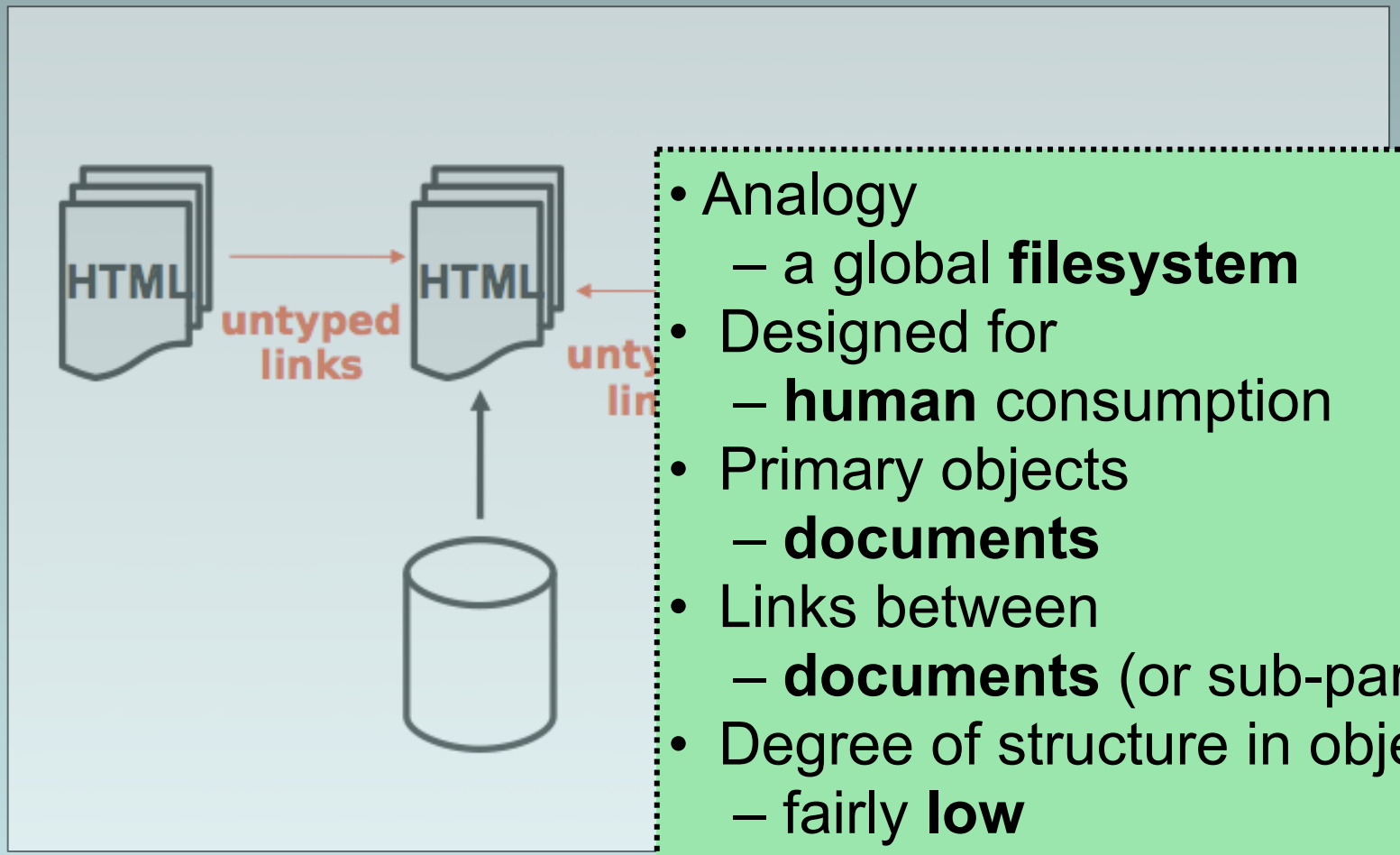





Heath. An Introduction to Linked Data. (2007) http://tomheath.com/slides/2009-02-austin-linkeddata-tutorial.pdf

# Ontologies in the Semantic Web / Web of Data

*from here.....*

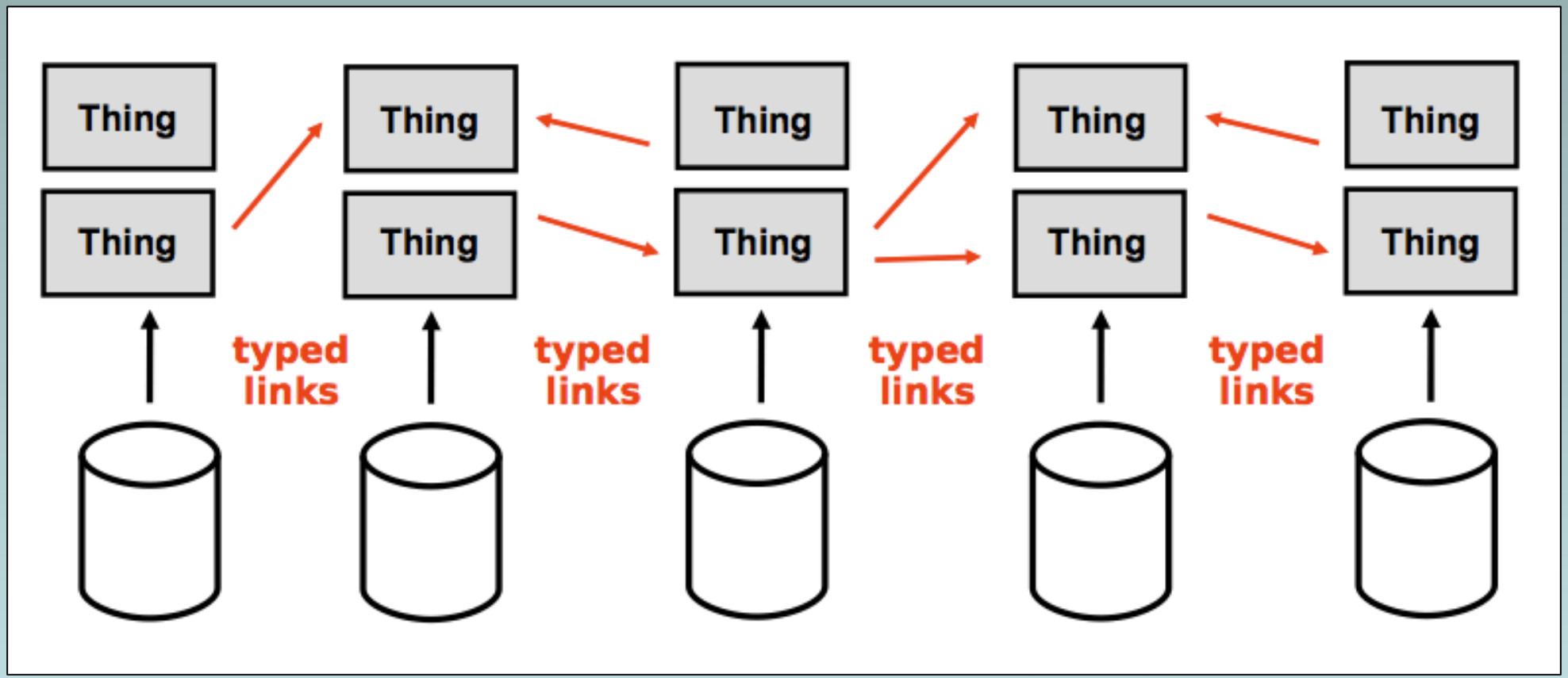



- Analogy
  - a global **filesystem**
- Designed for
  - **human** consumption
- Primary objects
  - **documents**
- Links between
  - **documents** (or sub-parts of)
- Degree of structure in objects
  - fairly **low**
- Semantics of content and links
  - **implicit**

Heath. An Introduction to Linked Data. (2007) http://tomhea

# Ontologies in the Semantic Web / Web of Data

*to here!*

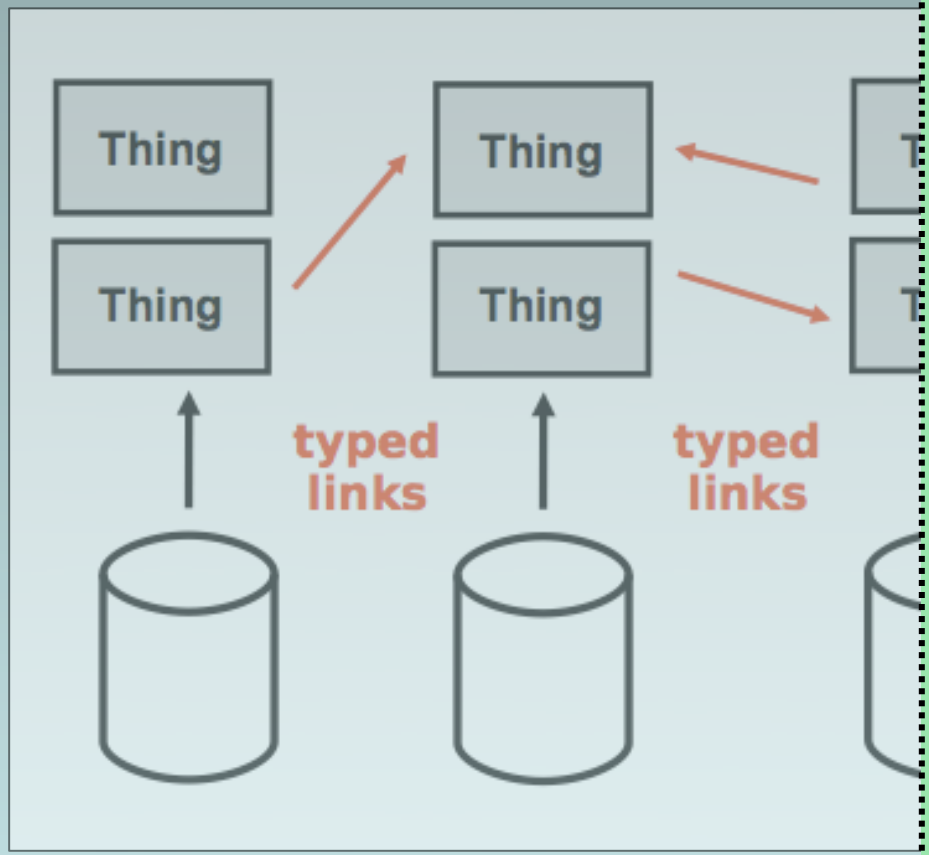





http://www.w3.org/2001/sw/

# Ontologies in the Semantic Web / Web of Data

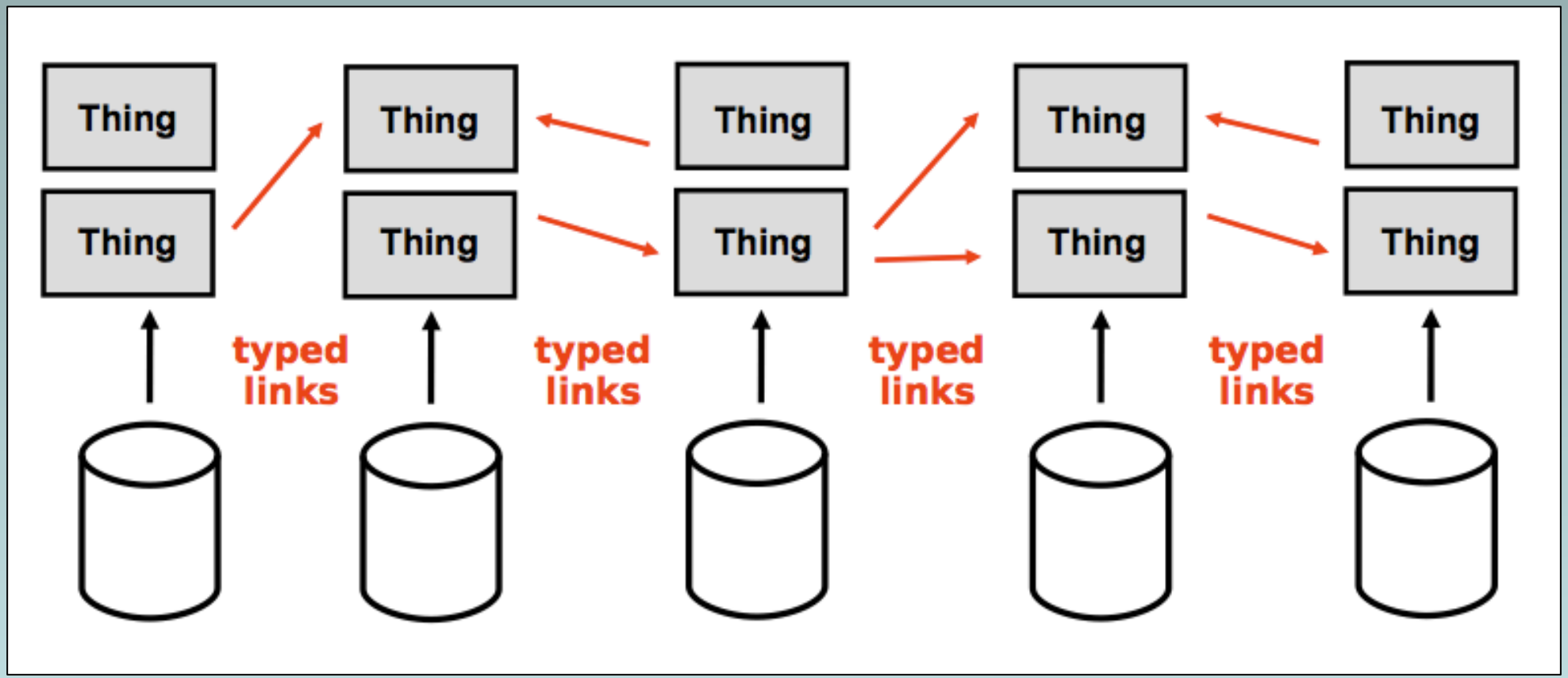
*to here!*



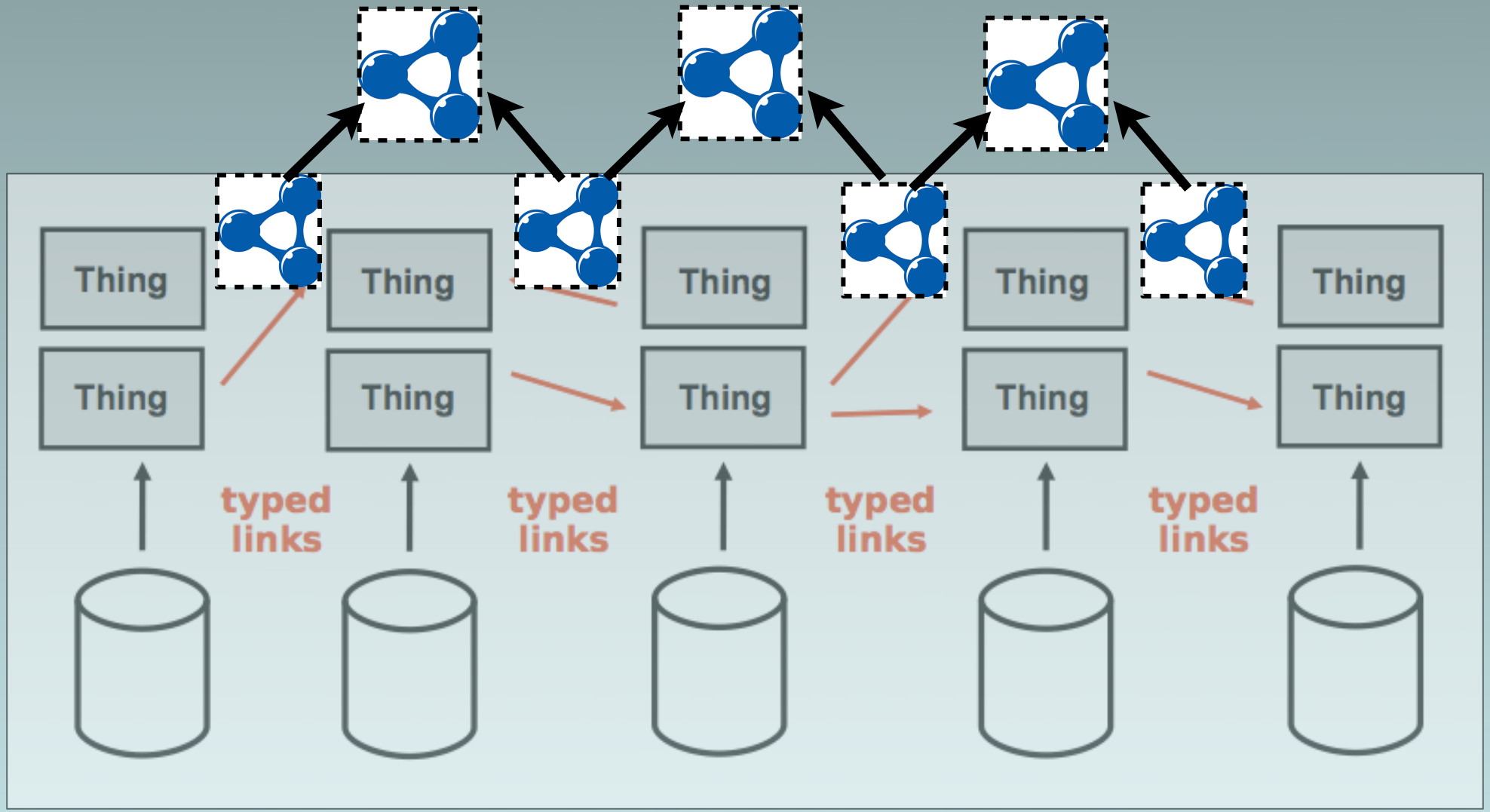

- Analogy
  - a global **database**
- Designed for
  - **machines** first, **humans** later
- Primary objects
  - **things** (or **descriptions** of things)
- Links between
  - **things**
- Degree of structure in (descriptions of) things
  - **high**
- Semantics of content and links
  - **explicit**

http://www.w3.org/2001/sw/

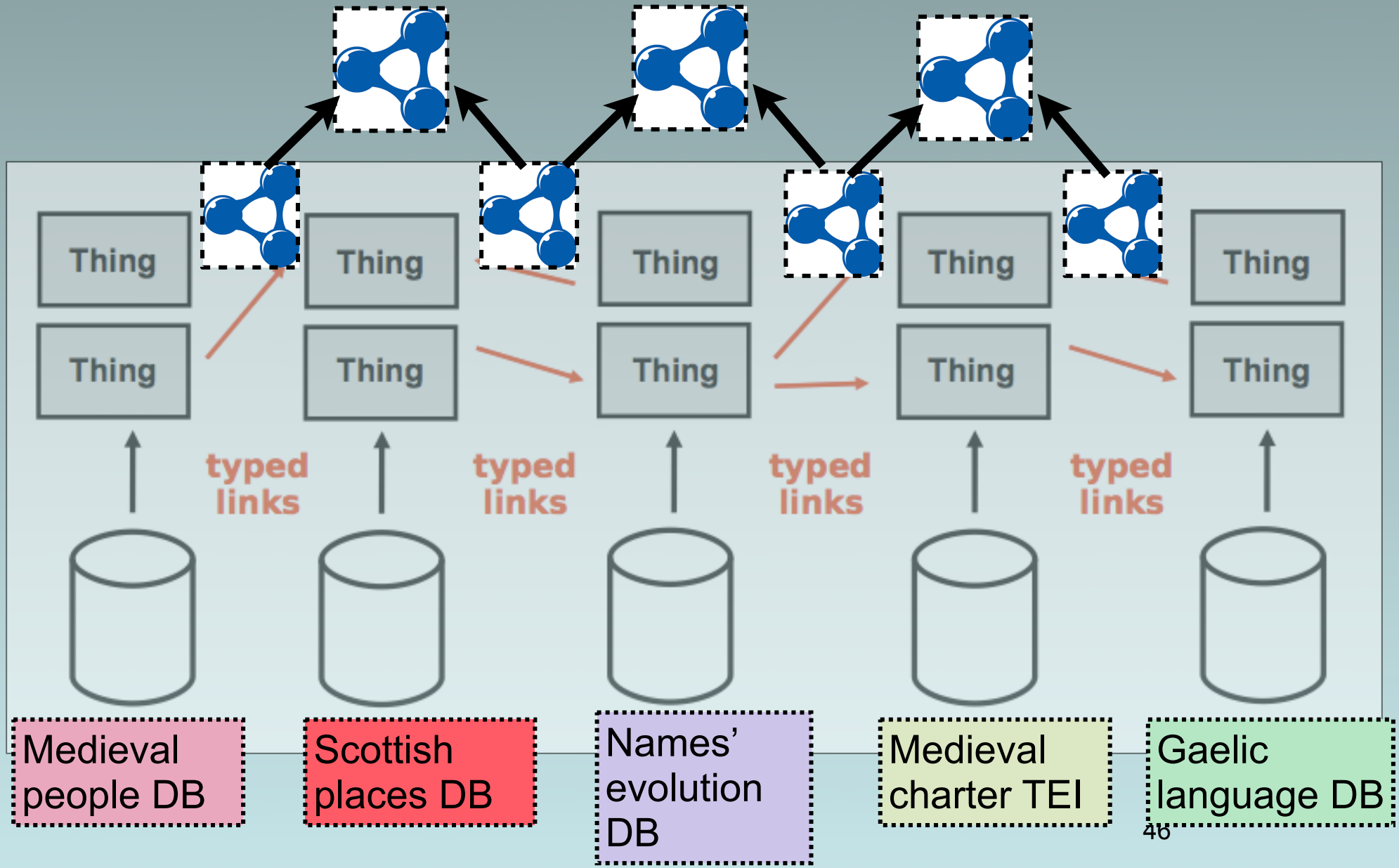
# Ontologies in the Semantic Web / Web of Data



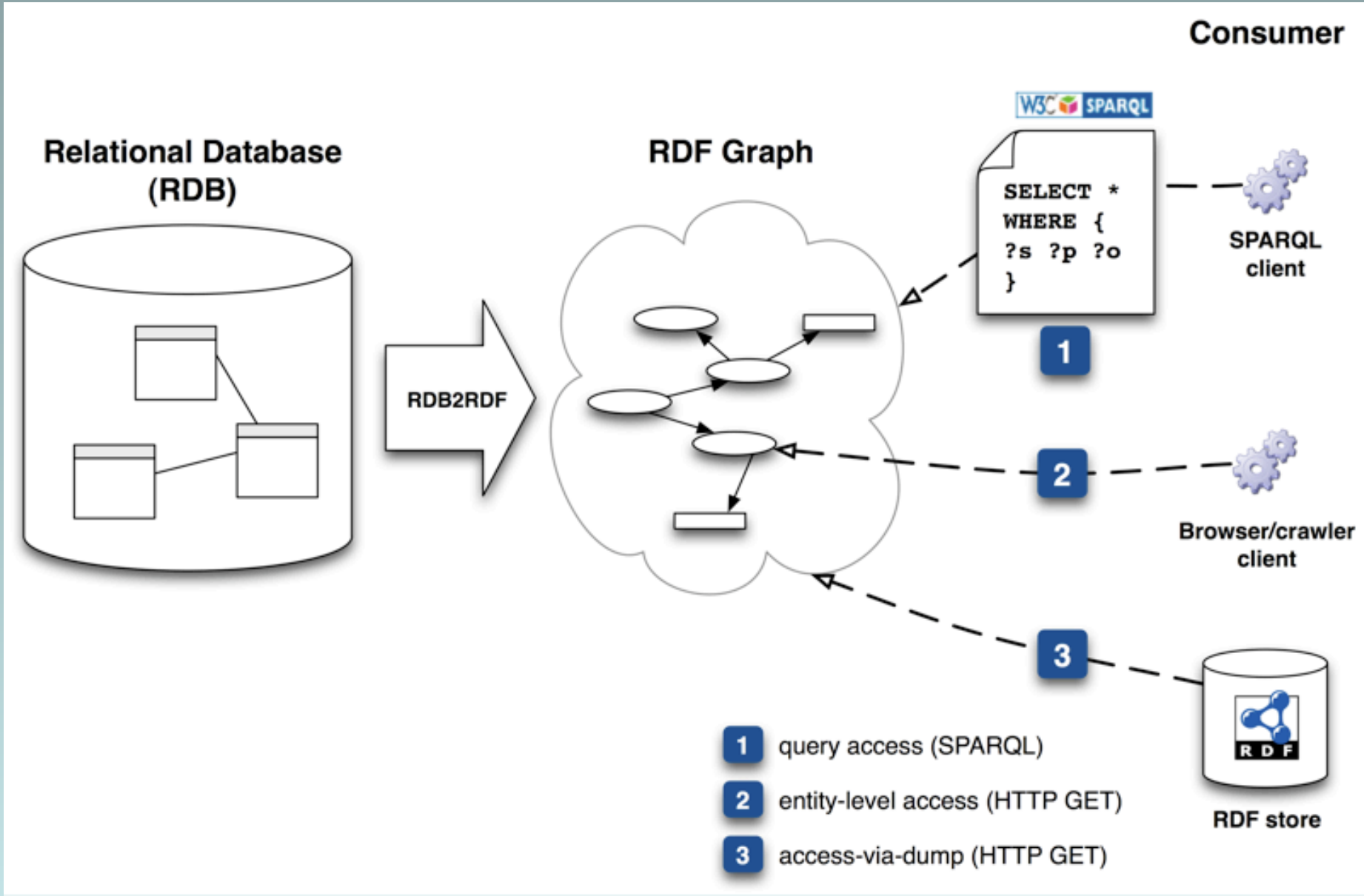
# Ontologies in the Semantic Web / Web of Data



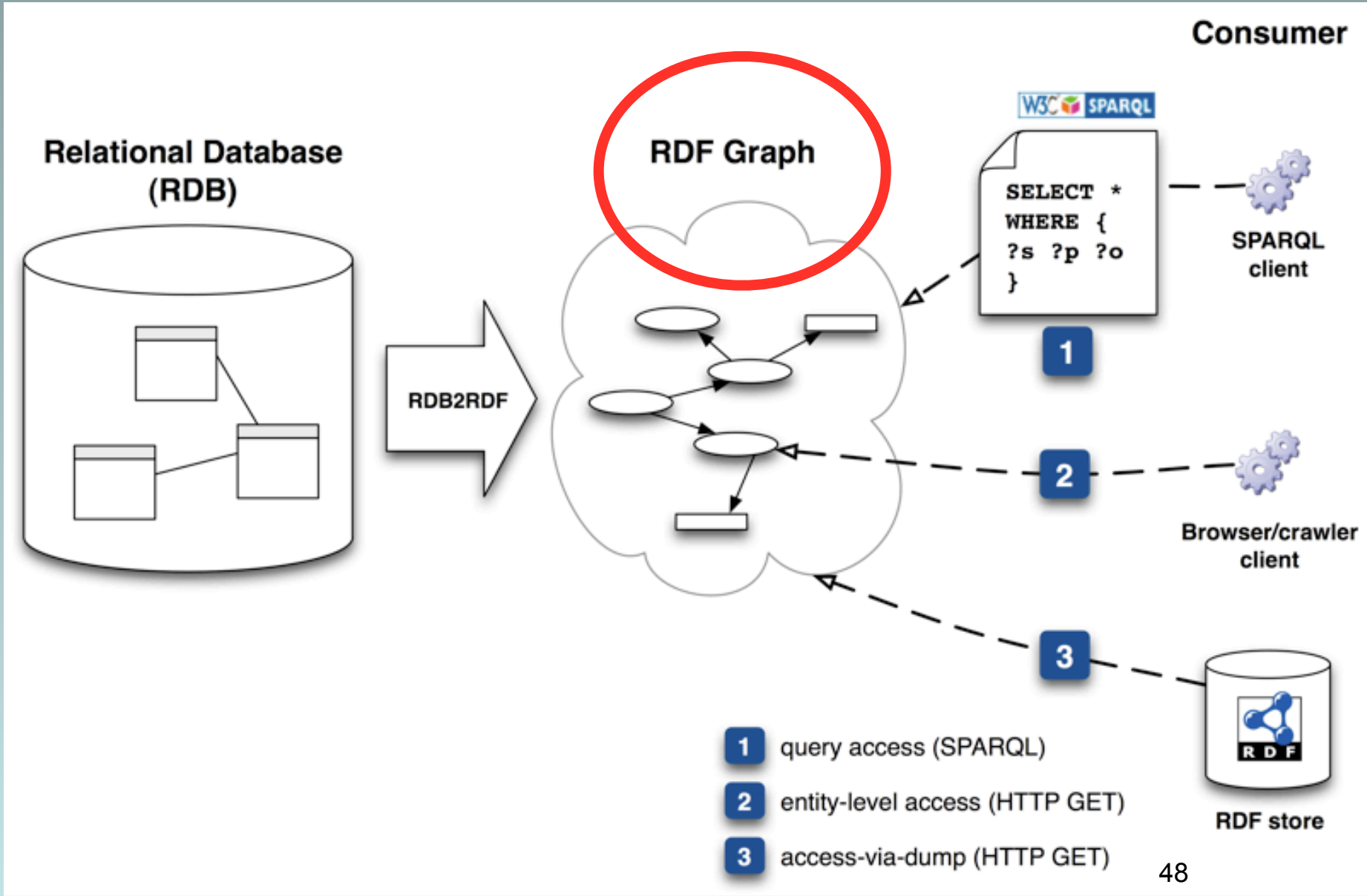
# Ontologies in the Semantic Web / Web of Data



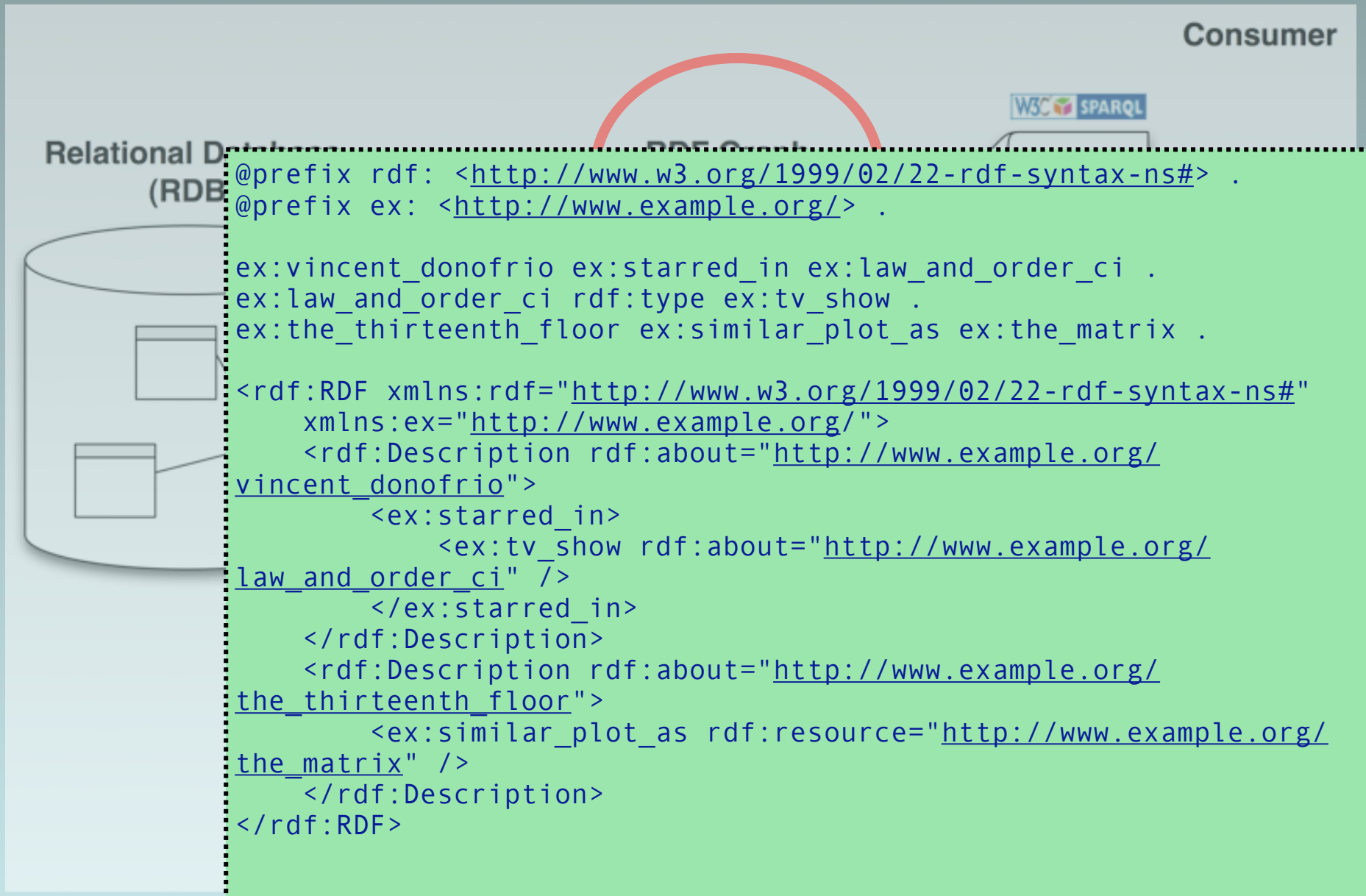
# Semantic Web: the technology stack



# The Resource Description Framework (RDF)

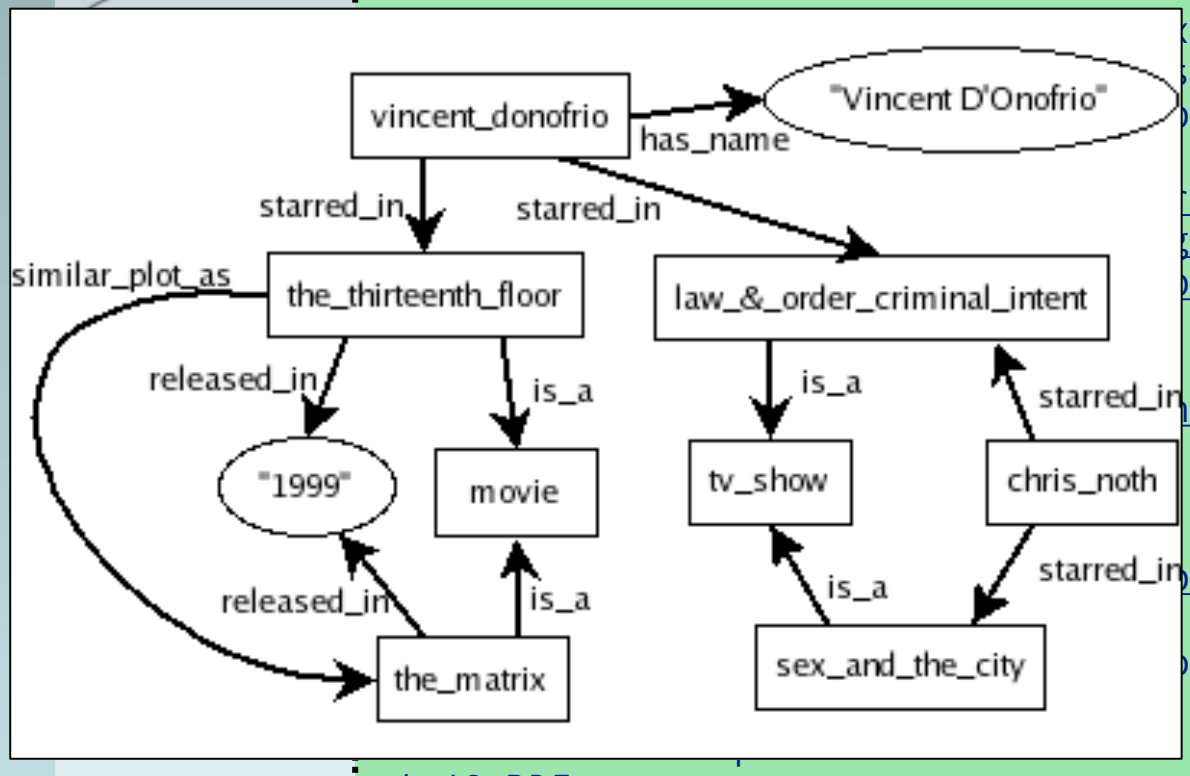


# The Resource Description Framework (RDF)



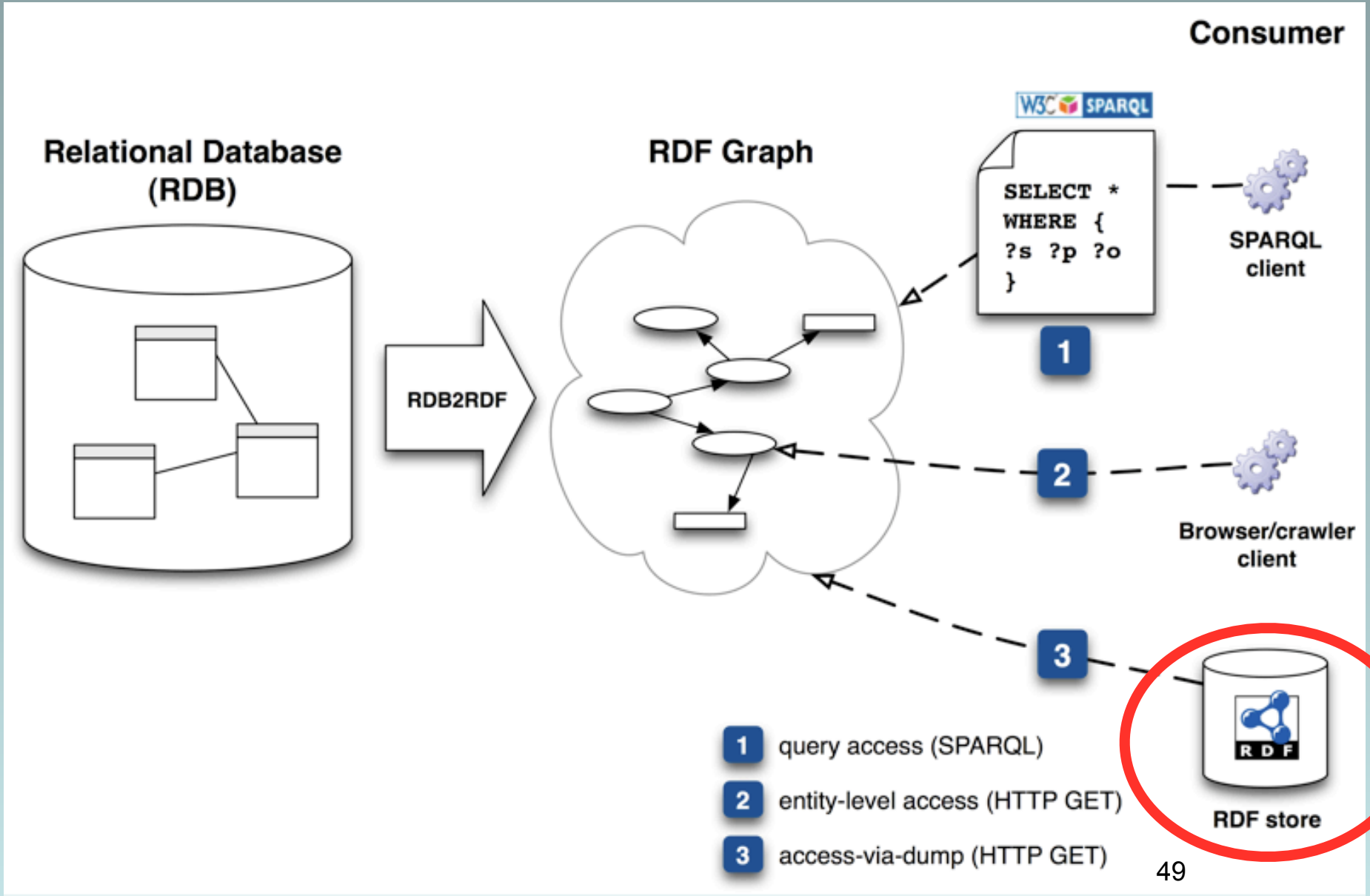

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ex: <http://www.example.org/> .

ex:vincent_donofrio ex:starred_in ex:law_and_order_ci .
ex:law_and_order_ci rdf:type ex:tv_show .
ex:the_thirteenth_floor ex:similar_plot_as ex:the_matrix .

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:ex="http://www.example.org/">
    <rdf:Description rdf:about="http://www.example.org/
vincent_donofrio">
        <ex:starred_in>
            <ex:tv_show rdf:about="http://www.example.org/
law_and_order_ci" />
        </ex:starred_in>
    </rdf:Description>
    <rdf:Description rdf:about="http://www.example.org/
the_thirteenth_floor">
        <ex:similar_plot_as rdf:resource="http://www.example.org/
the_matrix" />
    </rdf:Description>
</rdf:RDF>
```

# The Resource Description Framework (RDF)

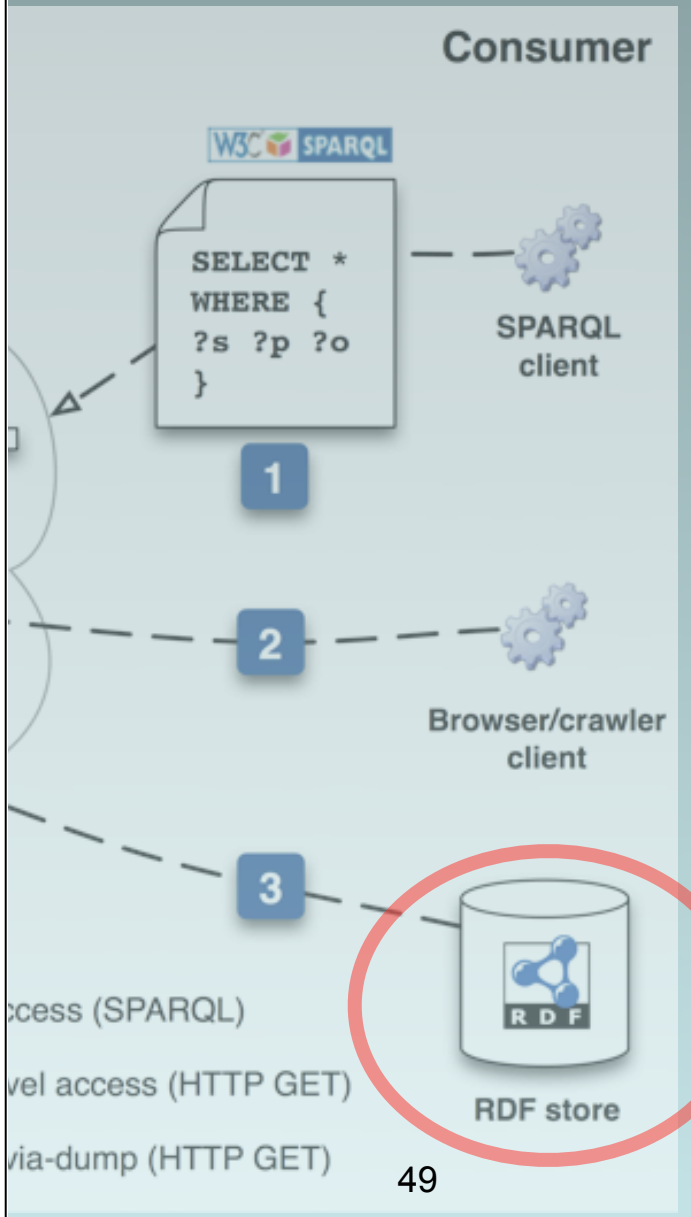



```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ex: <http://www.example.org/> .

                                        :law_and_order_ci .
                                        how .
                                        lot_as ex:the_matrix .

                                        g/1999/02/22-rdf-syntax-ns#"
                                        /">
                                        ://www.example.org/

                                        ttp://www.example.org/

                                        ://www.example.org/

                                        ource="http://www.example.org/

</rdf:RDF>
```

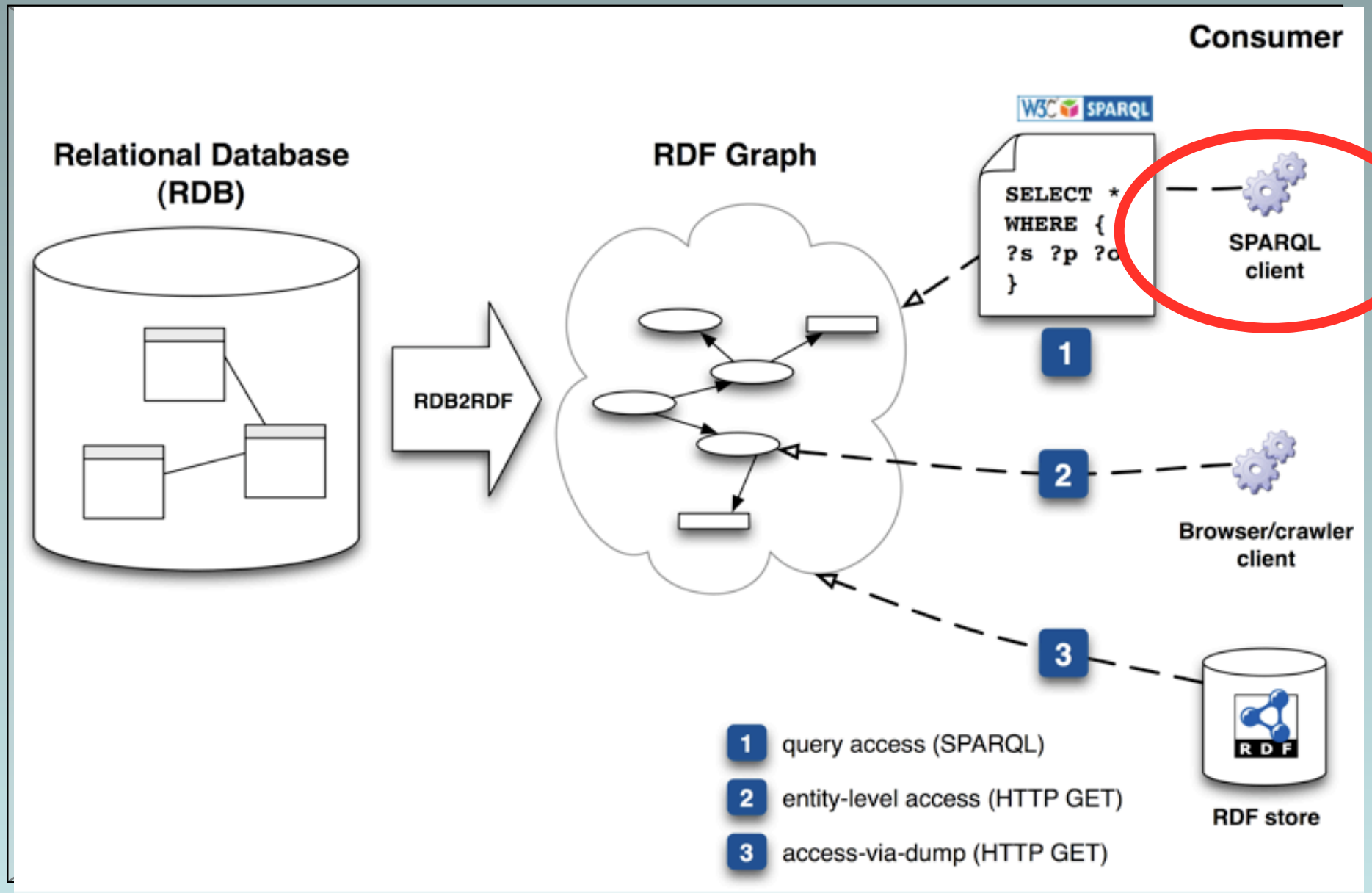
# RDF Triple Stores



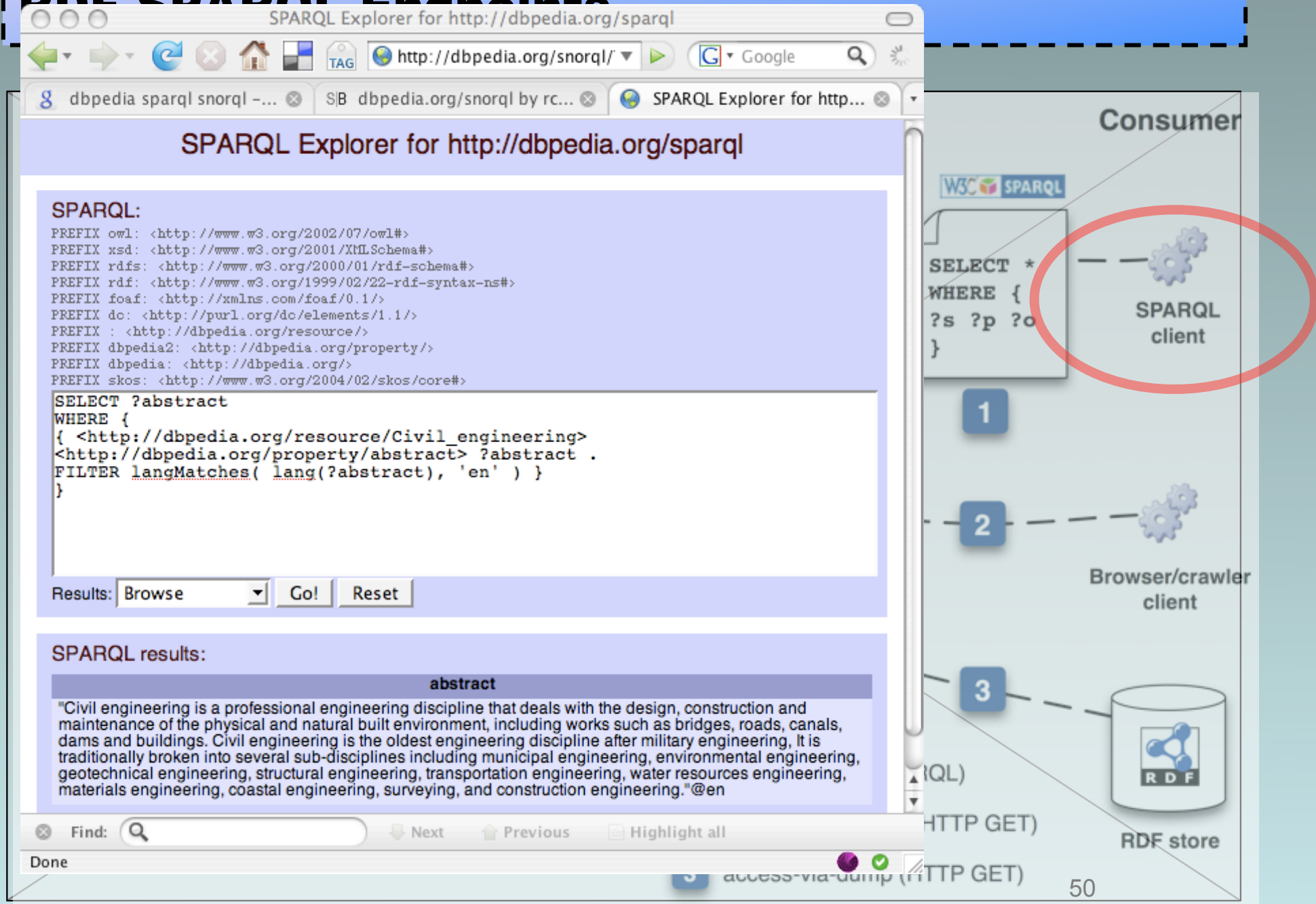
| Name | Language | Homepage |
|---|---|---|
| 3store | C | http://www.aktors.org/technologies/3store/ |
| 4store | C | http://www.4store.org/ |
| 5store | C | http://4store.org/trac/wiki/5store |
| AllegroGraph | ? | http://www.franz.com/agraph/allegrograph/ |
| ARC | PHP | http://arc.semsol.org/ |
| Bigdata | Java | http://www.bigdata.com/ |
| BigOWLIM | Java | http://www.ontotext.com/owlim/ |
| Jena | Java | http://jena.sourceforge.net/ |
| Joseki | Java | http://www.joseki.org/ |
| Mulgara | Java | http://www.mulgara.org/ |
| OpenAnzo | Java | http://www.openanzo.org/ |
| OntoBroker | Java | http://www.ontoprise.de/en/home/products/ontobroker/ |
| Parliament | Java/C++ | http://parliament.semwebcentral.org/ |
| Pointrel System | Java/Python | http://sourceforge.net/projects/pointrel/ |
| RAP | PHP | http://www4.wiwiss.fu-berlin.de/bizer/rdfapi/ |
| RDF::Core | Perl | http://search.cpan.org/dist/RDF-Core/ |
| RDF::Trine | Perl | http://www.perlrdf.org/ |
| RDFBroker | Java | http://rdfbroker.opendfki.de/ |
| Redland | C | http://librdf.org/ |
| RedStore | C | http://www.aelius.com/njh/redstore/ |
| Semantics Platform | C# | http://www.intellidimension.com/ |
| SemWeb-DotNet | C# | http://razor.occams.info/code/semweb/ |
| Sesame | Java | http://www.openrdf.org/ |
| Soprano | C++ | http://soprano.sourceforge.net/ |
| SwiftOWLIM | Java | http://www.ontotext.com/owlim/ |



# RDF SPARQL Endpoints



# RDF SPARQL Endpoints



# Semantic Web technology stack: conclusions

- based on the existing WWW architecture (URIs, HTTP)

- set of standards for encoding the semantics of your data (RDF, RDFS, OWL)

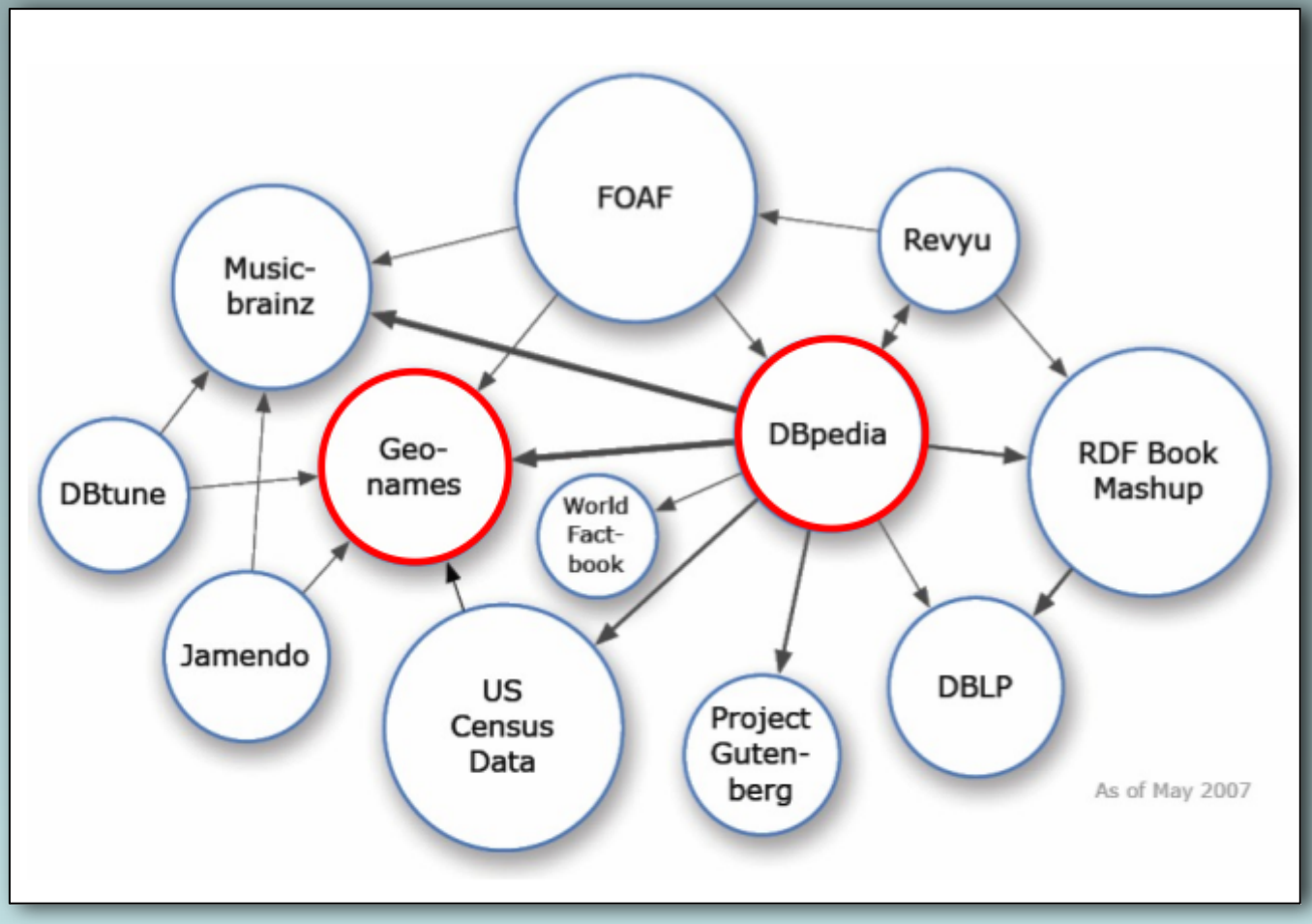
- allow reasoning tasks (eg inferences) using standard SW processors

- provide a single, standardized query mechanism (SPARQL)

- endorsed by the W3C, rapidly adopted by many..

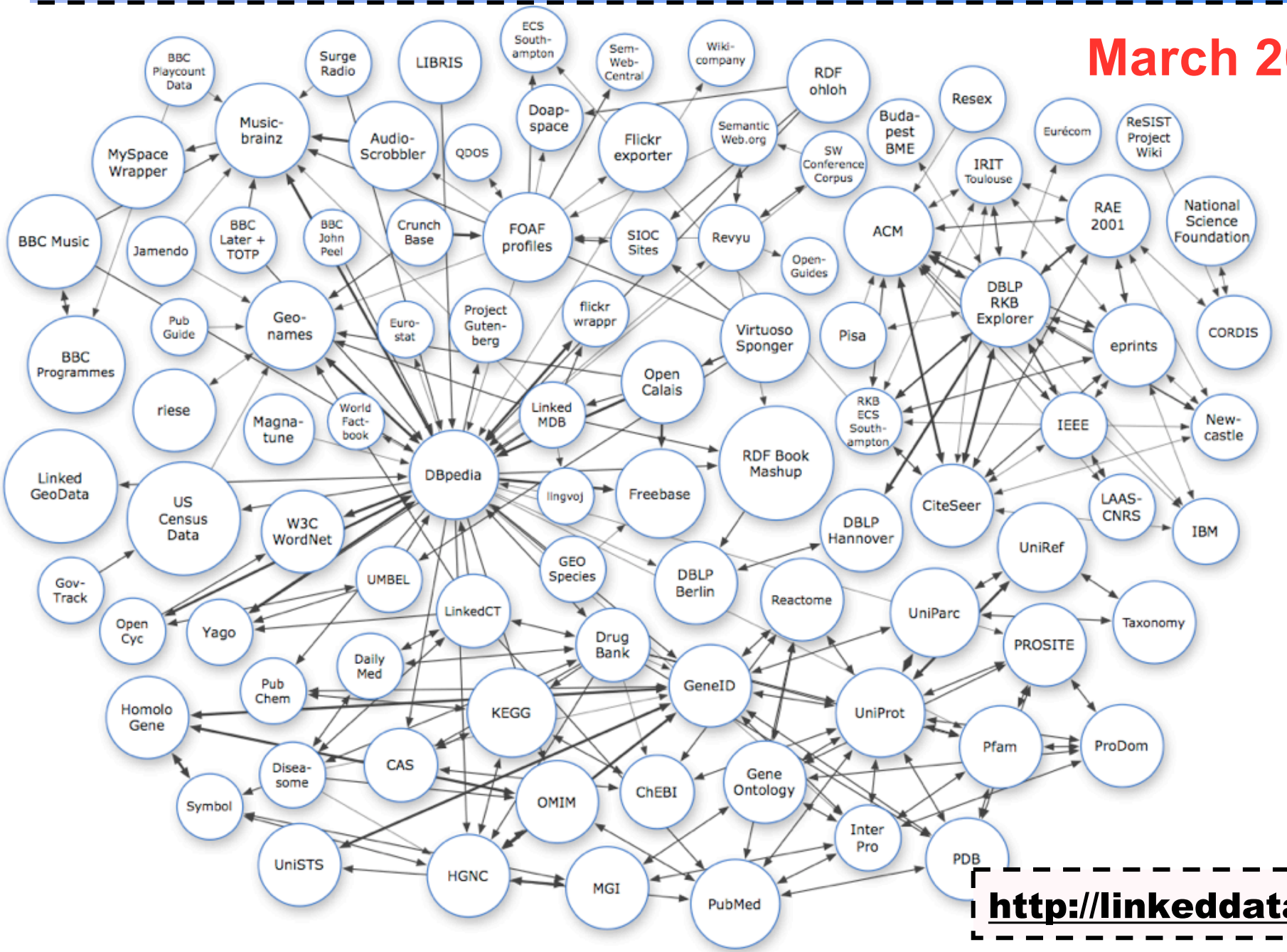
# Approaches: the Linked-Data initiative

**May 2007**




http://linkeddata.org/

# Examples: the Linked-Data initiative (II)



# Conclusion: possible research directions

- modeling exercise with a domain of your choice
  - lots of humanistic domains need more careful examination
- data integration @ CCH
  - lots of TEI / DBs that can be used as a starting point!
  - defining realistic usage scenarions
- do get in touch if you need more info!
  - domain ontologies you can reuse / extend
  - implementation languages, software tools for prototyping
  - examples of real-world applications

# thanks...

More info available at http://staff.cch.kcl.ac.uk/~mpasin/