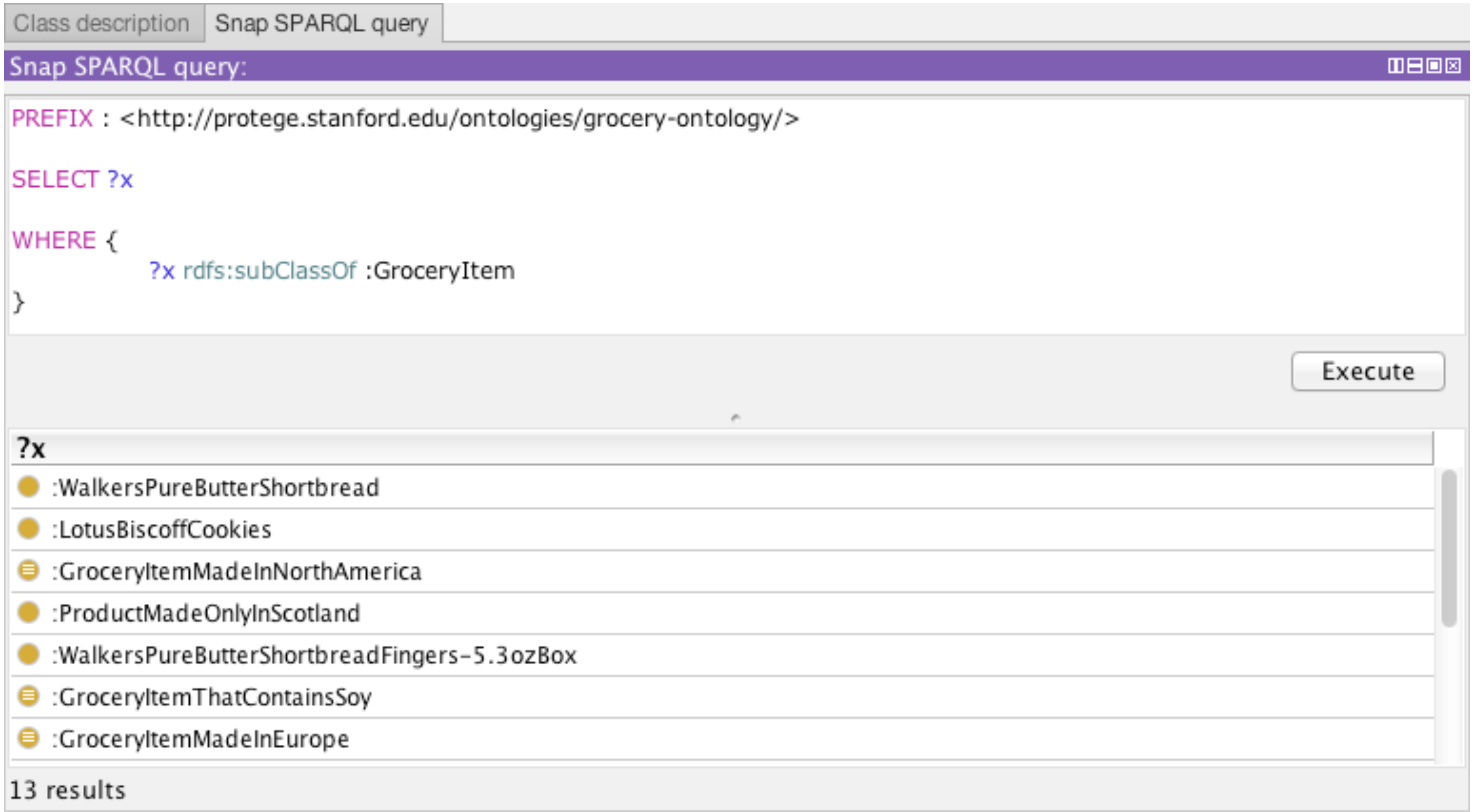
Querying with SPARQL

Protégé Short Course
9 – 11 October, 2017

An Example SPARQL Query

```
PREFIX : <http://protege.stanford.edu/ontologies/groceries/>

SELECT ?x

WHERE {
    ?x rdfs:subClassOf :GroceryItem
}
```

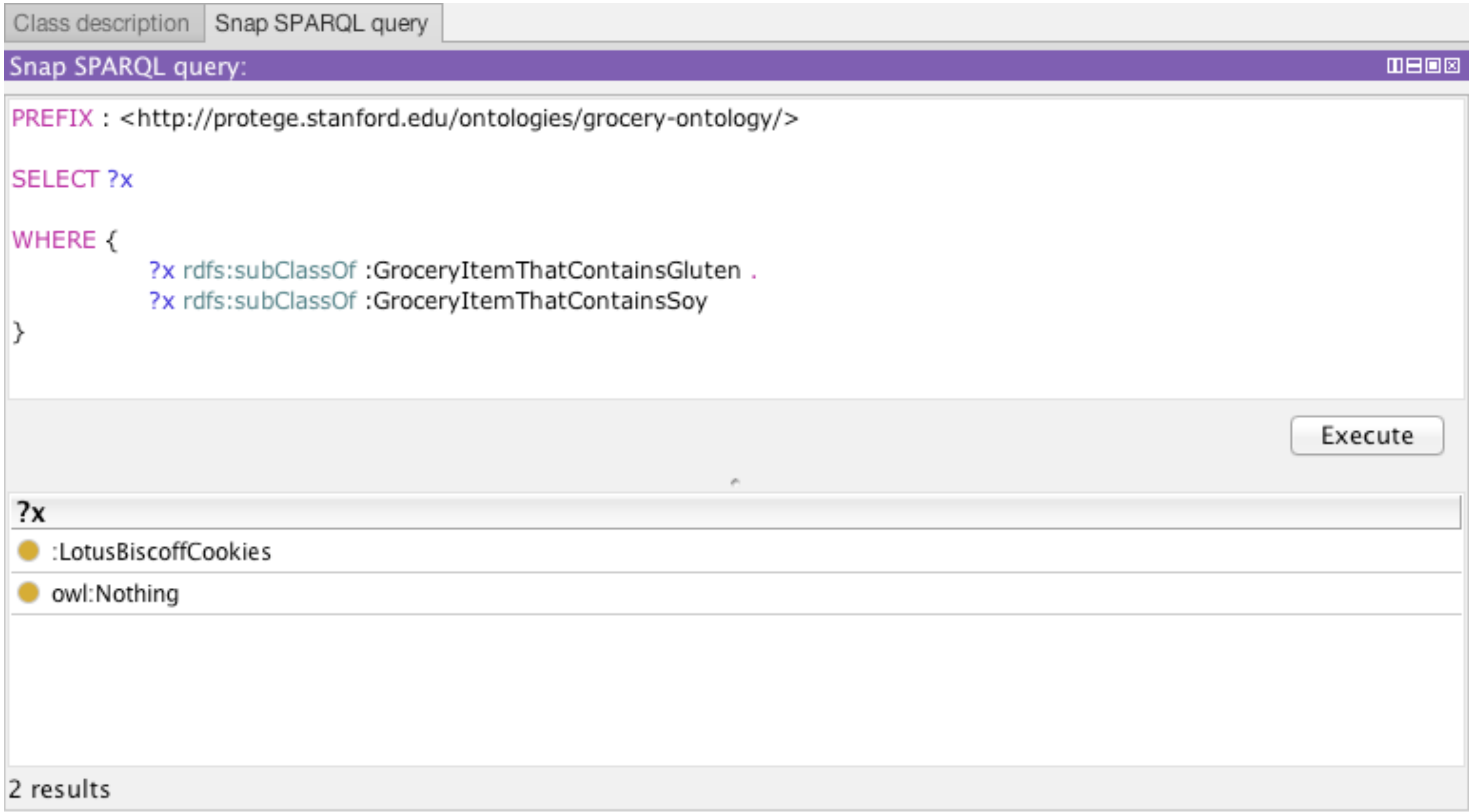


```
PREFIX : <http://protege.stanford.edu/ontologies/grocery-ontology/>

SELECT ?x

WHERE {
            ?x rdfs:subClassOf :GroceryItem
}
```

```
?x
:WalkersPureButterShortbread
:LotusBiscoffCookies
:GroceryItemMadeInNorthAmerica
:ProductMadeOnlyInScotland
:WalkersPureButterShortbreadFingers-5.3ozBox
:GroceryItemThatContainsSoy
:GroceryItemMadeInEurope
13 results
```

An Example SPARQL Query

```
PREFIX : <http://protege.stanford.edu/ontologies/groceries/>

SELECT ?x

WHERE {
      ?x rdfs:subClassOf :GroceryItemThatContainsGluten .
      ?x rdfs:subClassOf :GroceryItemThatContainsSoy
}
```



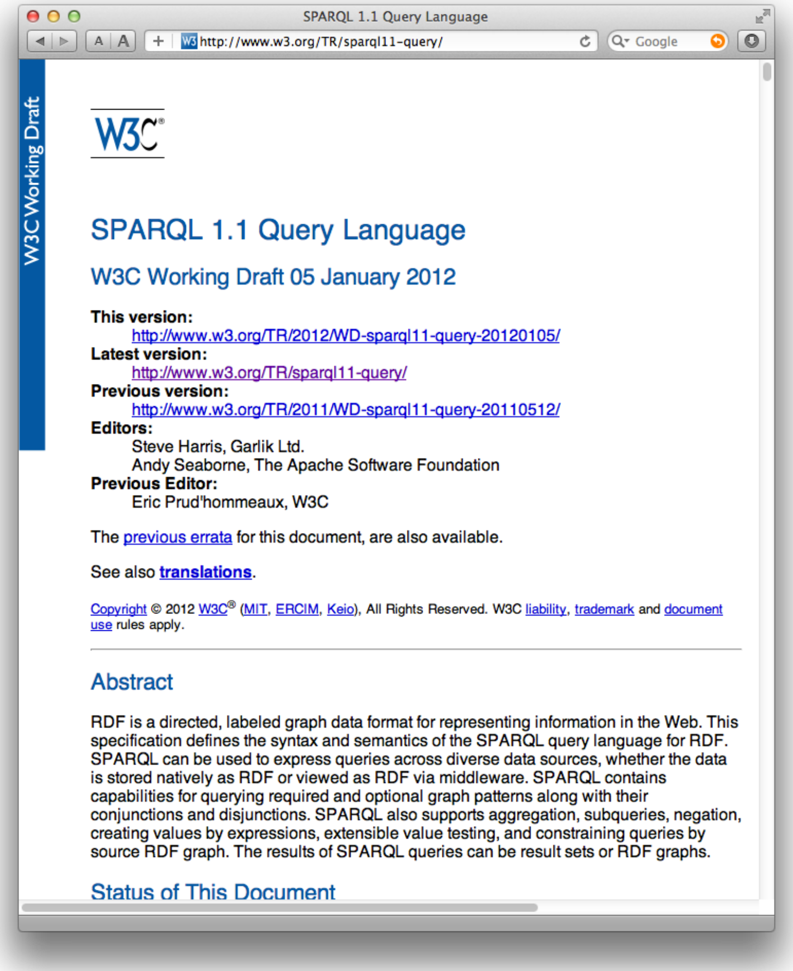
```
PREFIX : <http://protege.stanford.edu/ontologies/grocery-ontology/>

SELECT ?x

WHERE {
            ?x rdfs:subClassOf :GroceryItemThatContainsGluten .
            ?x rdfs:subClassOf :GroceryItemThatContainsSoy
}
```

?x
● :LotusBiscoffCookies
● owl:Nothing

2 results

SPARQL

SPARQL Protocol And RDF Query Language


SPARQL 1.1 Query Language
http://www.w3.org/TR/sparql11-query/
Google
W3C Working Draft

SPARQL 1.1 Query Language

W3C Working Draft 05 January 2012

This version:
http://www.w3.org/TR/2012/WD-sparql11-query-20120105/
Latest version:
http://www.w3.org/TR/sparql11-query/
Previous version:
http://www.w3.org/TR/2011/WD-sparql11-query-20110512/
Editors:
Steve Harris, Garlik Ltd.
Andy Seaborne, The Apache Software Foundation
Previous Editor:
Eric Prud'hommeaux, W3C

The previous errata for this document, are also available.

See also **translations**.

Abstract

RDF is a directed, labeled graph data format for representing information in the Web. This specification defines the syntax and semantics of the SPARQL query language for RDF. SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports aggregation, subqueries, negation, creating values by expressions, extensible value testing, and constraining queries by source RDF graph. The results of SPARQL queries can be result sets or RDF graphs.

Status of This Document

http://www.w3.org/TR/sparql11-query/

SPARQL

Primarily designed for querying RDF Graphs

... however, the **kinds** of queries that can be asked are applicable to OWL ontologies...

.... it's a **W3C Standard**

RDF

The Resource Description Framework

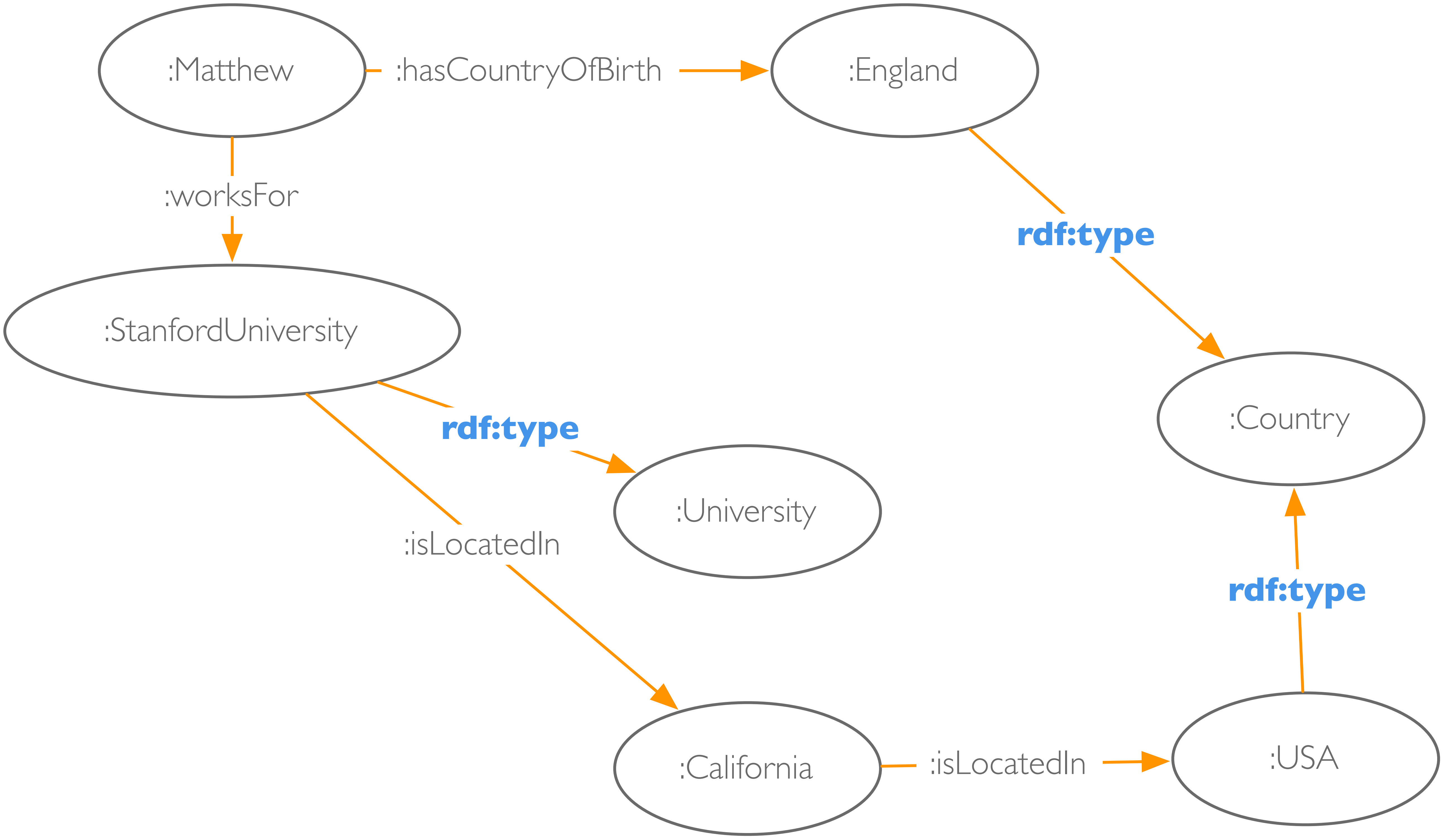
For describing properties of resources on the web

An RDF Graph is a set of statements - Triples

Subject - Predicate - Object

RDF Graph Example

:Matthew	:hasCountryOfBirth	:England
:England	rdf:type	:Country
:Matthew	:worksFor	:StanfordUniversity
:StanfordUniversity	rdf:type	:University
:StanfordUniversity	:locatedIn	:California
:California	:locatedIn	:USA


:Matthew
:hasCountryOfBirth
:England
:worksFor
rdf:type
:StanfordUniversity
:Country
rdf:type
:University
:isLocatedIn
rdf:type
:California
:isLocatedIn
:USA

An Example SPARQL Query

```
PREFIX : <http://protege.stanford.edu/rdf/employees#>

SELECT ?x ?y

WHERE {
      ?x :locatedIn ?y .
}
```

:StanfordUniversity — :locatedIn — :California

:California — :locatedIn — :USA

?x	?y
:StanfordUniversity	:California
:California	:USA

rdf:type

(built in vocabulary)

:Matthew	:hasCountryOfBirth	:England
:England	**rdf:type**	:Country
:Matthew	:worksFor	:StanfordUniversity
:StanfordUniversity	**rdf:type**	:University
:StanfordUniversity	:locatedIn	:California
:California	:locatedIn	:USA

RDF and Entailment

```
PREFIX : <http://protege.stanford.edu/rdf/employees#>

SELECT ?x ?y

WHERE {
      ?x :locatedIn ?y .
}
```

?x	?y
:StanfordUniversity	:California
:California	:USA

Not entailed!

?x	?y
:StanfordUniversity	:USA

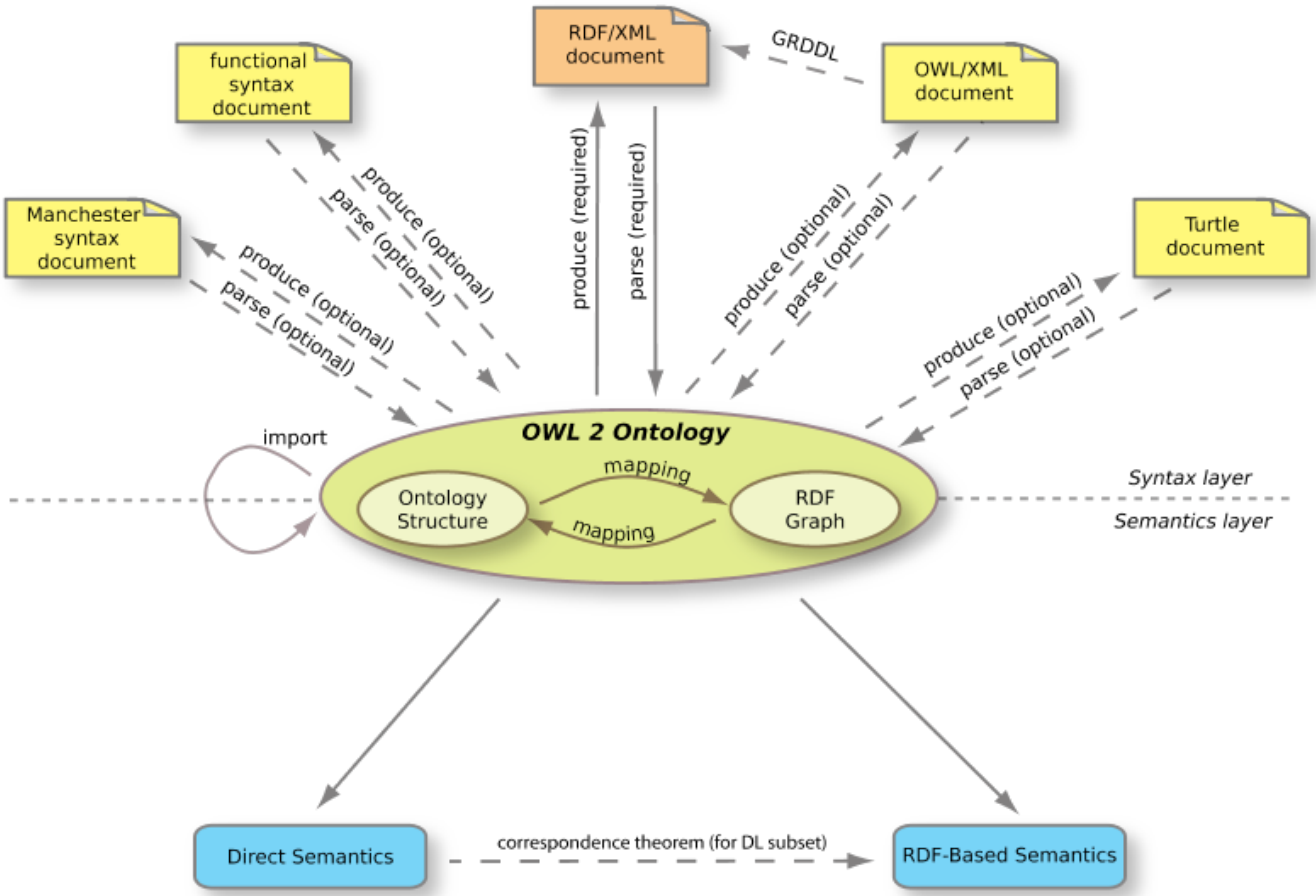
(Can't say locatedIn is transitive in RDF)

What is RDF good for?

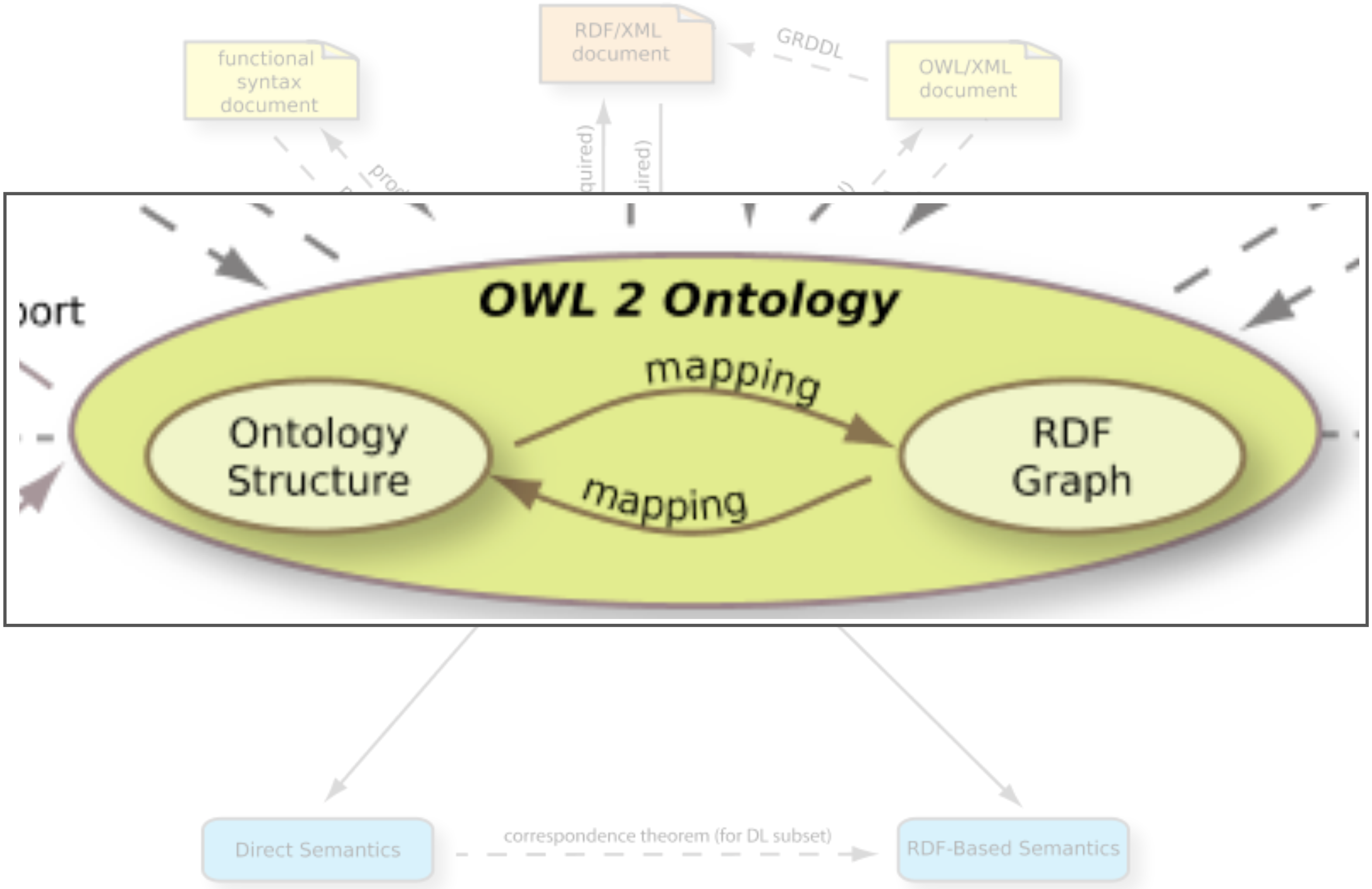
Storing (semi-structured) data - facts about things

Off-the-shelf tools for storing and querying with standard query languages (SPARQL)

SPARQL and OWL



SPARQL and OWL



Mapping an OWL Ontology into an RDF Graph

(Example I)

WheatFlour SubClassOf Flour

↓

```
<http://protege.stanford.edu/ontologies/groceries/WheatFlour>
<http://www.w3.org/2000/01/rdf-schema#subClassOf>
<http://protege.stanford.edu/ontologies/groceries/Flour> .
```

↓

```
PREFIX : <http://protege.stanford.edu/ontologies/groceries/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

    :WheatFlour rdfs:subClassOf :Flour .
```

Mapping an OWL Ontology into an RDF Graph

(Example II)

hasIngredient Domain FoodStuff

↓ (Assuming prefixes have been specified)

:hasIngredient rdfs:domain :FoodStuff

Mapping an OWL Ontology into an RDF Graph

(Example III)

Shortbread SubClassOf hasIngredient some Butter

↓ (Assuming prefixes have been specified)

```
:Shortbread rdfs:subClassOf _:x

_:x rdf:type owl:Restriction

_:x owl:onProperty :hasIngredient

_:x owl:someValuesFrom :Butter
```

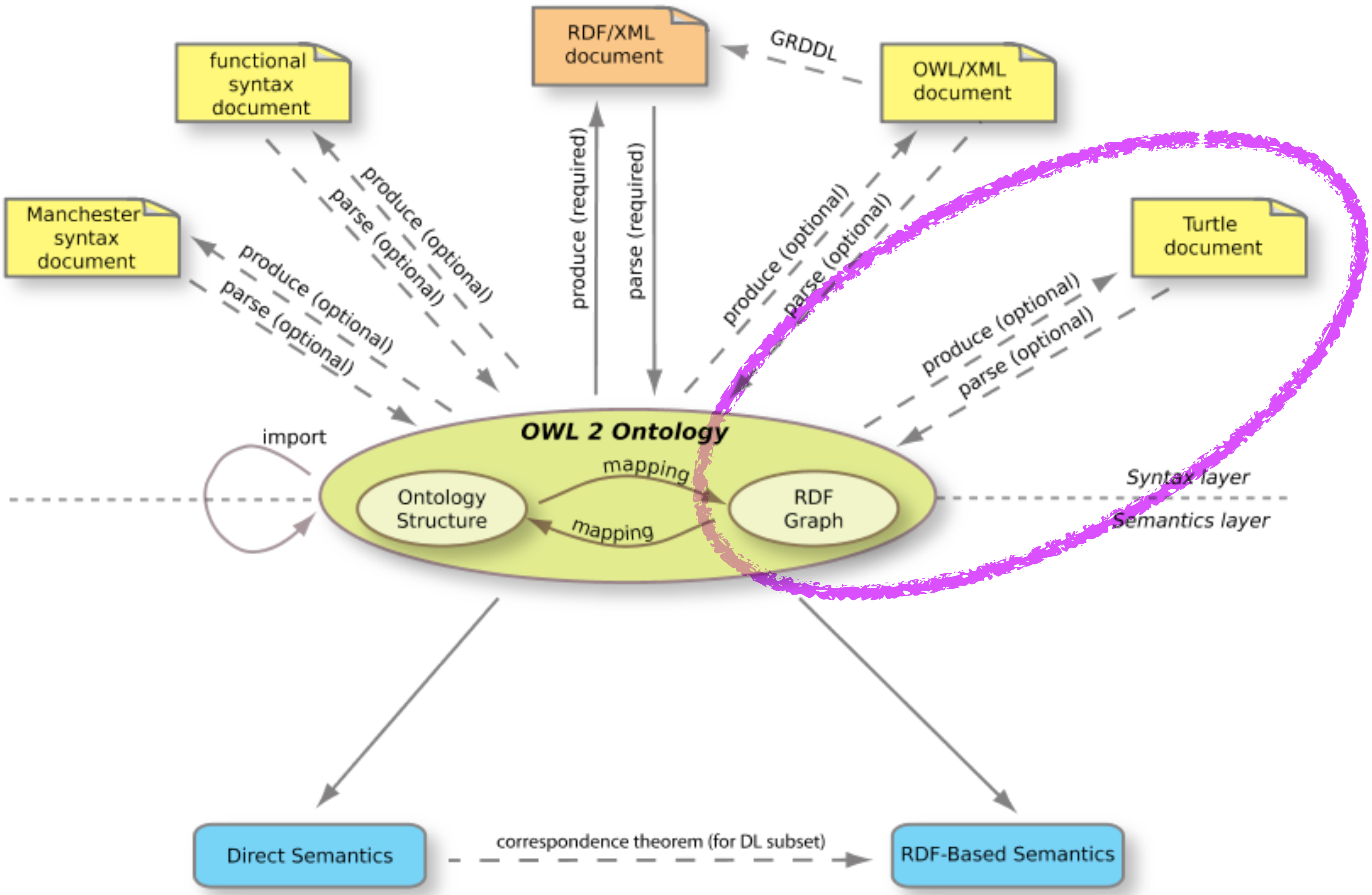
Mapping an OWL Ontology into an RDF Graph

(Example IV)

BiscoffCookie SubClassOf hasIngredient some (CanolaOil or PalmOil)

```
:BiscoffCookie rdfs:subClassOf _:x1 .
        _:x1 owl:onProperty :hasIngredient .
        _:x1 owl:someValuesFrom _:x2 .
                _:x2 owl:unionOf _:x3 .
                        _:x3 rdf:type rdf:List .
                        _:x3 rdf:first :CanolaOil .
                        _:x3 rdf:rest _:x4 .
                             _:x4 rdf:type rdf:List .
                             _:x4 rdf:first :PalmOil .
                             _:x4 rdf:rest rdf:nill.
                _:x2 rdf:type owl:Class .
        _:x1 rdf:type owl:Restriction .
```

SPARQL Syntax



SPARQL Graph Pattern Syntax

(Turtle Syntax)

```
?x rdfs:subClassOf :GroceryItemThatContainsGluten .
?x rdfs:subClassOf :HighFatGroceryItem
```

SPARQL Graph Pattern Syntax
(Turtle Syntax)

```
?x rdfs:subClassOf :GroceryItemThatContainsGluten .
?x rdfs:subClassOf :HighFatGroceryItem
```

Don’t forget the dot between multiple triples!

Anatomy of a
(simple) SPARQL Query

```
PREFIX : <http://protege.stanford.edu/ontologies/groceries/>

SELECT ?x

WHERE {
      ?x rdfs:subClassOf :GroceryItemThatContainsSoy .
      ?x rdfs:subClassOf :GroceryItemThatIsHighInSaturatedFat
}
```

Anatomy of a (simple) SPARQL Query

```
PREFIX : <http://protege.stanford.edu/ontologies/groceries/>

SELECT ?x

WHERE {
      ?x rdfs:subClassOf :GroceryItemThatContainsSoy .
      ?x rdfs:subClassOf :GroceryItemThatIsHighInSaturatedFat
}
```

PREFIX declaration (part of prologue)
BASE declarations are also possible here

Anatomy of a (simple) SPARQL Query

```
PREFIX : <http://protege.stanford.edu/ontologies/groceries/>

SELECT ?x

WHERE {
      ?x rdfs:subClassOf :GroceryItemThatContainsSoy .
      ?x rdfs:subClassOf :GroceryItemThatIsHighInSaturatedFat
}
```

SELECT keyword (query form).

Other possibilities include ASK, DESCRIBE, CONSTRUCT etc.

Variable name to show in results list

Anatomy of a (simple) SPARQL Query

```
PREFIX : <http://protege.stanford.edu/ontologies/groceries/>

SELECT ?x

WHERE {
      ?x rdfs:subClassOf :GroceryItemThatContainsSoy .
      ?x rdfs:subClassOf :GroceryItemThatIsHighInSaturatedFat
}
```

Basic Graph Pattern
(Specifies axiom template to match)

WHERE keyword

Some SPARQL Queries

(which make sense for OWL)

A few examples:

“Give me all foods that are DairyProducts”

“What is WalkersShortbread a subclass of?”

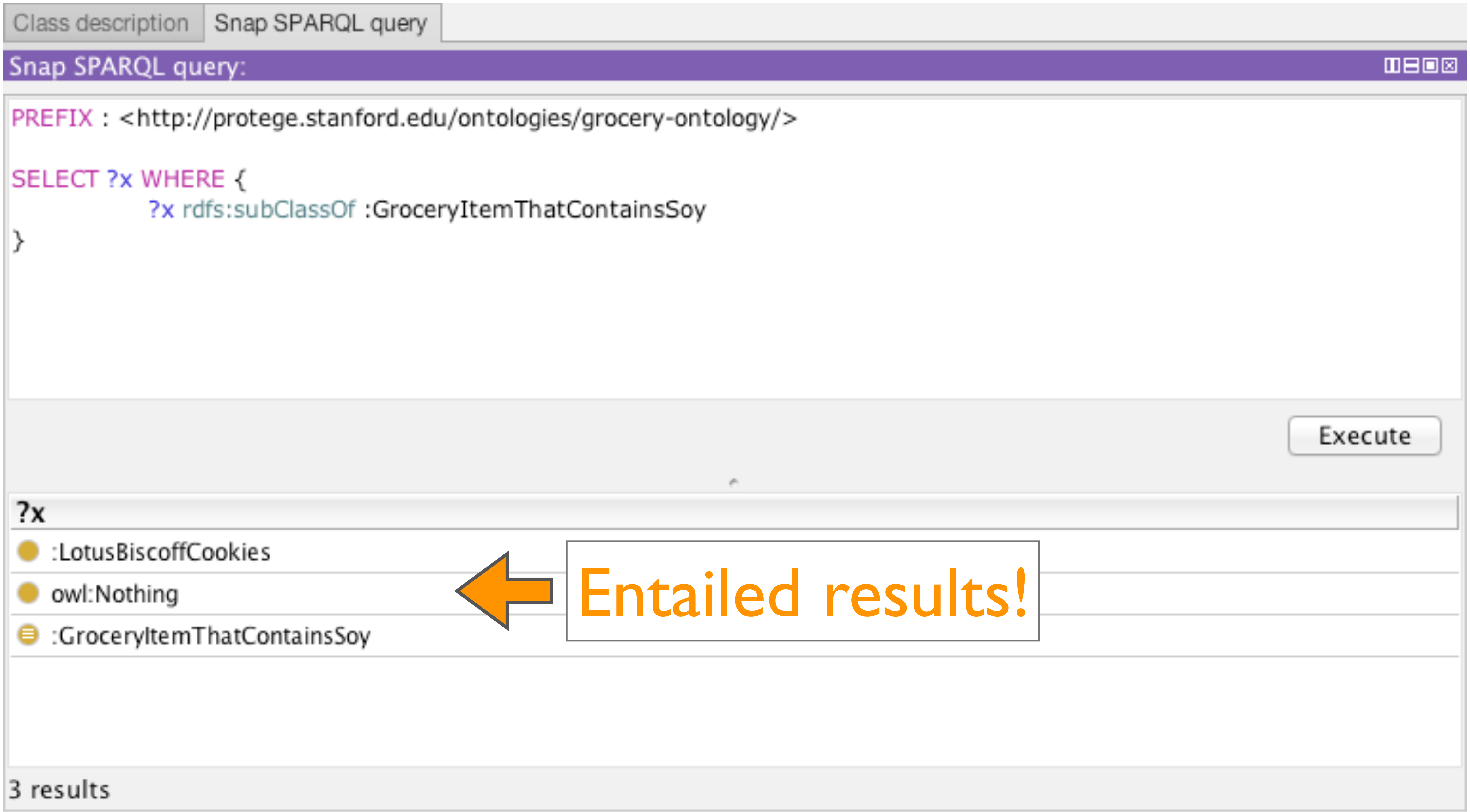
“What are the instances of Country?”

“List places and their locations”

“List classes that have rdfs:comment annotations on them”

SPARQL and Entailment

“Give me all the GroceryItem that contains Soy”

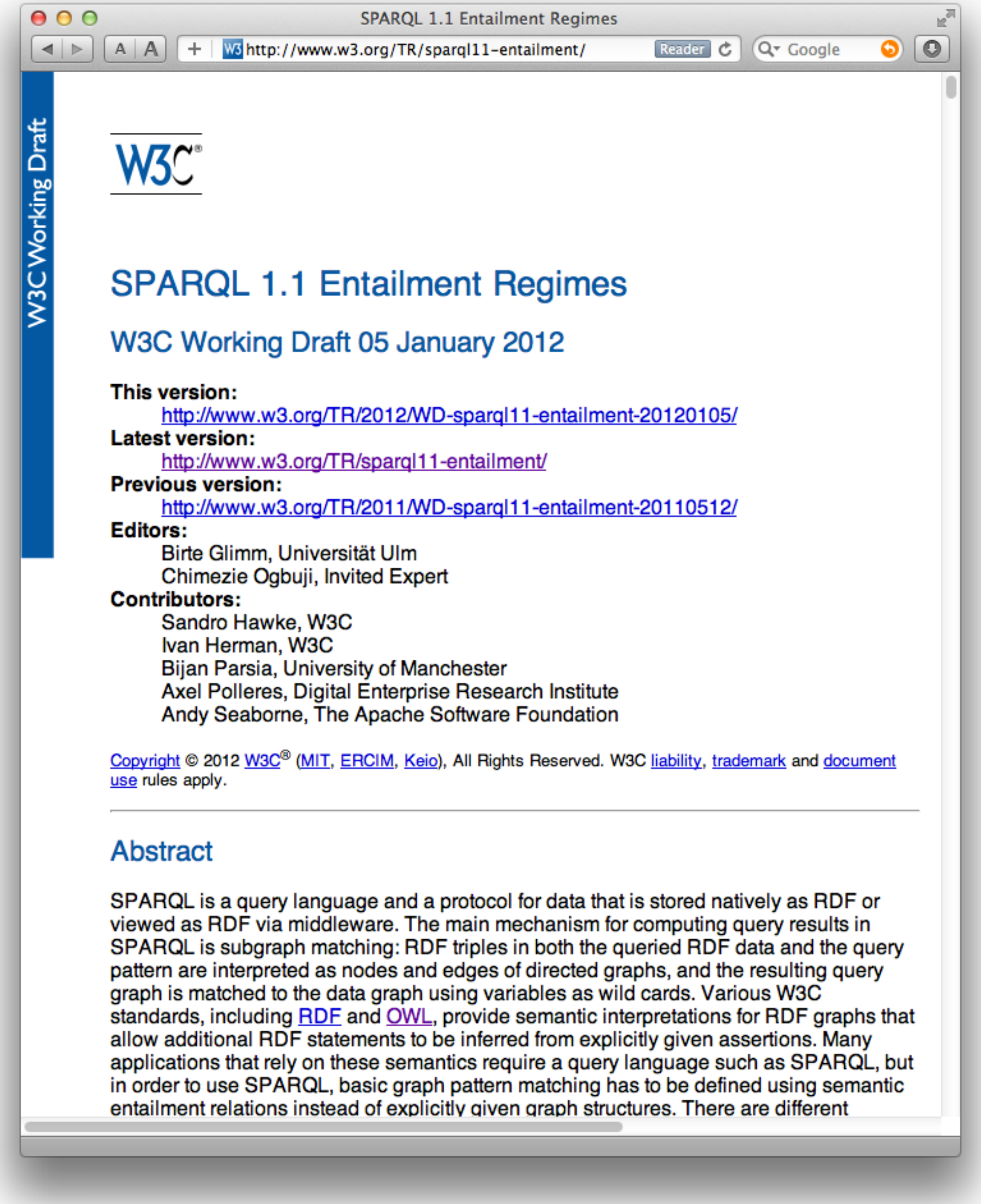


```
PREFIX : <http://protege.stanford.edu/ontologies/grocery-ontology/>

SELECT ?x WHERE {
            ?x rdfs:subClassOf :GroceryItemThatContainsSoy
}
```

```
?x
● :LotusBiscoffCookies
● owl:Nothing
● :GroceryItemThatContainsSoy

3 results
```

Entailed results!

SPARQL Entailment Regimes


SPARQL 1.1 Entailment Regimes
http://www.w3.org/TR/sparql11-entailment/
Reader
Google
W3C Working Draft
W3C

SPARQL 1.1 Entailment Regimes

W3C Working Draft 05 January 2012

This version:
http://www.w3.org/TR/2012/WD-sparql11-entailment-20120105/
Latest version:
http://www.w3.org/TR/sparql11-entailment/
Previous version:
http://www.w3.org/TR/2011/WD-sparql11-entailment-20110512/
Editors:
Birte Glimm, Universität Ulm
Chimezie Ogbuji, Invited Expert
Contributors:
Sandro Hawke, W3C
Ivan Herman, W3C
Bijan Parsia, University of Manchester
Axel Polleres, Digital Enterprise Research Institute
Andy Seaborne, The Apache Software Foundation

Abstract

SPARQL is a query language and a protocol for data that is stored natively as RDF or viewed as RDF via middleware. The main mechanism for computing query results in SPARQL is subgraph matching: RDF triples in both the queried RDF data and the query pattern are interpreted as nodes and edges of directed graphs, and the resulting query graph is matched to the data graph using variables as wild cards. Various W3C standards, including RDF and OWL, provide semantic interpretations for RDF graphs that allow additional RDF statements to be inferred from explicitly given assertions. Many applications that rely on these semantics require a query language such as SPARQL, but in order to use SPARQL, basic graph pattern matching has to be defined using semantic entailment relations instead of explicitly given graph structures. There are different

SPARQL Syntax Issues

```
:MargheritaPizza rdfs:subClassOf _:x1 .
_:x1 owl:onProperty :hasTopping .
_:x1 owl:allValuesFrom _:x2 .
                _:x2 owl:unionOf _:x3 .
                                _:x3 rdf:type rdf:List .
                              _:x3 rdf:first :MozzarellaTopping .
                                _:x3 rdf:rest _:x4 .
                                        _:x4 rdf:type rdf:List .
                                        _:x4 rdf:first :TomatoTopping .
                                        _:x4 rdf:rest rdf:nill.
                _:x2 rdf:type owl:Class .
_:x1 rdf:type owl:Restriction .
```

From this to this

```
:MargheritaPizza rdfs:subClassOf [ owl:onProperty :hasTopping ;
                                   owl:allValuesFrom
                                         [ owl:unionOf
                                                (:MozzarellaTopping
                                                 :TomatoTopping) ;
                                            rdf:type owl:Class
                                         ] ;
                                   rdf:type owl:Restriction
                                  ]
```

SPARQL + Manchester Syntax
(Terp)

Pure Turtle

```
:MargheritaPizza rdfs:subClassOf [ owl:onProperty :hasTopping ;
                                   owl:allValuesFrom
                                       [ owl:unionOf
                                             (:MozzarellaTopping
                                              :TomatoTopping) ;
                                         rdf:type owl:Class
                                       ] ;
                                   rdf:type owl:Restriction
                                  ]
```

Turtle + Manchester Syntax (for class expressions)

```
PREFIX : <http://protege.stanford.edu/ontologies/pizza#>

SELECT ?x

WHERE {
      ?x rdfs:subClassOf (hasTopping some (MozzarellaTopping or TomatoTopping) .
}
```

SPARQL + Manchester Syntax

(Terp)

```
PREFIX : <http://protege.stanford.edu/ontologies/pizza#>

SELECT ?x

WHERE {
        ?x rdfs:subClassOf (hasTopping some ?y) .
}
```

Turtle + Manchester Syntax with variables!

SPARQL Summary

Certain classes of SPARQL queries make sense in the context of OWL

SPARQL can be used to write expressive queries for OWL ontologies
(multiple variables in most positions are allowed)

SPARQL 1.1 has an entailment regimes specification, which makes it clear how to use reasoning when answering SPARQL queries

Native SPARQL syntax is based on Turtle, which can be ugly for complex class expressions in OWL, but for small queries is perfectly readable

Where do I store my ontology for others to query?

Triple Stores

Purpose-built data stores for RDF graphs…

…and for retrieving RDF data through SPARQL

When to use a triple store

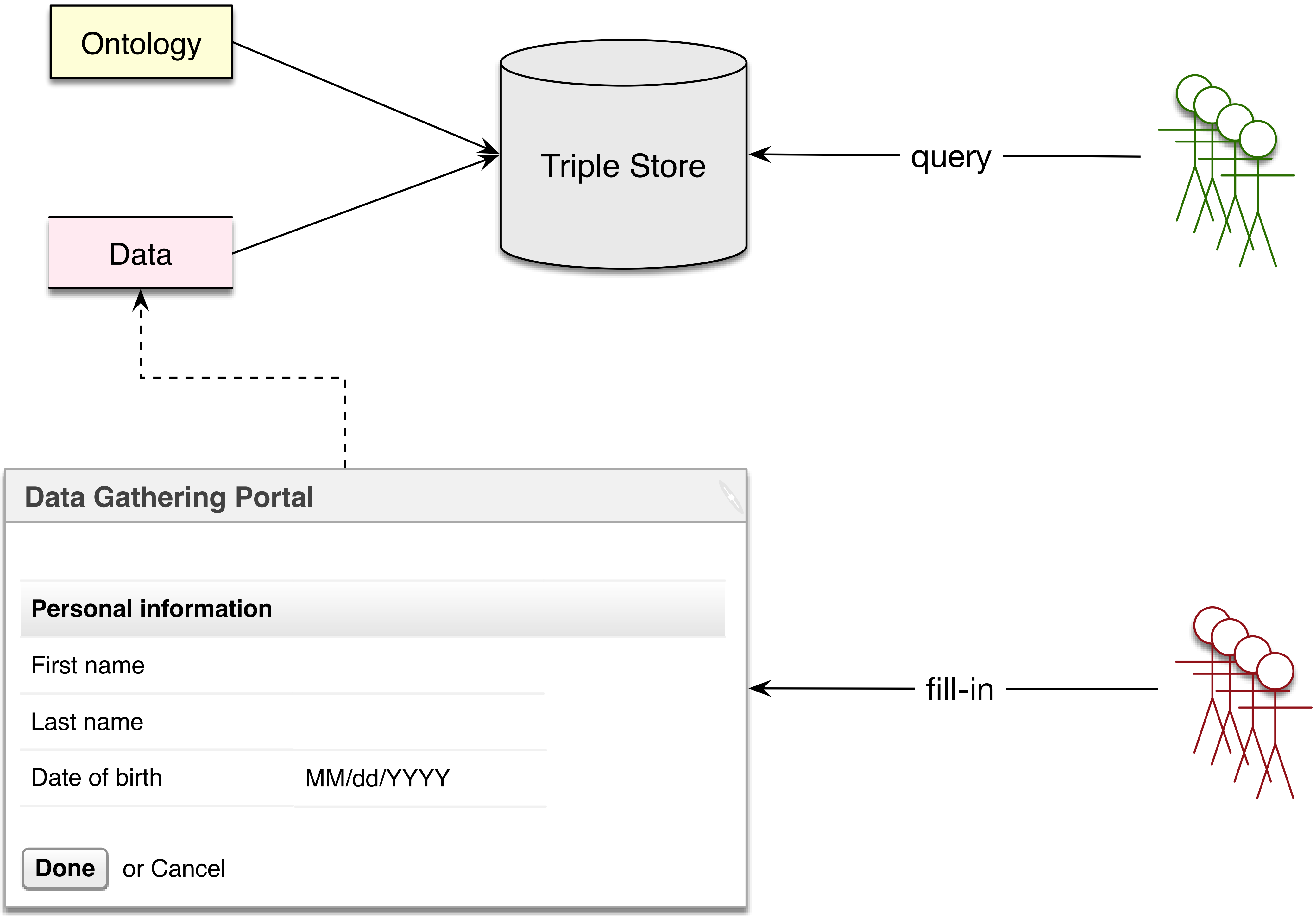
W3C standards: RDF, RDFS, SPARQL, OWL, …

Portability

Fast query answering over RDF using SPARQL

Reasoning support

Example Usage Scenario



Data Gathering Portal

Personal information

First name

Last name

Date of birth MM/dd/YYYY

Done or Cancel

Choosing a triple store

How expressive is the ontology?

(i.e., do I have to compromise the "truth")

What kind of reasoning is needed?

Does the ontology change frequently?

What infrastructure is available?

(memory size vs high clock speed (distributed) machine setup)

Triple Stores Summary

Triple stores excel at storing and querying data that can be represented in RDF (e.g., OWL ontologies)

Provide some reasoning support

Versatile

Based on W3C standards