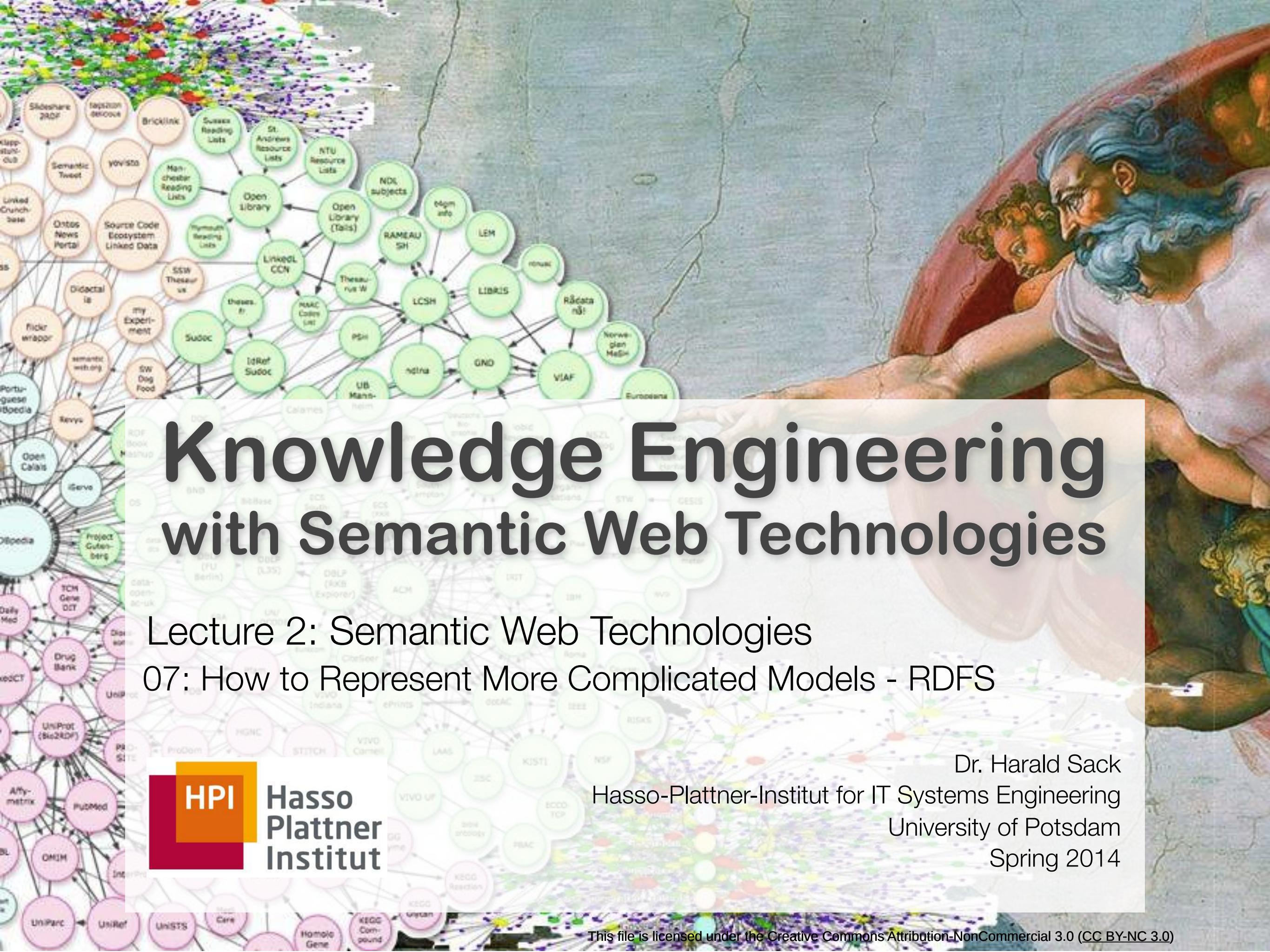




Knowledge Engineering with Semantic Web Technologies

Lecture 2: Semantic Web Technologies

07: How to Represent More Complicated Models - RDFS



Dr. Harald Sack
Hasso-Plattner-Institut for IT Systems Engineering
University of Potsdam
Spring 2014



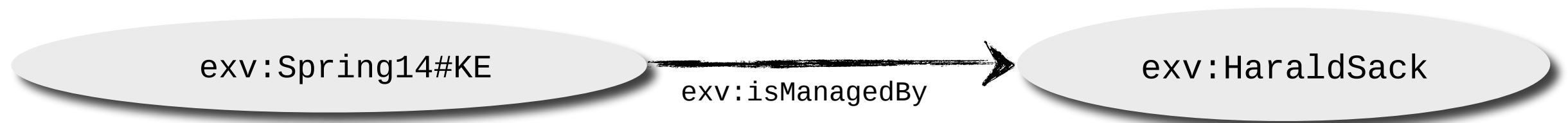
Lecture 2: Semantic Web Technologies - Part 1

Open HPI - Course: Knowledge Engineering with Semantic Web Technologies

What about the Meaning?

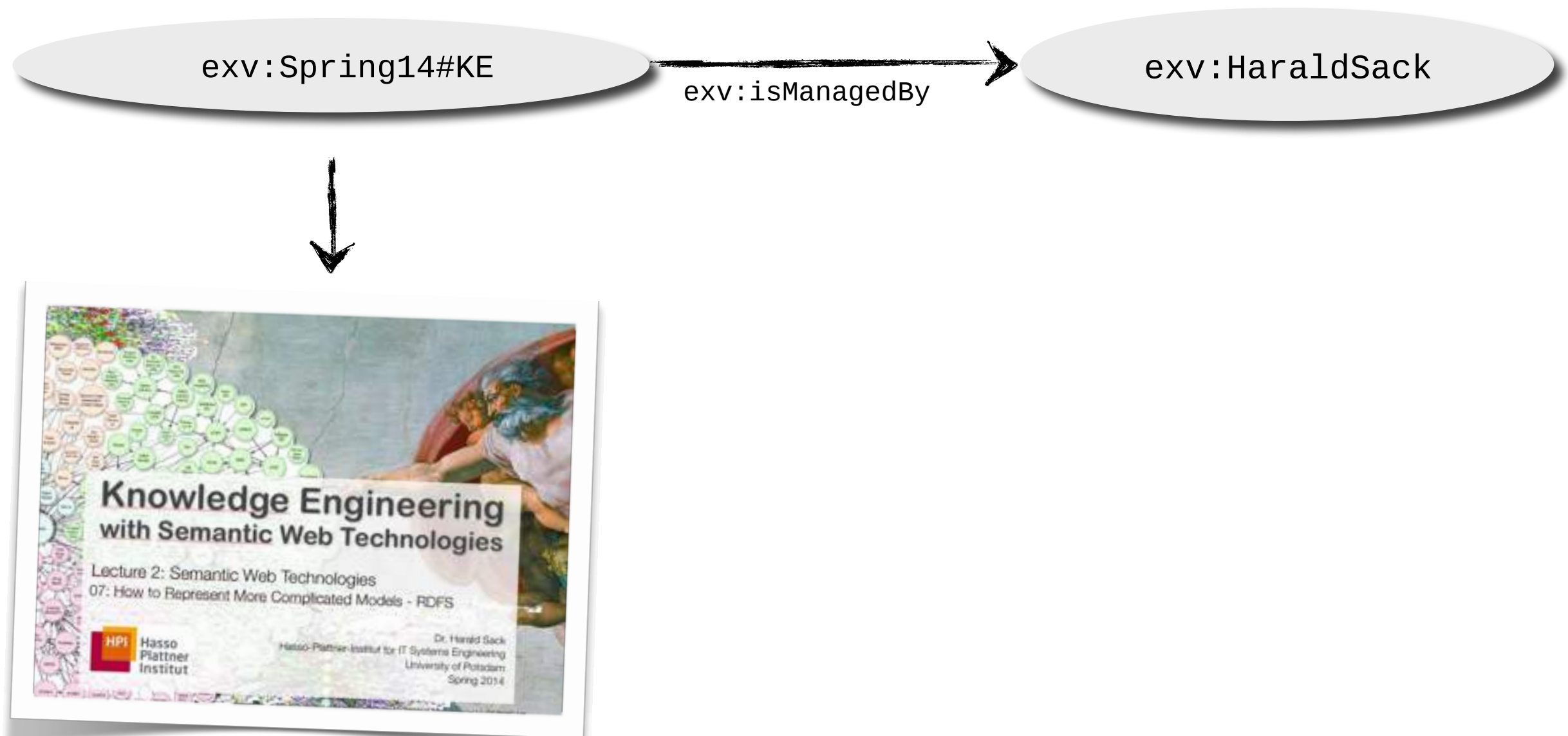
3

RDF



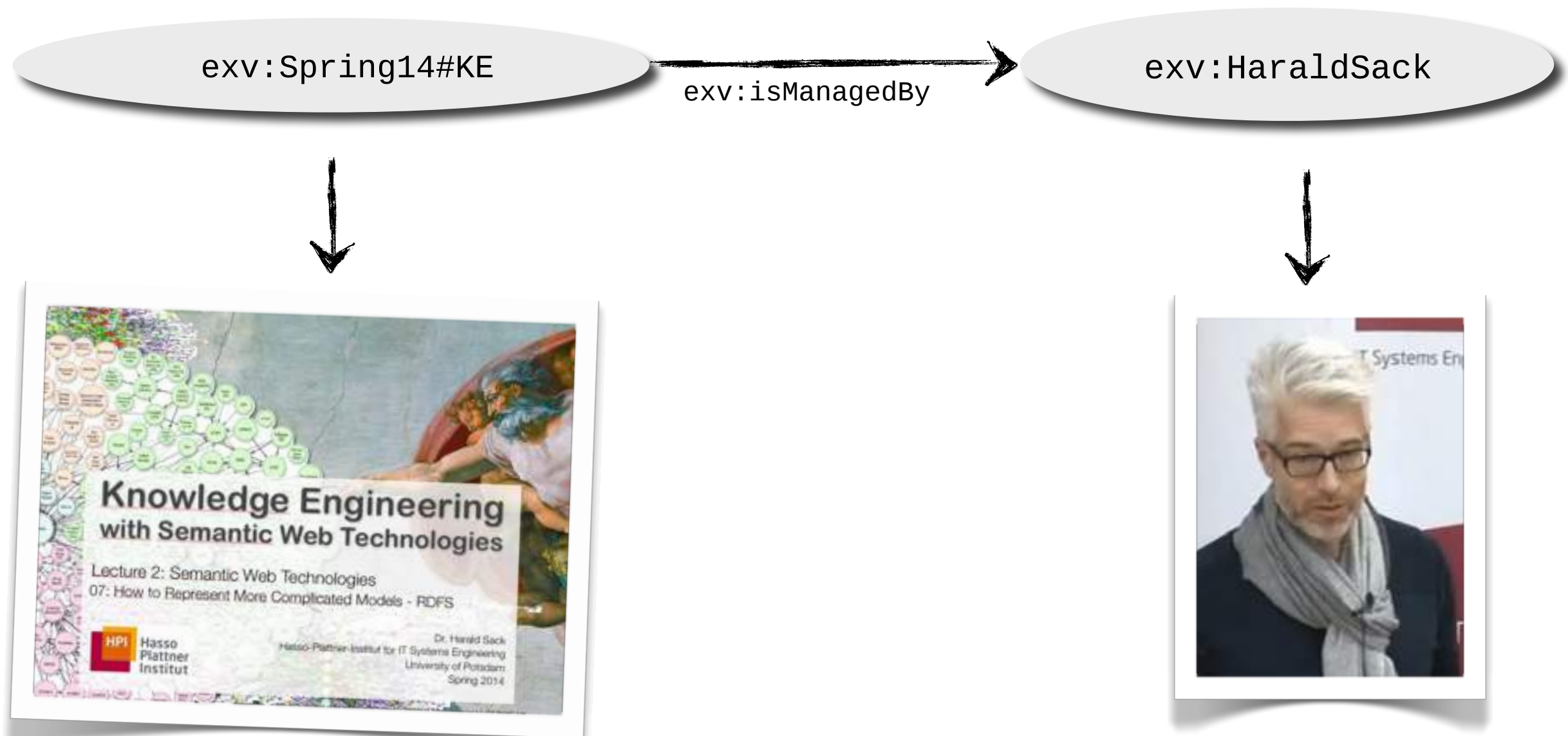
What about the Meaning?

RDF



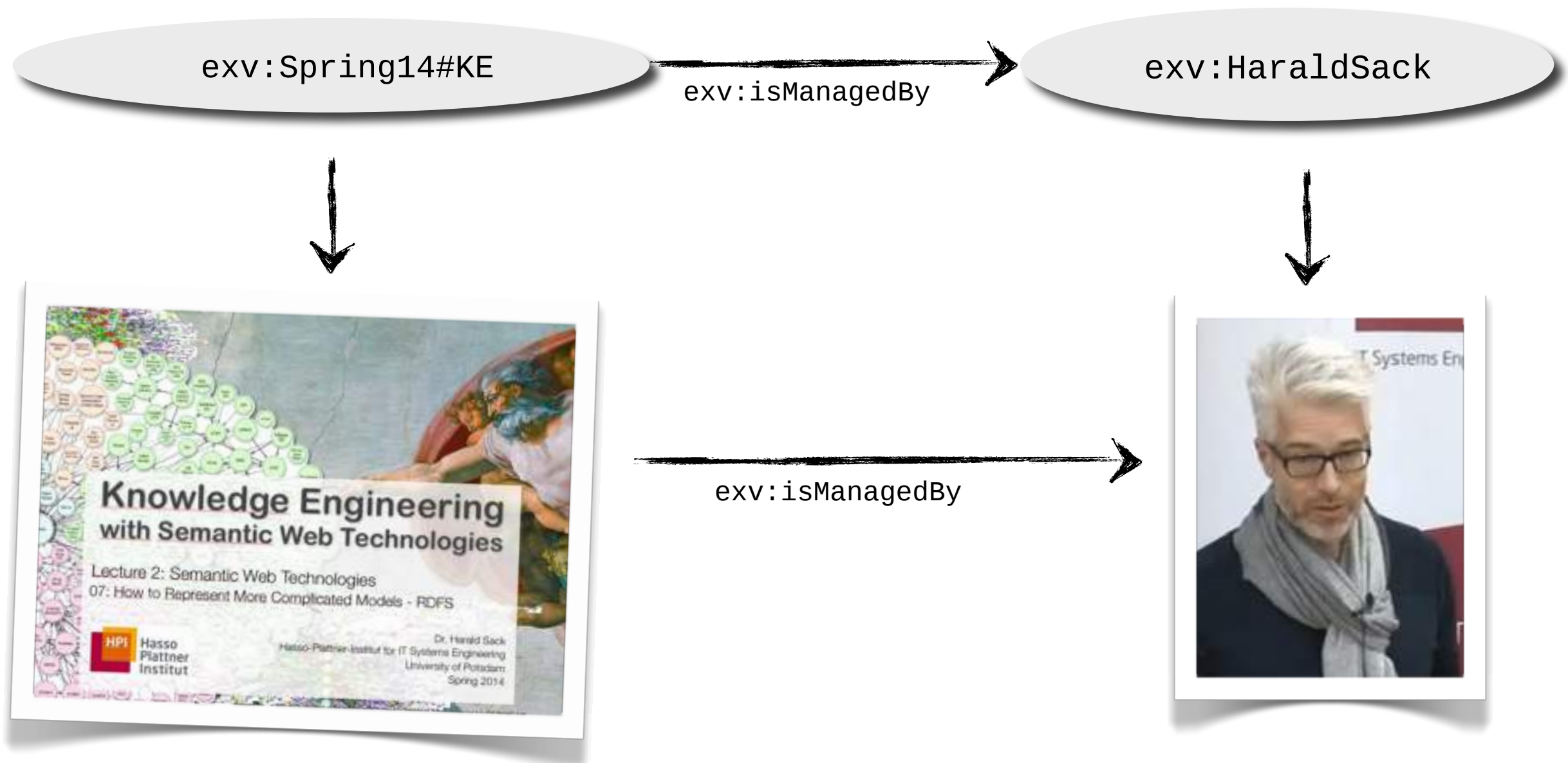
What about the Meaning?

RDF



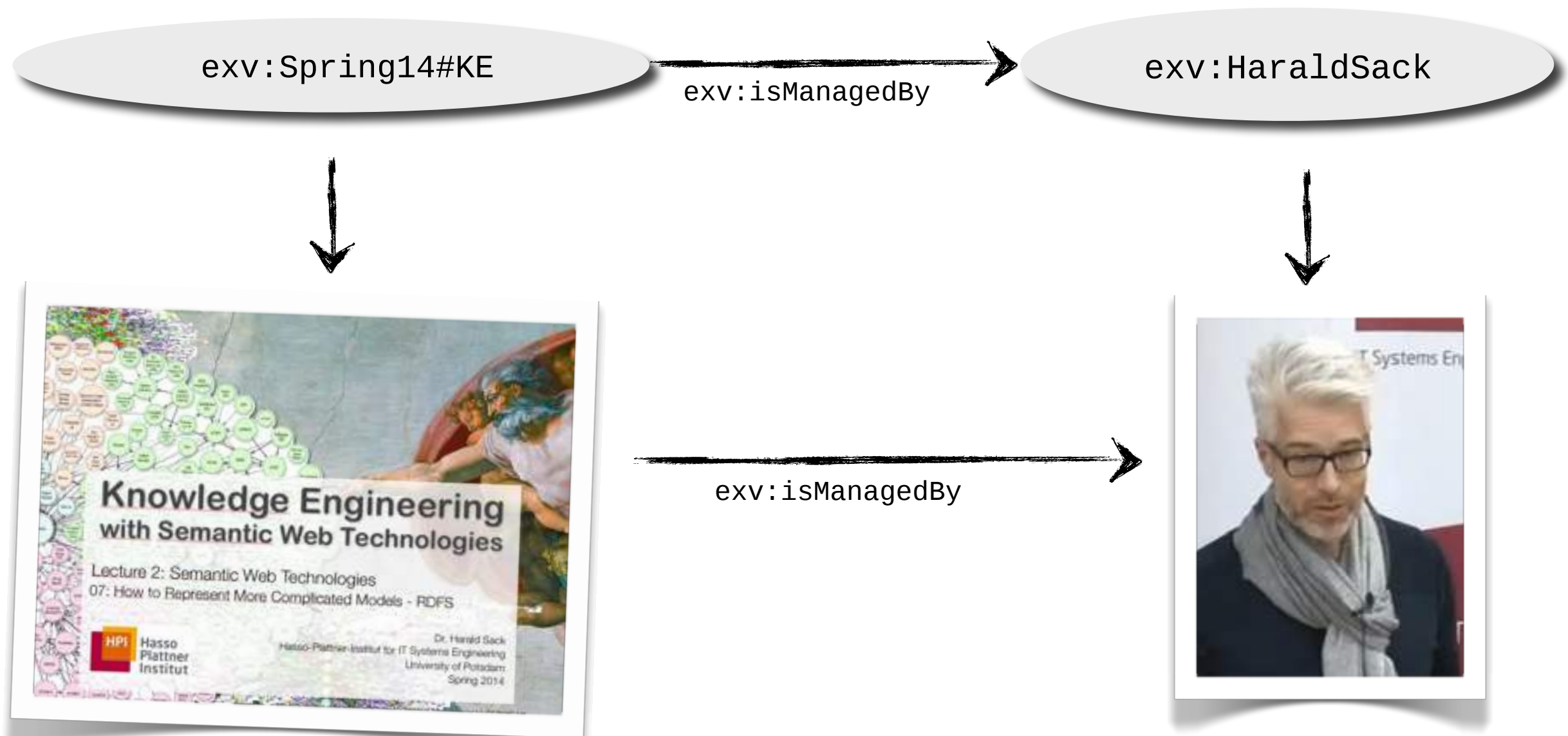
What about the Meaning?

RDF



What about the Meaning?

RDF

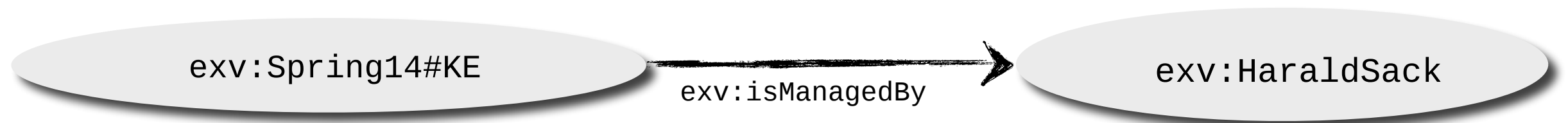


Where does the intended Meaning come from?

What about the Meaning?

4

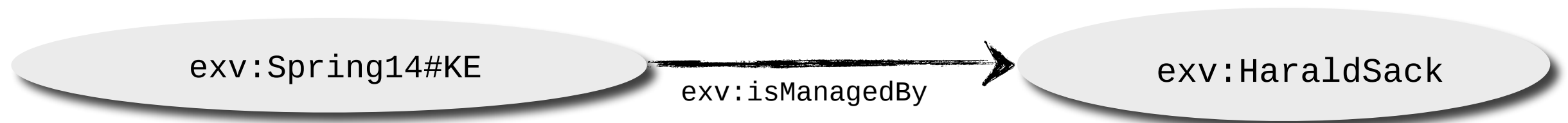
RDF



What about the Meaning?

4

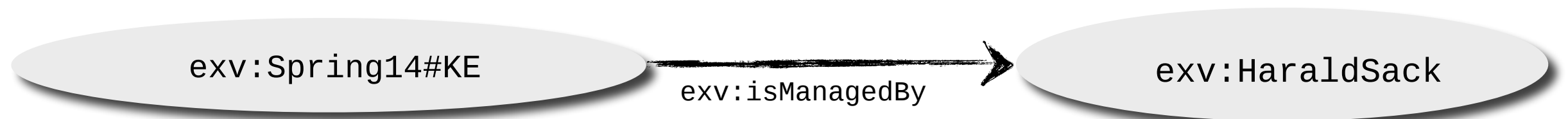
RDF



What about the Meaning?

4

RDF

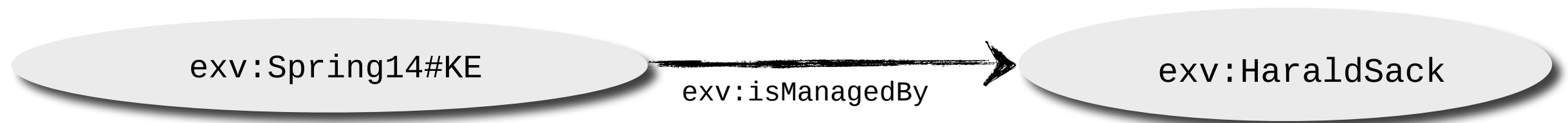


*something from the **namespace** **exv:**
with the **URI** **Spring14#KE***

What about the Meaning?

4

RDF



*something from the **namespace** **exv:**
with the **URI Spring14#KE***

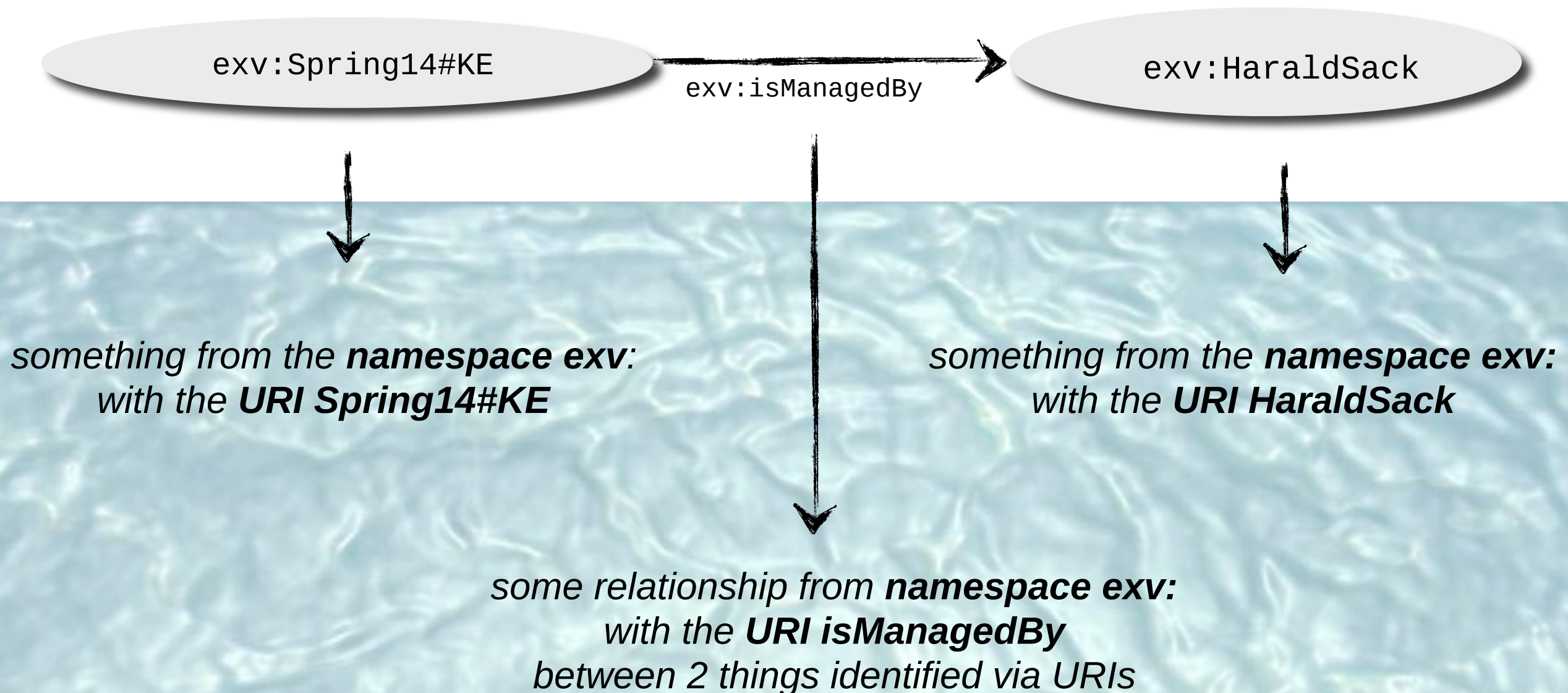


*something from the **namespace** **exv:**
with the **URI HaraldSack***

What about the Meaning?

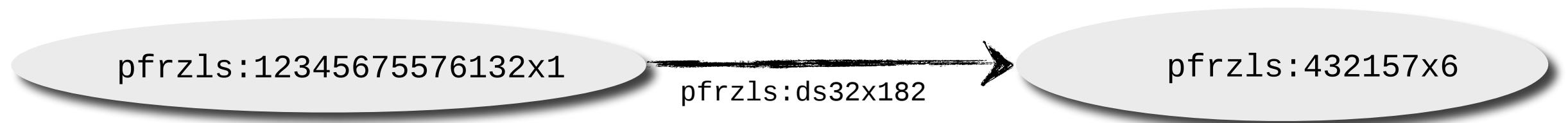
4

RDF



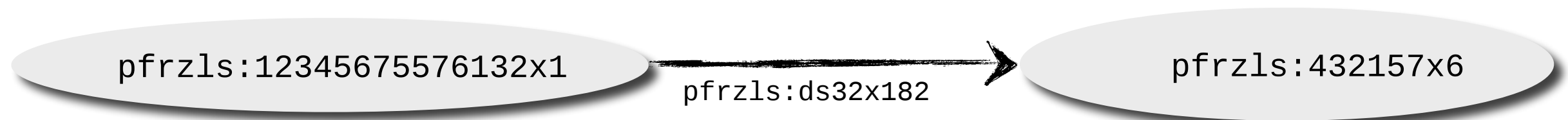
What about the Meaning?

RDF



What about the Meaning?

RDF



Probably we need a little
bit more to make something
out of it....

07 How to represent more complicated models? - RDFS

Open HPI - Course: Knowledge Engineering with Semantic Web Technologies
Lecture 2: Semantic Web Technologies Part I



**SO ... WE NEED A WAY TO
DEFINE RDF SCHEMATA FOR
CLASSES AND RELATIONS ?!**

2. Semantic Web Technologies

Part 1

The Semantic Web Technology Stack (not a piece of cake...)

Most apps use only a subset of the stack

Querying allows fine-grained data access

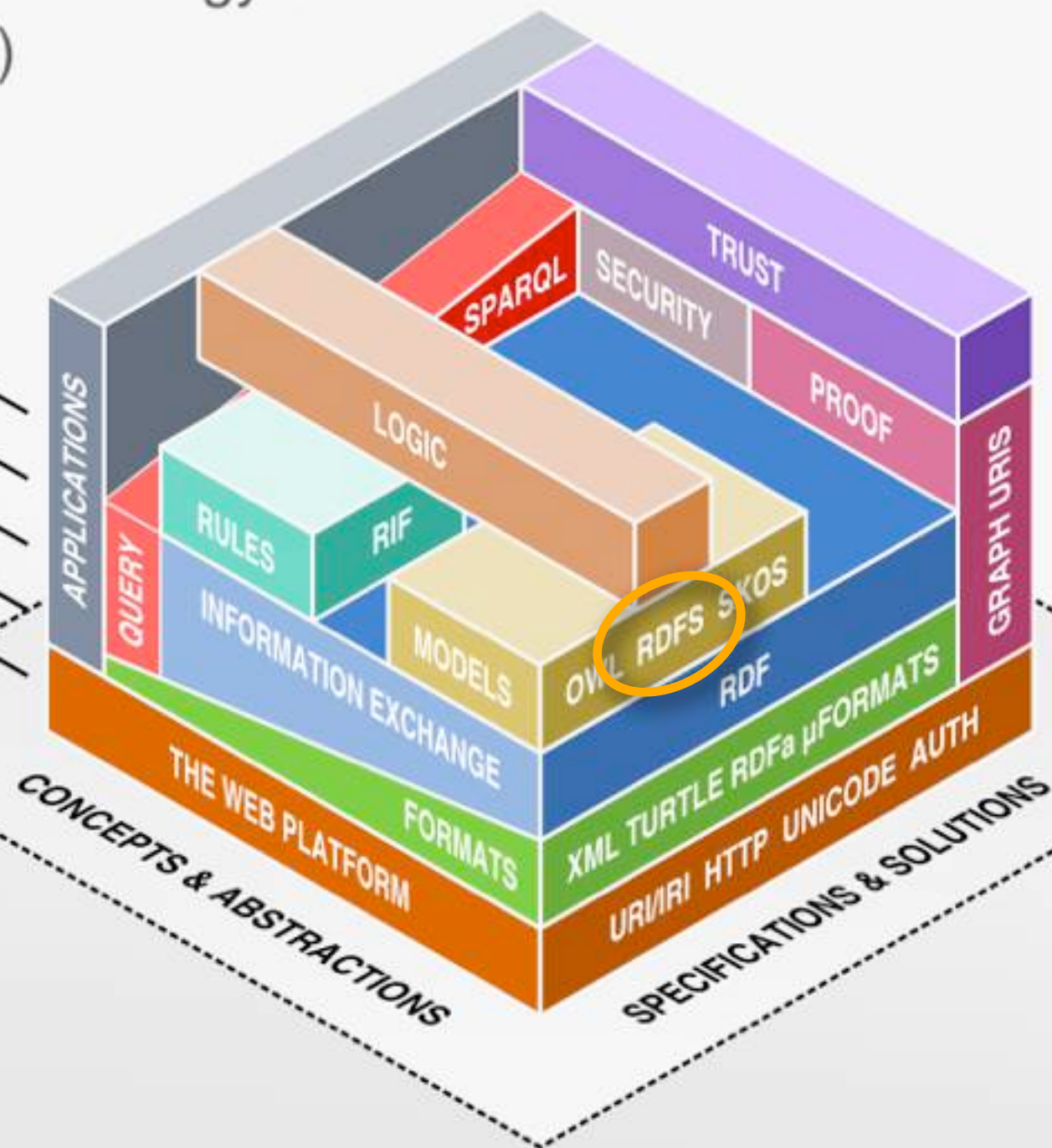
Standardized information exchange is key

Formats are necessary, but not too important

The Semantic Web is based on the Web

Linked Data uses a small
selection of technologies

LINKED DATA



Why RDFSchema...?

Why RDFSchema...?

- An **application program** can define and use **RDF data**

Why RDFSchema...?

- An **application program** can define and use **RDF data**
-if the application program knows, **which terms and classes to use**, as e.g.

Why RDFSchema...?

- An **application program** can define and use **RDF data**
-if the application program knows, **which terms and classes to use**, as e.g.
 - name, title, year, ...

Why RDFSchema...?

- An **application program** can define and use **RDF data**
-if the application program knows, **which terms and classes to use**, as e.g.
 - name, title, year, ...
 - name, blog, phone number...

Why RDFSchema...?

- An **application program** can define and use **RDF data**
-if the application program knows, **which terms and classes to use**, as e.g.
 - name, title, year, ...
 - name, blog, phone number...
 - author, cites, ...

Why RDFSchema...?

- An **application program** can define and use **RDF data**
-if the application program knows, **which terms and classes to use**, as e.g.
 - name, title, year, ...
 - name, blog, phone number...
 - author, cites, ...
- But.....

Why RDFSchema...?

- An **application program** can define and use **RDF data**
-if the application program knows, **which terms and classes to use**, as e.g.
 - name, title, year, ...
 - name, blog, phone number...
 - author, cites, ...
- But.....
 - Are all terms known?

Why RDFSchema...?

- An **application program** can define and use **RDF data**
-if the application program knows, **which terms and classes to use**, as e.g.
 - name, title, year, ...
 - name, blog, phone number...
 - author, cites, ...
- But.....
 - Are all terms known?
 - Are all terms correct?

Why RDFSchema...?

- An **application program** can define and use **RDF data**
-if the application program knows, **which terms and classes to use**, as e.g.
 - name, title, year, ...
 - name, blog, phone number...
 - author, cites, ...
- But.....
 - Are all terms known?
 - Are all terms correct?
 - Are there (logical) relations among the terms?

Why RDFSchema...?

- An **application program** can define and use **RDF data**
-if the application program knows, **which terms and classes to use**, as e.g.
 - name, title, year, ...
 - name, blog, phone number...
 - author, cites, ...
- But.....
 - Are all terms known?
 - Are all terms correct?
 - Are there (logical) relations among the terms?
- We need a language for data definition: **RDF Schema**

Why RDFSchema...?

- An **application program** can define and use **RDF data**
-if the application program knows, **which terms and classes to use**, as e.g.
 - name, title, year, ...
 - name, blog, phone number...
 - author, cites, ...
- But.....
 - Are all terms known?
 - Are all terms correct?
 - Are there (logical) relations among the terms?
- We need a language for data definition: **RDF Schema**
 - officially called “**RDF Vocabulary Description Language**”

RDFSchema



- RDFSchema allows:
 - Definition of **classes**
 - **Class instantiation** in RDF via `<rdf:type>`
 - Definition of **properties** and **restrictions**
 - Definition of **hierarchies**
 - Subclasses and superclasses
 - Subproperties and superproperties

• Classes

- **`rdfs:Class`**

Concept of a class, defines an abstract object and is applied (with `rdf:type`) to create instances

- **`rdf:Property`**

Base class for properties

- **`rdfs:Literal`**

Class for literals

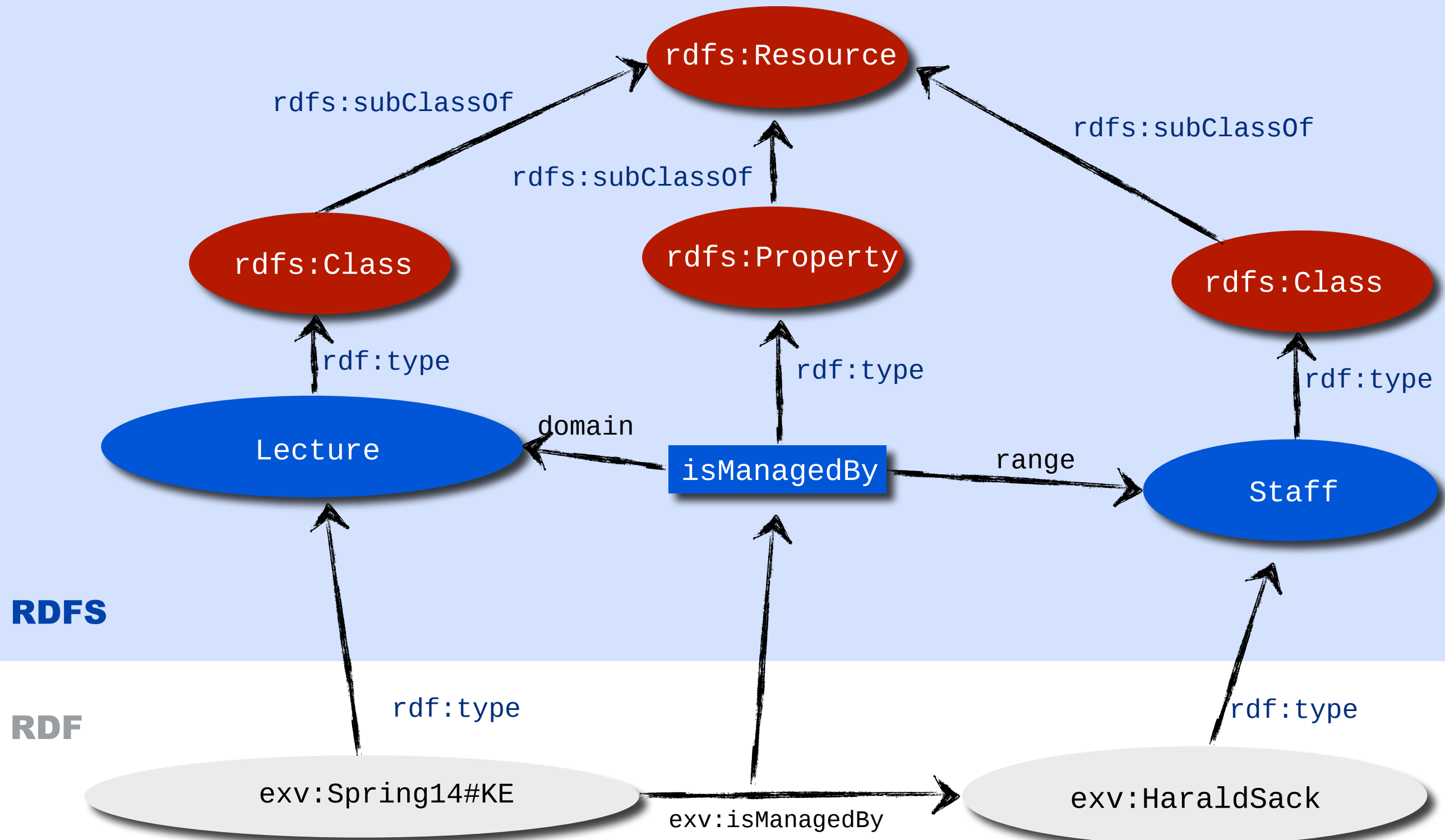
- **`rdfs:Resource`**

every entity of an RDF model is instance of this class

- and additionally

**`rdfs:Datatype`, `rdf:XMLLiteral`, `rdfs:Container`,
`rdfs:ContainerMembershipProperty`**

11



- Properties

- **`rdfs:subClassOf`**

- transitive property to define inheritance hierarchies for classes

- **`rdfs:subPropertyOf`**

- transitive property to define inheritance hierarchies for properties

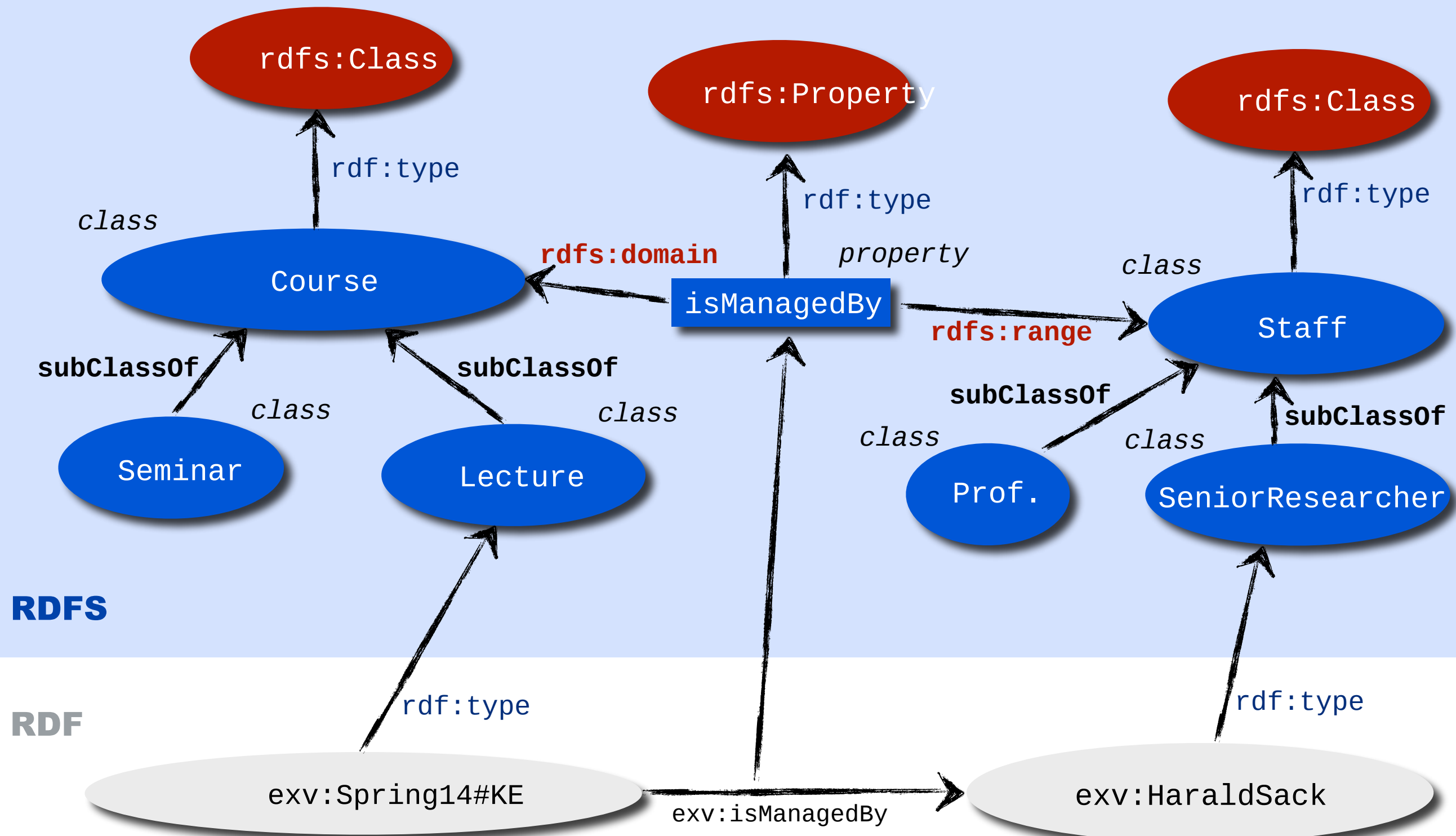
- **`rdfs:domain`**

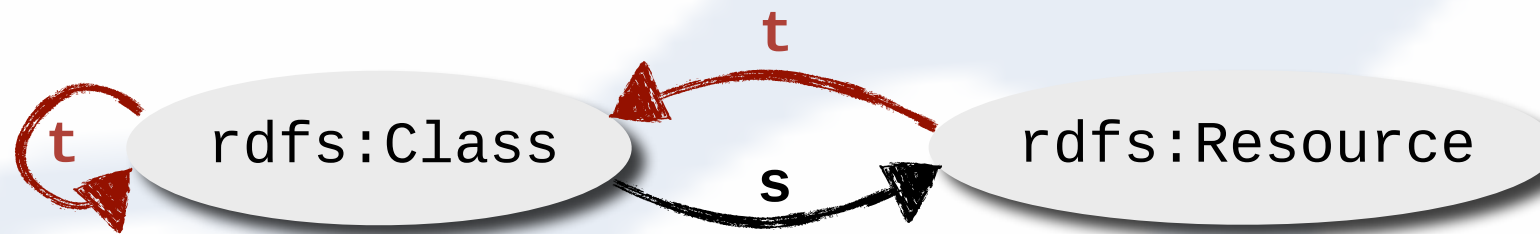
- defines the domain of a property concerning a class

- **`rdfs:range`**

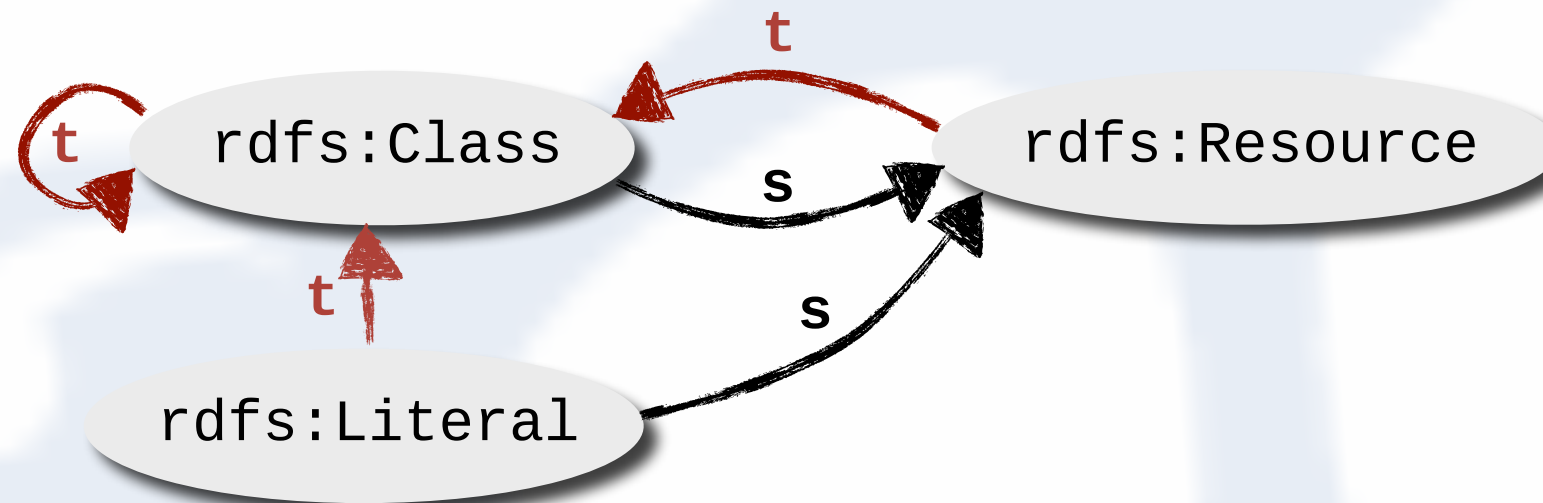
- defines range of a property concerning a class

13

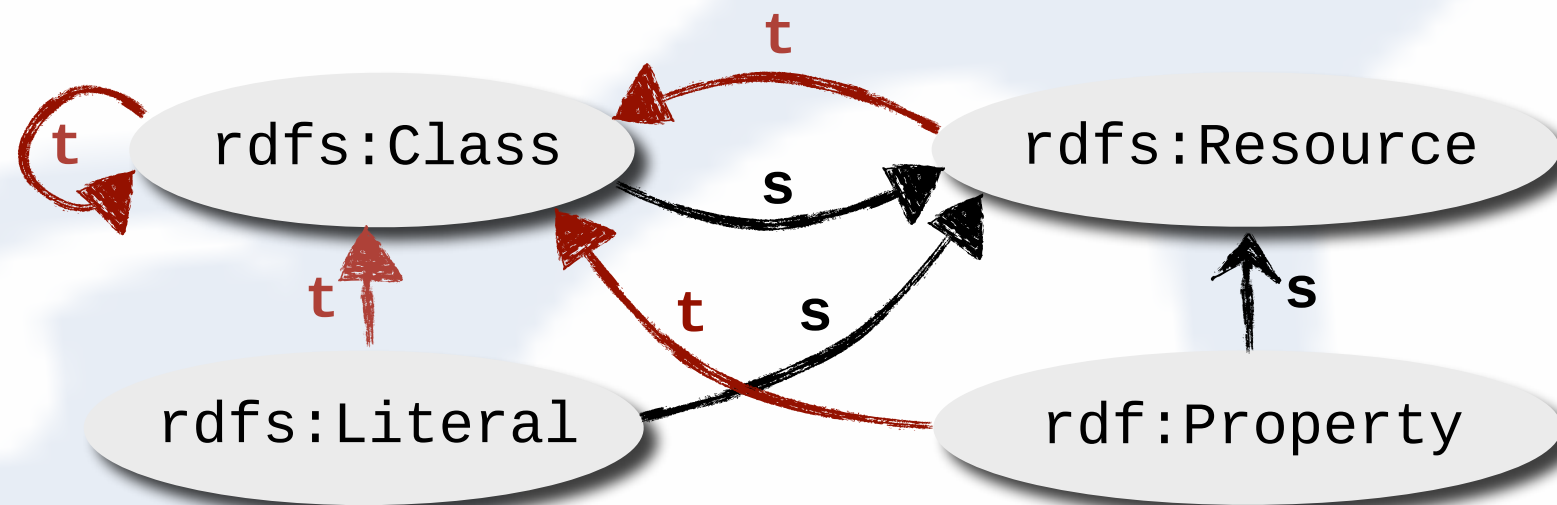




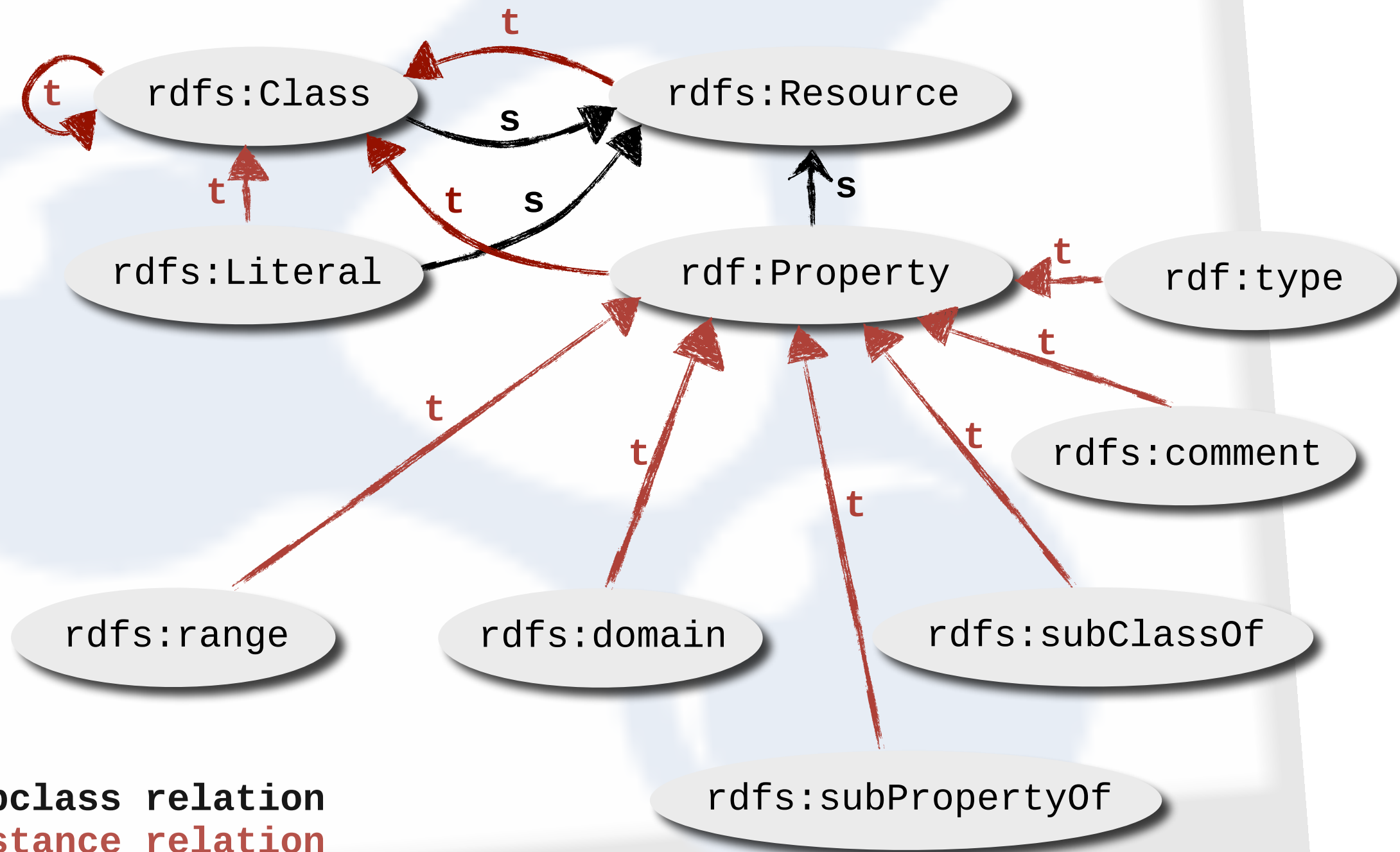
s - subclass relation
t - instance relation



s - subclass relation
t - instance relation



s - subclass relation
t - instance relation

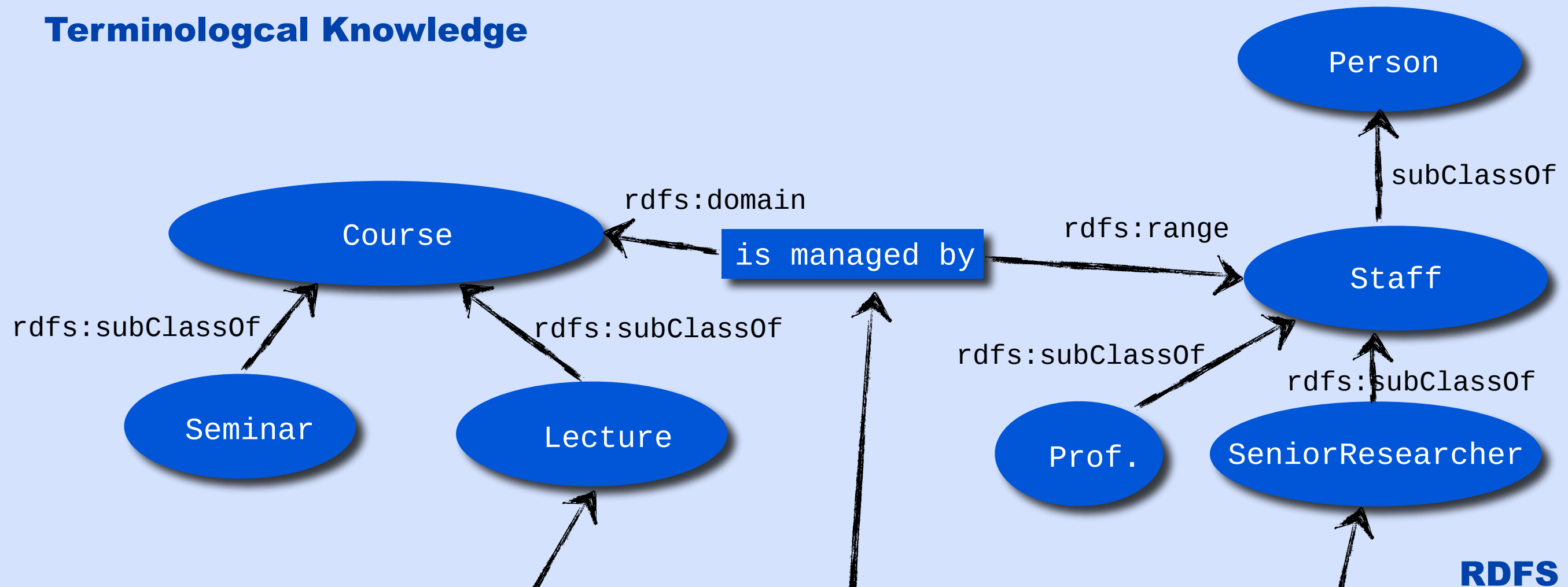


RDF(S) Knowledge Base

15

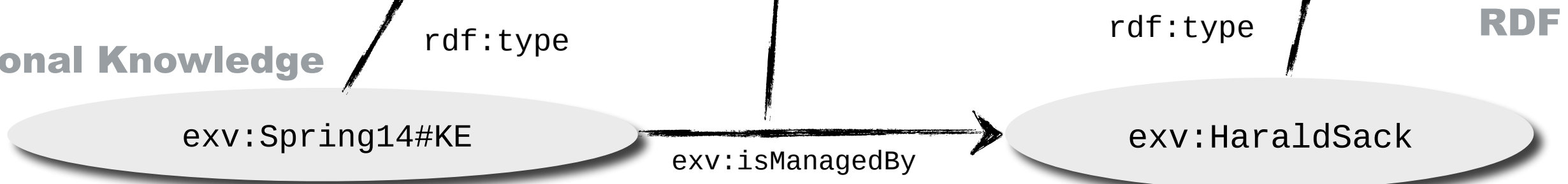
T-Box

Terminological Knowledge



A-Box

Assertional Knowledge



```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> . Turtle
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix : <http://example.org/>.
```

```
:Lecture a rdfs:Class;
    rdfs:subClassOf :Course.
:Seminar a rdfs:Class ;
    rdfs:subClassOf :Course.
```

```
:Person a rdfs:Class .
:Staff a rdfs:Class ;
    rdfs:subClassOf :Person .
:SeniorResearcher a rdfs:Class ;
    rdfs:subClassOf :Staff .
:Professor a rdfs:Class ;
    rdfs:subClassOf :Staff.
:isManagedBy a rdf:Property ;
    rdfs:domain :Course ;
    rdfs:range :Staff .
```

```
<Spring14#KE> a :Lecture .
:HaraldSack a :SeniorResearcher .
<Spring14#KE> :isManagedBy :HaraldSack .
```

- RDFSchema defines a data model for the definition of simple ontologies (knowledge representations).
- Via RDFSchema ontologies RDF statements (facts) can be expressed
- More than XML:
 - (small) ontological agreement about modelling primitives
 - Possibility to define own vocabularies



08 How much semantics is there in RDF(S)

Open HPI - Course: Knowledge Engineering with Semantic Web Technologies
Lecture 2: Semantic Web Technologies Part I