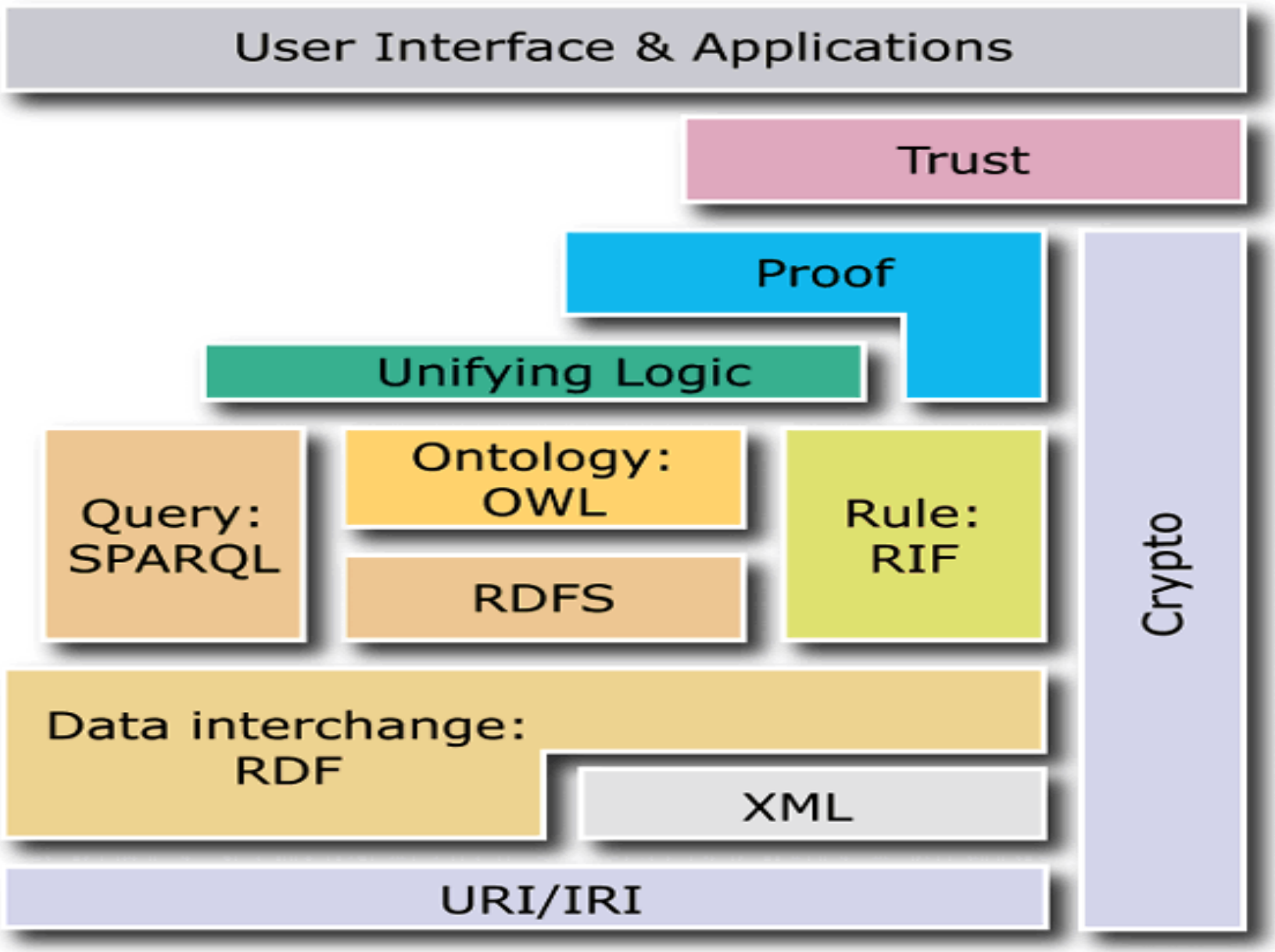
ONTLOLOGY LANGUAGE : RDFS SCHEMA

Hala Skaf-Molli
Associate Professor
Nantes University
Hala.Skaf@univ-nantes.fr
http://pagesperso.lina.univ-nantes.fr/~skaf-h

Semantic Web Cake



Ontology

- **A *specification of a conceptualization***.
- Une ontologie est une «spécification formelle et explicite d'une conceptualisation partagée» (Gruber, 1993)
- Why ?
  - Enable knowledge sharing and reuse.

- http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

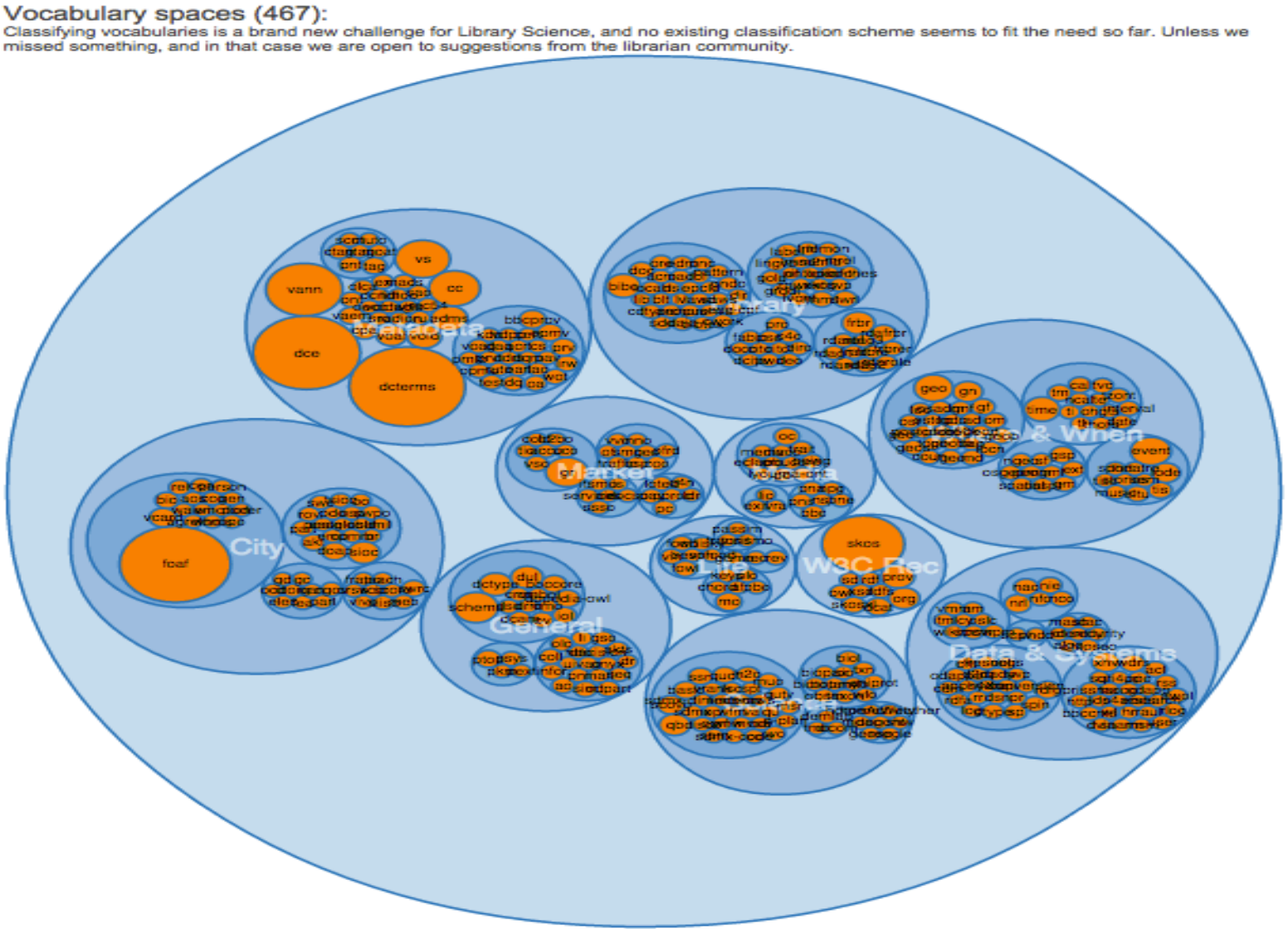
Ontology

- A vocabulary used to describe (a particular viewof) some domain
- An explicit specification of the intended meaning of the vocabulary.
  - Often includes classification based information
- Constraints capturing additional knowledge about the domain
  - Capture a shared understanding of a domain of interest
  - Provide a formal and machine manipulable model of the domain

Where are ontologies used?

- e-Science, e.g., Bioinformatics
  - Open Biomedical Ontologies Consortium (GO, MGED)
- Medicine
  - Building/maintaining terminologies such as Snomed (clinical term)
- Organising complex and semi-structured information
  - UN-FAO, NASA, General Motors, Lockheed Martin, …
- Military/Government
  - DARPA, NIST, SAIC, Department of Homeland Security, …
- E-Commerce
  - goodRelations : Google, Yaho, BestBuy,Kmart,…

- [Linked Open Vocabularies (LOV): ontologies used in the LOD]
  - [http://lov.okfn.org/dataset/lov/]



Aslo

- http://schema.org/
- http://www.heppnetz.de/projects/goodrelations/
- http://disease-ontology.org/
- ….

Ontology languages

- Ontology languages allow users to write explicit, formal conceptualizations of domain models
- The main requirements are:
  - a well-defined syntax
  - a formal semantics
  - sufficient expressive power
  - efficient reasoning support

Ontology= Schema +instances

- **Schema**
  - The set of **class** and **relation** names
  - The signatures of relations and also constraints
  - The **constraints** that are used for two purposes
    - checking data consistency (like dependencies in databases)
    - inferring new facts
- **Instance**
  - The set of facts
  - The set of base facts together with the inferred facts should satisfy the constraints
- Ontology (i.e., Knowledge Base) = Schema + Instance

Ontology Languages

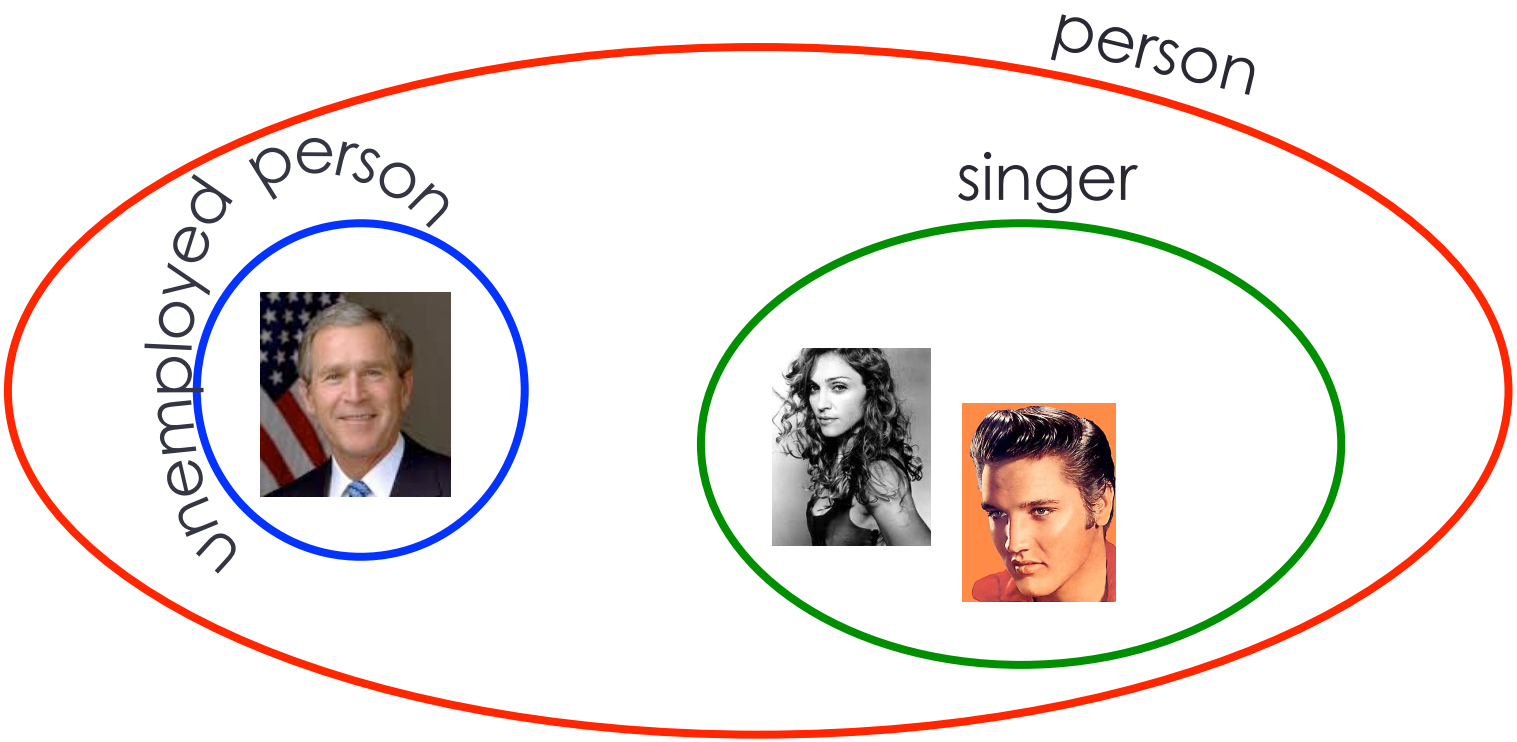
- RDFS (RDF Schema)
- OWL (Web Ontology Language) which is an extension to RDFS (RDF Schema) based on description logic (DL)

RDFS

- **RDF** is a very simple language that lets users describe resources in their **own vocabularies**
- RDF does not assume, nor does it define semantics of any particular application domain
- The user can do so in **RDF Schema** using predefined vocabularies:
  - **Classes** and **Properties**
  - **Class Hierarchies** and **Inheritance**
  - **Property Hierarchies** and **Inheritance**

Classes

- A **class** (also called concept) can be understood as a set of similar entities.



Slides Fabian Suchanek

Classes in RDF

The fact that an entity belongs to a class is expressed by the **type** predicate from the standard namespace rdf (http://www.w3.org/2000/01/rdf-schema#").

The fact that a class is a sub-class of another class is expressed by the **subclassOf** predicate from the standard namespace rdfs (http://w3c.org/...).

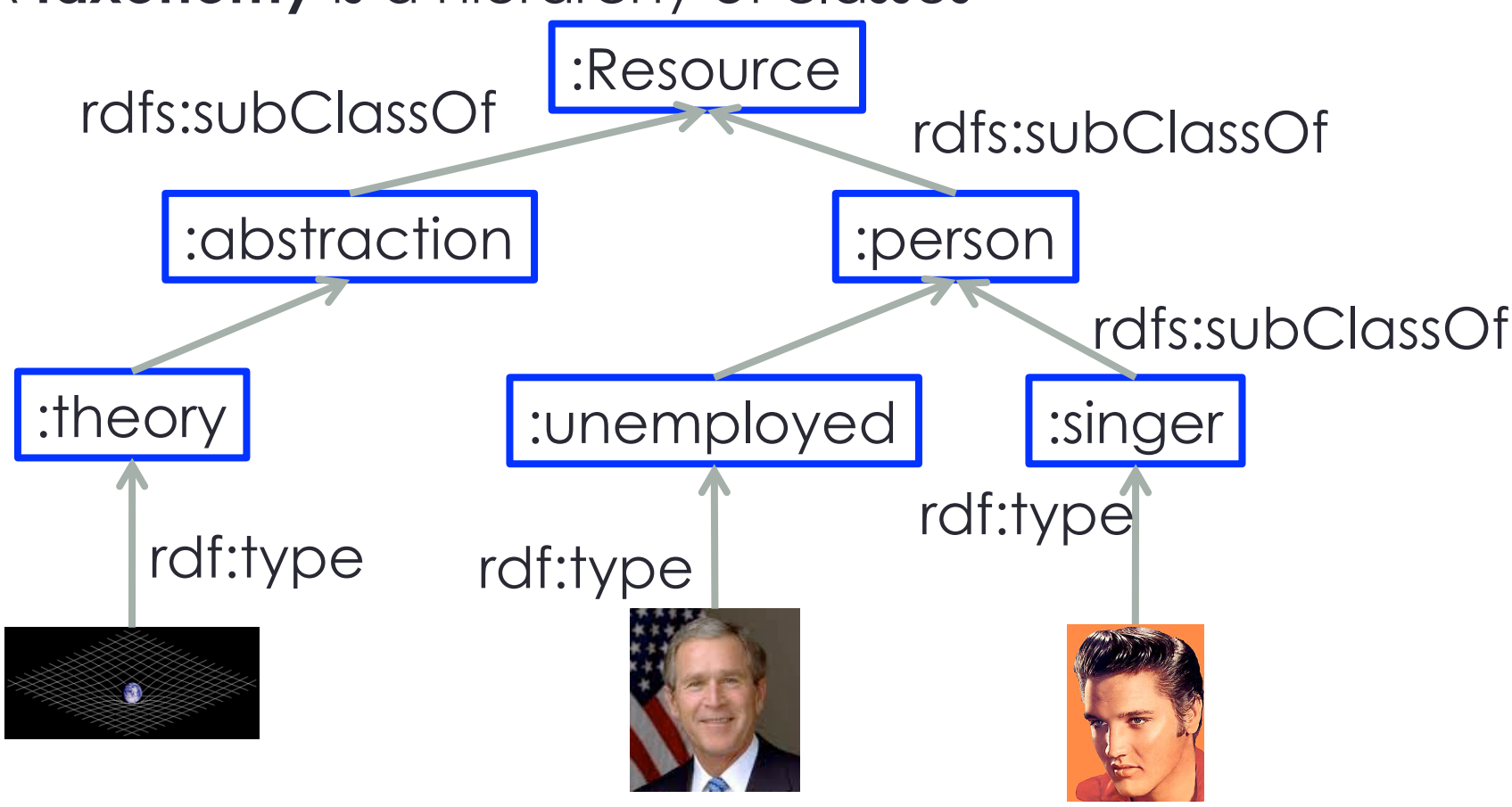
Classes in RDF

The fact that an entity belongs to a class is expressed by the **type** predicate from the standard namespace **rdf** (http://w3c.org/...).

The fact that a class is a sub-class of another class is expressed by the **subclassOf** predicate from the standard namespace **rdfs** (http://w3c.org/...).

Taxonomy

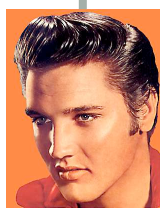
A **taxonomy** is a hierarchy of classes



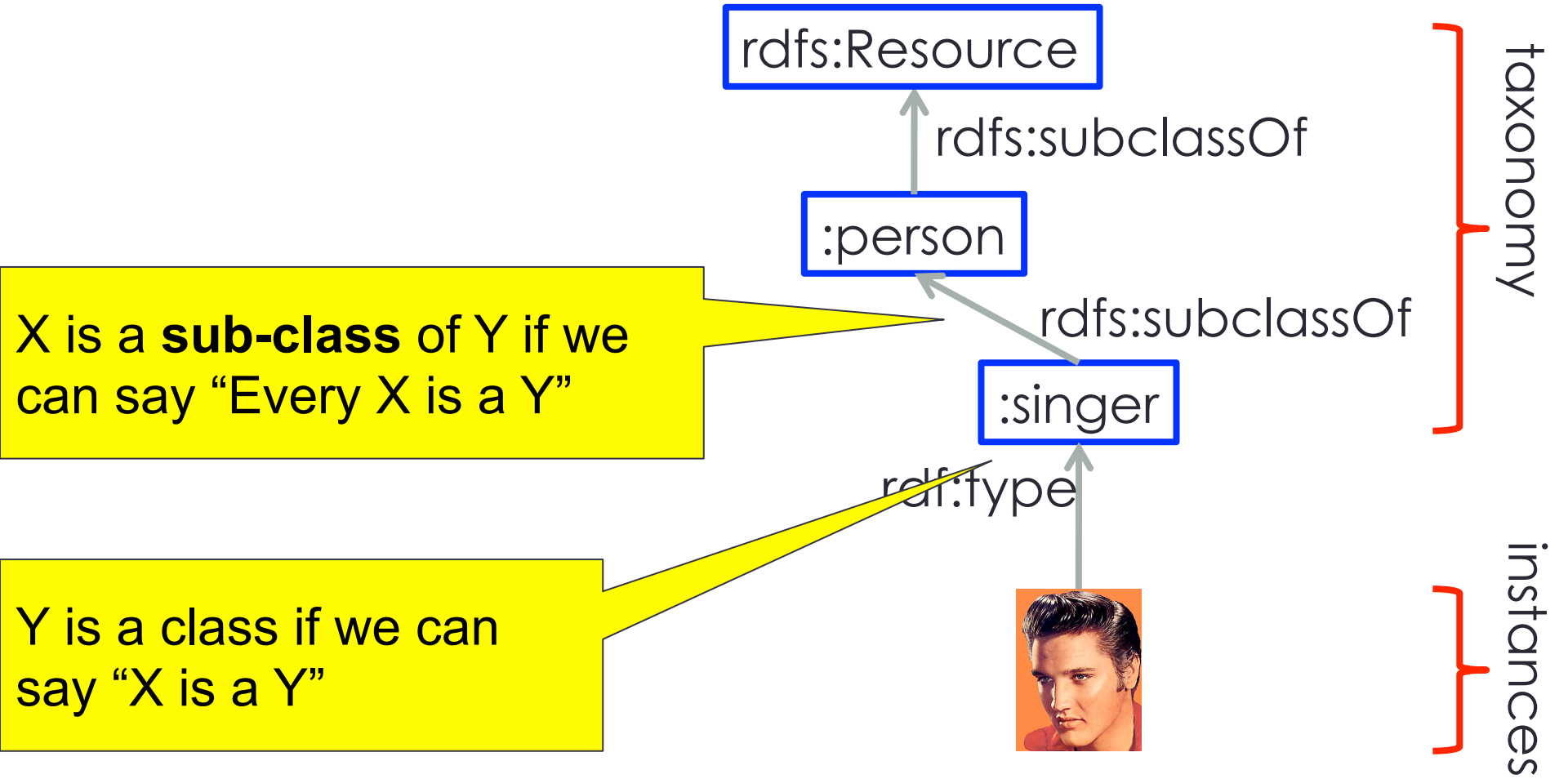
The most general class is **rdfs:Resource** – everything is a resource.

More general class

More specific class

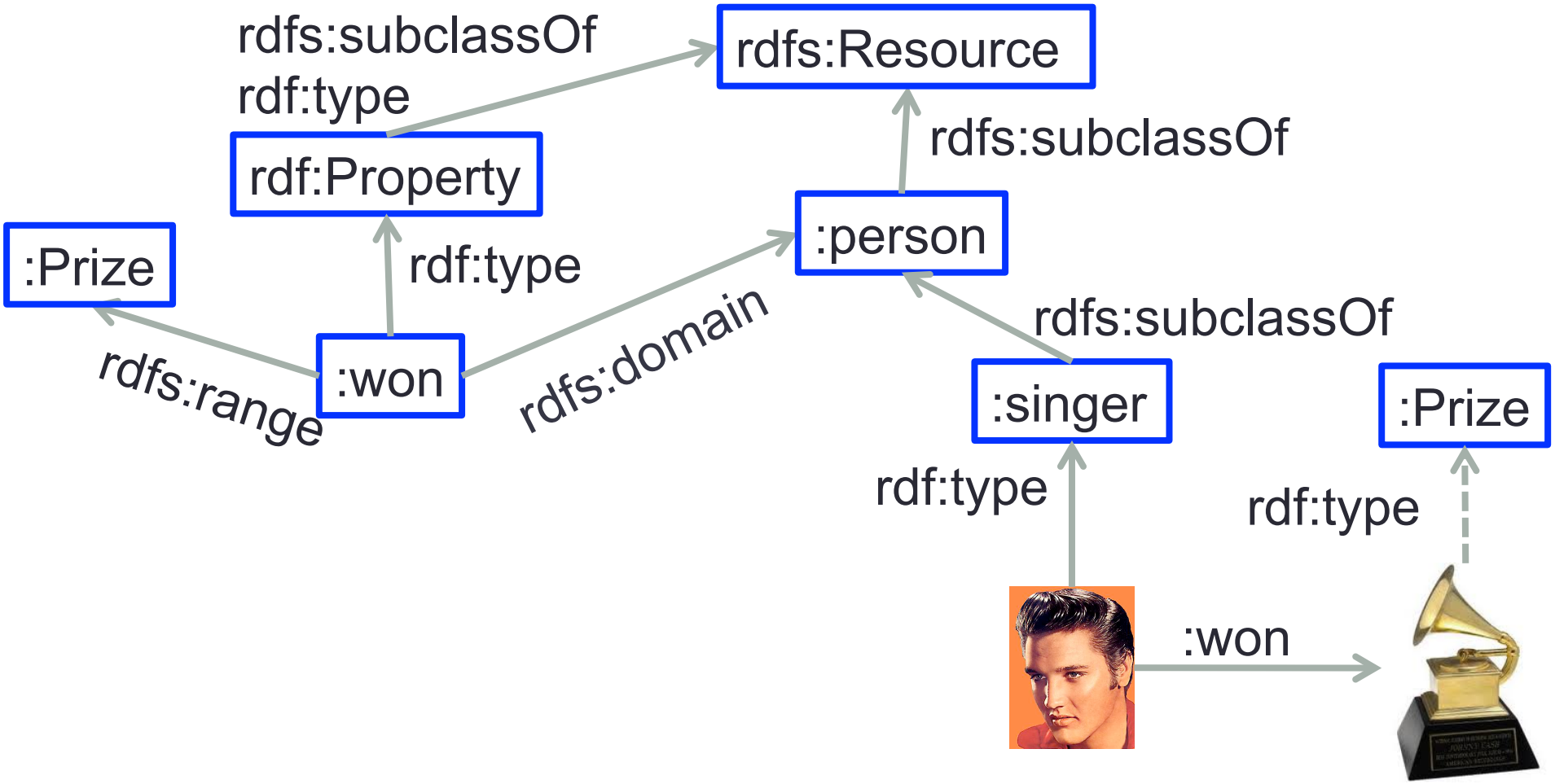


Taxonomy

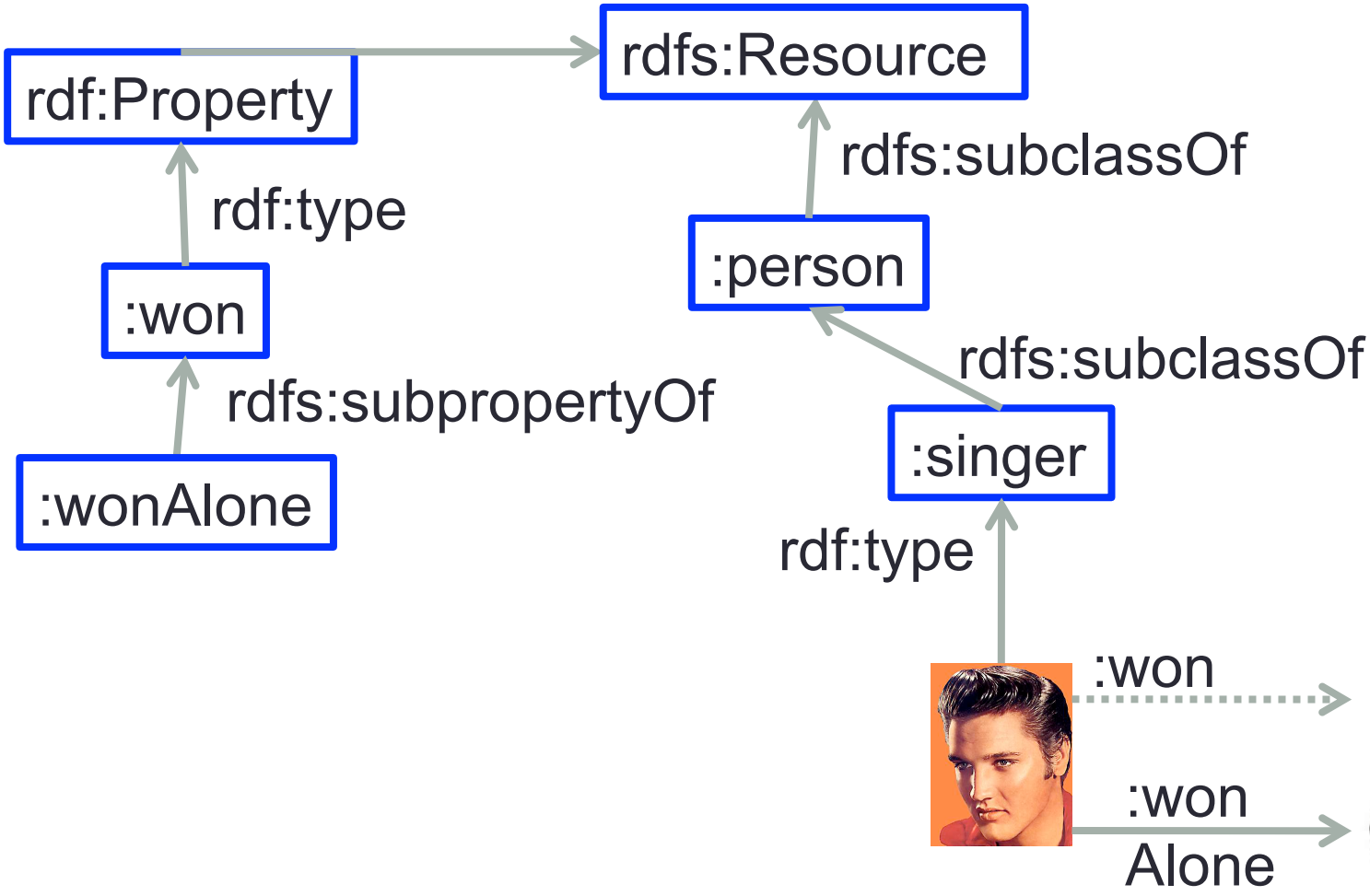

rdfs:Resource
rdfs:subclassOf
:person
rdfs:subclassOf
:singer
rdf:type
taxonomy
instances
X is a **sub-class** of Y if we can say “Every X is a Y”
Y is a class if we can say “X is a Y”

Domain & Range

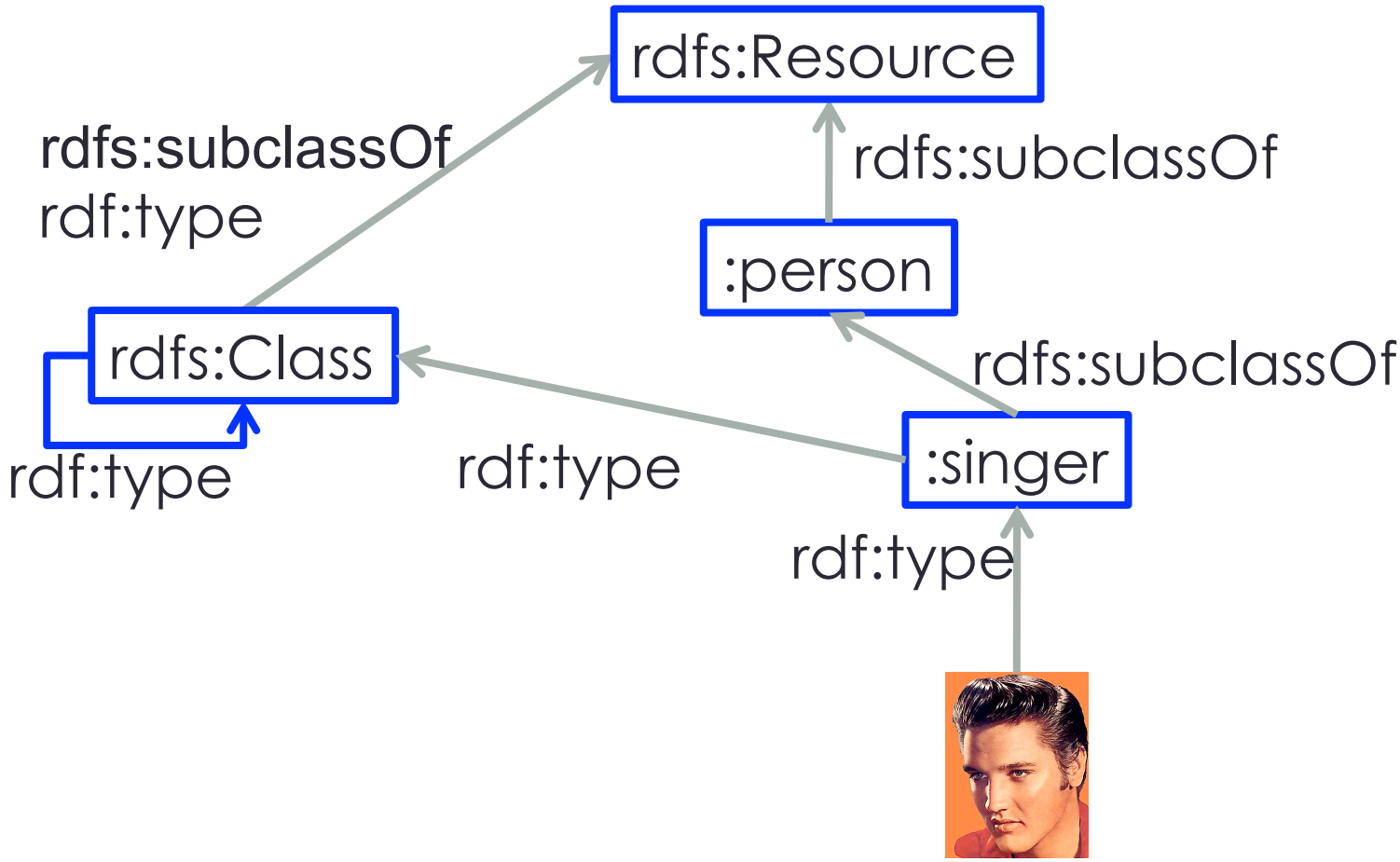
Properties (predicates) are instances of rdf:Property.



Sub-Properties



RDFS: “Meta Facts”



Core Classes

- **rdfs:Resource**, the class of all resources
  - All other classes are subclasses of this
  - rdfs:Resource is an instance of rdfs:Class
- **rdfs:Class**, the class of all RDF classes
  - rdfs:Class is an instance of rdfs:Class
  - A class can be instance of itself
- **rdfs:Literal**, the class of all literals
  - rdf:Literal is an instance of rdfs:Class, and sub
- **rdf:Property**, the class of all properties.
  - Rdf:Property is an instance of rdfs:Class
- **rdf:Datatype, rdf:langString, rdf:HTML**

Core Properties (1)

- **rdfs:domain** is an instance of rdf:Property
  - P rdfs:domain C states that:
  - P is an instance of the class rdf:Property
  - C is an instance of the class rdfs:Class
  - The subject of triples whose predicate is P are instances of the class C
- **rdfs:range** is an instance of rdf:Property
  - P rdfs:rang C states that:
  - P is an instance of the class rdf:Property
  - C is an instance of the class rdfs:Class
  - The objects of triples whose predicate is P are instances of the class C

Core Properties (2)

- **rdf:type**, which relates a resource to its class
  - rdf:type is an instance of rdf:Property
  - R rdf:type C :
    - C is an instance of rdfs:Class and R is an instance of C.
- **rdfs:subClassOf**, is an instance of rdf:Property. It relates a class to one of its superclasses
  - All instances of a class are instances of its superclass interpreted as set inclusion
  - C1 rdfs:subClassOf C2 :
    - C1 is an instance of rdfs:Class, C2 is an instance of rdfs:Class
    - The rdfs:domain, the rdfs:range of rdfs:subClassOf are rdfs:Class

Core Properties (2)

- **rdfs:subPropertyOf** is an instance of rdf:Propery.
  - It relates a property to one of its superproperties
  - P1 rdfs:subProperty Of P2 :
    - P1 is an instance of rdf:Property, P2 is an instance of rdf:Property
    - rdfs:domain of rdfs:subPropertyOf is rdf:Property
    - Rdfs:range of rdfs:subPropertyOf is rdf:Property
- **rdfs:label:** is an instance of rdf:Property that may be use to provide human-readable version of a resource's name.
  - R rdfs:lablel L
  - Domain: rdfs:Resource
  - Range: rdfs:Literal
- **rdfs:comment** provide human-readable description of a resource
  - R rdfs:lablel L
  - Domain: rdfs:Resource
  - Range: rdfs:Literal
- Rdf:subject, rdf:predicate, rdf:object

Utility Properties

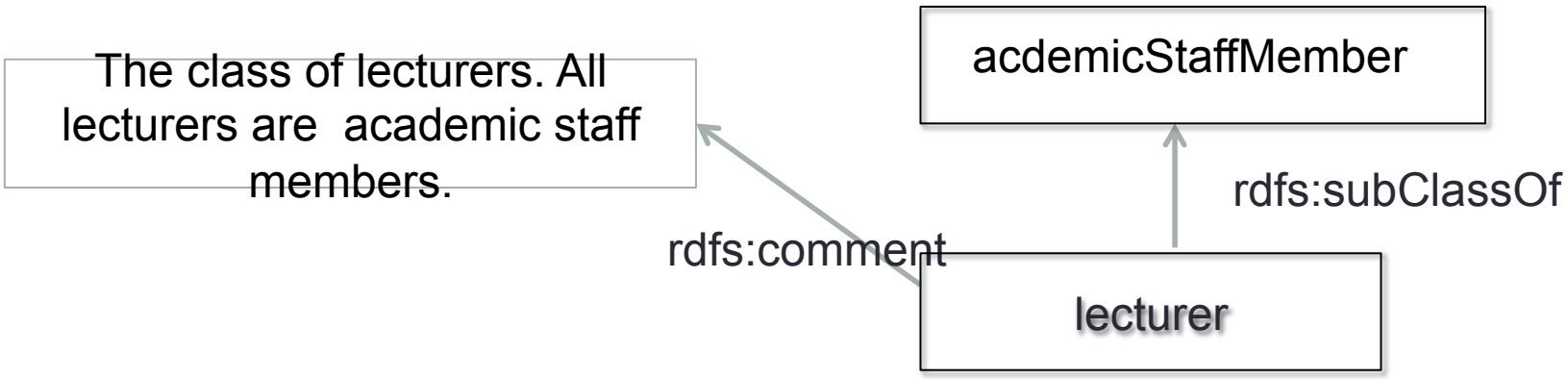
- **rdfs:seeAlso** relates a resource to another resource that explains it
  - S rdfs:seeAlso O
  - O may provide additional information about S.
  - The rdfs:domain of rdfs:seeAlso is rdfs:Resource.
    The rdfs:range of rdfs:seeAlso is rdfs:Resource.
- **rdfs:isDefinedBy** is a subproperty of **rdfs:seeAlso** and relates a resource to the place where its definition, typically an RDF schema, is found
  - Domain, range are rdfs:Resouce

Relationships Between Core Classes and Properties

- **rdfs:subClassOf** and **rdfs:subPropertyOf** are transitive, by definition
- **rdfs:Class** is a subclass of rdfs:Resource
  - Because every class is a resource
- **rdfs:Resource** is an instance of **rdfs:Class**
  - **rdfs:Resource** is the class of all resources, so it is a class
- Every class is an instance of **rdfs:Class**
  - For the same reason

XML-based syntax of RDF for RDFS: Example

```
<rdfs:Class rdf:ID="lecturer">
    <rdfs:comment>
        The class of lecturers. All lecturers are academic staff members.
    </rdfs:comment>
    <rdfs:subClassOf rdf:resource="#academicStaffMember"/>
</rdfs:Class>
```



Example (2)

```
<rdfs:Class rdf:about="#lecturer">
  <rdfs:subClassOf rdf:resource="#staffMember"/>
</rdfs:Class>

<rdf:Property rdf:ID="phone">
        <rdfs:domain rdf:resource="#staffMember"/>
        <rdfs:range rdf:resource="http://www.w3.org/
                2000/01/rdf-schema#Literal"/>
</rdf:Property>
```

Example (3)

```
<rdfs:Class rdf:ID="course">
      <rdfs:comment>The class of courses</rdfs:comment>
</rdfs:Class>

<rdf:Property rdf:ID="isTaughtBy">
      <rdfs:comment>
          Inherits its domain ("course") and range ("lecturer")
          from its superproperty "involves"
      </rdfs:comment>
      <rdfs:subPropertyOf rdf:resource="#involves"/>
</rdf:Property>
```

RDFS logical semantics

RDF and RDFS statements	FOL translation	DL notation
⟨ i rdf:type C ⟩	$C(i)$	$i : C$ or $C(i)$
⟨ i P j ⟩	$P(i,j)$	iPj or $P(i,j)$
⟨ C rdfs:subClassOf D ⟩	$\forall X\,(C(X) \Rightarrow D(X))$	$C \sqsubseteq D$
⟨ P rdfs:subPropertyOf R ⟩	$\forall X \forall Y\,(P(X,Y) \Rightarrow R(X,Y))$	$P \sqsubseteq R$
⟨ P rdfs:domain C ⟩	$\forall X \forall Y\,(P(X,Y) \Rightarrow C(X))$	$\exists P \sqsubseteq C$
⟨ P rdfs:range D ⟩	$\forall X \forall Y\,(P(X,Y) \Rightarrow D(Y))$	$\exists P^{-} \sqsubseteq D$

Semantics based on Inference Rules

- Reasoning:
  - Deduce new facts based on existing ones
- This inference system consists of inference rules of the form:
- *IF E contains certain triples THEN*

  add to E certain additional triples

  where E is an arbitrary set of RDF triples

RDFS Semantics

- RDFS specifies 44 entailment rules
- The entailment rules are applied recursively until the graph does not change any more.

- The result is called the **deductive closure**.

Type Semantics

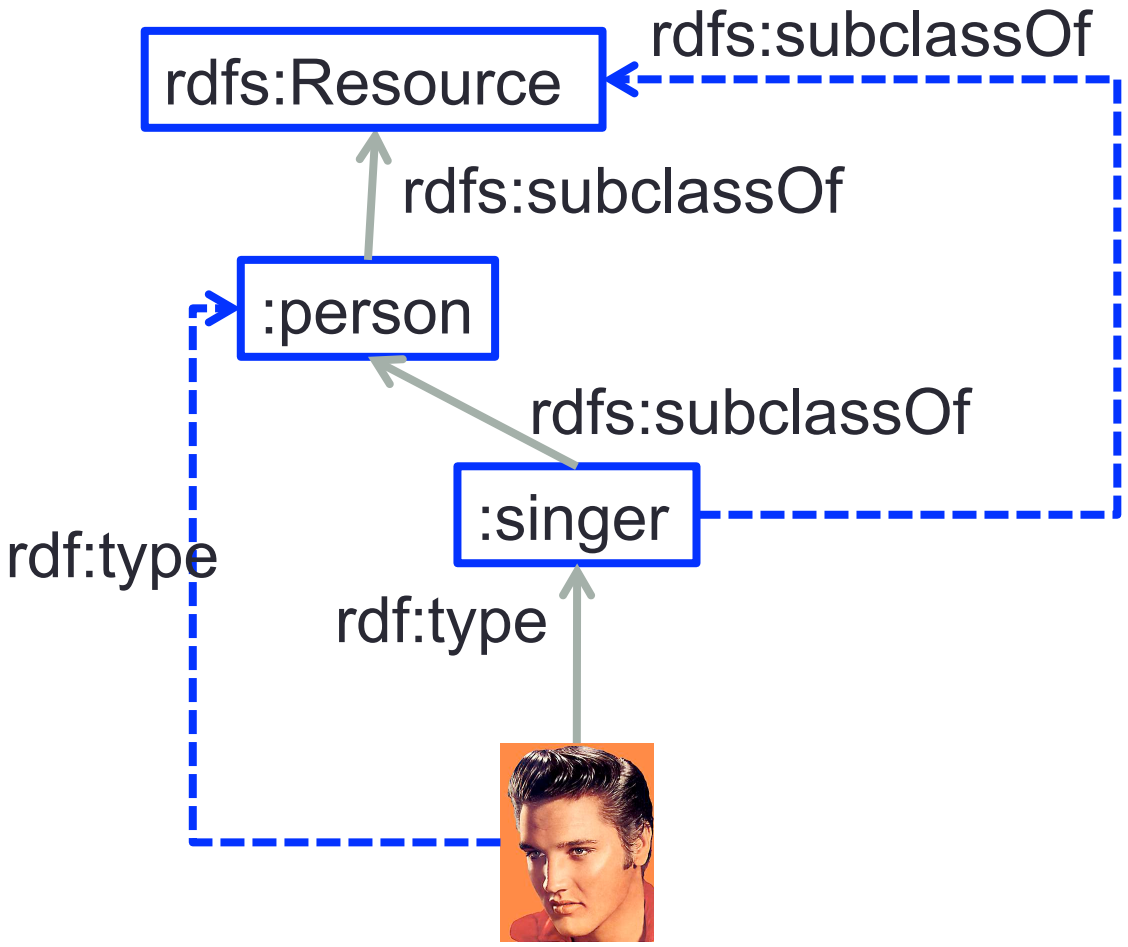
(:s rdf:type :t)

(:t rdf:type rdfs:Class)

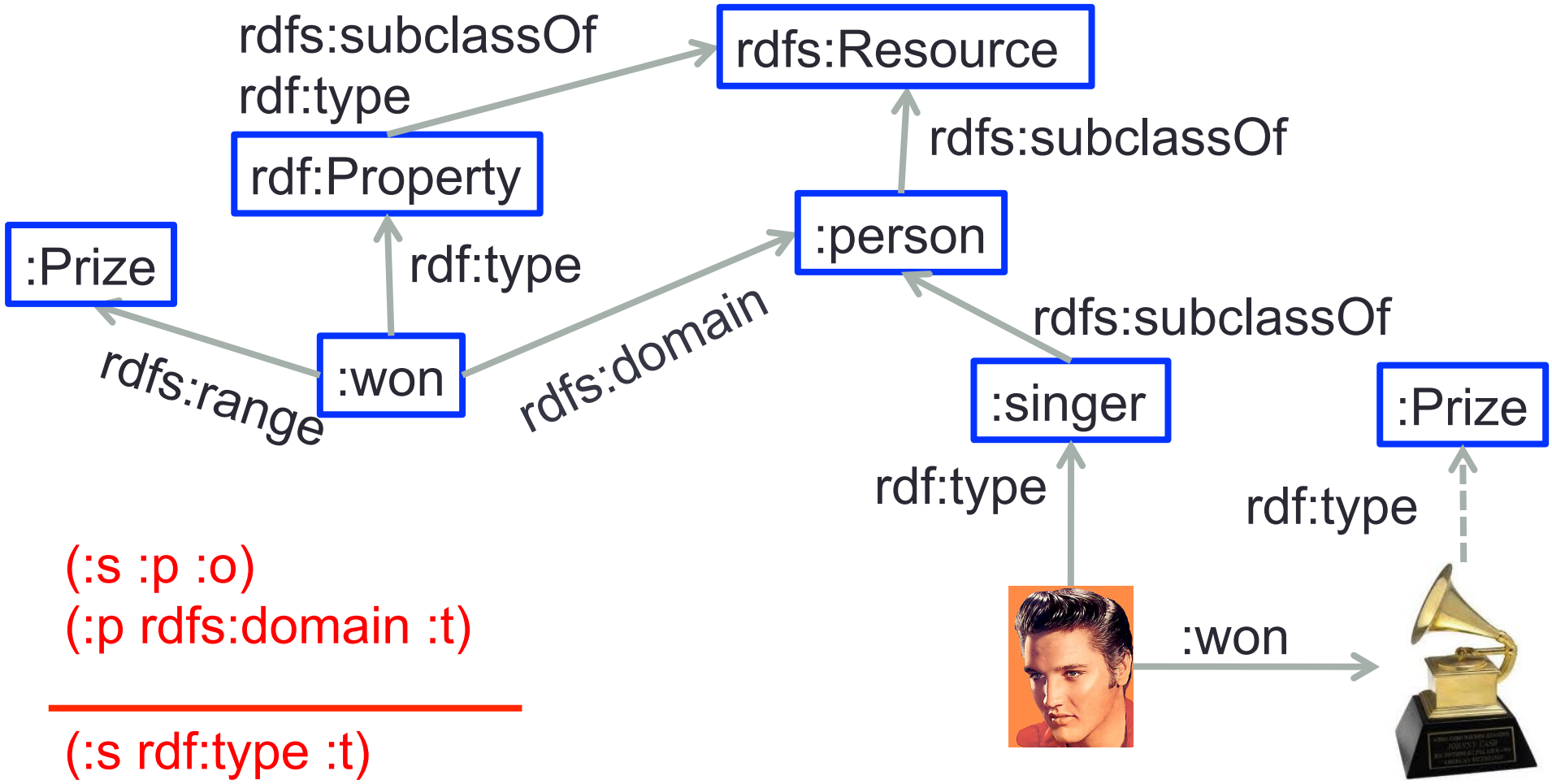
(:x rdf:type :t)
(:t rdfs:subClassOf: :z)

(:x rdf:type :z)

Every instance is an instance of all more general classes



Domain & Range Semantics



(:s :p :o)
(:p rdfs:domain :t)

(:s rdf:type :t)

Same for range

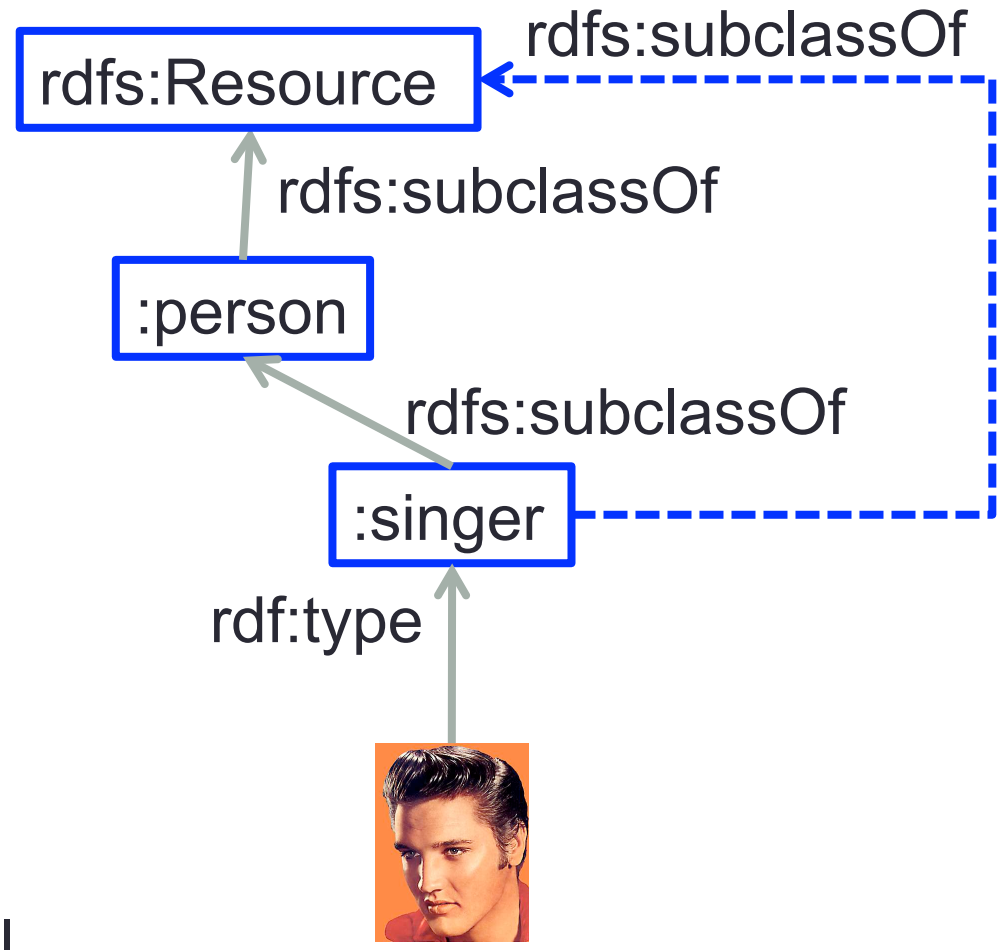
SubclassOf Semantics

(:s rdfs:subClassOf :o)

(:s rdf:type : rdfs:Class)
(:o rdf:type rdfs:Class)

(:s rdfs:subClassOf :o)
(:o rdfs:subClassOf :c)

(:s rdfs:subClassOf :c)



Every class is a subclass of all more general classes

(:p rdfs:subpropertyOf :r)
(:x :p :y)

(:x :r :y)

(:s :p :o)
(:p rdfs:domain :t)

(:s rdf:type :t)

(:s :p :o)
(:p rdfs:range :t)

(:o rdf:type :t)

(:s rdf:type :o)
(:o rdfs:subclassOf :c)

(:s rdf:type :c)

(:s rdfs:subClassOf :o)
(:o rdfs:subClassOf :c)

(:s rdfs:subClassOf :c)

(:won rdfs:domain :person)
(:Elvis :won ‘awards’)
=>

(:hasMother rdfs:range :Women)
(:Sheldon :hasMother :MrsCooper)
⇒

(:Homme rdfs:subClassOf :Humain)
(:Sheldon rdf:type :Homme)
⇒

Exercise (1)

(:p rdfs:subpropertyOf :r)
(:x :p :y)

(:x :r :y)

(:s :p :o)
(:p rdfs:domain :t)

(:s rdf:type :t)

(:s :p :o)
(:p rdfs:range :t)

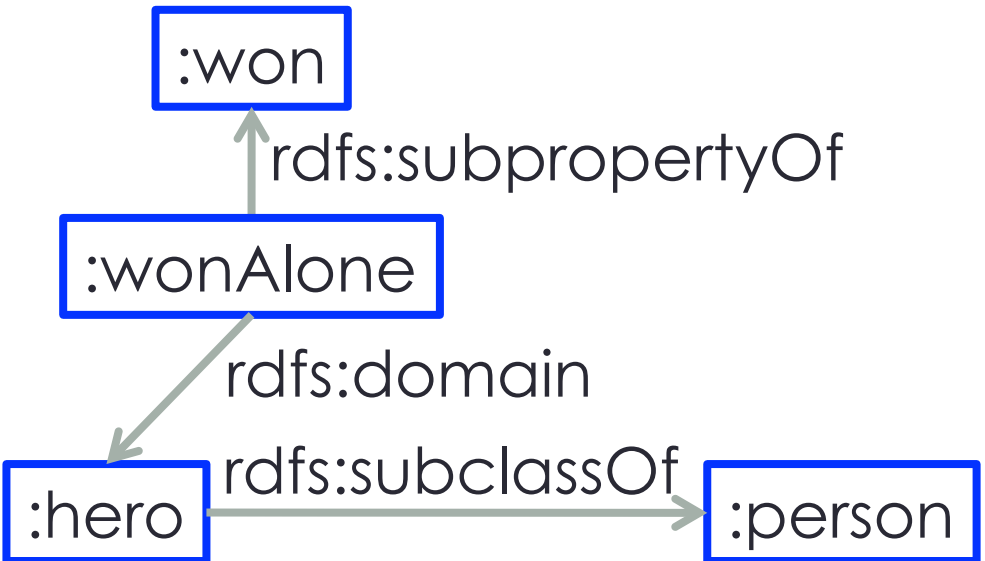
(:o rdf:type :t)

(:s rdf:type :o)
(:o rdfs:subclassOf :c)

(:s rdf:type :c)

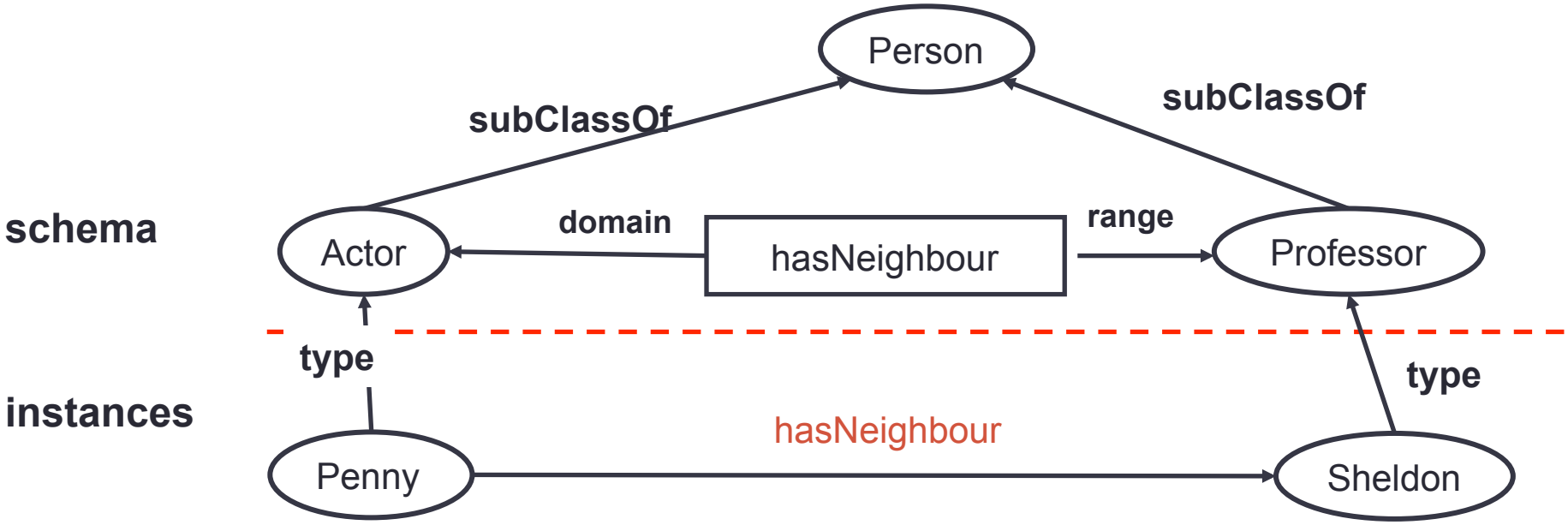
(:s rdfs:subClassOf :o)
(:o rdfs:subClassOf :c)

(:s rdfs:subClassOf :c)



Compute the deductive closure

Exercise (2)


Person
subClassOf
subClassOf
schema
Actor
domain
hasNeighbour
range
Professor
type
type
instances
hasNeighbour
Penny
Sheldon

Standard Vocabulary

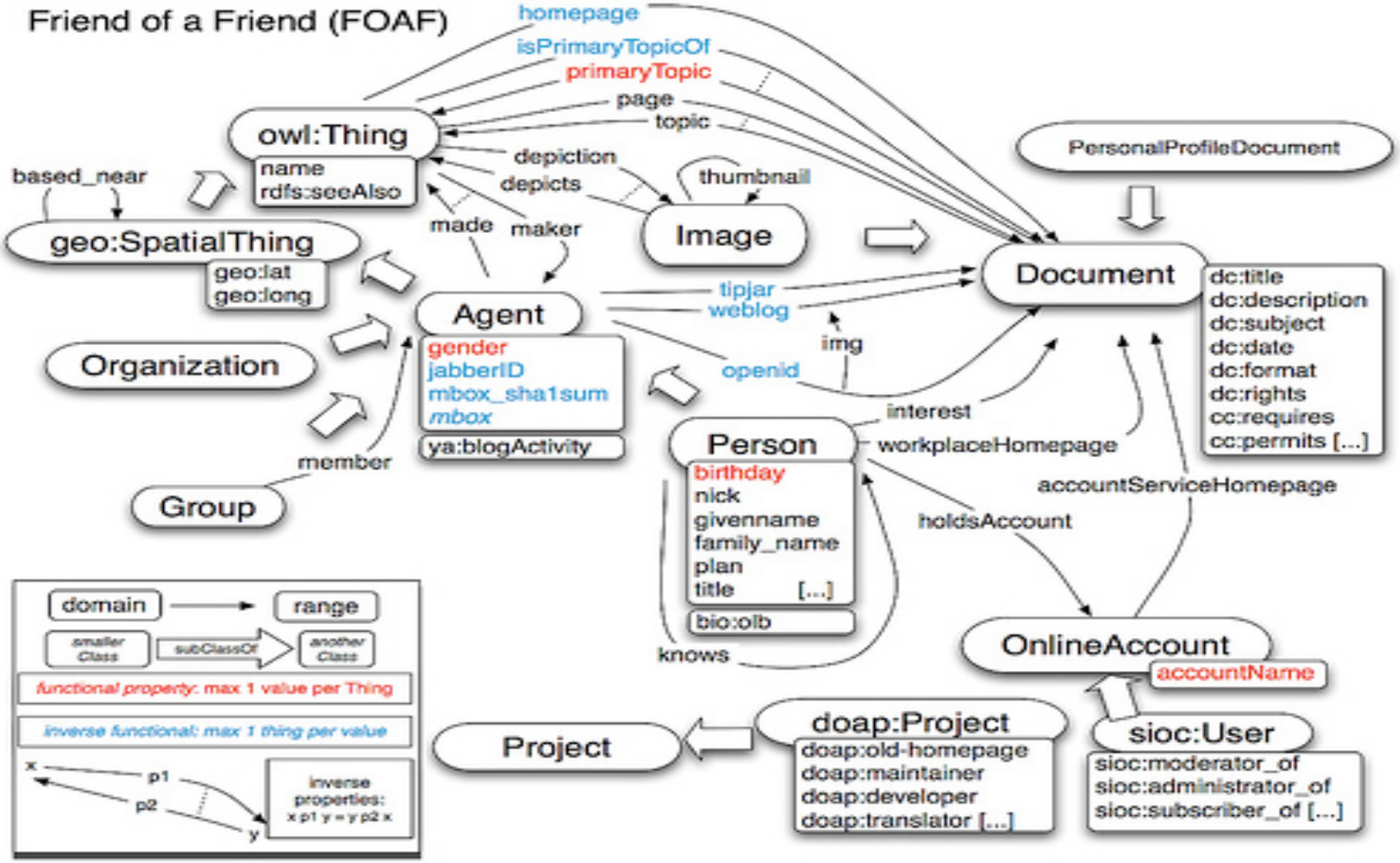
A number of standard vocabularies have evolved

dc: Dublin Core (predicates for describing documents)
http://purl.org/dc/elements/1.1/

foaf: Friend Of A Friend (relationships between people)
http://xmlns.com/foaf/0.1/

cc: Creative Commons (types of licences)
http://creativecommons.org/ns#

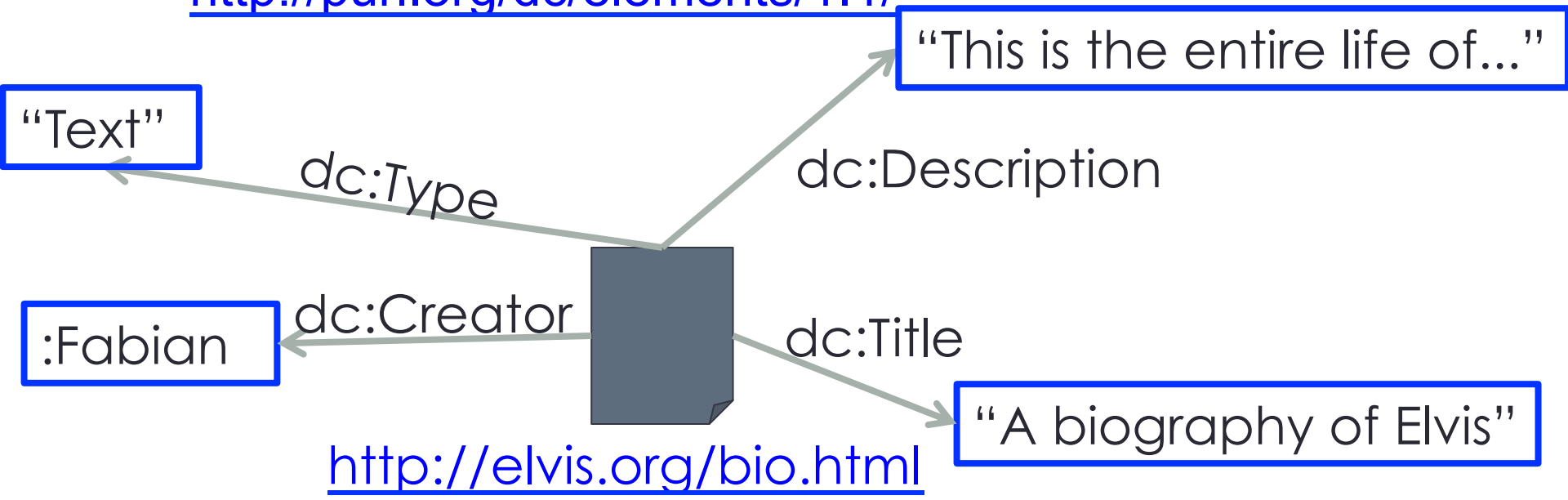
Main classes and properties of FOAF



http://danbri.org/words/2007/11/04/223

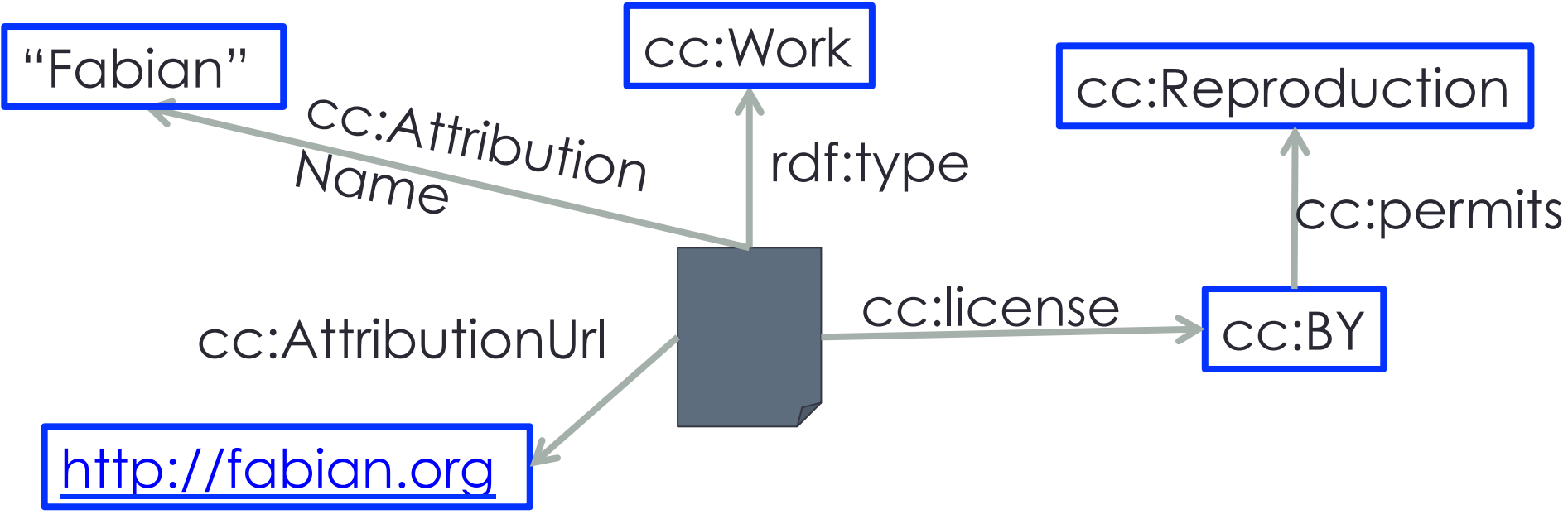
Dublin Core

dc: Dublin Core (predicates for describing documents)
http://purl.org/dc/elements/1.1/



Creative Commons

cc: Creative Commons (types of licences)
http://creativecommons.org/ns



Creative Commons is a non-profit organization, which defines very popular licenses, notably

- CC-BY: Free for reuse, just give credit to the author
- CC-BY-NC: Free for reuse, give credit, non-commercial use only
- CC-BY-ND: Free for reuse, give credit, do not create derivative works

Problems with RDFS 'liberal' Semantic

- No distinction between classes and instances
  - (:Species rdf:type rdfs:Class)
  - (:Lion rdf:type Species)
  - (:Leo rdf:type :Lion)
- Properties themselves have properties
  - (:hasDaughter rdfs:subPropertyOf :hasChild)
  - (:hasDaughter rdf:type rdf:Property)
- No localized domain and range constraints
  - Cannot say range of hasChild is person in the context of person and elephant in the context of elephant
- No cardinality on properties: has exactly two parent
- No inverse, transitive,.. properties

Conclusion

Classes	Properties	Properties-constraints
rdfs:Resource rdf:Property rdfs:Class	rdf:type rdfs:subClassOf rdfs:subPropertyOf	**rdfs:domain** **rdfs:range**

We need a more expressive and well-grounded ontology language….

- OWL Full
- OWL DL
- OWL Lite