Knowledge Representation

Craig Knoblock, DSCI 558
02/01/2021
Slides from Pedro Szekely
USC Information Sciences Institute

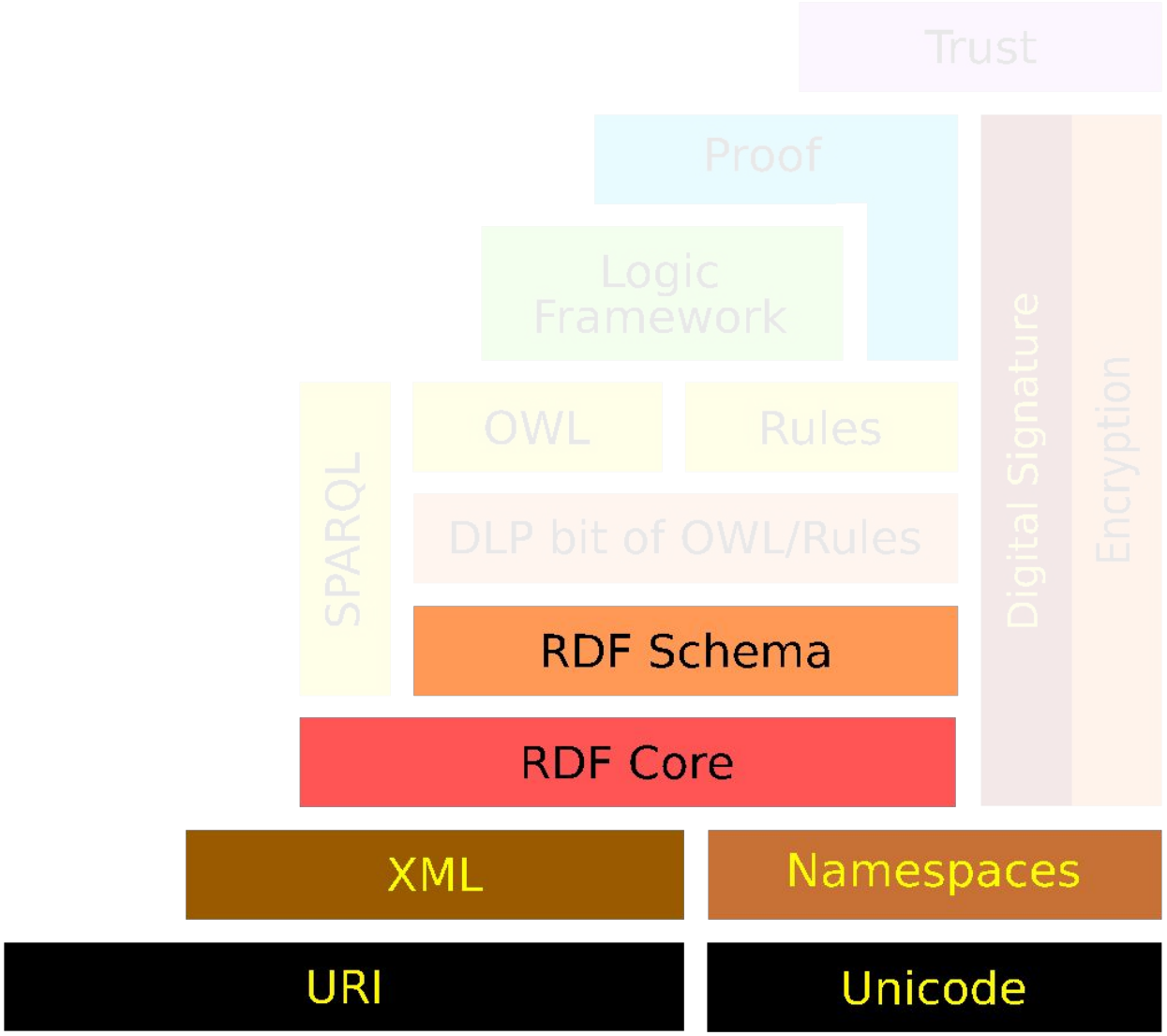
Outline

1. RDF representation framework
   a. URI, triple, graph
   b. syntax
   c. modeling more complex structures
2. RDFS (RDF schema)
   a. classes
   b. properties
3. RDF and KGs
   a. representation example
   b. storage of KGs

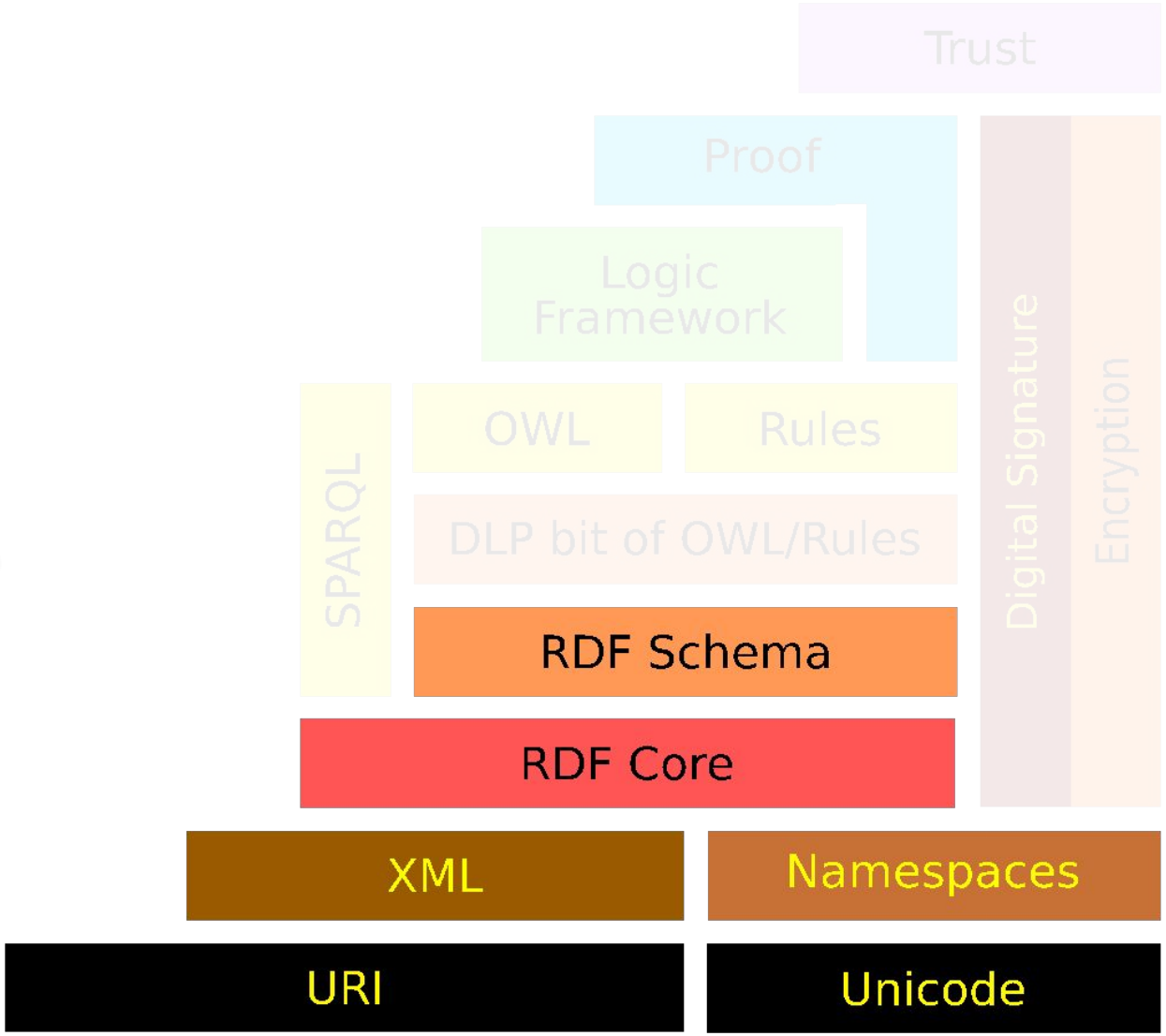
RDF

slide by Pedro Szekely

Semantic Web Layer Cake



Unicode



slide by Pedro Szekely

Why Unicode?

http://site.com/Македонски.html

http://site.com/Μία_Σελίδα

http://www.中国政府.政务.cn

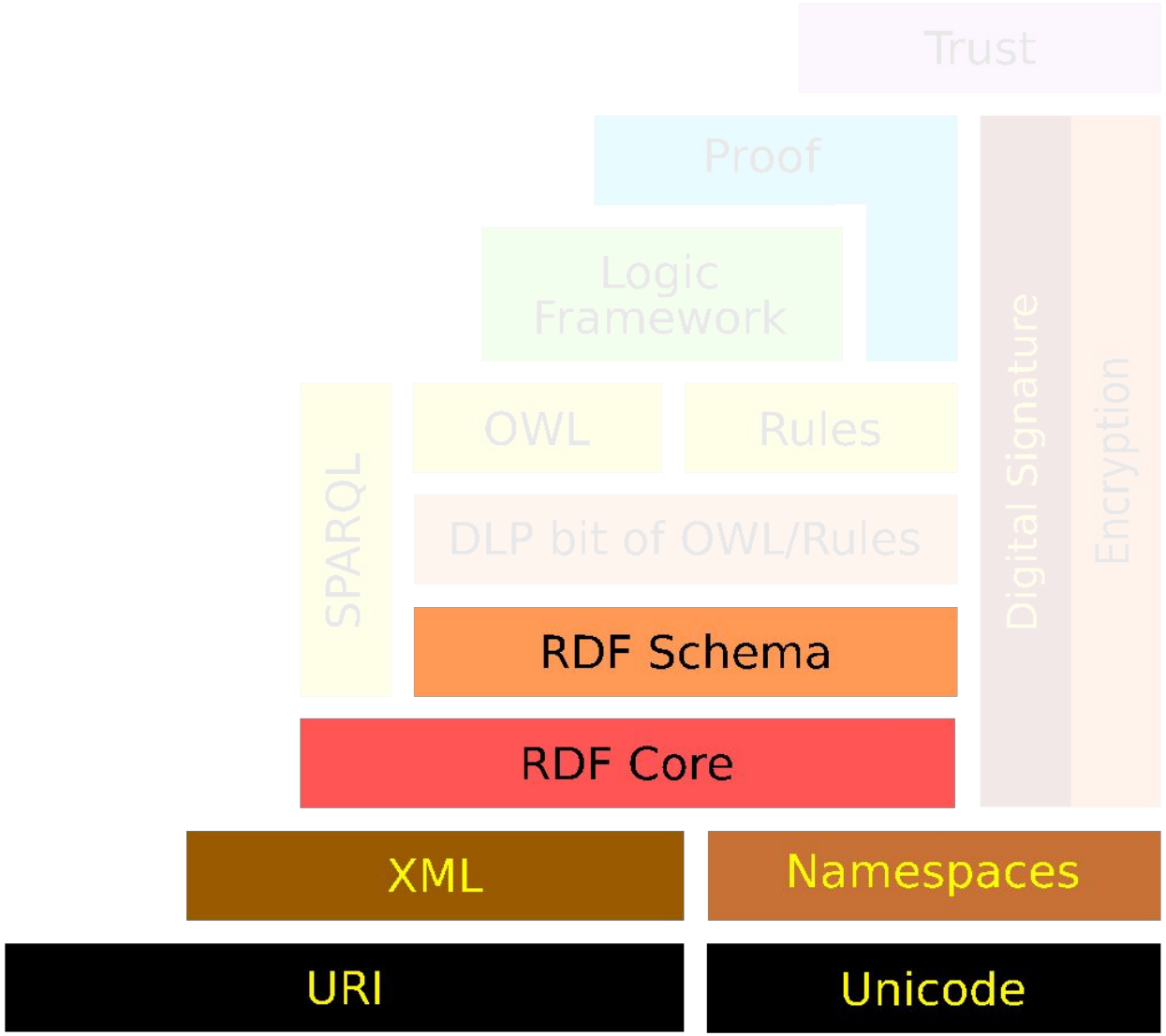
Unicode

Unicode is a computing industry standard for the consistent encoding, representation and handling of text expressed in most of the world's writing systems.

… the latest version of Unicode consists of a repertoire of more than 110,000 characters covering over 100 scripts

Wikipedia

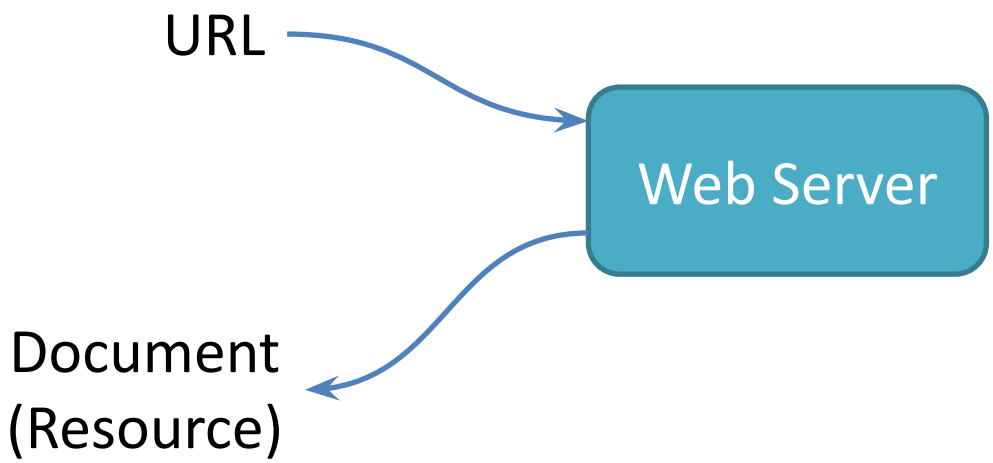
URI



slide by Pedro Szekely

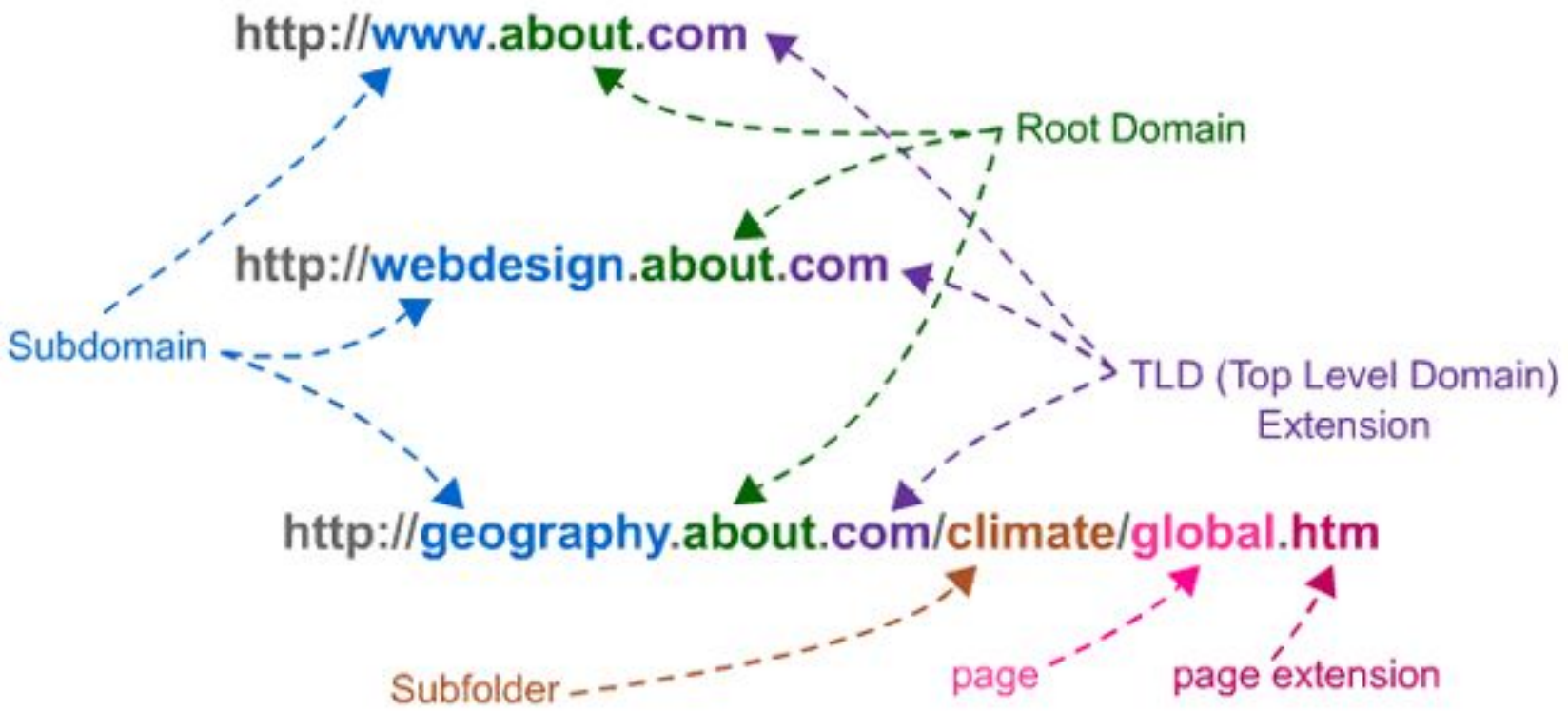
URL: Uniform Resource Locator

A reference to an Internet resource



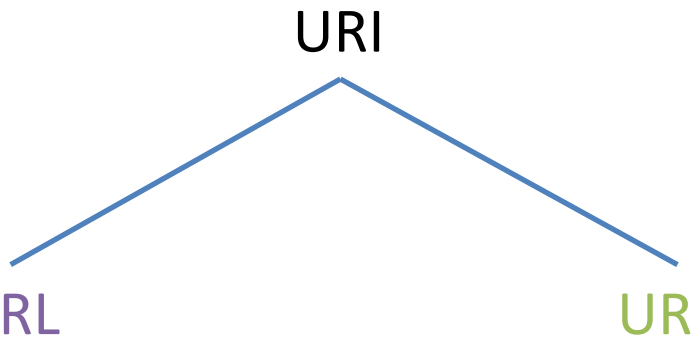
slide by Pedro Szekely

URL: Uniform Resource Locator



http://www.seomoz.org/blog/subfolders-root-domains-linkscape-update-more

URL vs URI



locators

like person's street address

method for finding it

names

like a person's name

item's identity

Can USC Have a URI?

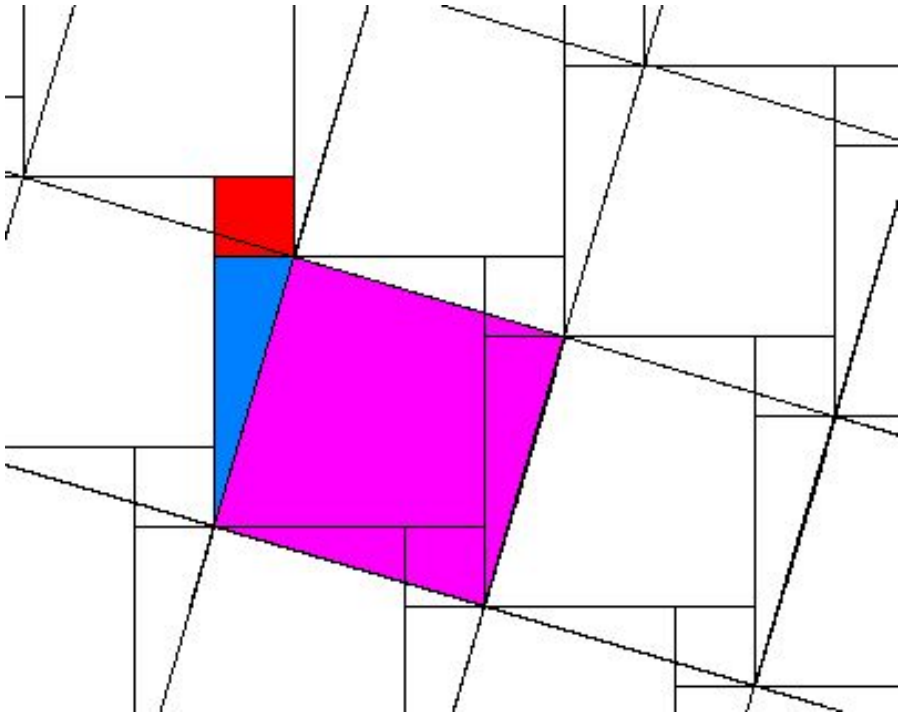
Can USC Have a URI?

http://dbpedia.org/page/University_of_Southern_California

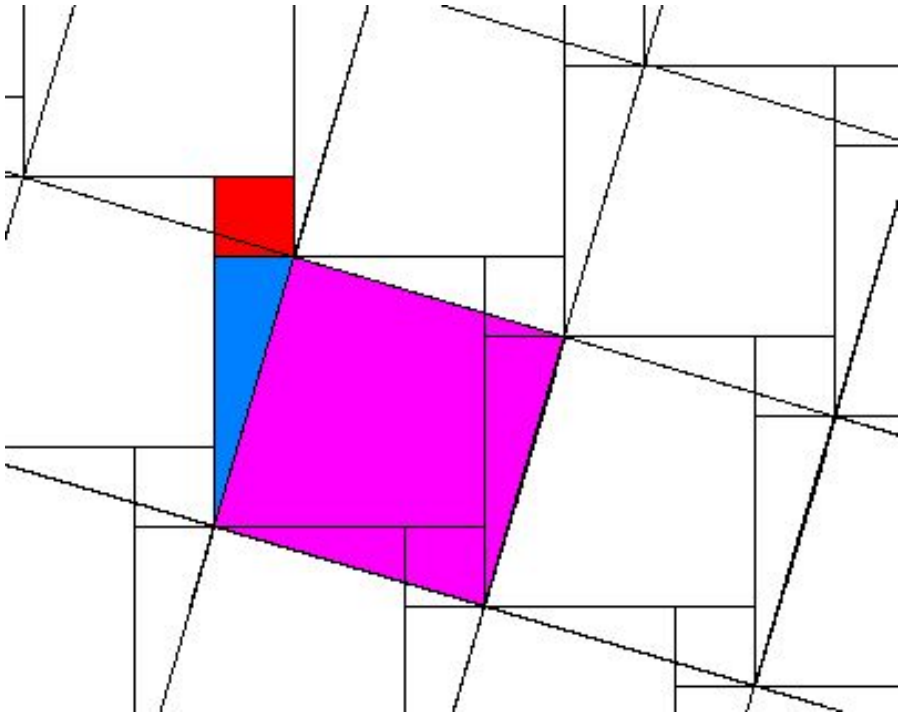
slide by Pedro Szekely

Things can have URIs

Can the Pythagoras Theorem Have a URI?



Can the Pythagoras Theorem Have a URI?



http://www.freebase.com/view/en/pythagorean_theorem

Ideas can have URIs

My Dog: Can He Have a URI?

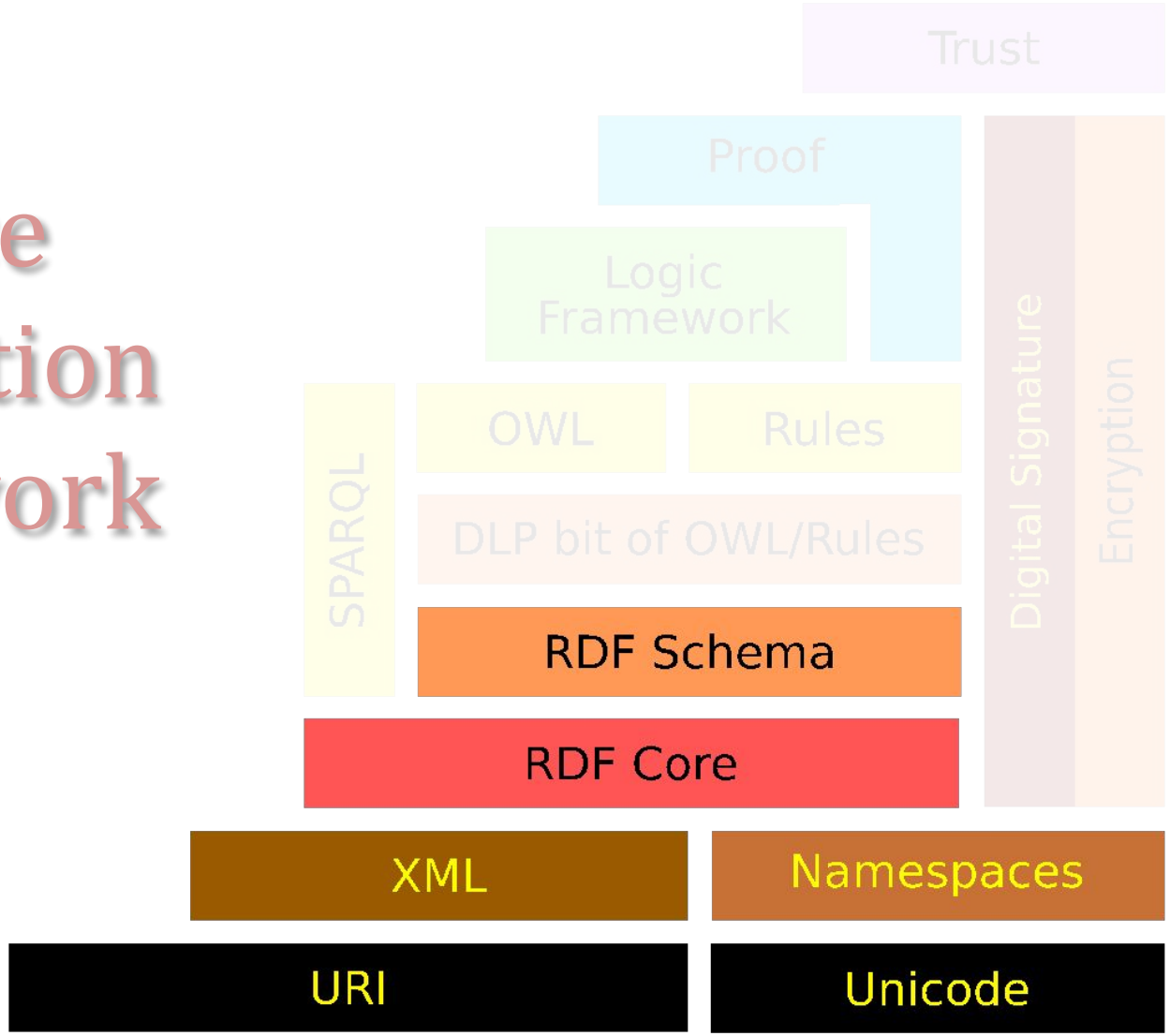
slide by Pedro Szekely

My Dog: Can He Have a URI?

http://szekelys.com/diego

It does not have to be “important” to have a URI

Resource Description Framework



slide by Pedro Szekely

The Resource Description Framework (RDF)
is a language for
representing information about resources
in the World Wide Web

http://www.w3.org/TR/rdf-primer/

slide by Pedro Szekely

Resource Description Framework

Intended for representing metadata about Web resources, such as the title, author, and modification date of a Web document

… also be used to represent information about things that can be *identified* on the Web, even when they cannot be directly *retrieved* on the Web

examples include information about items available from on-line shopping facilities (e.g., prices and availability)

slide by Pedro Szekely

Represent Resources Using URIs

That guy has first name "Pedro"

http://szekelys.com/family#pedro

http://xmlns.com/foaf/0.1/firstName

"Pedro"

Represent Information as Triples

Subject

The resource being described

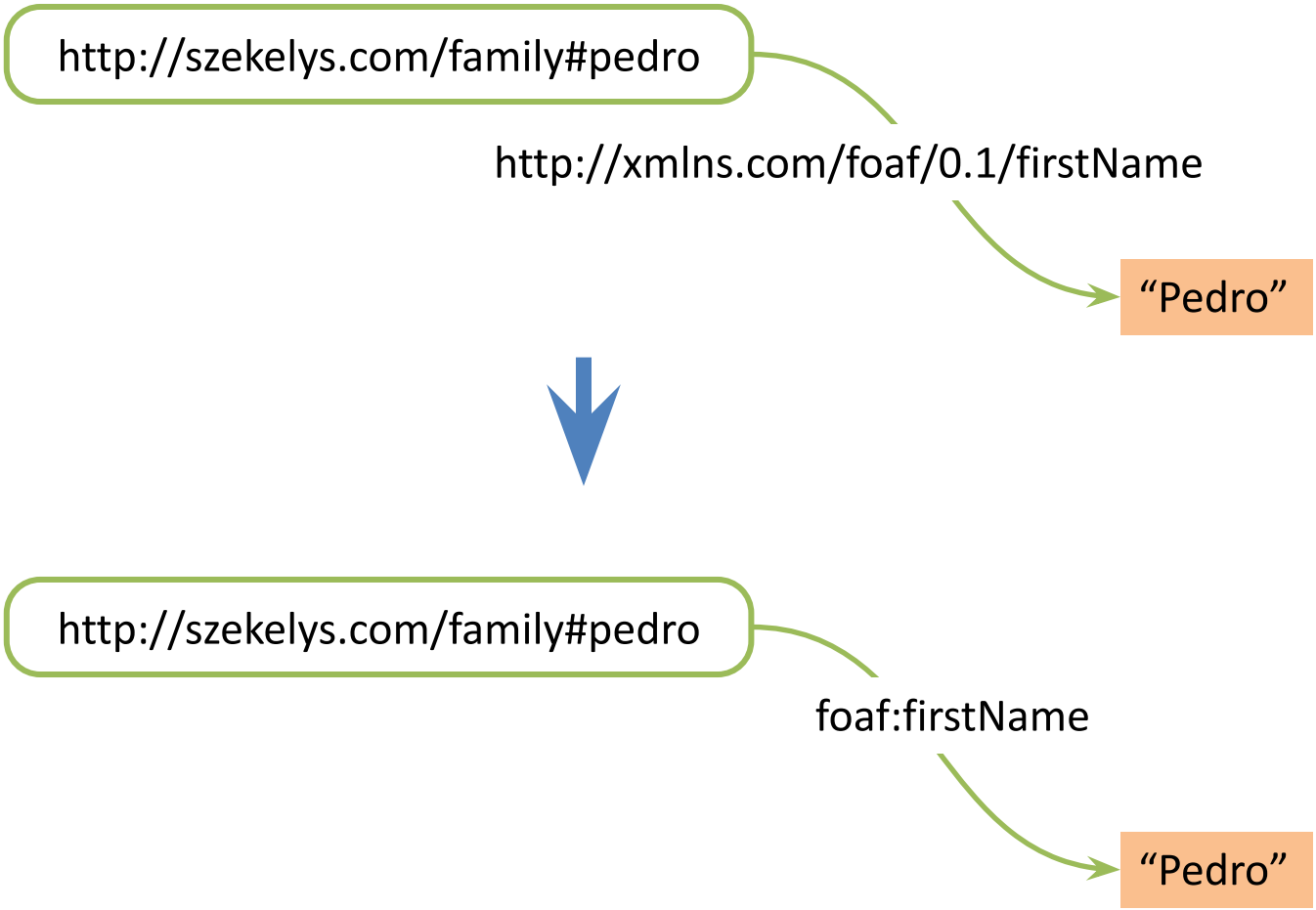
Predicate

A property of the resource

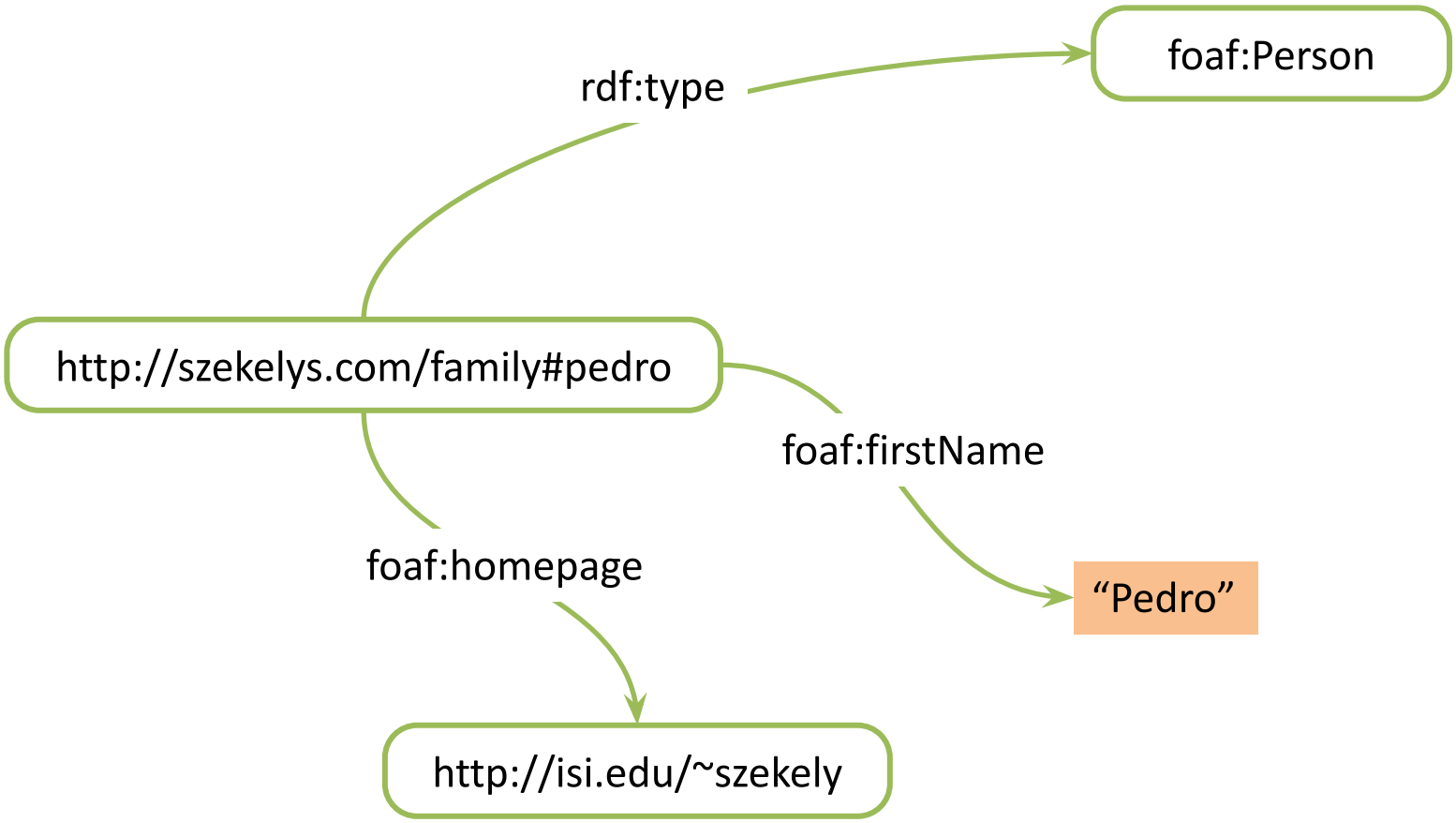
Object

The value of the property

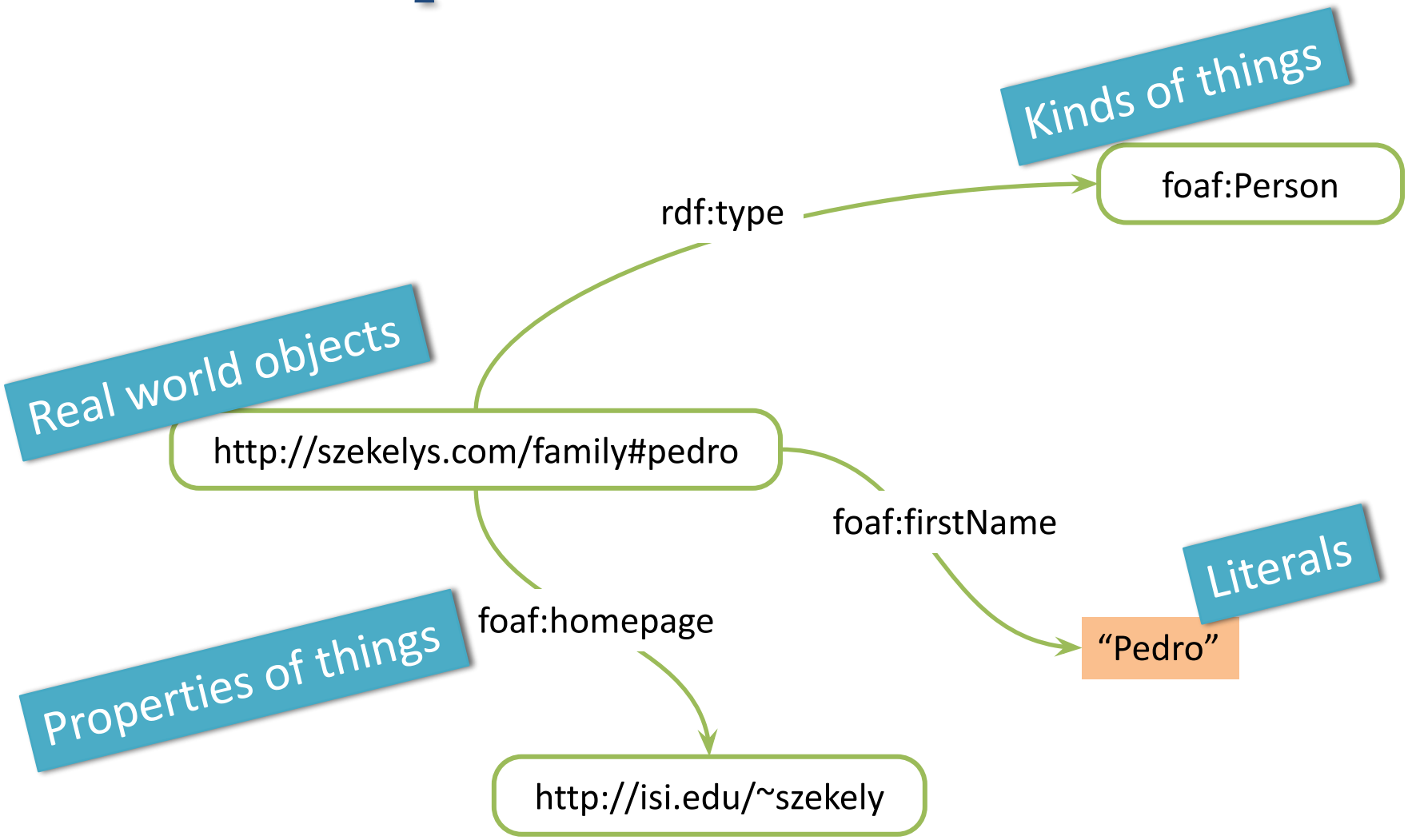
Use Namespaces



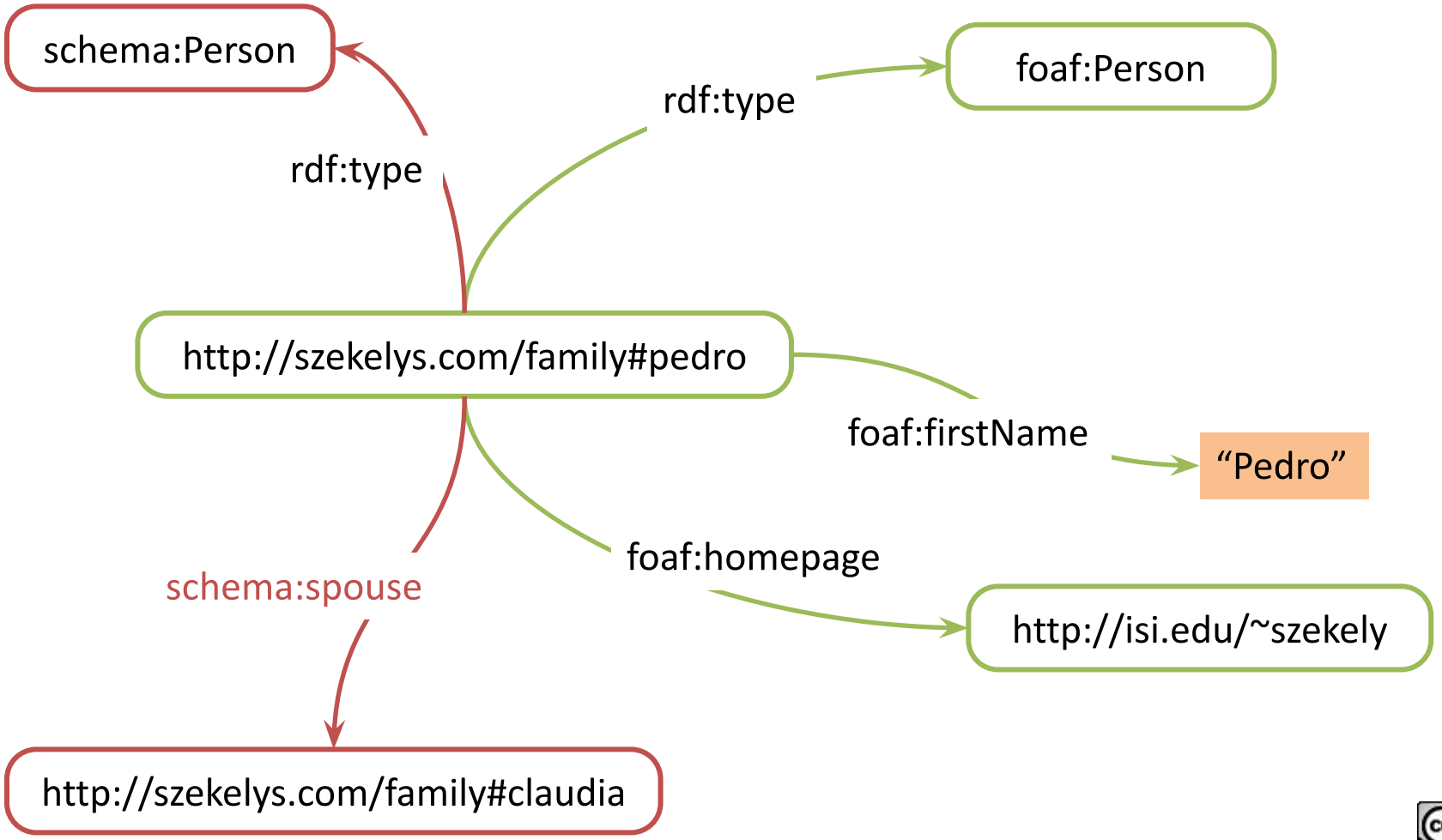
RDF Graphs



RDF Graphs



Mix Vocabularies



Why Use URIs?

- URIs look cool

Why Use URIs?

- URIs look cool
- Precisely identify resources
  - Avoid confusion among different “Jose Lopez”
- Precisely identify properties
  - E.g., name of a company or name of a person
- Provide information about properties
- Look them up on the web

slide by Pedro Szekely

RDF Syntaxes

XML	Leverages XML tools Hard for humans to read
N3, Turtle	Terse RDF Triple Language Human readable format Works with software too
N-Triples	Subset of turtle, supports streaming Standard for large RDF dumps
RDFa	Allows embedding RDF in HTML pages
JSON-LD	RDF in JSON

slide by Pedro Szekely

XML Syntax

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:foaf="http://xmlns.com/foaf/0.1/">

  <rdf:Description rdf:about="http://szekelys.com/family#pedro">
    <foaf:firstName>Pedro</foaf:firstName>
    <foaf:homepage rdf:resource="http://isi.edu/~szekely"/>
  </rdf:Description>

</rdf:RDF>
```

Pedro's homepage is "http://isi.edu/~szekely"

XML Syntax

It's an XML document

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:foaf="http://xmlns.com/foaf/0.1/">

  <rdf:Description rdf:about="http://szekelys.com/family#pedro">
    <foaf:firstName>Pedro</foaf:firstName>
    <foaf:homepage rdf:resource="http://isi.edu/~szekely"/>
  </rdf:Description>

</rdf:RDF>
```

Pedro's homepage is "http://isi.edu/~szekely"

XML Syntax

Here comes some RDF

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:foaf="http://xmlns.com/foaf/0.1/">

  <rdf:Description rdf:about="http://szekelys.com/family#pedro">
    <foaf:firstName>Pedro</foaf:firstName>
    <foaf:homepage rdf:resource="http://isi.edu/~szekely"/>
  </rdf:Description>

</rdf:RDF>
```

Pedro's homepage is "http://isi.edu/~szekely"

XML Syntax

Namespace declarations

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:foaf="http://xmlns.com/foaf/0.1/">

  <rdf:Description rdf:about="http://szekelys.com/family#pedro">
    <foaf:firstName>Pedro</foaf:firstName>
    <foaf:homepage rdf:resource="http://isi.edu/~szekely"/>
  </rdf:Description>

</rdf:RDF>
```

Pedro's homepage is "http://isi.edu/~szekely"

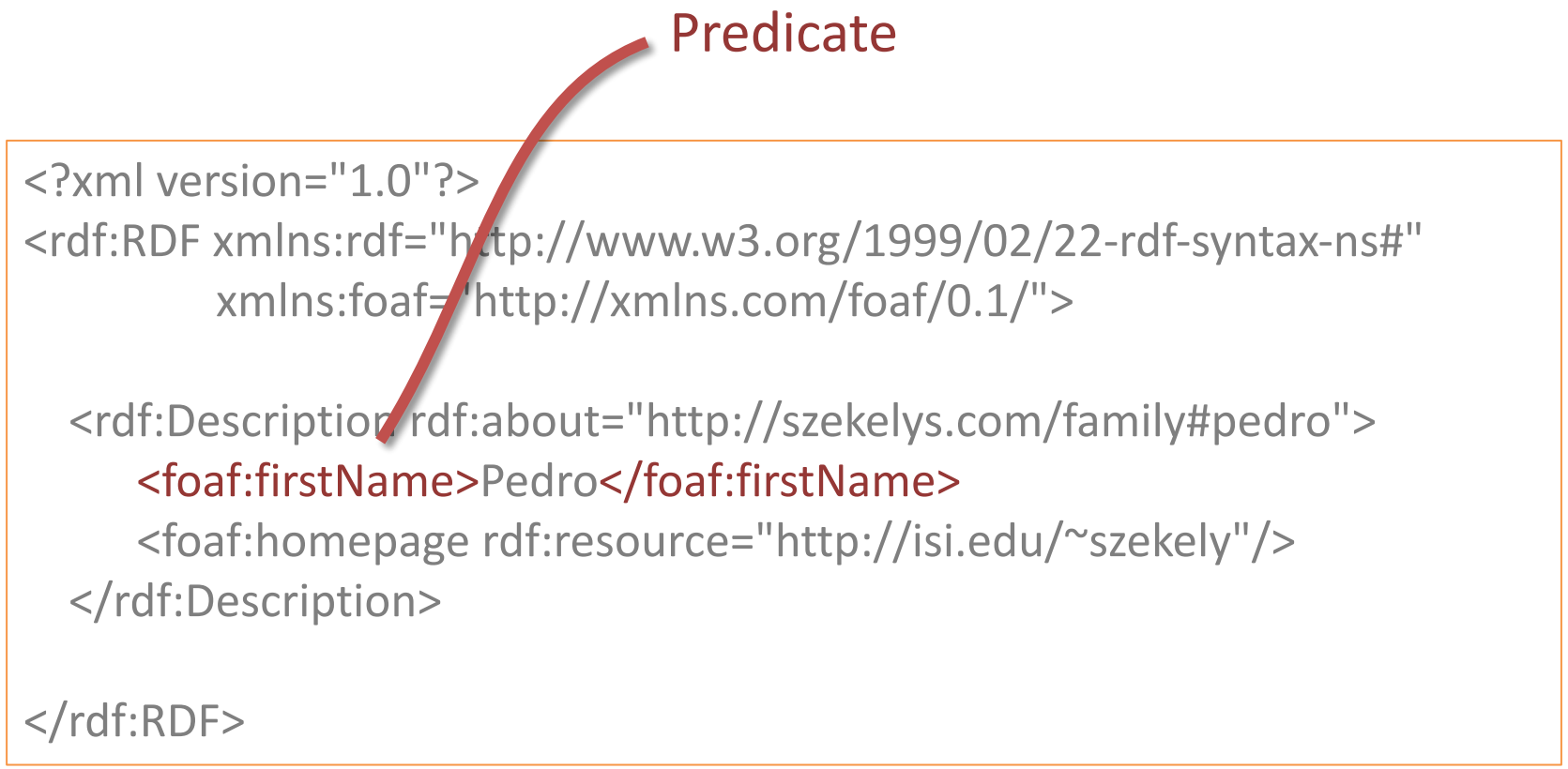
XML Syntax

Subject

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:foaf="http://xmlns.com/foaf/0.1/">

  <rdf:Description rdf:about="http://szekelys.com/family#pedro">
    <foaf:firstName>Pedro</foaf:firstName>
    <foaf:homepage rdf:resource="http://isi.edu/~szekely"/>
  </rdf:Description>

</rdf:RDF>
```

Pedro's homepage is "http://isi.edu/~szekely"

XML Syntax



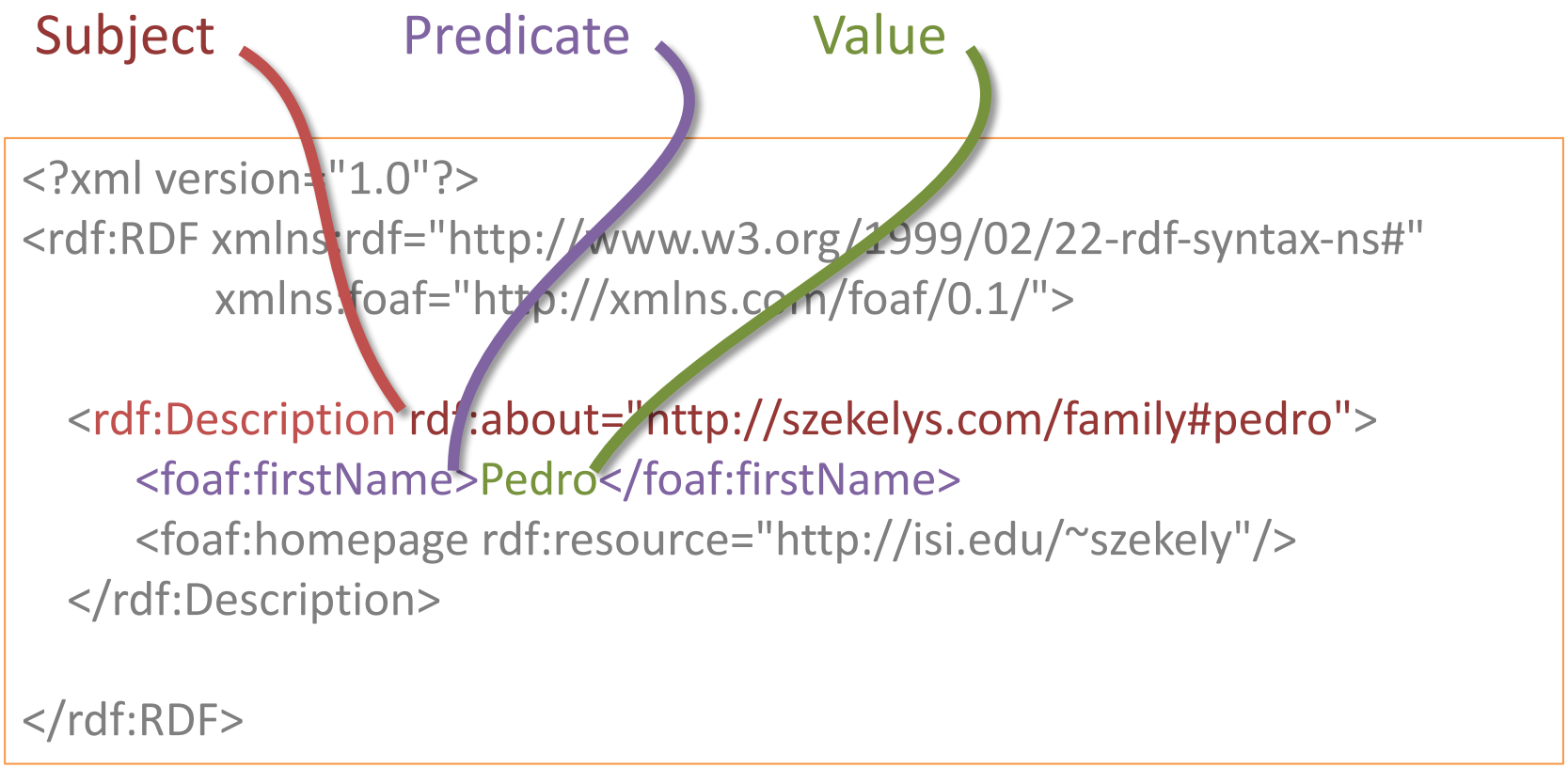
Pedro's homepage is "http://isi.edu/~szekely"

XML Syntax


```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:foaf="http://xmlns.com/foaf/0.1/">

  <rdf:Description rdf:about="http://szekelys.com/family#pedro">
    <foaf:firstName>Pedro</foaf:firstName>
    <foaf:homepage rdf:resource="http://isi.edu/~szekely"/>
  </rdf:Description>

</rdf:RDF>
```

Pedro's homepage is "http://isi.edu/~szekely"

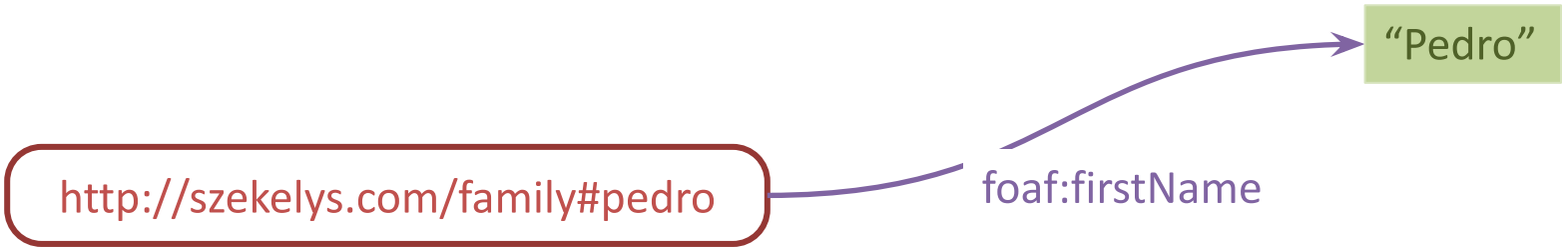
XML Syntax



Pedro's homepage is "http://isi.edu/~szekely"

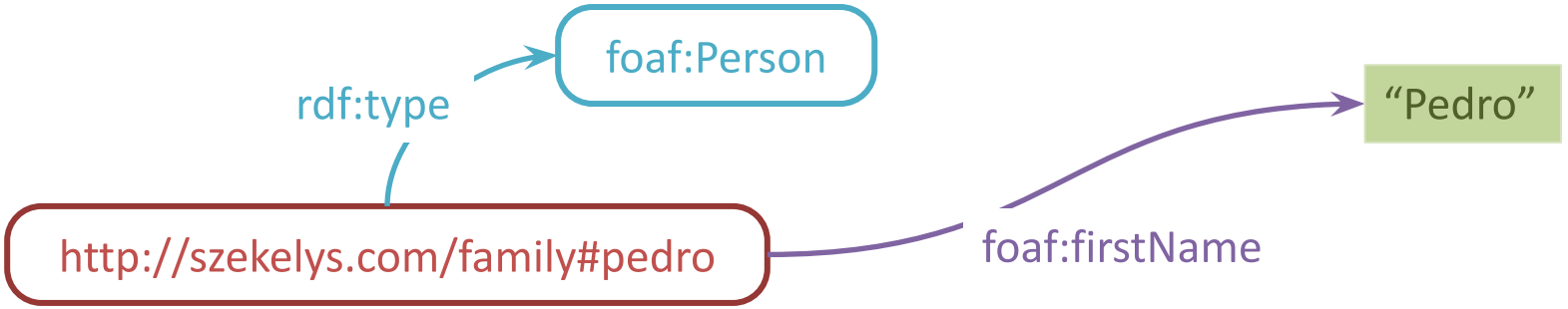
slide by Pedro Szekely

XML Syntax



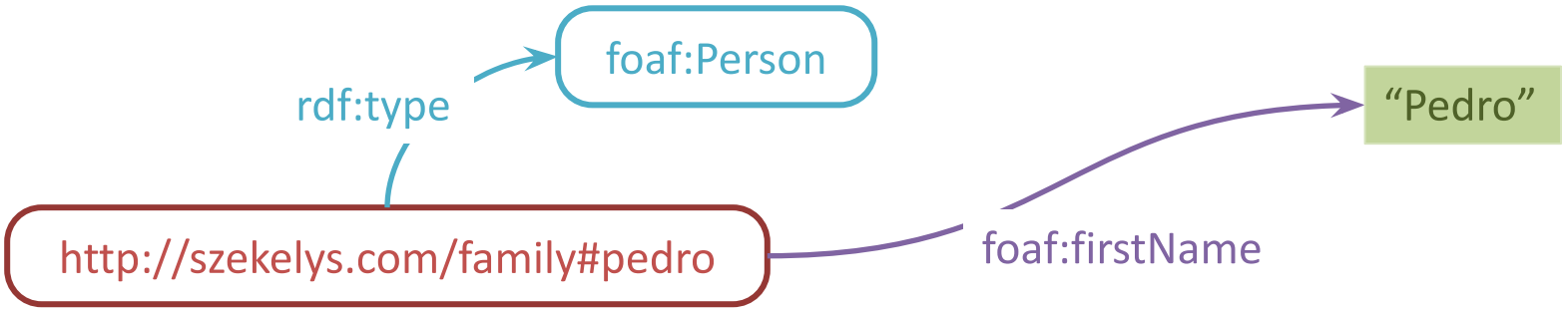

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:foaf="http://xmlns.com/foaf/0.1/">

  <rdf:Description rdf:about="http://szekelys.com/family#pedro">
    <foaf:firstName>Pedro</foaf:firstName>
    <foaf:homepage rdf:resource="http://isi.edu/~szekely"/>
  </rdf:Description>

</rdf:RDF>
```

XML Syntax




```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:foaf="http://xmlns.com/foaf/0.1/">

  <foaf:Person rdf:about="http://szekelys.com/family#pedro">
    <foaf:firstName>Pedro</foaf:firstName>
    <foaf:homepage rdf:resource="http://isi.edu/~szekely"/>
  </foaf:Person>

</rdf:RDF>
```

N3 and Turtle Syntaxes




```
@prefix  rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix  foaf:      <http://xmlns.com/foaf/0.1/> .

<http://szekelys.com/family#pedro> foaf:firstName   "Pedro" .
<http://szekelys.com/family#pedro> rdf:type   foaf:Person .
```

Each triple ends with a dot

More Complex Structures

English

"USC/ISI's address is
4676 Admiralty Way, Marina del Rey, CA 90292"

RDF

```
usc:isi
    schema:address
    "4676 Admiralty Way, Marina del Rey, CA 90292"
    .
```

~~In what city is USC/ISI located?~~

~~Find all universities in California~~

slide by Pedro Szekely

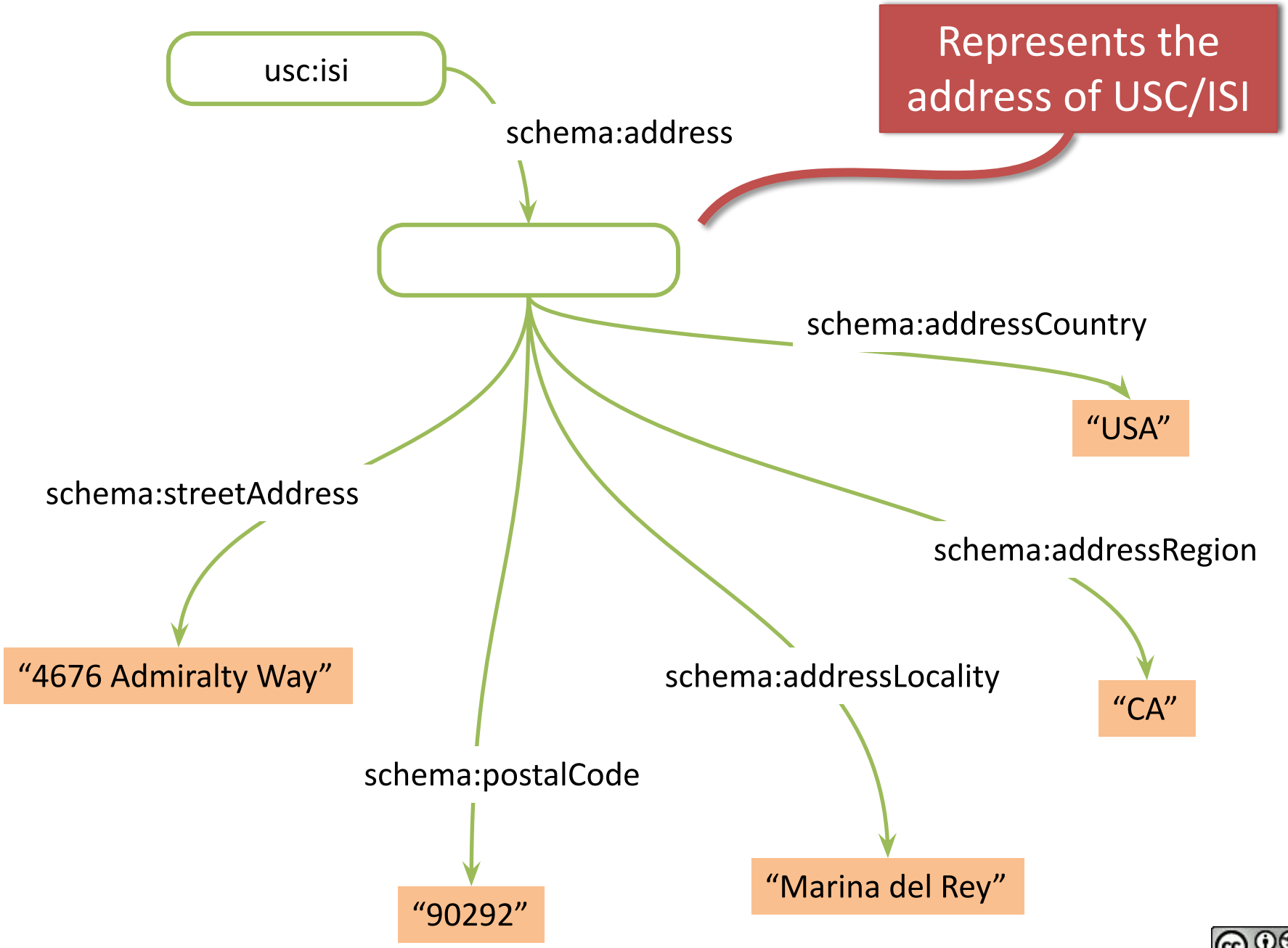
English

"USC/ISI's address is
4676 Admiralty Way, Marina del Rey, CA 90292"

RDF

```
usc:isi
    schema:address
    "4676 Admiralty Way, Marina del Rey, CA 90292"
    .
```

How to represent nested structures?



English

"USC/ISI's address is
4676 Admiralty Way, Marina del Rey, CA 90292"

RDF

```
usc:isi      schema:address      usc:isi-address .

usc:isi-address
     schema:addressCountry   "USA" ;
     schema:addressRegion    "CA";
     schema:addressLocality  "Marina del Rey" ;
     schema:postalCode    "90292" ;
     schema:streetAddress      "4676 Admiralty Way" .
```


slide by Pedro Szekely

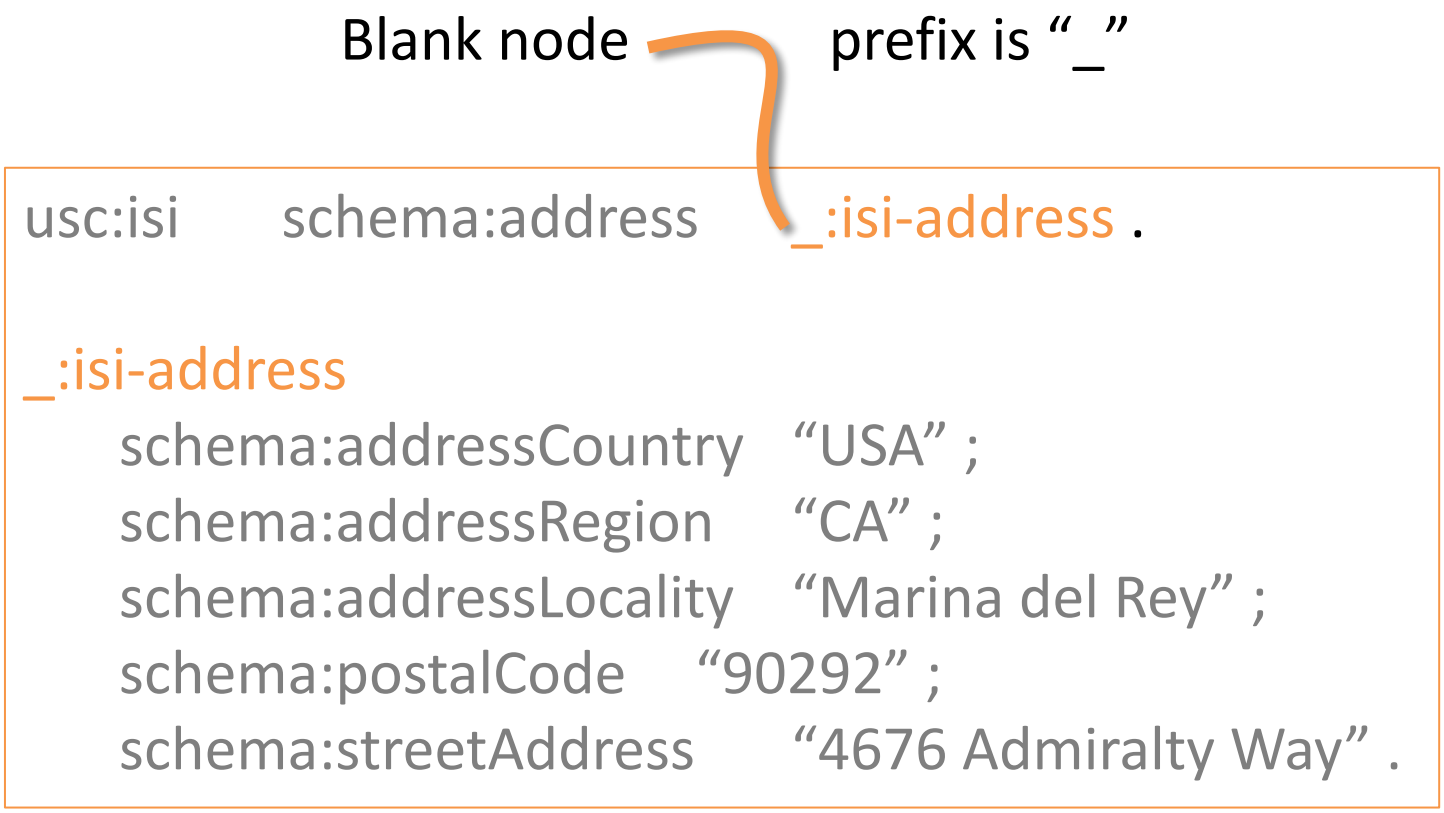
We minted a URI for USC/ISI's address

```
usc:isi      schema:address      usc:isi-address .

usc:isi-address
      schema:addressCountry    "USA" ;
      schema:addressRegion     "CA"
      schema:addressLocality   "Marina del Rey" ;
      schema:postalCode     "90292" ;
      schema:streetAddress      "4676 Admiralty Way" .
```

… but sometimes we don't want to mint URIs

slide by Pedro Szekely

Blank Nodes



… can be improved

slide by Pedro Szekely

What If I Don't Know the URI?

English

"Pedro Szekely lives in Los Angeles"

Blank node

RDF

```
_:pedro
    foaf:firstName          "Pedro" ;
    foaf:lastName           "Szekely" ;
    foaf:mbox               "szekely1401@gmail.com" ;
    schema:addressLocality  "Los Angeles" .
```

… is this useful? …
maybe

Typed Literals

Compact blank node syntax

```
gn:bogota    weather:event  [
     weather:temperature      “10” ;
     weather:date    ”18 June 2012”
     ] .
```

… what is the meaning of the strings?

… how do I specify numbers?

… how about dates?

… how do I specify 10 degrees centigrade?

slide by Pedro Szekely

Typed Literals

```
gn:bogota     weather:event  [
      weather:temperature
"10"^^<http://www.w3.org/2001/XMLSchema#integer> ;
      weather:date                    "18 June 2012" ;
      ] .
```

URI specifies the type

Typed Literals

```
gn:bogota     weather:event   [
      weather:temperature
"10"^^<http://www.w3.org/2001/XMLSchema#integer> ;
      weather:date       "18 June 2012" ;
      weather:date       "2012-06-18"^^xsd:date ;
] .
```

URI from the XML Schema namespace are popular

… No set of predefined types defined in RDF

… Software that consumes RDF must process types

… XSD types commonly used

Reification

Why Do We Need Reification?

English

"On June 19 2012, Claudia said that
Pedro's email address is szekely1401@gmail.com"

RDF

<http://szekelys.com/family#pedro> foaf:mbox <szekely1401@gmail.com>

Correct? ……..

… No!

We need to make a statement about a statement

Reification

English

"On June 19 2012, Claudia said that
Pedro's email address is szekely1401@gmail.com"

RDF

```
_:s  rdf:type  rdf:Statement .
_:s  rdf:subject    <http://szekelys.com/family#pedro> .
_:s  rdf:predicate        foaf:mbox .
_:s  rdf:object      <szekely1401@gmail.com> .

_:s  dcterms:date"2012-06-19"^^xsd:date .
_:s  dcterms:creator   <http://uniandes.edu.co/faculty#claudiaj> .
```

RDF Syntax

Turtle Syntax

http://www.w3.org/TR/2011/WD-turtle-20110809/

Turtle URIs aka IRI

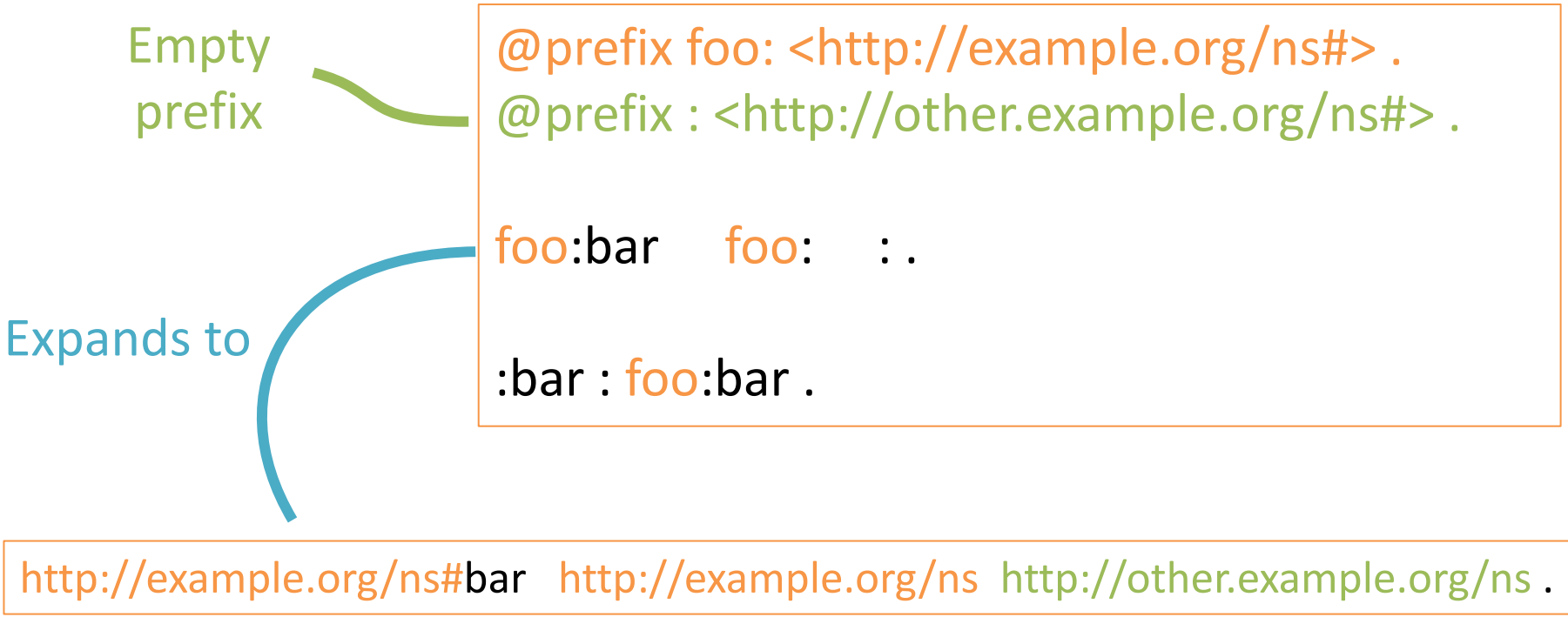
URIs are in <>

Absolute

Relative to the base document

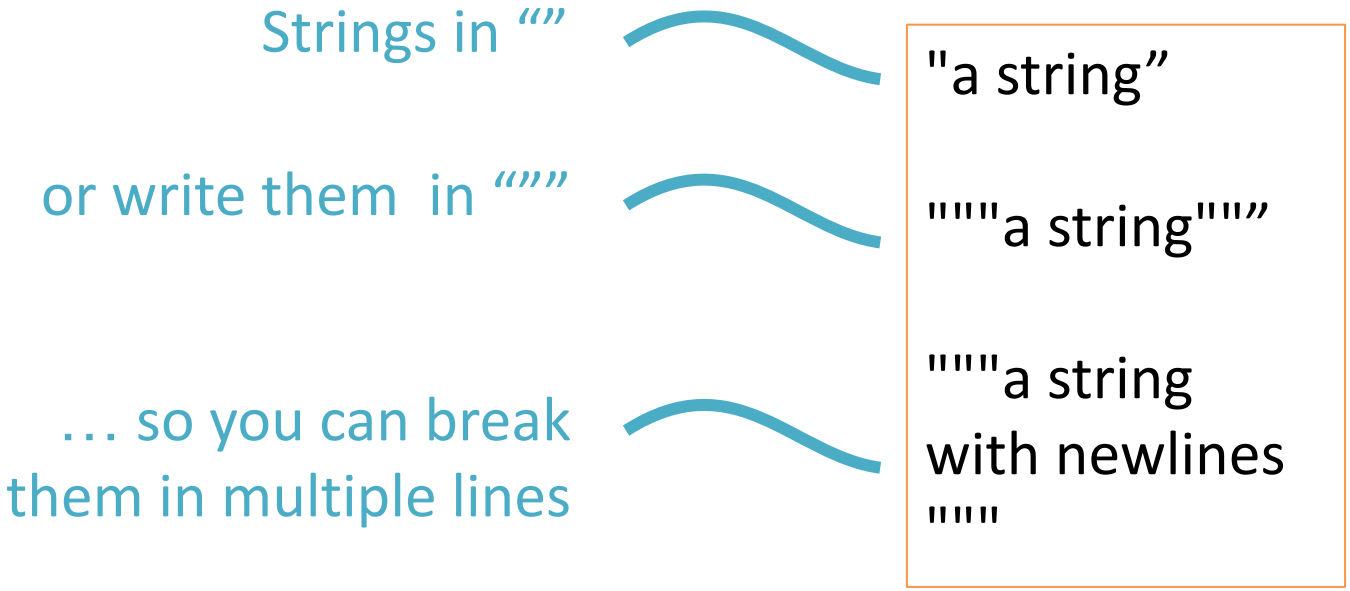
```
<http://example.org/path/>
<http://example.org/path/#fragment>

</path>
<#fragment>
<>
```

Turtle Prefixes (Namespaces)



slide by Pedro Szekely

Turtle Literals



Turtle Literals

"That Seventies Show"

"That Seventies Show"@en

"Cette Série des Années Soixante-dix"@fr

"Cette Série des Années Septante"@fr-be

"mylexicaldata"^^<http://example.org/my/datatype>

"""10"""^^xsd:decimal

Kinds of literals:

- untyped (“London” equivalent to “London”^^xsd:string)
- language tag
- data type (with a URI)

slide by Pedro Szekely

Turtle Blank Nodes

Blank nodes

```
_:me
_:a1234
[]
```

two blank nodes

Examples

```
_:me    foaf:firstName “Pedro” .
_:me    foaf:lastName “Szekely” .
```

```
[]    foaf:firstName “Pedro” .
[]    foaf:lastName “Szekely” .
```

```
[     foaf:firstName “Pedro” ;
      foaf:lastName “Szekely” ;
]
```

Turtle Base URI

RDF document stored at http://isi.edu/szekely/example.rdf

```
</path>
<#fragment>
<>
```

</path>	→	<http://isi.edu/szekely/example/path>
<#fragment>	→	<http://isi.edu/szekely/example/#fragment>
<>	→	<http://isi.edu/szekely/example>

Turtle Base URI

```
# this is a complete turtle document
# In-scope base URI is the document URI at this point
<a1> <b1> <c1> .

@base <http://example.org/ns/> .
# In-scope base URI is http://example.org/ns/ at this point
<a2> <http://example.org/ns/b2> <c2> .

@base <foo/> .
# In-scope base URI is http://example.org/ns/foo/ at this point
<a3> <b3> <c3> .

@prefix : <bar#> .
:a4 :b4 :c4 .

@prefix : <http://example.org/ns2#> .
:a5 :b5 :c5 .
```

<a2> is <http://example.org/ns/a2>

<a3> is <http://example.org/ns/foo/a3>

:a4 is <http://example.org/ns/foo/bar#a4>

:a5 is <http://example.org/ns2#a5>

slide by Pedro Szekely

Turtle "Type" Shortcut

You can write:

```
<http://szekelys.com/family#pedro> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> foaf:Person .
```

You can abbreviate it to:

```
<http://szekelys.com/family#pedro>    rdf:type      foaf:Person .
```

Or even better:

```
<http://szekelys.com/family#pedro>    a      foaf:Person .
```


slide by Pedro Szekely

Turtle Literals Revisited

Integers:

```
-5
0
1
10
+1
```

Floating Point:

```
0.0
1.0
1.23456789012345678 9
-5.0
1.3e2
10e0
-12.5e10
```

Booleans:

```
true
false
```

Integers the hard way:

```
"-5"^^xsd:integer
"10"^^<http://www.w3.org/2001/XMLSchema#integer>
```

Booleans the hard way:

```
"true"^^xsd:boolean
```

Floating point the hard way:

```
"1.3e2"^^xsd:double
"-5.0"^^<http://www.w3.org/2001/XMLSchema#decimal>
```


slide by Pedro Szekely

Turtle Triple Abbreviations

```
:subject   :predicate   :object-1 .
:subject   :predicate   :object-2 .
```

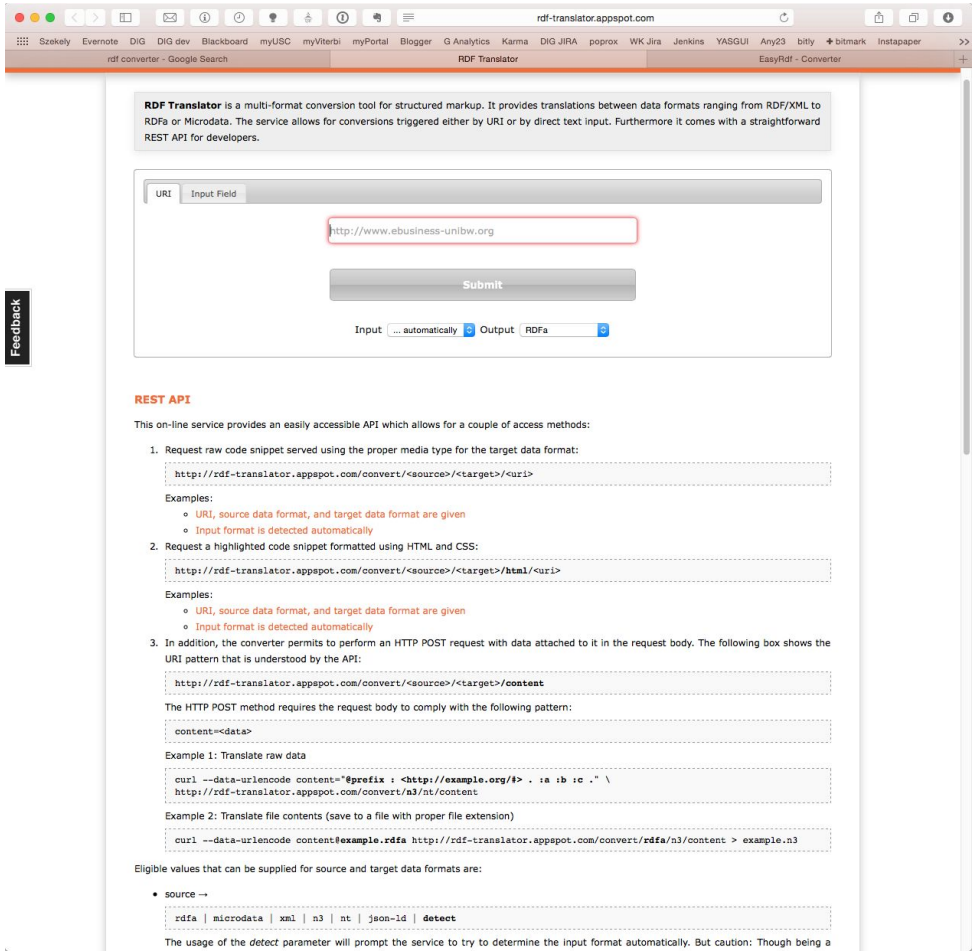
```
:subject   :predicate   :object-1, :object-2 .
```

```
:subject   :predicate-1   :object-1 .
:subject   :predicate-2   :object-2 .
```

```
:subject
   :predicate-1   :object-1 ;
   :predicate-2   :object-2 .
```

RDF Converter

http://rdf-translator.appspot.com



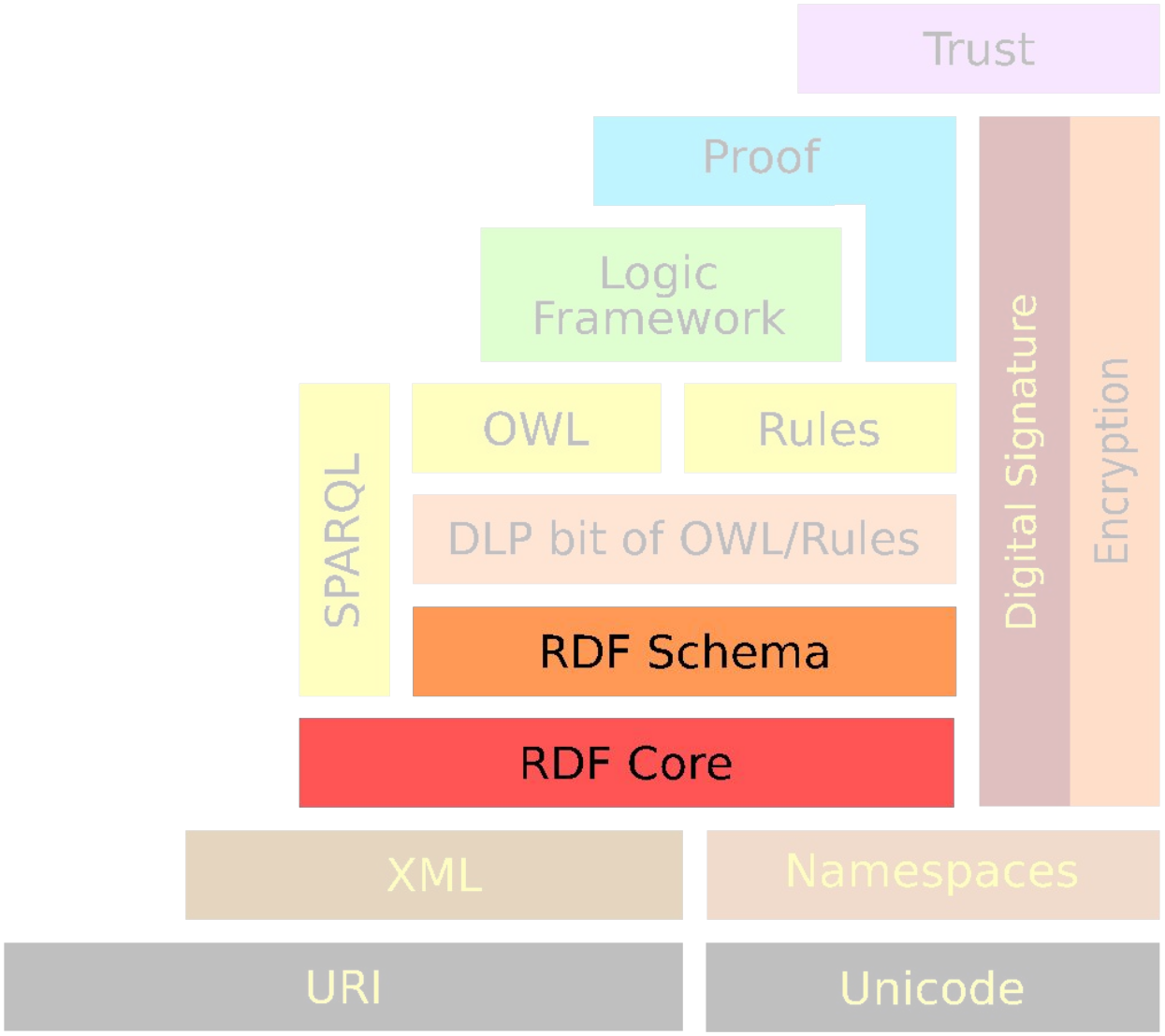
slide by Pedro Szekely

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
>
  <foaf:Person rdf:about="http://szekelys.com/family#pedro">
    <foaf:firstName>Pedro</foaf:firstName>
    <foaf:homepage rdf:resource="http://isi.edu/~szekely"/>
  </foaf:Person>
</rdf:RDF>
```

RDF Schema

slide by Pedro Szekely

Semantic Web Layer Cake



RDF Schema

RDF Schema is the language for defining RDF vocabularies

It specifies the RDF inference rules: the triples that are implied by the triples you have

RDF Schema Vocabulary

xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

Classes

rdfs:Resource

rdfs:Class

rdfs:Literal

rdfs:Datatype

rdf:XMLLiteral

rdf:Property

Properties

rdfs:range

rdfs:domain

rdf:type

rdfs:subClassOf

rdfs:subPropertyOf

rdfs:label

rdfs:comment

Utility Properties

rdfs:seeAlso

rdfs:isDefinedBy

rdf:value

rdfs:Resource

All things described by RDF are called *resources*, and are instances of the class rdfs:Resource

rdfs:Class

the class of resources that are RDF classes

rdfs:Class is a resource

rdfs:Class is an instance of rdfs:Class

rdfs:Literal, rdfs:Datatype

class rdfs:Literal is the class of literal values such as strings and integers

typed literals

class rdfs:Datatype is the class of typed literals

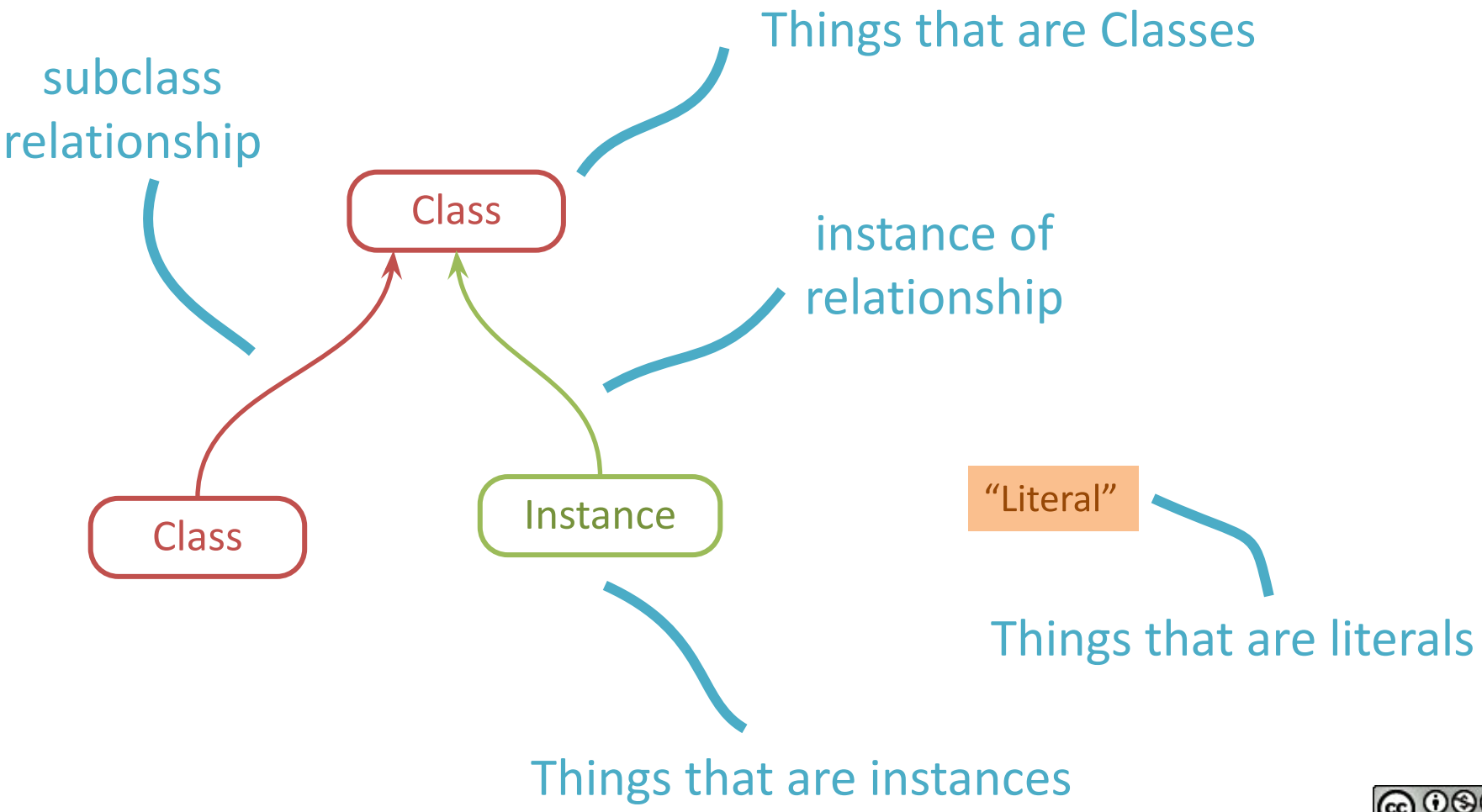
"13"^^xsd:int

plain literals

there is no class of plain literals

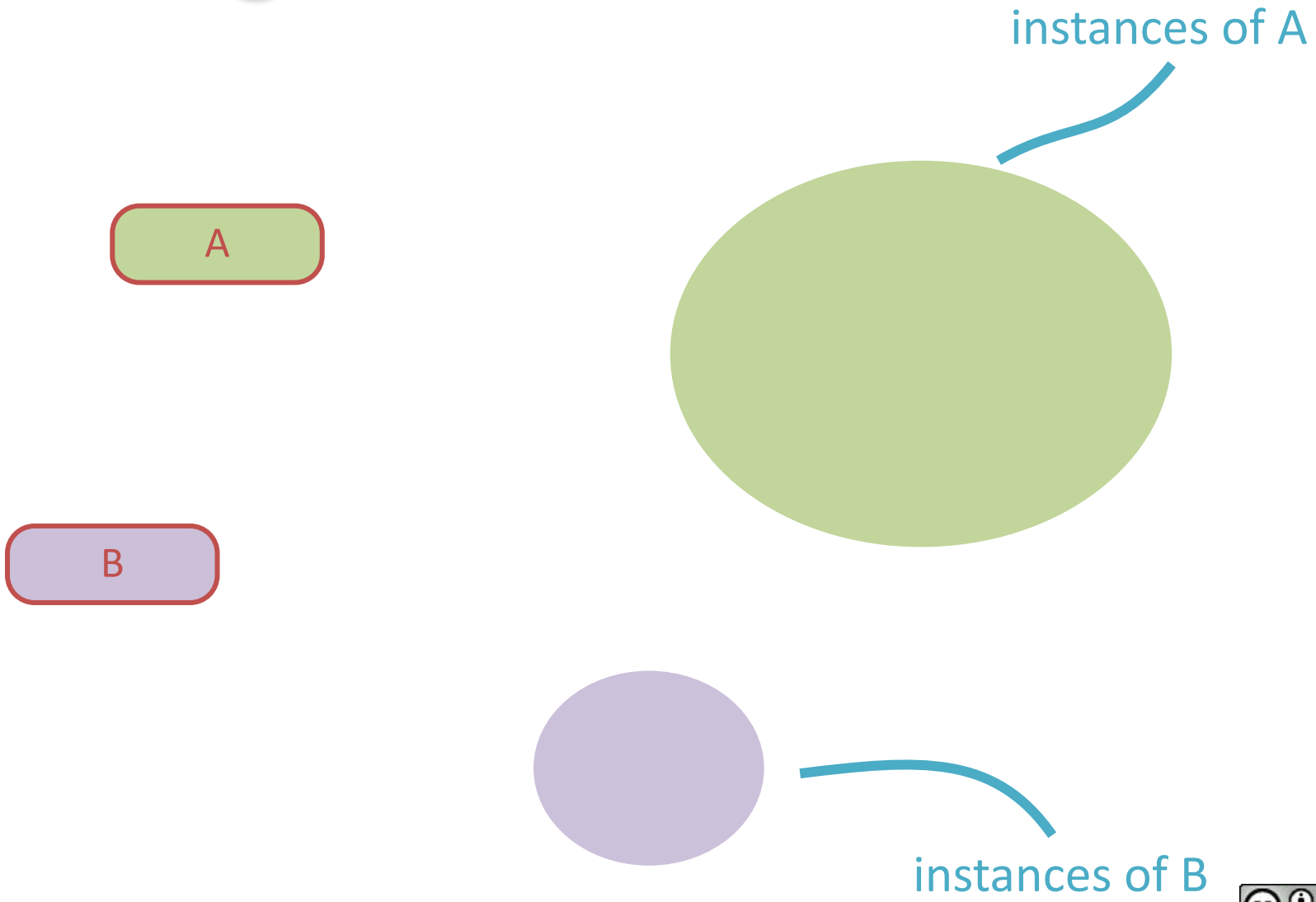
"Bob"

Notation and Conventions



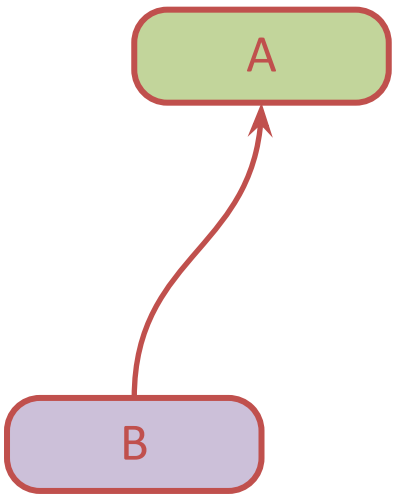
slide by Pedro Szekely

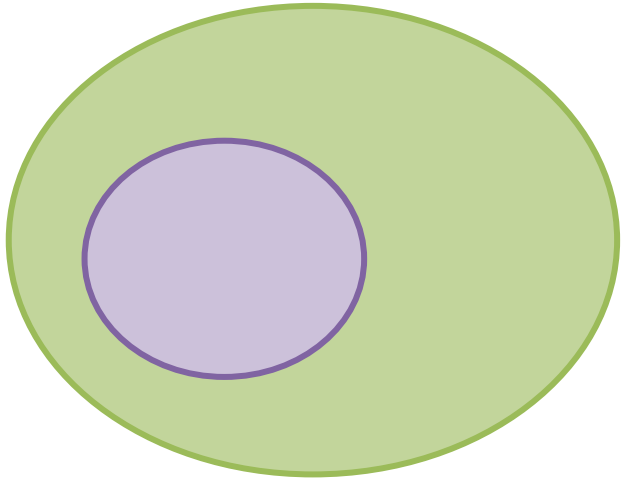
Meaning of Subclass



slide by Pedro Szekely

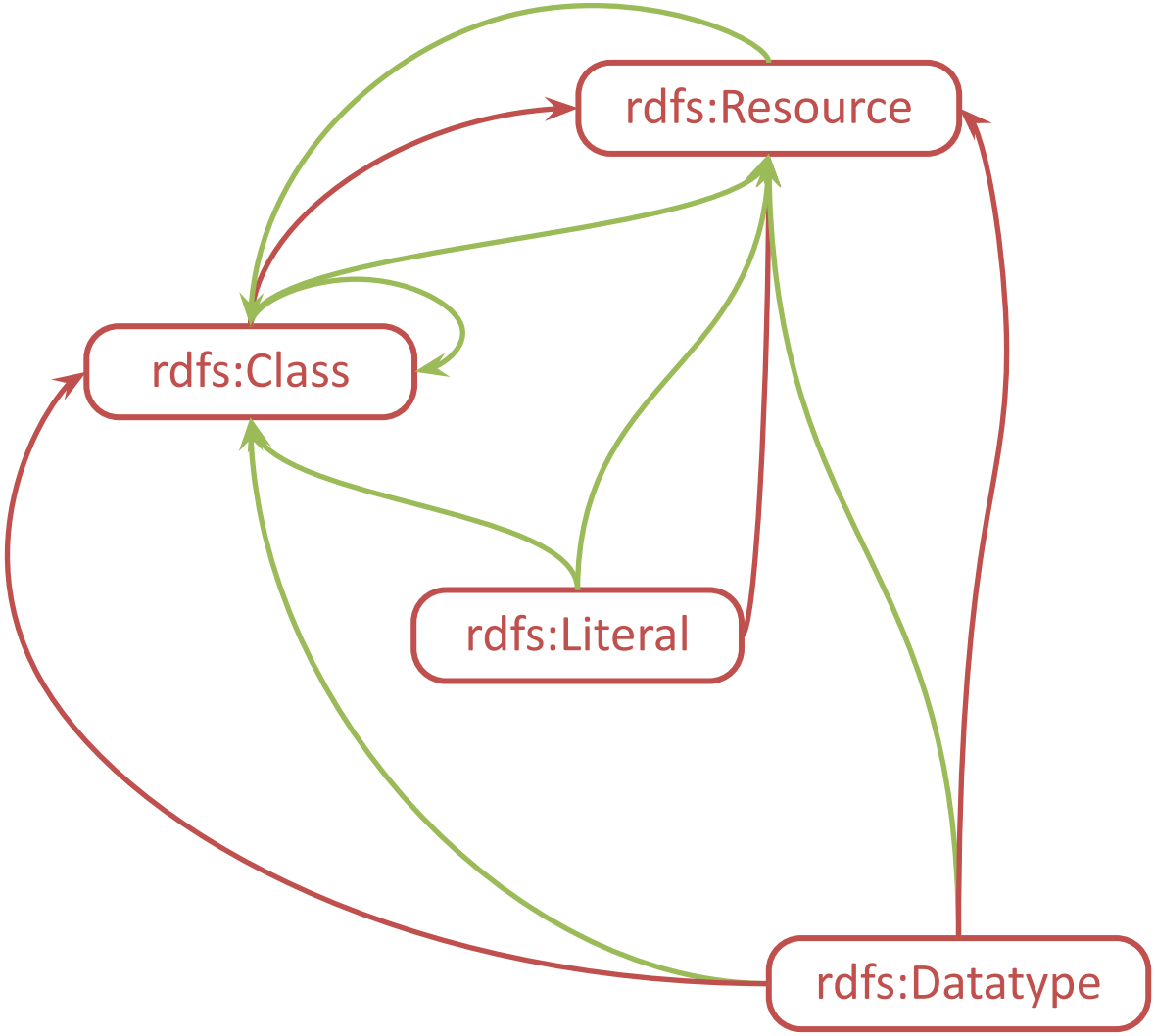
Meaning of Subclass





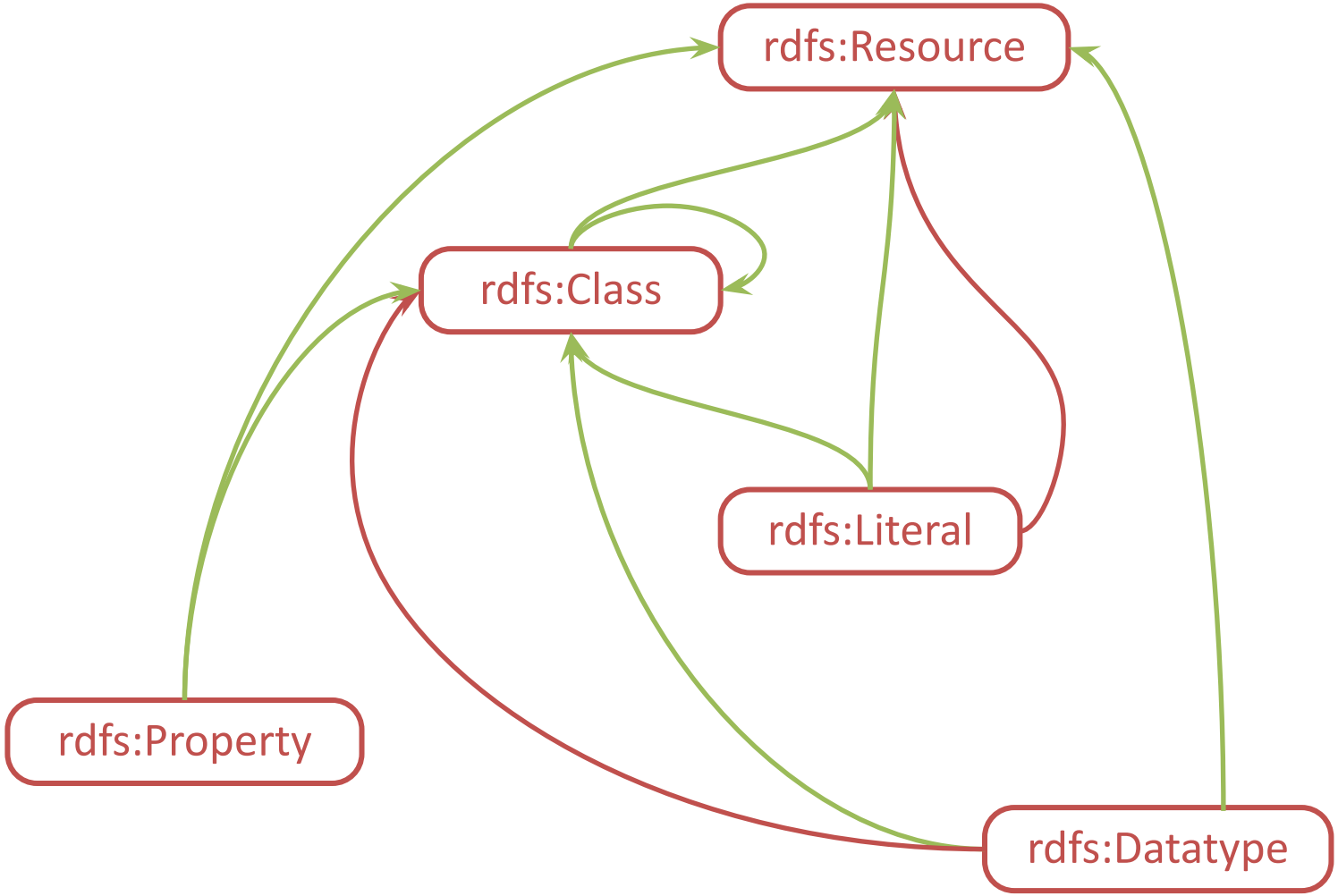
if B is a subclass of A,
the instances of B are a subset of the instances of A

The Story So Far

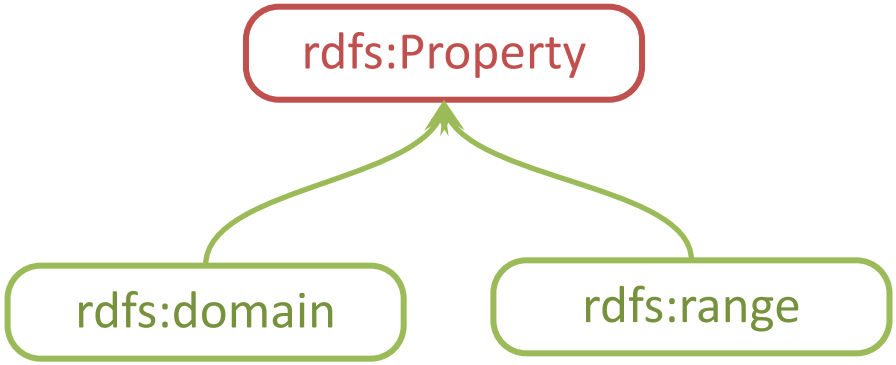


rdfs:Property

the class of resources that are RDF properties

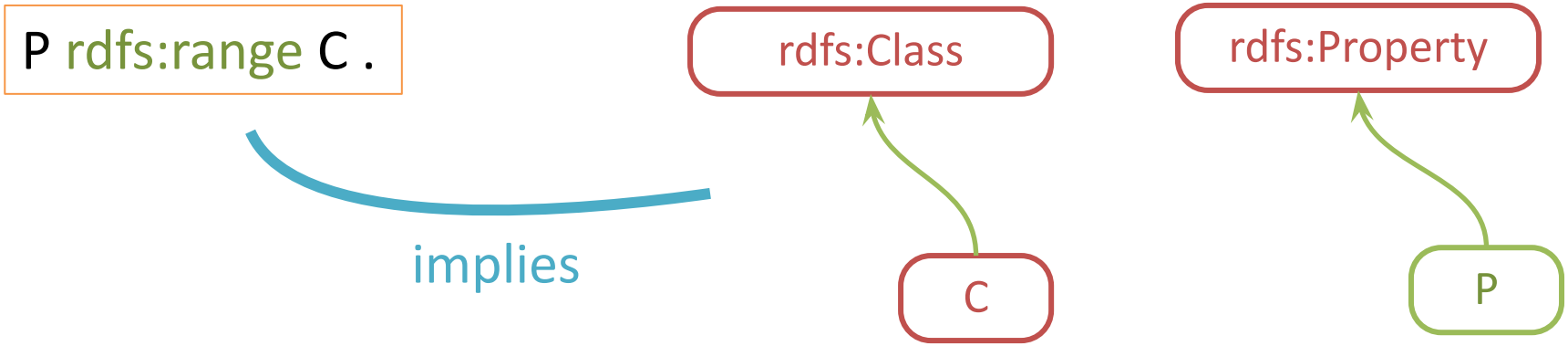


rdfs:domain, rdfs:range



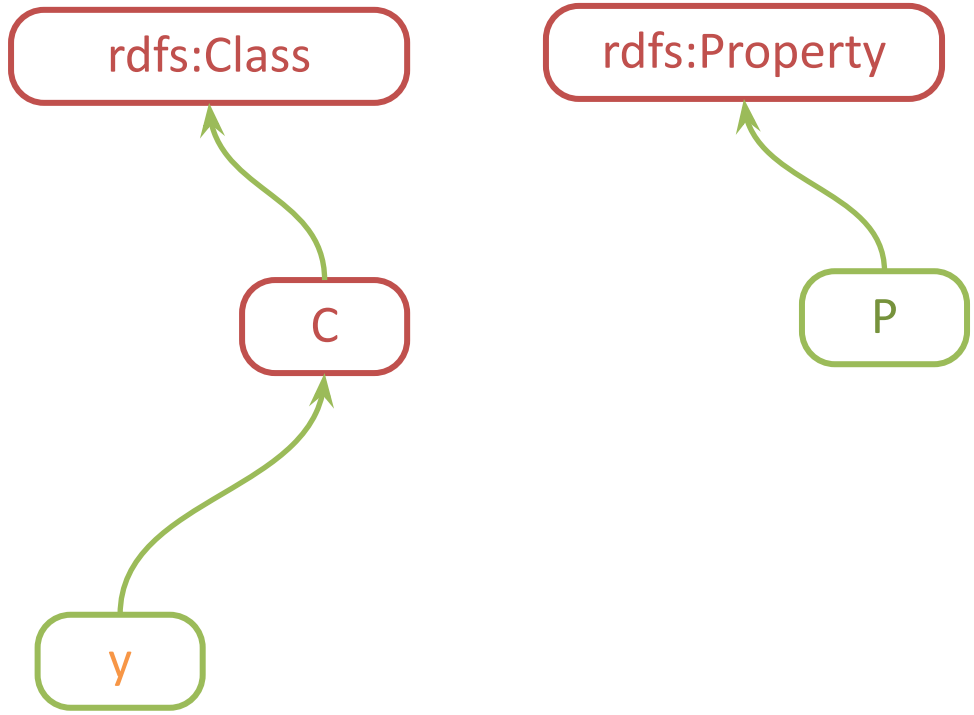
rdfs:domain and rdfs:range are instances of rdfs:property

Semantics of rdfs:range



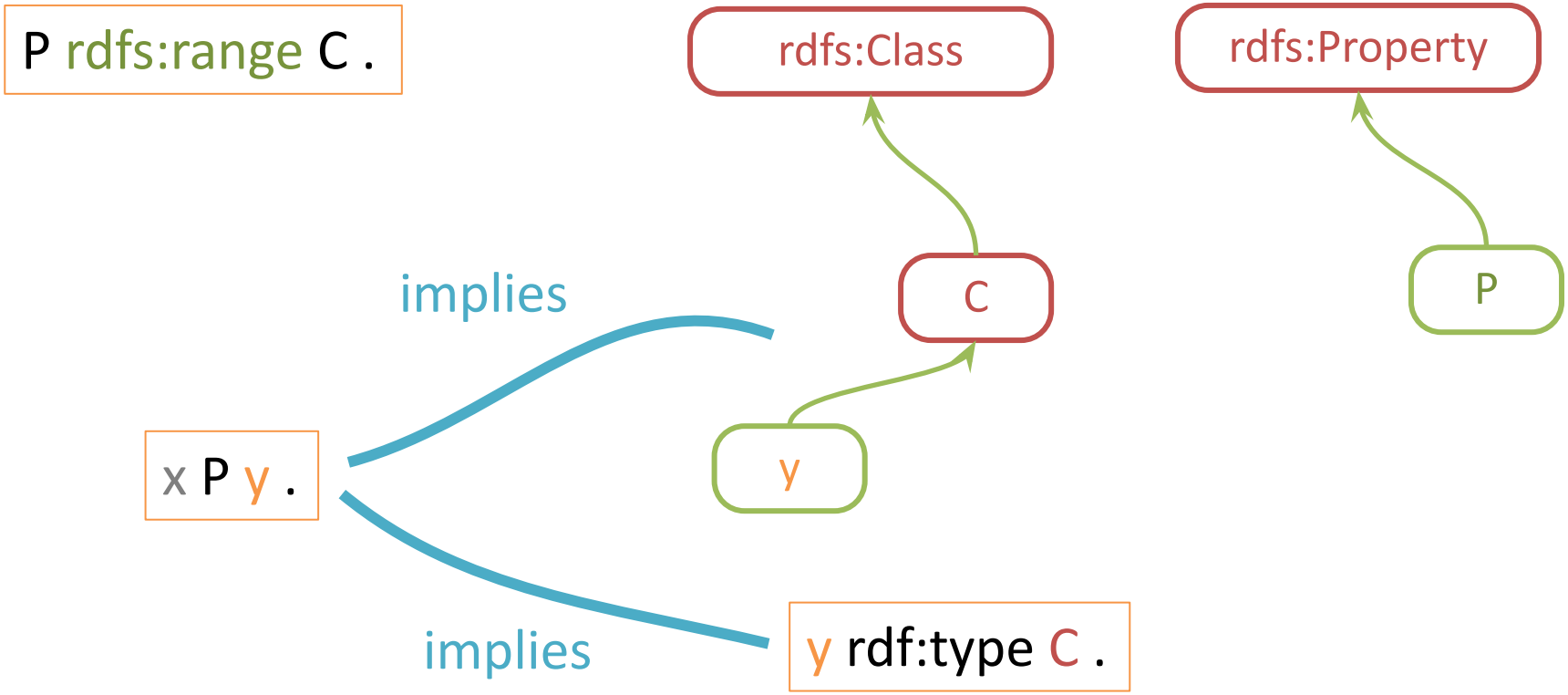
Semantics of rdfs:range

P rdfs:range C .



x P y .

Semantics of rdfs:range



slide by Pedro Szekely

rdfs:range Example

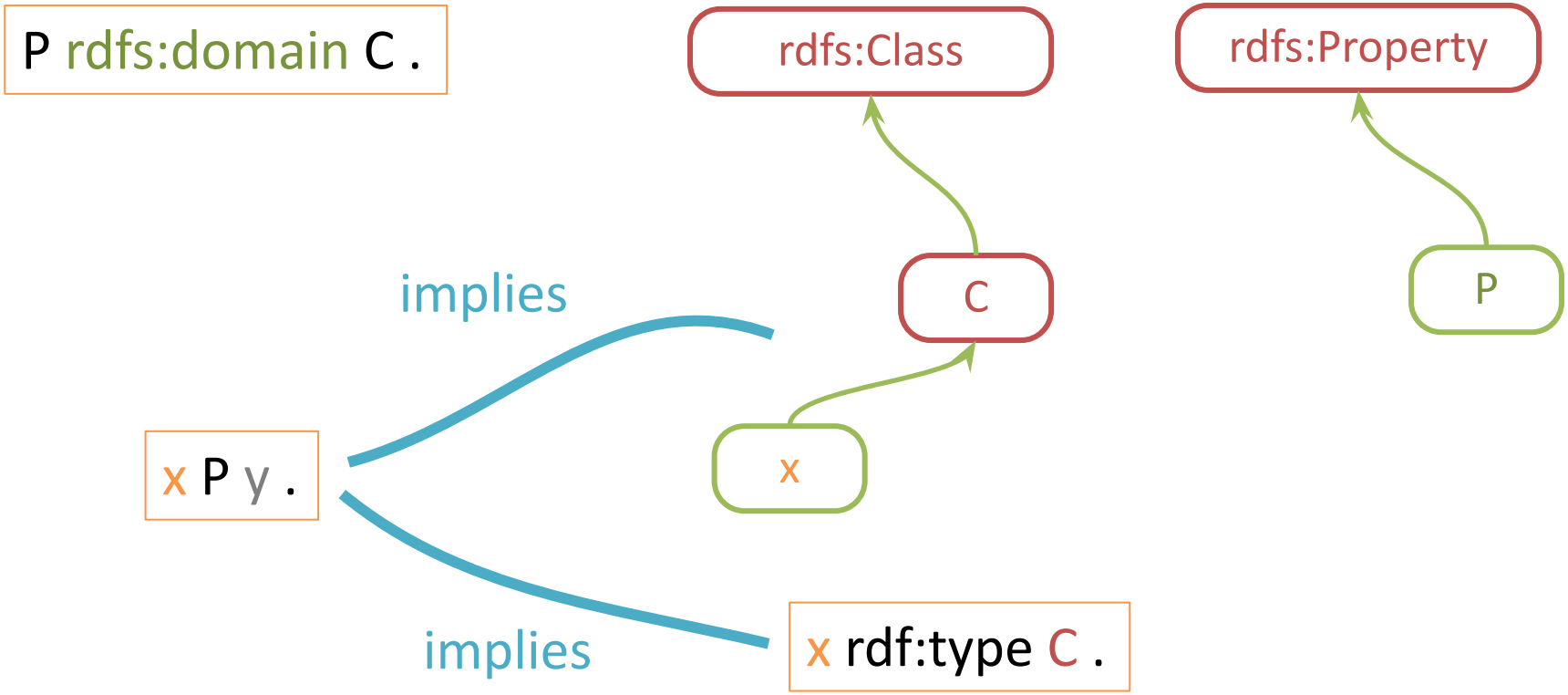
```
foaf:based_near     a rdf:Property;
        rdfs:domain  foaf:Person;
        rdfs:range     wgs84:SpatialThing .
```

```
ex:pedro    foaf:based_near    dbpedia:Los_Angeles .
```

implies

```
dbpedia:Los_Angeles    rdf:type    wgs84:SpatialThing  .
```

Semantics of rdfs:domain



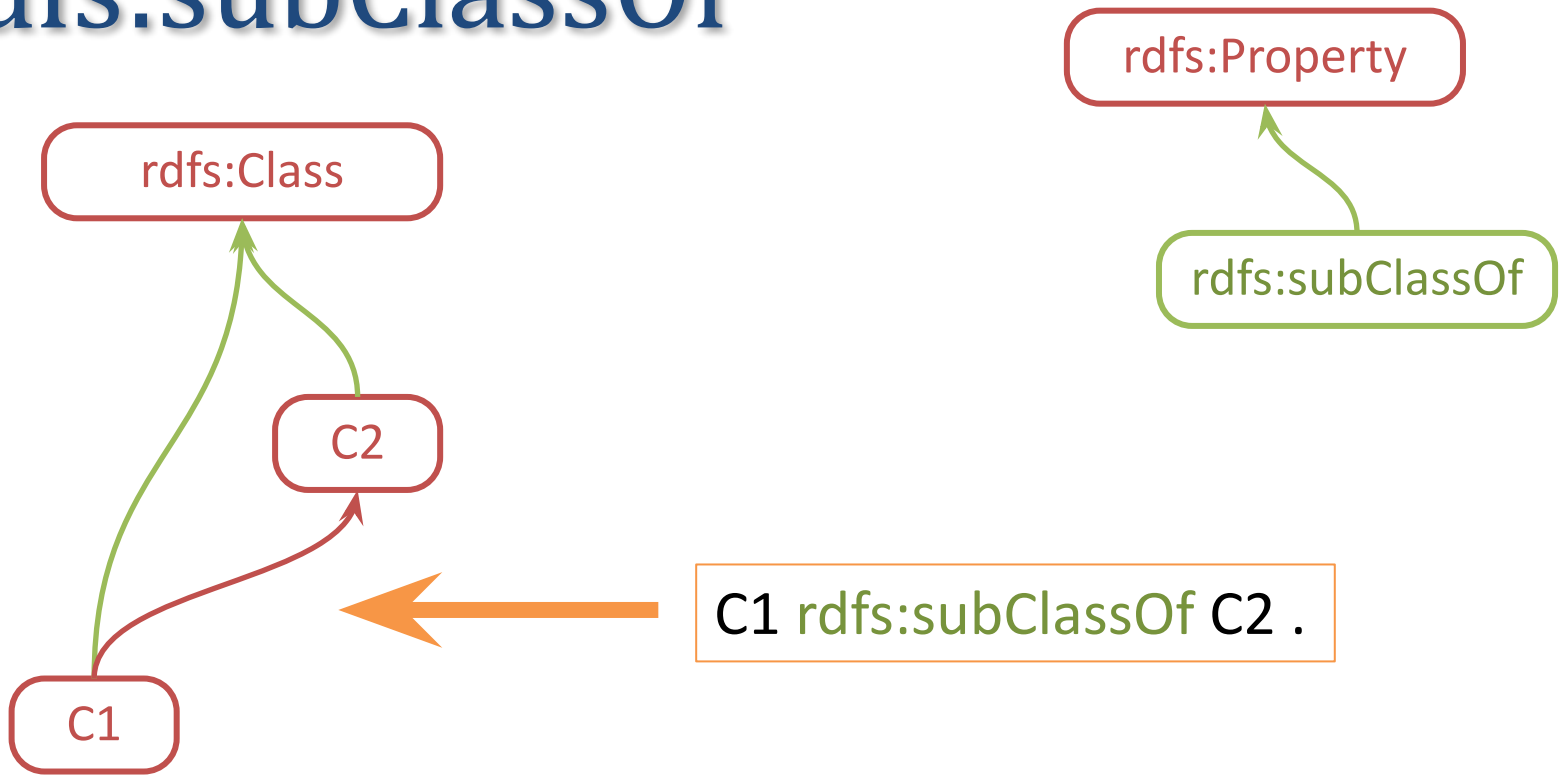
Multiple rdfs:domain, rdfs:range

P rdfs:domain C1 .

P rdfs:domain C2 .

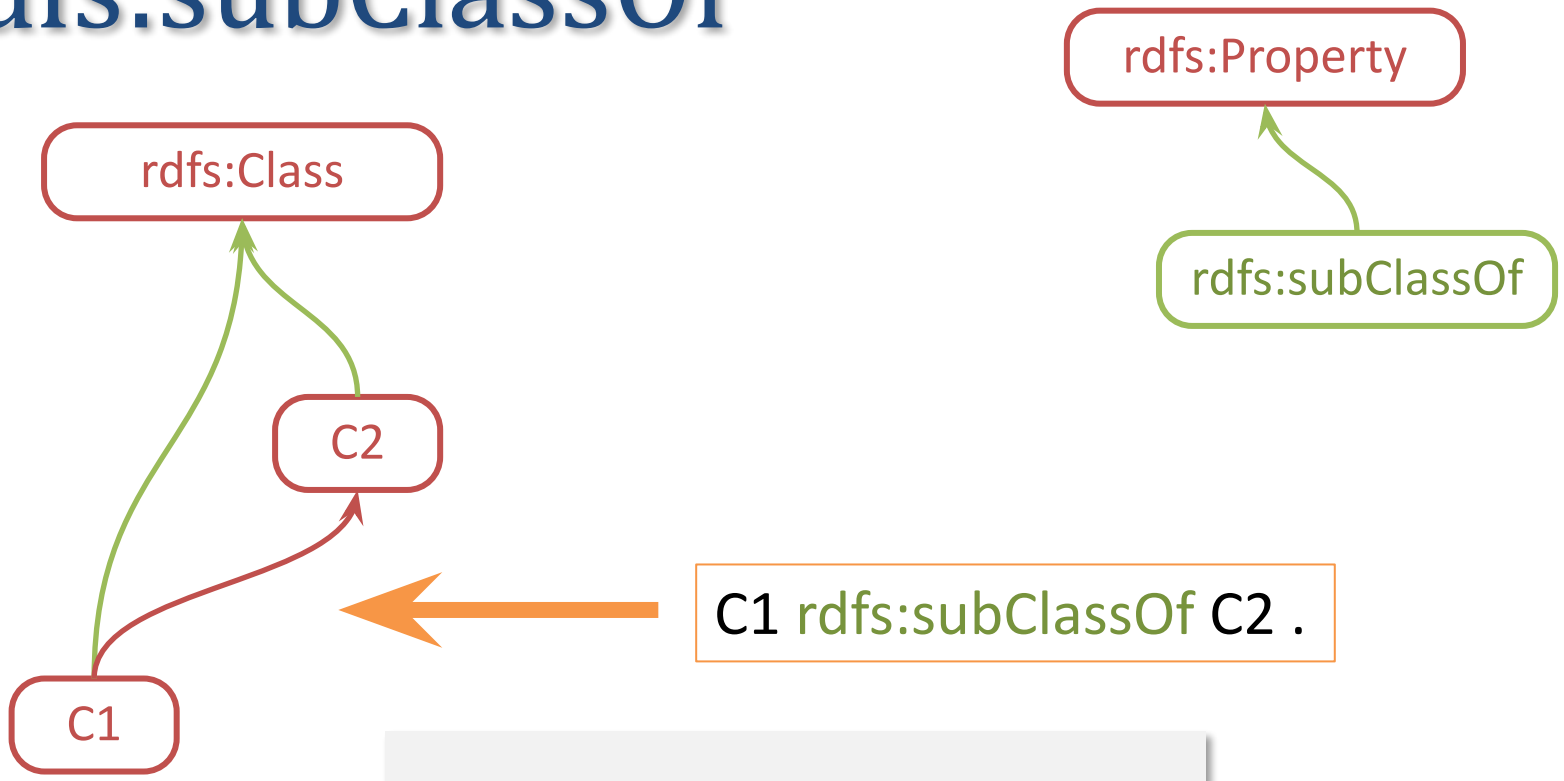
x P y .  x rdf:type C1 . x rdf:type C2 .

rdfs:subClassOf



Transitive

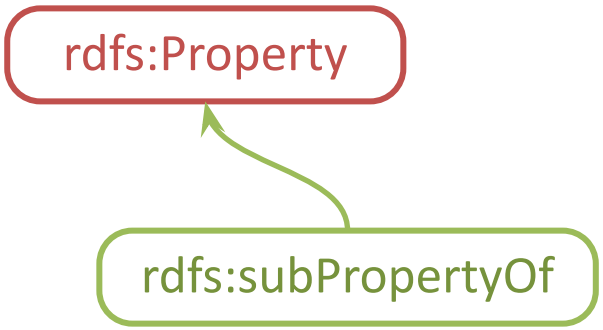
rdfs:subClassOf

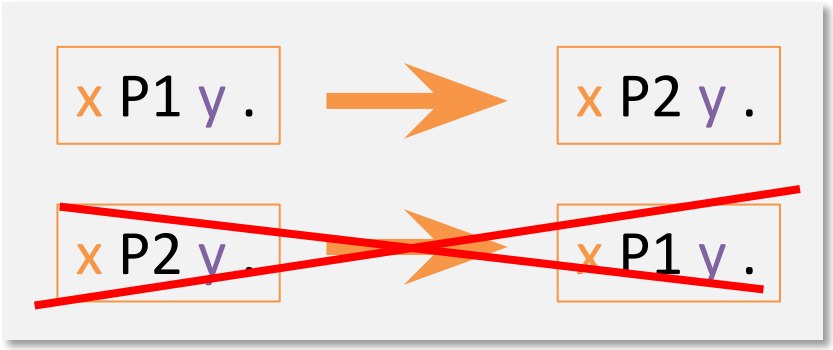


Transitive

rdfs:subPropertyOf

P1 rdfs:subPropertyOf P2 .





Transitive

Multiple Assertions

P1 rdfs:subPropertyOf P2 .

x P1 y .

x P2 y .

P2 rdfs:domain C .

x a C .

rdfs:label, rdfs:comment

rdfs:label is an instance of rdf:Property

used to provide a human-readable version of a resource's name

every resource should have a label in multiple languages

because browsers show the label when they show a resource

rdfs:label, rdfs:comment

rdfs:label is an instance of rdf:Property

used to provide a human-readable version of a resource's name

every resource should have a label in multiple languages

because browsers show the label when they show a resource

rdfs:comment is an instance of rdf:Property

used to provide a human-readable description of a resource

rdfs:seeAlso

rdfs:seeAlso is an instance of rdf:Property

used to indicate a resource that might provide additional information about the subject resource

rdfs:seeAlso

```
<http://schema.rdfs.org/all> a owl:Ontology;
   dct:title       "The schema.org terms in RDFS+OWL"@en;
   dct:description    "This is a conversion of the terms defined at schema.org to RDFS and
OWL."@en;
   foaf:page     <http://schema.rdfs.org/>;
   rdfs:seeAlso        <http://schema.org/>;
   rdfs:seeAlso        <http://github.com/mhausenblas/schema-org-rdf>;
   dct:hasFormat     <http://schema.rdfs.org/all.ttl>;
   dct:hasFormat     <http://schema.rdfs.org/all.rdf>;
   dct:hasFormat     <http://schema.rdfs.org/all.nt>;
   dct:hasFormat      <http://schema.rdfs.org/all.json>;
   dct:hasFormat     [
        dct:hasPart <http://schema.rdfs.org/all-classes.csv>;
        dct:hasPart <http://schema.rdfs.org/all-properties.csv>;
       ];
   dct:source <http://schema.org/>;
   dct:license <http://schema.org/docs/terms.html>;
   dct:valid "2012-06-16"^^xsd:date;
   .
```


slide by Pedro Szekely

rdfs:isDefinedBy

rdfs:isDefinedBy is an instance of rdf:Property
a subproperty of rdf:seeAlso

used to indicate a resource defining the subject resource

this property may be used to indicate an RDF vocabulary in which a resource is described

slide by Pedro Szekely

Example

… from the schema.org vocabulary definition

```
schema:ContactPage  a rdfs:Class;
   rdfs:label   "Contact Page"@en;
   rdfs:comment   "Web page type: Contact page."@en;
   rdfs:subClassOf       schema:WebPage;
   rdfs:isDefinedBy      <http://schema.org/ContactPage>;
   .
```

```
schema:cookTime       a rdf:Property;
   rdfs:label   "Cook Time"@en;
   rdfs:comment   """The time it takes to actually cook the dish, in ISO 8601
duration format. """@en;
   rdfs:domain      schema:Recipe;
   rdfs:range  schema:Duration;
   rdfs:isDefinedBy      <http://schema.org/Recipe>;
   .
```


slide by Pedro Szekely

how to represent KGs?

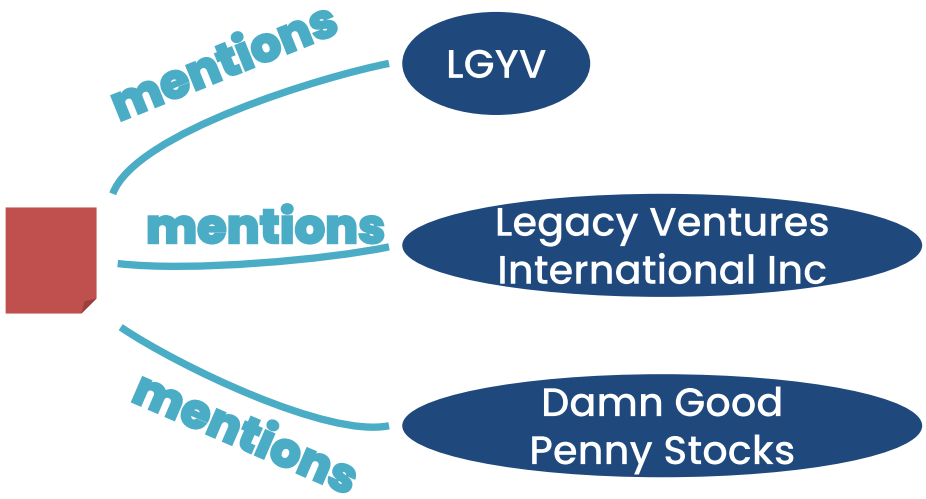
KG Definition

a directed, labeled multi-relational graph representing facts/assertions as triples

(h, r, t) **head entity, relation, tail entity**
(s, p, o) **subject, predicate, object**

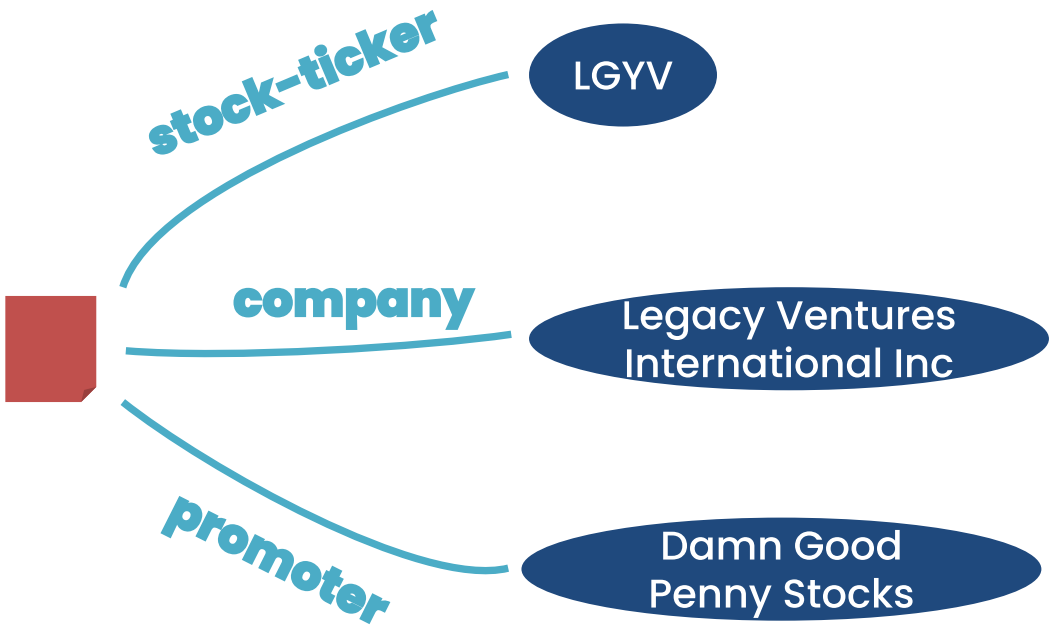
Simplest Knowledge Graph

Entities



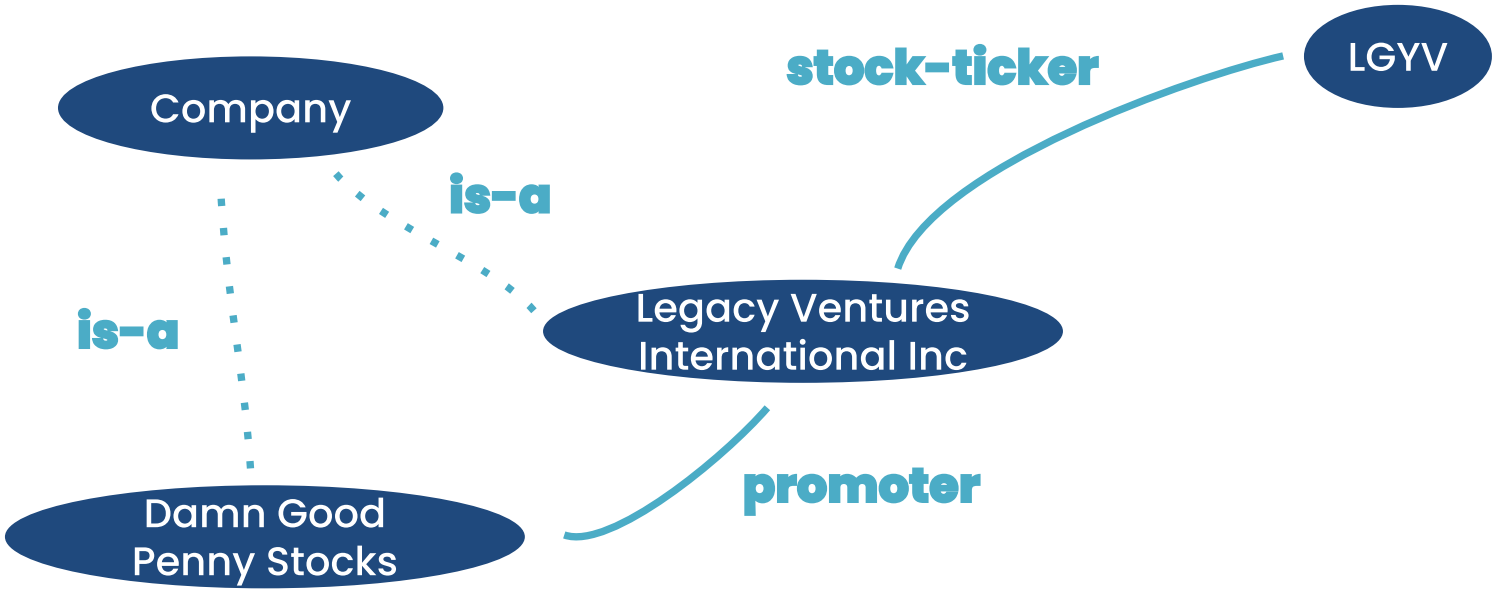
Simple, But Useful KG

Entities + properties (relation extraction)



Semantic Web KG (RDF/OWL)

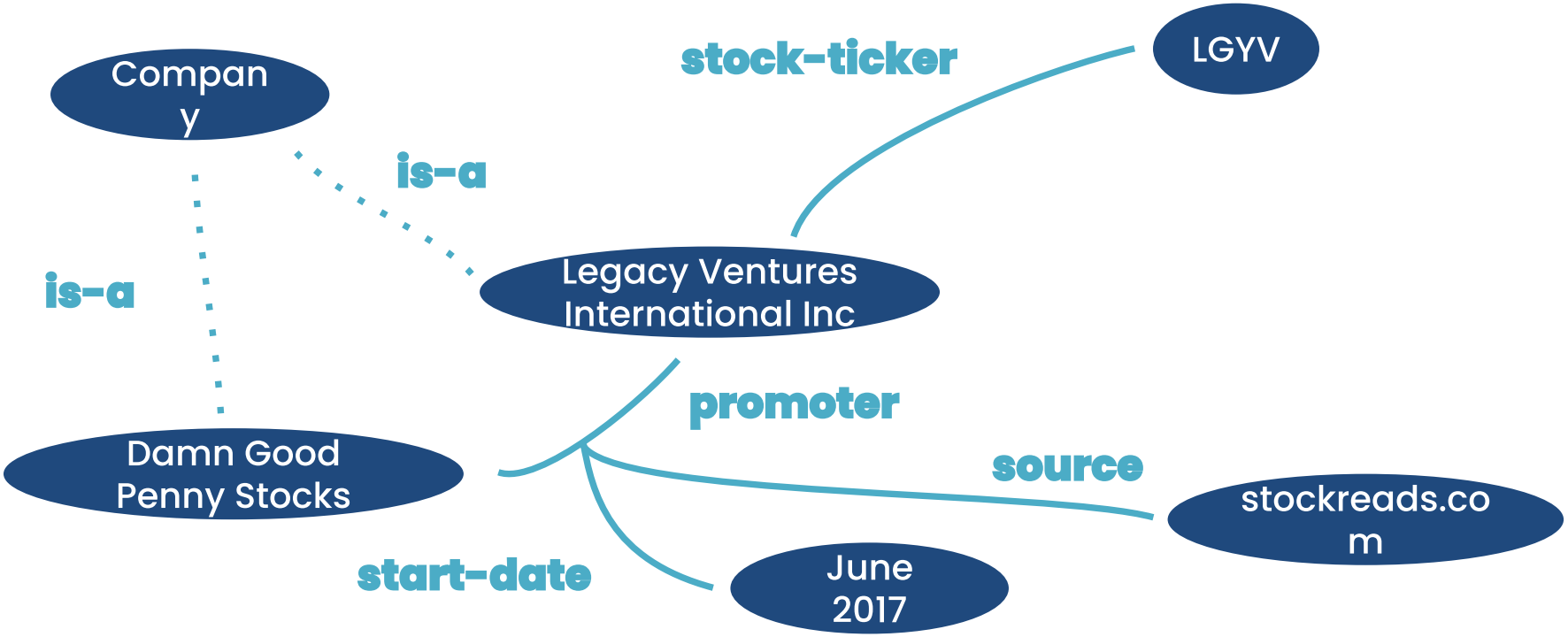
Entities + properties + classes



Kejriwal, Szekely

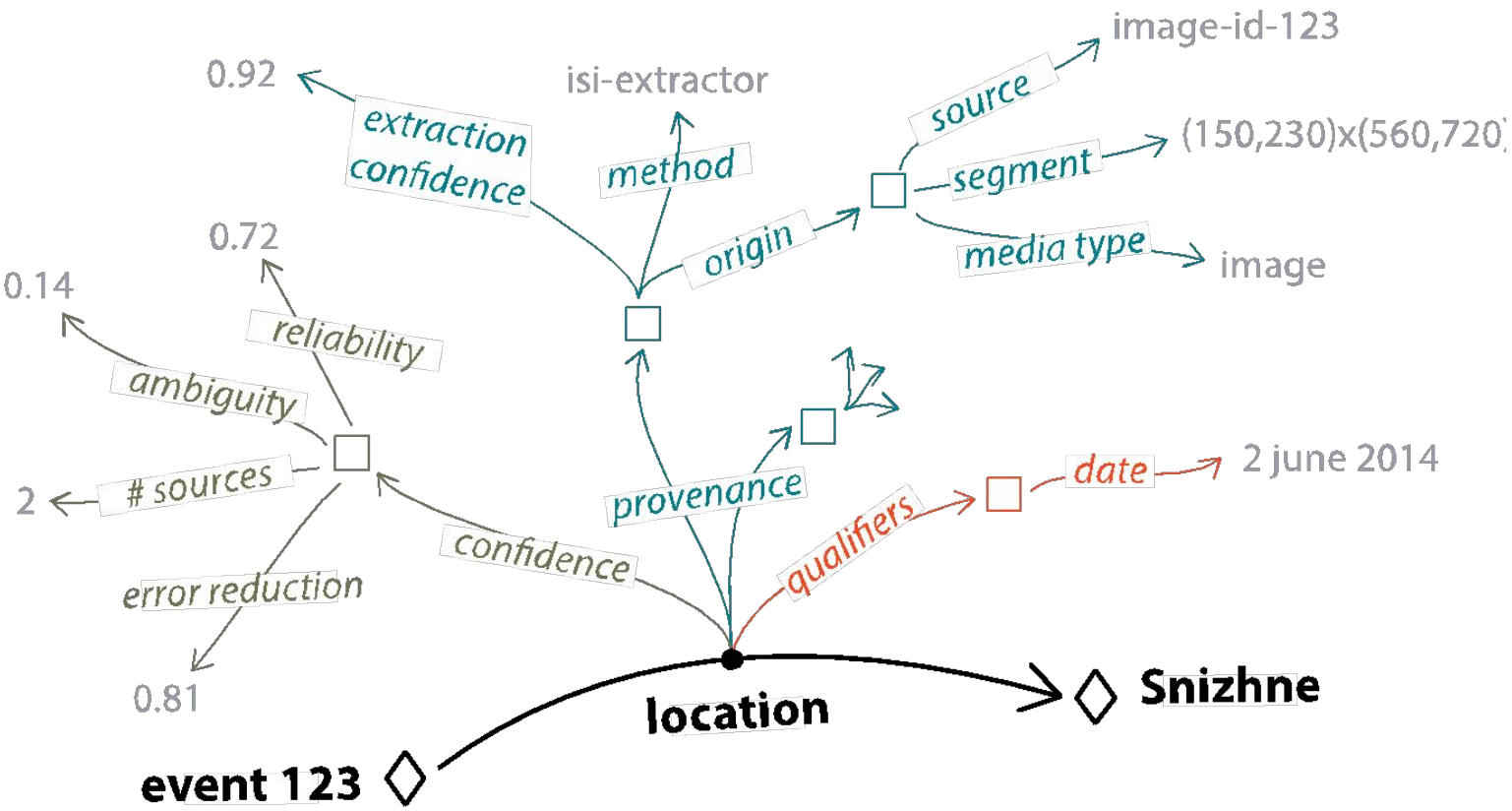
“Ideal” KG

Entities + properties + classes + qualifiers



Semi-Structured KG

Entities + properties + text + provenance + confidence

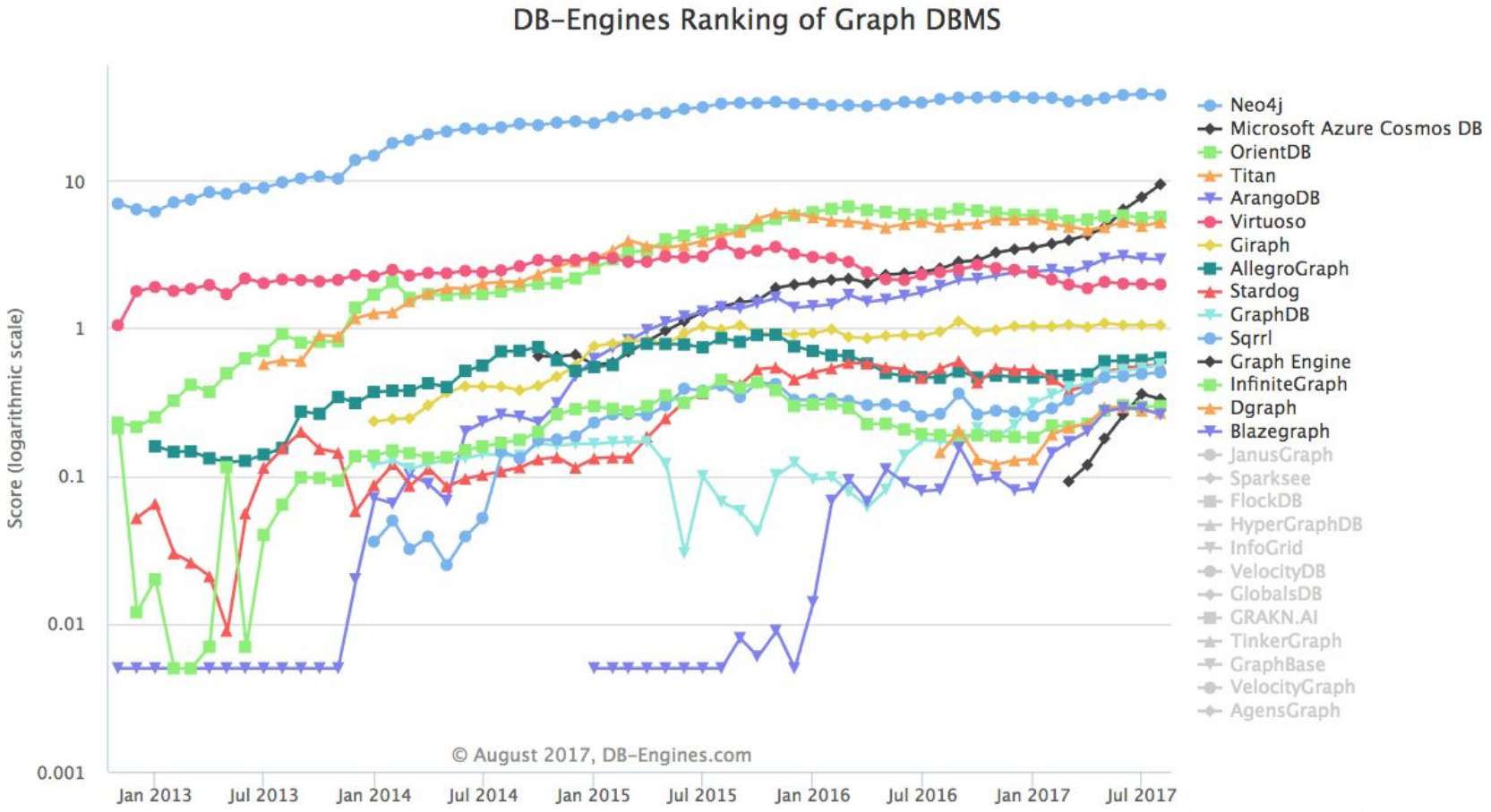


Where to Store KGs?

Serializing Knowledge Graphs

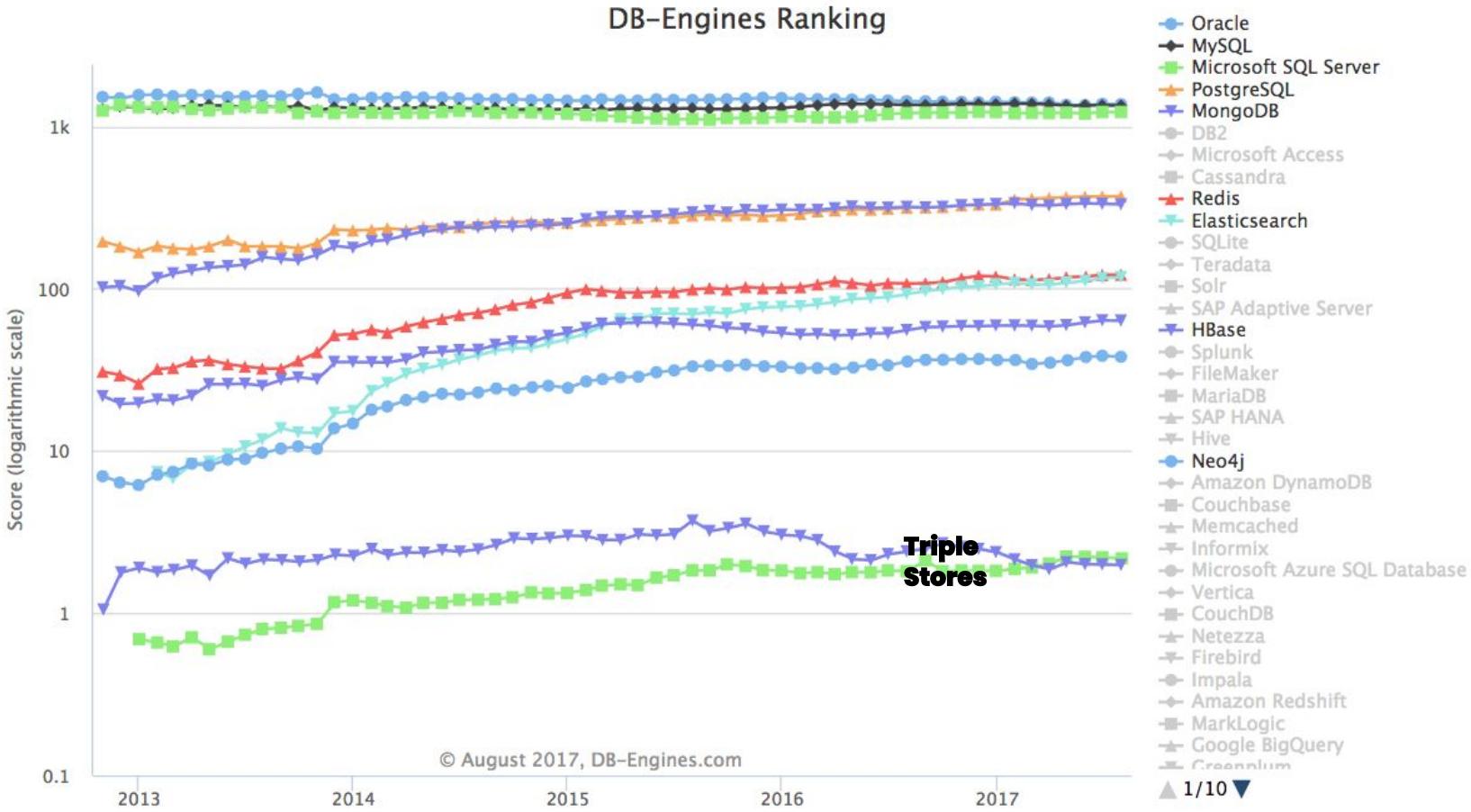
- Resource Description Framework (RDF) Stores
  - Database (triple store): AllegroGraph, Virtuoso,
  - Query: SPARQL (SQL-like)
- Key-Value, Document Stores
  - Data model: Node-centric
  - Databases: Hbase, MongoDB, Elastic Search, …
  - Query: filters, keywords, aggregation (no joins)
- Graph Databases
  - Data model: graph
  - Databases: Neo4J, Cayley, MarkLogic, GraphDB, Titan, OrientDB, Oracle, …
  - Query: GraphQL, Gremlin, Cypher

Popularity Ranking Of Graph Databases



https://db-engines.com/en/ranking_trend/graph+dbms

ElasticSearch, MongoDB & Neo4J Have Wide Adoption



https://db-engines.com/en/ranking_trend/graph+dbms