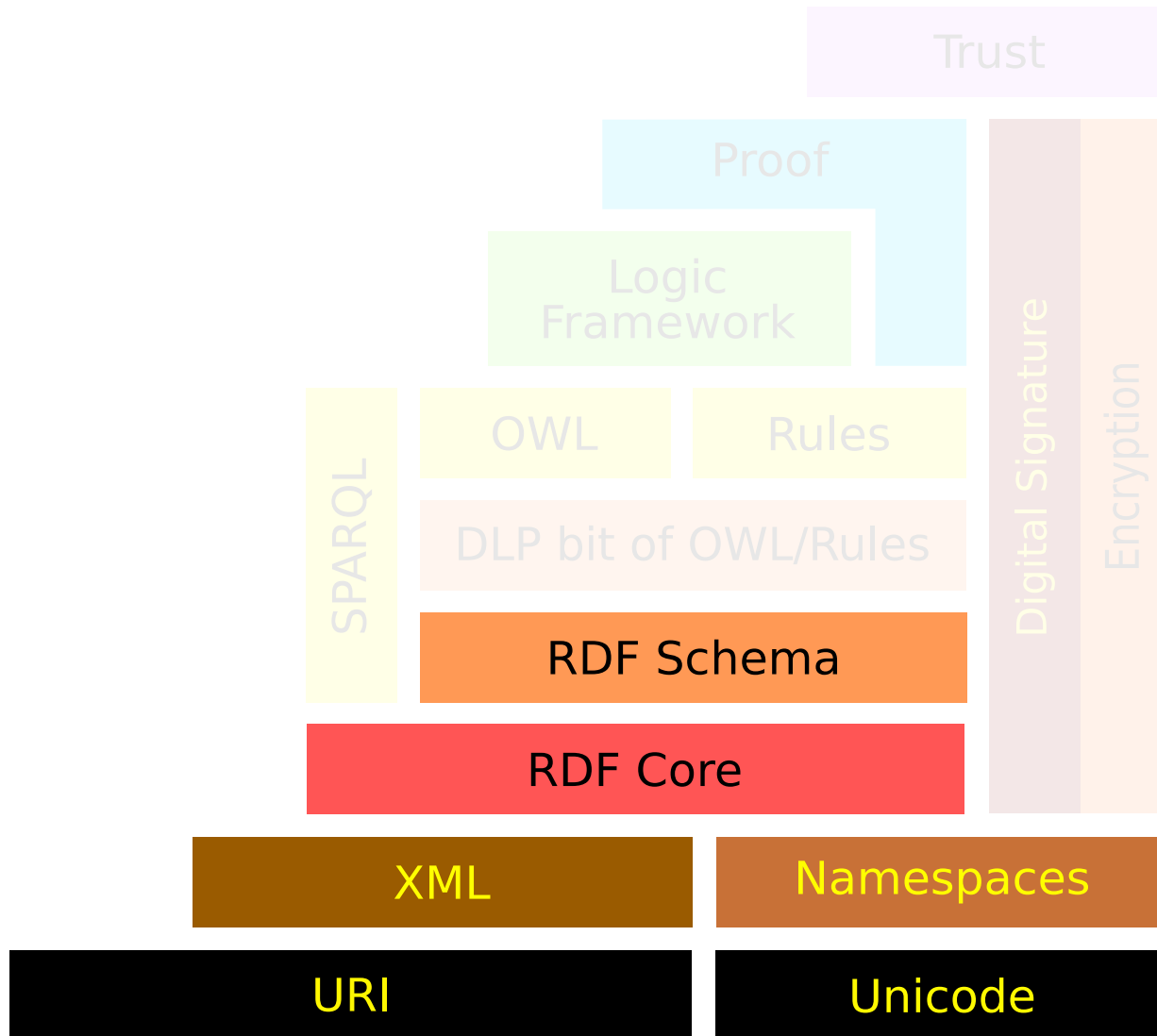


RDF

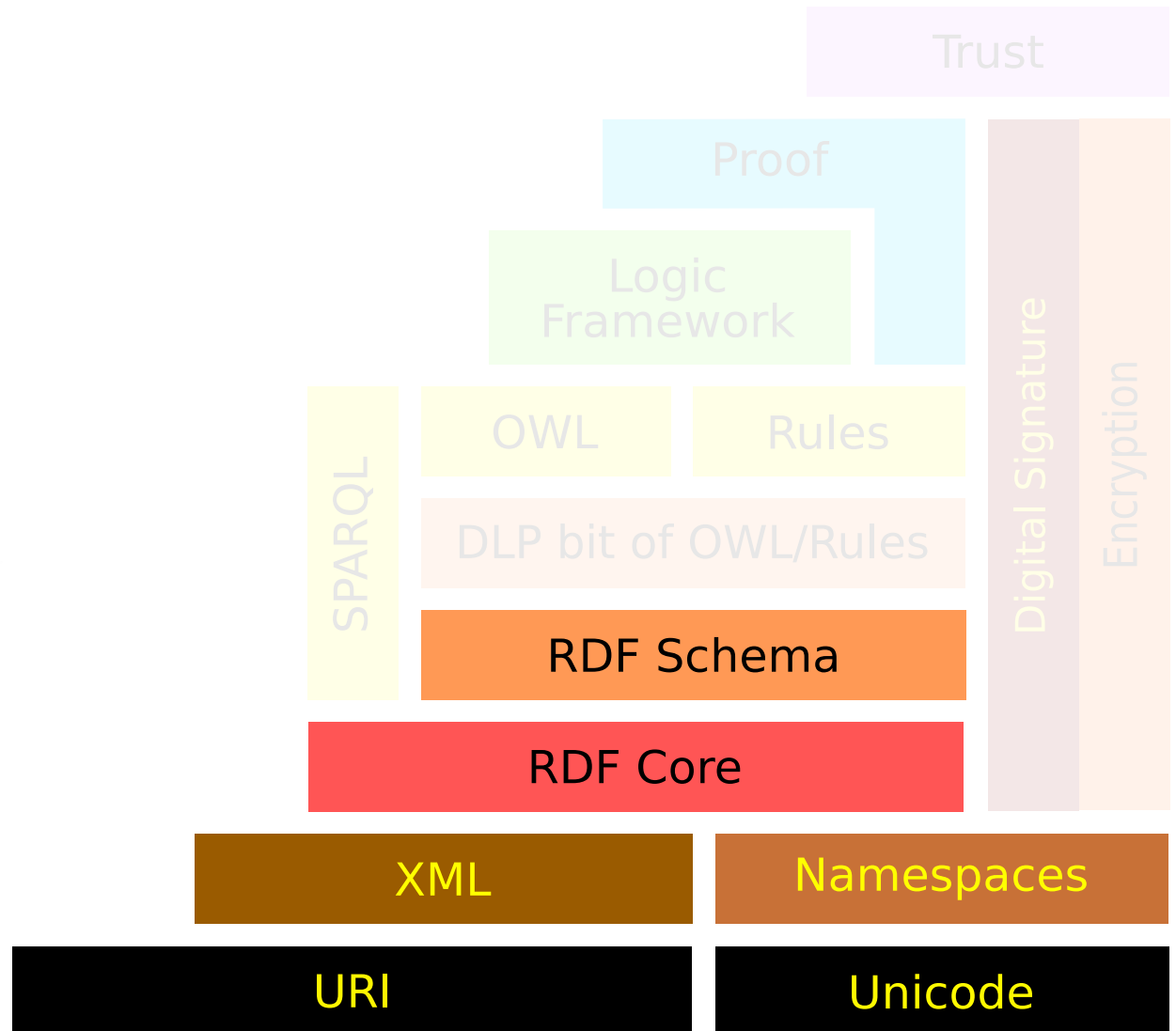
Pedro Szekely

University of Southern California/ISI

Semantic Web Layer Cake



Unicode



Why Unicode?

<http://site.com/Македонски.html>

http://site.com/Μία_Σελίδα

<http://www.中国政府.政务.cn>

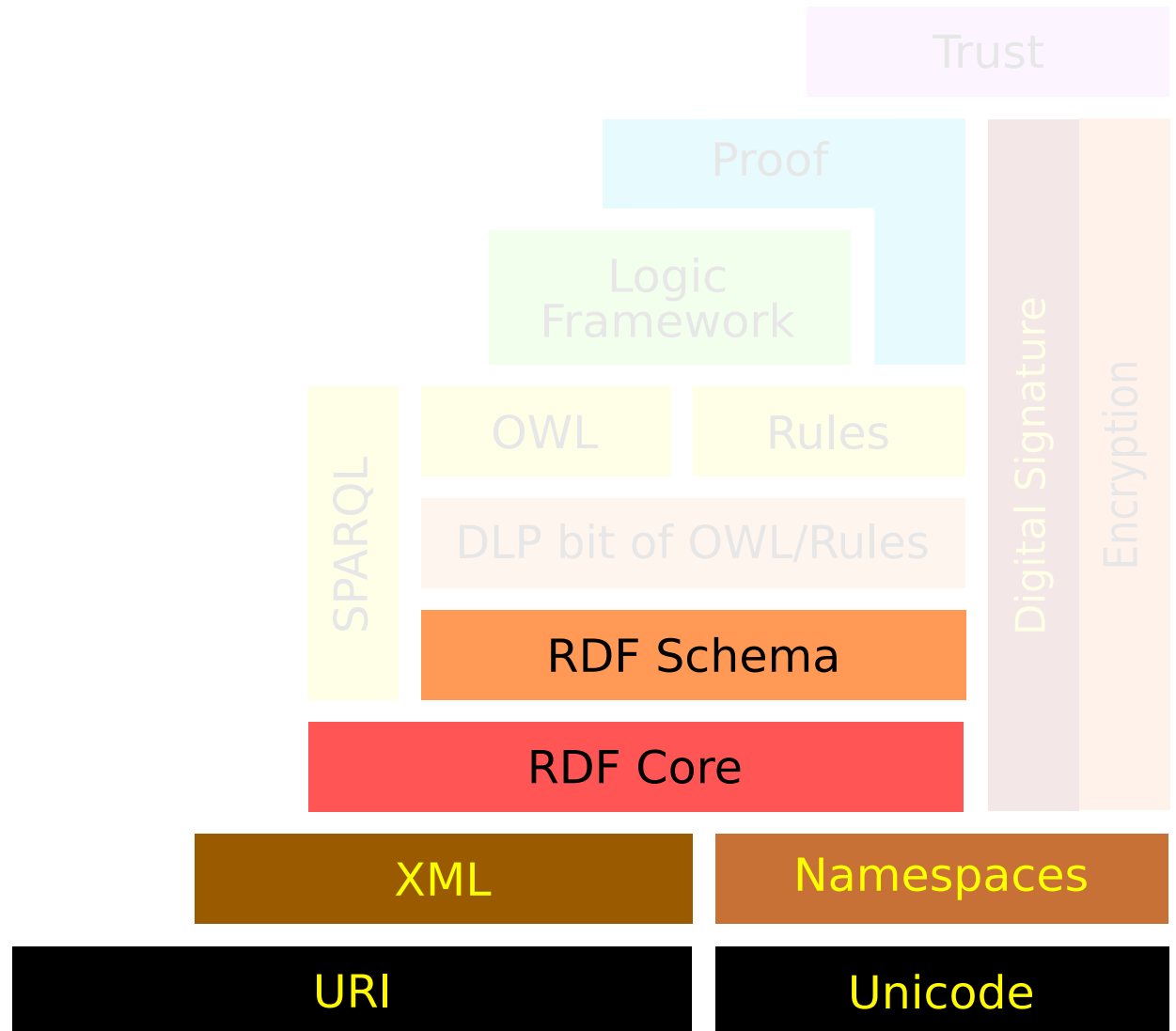
Unicode

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

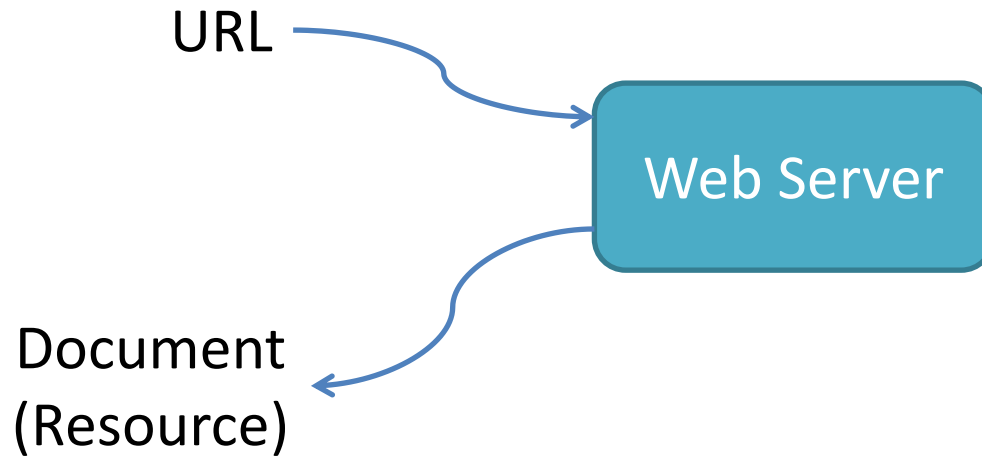
[Wikipedia](#)

URI

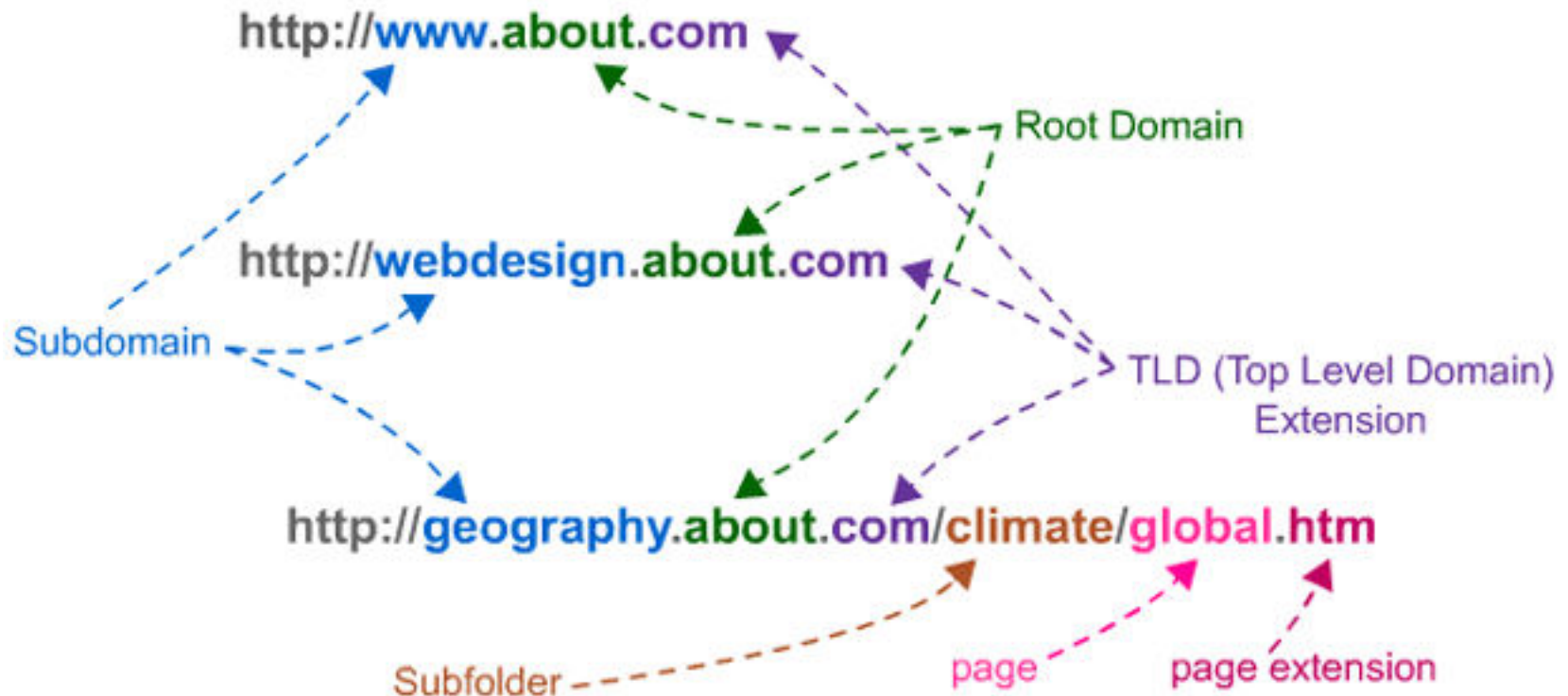


URL: Uniform Resource Locator

A reference to an Internet resource



URL: Uniform Resource Locator



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

URL vs URI

URI

URL

URN

locators

names

like person's street address
method for finding it

like a person's name
item's identity

Can USC Have a URI?



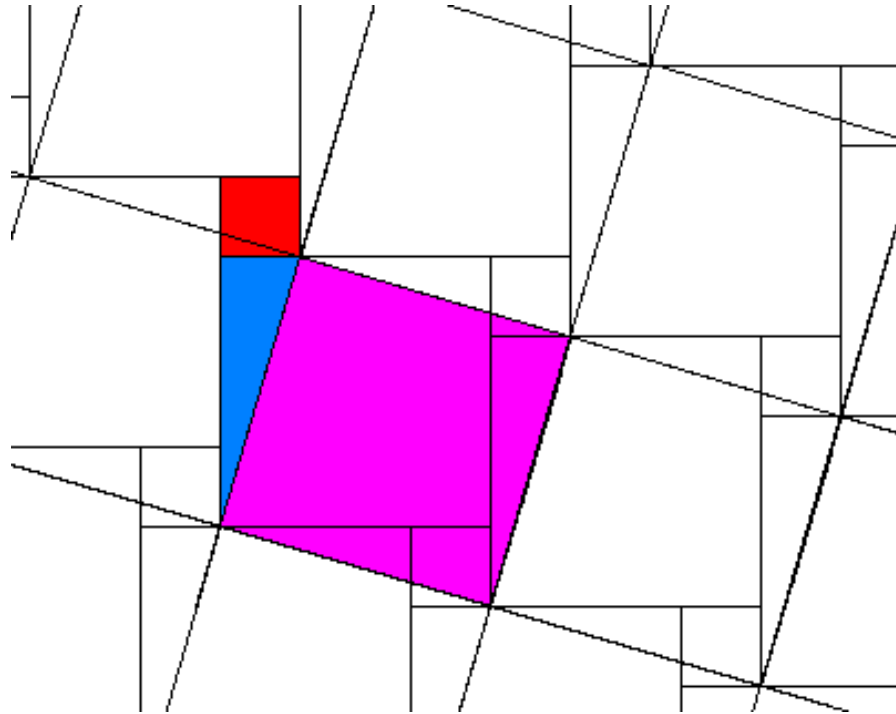
Can USC Have a URI?



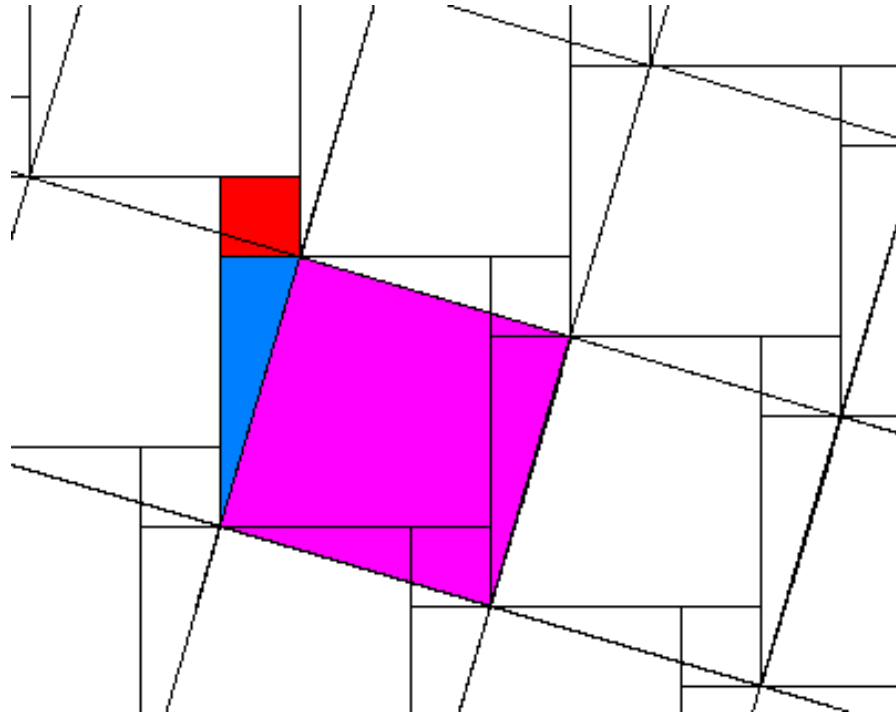
http://dbpedia.org/page/University_of_Southern_California

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?



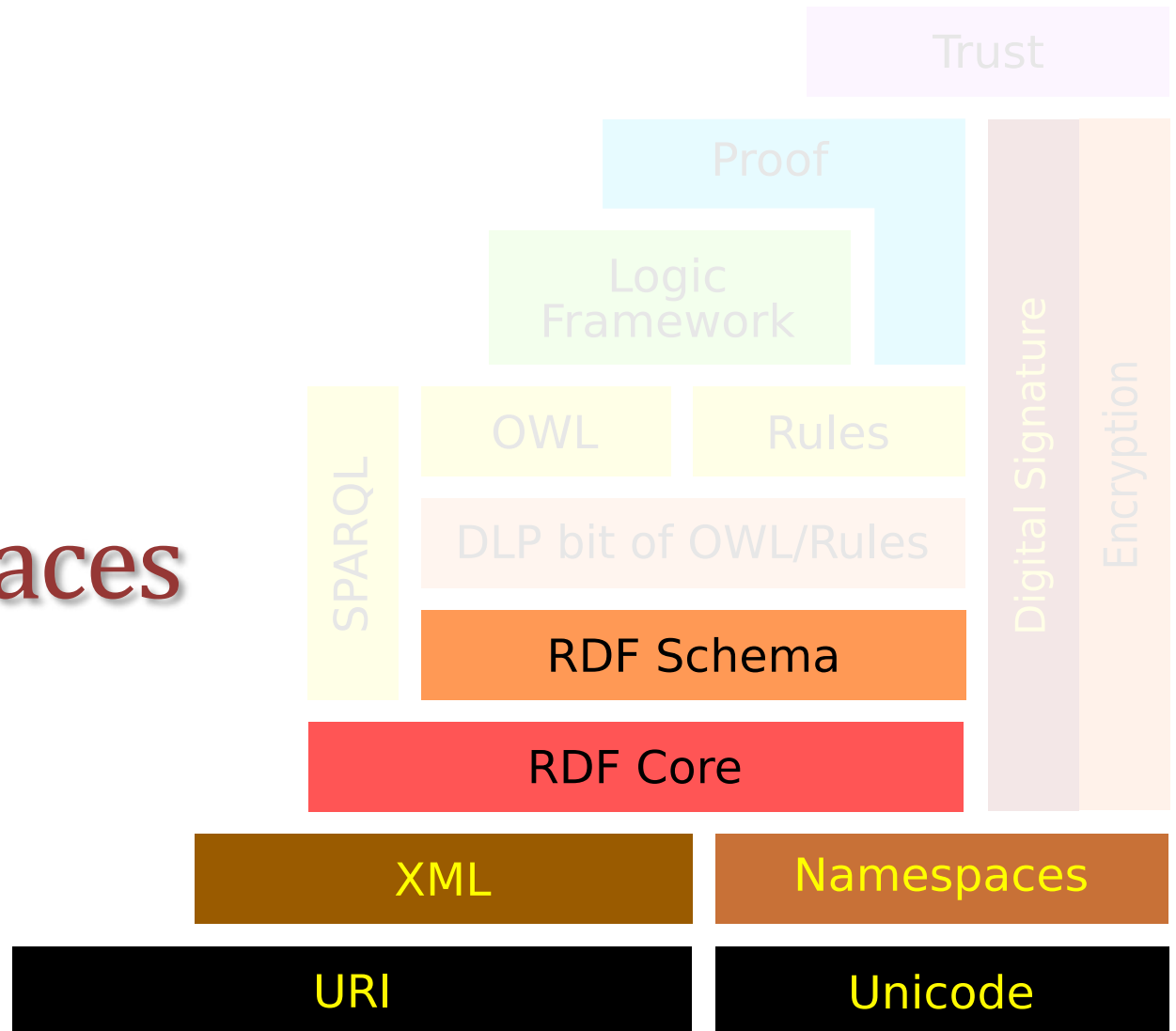
My Dog: Can He Have a URI?



<http://szekelys.com/diego>

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

Namespaces



Are These the Same?

```
<Bookstore>
<Book>
  <Author>John Doe</Author>
  <Title>Introduction to XML</Title>
  <Publisher>XYZ</Publisher>
</Book>
</Bookstore>
```

```
<http://amazon.com/store/Bookstore>
<http://amazon.com/store/Book>
  <http://amazon.com/store/Author>John Doe</http://amazon.com/store/Author>
  <http://amazon.com/store/Title>Introduction to XML</http://amazon.com/store/Title>
  <http://amazon.com/store/Publisher>XYZ</http://amazon.com/store/Publisher>
</http://amazon.com/store/Book>
</http://amazon.com/store/Bookstore>
```

```
<http://barnesandnoble.com/store/Bookstore>
<http://barnesandnoble.com/store/Book>
  <http://barnesandnoble.com/store/Author>John Doe</http://barnesandnoble.com/store/Author>
  <http://barnesandnoble.com/store/Title>Introduction to XML</http://barnesandnoble.com/store/Title>
  <http://barnesandnoble.com/store/Publisher>XYZ</http://barnesandnoble.com/store/Publisher>
</http://barnesandnoble.com/store/Book>
</http://barnesandnoble.com/store/Bookstore>
```

Namespaces

XML namespaces are used for providing uniquely named elements and attributes in an XML document

[Wikipedia](#)

```
xmlns="http://amazon.com/store"
```

Using a Namespace Declaration

```
<http://amazon.com/store/Bookstore>  
<http://amazon.com/store/Book>  
  <http://amazon.com/store/Author>John Doe</http://amazon.com/store/Author>  
  <http://amazon.com/store/Title>Introduction to XML</http://amazon.com/store/Title>  
  <http://amazon.com/store/Publisher>XYZ</http://amazon.com/store/Publisher>  
</http://amazon.com/store/Book>  
</http://amazon.com/store/Bookstore>
```

=

```
<Bookstore xmlns="http://amazon.com/store">  
  <Book>  
    <Author>John Doe</Author>  
    <http://amazon.com/store/Title>Introduction to XML</Title>  
    <http://amazon.com/store/Publisher>XYZ</Publisher>  
  </Book>  
</Bookstore>
```

Default and Prefix Namespaces

```
<http://amazon.com/store/Bookstore>
<http://amazon.com/store/Book>
  <http://amazon.com/store/Author>John Doe</http://amazon.com/store/Author>
  <http://amazon.com/store/Title>Introduction to XML</http://amazon.com/store/Title>
  <http://amazon.com/store/Publisher>XYZ</http://amazon.com/store/Publisher>
</http://amazon.com/store/Book>
</http://amazon.com/store/Bookstore>
```

```
<Bookstore xmlns="http://amazon.com/store">
<Book>
  <Author>John Doe</Author>
  <Title>Introduction to XML</Title>
  <Publisher>XYZ</Publisher>
</Book>
</Bookstore>
```

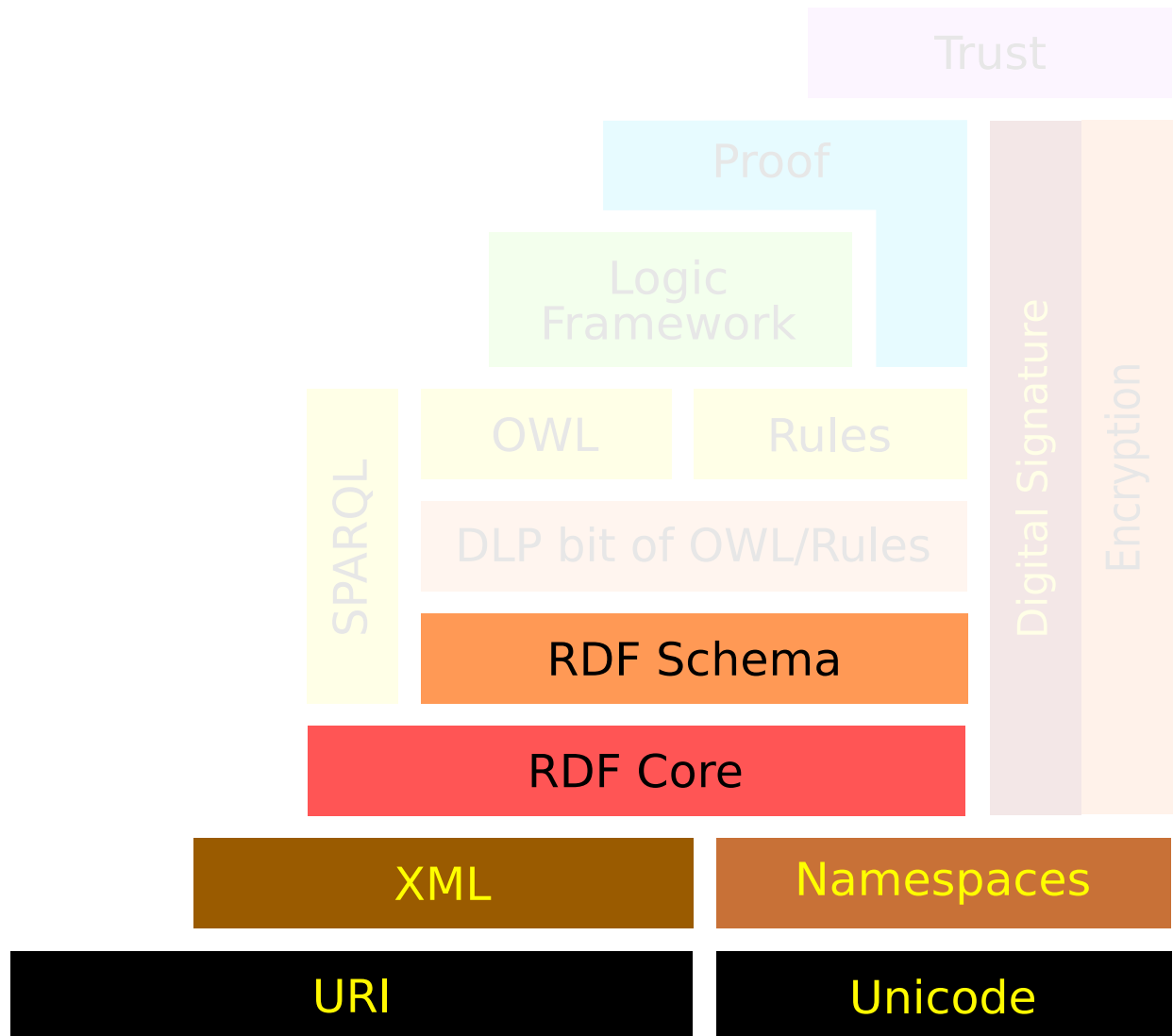
```
<am:Bookstore xmlns:am="http://amazon.com/store">
<am:Book>
  <am:Author>John Doe</am:Author>
  <am:Title>Introduction to XML</am:Title>
  <am:Publisher>XYZ</am:Publisher>
</am:Book>
</am:Bookstore>
```

Default and Prefix Namespaces

```
<am:Bookstore
  xmlns:am="http://amazon.com/store"
  xmlns:bn=http://barnesandnoble.com/store>
<am:Book>
  <am:Author>John Doe</am:Author>
  <bn:Author>Jane Doe</bn:Author>
  <am:Title>Introduction to XML</am:Title>
  <am:Publisher>XYZ</am:Publisher>
</am:Book>
</am:Bookstore>
```

If elements were defined within a global scope,
it would be a problem to
combine elements from multiple documents

XML



eXtensible Markup Language

HTML specifies how to display data

fixed
set of tags

```
<h2>Nonmonotonic Reasoning</h2>  
<i>by <b>V. Marek</b> and <b>M. Truszczyński</b></i><br>  
Springer 1993<br>  
ISBN 0387976892
```

XML specifies data

extensible
set of tags

```
<book>  
  <title>Nonmonotonic Reasoning</title>  
  <author>V. Marek</author>  
  <author>M. Truszczyński</author>  
  <publisher>Springer</publisher>  
  <year>1993</year>  
  <ISBN>0387976892</ISBN>  
</book>
```

Design of XML

- Tags can be used to indicate the meaning of data
- No fixed set of markup tags: new tags can be defined
- Underlying data model is a tree structure
 - Actually XML can represent graphs through IDs and IDREFs, but it's a bit cumbersome
- XML provides a common exchange format
- W3C Recommendation:
<http://www.w3.org/TR/REC-xml/>

Merging Problem in XML

Document 1

```
<Bookstore xmlns="http://amazon.com">
  <Book id="2">
    <Publisher>Springer</Publisher>
  </Book>
  <Book id="1">
    <Publisher>ACM</Publisher>
  </Book>
</Bookstore>
```

Document 2

```
<Bookstore xmlns="http://amazon.com">
  <Book id="1">
    <Author>John</Author>
    <Title>Introduction to XML</Title>
  </Book>
  <Book id="2">
    <Author>Susan</Author>
    <Title>Advanced</Title>
  </Book>
</Bookstore>
```

Merged Document

```
<Bookstore xmlns="http://amazon.com">
  <Book id="1">
    <Author>John</Author>
    <Title>Introduction to XML</Title>
    <Publisher>ACM</Publisher>
  </Book>
  <Book id="2">
    <Author>Susan</Author>
    <Title>Advanced</Title>
    <Publisher>Springer</Publisher>
  </Book>
</Bookstore>
```

... is difficult

Does XML Represent Meaning?

John is an instructor for CS101

```
<instructor name="John">  
  <teaches>CS 101</teaches>  
</instructor>
```

```
<course name="CS101">  
  <instructor> John </instructor>  
</course>
```

Opposite nesting, same information!

Does XML Represent Meaning?

John is an instructor for CS101

```
<instructor name="John">  
  <teaches>CS 101</teaches>  
</instructor>
```

```
<course name="CS101">  
  <instructor> John </instructor>  
</course>
```

hasInstructor inverseOf teaches
 $\forall C, I \text{ hasInstructor}(C, I) \leftrightarrow \text{teaches}(I, C)$

range(hasInstructor) = Person
 $\forall C, I \text{ hasInstructor}(C, I) \rightarrow \Box \text{ Person}(I)$

Meaning of Data in XML?

```
...  
<Book>  
  <Author>John</Author>  
  <Title>Introduction to XML</Title>  
  <Publisher>ACM</Publisher>  
  <Country>USA</Country>  
</Book>  
...
```

What is the meaning of **Country**?

- ... where the book is sold?
- ... where it is published?
- ... where the author lives?
- ... ???

XML Schema

The purpose of a schema is to
define a class of XML documents, and so the term
"instance document" is often used to
describe an XML document
that conforms to a particular schema

<http://www.w3.org/TR/xmlschema-0/>

a syntax checker

Example

there are simple and complex types
simple can contain only text and no
elements or attributes

Defining the USAddress Type

```
<xsd:complexType name="USAddress" >  
  <xsd:sequence>  
    <xsd:element name="name" type="xsd:string"/>  
    <xsd:element name="street" type="xsd:string"/>  
    <xsd:element name="city" type="xsd:string"/>  
    <xsd:element name="state" type="xsd:string"/>  
    <xsd:element name="zip" type="xsd:decimal"/>  
  </xsd:sequence>  
  <xsd:attribute name="country" type="xsd:NMTOKEN" fixed="US"/>  
</xsd:complexType>
```

... must have specific elements
... in a specific order
... filled with specific types of data

IMP

XML Schema by Example

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.books.org"
  xmlns=http://www.books.org>
  <xsd:element name="Bookstore">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Book" minOccurs="1" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Book">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Title" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="Author" minOccurs="1" maxOccurs="unbounded"/>
        <xsd:element ref="Date" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="ISBN" minOccurs="1" maxOccurs="1"/>
        <xsd:element ref="Publisher" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Title" type="xsd:string"/>
  <xsd:element name="Author" type="xsd:string"/>
  <xsd:element name="Date" type="xsd:date"/>
  <xsd:element name="ISBN" type="xsd:integer"/>
  <xsd:element name="Publisher" type="xsd:string"/>
</xsd:schema>
```

“Bookstore” is a complex Type

A sequence of 1 or more “Book” elements

When referring to another Element, use “ref”

Notice the use of more meaningful data types

XML Schema Primitive Types

string

boolean

decimal

float

double

duration

dateTime

time

date

gYearMonth

gYear

gMonthDay

gDay

gMonth

hexBinary

base64Binary

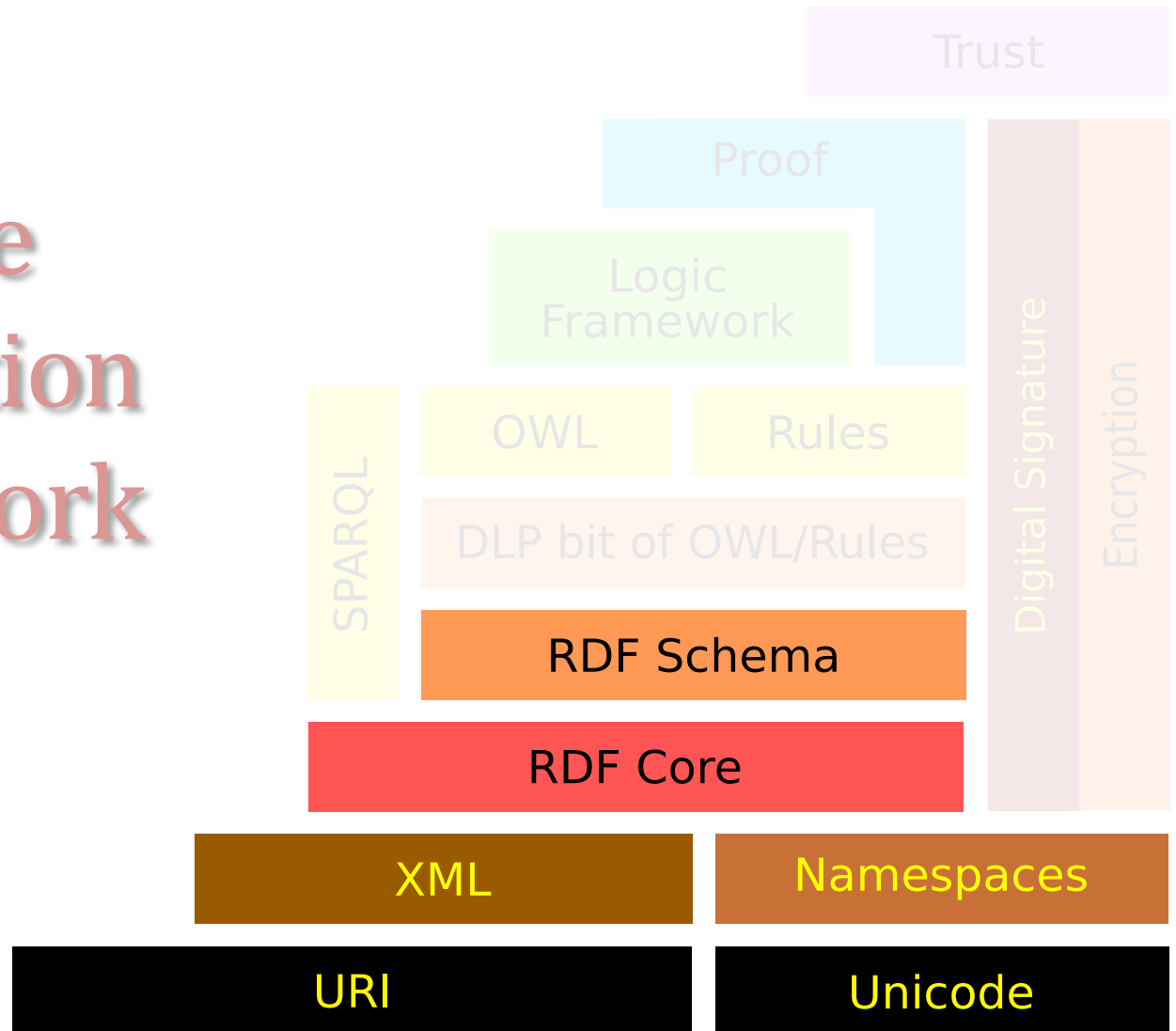
anyURI

Qname

NOTATION

useful in RDF

Resource Description Framework



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/>

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)

Represent Resources Using URIs



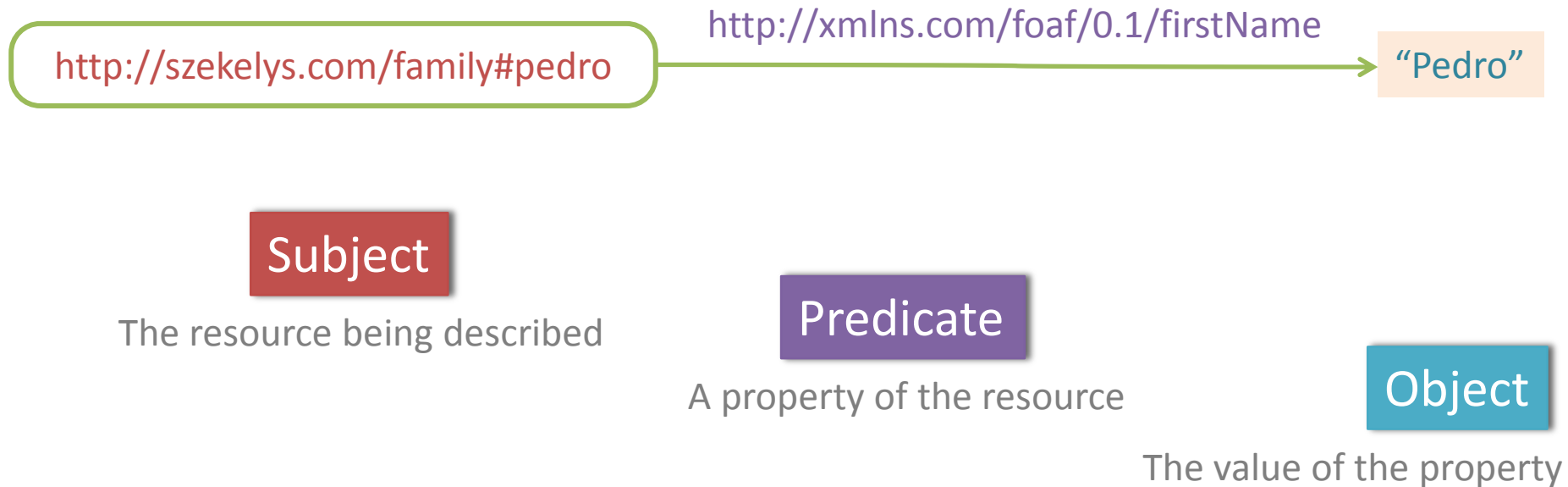
That **guy** has first name **“Pedro”**

<http://szekelys.com/family#pedro>

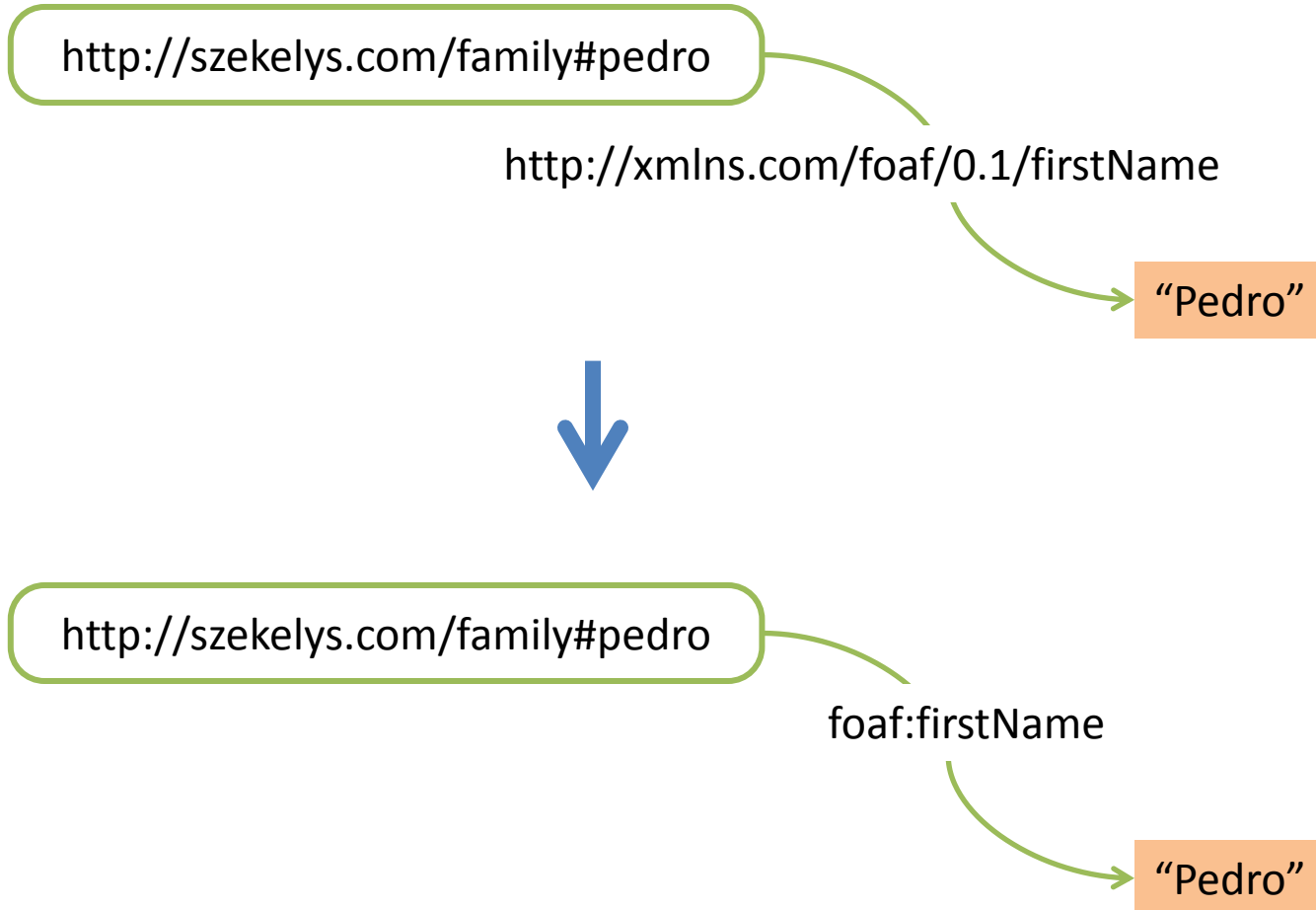
<http://xmlns.com/foaf/0.1/firstName>

“Pedro”

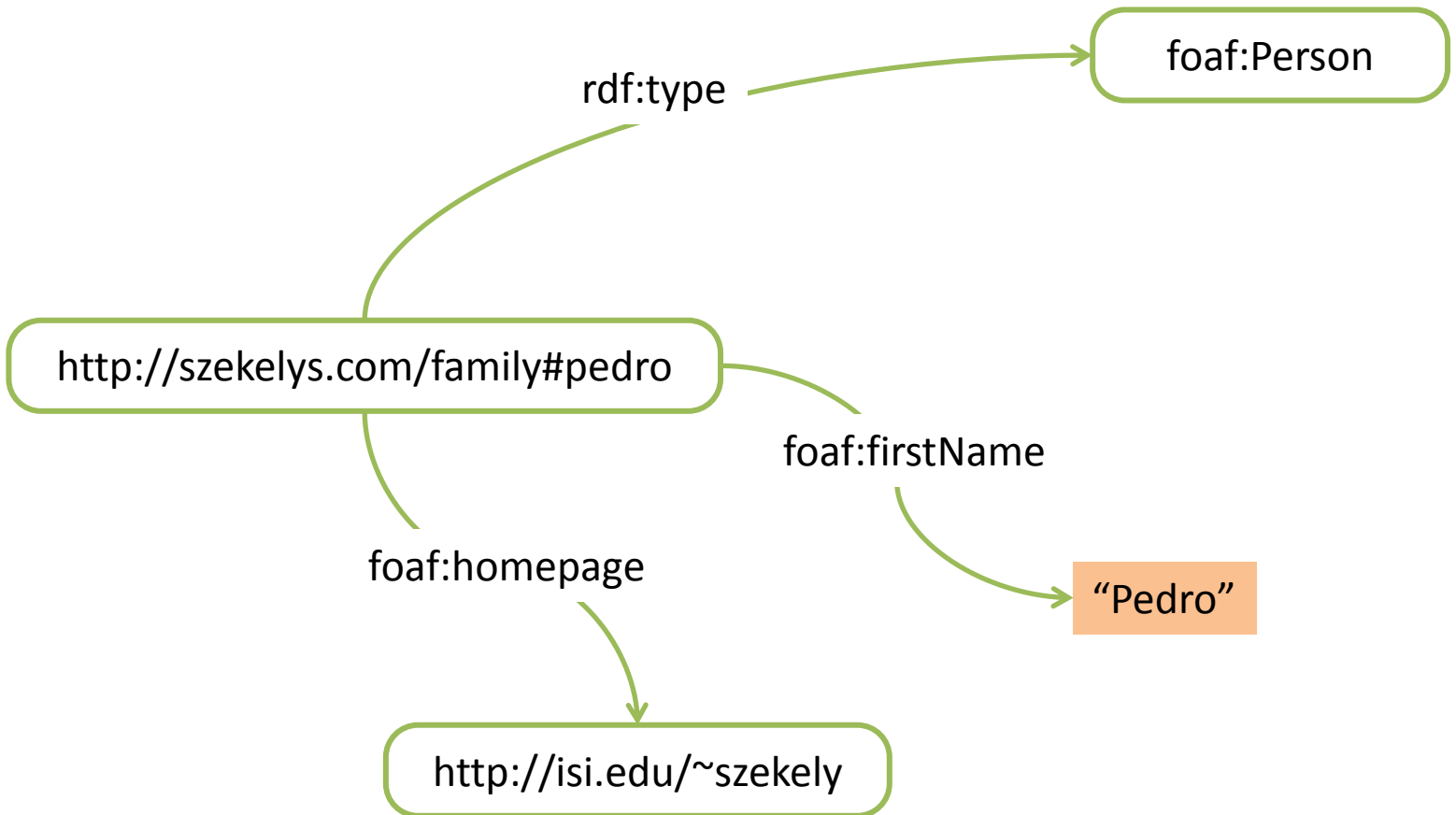
Represent Information as Triples



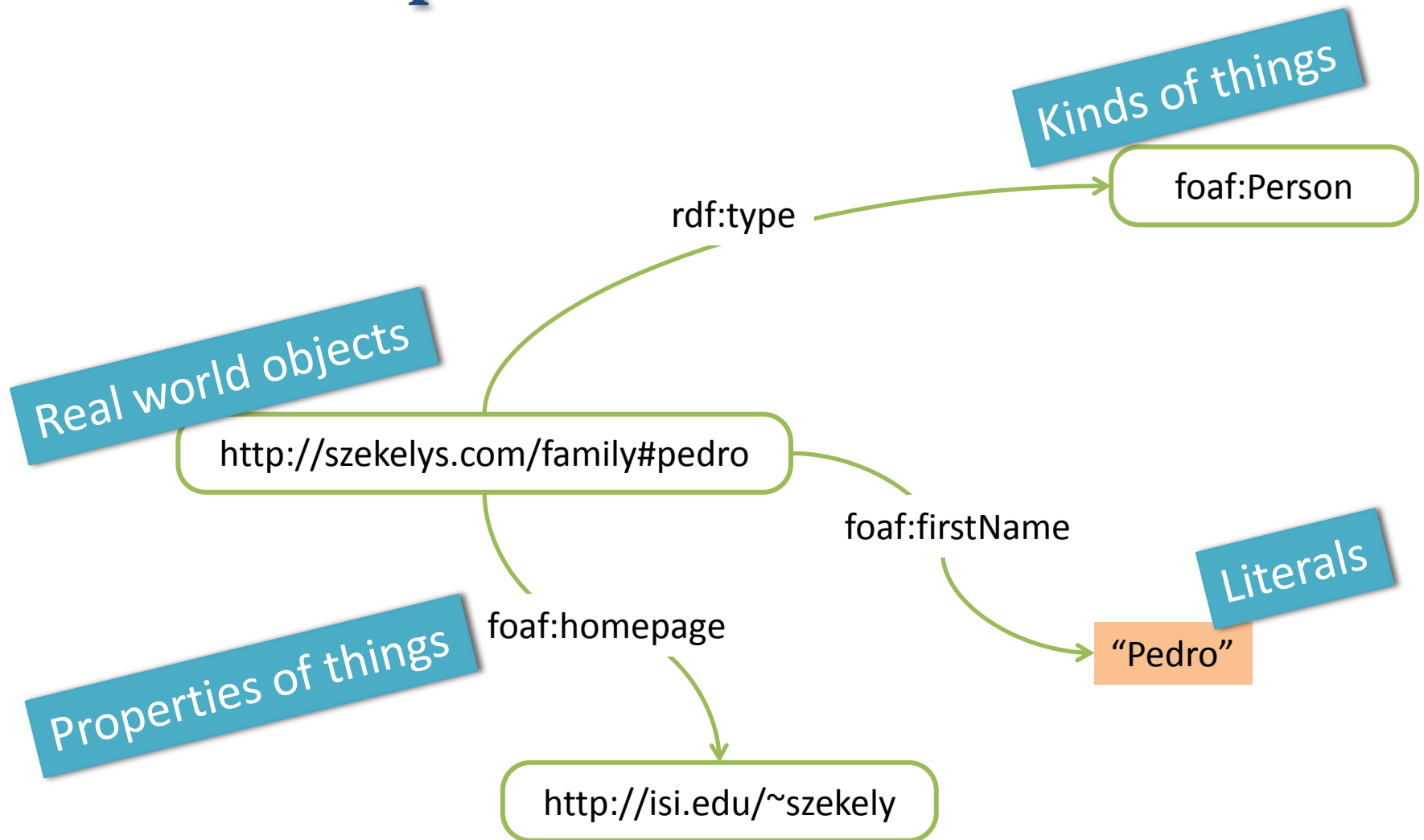
Use Namespaces



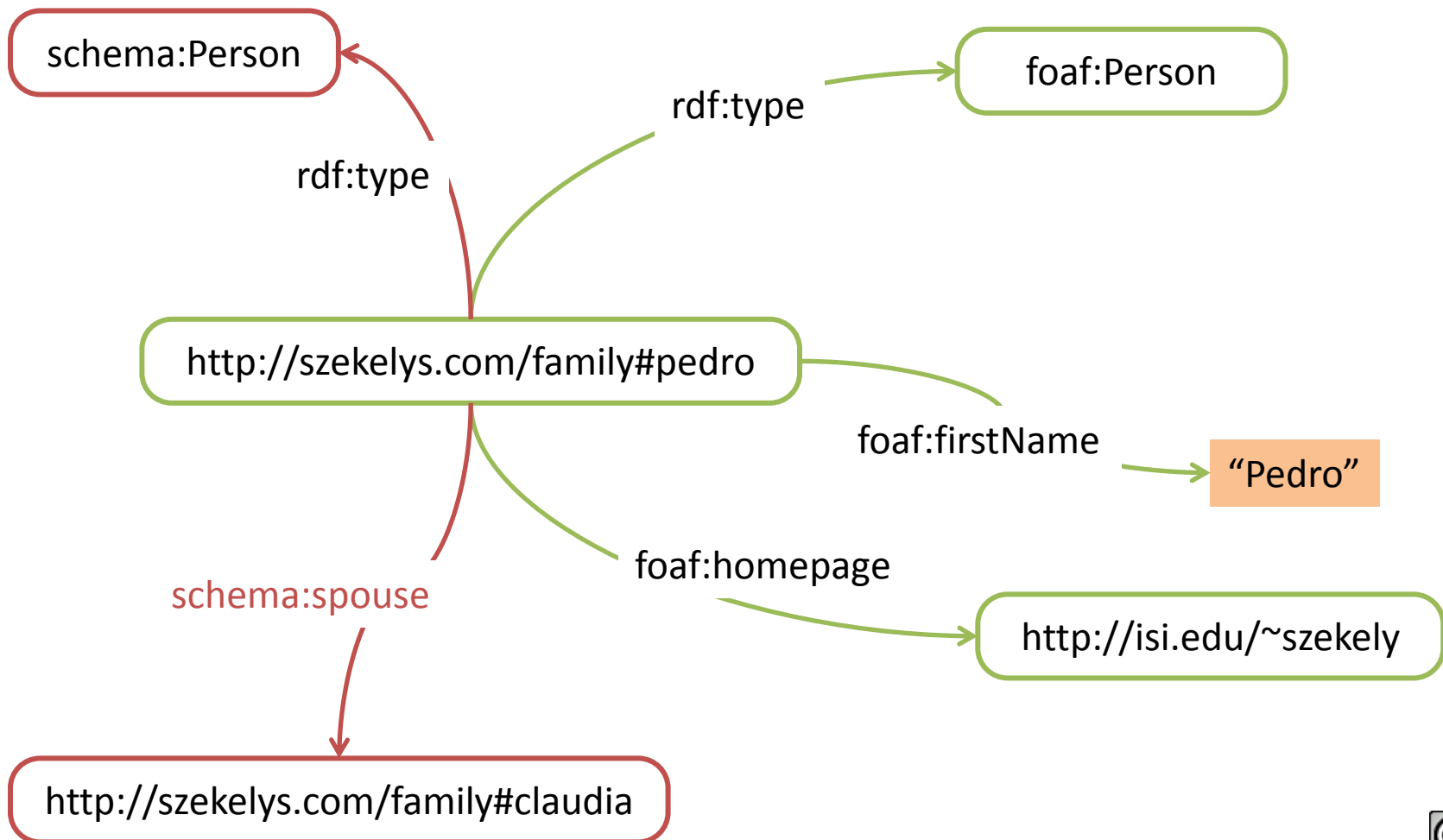
RDF Graphs



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

XML vs RDF

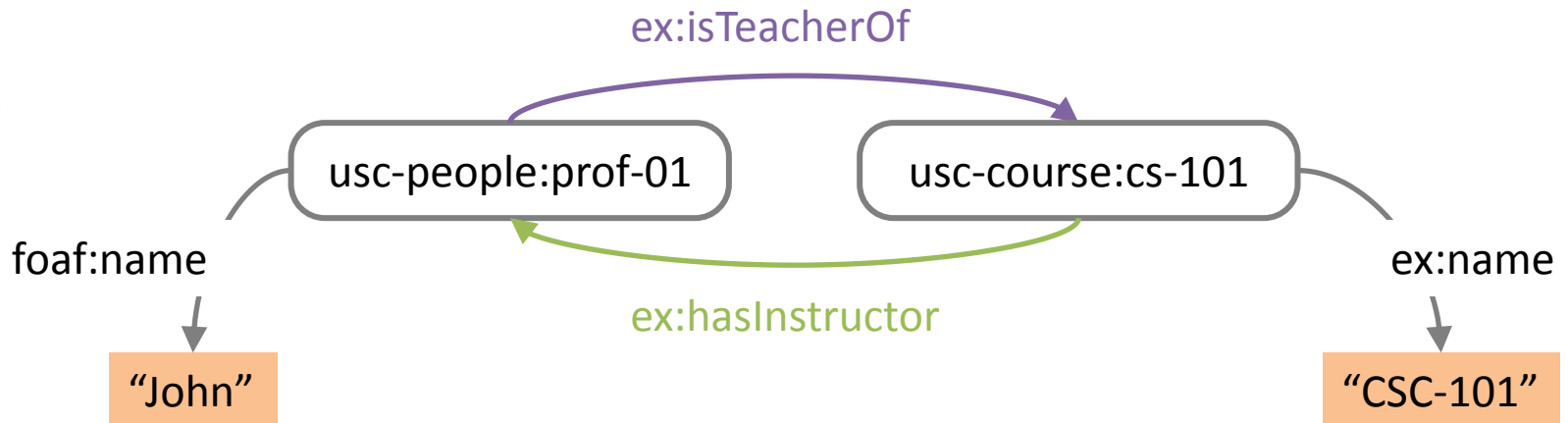
John is an instructor for CS101

XML

```
<instructor name="John">  
  <teaches>CS 101</teaches>  
</instructor>
```

```
<course name="CS101">  
  <instructor> John </instructor>  
</course>
```

RDF



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

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/">

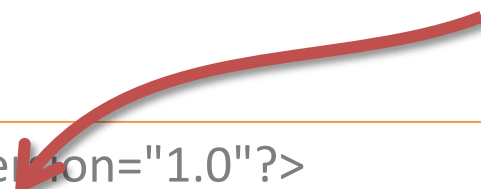
  <rdf:Description rdf:about="http://szekelys.com/family#pedro">
    <foaf:firstName>Pedro</foaf:firstName>
    <foaf:homepage rdf:resource="http://isi.edu/~szekely"/>
  </rdf:Description>

</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

Predicate

```
<?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

Value

```
<?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


Predicate

Value

```
<?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/">

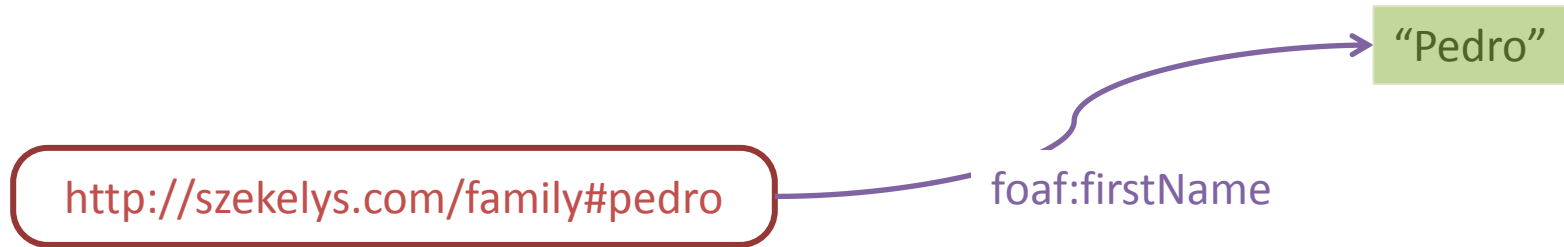
  <rdf:Description rdf:about="http://szekelys.com/family#pedro">
    <foaf:firstName>Pedro</foaf:firstName>
    <foaf:homepage rdf:resource="http://isi.edu/~szekely"/>
  </rdf:Description>

</rdf:RDF>
```



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

XML Syntax

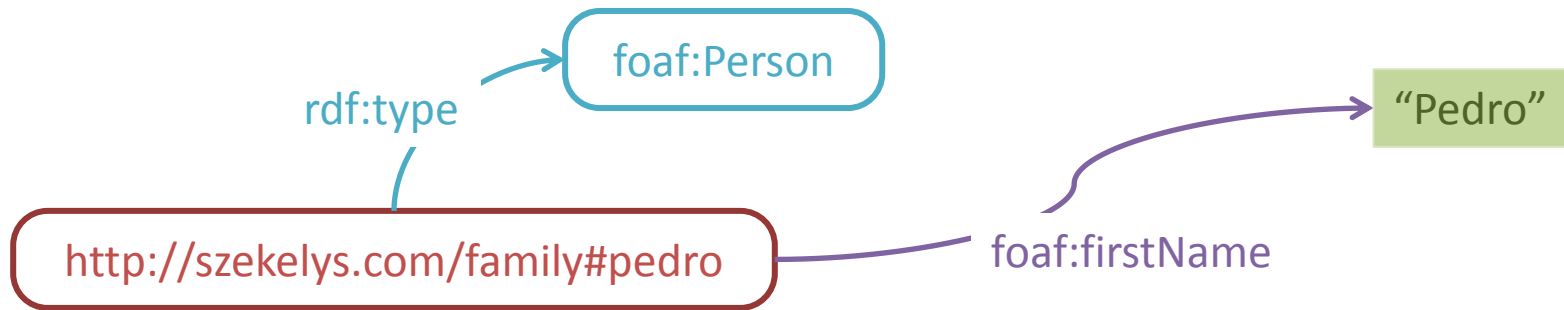


```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">

  <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>
```

RDF Syntaxes

XML

Leverages XML tools
Hard for humans to read

N3, Turtle

Terse RDF Triple Language
Human readable format
Works with software too

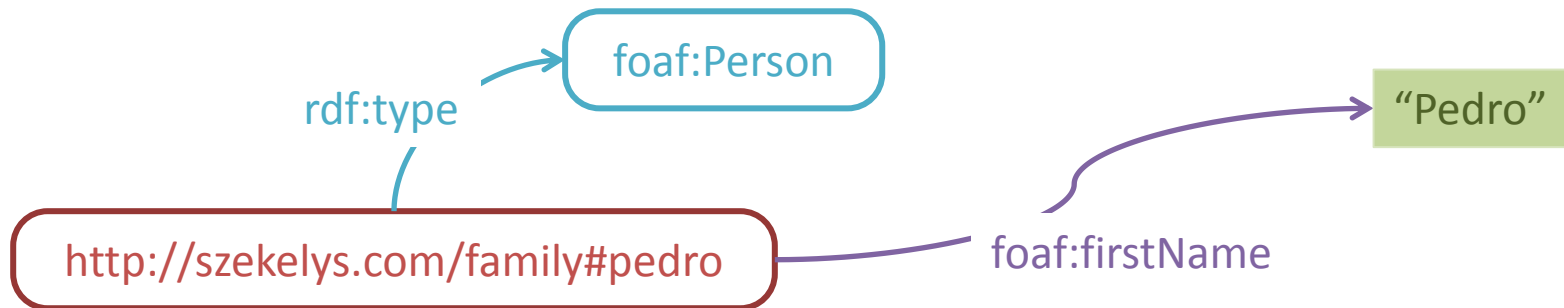
N-Triples

Subset of turtle, supports streaming
Standard for large RDF dumps

RDFa

Allows embedding RDF in HTML pages

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~~

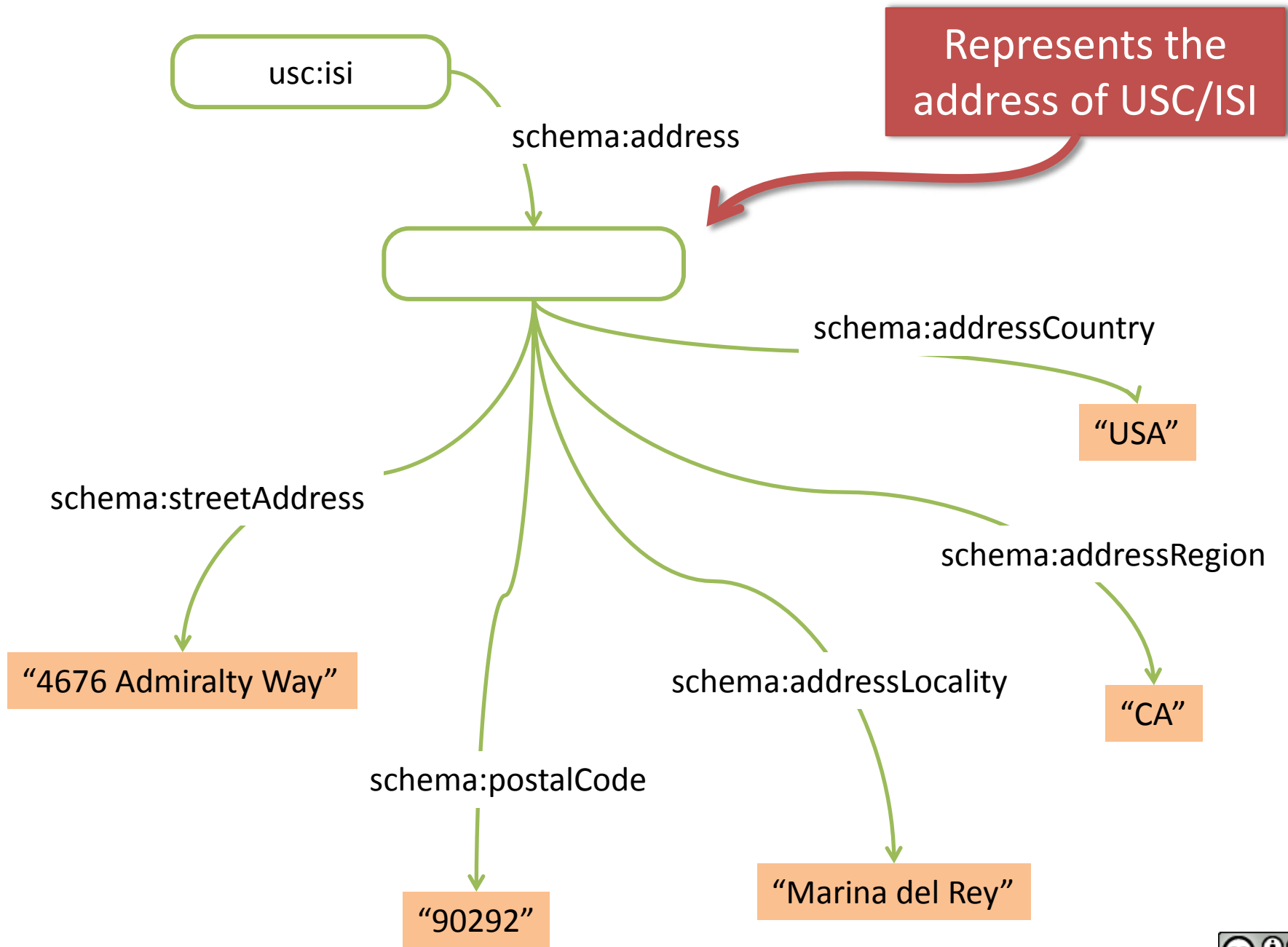
English

“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” .

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

Blank Nodes

Blank node prefix is “_”

usc:isi schema:address :isi-address .

:isi-address

schema:addressCountry “USA” ;
schema:addressRegion “CA” ;
schema:addressLocality “Marina del Rey” ;
schema:postalCode “90292” ;
schema:streetAddress “4676 Admiralty Way” .

... can be improved ...

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?

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

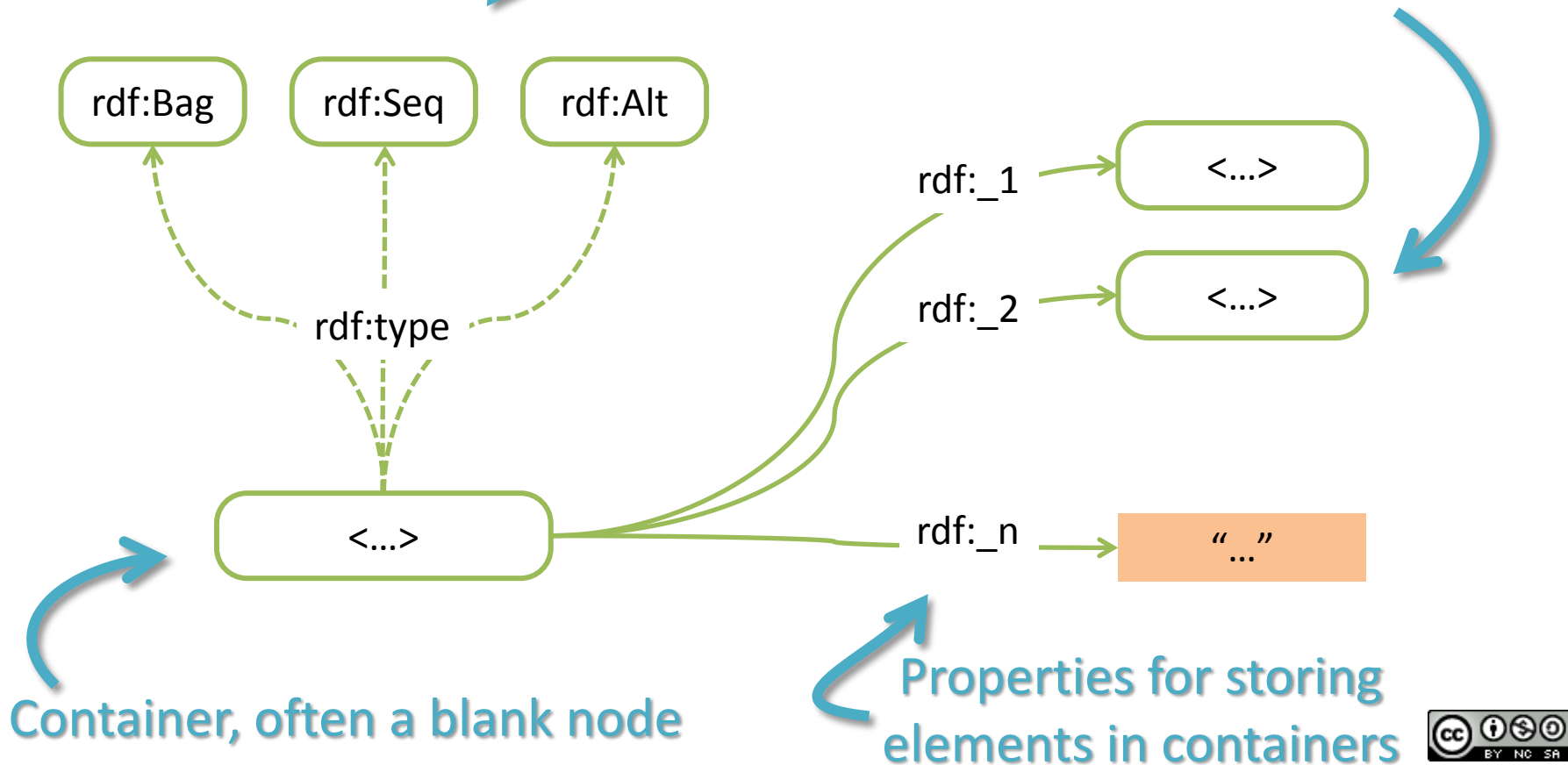
- ... No set of predefined types defined in RDF
- ... Software that consumes RDF must process types
- ... XSD types commonly used

Containers and Collections

Bag, Sequence, Alternative

Kinds of containers

Elements, can be URI or literal



Bag Example

“Three papers that Sue published”

```
exstaff:Sue  exterm:publication  _:z .  
_:z         rdf:type             rdf:Bag .  
_:z         rdf:_1               ex:AnthologyOfTime .  
_:z         rdf:_2               ex:ZoologicalReasoning .  
_:z         rdf:_3               ex:GravitationalReflections .
```

What's the Difference?

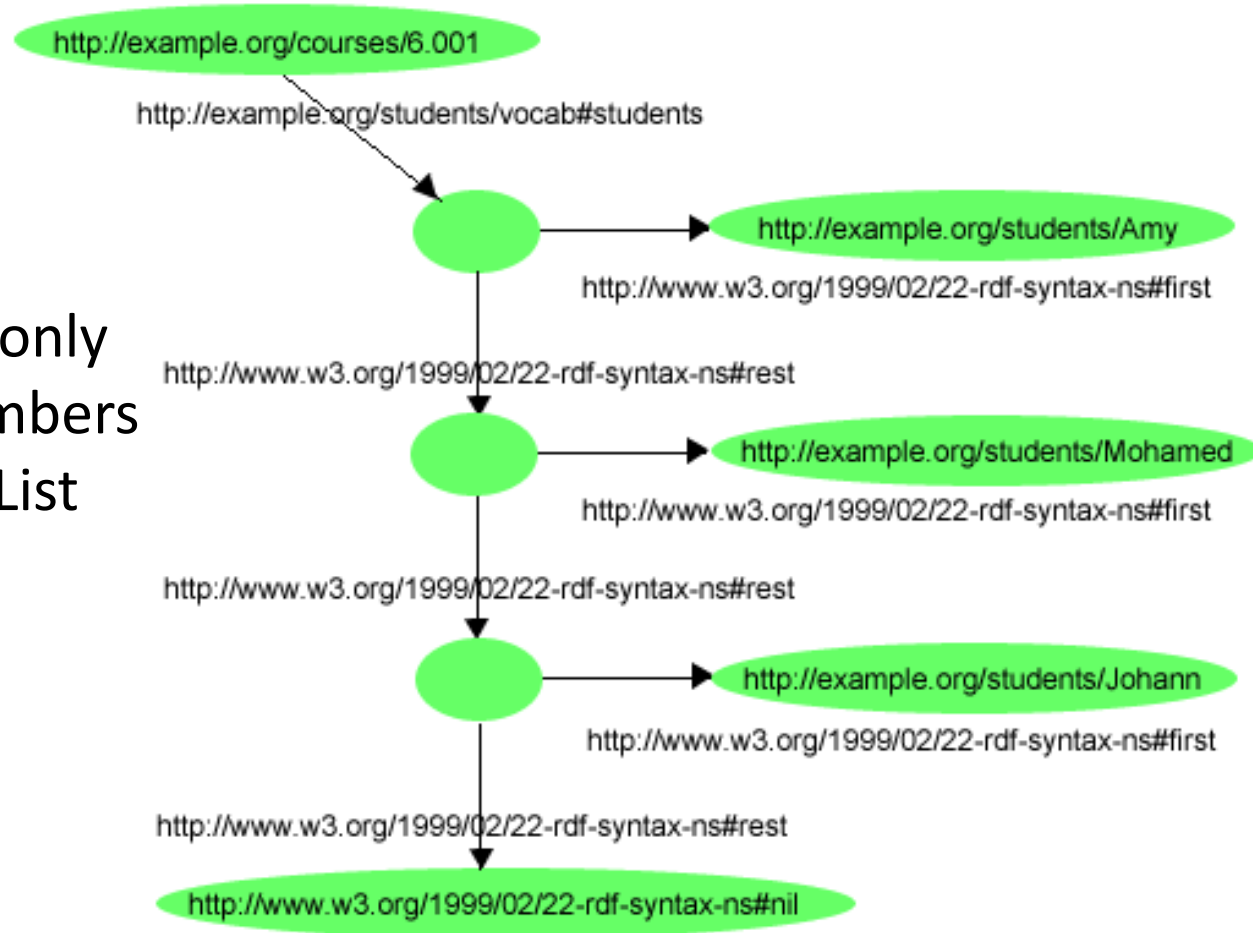
```
exstaff:Sue  exterm:s:publication  _:z .  
_:z         rdf:type              rdf:Bag .  
_:z         rdf:_1                ex:AnthologyOfTime .  
_:z         rdf:_2                ex:ZoologicalReasoning .  
_:z         rdf:_3                ex:GravitationalReflections .
```

```
exstaff:Sue  exterm:s:publication  ex:AnthologyOfTime .  
exstaff:Sue  exterm:s:publication  ex:ZoologicalReasoning .  
exstaff:Sue  exterm:s:publication  ex:GravitationalReflections .
```

List

RDF Collection:

- group containing only the specified members
- represented as a List structure



```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://example.org/students/vocab#">
  <rdf:Description rdf:about="http://example.org/courses/6.001">
    <s:students rdf:parseType="Collection">
      <rdf:Description rdf:about="http://example.org/students/Amy"/>
      <rdf:Description rdf:about="http://example.org/students/Mohamed"/>
      <rdf:Description rdf:about="http://example.org/students/Johann"/>
    </s:students>
  </rdf:Description>
</rdf:RDF>
```

Containers

vs

Collections

RDF philosophy



Open World

Incomplete information

There are things I don't know

Scales to the whole Web

Containers



open world sets

Closed World

Complete Information

If I don't know it, it does not exist

Does not scale

Collections



closed world sets

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>](http://szekelys.com/family#pedro) foaf:mbox [<szekely1401@gmail.com>](mailto: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> .

Problems With Reification

RDF 1

<code>_:s</code>	<code>rdf:type</code>	<code>rdf:Statement .</code>
<code>_:s</code>	<code>rdf:subject</code>	<code><http://szekelys.com/family#pedro> .</code>
<code>_:s</code>	<code>rdf:predicate</code>	<code>foaf:mbox .</code>
<code>_:s</code>	<code>rdf:object</code>	<code><szekely1401@gmail.com> .</code>
<code>_:s</code>	<code>dcterms:date</code>	<code>"2012-06-19"^^xsd:date .</code>
<code>_:s</code>	<code>dcterms:creator</code>	<code><http://uniandes.edu.co/faculty#claudiaj> .</code>

RDF 2

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

RDF 1 implies RDF 2? No!

Problems With Reification

<code>_:s</code>	<code>rdf:subject</code>	<code><http://szekelys.com/family#pedro> .</code>
<code>_:s</code>	<code>rdf:predicate</code>	<code>foaf:mbox .</code>
<code>_:s</code>	<code>rdf:object</code>	<code><szekely1401@gmail.com> .</code>
<code>_:s</code>	<code>dcterms:date</code>	<code>"2012-06-19"^^xsd:date .</code>
<code>_:s</code>	<code>dcterms:creator</code>	<code><http://uniandes.edu.co/faculty#claudiaj> .</code>

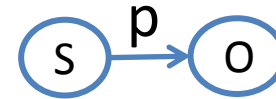
- Needs 3 times more triples
- Most software cannot reason with it
- Nice idea that does not work well!
- Use sparingly, often there is a better way

Views of RDF statements and documents

An RDF statement can be viewed as:

- A triple (subject, predicate, object):
- A piece of a labeled graph:
- A piece of XML code:
- A binary predicate in logic:

s p o .



<s> <p> o </p> <s>

p(s, o)

Thus an RDF document can be viewed as:

- A set of triples
- A directed labeled graph (semantic net)
- An XML document
- A set of logical facts