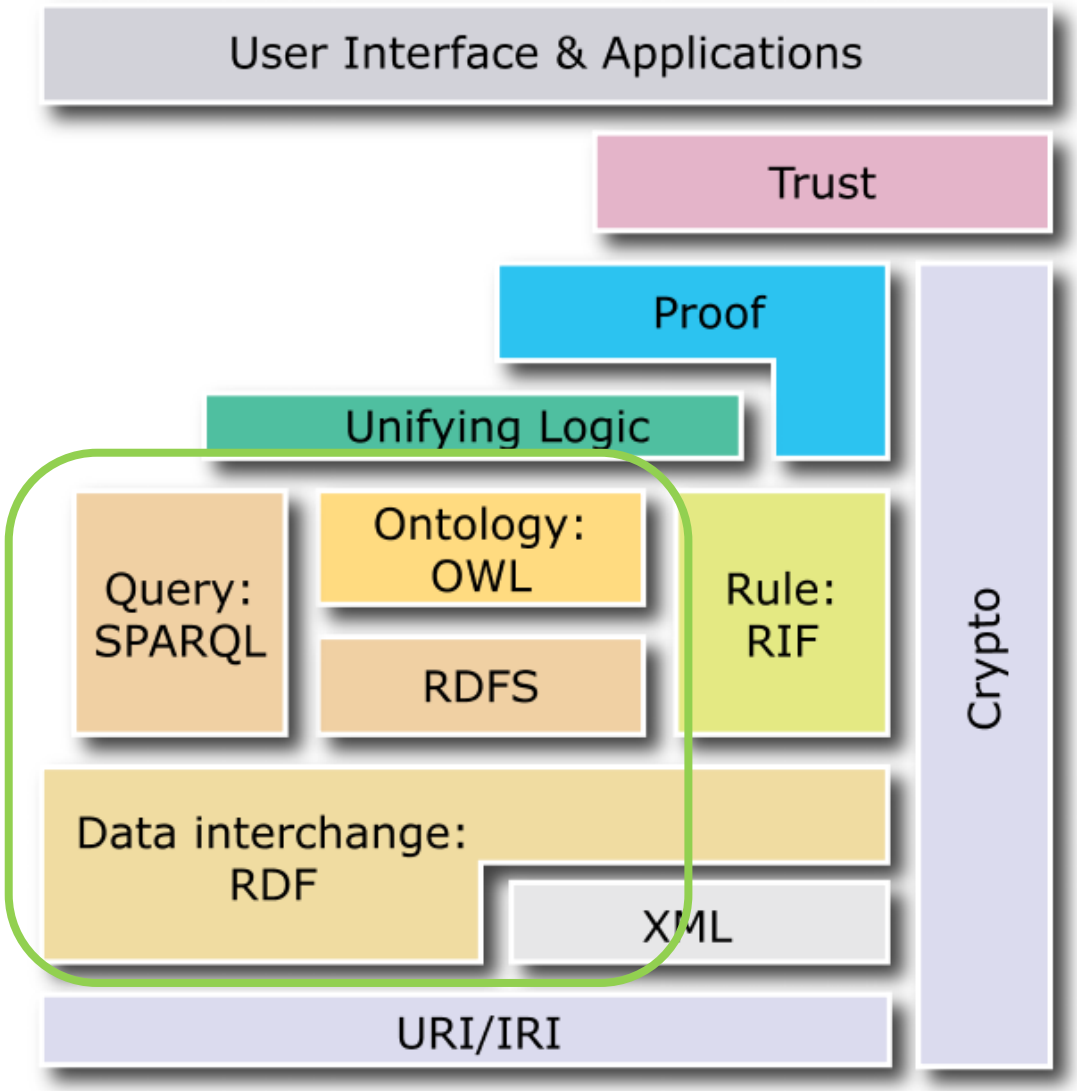
29/6/2022

Basic Ontology Languages

Dr Eleni Tsalapati

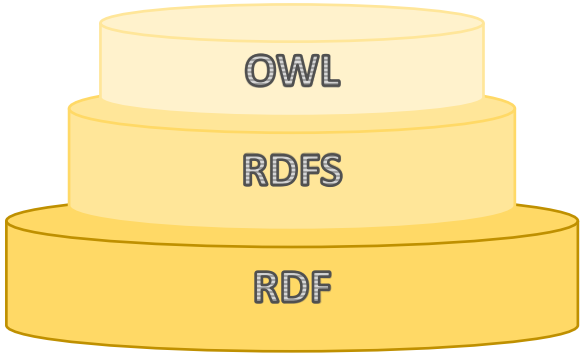
Marie Curie Fellow

The Semantic Web Layer Cake



Standardized Machine Readable KG/Ontology Languages

- RDF
- RDFS
- OWL 2.0
- SWRL



Standardized Machine Readable KG/Ontology Languages

RDF: represent data as URIs and in graphical form

For describing facts
Data integration from multiple resources
Detachment of data from their schema

RDFS: adds schema to the RDF

Adds constraints on the facts
Hierarchies, domains & ranges of properties
Enables basic inferencing – infer new triples

OWL 2.0: higher expressivity –adds more constraints

Enables more complex inferencing

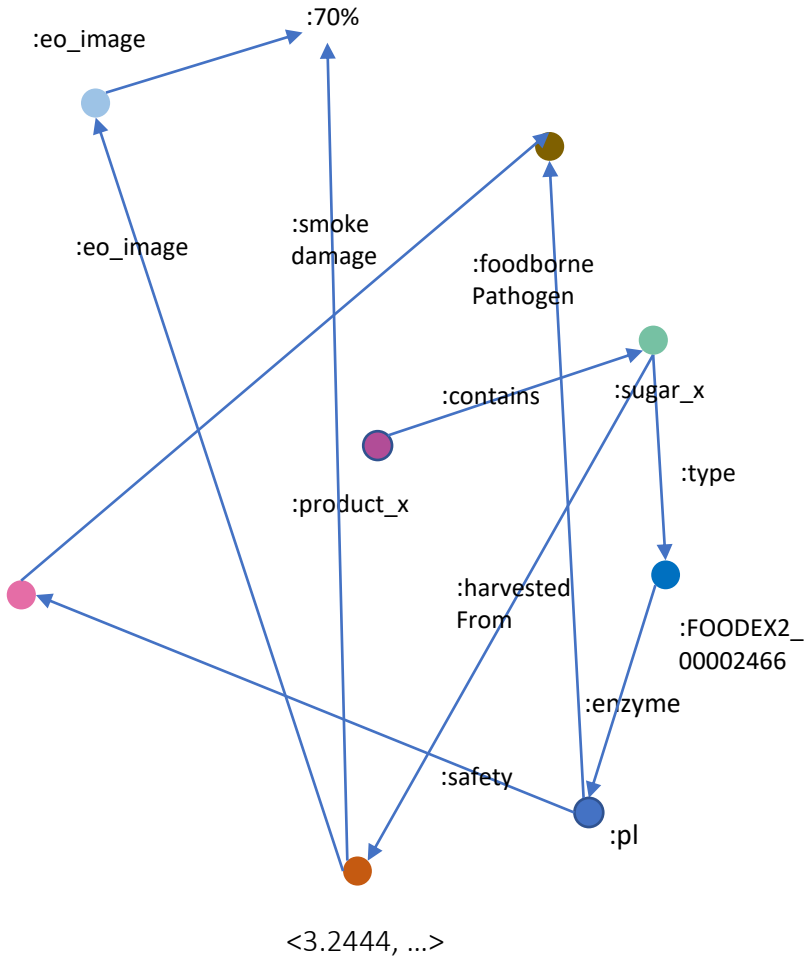
SWRL: rule language, intuitive, adds expressivity

Core Idea

- Information no longer on sheer sheets of data

But:

- **Each** piece of information is represented as a **unique node** and the nodes are **interrelated** with **labeled links**
- From human readability to machine **processibility**



<3.2444, …>

Resource Description Framework

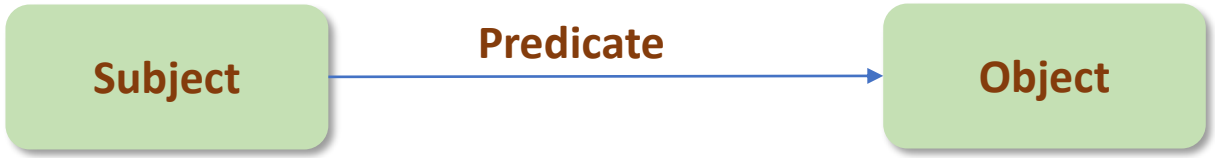
- The Resource Description Framework (RDF):
  - **Data model** originally for representing information (especially **metadata**) about web resources
- Now is used to describe any data and not only metadata
- Easy, powerful, expressive W3C **standard**
- To represent information about things that can be **identified,** even when they **cannot be directly retrieved (e.g., a book or a person).**
- For data to be **processed by applications**, rather than being only displayed to people.

Resource Description Framework

- RDF draws upon ideas from knowledge representation, artificial intelligence, and data management, including:
  - Semantic networks
  - Frames
  - Conceptual graphs
  - Logic-based knowledge representation
  - Relational databases
- The closest to RDF, pre-Web knowledge representation language is Telos:
  - John Mylopoulos, Alexander Borgida, Matthias Jarke, Manolis Koubarakis: Telos: Representing Knowledge About Information Systems. ACM Trans. Inf. Syst. 8(4): 325-362 (1990).

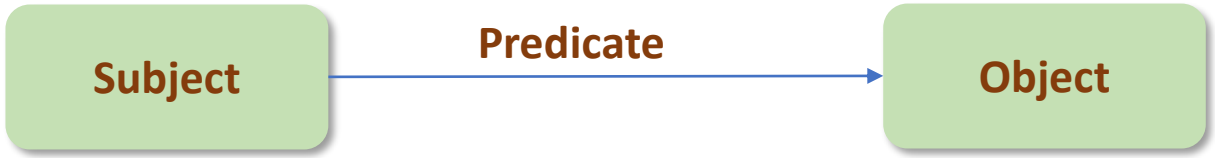
Resource Description Framework

- **Basic idea:**
  - Data objects are identified as web identifiers (URIs)
  - Definition of relationships (URIs) between data objects



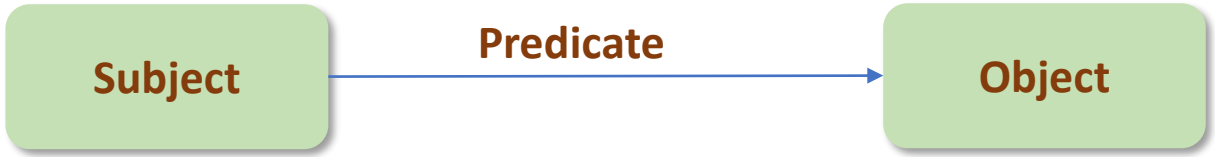
Resource Description Framework

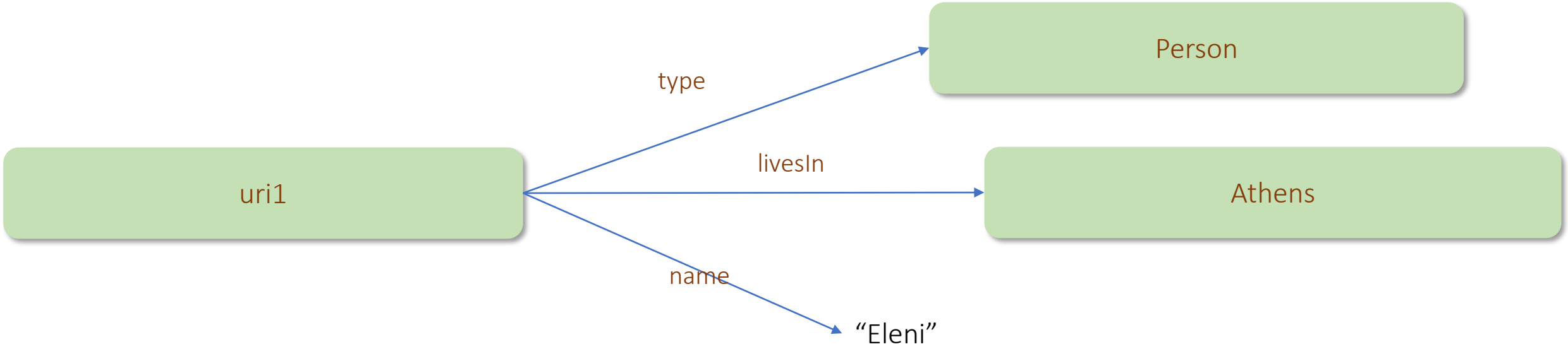
- **Basic idea:**
  - Data objects are identified as web identifiers (URIs)
  - Definition of relationships (URIs) between data objects



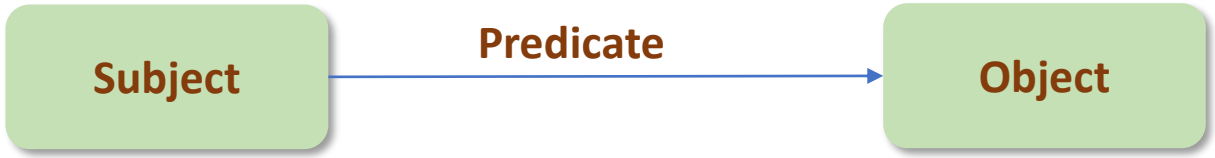
Represent in RDF this: There is a Person identified by uri1 whose name is Eleni and lives in Athens

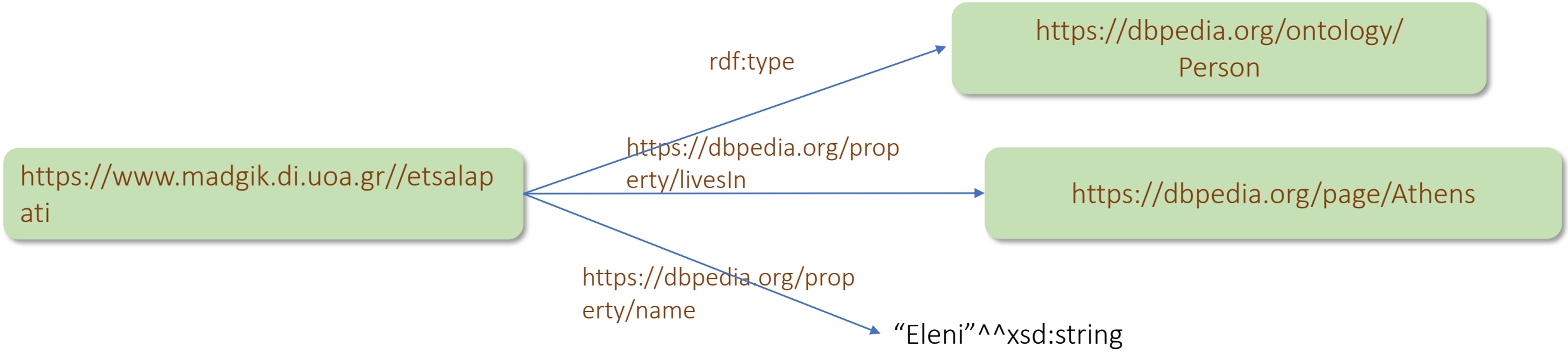
Resource Description Framework



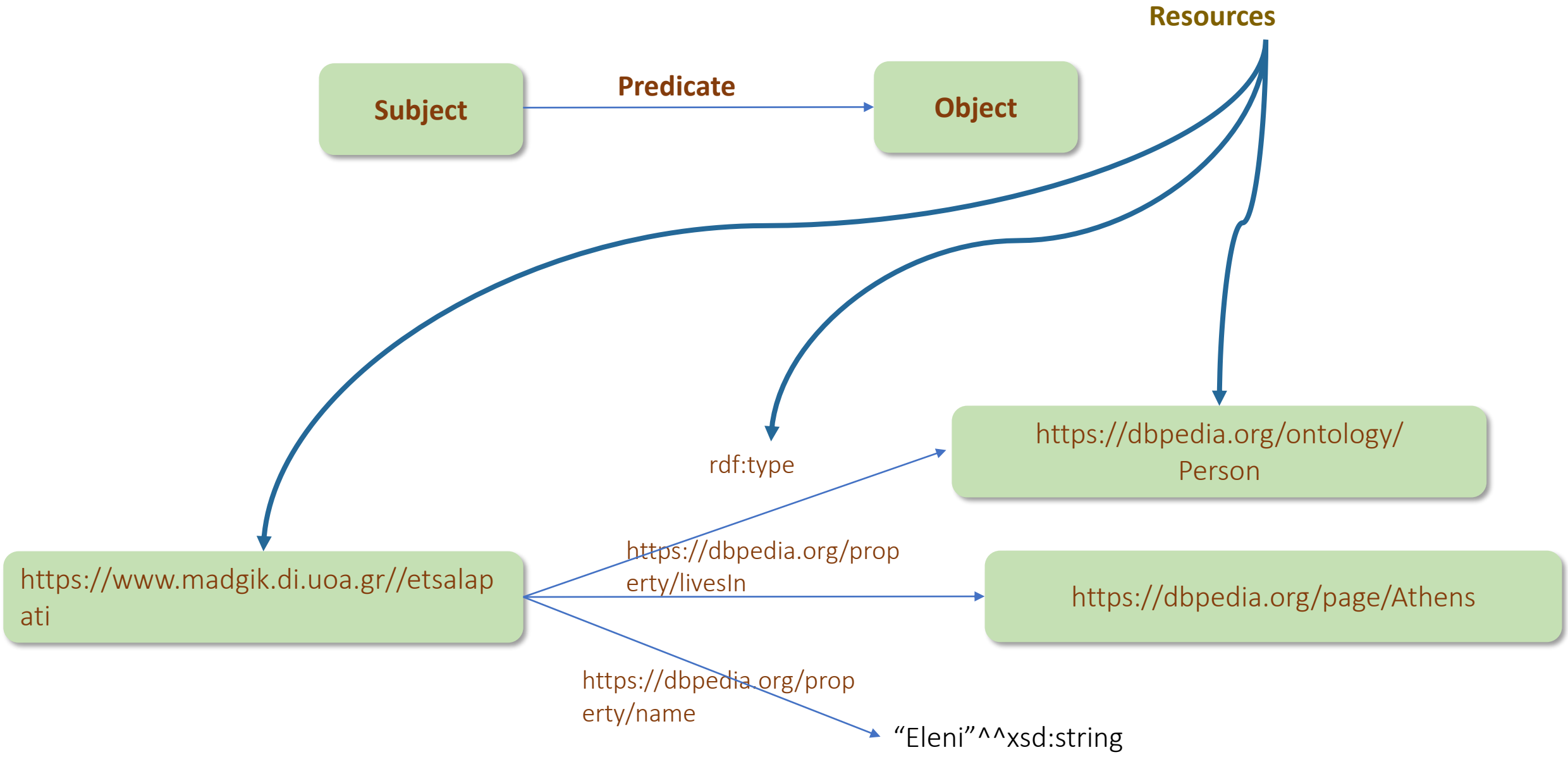


Resource Description Framework

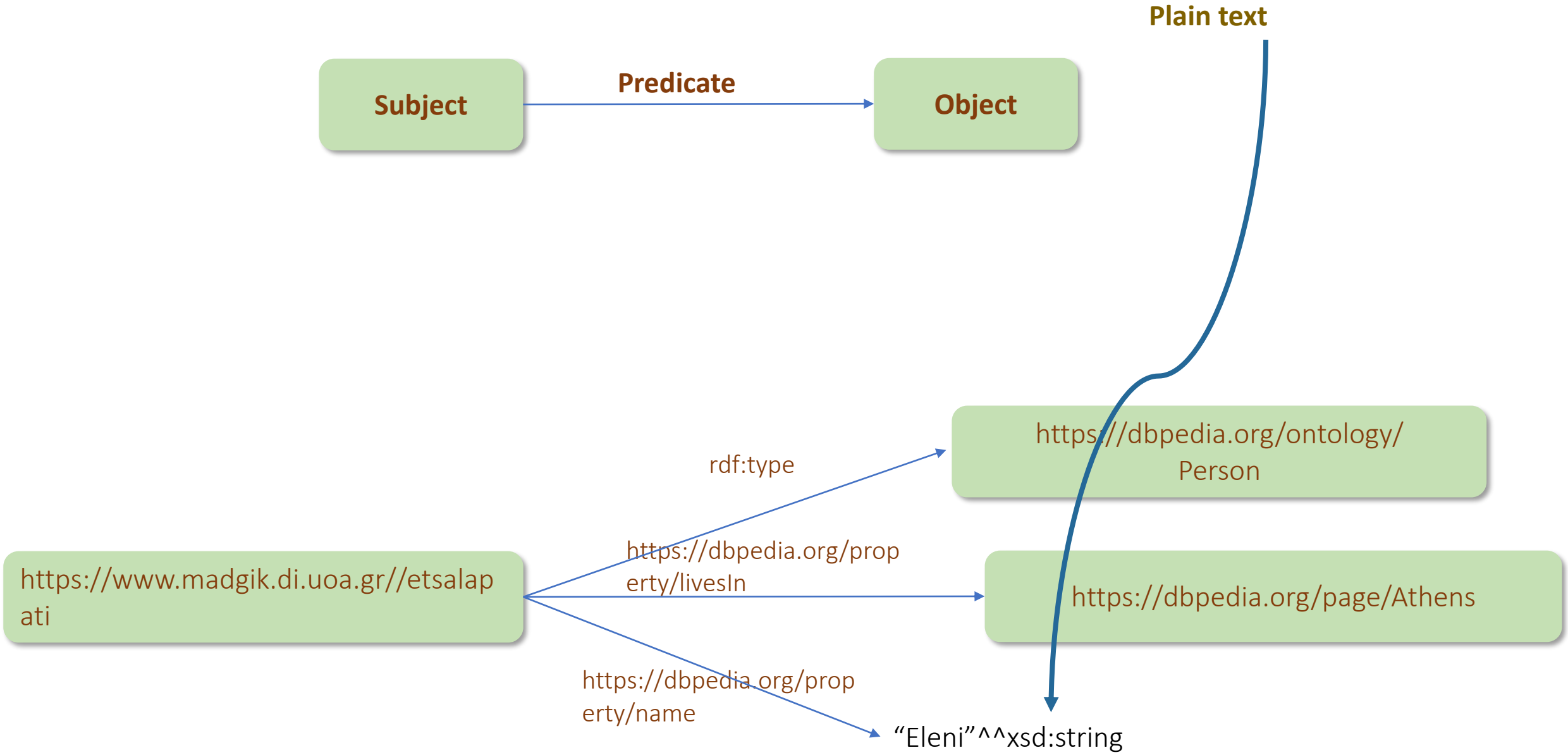




Resource Description Framework



Resource Description Framework



Resource Description Framework

- RDF statements can be written in triple notation:

subject predicate object .

<https://www.madgik.di.uoa.gr/etsalapati>
<https://dbpedia.org/property/livesIn>
<https://dbpedia.org/page/Athens> .

<https://www.madgik.di.uoa.gr/etsalapati>
<https://dbpedia.org/property/name> “Eleni”^^xsd:string .

Resource Description Framework

- RDF statements can be written in triple notation:

subject predicate object .

<https://www.madgik.di.uoa.gr/etsalapati>
<https://dbpedia.org/property/livesIn>
<https://dbpedia.org/page/Athens> .

<https://www.madgik.di.uoa.gr/etsalapati>
<https://dbpedia.org/property/name> “Eleni”^^xsd:string .

URI reuse

- Reuse existing URIs from well-known vocabularies/ontologies (e.g., DBPedia, Dublin Core)
  - Less effort
  - Shared understanding of the resulted ontology
  - Ambiguity is eliminated
  - Can be looked-up

NY or GR?

<https://www.madgik.di.uoa.gr/etsalapati>
<https://dbpedia.org/property/livesIn>
<https://dbpedia.org/page/Athens> .

RDF and Related Data Models

- In terms of the **relational model**, an RDF statement is similar to a *tuple* in a relation called *Triples* or *Graph* with attributes *Subject, Predicate and Object*.

- In terms of **first-order logic**, an RDF statement is similar to an atomic formula
  - triple(subj,pred,obj)

where triple is a first-order logic predicate and subj, pred and obj are constants.

Uniform Resource Identifiers

- **Uniform:** URI follows syntax rules to ensure uniformity
- **Identity**: **identify (name) resources on the Web**.
- URIs are **not** limited to identifying things that have network locations, or use other computer access mechanisms.
- A number of different **URI schemes** (URI forms) have been already been developed, and are being used, for various purposes.
- Examples:
  - http: (Hypertext Transfer Protocol, for Web pages)
  - mailto: (email addresses), e.g., mailto:em@w3.org
  - ftp: (File Transfer Protocol)

Syntax of URI

URI = scheme:[//authority]path[?query][#fragment]

scheme:[LETTER][DIGIT|LETTER|.|-|+]

authority (optional):

user info component (optional)

host subcomponent: IP address/registered name

: [port] (optional)

path: sequence of segments that are separated by a slash

query(optional): query string of non-hierarchical data

fragment: identifier giving direction to a secondary resource (then we have a URI ref)

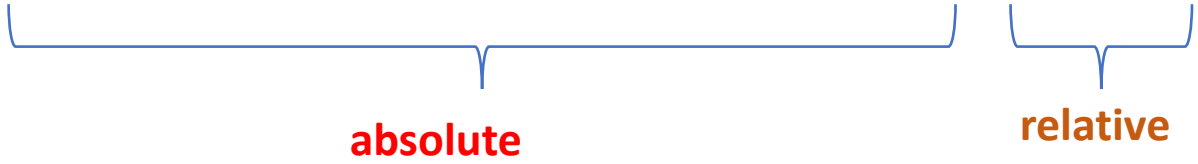
URI refs

- A URI reference (or URIref) is a URI, together with an optional fragment identifier at the end.

- The URIref: http://www.example.org/index.html#section2

URI refs

- URIrefs may be either absolute or relative.
- The URIref: http://www.example.org/index.html#section2



- An absolute URIref refers to a resource independently of the context in which the URIref appears
- A relative URIref is a shorthand form of an absolute URIref, where some prefix of the URIref is missing

URI refs

- A relative URIref consisting of just a fragment identifier:

- The URIref: #section2

is considered equivalent to the URIref of the document in which it appears (http://www.example.org/index.html)

with the fragment identifier appended to it:
http://www.example.org/index.html#section2

URI refs

- **RDF** defines a **resource** as anything that is identifiable by a URI reference
- Using URIrefs allows RDF to describe practically **anything**, and to state relationships between such things as well

- RDF uses **Internationalized Resource Identifiers** (**IRIs**) and IRIrefs.
- To have also **non-latin languages** for the description of resources.

URI refs in RDF

- RDF and Web browsers use URIrefs to identify things.
- They interpret URIrefs in slightly different ways:
  - RDF uses URIrefs only to identify things.
  - Browsers also use URIrefs to retrieve things.

- What is the difference?
  - In a browser, identifies a resource that can actually be retrieved: that something is actually "at" the location identified by the URI.
  - In RDF, identifies something, such as a person, that cannot be retrieved on the Web.
  - But important uses of RDF, like Linked Data (http://linkeddata.org/), insist that we use HTTP URIs so data identified by a URI can be retrieved.

URI refs in RDF

- Another difference is in the way URIrefs with fragment identifiers are handled. Consider the following URIrefs:
  - http://www.example.org/index.html
  - http://www.example.org/index.html#Section2

- In normal HTML usage, these URIrefs are related (they both refer to the same document, the second one identifying a location within the first one).

- RDF assumes **no particular relationship** between these two URIrefs. As far as RDF is concerned, they are syntactically different URI references, and hence may refer to unrelated things.

URIs

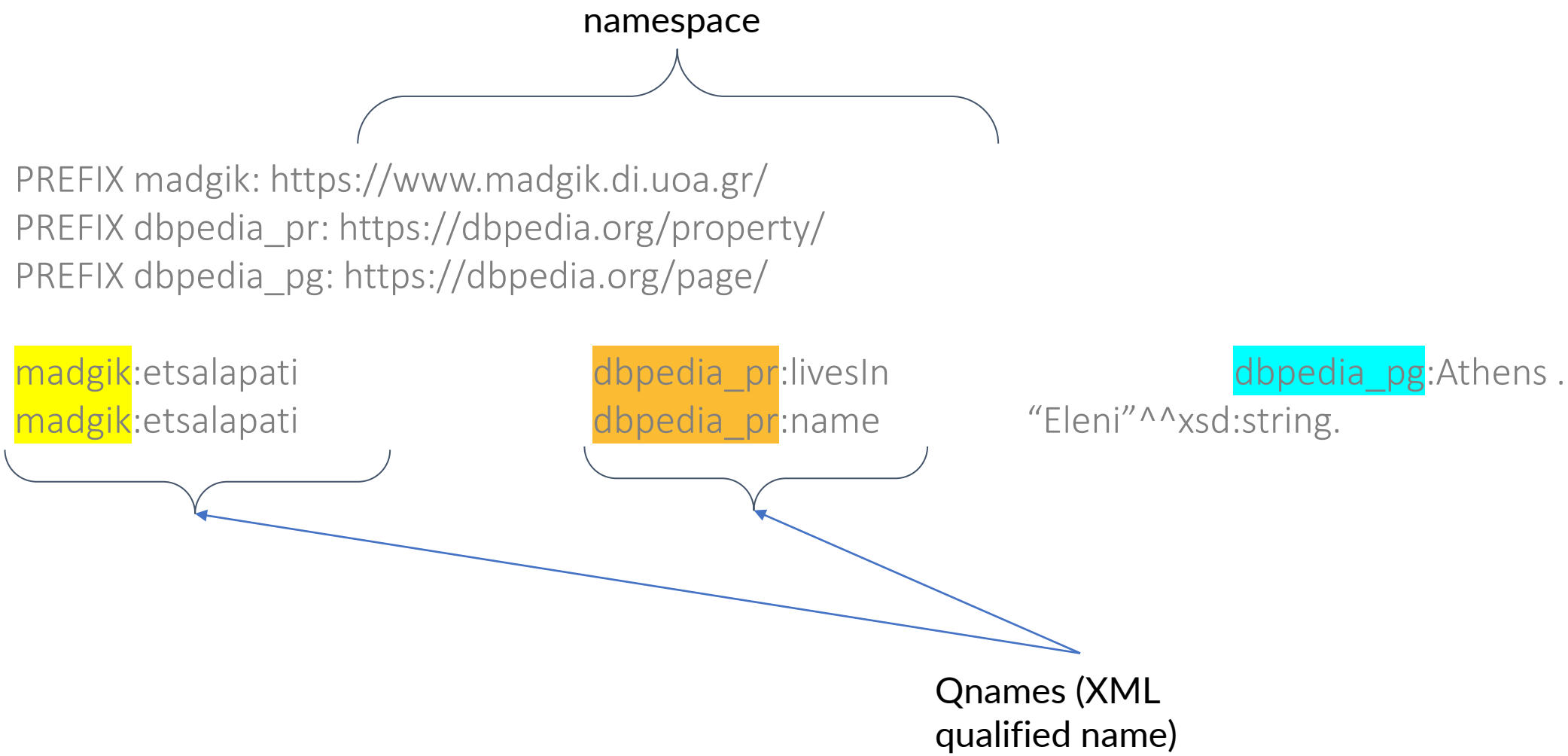
- The full URIs are too verbose:

<https://www.madgik.di.uoa.gr/etsalapati>
<https://dbpedia.org/property/livesIn>
<https://dbpedia.org/page/Athens> .

<https://www.madgik.di.uoa.gr/etsalapati>
<https://dbpedia.org/property/name> “Eleni”^^xsd:string.

Prefixes

- The full URIs are too verbose:



Prefixes

- The full URIs are too verbose:

PREFIX madgik: https://www.madgik.di.uoa.gr/
PREFIX dbpedia_pr: https://dbpedia.org/property/
PREFIX dbpedia_pg: https://dbpedia.org/page/

madgik:etsalapati dbpedia_pr:livesIn dbpedia_pg:Athens .
madgik:etsalapati dbpedia_pr:name “Eleni”^^xsd:string.

Prefixes- More Examples:

prefix rdf:, namespace URI: http://www.w3.org/1999/02/22-rdf-syntax-ns#
prefix rdfs:, namespace URI: http://www.w3.org/2000/01/rdf-schema#
prefix dc:, namespace URI: http://purl.org/dc/elements/1.1/
prefix owl:, namespace URI: http://www.w3.org/2002/07/owl#
prefix ex:, namespace URI: http://www.example.org/ (or http://www.example.com/)
prefix xsd:, namespace URI: http://www.w3.org/2001/XMLSchema#

XML Namespaces

- A namespace is a way of **identifying a subset of a set of names** (e.g., the set of possible names of resources in the Web) **which acts as a qualifier for the names in this subset**.

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

- XML namespaces help us eliminate ambiguity in an XML document. For example, an XML document can use **id** to refer to **both identifiers of customers and products** if id is prefixed by an appropriate name space (e.g., http://customers.org and http://products.com).

- A namespace **is created by creating a URI** for it. By qualifying names with the URIs of their namespaces, **anyone can create their own names** and properly **distinguish** them from names with identical spellings **created by others**.

URIrefs as Vocabulary

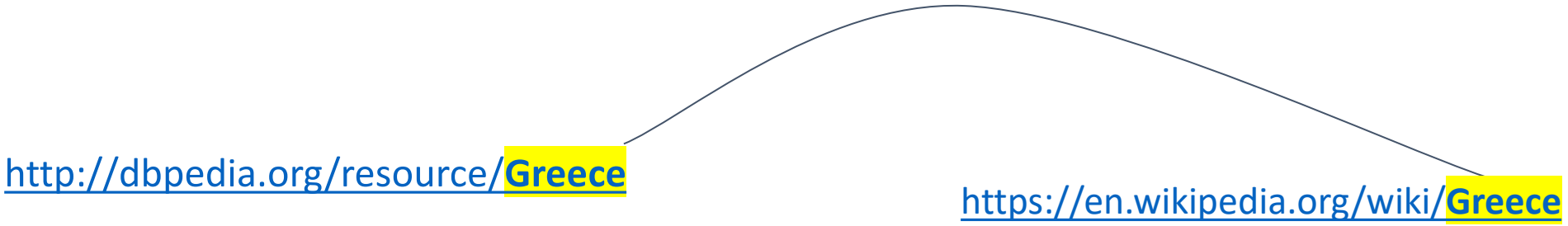
- Since RDF uses **URIrefs instead of words** to name things in statements, URIrefs define **vocabularies in RDF**.

- The URIrefs in RDF vocabularies are typically **organized** so that they can be represented as **a set of QNames with a common prefix**:
- A **common namespace URIref** is chosen for all terms in a vocabulary, typically a URIref under the control of whoever is defining the vocabulary.
- URIrefs that are **contained in the vocabulary** are formed by appending **individual local names** to the **end of the common URIref**.

Example

- DBpedia (http://wiki.dbpedia.org/About) is a large knowledge base which has been derived from **Wikipedia** by extracting various kinds of **structured information** from Wikipedia editions in 14 languages and combining this information into a huge, cross-domain knowledge base.
- In the DBpedia data set, each **thing** is identified by a URIref of the form **http://dbpedia.org/resource/Name**, where term "**Name**" is taken from the URL of the source Wikipedia article, which has the form **http://en.wikipedia.org/wiki/Name**.
- Thus, each resource is tied directly to an English-language Wikipedia article.

DBpedia (cont'd)



- The prefix **dbpedia** can be used instead of http://dbpedia.org/resource/
- For example: dbpedia:Greece

URIrefs as Vocabulary (cont'd)

- **RDF** uses **this same approach to define its own vocabulary of** terms with special meanings in RDF:
- The URIrefs in the RDF vocabulary all begin with:
  - **http://www.w3.org/1999/02/22-rdf-syntax-ns#,** conventionally, with prefix rdf:.
- The RDF Vocabulary Description Language defines an additional set of terms having URIrefs that begin with
  - **http://www.w3.org/2000/01/rdf-schema#,** conventionally prefix rdfs:.
- QName prefix itself is sometimes used as the name of the vocabulary. For example, someone might refer to "the rdfs: vocabulary".

URIrefs as Vocabulary (cont'd)

- **Convention:** Organizations typically use a **vocabulary's namespace URIref as the URL** of a Web resource that **provides further information** about that vocabulary.

- **Example:** the QName prefix **dc**: with the namespace URIref `http://purl.org/dc/elements/1.1` refers to the **Dublin Core vocabulary**.
  - Accessing this namespace URIref in a Web browser will retrieve additional information about the Dublin Core vocabulary (specifically, RDFS definitions of the Dublin core vocabulary).
  - **Reminder:** this is just a useful convention. RDF does not assume that a namespace URI identifies a retrievable Web resource.

URIrefs as Vocabulary (cont'd)

- Using URIrefs as subjects, predicates, and objects in RDF statements supports the **development and use of shared vocabularies** on the Web.

- People can **discover and begin using vocabularies** already used by others to describe things, reflecting a **shared understanding of those concepts.**

Example

- Consider the triple

```
ex:index.html dc:creator exstaff:85740.
```

The predicate dc:creator, when fully expanded as a URIref, is an **unambiguous reference** to the "creator" attribute in the Dublin Core metadata attribute set, a widely-used set of attributes (properties) for describing a wide range of networked resources (see http://dublincore.org/documents/usageguide/).

The writer of this triple is effectively saying that the **relationship** between the Web page and the creator of the page is exactly the concept identified by http://purl.org/dc/elements/1.1/creator.

Another person familiar with the Dublin Core vocabulary, or who finds out what dc:creator means (say by looking up its definition on the Web) **will know what is meant by this relationship**. In addition, based on this understanding, people can **write programs to behave in accordance with that meaning** when processing triples containing the predicate dc:creator.

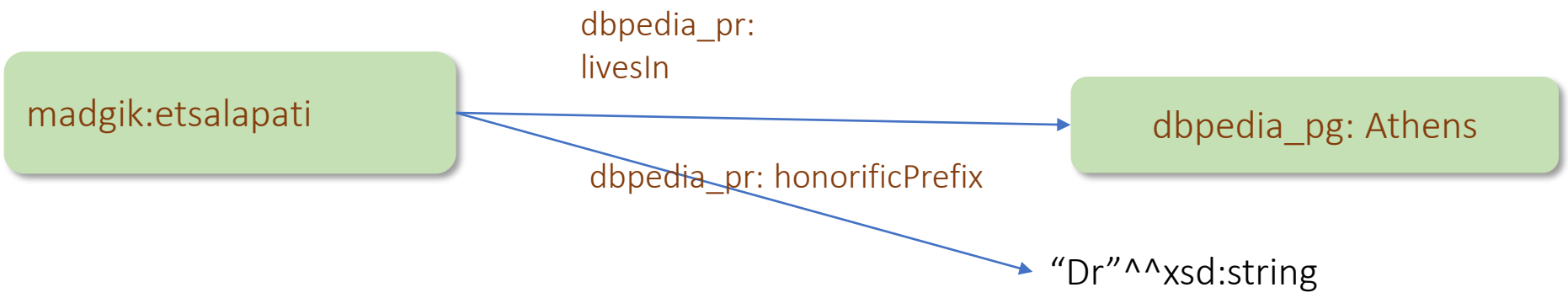
Another Example

- The Friend of a Friend (FOAF) vocabulary at http://xmlns.com/foaf/spec/.

- The FOAF project is creating a Web of machine-readable pages (written in RDF) describing people, the links between them and the things they create and do.

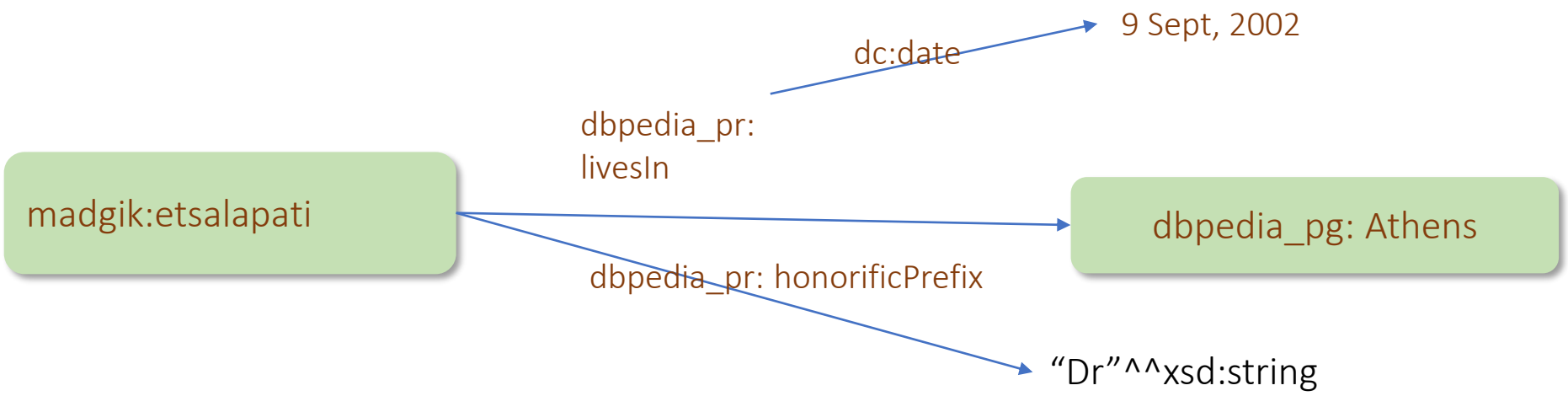
RDF Graph

- There can be three kinds of nodes in an RDF graph:
  - URIs/IRIs
  - RDF literals
  - Blank nodes



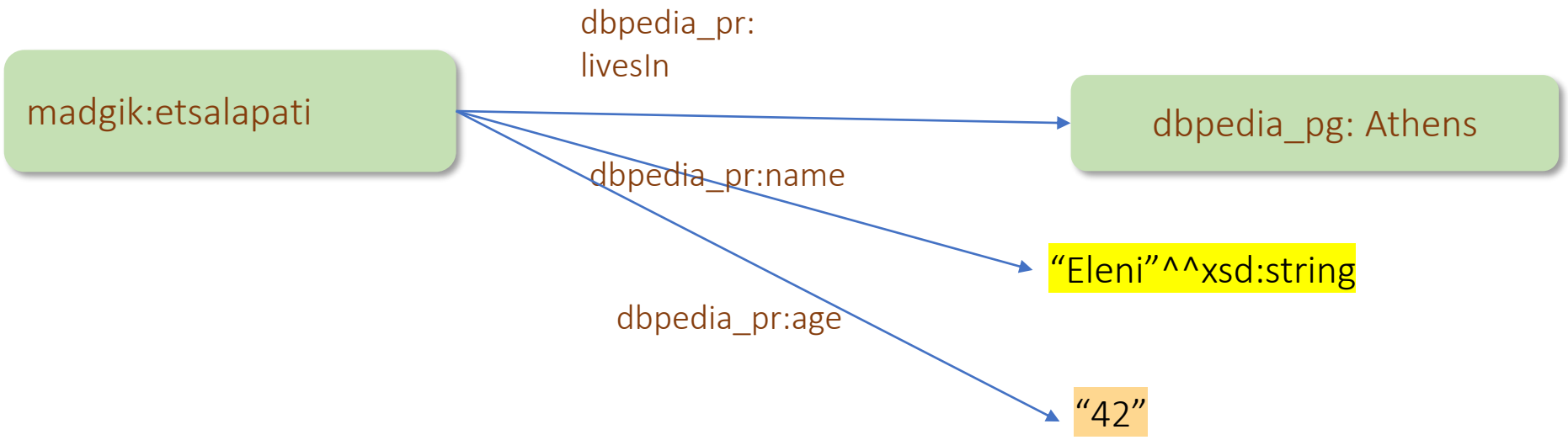
RDF Graph (??)

- We draw RDF graphs as directed graphs.
- But **directed graphs are not sufficient for capturing all of RDF**
- Directed graphs assume that the sets of nodes and arcs are disjoint
- But: RDF allows a property as a subject of a statement).



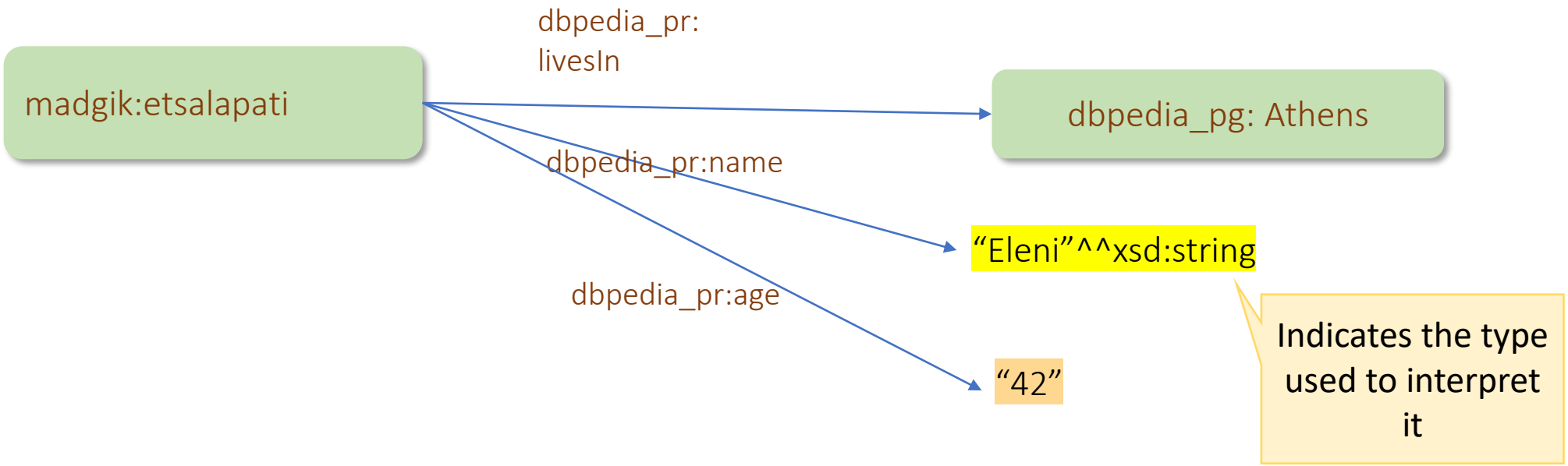
RDF typed literals

- There can be three kinds of nodes in an RDF graph:
  - URIs/IRIs
  - **RDF literals**
    - **Typed literals**
    - **Plain literals**



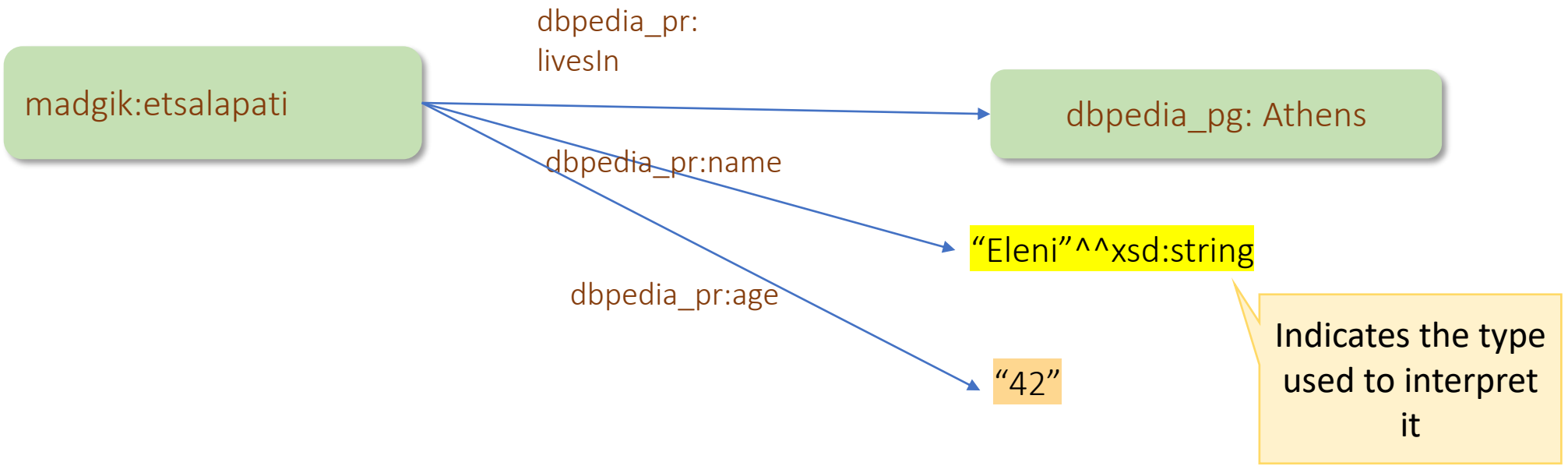
RDF typed literals

- There can be three kinds of nodes in an RDF graph:
  - URIs/IRIs
  - **RDF literals**
    - **Typed literals**
    - **Plain literals**



RDF typed literals

- An RDF typed literal is formed by pairing a string with a **URIref** that identifies a particular datatype:
- “27”^^http://www.w3.org/2001/XMLSchema#integer



Typed Literals

- RDF defines one built-in datatype with the URIref rdf:XMLLiteral to represent XML content as a literal value.
- For bringing into an RDF sentence, content that has already been defined in XML
  - e.g., in the case of spatial data, a polygon defined in GML
- For the rest of the literals we can use datatypes from other formalisms or we define own datatypes.

Typed Literals

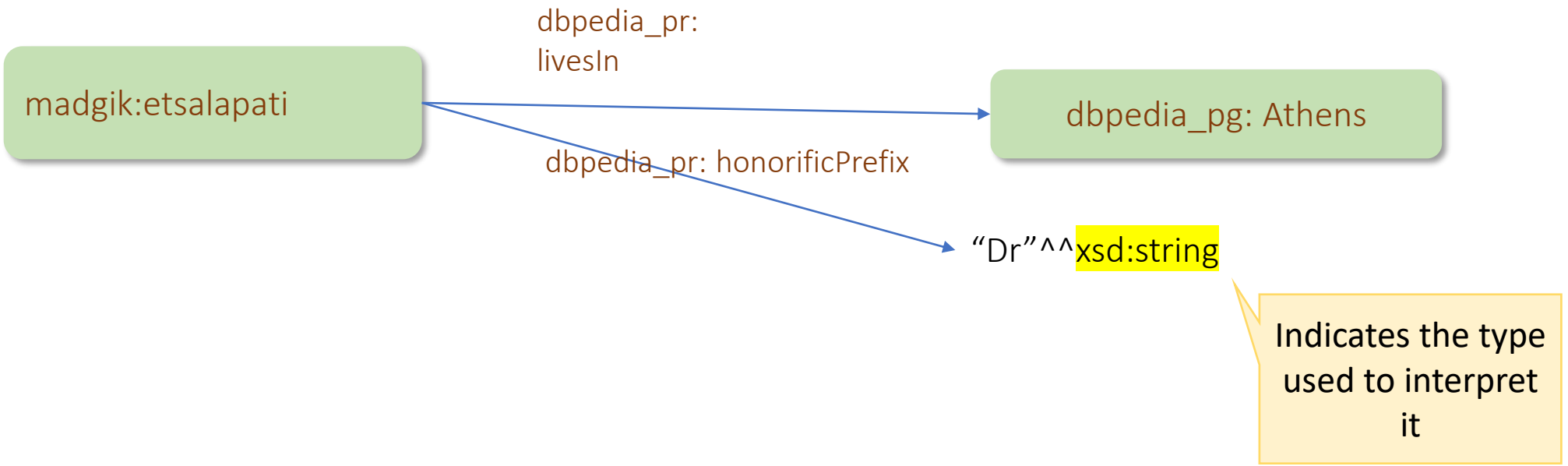
- RDF datatype concepts are based on a conceptual framework from XML Schema datatypes.
- This framework defines the value space, the lexical space and the lexical-to-value mapping for a datatype (see the RDF specifications for more details).

XML Schema datatypes:

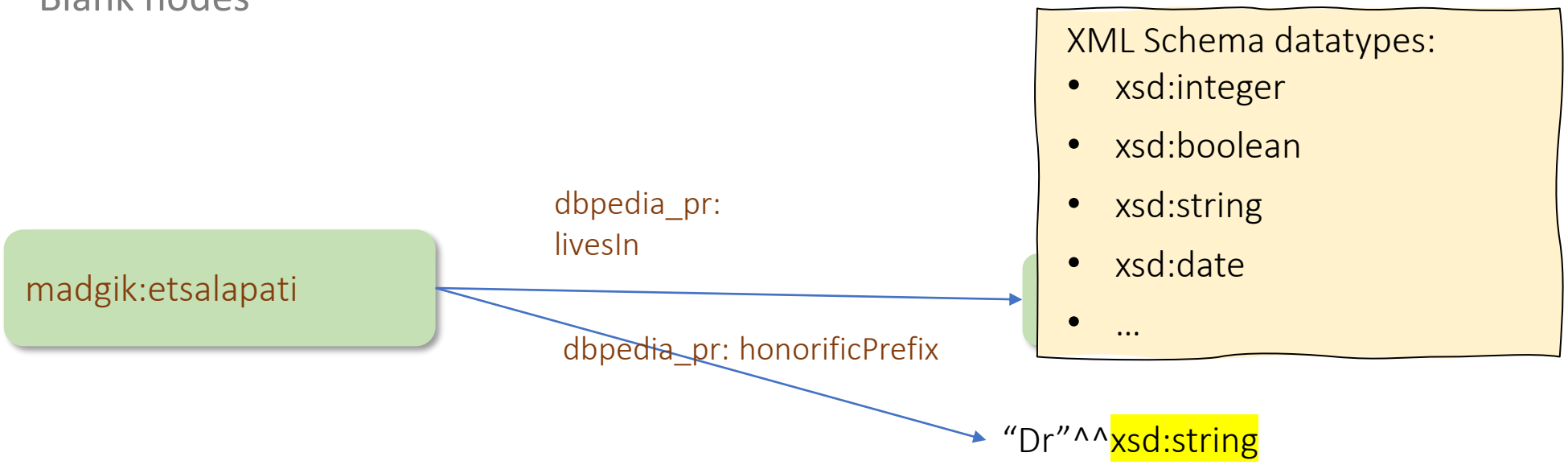
- xsd:integer
- xsd:boolean
- xsd:string
- xsd:date
- …

RDF typed literals

- There can be three kinds of nodes in an RDF graph:
  - URIs/IRIs
  - **RDF typed literals**
  - Blank nodes

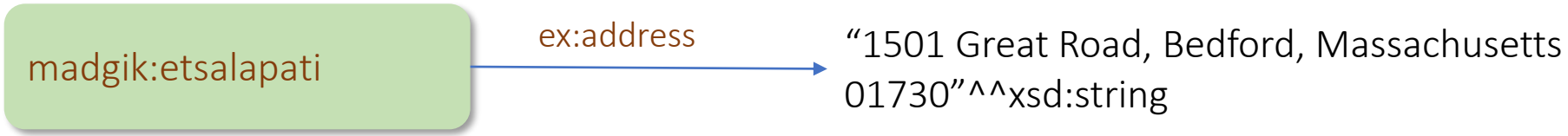


- There can be three kinds of nodes in an RDF graph:
  - URIs/IRIs
  - **RDF typed literals**
  - Blank nodes



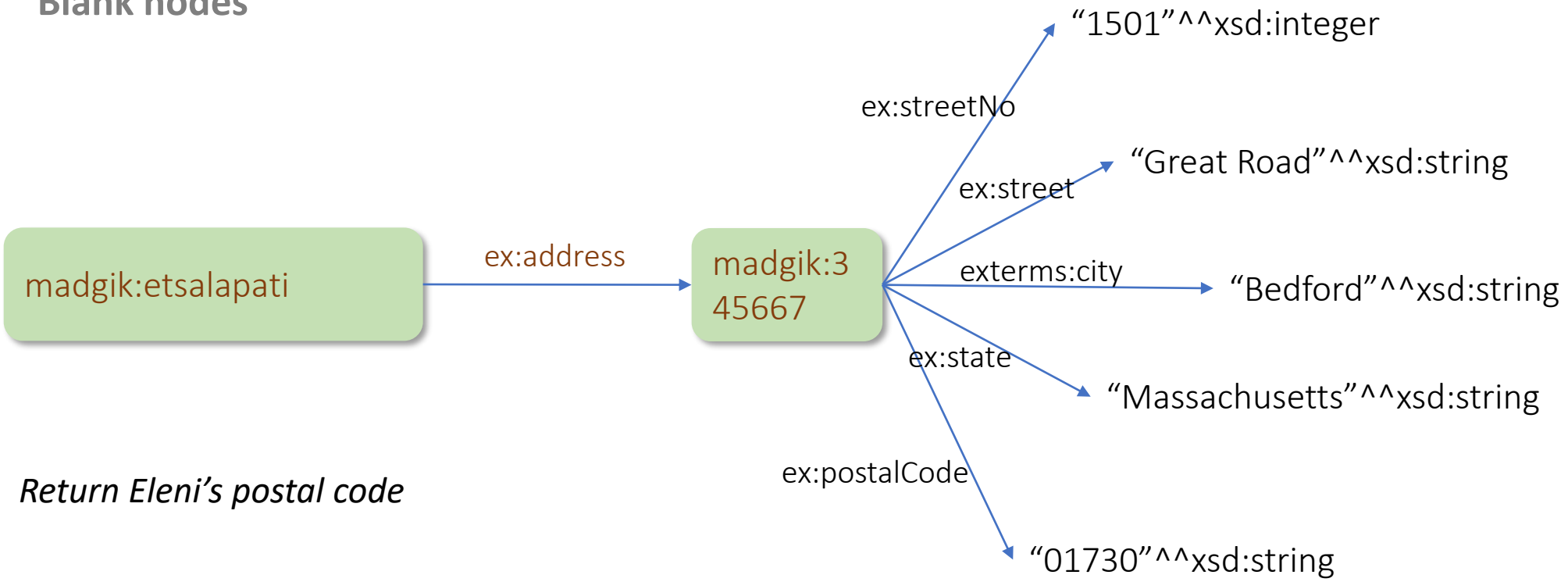
RDF: Blank Nodes

- There can be three kinds of nodes in an RDF graph:
  - URIs/IRIs
  - RDF typed literals
  - **Blank nodes**



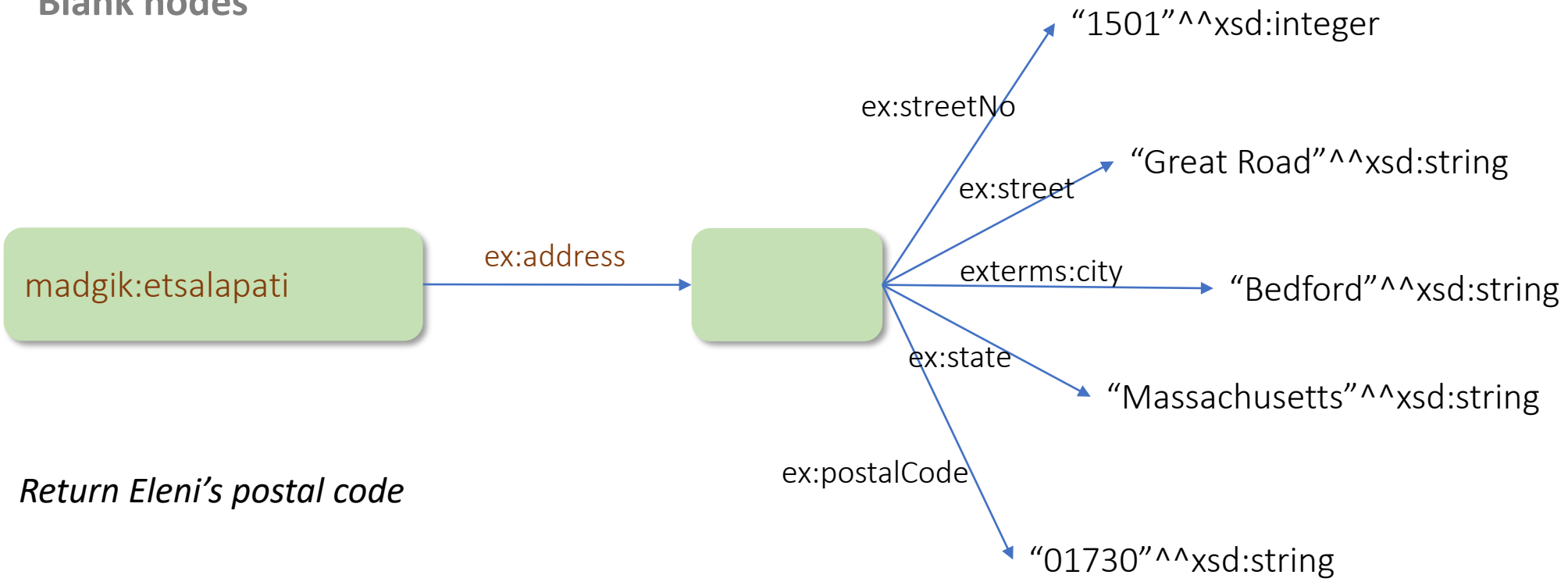
Return Eleni’s postal code

- There can be three kinds of nodes in an RDF graph:
  - URIs/IRIs
  - RDF typed literals
  - **Blank nodes**



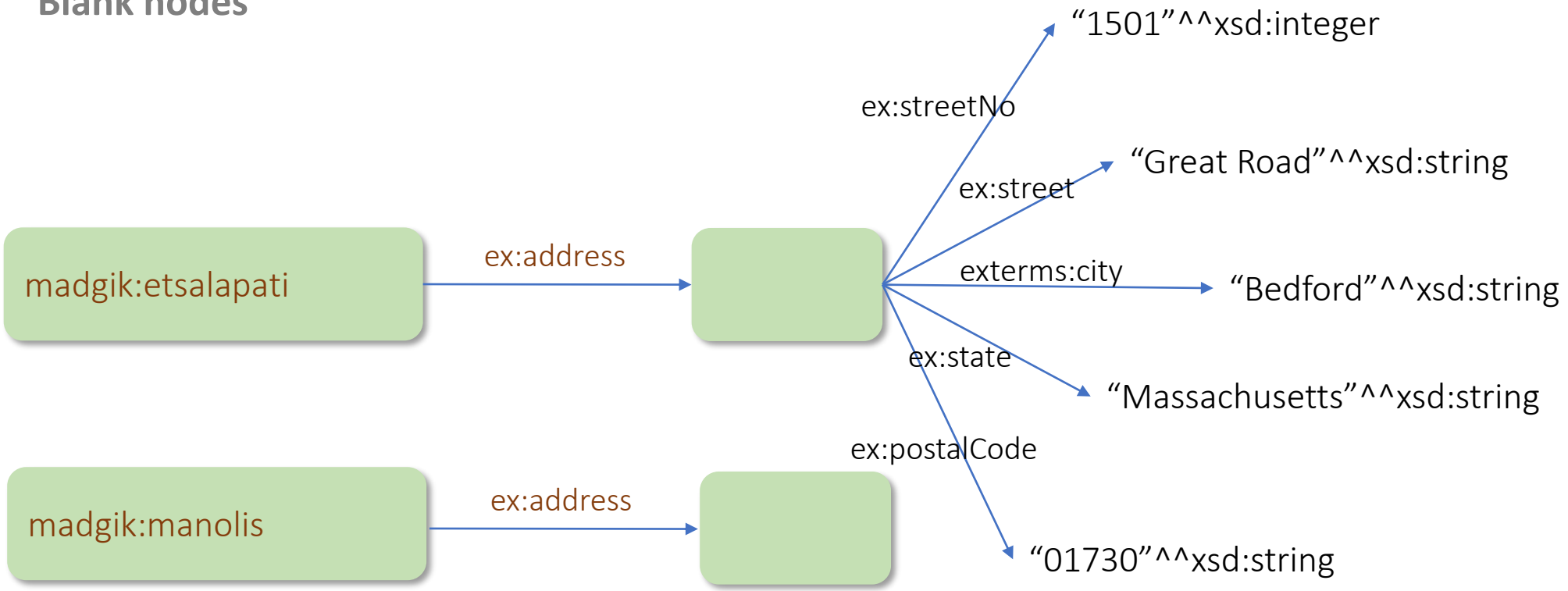
Return Eleni's postal code

- There can be three kinds of nodes in an RDF graph:
  - URIs/IRIs
  - RDF typed literals
  - **Blank nodes**

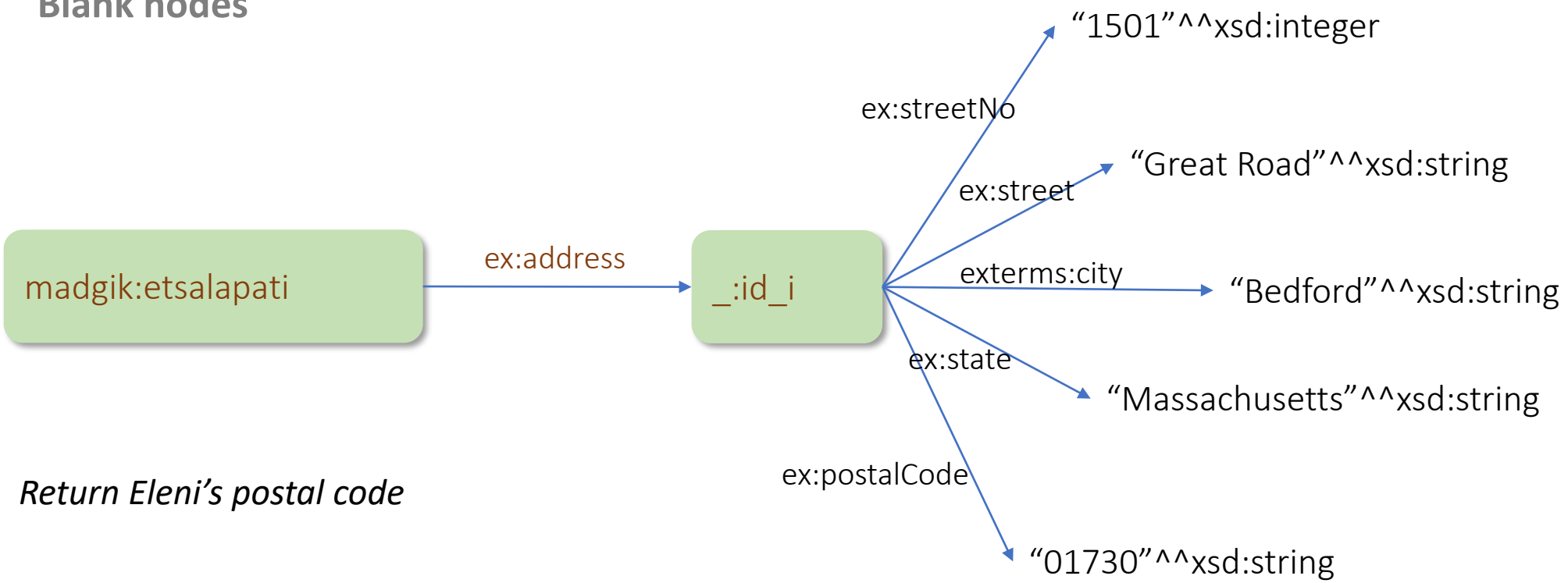


Return Eleni's postal code

- There can be three kinds of nodes in an RDF graph:
  - URIs/IRIs
  - RDF typed literals
  - **Blank nodes**



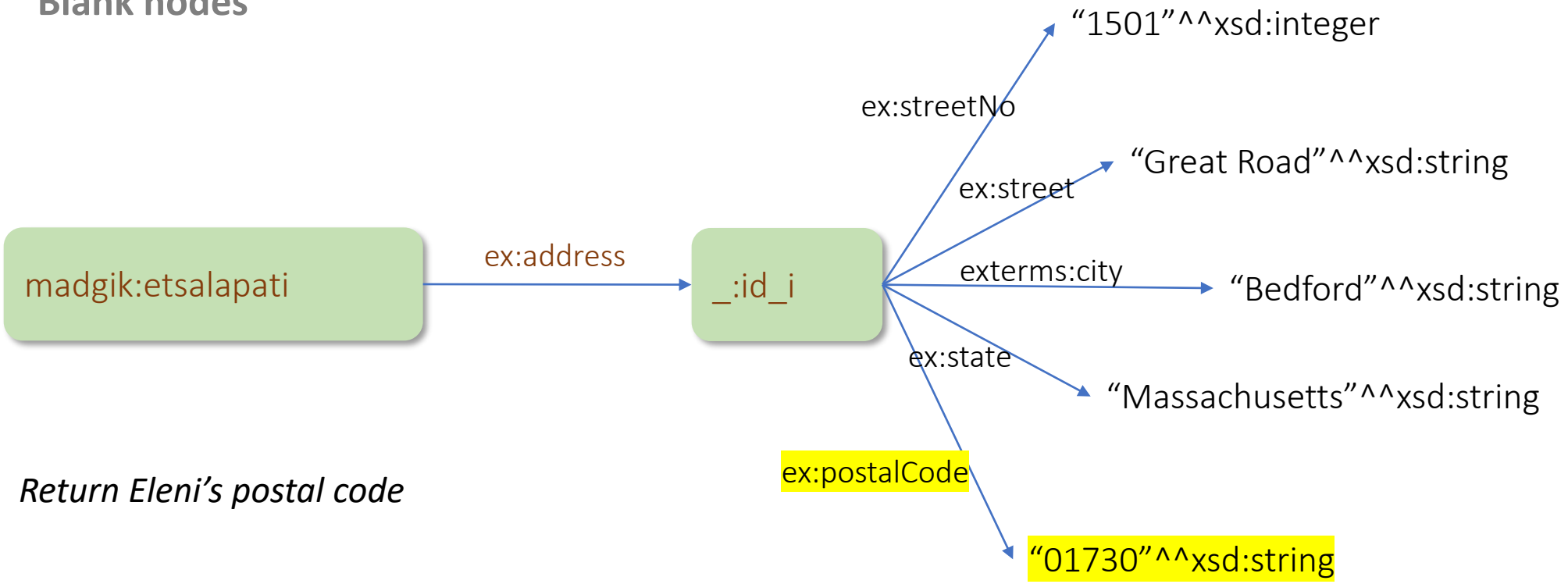
- There can be three kinds of nodes in an RDF graph:
  - URIs/IRIs
  - RDF typed literals
  - **Blank nodes**



Return Eleni's postal code

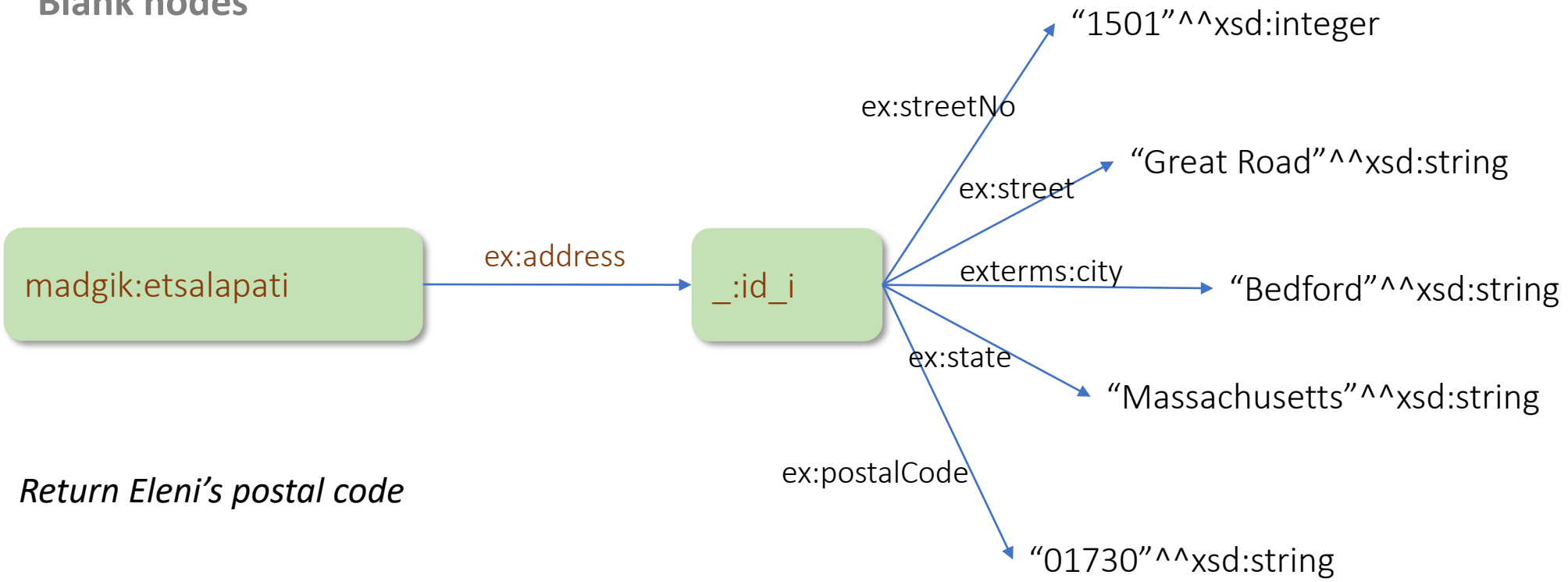
HOW WILL YOU REPRESENT THIS IN A TRIPLE FORM?

- There can be three kinds of nodes in an RDF graph:
  - URIs/IRIs
  - RDF typed literals
  - **Blank nodes**



Return Eleni's postal code

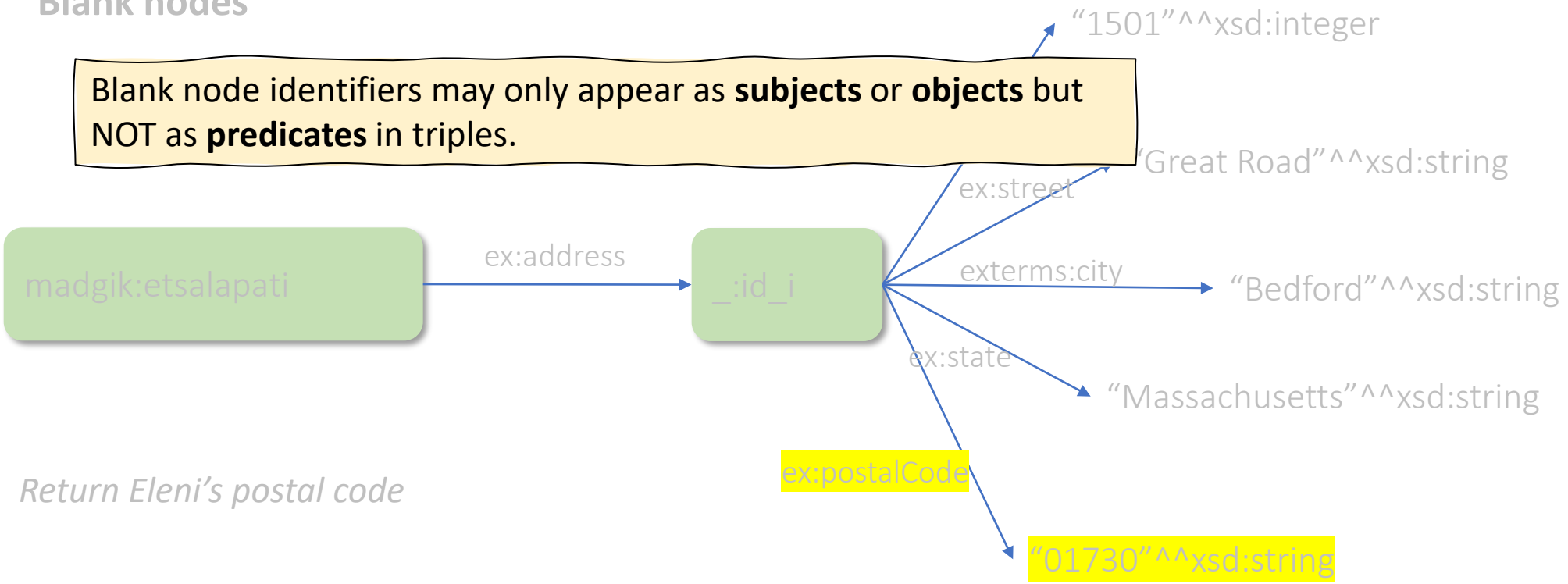
- There can be three kinds of nodes in an RDF graph:
  - URIs/IRIs
  - RDF typed literals
  - **Blank nodes**



Return Eleni's postal code

RDF Graph

- There can be three kinds of nodes in an RDF graph:
  - URIs/IRIs
  - RDF typed literals
  - **Blank nodes**



Return Eleni's postal code

Semantics of Blank Nodes

- In terms of **first-order logic**, a blank node corresponds to an **existentially quantified variable** (or a Skolem constant). Thus, a graph with blank nodes is similar to an **existentially quantified first order logic statement** or **a first-order database** in the sense of Reiter:

  Raymond Reiter: Towards a Logical Reconstruction of Relational Database Theory. Appears in the collection: Brodie M. L., Mylopoulos J., and Schmidt J. W. (eds.), On Conceptual Modelling: Perspectives from Artificial Intelligence, Database and Programming Languages, pp. 191-233, Springer-Verlag.

- In terms of the **relational model**, a blank node corresponds to a **marked null value**. Thus, a statement with a blank node identifier is similar to a **tuple in a relation** in the marked nulls model of Imielinski and Lipski:

  Tomasz Imielinski, Witold Lipski Jr.: Incomplete Information in Relational Databases. J. ACM 31(4): 761-791 (1984).

Semantics of Blank Nodes (cont'd)

- The RDF semantics (http://www.w3.org/TR/rdf-mt/) takes this "**existential view**" of blank nodes and defines appropriate semantics for them.

- SPARQL, the standard query language for RDF, considers blank nodes **as constants scoped in the graph where they appear.**

- In practice, it varies how people use blank when they publish linked data.

- For a nice study of the relevant issues see the paper
  Alejandro Mallea, Marcelo Arenas, Aidan Hogan, Axel Polleres.
  On Blank Nodes. Proceedings of the International Semantic Web Conference 2011, pages 421-437.

Blank Nodes (cont'd)

- Blank nodes are useful to represent **n-ary relationships** in RDF e.g., the relationship between John Smith and the street, city, state, and postal code components of his address.

- Blank nodes are also useful to more accurately make statements about **resources that may not have URIs,** but that are described in terms of relationships with other resources that do have URIs.

Example

- When making statements about a person, say Jane Smith, is it natural to use a URI based on that person's email address as her URI, e.g., mailto:jane@example.org ?

- Well, if we do so, how are we going to record information both about **Jane's mailbox** (e.g., the server it is on) as well as about **Jane herself** (e.g., her current physical address)? Similarly, if we use her Web page URI etc.

Example (cont'd)

- **Blank nodes to the rescue:** When Jane herself does not have a URI, a blank node provides a more accurate way of modeling this situation.

```
_:jane exterms:mailbox <mailto:jane@example.org>
  .
 _:jane rdf:type exterms:Person .
_:jane exterms:name "Jane Smith" .
_:jane exterms:empID "23748" .
_:jane exterms:age "26" .
```

The Property `rdf:value`

- When we have a structured value, RDF provides a way to define the **“main value”** of this structured value, with the other parts providing additional contextual or other information that qualifies the main value.

- **Example:**

```
exproduct:item10245 exterms:weight "2.4"^^xsd:decimal .
```

- In this case, it is better to have **2.4 kilograms** rather than just the decimal value 2.4.

Example (cont'd)

- Here we have a structured value (2.4 kilograms) with a **main value** (2.4) and another part (the unit of measure).

- RDF allows us to write:

```
exproduct:item10245 exterms:weight _:weight10245 .
_:weight10245 rdf:value "2.4"^^xsd:decimal .
_:weight10245 exterms:units exunits:kilograms .
```

- RDF does not have a special semantics for `rdf:value`; it simply offers it as a utility property.

Machine Readable Formats for RDF

RDF has a number of machine readable formats:

- XML/RDF
- Turtle (Terse RDF Triple Language)
- N3
- …

Anonymous Blank Nodes

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix dc: <http://purl.org/dc/elements/1.1/#>.
@prefix exterms: <http://www.example.org/terms/>.

<http://www.w3.org/TR/rdf-syntax-grammar>
    dc:title "RDF/XML Syntax Specification (Revised)";
    exterm:editor [
        exterms:fullName "Dave Beckett";
        exterms:homePage <http://purl.org/net/dajobe/>.
    ].
```

- **Notation:** This is reminiscent of notations for complex objects from the area of database systems.

RDF 1.1

- RDF, as we presented it, became a W3C standard in 2004. This is now referred as **RDF 1.0.**
- In 2014, **RDF1.1** was defined. It is a simple extension of RDF1.0.

What's New in RDF 1.1

- Identifiers are IRIs.
- **All literals have a data type.**
- Literals with a language tag have data type rdf:langString.
- The **concept of RDF dataset was defined.** An **RDF dataset** is a collection of RDF graphs. We will see again this concept when we discuss the query language SPARQL.
- Compatible XSD data types were specified.
- The datatype **rdf:HTML was introduced so that we can have objects of triples to be HTML code.**
- More serialization formats: RDF/XML, RDFa, N-Triples, N-Quads, Turtle, TriG and JSON-LD.

- You can see more details in https://www.w3.org/TR/rdf11-new/ .

RDF Serialization

RDF has a number of machine-readable formats:

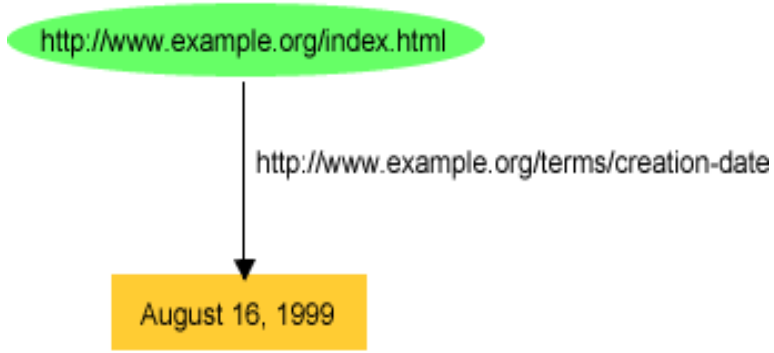
- XML/RDF
- Turtle (Terse RDF Triple Language)
- N3
- …

An XML Syntax for RDF: RDF/XML

- The conceptual model for RDF is a graph.
- RDF provides an XML syntax for **writing down and exchanging RDF graphs**, called **RDF/XML**.
- RDF/XML is **the normative syntax** for writing RDF.

Example

http://www.example.org/index.html has a creation-date whose value is August 16, 1999.



Example in XML/RDF

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
           xmlns:exterms="http://www.example.org/terms/">
     <rdf:Description rdf:about="http://www.example.org/index.html">
          <exterms:creation-date>August 16, 1999</exterms:creation-date>
     </rdf:Description>
</rdf:RDF>
```

Blank Nodes in Turtle

"RDF/XML Syntax Specification (Revised)“ has editor Dave Beckett,
who has homepage: http://purl.org/net/dajobe/

```
<http://www.w3.org/TR/rdf-syntax-grammar>
        dc:title "RDF/XML Syntax Specification (Revised)“^^xsd:string;
        exterm:editor _:abc.

 _:abc
        exterms:fullName "Dave Beckett";
        exterms:homePage <http://purl.org/net/dajobe/>.
```

```
<http://www.w3.org/TR/rdf-syntax-grammar>
        dc:title "RDF/XML Syntax Specification (Revised)";
        exterm:editor [
                exterms:fullName "Dave Beckett";
                exterms:homePage <http://purl.org/net/dajobe/>.
        ].
```

Summary

- In RDF, we identify things with **URIs**
- Statements about a domain are encoded by **triples:**

subject predicate object .

- The **subject** of a triple must be a **URI** or a **blank node**.
- The **predicate** of a triple must be a **URI**.
- The **object** of a triple must be a **URI**, a **literal** or a **blank node**.

What can we do with RDF

- Adding machine-readable information using **well-known vocabularies**, e.g. schema.org:
  - Ambiguity is eliminated
  - Shareability is established
- **Enriching** dataset by linking to external datasets.
  - e.g. linking paintings dataset to artists dataset
- Building **aggregations** of data about specific topics
  - e.g., distributed social networks by linking RDF descriptions of people across multiple Web sites
  - e.g., Interlinking various datasets within an organization: cross-dataset QA
- Providing **standard-compliant** way for exchanging data between DBs

Readings

- Chapter 2 of the book “Foundations of Semantic Web Technologies” or Chapters 2 and 3 of the Semantic Web Primer available from http://www.csd.uoc.gr/~hy566/ .

- The following material from the Semantic Web Activity Web page on RDF http://www.w3.org/RDF/ :
  - RDF Primer (https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140225/RDF Primer (https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140225/ and https://www.w3.org/TR/2004/REC-rdf-primer-20040210/)
  - Resource Description Framework (RDF): Concepts and Abstract Syntax (https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/)
  - The Turtle language (https://www.w3.org/TR/2014/REC-turtle-20140225/)

- Check out the content published at the RDF namespace URI:
  - http://www.w3.org/1999/02/22-rdf-syntax-ns#

  where you will find an RDF Schema description of the RDF vocabulary given in RDF/XML!

- The DBpedia project (http://dbpedia.org/AboutThe DBpedia project (http://dbpedia.org/About), a nice application of RDF and Linked Data (http://linkeddata.org/).

Exercise I

Create an RDF graph, using DBPedia, from the following sentence:

"The music genre of Chet Baker was cool jazz and west coast jazz. Both genres have stylistic origin bepop"

You will need the following URLs:

dbr:Chet_Baker

dbo:genre

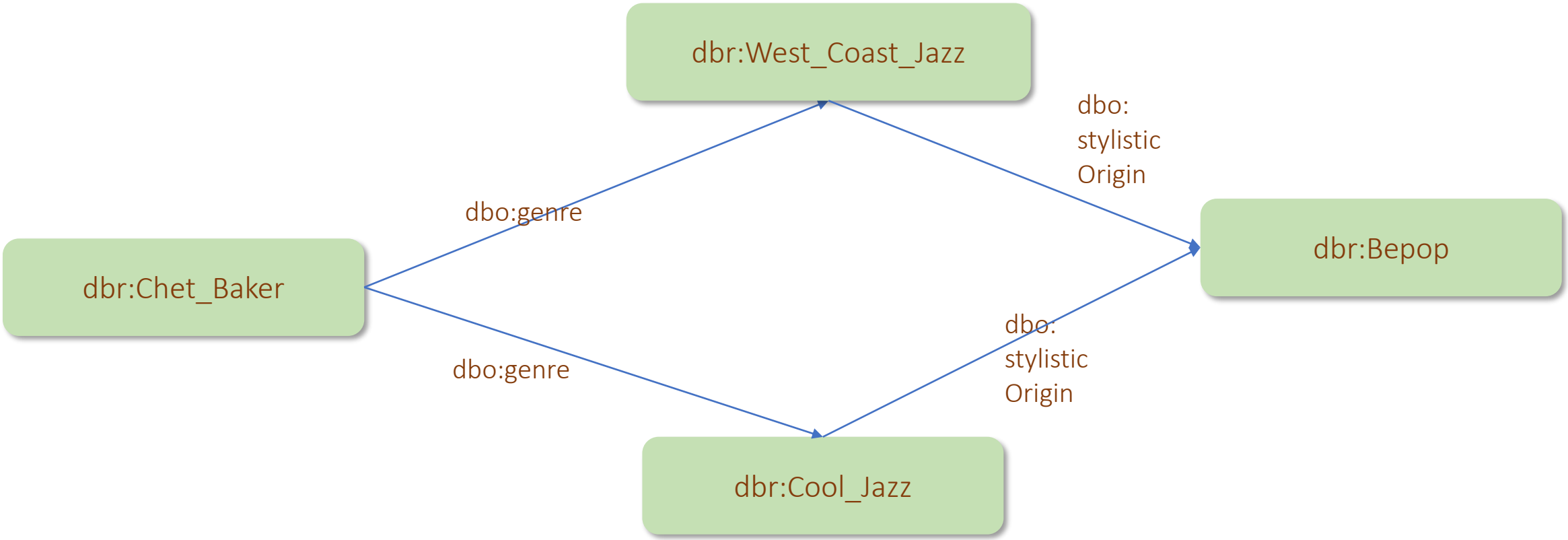
dbr:Cool_Jazz

dbr:West_Coast_Jazz

dbo:stylisticOrigin

dbr:Bepop

Solution


dbr:West_Coast_Jazz
dbo:
stylistic
Origin
dbo:genre
dbr:Chet_Baker
dbr:Bepop
dbo:genre
dbo:
stylistic
Origin
dbr:Cool_Jazz