

SWIB 2012: Workshop on Metadata Provenance

Part 2: Provenance of RDF data

Agenda

Linked Metadata (use the principles)

Named Graphs

RDF 1.1

OAI-ORE

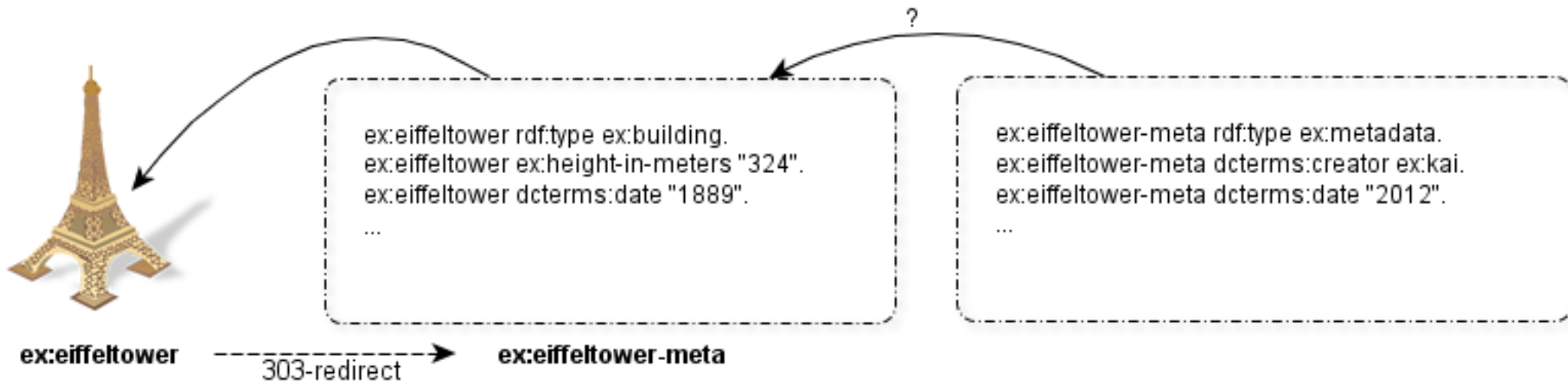
Linked Metadata

Linked Data Principles

- 1) Use URIs as names for things.
- 2) Use HTTP URIs so that people can look up those names.
- 3) When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL).
- 4) Include links to other URIs, so that they can discover more things.

<http://www.w3.org/DesignIssues/LinkedData.html>

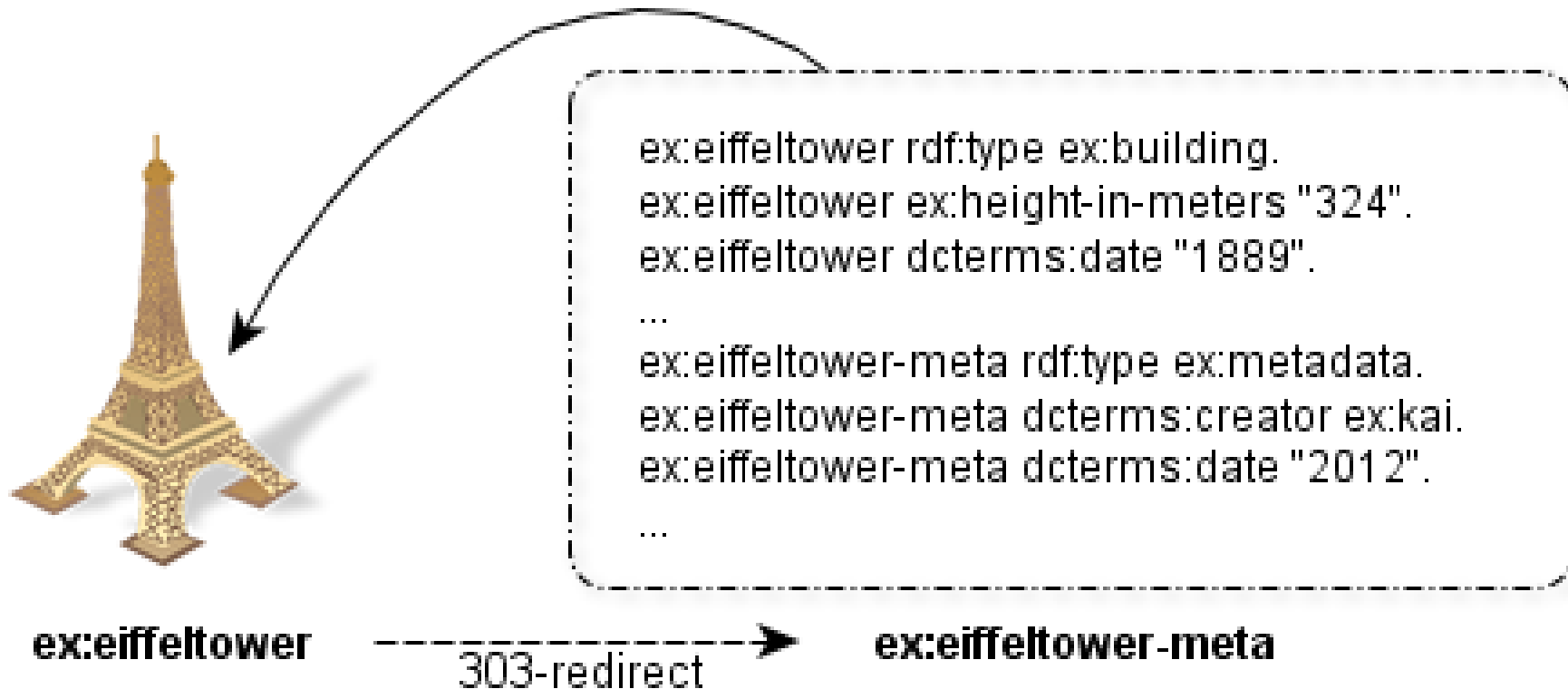
Linked Metadata



How do we get the metadata provenance?

Usual best practice: deliver it with the metadata.

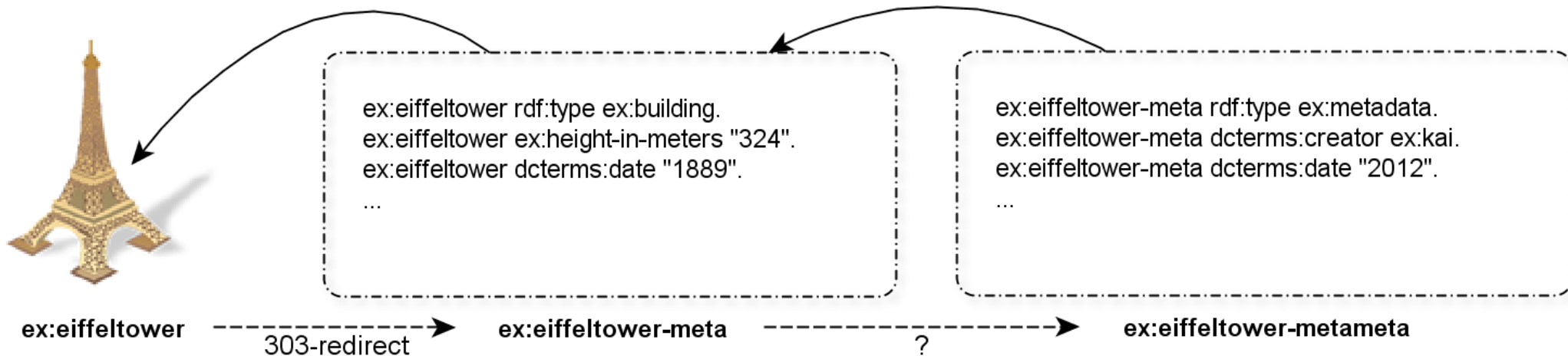
Embedded Linked Metadata (Method 1)



Drawback:

What about the provenance of the provenance?
There is no URI for the metadata provenance.

Linked Metadata



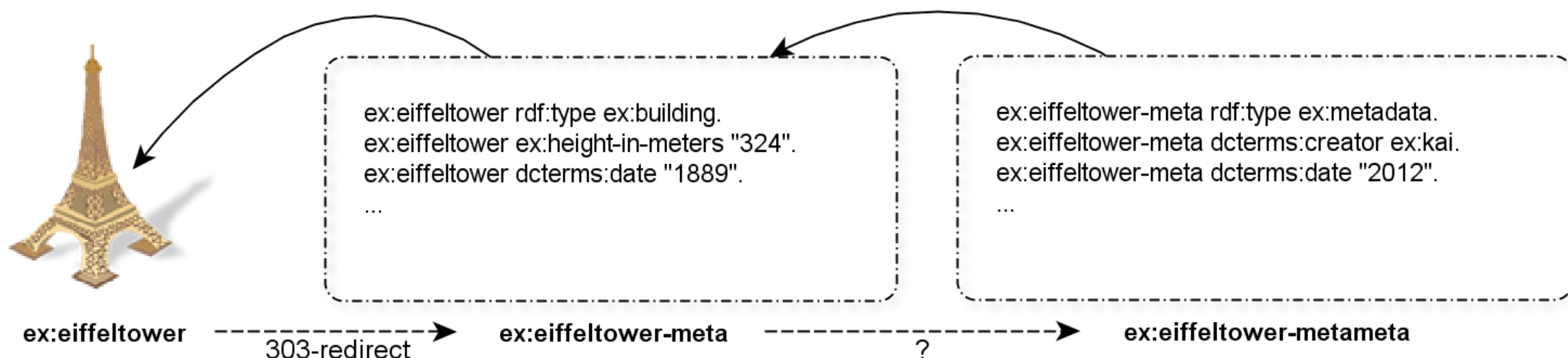
Then we give the metadata provenance a URI!

Problem: How to tell that we want the provenance.

Content negotiation is not working any more, as both contents are RDF.

Missing: A **request header** that asks for provenance.

The Link Header (Method 2)

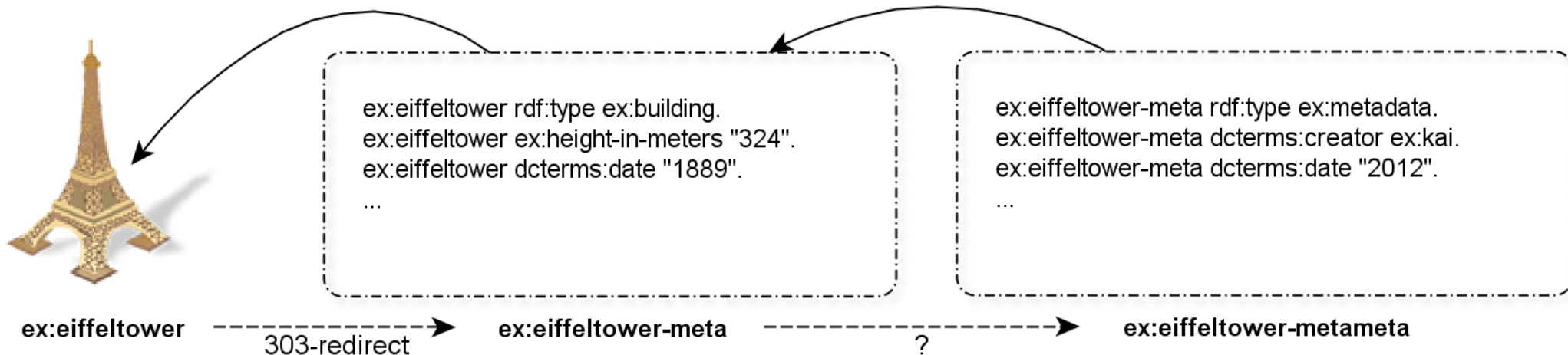


Response header sent by `ex:eiffeltower-meta`:

Link: <<http://example.org/eiffeltower-metameta>>; rel=meta

Drawback: Additional (head) request needed.

Additional Statements (Method 3)



Provide a **reference** to the provenance data:

For example:

```
ex:eiffeltower-meta rdfs:seeAlso ex:eiffeltower-metameta.
```

Drawback: `rdfs:seeAlso` very general. There is no commonly accepted property for provenance.

Linked Metadata Summary

- + Based on Linked Data Principles.
- + Current "best practice."
- Not suitable for provenance on statement level.
- Requires full control over web server.
- No URI for provenance information, **or**
- no accepted provenance retrieval mechanism.

Despite the drawbacks: a good starting point, as every provenance mechanism has to fit with the linked data principles.

Named Graphs

Named Graphs

A Named Graph is an RDF graph with an assigned URI as name.

Serialization in TriG:

```
ex:eiffeltower-meta {  
    ex:eiffeltower rdf:type ex:building.  
    ex:eiffeltower ex:height-in-meters "324".  
    ex:eiffeltower dcterms:date "1889".  
    ...  
}
```

Named Graphs are not (yet) part of the RDF standard, but they are supported in SPARQL.

Named Graphs in RDF Stores

RDF-Stores today are usually **quad-stores**.
(not triple-stores, even if we call them that way)

Each triple is assigned to a graph via the **fourth** quad element.

If the fourth element contains a URI, the URI is interpreted as the **name of the graph** that contains all triples with the same graph URI.

Named Graphs and SPARQL

SPARQL supports Named Graphs:

```
SELECT ?origin ?p ?o WHERE {  
  GRAPH ?origin {  
    :MonaLisa dc:creator :LeonardoDaVinci .  
  }  
  ?origin ?p ?o .  
}
```

This retrieves all statements about graph URIs containing a certain statement (e.g., provenance).

Named Graphs and Linked Data

A client that fetches linked data via a URI **usually** stores this URI as graph URI in a quad store.

This is **great**, because this way we can talk about the fetched RDF data and store provenance in our RDF store.

This is **only half way there**, because we can not reexpose the provenance information easily.

Because it is not part of RDF.

RDF 1.1

RDF WG

Mission:

Update the 2004 RDF Recommendations, extending RDF to include features desirable and important for interoperability, but without a negative effect on deployment.

Required Feature (Charter) among others:

Support for **Multiple Graphs** and Graph Stores.

Standardize the Turtle RDF Syntax. Either that syntax or a related syntax should also support **multiple graphs**.

=> Multiple Graph is a neutral term for: something like named graphs. Yeah! :-)

<http://www.w3.org/2011/01/rdf-wg-charter>

Named Graphs in RDF 1.1 (Work in Progress!)

From RDF 1.1 Concepts and Abstract Syntax
(Editor's Draft, June 05, 2012):

An RDF Dataset is a collection of RDF graphs and comprises [...] zero or more named graphs.

Each named graph is a pair consisting of an IRI (the **graph name**), and an **RDF graph**.

Note:

The graph name does **not** formally denote the graph.

RDF does **not** place any formal **restrictions** on **what resource the graph name may denote**, nor on the **relationship between that resource and the graph**.

RDF Graphs

What is an RDF Graph?

An RDF graph is a set of RDF triples.

That means that a (named) RDF Graph does not contain other (named) graphs.

Consequences:

You can reexpose graphs with names (e.g., with TriG),
but: no directions how to interpret the graph URI,
and: when the TriG file is fetched, no possibility to store the graphs inside another graph with the URI of the TriG file.

Preliminary RDF 1.1 Summary

Half way there,
but still enough room
for own decisions and
developments.

Positive thinking ;-)

OAI-ORE

OAI-ORE

Open Archives Initiative - Object Reuse and Exchange

Originally addresses another problem that lacks a solution in RDF:

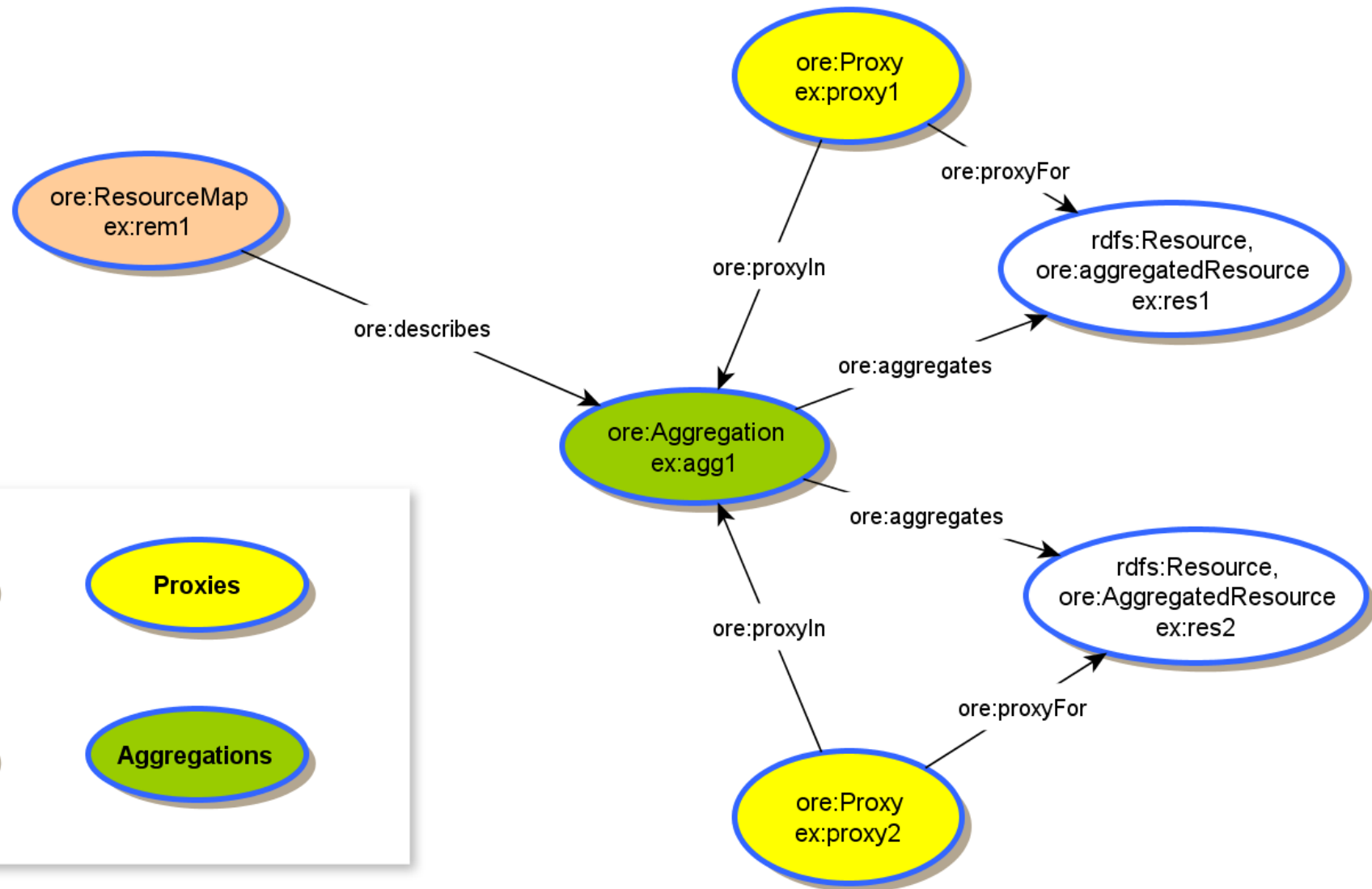
How to make a statement about a resource that is only valid in a special context?

Example: The ordering of resources in an aggregation, like the ordering of articles in a bibliography.

Adaption for provenance:

All statements are provided within such a context, the context can be identified and further described by provenance statements.

OAI-ORE Graph



OAI-ORE and Linked Data

The Resource Map is just a web resource with an own URI.

The Resource Map is connected to the Aggregation via ore:describes.

The Aggregation and the Proxies provide the scaffolding for the statements that are made in the context of the Aggregation.

Drawback: An application has to be “ore-aware” to make sense of all this, as the concept of a proxy resource is not known in RDF.

Summary

Many different approaches:

- 1) ~~Reification~~
- 2) "Simple" application of Linked Data principles.
- 3) Named Graphs
- 4) OAI-ORE
- 5) Own models and extensions

In practice, we have to combine them to create flexible solutions.

Unfortunately, the full understandability of linked data provenance is not (yet) guaranteed.