

# The MediaSpaces Mapping Framework

Bernhard Haslhofer ■ [bernhard.haslhofer@univie.ac.at](mailto:bernhard.haslhofer@univie.ac.at)

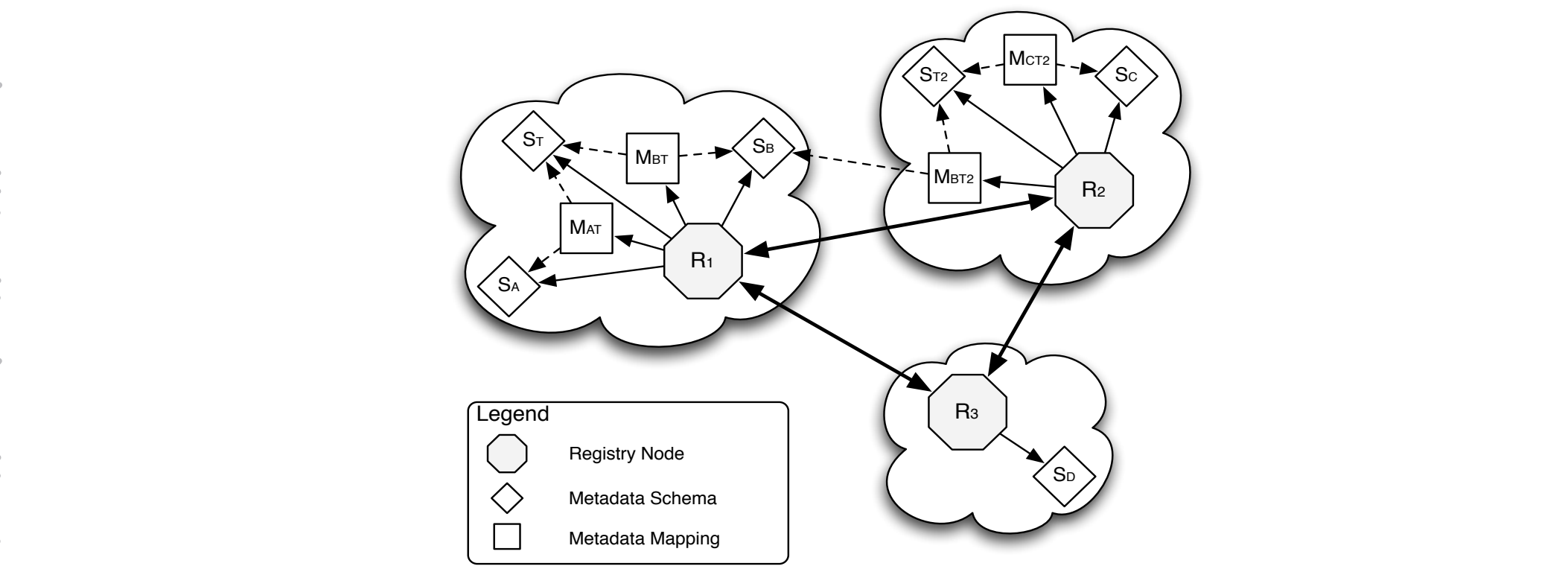
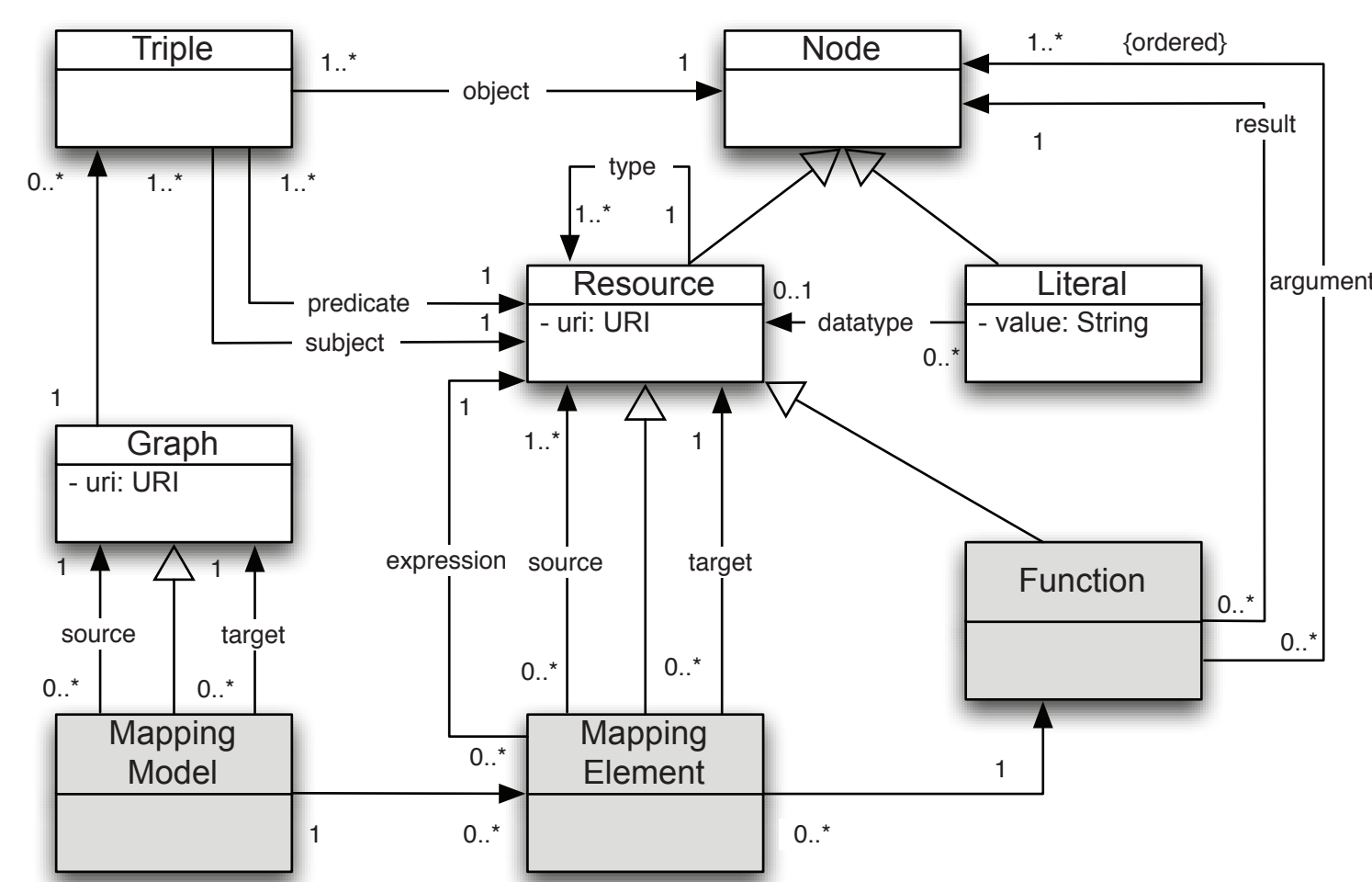
Department of Distributed and Multimedia Systems  
Multimedia Information Systems Group

## Mapping Maintenance

As a result of the Linked Data initiative, many institutions expose RDF data on the Web and provide a SPARQL endpoint that allows clients to execute structured queries against a data source. If, however, a client needs to access several autonomous data sources with a single query, it must first deal with the **heterogeneities** caused by the **different vocabularies/ontologies/schemes** in use.

The MediaSpaces Mapping API allows for the **declaration of mapping relationships** between incompatible vocabularies on the Web. It includes an extensible function framework to bridge heterogeneities also on the **instance level**. Mapping specifications are compiled into executable SPARQL query templates, which allows developer to set up executable **SPARQL mediation endpoints** in order to integrate metadata from different heterogeneous SPARQL data sources at run-time.

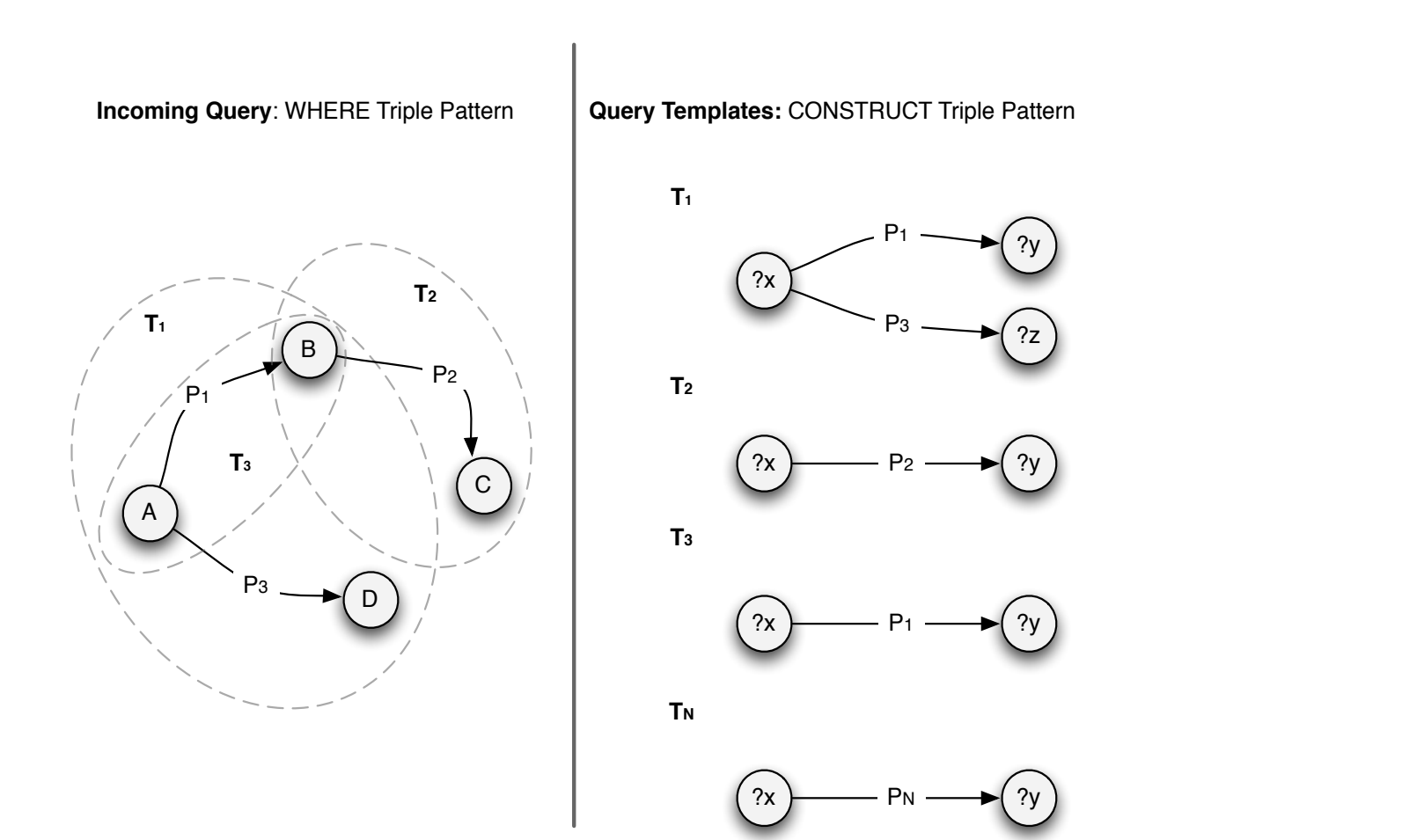
The underlying **Abstract Mapping Model** defines the main mapping concepts, independent of any particular language. It is based on a simple graph model.



```
reg:r1 a rm:RegistryModel;
rdfs:label "Institution X Metadata Registry";
rm:linkedRegistry <http://www.institutionY.com/registry/r2>;
rm:registeredMapping
<http://www.institution.com/mapping/onb_dc>;
.

http://www.institution.com/mapping/onb_dc am:MappingModel;
rdfs:label
"Mapping between the ONB Metadata Format and the Dublin
Core Element Set";
am:source <http://www.bildarchiv.com/schema/onb>;
am:target <http://purl.org/dc/elements/1.1/>;
.
```

```
SELECT
{
  ?x dc:creator ?y.
}
WHERE
{
  ?x onb:firstName ?a.
  ?x onb:lastName ?b.
  ?y fn:concat (?b " ", " ?a).
  ?x rdfs:type onb:Person.
}
```



## Mapping Execution

## Mapping Discovery

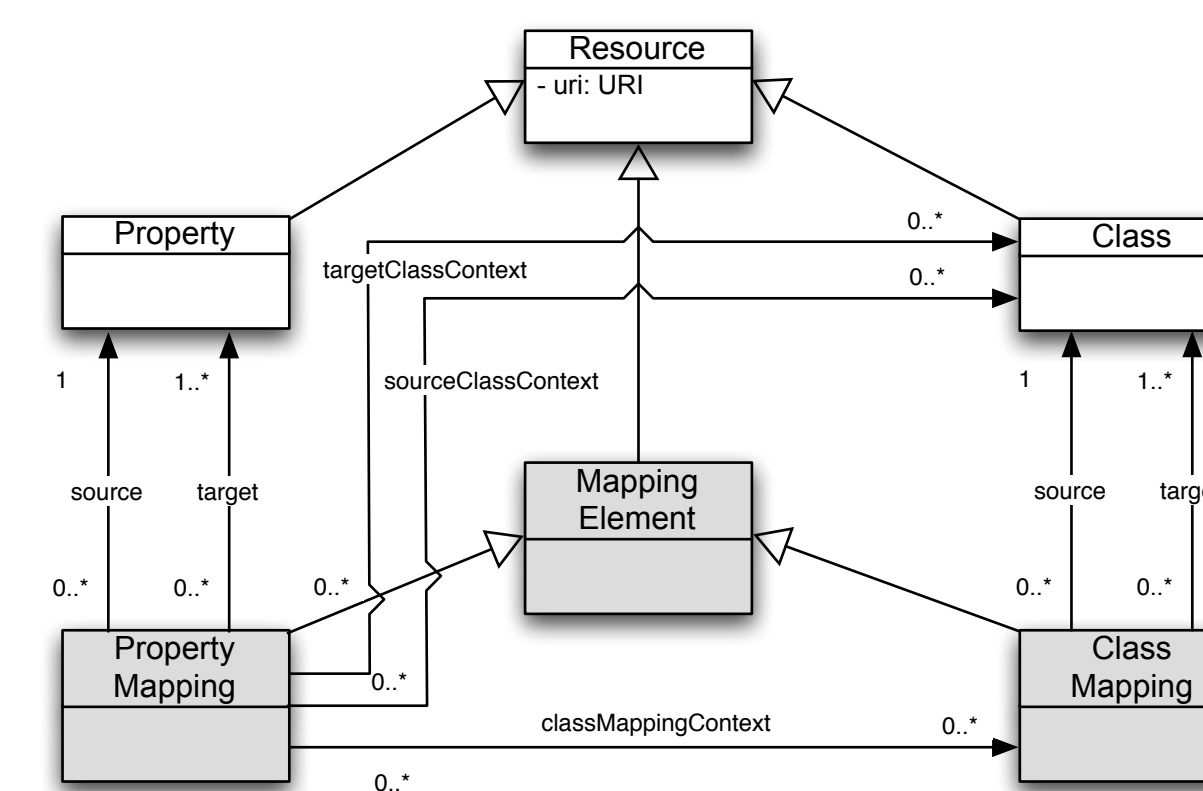
```
MappingModel
+ findMappings(): MappingElement[]
+ executeMapping(Graph graph): Graph
+ registerMapping(MappingRegistry registry)
```

For the discovery of mapping relationships we rely on **third-party schema/ontology matching algorithms**. They can be integrated into the MediaSpaces Mapping Framework via plug-in connectors.

Mapping discovery also includes the **reuse of existing metadata mappings** that are available in some Web accessible location (see Mapping Maintenance).

```
map:fnln2creator a mm:PropertyMapping;
am:expression am:targetInclude;
mm:sourceClassContext onb:Person;
am:sourceElement onb:firstName;
am:sourceElement onb:lastName;
am:targetElement dc:creator;
am:transFunction map:fnlnConcat;
.

map:fnlnConcat a am:Function;
am:URI fn:concat;
am:argument (onb:lastname, " ", onb:firstname);
am:result dc:creator;
.
```

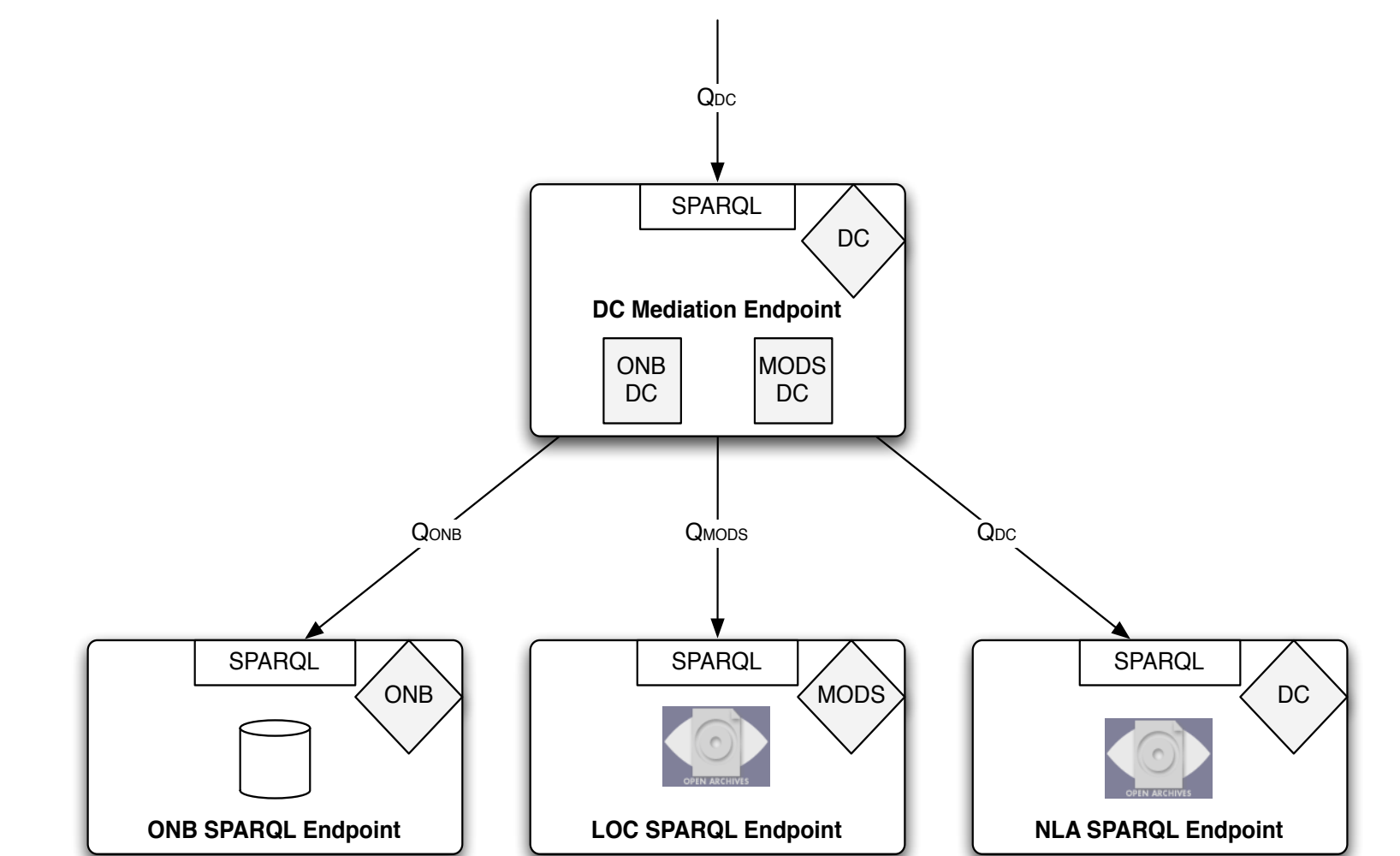


## Mapping Representation

The MediaSpaces Mapping Framework follows the **Linked Data principles**.

- Mapping elements
- \* use URIs as identifiers
  - \* are exposed as de-referencable HTTP resources
  - \* are presented in a human- and machine readable way
  - \* contain links to the schema elements they map

The Mapping API is an **extension of the Jena Semantic Web Framework** and freely available. For demonstration purposes we have implemented a **mediator component** that uses the Mapping API and provides uniform query access to a set of distributed, heterogeneous data sources via a single query interface.



Further Information:

All software components are documented and available at <http://www.mediaspaces.info/tools/mapping>

Haslhofer, B.: A Web-based Mapping Technique for Establishing Metadata Interoperability. PhD Thesis, University of Vienna (2008)