

Data mediation in SPARQL from alignments

Jérôme Euzenat



October 3, 2013

Jérôme Euzenat
Pavel Shvaiko

Ontology Matching

Second Edition

 Springer

- ▶ Affiliated with INRIA and LIG, Grenoble
- ▶ **Semantic web – linked data**
- ▶ Specialty: Ontology matching
- ▶ Logic, semantics, similarities, others
- ▶ <http://exmo.inria.fr>

Representing declaratively
relations between heterogeneous models
(ontology alignments)

is a valuable way to perform mediation
through SPARQL

Table of content

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

Mediating RDF data and alignments

Alignment API and mediators

Data interlinking example

RDB2RDF

Conclusions

Table of Contents

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

Mediating RDF data and alignments

Alignment API and mediators

Data interlinking example

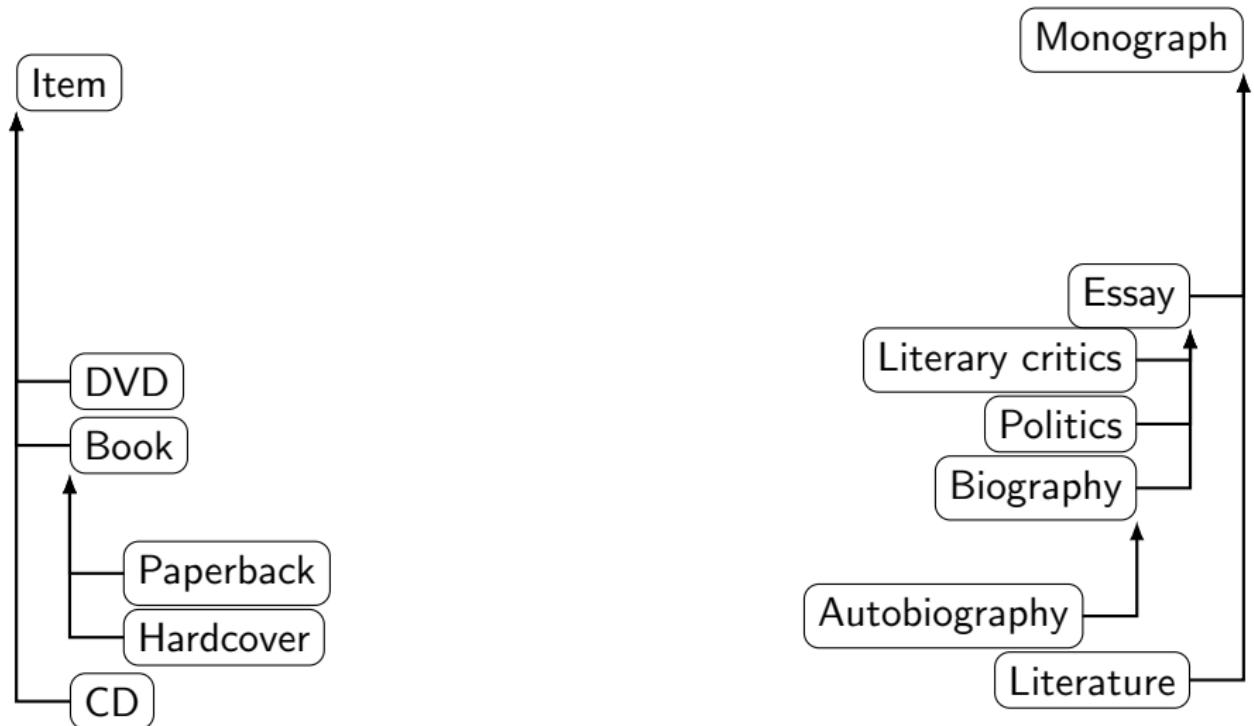
RDB2RDF

Conclusions

- ▶ Mediation is necessary because of heterogeneity
- ▶ We want to reduce heterogeneity with alignments
- ▶ They are declarative expression of correspondences between resources
- ▶ Which can be interpreted for generating mediators

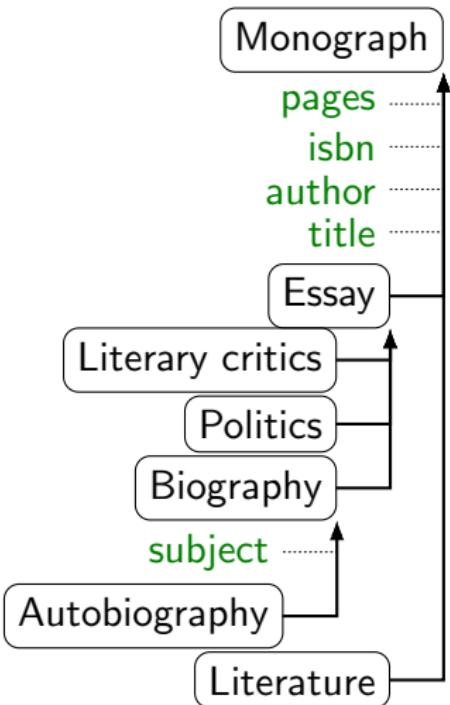
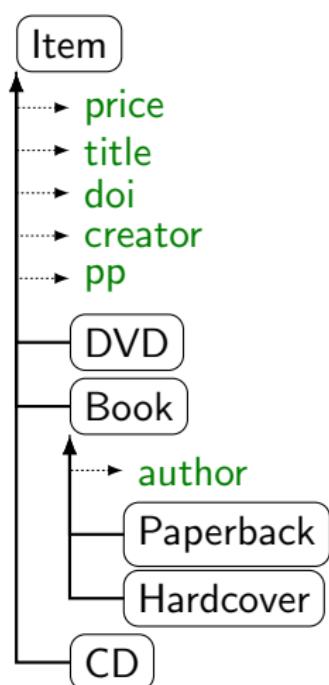
Ontology heterogeneity

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

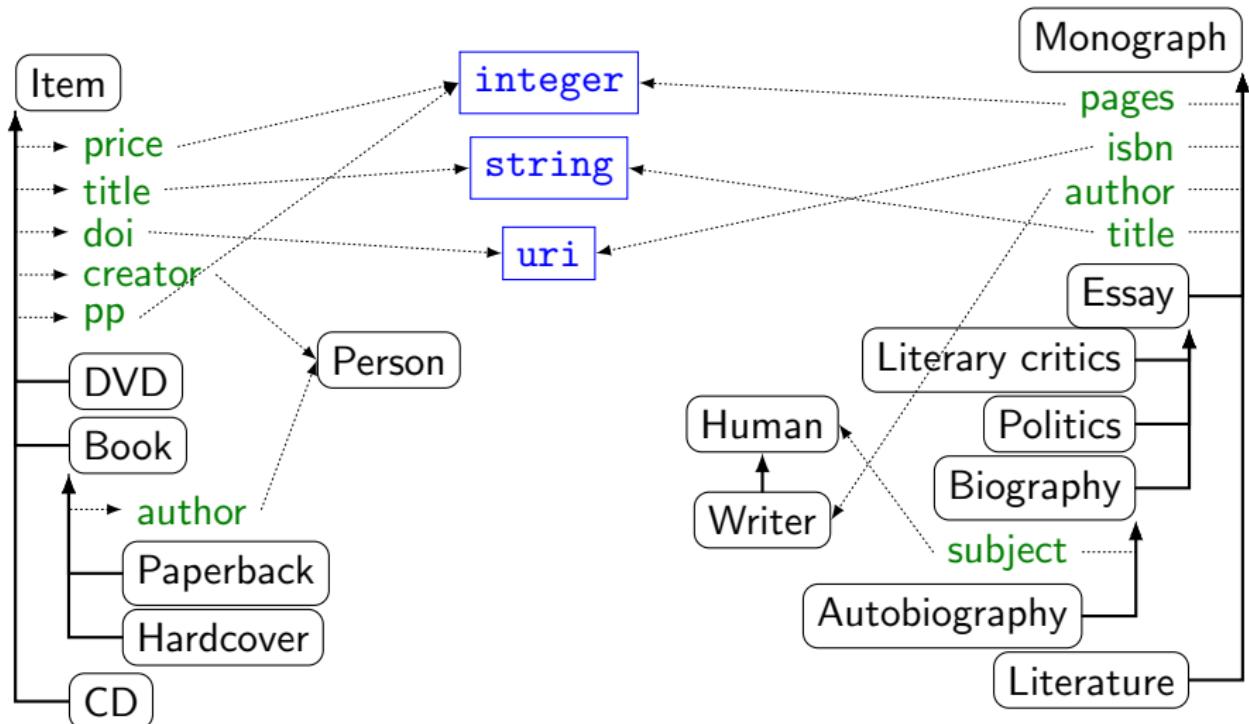


Ontology heterogeneity

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



Ontology heterogeneity



Resources being expressed in different ways must be reconciled before being used.

Mismatch between formalized knowledge can occur when:

- ▶ different languages are used (OWL vs. Topic maps);

Resources being expressed in different ways must be reconciled before being used.

Mismatch between formalized knowledge can occur when:

- ▶ different languages are used (OWL vs. Topic maps);
- ▶ **different terminologies are used:**
 - ▶ English vs. Chinese;
 - ▶ Book vs. Monograph.

Resources being expressed in different ways must be reconciled before being used.

Mismatch between formalized knowledge can occur when:

- ▶ different languages are used (OWL vs. Topic maps);
- ▶ **different terminologies are used:**
 - ▶ English vs. Chinese;
 - ▶ Book vs. Monograph.
- ▶ **different models are used:**
 - ▶ different classes: Autobiography vs. Paperback;
 - ▶ classes vs. property: Essay vs. literarygenre;
 - ▶ classes vs. instances: One physical book as an instance vs. one work as an instance.

Resources being expressed in different ways must be reconciled before being used.

Mismatch between formalized knowledge can occur when:

- ▶ different languages are used (OWL vs. Topic maps);
- ▶ **different terminologies are used:**
 - ▶ English vs. Chinese;
 - ▶ Book vs. Monograph.
- ▶ **different models are used:**
 - ▶ different classes: Autobiography vs. Paperback;
 - ▶ classes vs. property: Essay vs. literarygenre;
 - ▶ classes vs. instances: One physical book as an instance vs. one work as an instance.
- ▶ **different scopes and granularity are used.**
 - ▶ Only books vs. cultural items vs. any product;
 - ▶ Books detailed to the print and translation level vs. books as works.

How can we address the problem?

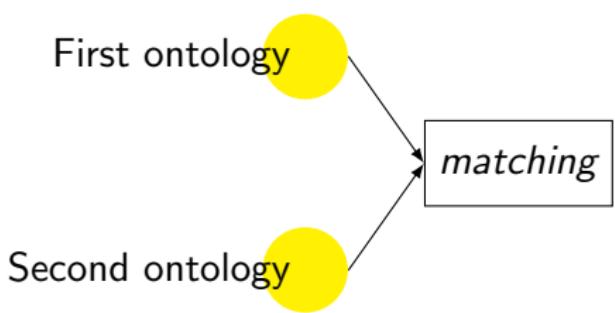
Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

First ontology

Second ontology

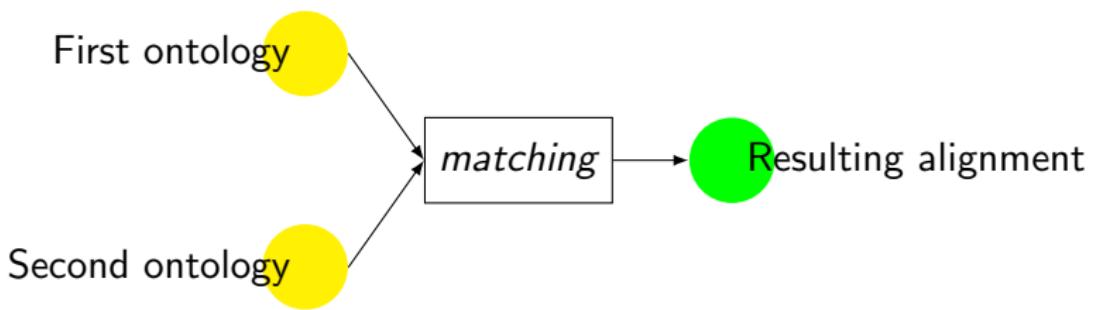
How can we address the problem?

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



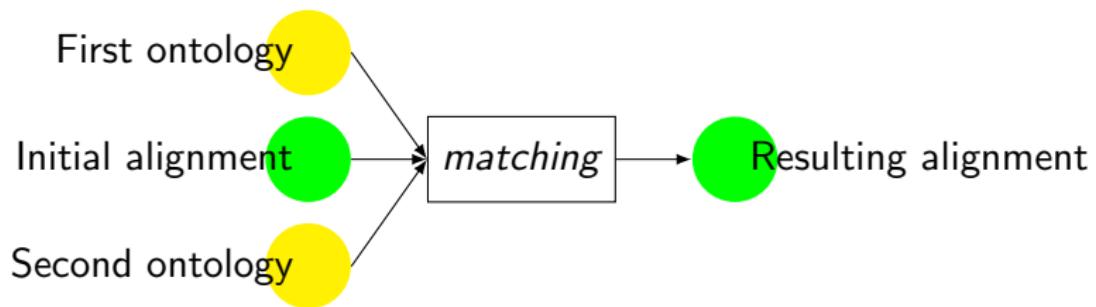
How can we address the problem?

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



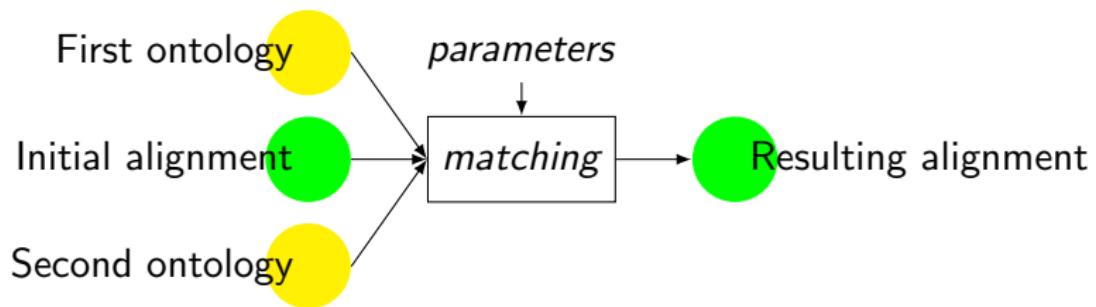
How can we address the problem?

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



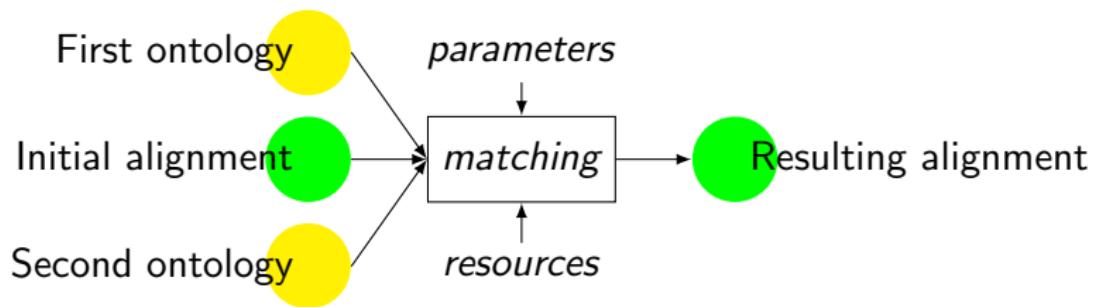
How can we address the problem?

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

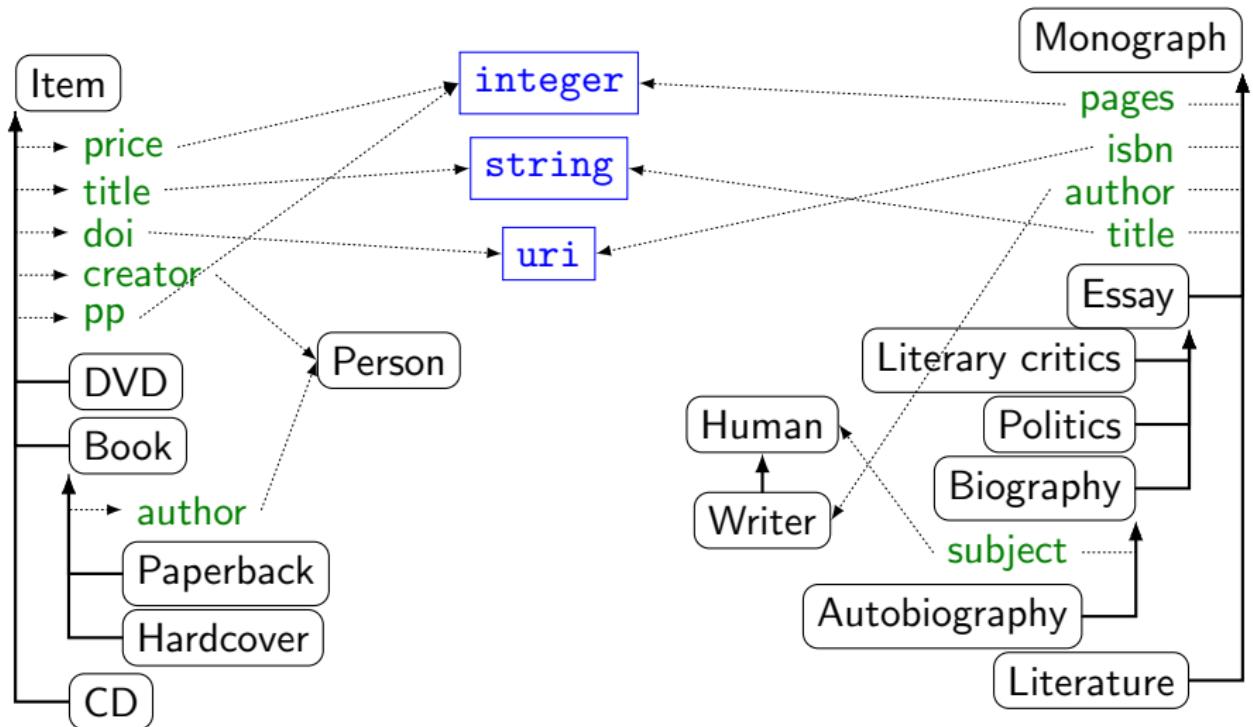


How can we address the problem?

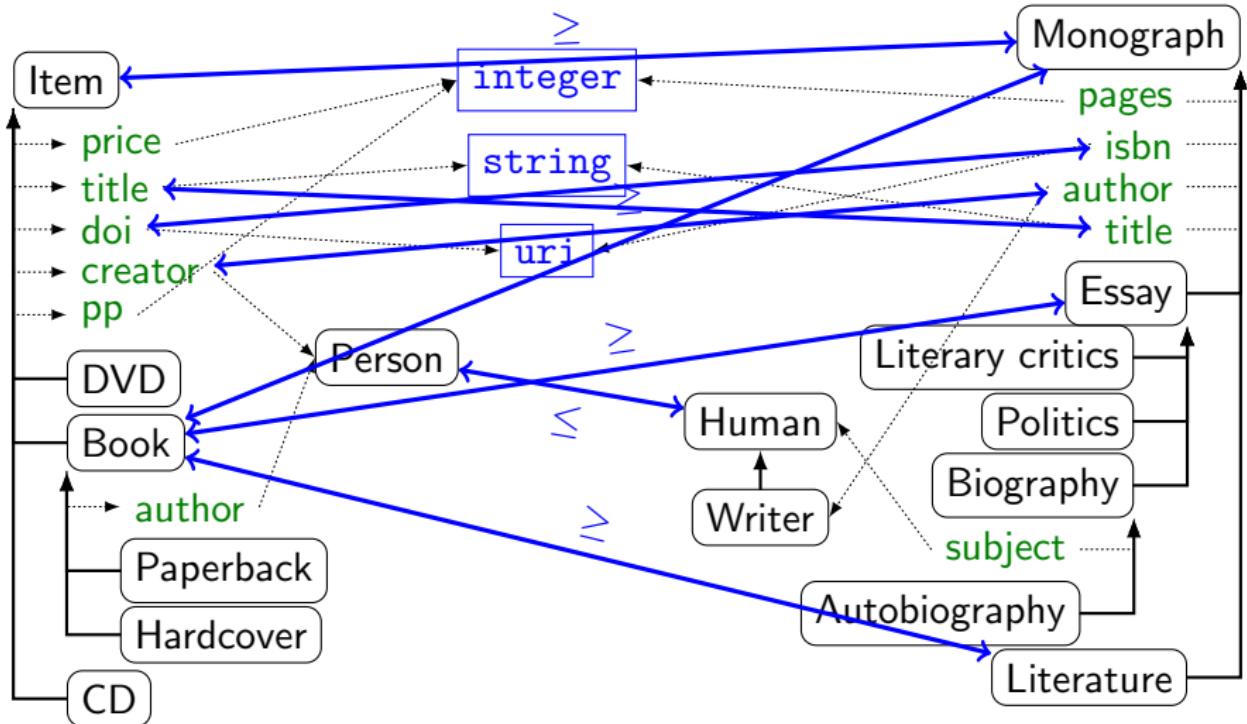
Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



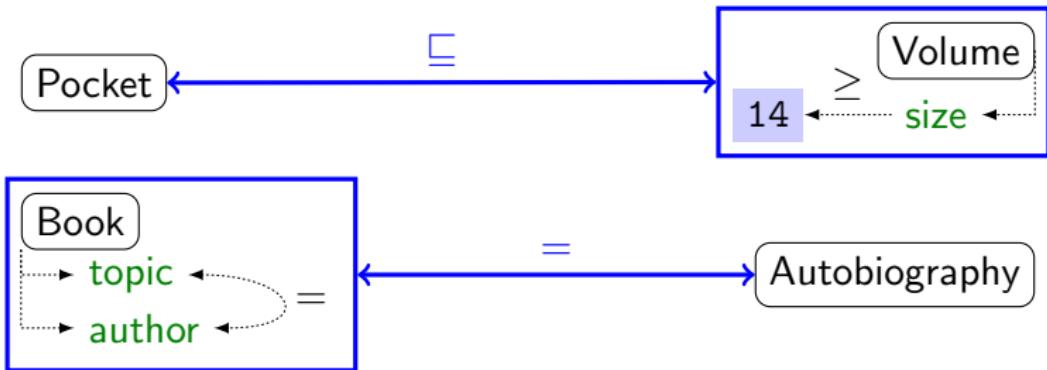
Ontology alignment



Ontology alignment



Expressive alignments (EDOAL)


$$\forall x, Pocket(x) \Leftarrow Volume(x) \wedge size(x, y) \wedge y \leq 14$$
$$\forall x, Book(x) \wedge author(x, y) \wedge topic(x, y) \equiv Autobiography(x)$$

```
SELECT ?i  
WHERE { ?x rdf:type o':Autobiography .  
        ?x o':author/o':name "Bertrand Russell" .  
        ?x o':isbn ?i .}
```

```
SELECT ?i  
WHERE { ?x rdf:type o':Autobiography .  
?x o':author/o':name "Bertrand Russell" .  
?x o':isbn ?i .}
```

mediator

Transformation and mediation

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

```
SELECT ?d
WHERE {?x rdf:type o:Book .
      ?x o:creator ?y .
      ?x o:topic ?y .
      ?y o:name "Bertrand Russell" .
      ?x o:doi ?d .}
```

```
SELECT ?i
WHERE {?x rdf:type o':Autobiography .
      ?x o':author/o':name "Bertrand Russell" .
      ?x o':isbn ?i .}
```

mediator

Transformation and mediation

```
SELECT ?d
WHERE {?x rdf:type o:Book .
      ?x o:creator ?y .
      ?x o:topic ?y .
      ?y o:name "Bertrand Russell" .
      ?x o:doi ?d .}
```

```
SELECT ?i
WHERE {?x rdf:type o':Autobiography .
      ?x o':author/o':name "Bertrand Russell" .
      ?x o':isbn ?i .}
```

mediator

x.doi=http://dx.doi.org/10.1080/041522862X

Transformation and mediation

```
SELECT ?d
WHERE {?x rdf:type o:Book .
      ?x o:creator ?y .
      ?x o:topic ?y .
      ?y o:name "Bertrand Russell" .
      ?x o:doi ?d .}
```

```
SELECT ?i
WHERE {?x rdf:type o:Autobiography .
      ?x o:author/o':name "Bertrand Russell" .
      ?x o:isbn ?i .}
```

mediator

x.doi=http://dx.doi.org/10.1080/041522862X

x.isbn=041522862X

Table of Contents

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

Mediating RDF data and alignments

Alignment API and mediators

Data interlinking example

RDB2RDF

Conclusions

Ontology matching in three steps

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

Reconciliation can be performed in 3 steps

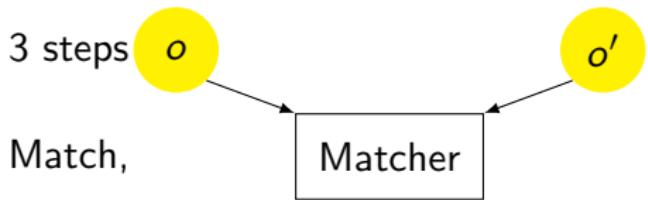
o

o'

Ontology matching in three steps

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

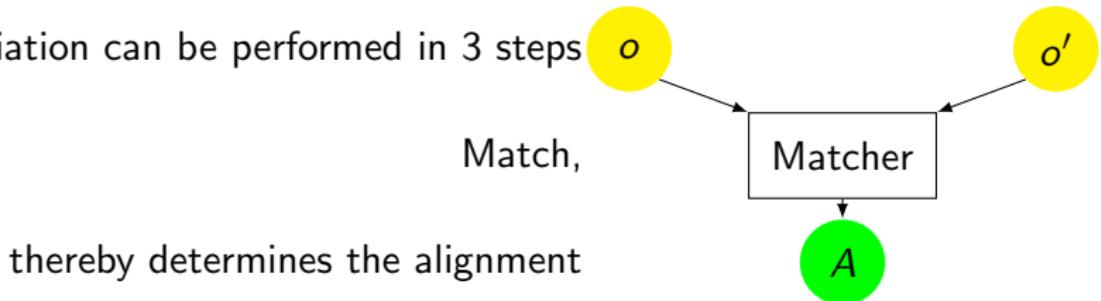
Reconciliation can be performed in 3 steps



Ontology matching in three steps

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

Reconciliation can be performed in 3 steps



Ontology matching in three steps

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

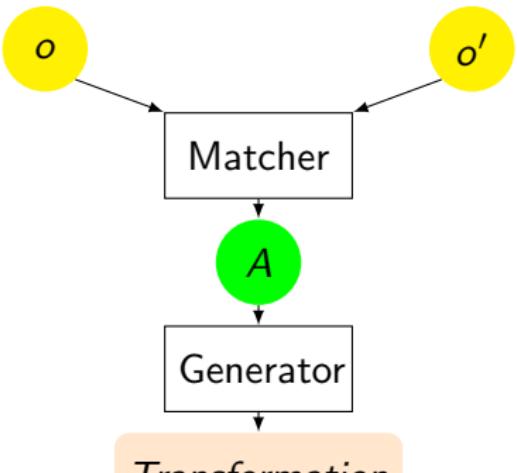
Reconciliation can be performed in 3 steps

Match,

thereby determines the alignment

Generate

a processor (for merging, transforming, etc.)



Ontology matching in three steps

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

Reconciliation can be performed in 3 steps

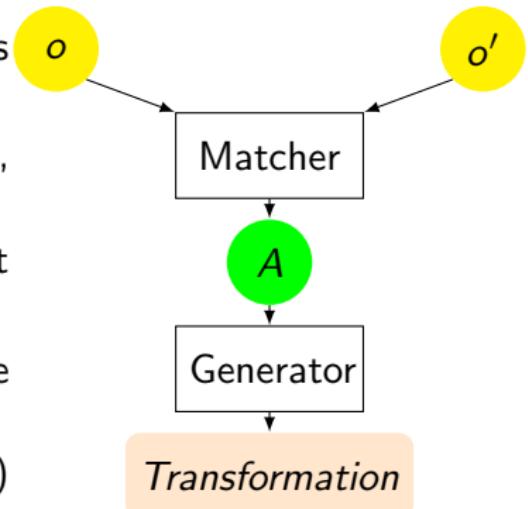
Match,

thereby determines the alignment

Generate

a processor (for merging, transforming, etc.)

Apply



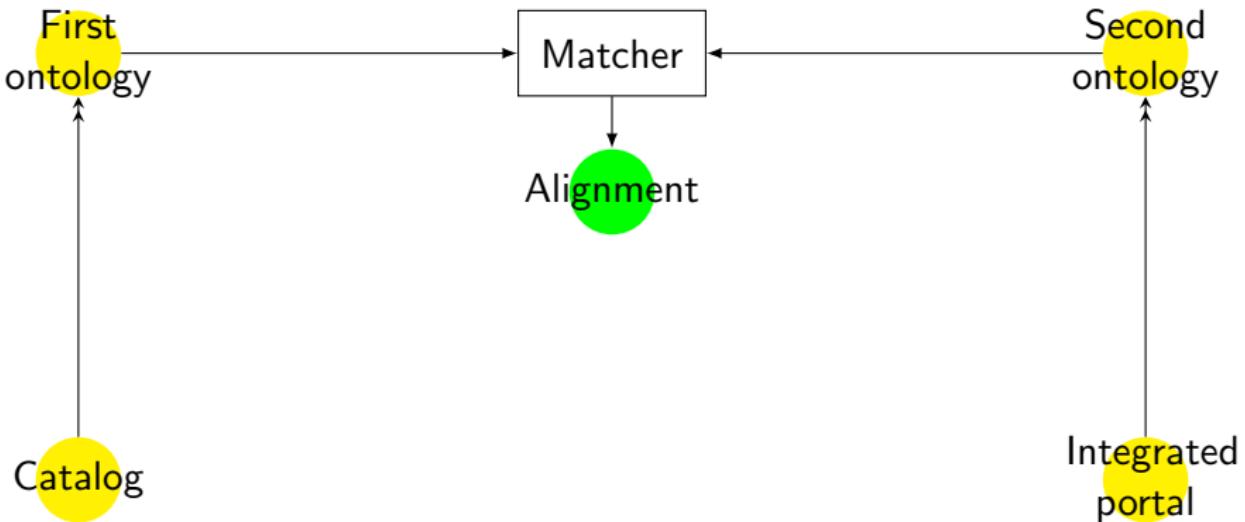
Application: Catalog integration

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



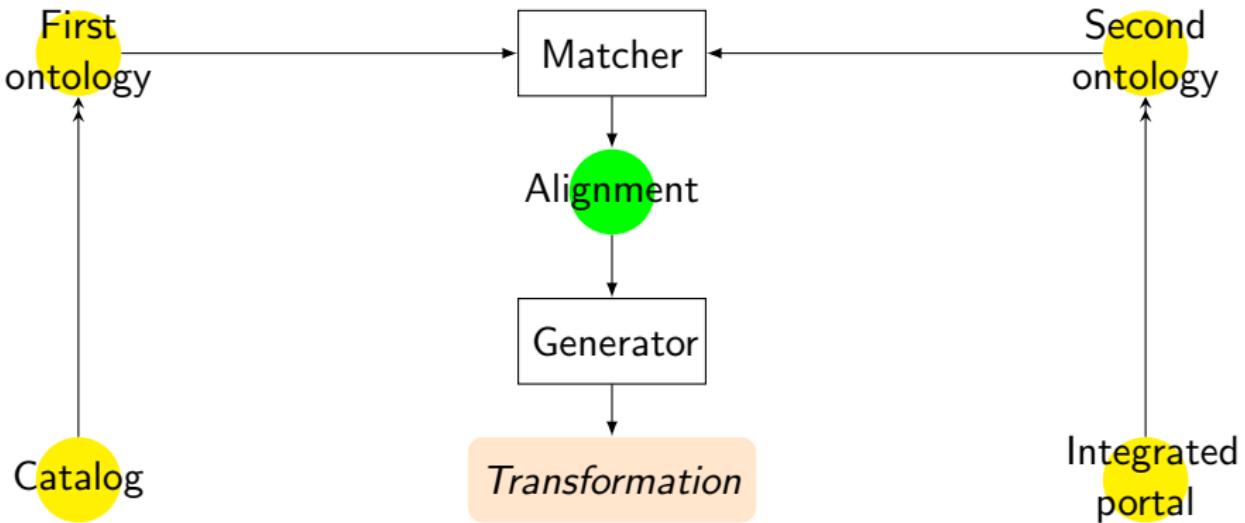
Application: Catalog integration

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



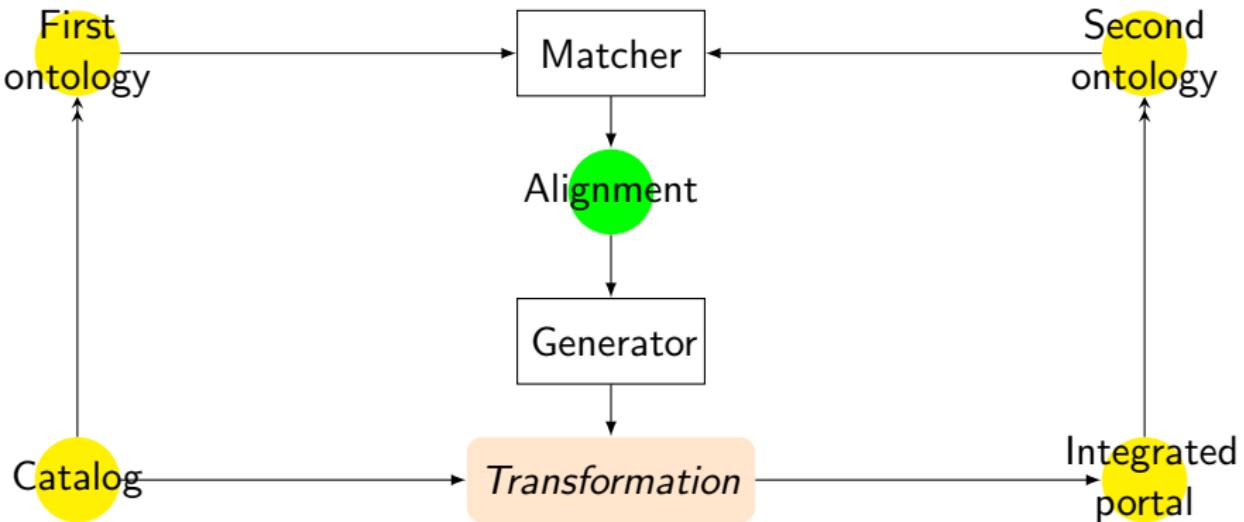
Application: Catalog integration

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



Application: Catalog integration

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



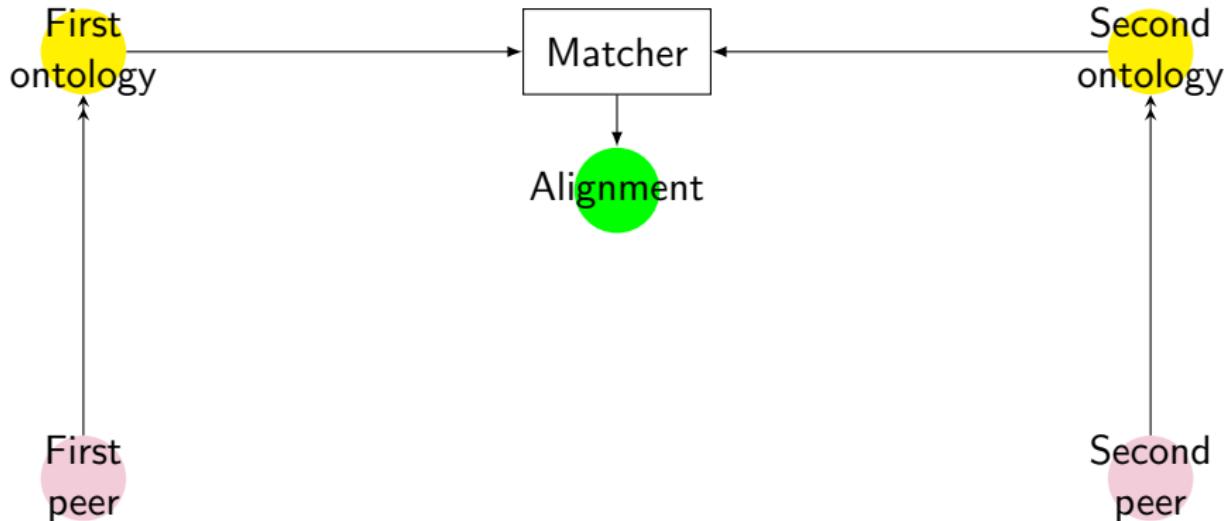
Applications: Query answering

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



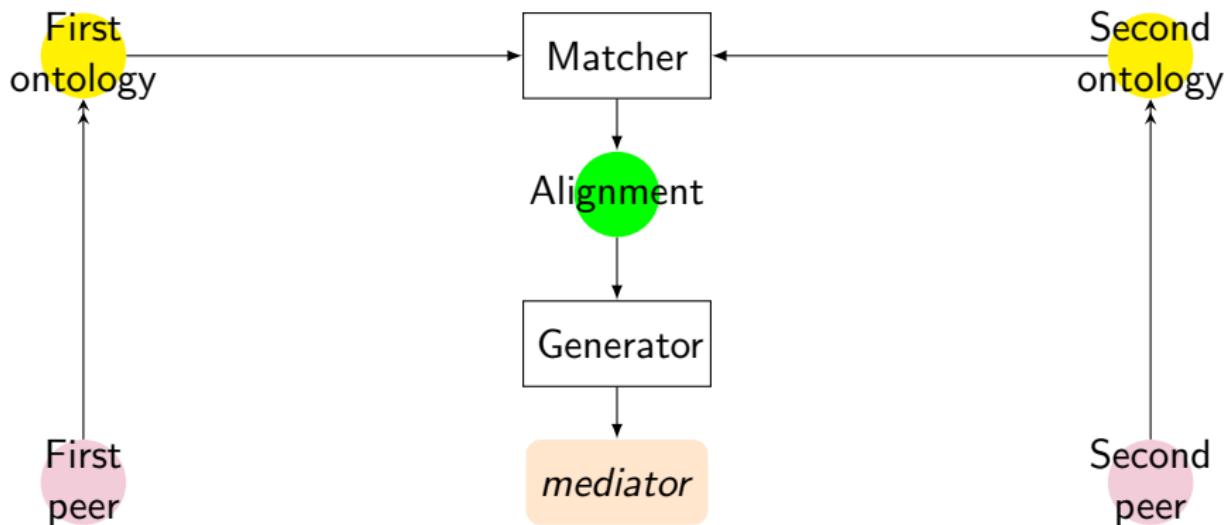
Applications: Query answering

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



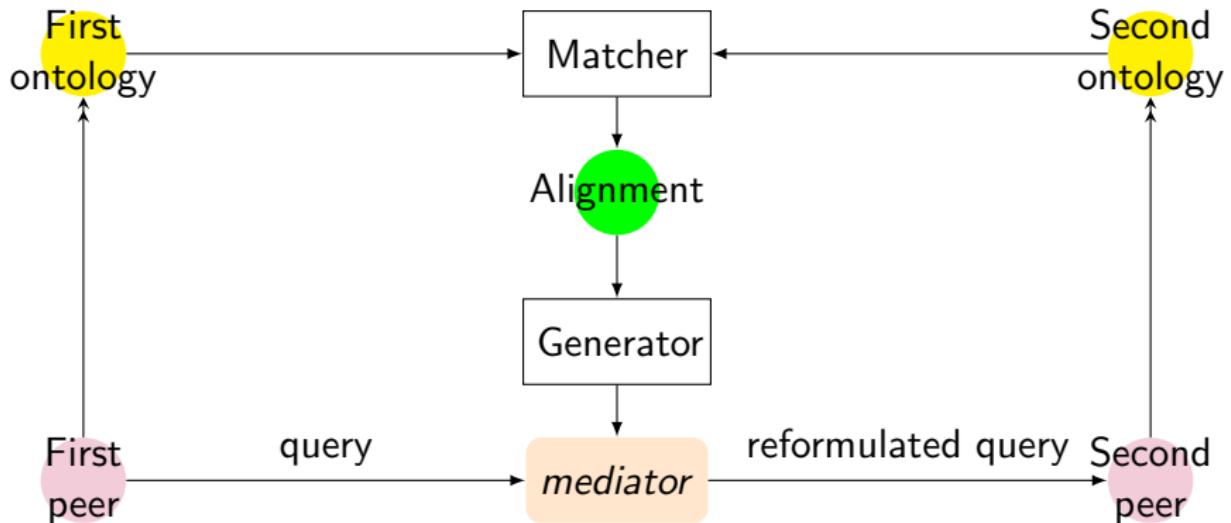
Applications: Query answering

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



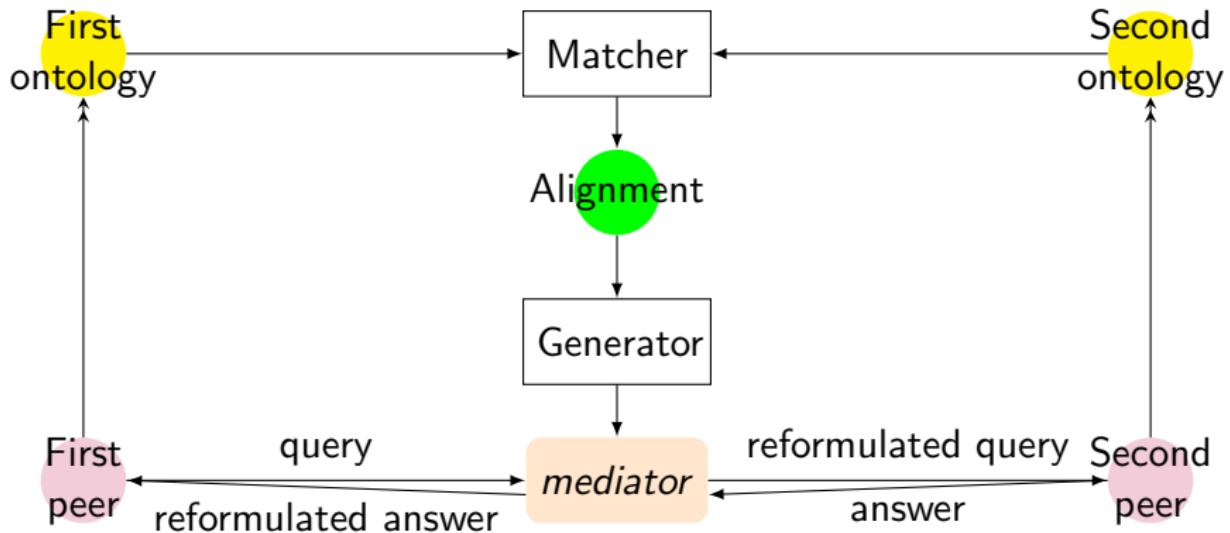
Applications: Query answering

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



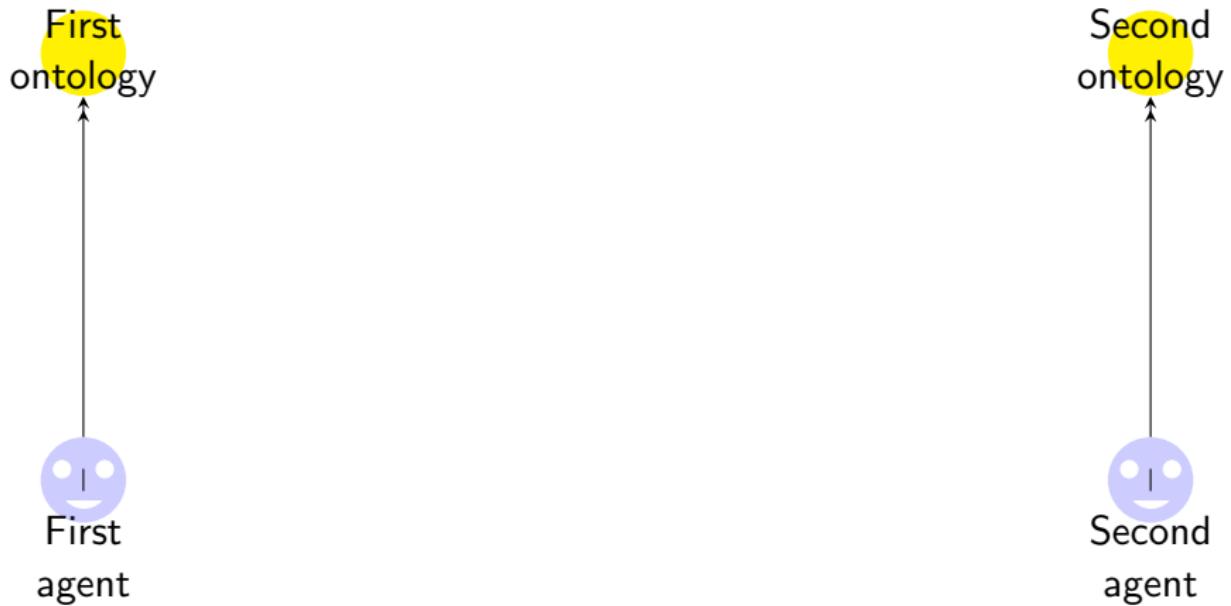
Applications: Query answering

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



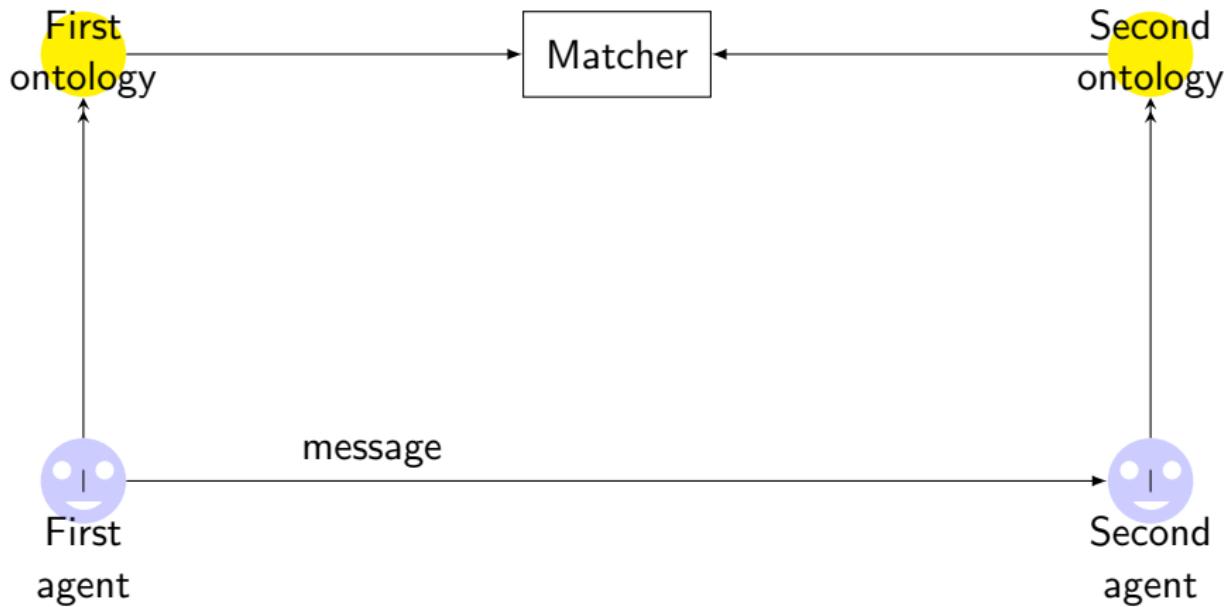
Applications: Agent communication

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



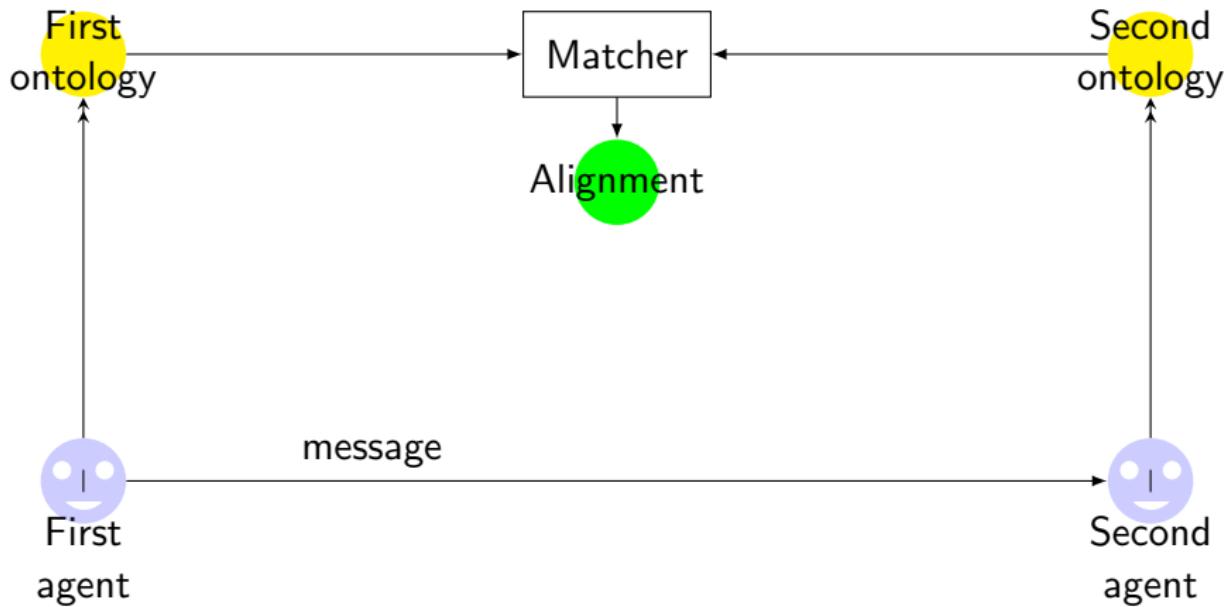
Applications: Agent communication

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



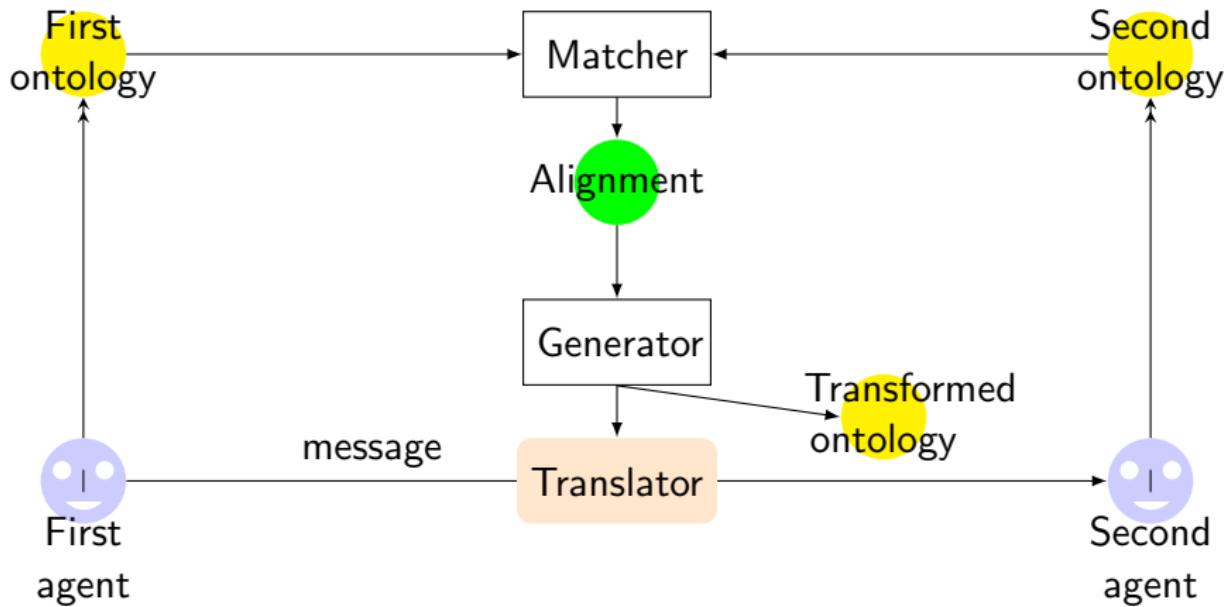
Applications: Agent communication

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



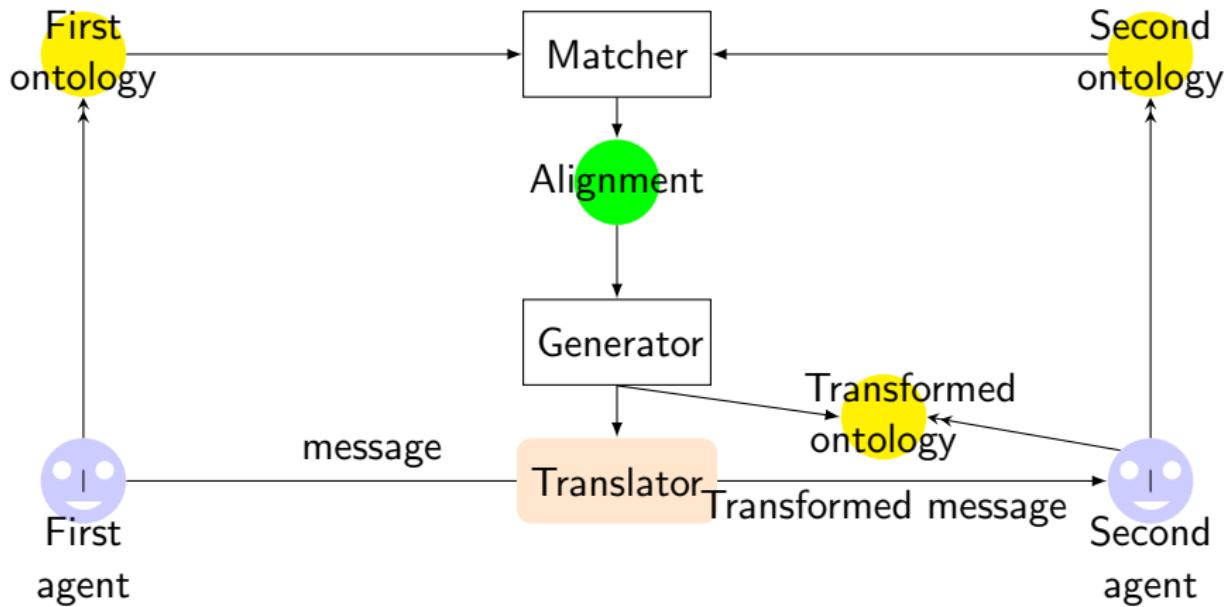
Applications: Agent communication

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



Applications: Agent communication

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



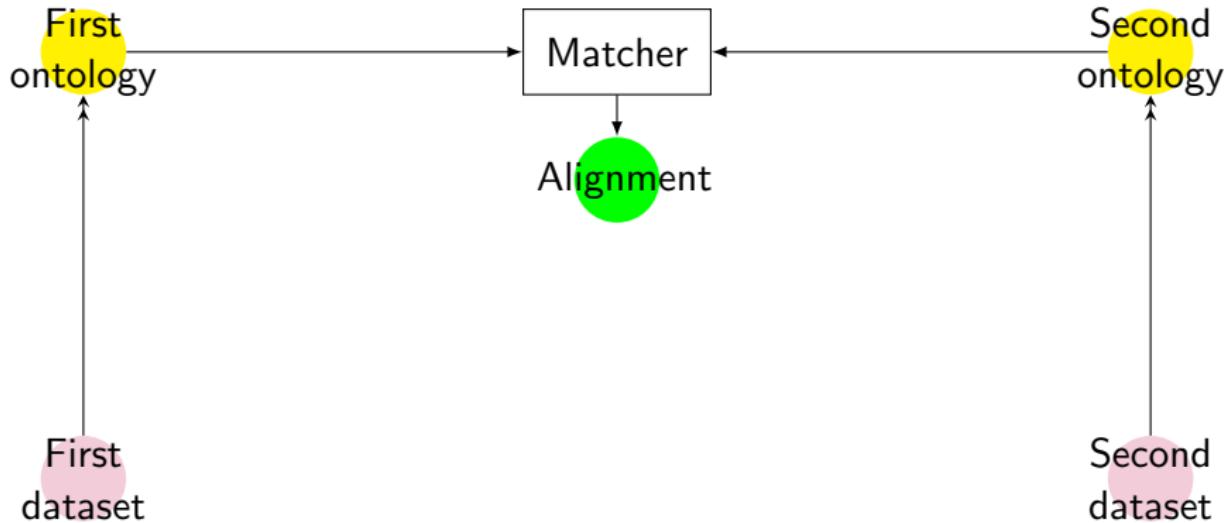
Data interlinking

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



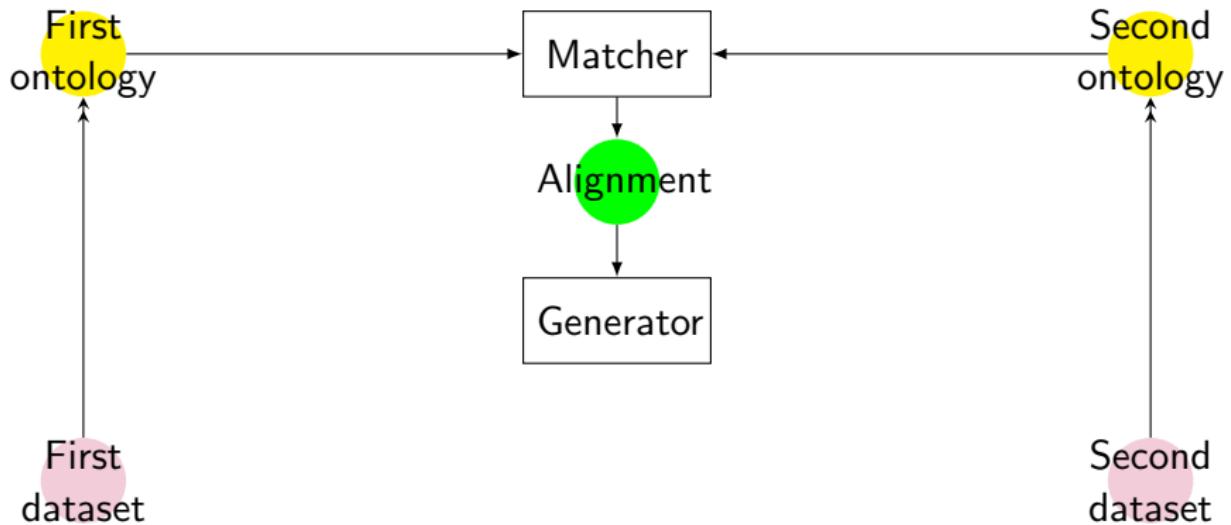
Data interlinking

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



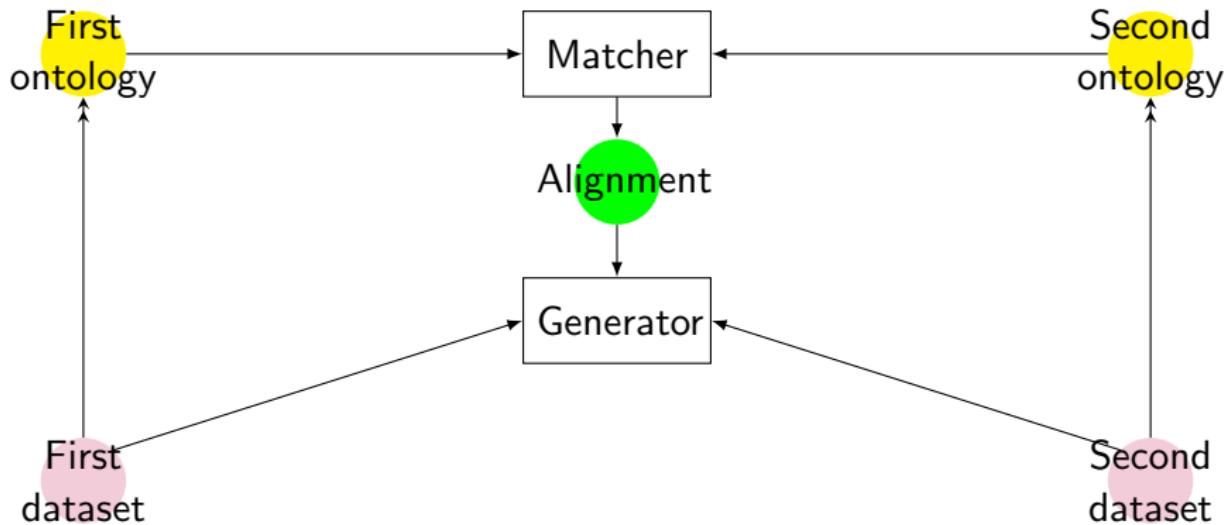
Data interlinking

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



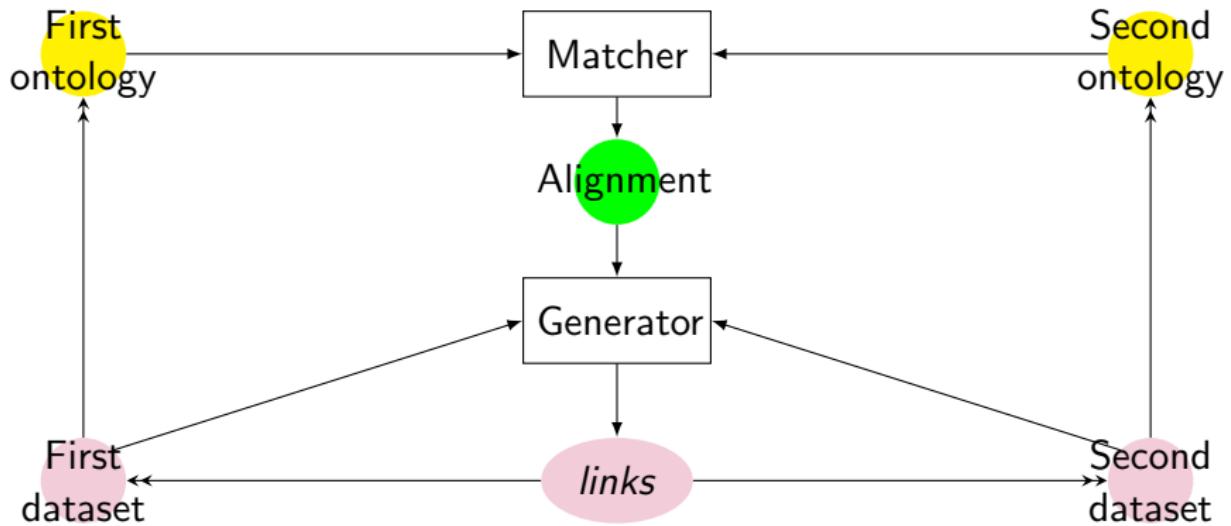
Data interlinking

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



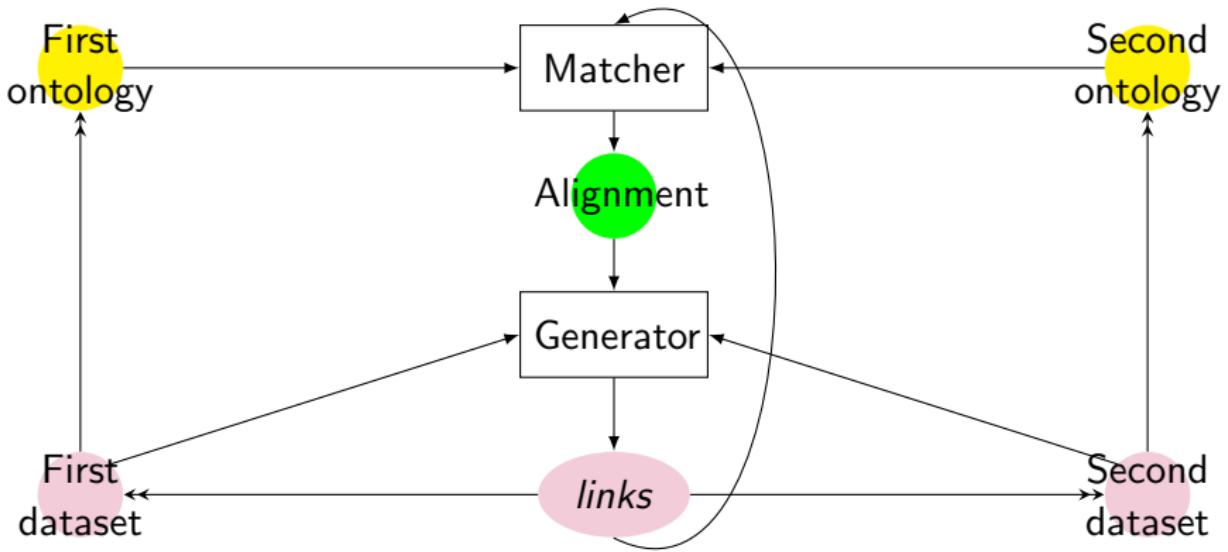
Data interlinking

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



Data interlinking

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



- ▶ Framework: Alignment API;
- ▶ <http://alignapi.gforge.inria.fr>
- ▶ Usable with various ontology APIs (Jena, OWL API, SKOS API);
- ▶ Offers an expressive alignment language (EDOAL)
- ▶ Alignment server infrastructure;
- ▶ Ontology alignment evaluation initiative (OAEI & SEALS);
- ▶ Used in more than 30 systems with published papers;

Alignment API Renderers as mediator generators

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

- ▶ XSLT: transforms XML data
- ▶ SPARQLConstruct: transforms SPARQL data
- ▶ SPARQLSelect
- ▶ SILK: generates links within RDF
- ▶ COWLMapping: generates mappings between OWL ontologies
- ▶ SWRL: generates Horn-clause-like rules
- ▶ OWLAxioms: merges two ontologies
- ▶ SKOS: generates SKOS mappings
- ▶ . . . missing is the query mediator, because they interpret alignments.

Table of Contents

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

Mediating RDF data and alignments

Alignment API and mediators

Data interlinking example

RDB2RDF

Conclusions

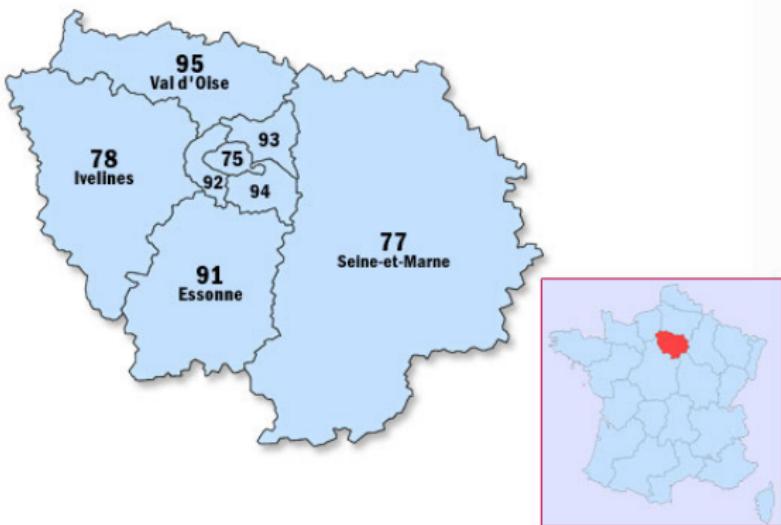
Example: INSEE dataset

Région table:

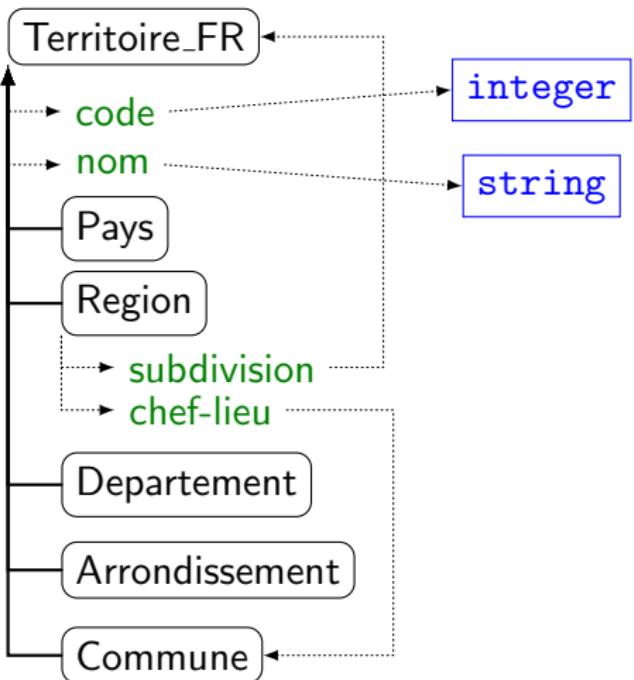
code	nom	chef-lieu
11	Île-de-France	75056
21	Champagne-Ardenne	51108
22	Picardie	80021

Sous-région table:

région	département
11	75
11	77
11	78
11	91
11	92
11	93



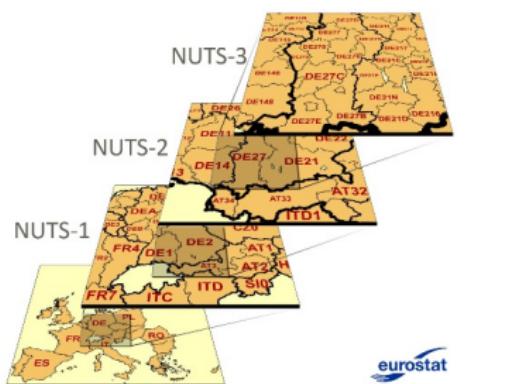
Example: Administrative ontology



Example: NUTS dataset

NUTSRegion table:

level	code	name	hasParentRegion
0	FR	FRANCE	
1	FR1	ÎLE DE FRANCE	FR
2	FR10	Île de France	FR1
3	FR101	Paris	FR10
3	FR104	Essonne	FR10

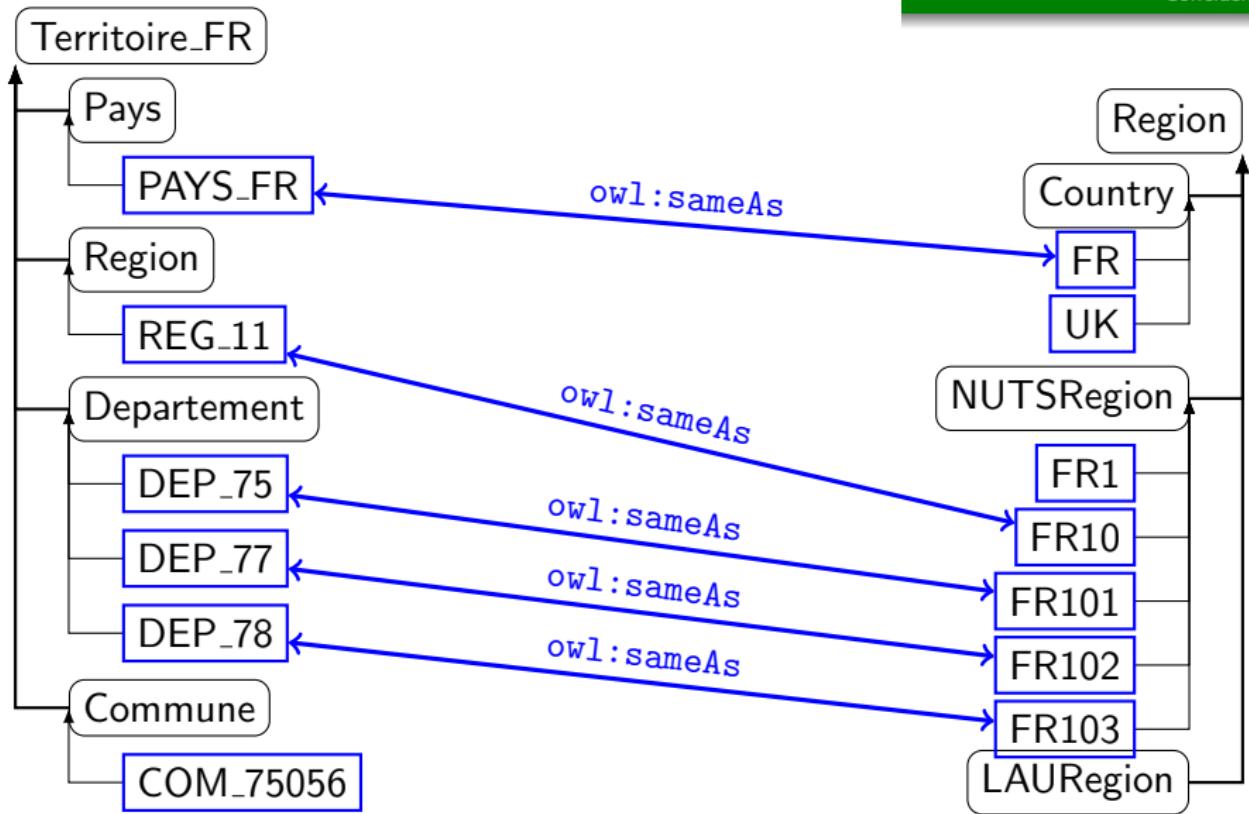


Example: Linking INSEE and NUTS

NUTS: Nomenclature of territorial units for statistics

#INSEE	INSEE name	NUTS Level	#NUTS
1	Pays	0	34
26	Région	1	142
100	Département	2	344
342	Arrondissement	3	1488
4036	Canton	4	
52422	Commune	5	

Example: Linking INSEE and NUTS

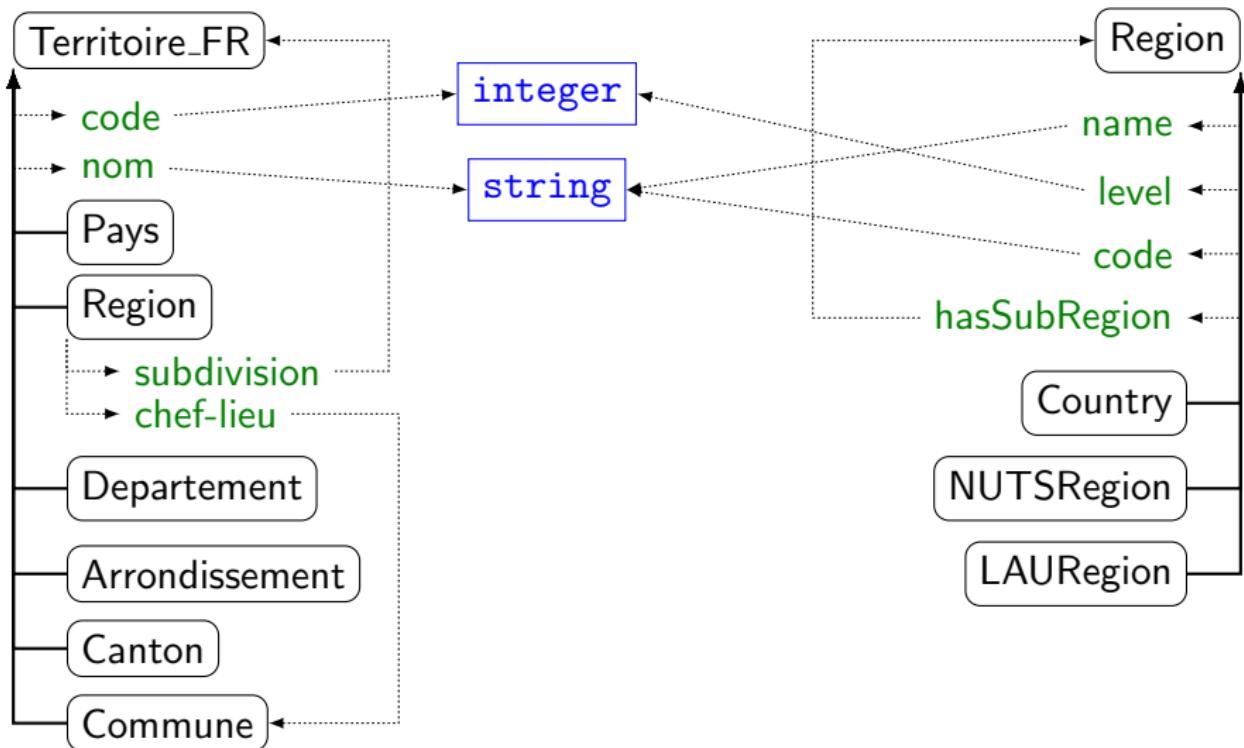


Data interlinking techniques may be based on:

- ▶ Data ID (URLs);
- ▶ Data keys
- ▶ Data content
- ▶ External relations (links)
- ▶ Common ontologies
- ▶ Ontology alignments

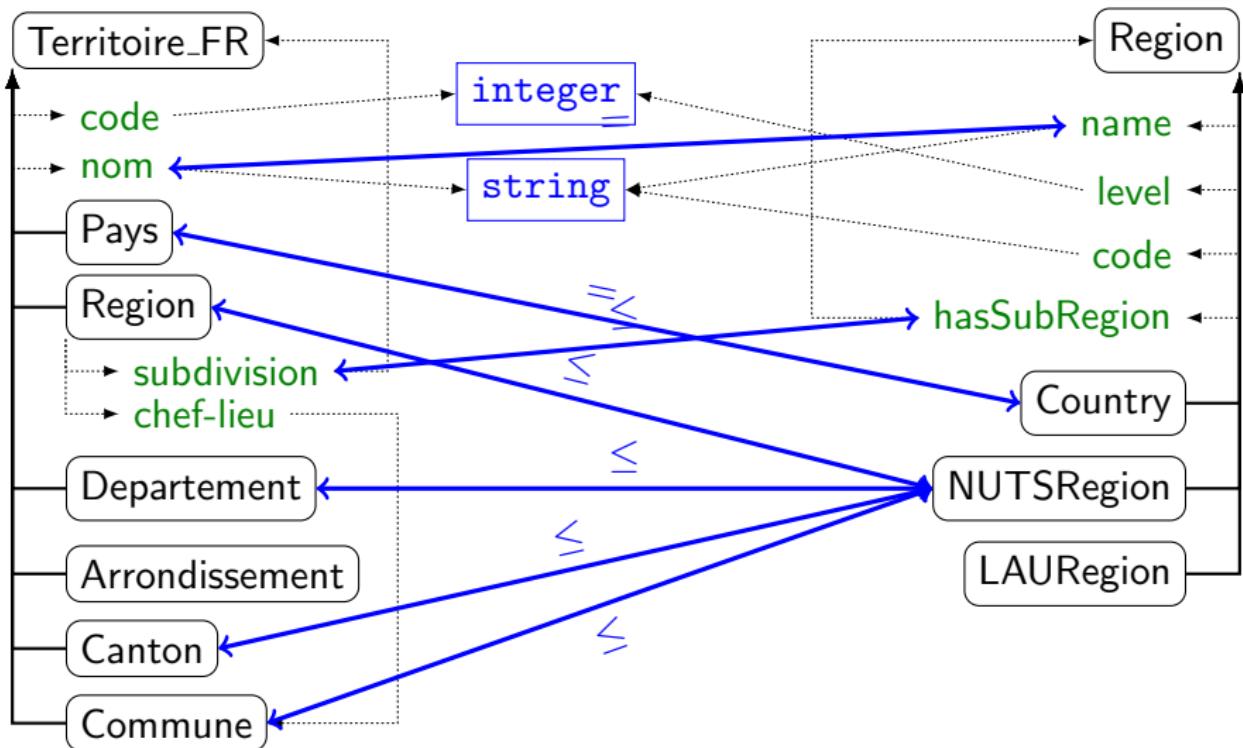
INSEE and NUTS: ontology alignment

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



INSEE and NUTS: ontology alignment

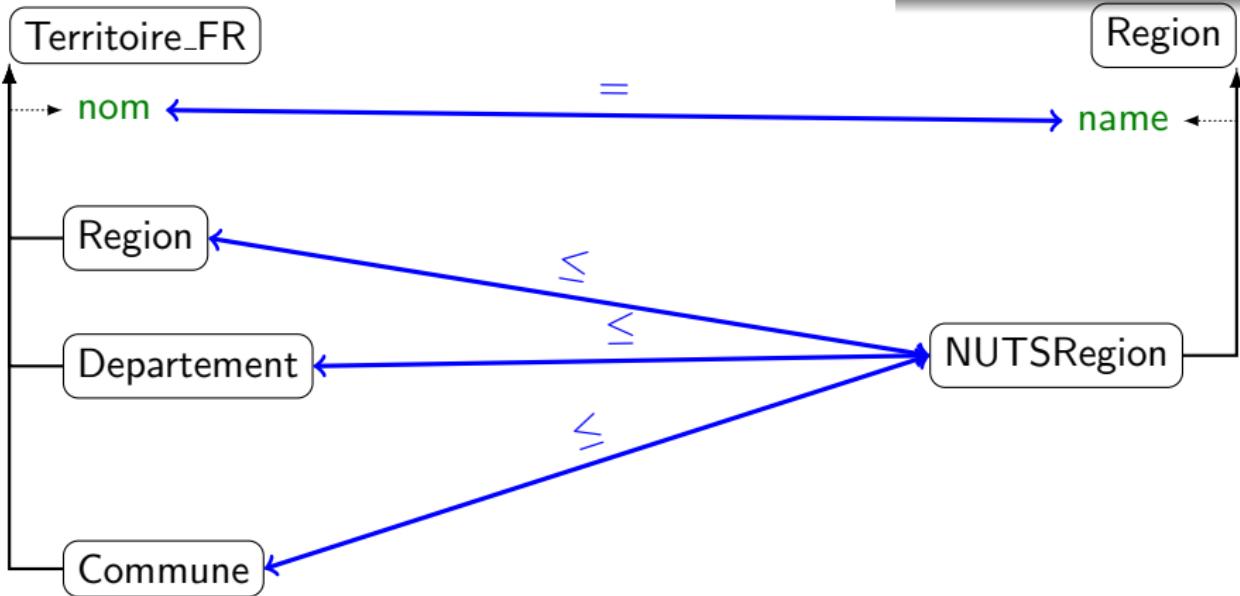
Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



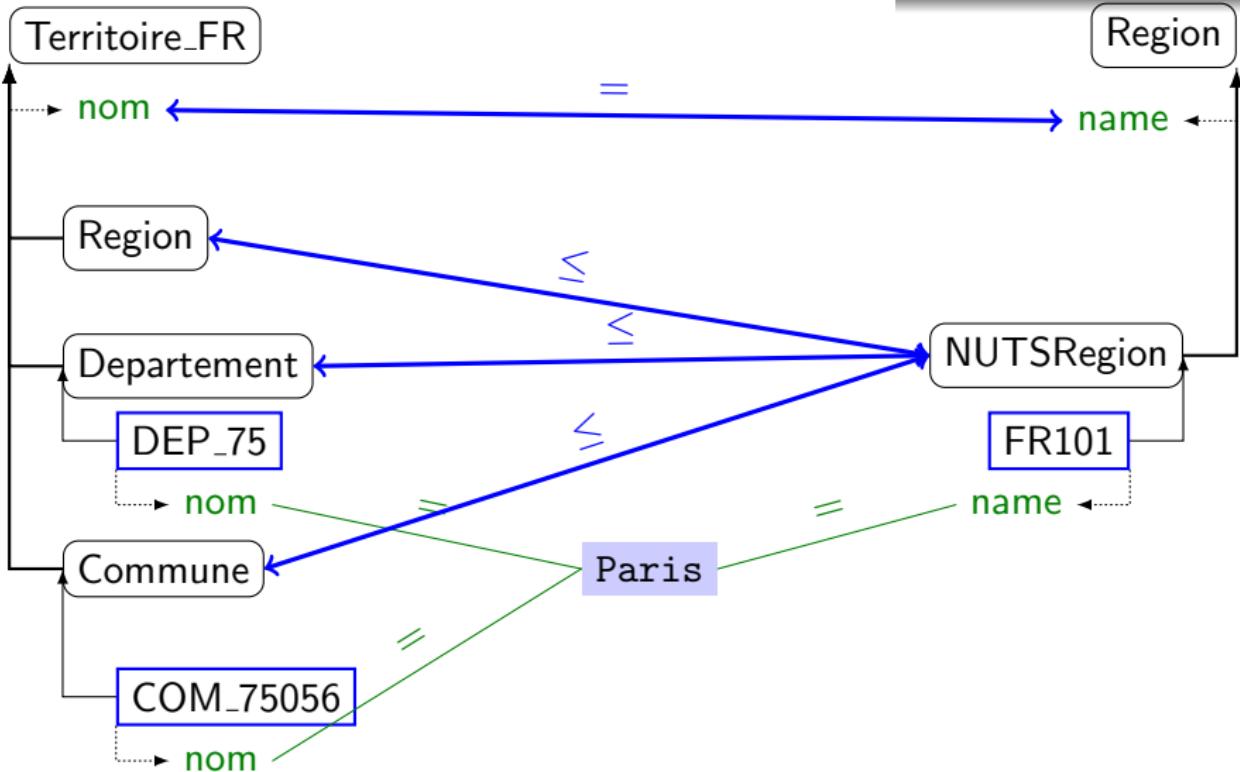
- ▶ Find matching concepts [concept matching];
- ▶ For each of them, determine matching properties based on the similarity between their values in both datasets [property matching];
- ▶ From them find property combinations identifying corresponding entities [key extraction];
- ▶ Link corresponding entities [link generation].

For instance, $\text{nom}/\text{Region}_{INSEE} \subseteq \text{name}/\text{NUTSRegion}_{NUTS}$ and moreover they are unambiguous.

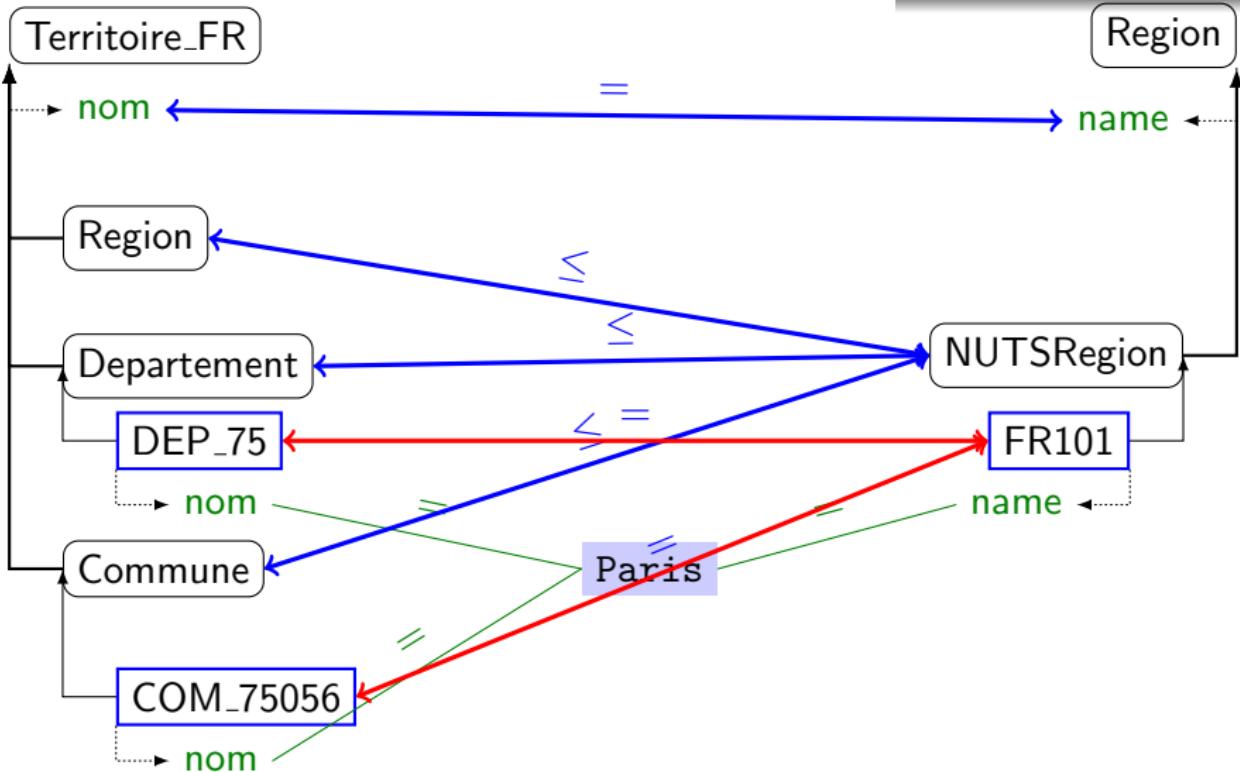
Simple alignments are not sufficient



Simple alignments are not sufficient

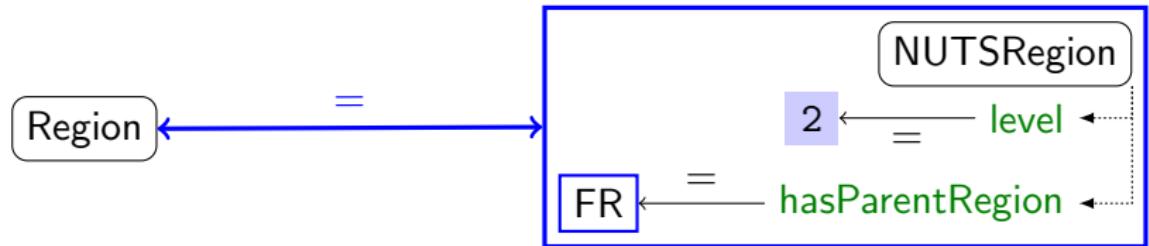


Simple alignments are not sufficient



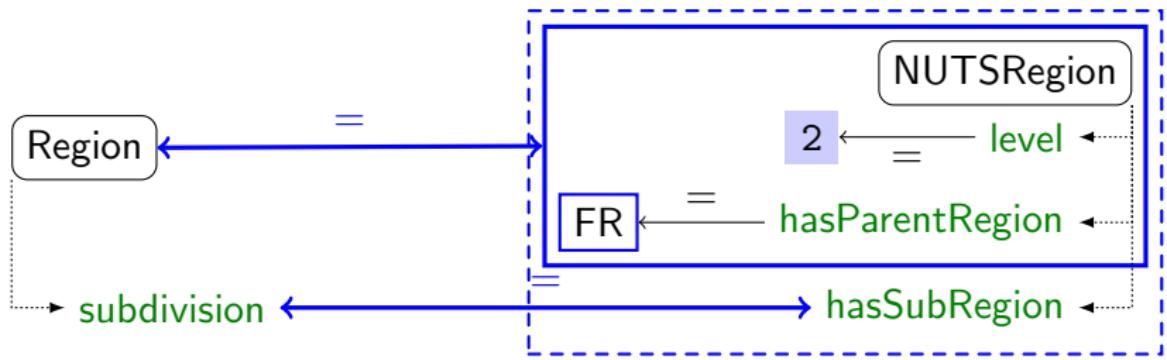
Expressive alignments are necessary

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



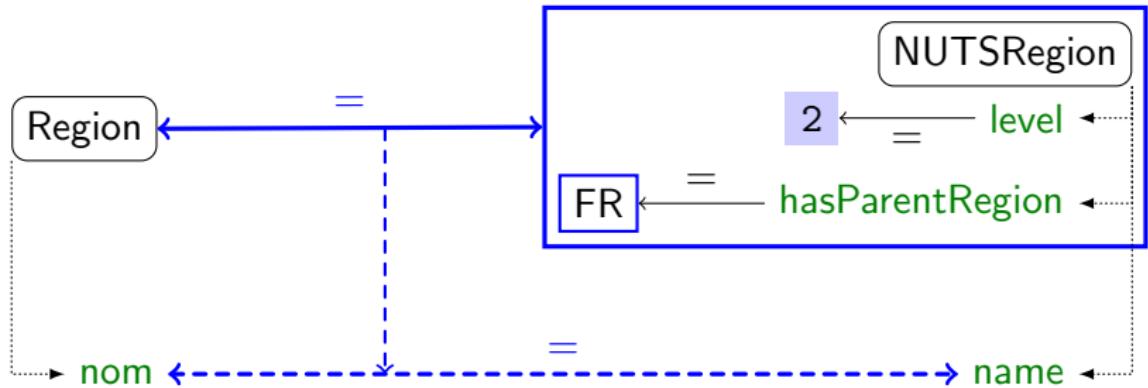
Expressive alignments are necessary

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



Expressive alignments are necessary

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



Query generation

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

```
PREFIX insee: <http://rdf.insee.fr/ontologie-geo-2006.rdf#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?r
FROM <http://rdf.insee.fr/geo/regions-2011.rdf>
WHERE {
    ?r rdf:type insee:Region .
}
```

```
PREFIX nuts: <http://ec.europa.eu/eurostat/ramon/ontologies/geographi
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?n
FROM <http://ec.europa.eu/eurostat/ramon/rdfdata/nuts2008/>
WHERE {
    ?n rdf:type nuts:NUTSRegion .
    ?n nuts:level 2^^xsd:int .
    ?n nuts:hasParentRegion nuts:FR1 .
}
```

```
PREFIX insee: <http://rdf.insee.fr/ontologie-geo-2006.rdf#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX insee: <http://rdf.insee.fr/ontologie-geo-2006.rdf#>
CONSTRUCT {
    ?r rdf:type nuts:NUTSRegion .
    ?r nuts:level 2^^xsd:int .
    ?r nuts:hasParentRegion nuts:FR1 .
}
FROM <http://rdf.insee.fr/geo/regions-2011.rdf>
WHERE {
    ?r rdf:type insee:Region .
}
```

SameAs link generation generation

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

```
CONSTRUCT { ?r owl:sameAs ?n . }
PREFIX insee: <http://rdf.insee.fr/ontologie-geo-2006.rdf#>
PREFIX nuts: <http://ec.europa.eu/eurostat/ramon/ontologies/geographi
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
FROM <http://rdf.insee.fr/geo/regions-2011.rdf>
FROM <http://ec.europa.eu/eurostat/ramon/rdfdata/nuts2008/>
WHERE {
    ?r rdf:type insee:Region .
    ?r insee:nom ?l .
    ?n rdf:type nuts:NUTSRegion .
    ?n nuts:name ?l .
    ?n nuts:level 2^^xsd:int .
    ?n nuts:hasParentRegion nuts:FR1 .
}
```

- ▶ Declare data sources (DataSource);
- ▶ Circumscribe entities to compare (Source/TargetDataset);
- ▶ Describe how to compare them (LinkageRule):
 - ▶ Select properties to compare through paths (Input);
 - ▶ Compute distances between them (Compare+threshold);
 - ▶ Aggregate all comparisons (Aggregate);
- ▶ Select those pairs of entities to be linked (Filter);
- ▶ Generate links (Output+thresholds).

A Silk script

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

Consider a linking script between INSEE and NUTS:

```
<Silk>
  <Prefix id="nuts"
    namespace="http://ec.europa.eu/.../geographic.rdf#" />
  <Prefix id="insee"
    namespace="http://rdf.insee.fr/geo/" />
  <Interlinks>
    <Interlink id="linkingNUTS">
      <LinkType>owl:sameAs</LinkType>

  <DataSource id="nuts2008"
    type="sparqlEndpoint">
    <Param name="endpointURI"
      value="http://localhost:9091/.../internal"/>
    <Param name="graph"
      value="http://localhost:9091/.../nuts2008-complete-1"/>
  </DataSource>
  <DataSource id="insee2010"
    type="sparqlEndpoint">
    <Param name="endpointURI"
      value="http://localhost:9091/.../internal"/>
    <Param name="graph"
      value="http://localhost:9091/.../source/regions-2010-1"/>
  </DataSource>
  <Thresholds accept="0.9" verify="0.7" />
  <Outputs>
    <Output type="sparql">
      <Param name="graphUri"
        value="http://localhost:9091/.../source/insee-nuts-silk"/>
      <Param name="uri"
        value="http://localhost:9091/.../lifted/"/>
      <Param name="parameter" value="update"/>
    </Output>
    <SourceDataset dataSource="nuts2008" var="s">
      <RestrictTo>?s rdf:type nuts:NUTSRegion.
      ?s nuts:level 2.
    </RestrictTo>
    </SourceDataset>
    <TargetDataset dataSource="insee2010" var="ss">
      <RestrictTo>?ss rdf:type insee:Region</RestrictTo>
    </TargetDataset>
    <LinkageRule>
      <Aggregate type="max">
        <Compare metric="levenshteinDistance"
          threshold=".2">
          <Input path="?s/nuts:name"/>
          <Input path="?ss/insee:nom"/>
        </Compare>
      </Aggregate>
      <LinkageRule>
        <Interlinks>
          <Interlink id="linkingNUTS">
            <LinkType>owl:sameAs</LinkType>
          </Interlink>
        </Interlinks>
      </LinkageRule>
    </LinkageRule>
  </Outputs>
</Silk>
```

- ▶ Ontology alignments are schema-level expression of correspondences;
- ▶ They are useful for focussing the search;
- ▶ Expressive alignments are necessary;
- ▶ They can be turned into SPARQL-based link generators.

but it is also necessary to express instance level constraints:

- ▶ for converting data (e.g., mph vs. m/s);
- ▶ for expressing matching constraint on data (e.g., similarity).

Table of Contents

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

Mediating RDF data and alignments

Alignment API and mediators

Data interlinking example

RDB2RDF

Conclusions

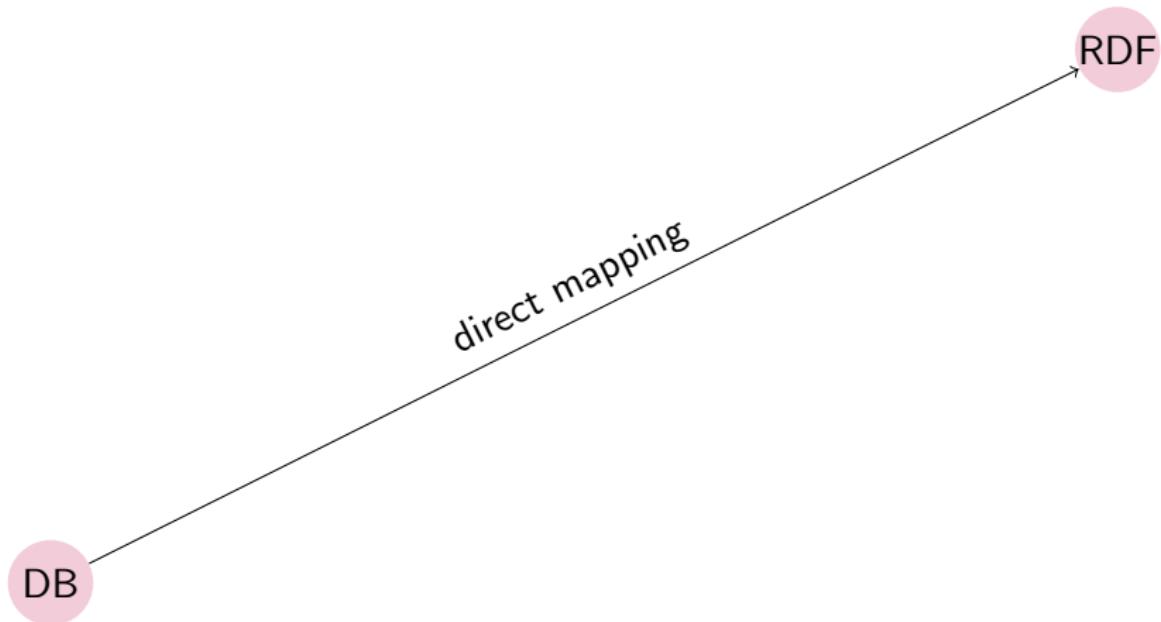
RDB2RDF principles

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

DB

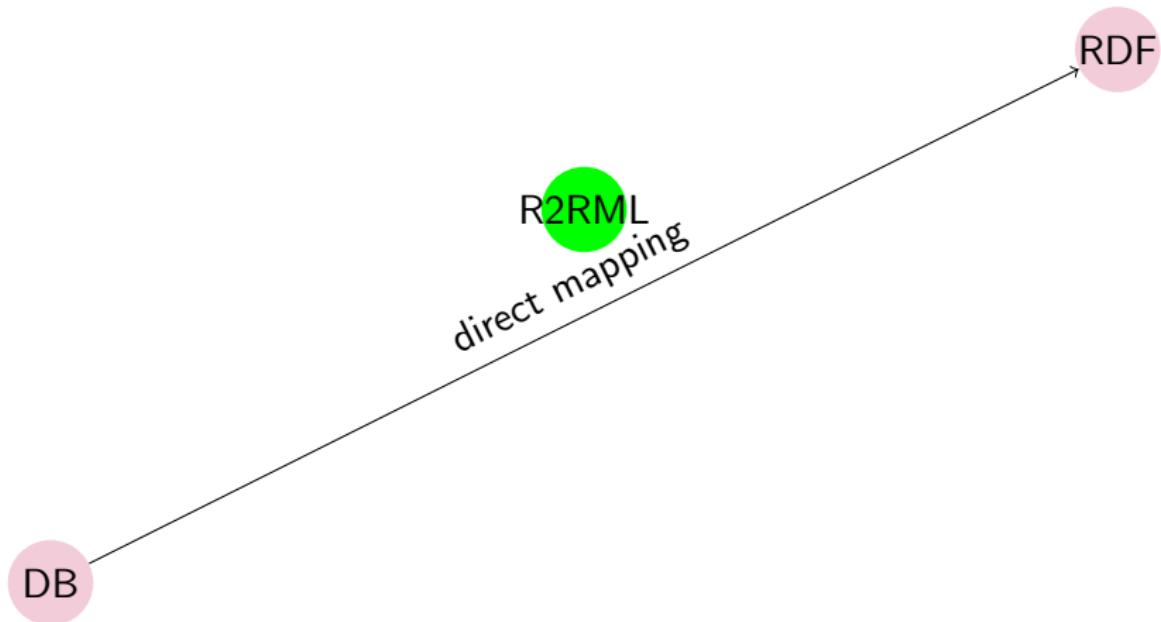
RDB2RDF principles

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



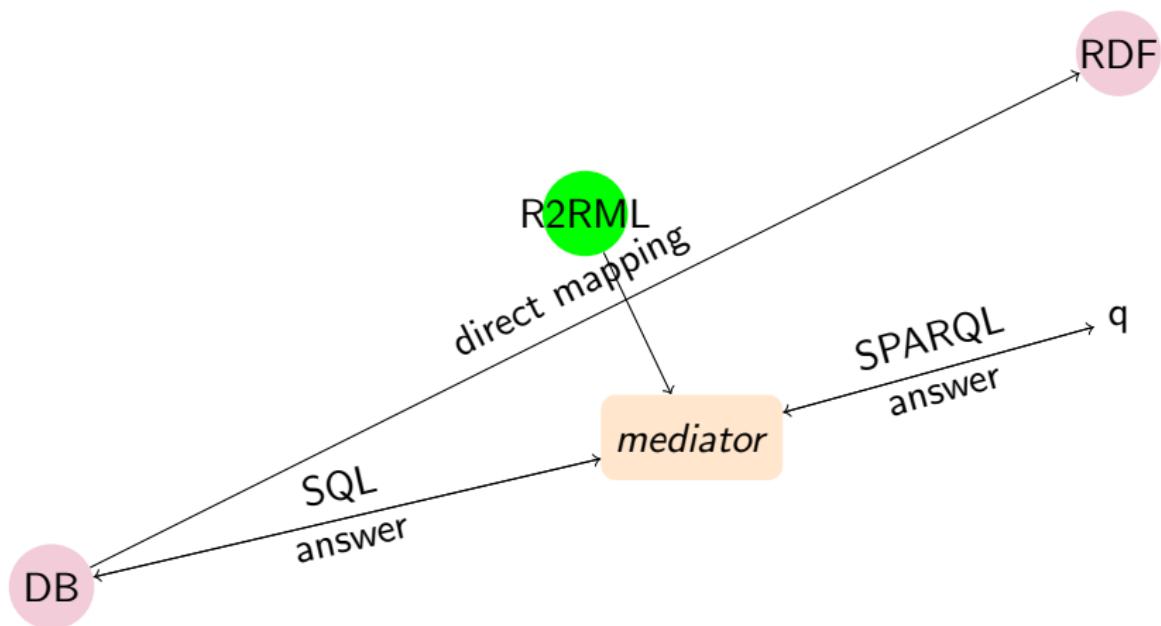
RDB2RDF principles

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



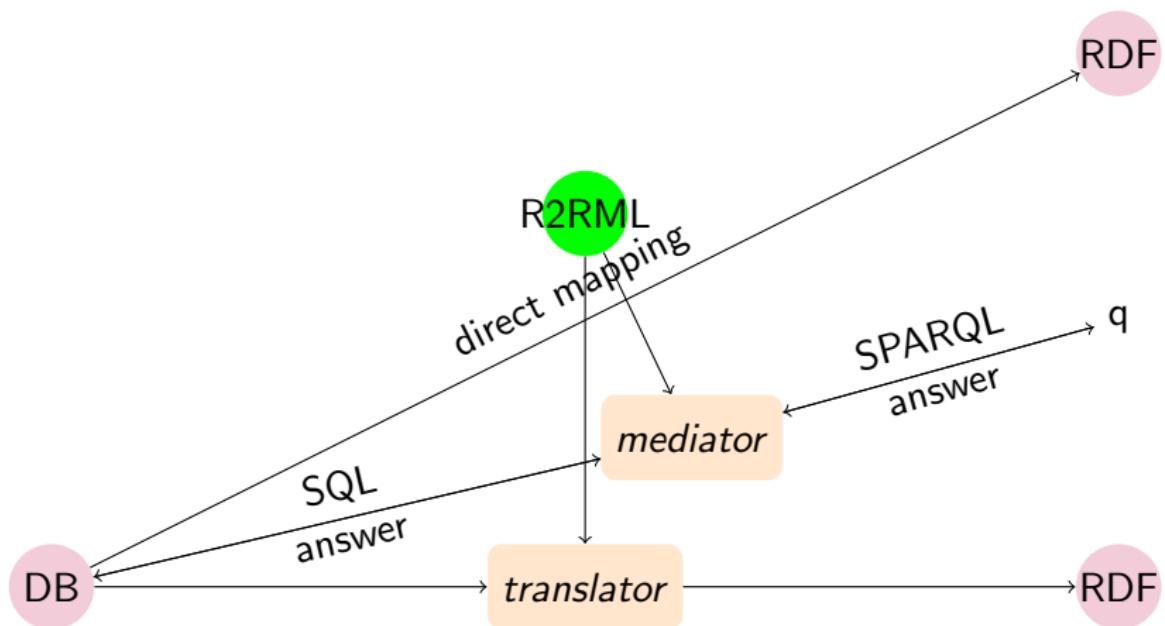
RDB2RDF principles

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



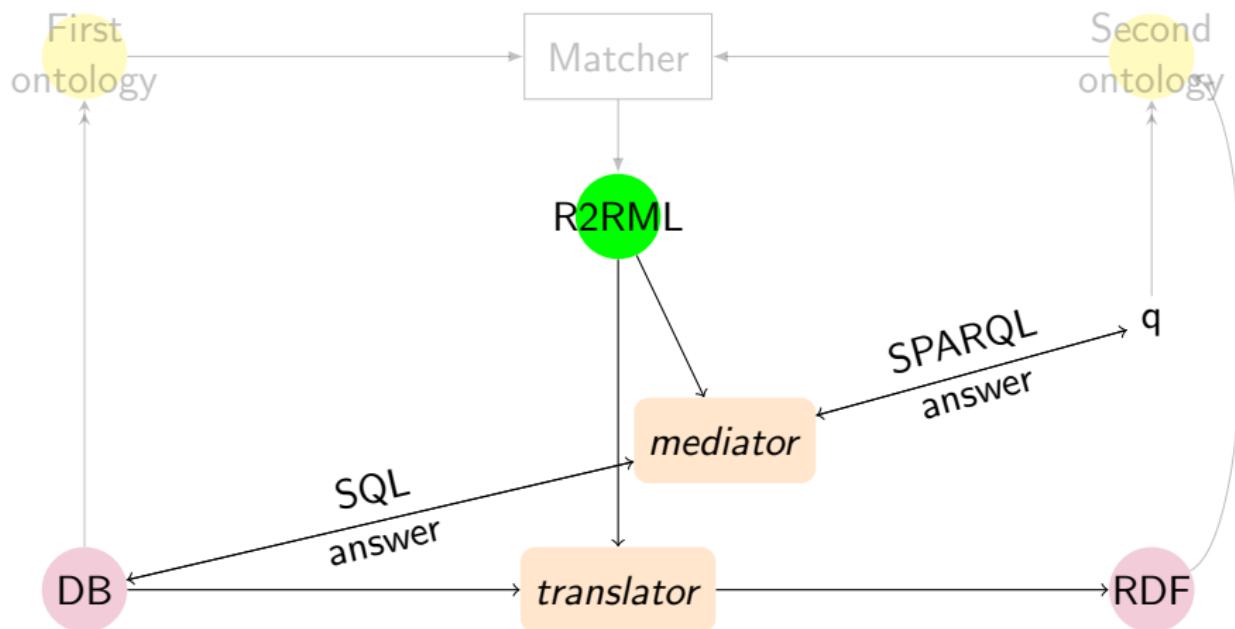
RDB2RDF principles

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



RDB2RDF principles

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions



R2RML vs. Alignments

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

R2RML	Alignments
generated by hand	
RDB specific	based on schemas
SPARQL specific	general purpose
efficient on RDB	other renderers

Now you have the choice!

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

q



Now you have the choice!

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

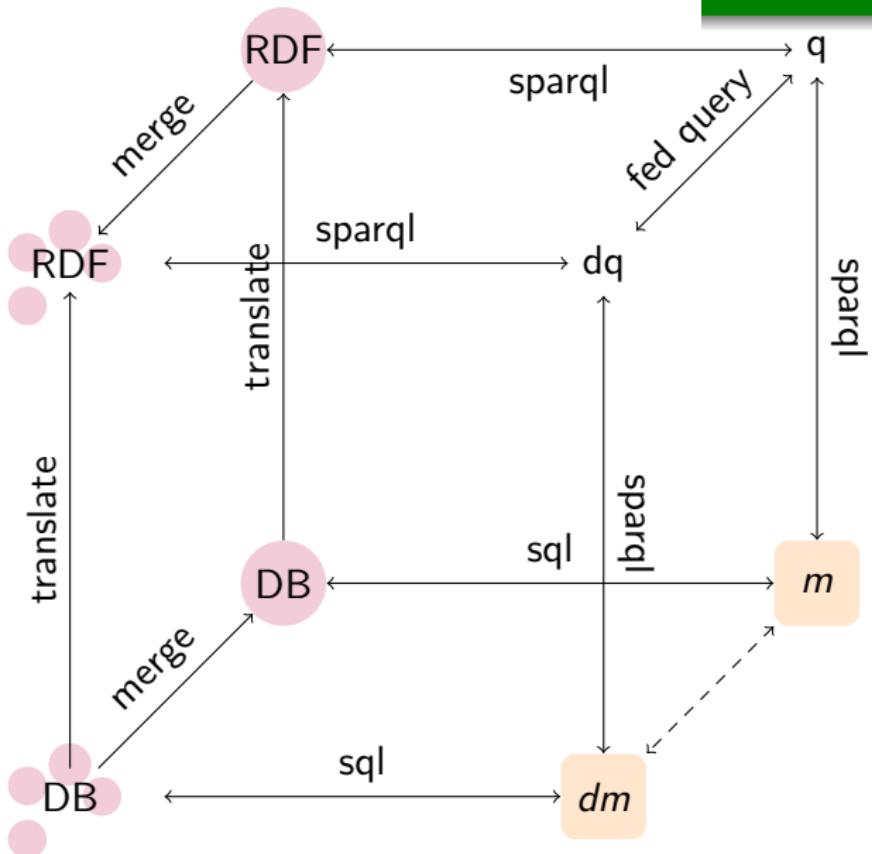


Table of Contents

Mediating RDF data and alignments
Alignment API and mediators
Data interlinking example
RDB2RDF
Conclusions

Mediating RDF data and alignments

Alignment API and mediators

Data interlinking example

RDB2RDF

Conclusions

1. Which reference model? Ontologies?
→ Yes
2. Is SPARQL the most appropriate query language?
→ Both SPARQL and follow-your-nose are decent ways of exploiting data.
3. Mediating various data sources. Statically? Dynamically?
→ See previous slide
4. Coarse vs. fine access control?
→ ??

Representing declaratively
relations between heterogeneous models
(ontology alignments)

is a valuable way to perform mediation
through SPARQL

There are tools for that

Jerome.Euzenat@inria.fr

<http://exmo.inria.fr>