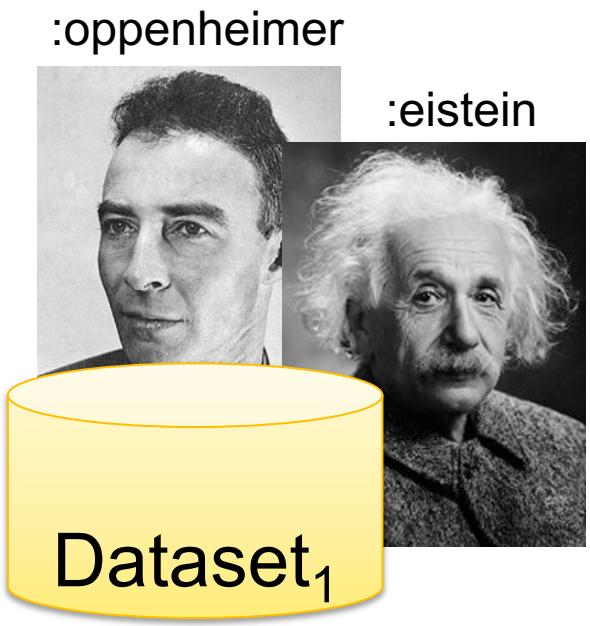
# SPARQL-based linking

cimmino@fi.upm.es

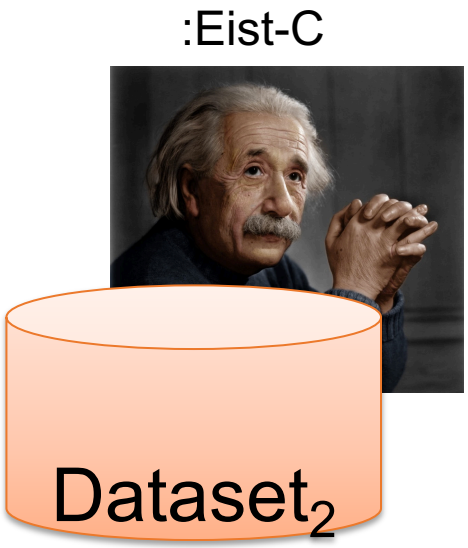
**Andrea Cimmino**
**Ontology Engineering Group**
**Universidad Politécnica de Madrid, Spain**

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 688467*

European Commission | Horizon 2020 European Union funding for Research & Innovation




e.g. Dbpedia

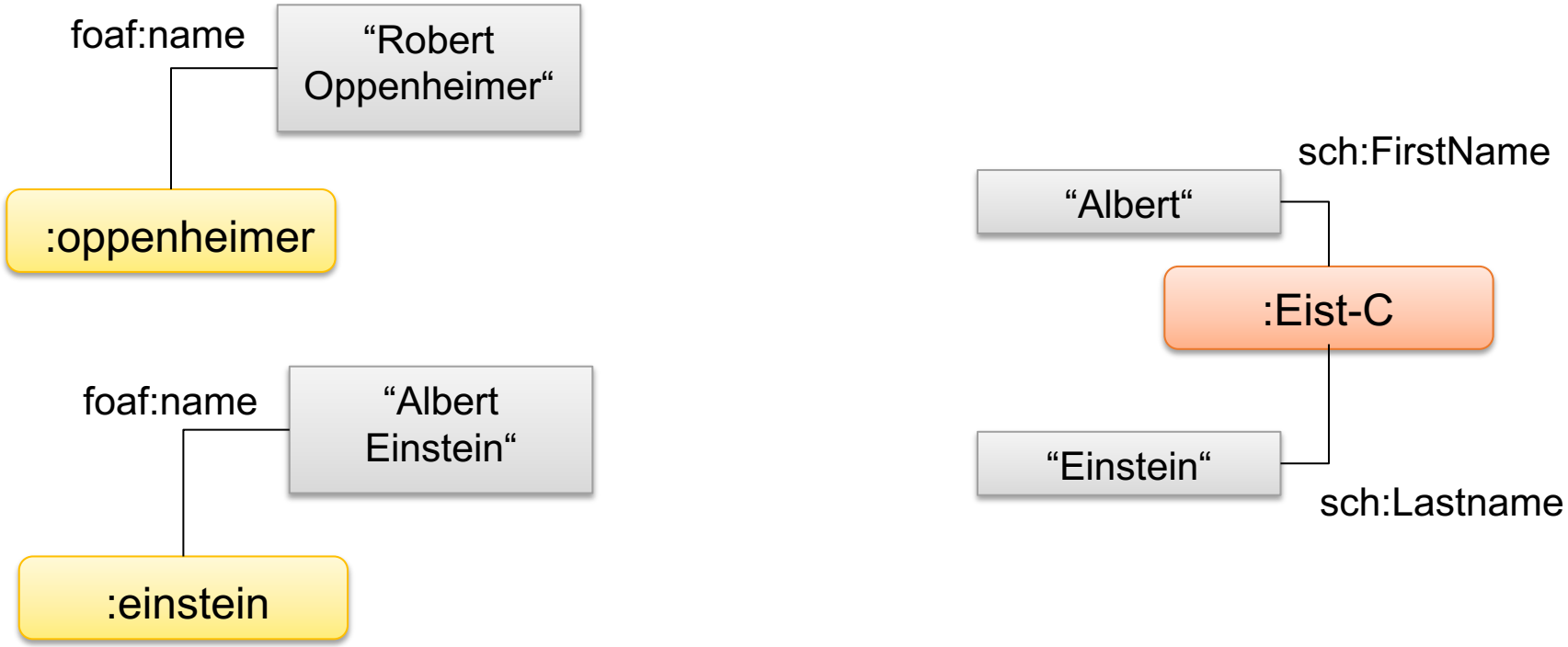



e.g. Wikidata

- RDF linking is a task that consist of generating relationships between RDF resources from different, or the same, dataset. In other words generate links between RDF resources, generally owl:sameAs.

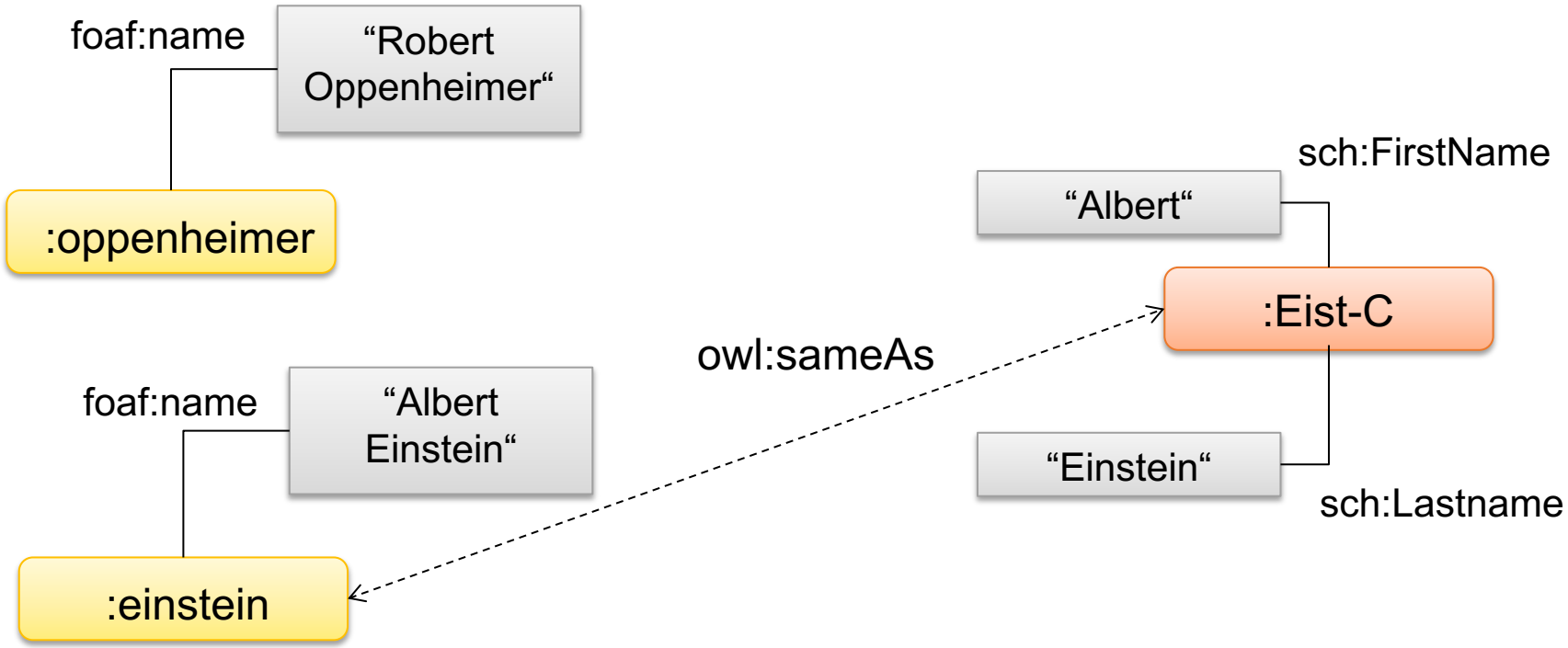
# RDF linking: link rules

- RDF linking relies on one or more link rules in order to generate the links. Link rules specify the conditions under which two RDF resources can be considered the same



# RDF linking: link rules

- *For instance: If the value of foaf:name is the same of the values of sch:FirstName plus sch:LastName, then the RDF resources are the same*



# RDF linking: issues

- RDF linking is a computational expensive task since it has to compare all the resources form one dataset, with all the resources from another. Then, for each RDF resource one or more data comparisons must be performed
- As a result, the way in which link rules are expressed (language), built (nº of comparisons, restrictions to reduce the number of RDF resources to be compared, etc.), and processed has a high impact on the performance results.

# RDF linking: outdated state of the art

- 2 implementations (engines) are the most well known
  - Limes → https://aksw.org/Projects/LIMES.html
  - Silk → http://silkframework.org/

- Usually implementations rely on their own link rule language.
  - This makes impossible to use the same link rule in two or more engines
  - This makes impossible to extend the link rules capabilities since they are tied to the implementation or the language
  - Not based on standards
- These implementations are usually hard to deploy, and to use. In addition, a user usually needs to learn the link rule language.
- These implementations are theoretically efficient

# RDF linking: outdated state of the art

- As a result, currently users find difficult to use these implementations and RDF linking is not a task performed commonly.

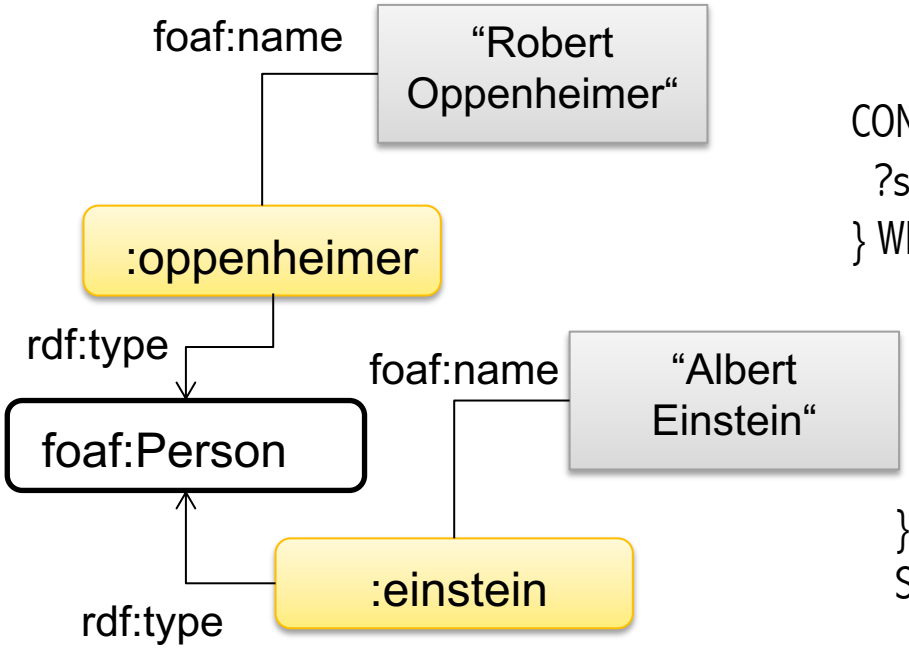
# SPARQL-based linking

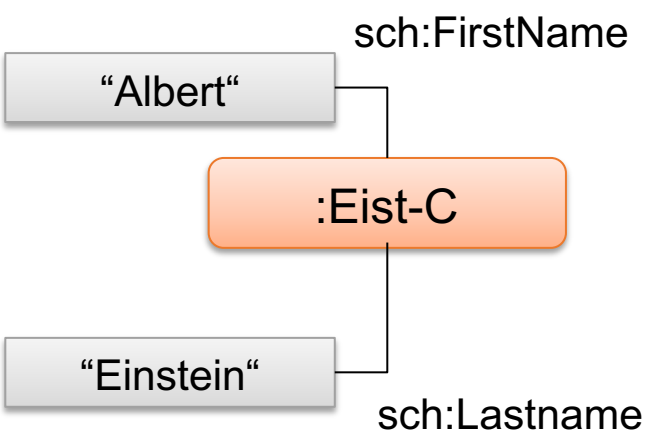
- How to change the current state of the art?
  - Allow expressing link rules with SPARQL
  - Generate the links among online datasources (through their SPARQL endpoints)
  - Provide this functionality through a web service easy to use or deploy

# SPARQL-based link rules

- In previous works we have already implemented the link rules as SPARQL queries

# SPARQL-based link rules




```
CONSTRUCT {
  ?scientist owl:sameAs ?person .
} WHERE {

    SERVICE <https://es.dbpedia.org/sparql> {
    ?scientist  rdf:type  foaf:Person ;
                foaf:name  ?fullName1 .
    }
    SERVICE <https://query.wikidata.org/#> {
      ?person  sch:fistName  ?firstName .
      ?person  sch:lastName  ?lastName .
      BIND ( CONCAT(?firstName, ?lastName) AS ?fullName2 ) .
    }

    FILTER ( ?fullName1 = ?fullName2 )
}
```

# Anatomy of a link rule

```
CONSTRUCT {
  ?scientist owl:sameAs ?person .
} WHERE {

   SERVICE <https://es.dbpedia.org/sparql> {
     ?scientist  rdf:type  foaf:Person ;
                   foaf:name  ?fullName1 .
   }
   SERVICE <https://query.wikidata.org/#> {
     ?person  sch:fistName  ?firstName .
     ?person  sch:lastName  ?lastName .
     BIND ( CONCAT(?firstName, ?lastName) AS ?fullName2 )
   }
   FILTER ( ?fullName1 = ?fullName2 )
}
```

- Datasets
- Restrictions over RDF resources & Data retrieval
- Transformation functions
- Link condition with similarity functions
- Link specification

# A quick sample

- https://es.dbpedia.org/sparql

```
CONSTRUCT {
  ?city1 owl:sameAs ?city2 .
} WHERE {
    ?city1 foaf:name  ?fullName1 .
    ?city2 foaf:name  ?fullName1 .
    FILTER ( ?fullName1 = ?fullName2 )
    VALUES ?city1 {
              <http://dbpedia.org/resource/Madrid>
          <http://dbpedia.org/resource/Soria>
          <http://es.dbpedia.org/resource/Sevilla>
      }
    VALUES ?city2 {
              <http://dbpedia.org/resource/Madrid>
          <http://dbpedia.org/resource/Soria>
      }
}
```

# The proposal

- Develop a web serivice that allows writting and running these SPARQL-based link rules

# Roadmap

1. Review the state of the art for RDF linking or link discovery
2. Develop a service that accepts SPARQL queries and runs them
   - For the specification use the standard https://www.w3.org/TR/sparql11-overview/
   - Use YASQUE/YASGUI for Web interface
     - E.g.: https://github.com/oeg-upm/helio-publisher/blob/master/src/main/resources/templates/sparql.html
   - Use Jena for the SPARQL processor
     - https://jena.apache.org/
   - Use https://sparkjava.com/ for the service

- Test the implementation with the query

```
CONSTRUCT {
  ?city1 owl:sameAs ?city2 .
} WHERE {
      SERVICE <https://es.dbpedia.org/sparql> {
      ?city1 foaf:name  ?fullName1 .
      VALUES ?city1 {
              <http://dbpedia.org/resource/Madrid>
                          <http://dbpedia.org/resource/Soria>
                          <http://es.dbpedia.org/resource/Sevilla>
          }
   }
      SERVICE <https://es.dbpedia.org/sparql> {
       ?city2 foaf:name  ?fullName1 .
         VALUES ?city2 {
              <http://dbpedia.org/resource/Madrid>
                          <http://dbpedia.org/resoAZurce/Soria>
           }
    }
    FILTER ( ?fullName1 = ?fullName2 )
}
```

# Roadmap

3. Add linking functions as Jena ARQ extensions to the implementation
   - Import functions from https://github.com/AndreaCimminoArriaga/EvA4LD/tree/master/tdg.link_discovery.connector.sparql/tdg/link_discovery/connector/sparql/evaluator/arq/linker/string_similarities

- Build a set of HTML views to assist users for writing the link rules
- Compare the time required by our proposal for linking two datasets with Limes or Silk for the same datasets.

# Considerations during implementation

- The service may require a lot of time for generating the links → this will be an issue with the time out
  - A possible solution would be to return instead of a time out error another response. Bulk in a file the links, provide in the response an access to such file (URL) so user can access. The file must be writting using a stream (FileWriter or ByteArrayOutputStream)
- Ad-hoc code for handling RDF or SPARQL queries must be avoided. The service must rely on mechanisms provided by the third party libraries
- Use the OEG github repository to backup the code

# Biobliography


- Nentwig, M., Hartung, M., Ngonga Ngomo, A. C., & Rahm, E. (2017). A survey of current link discovery frameworks. Semantic Web, 8(3), 419-436.