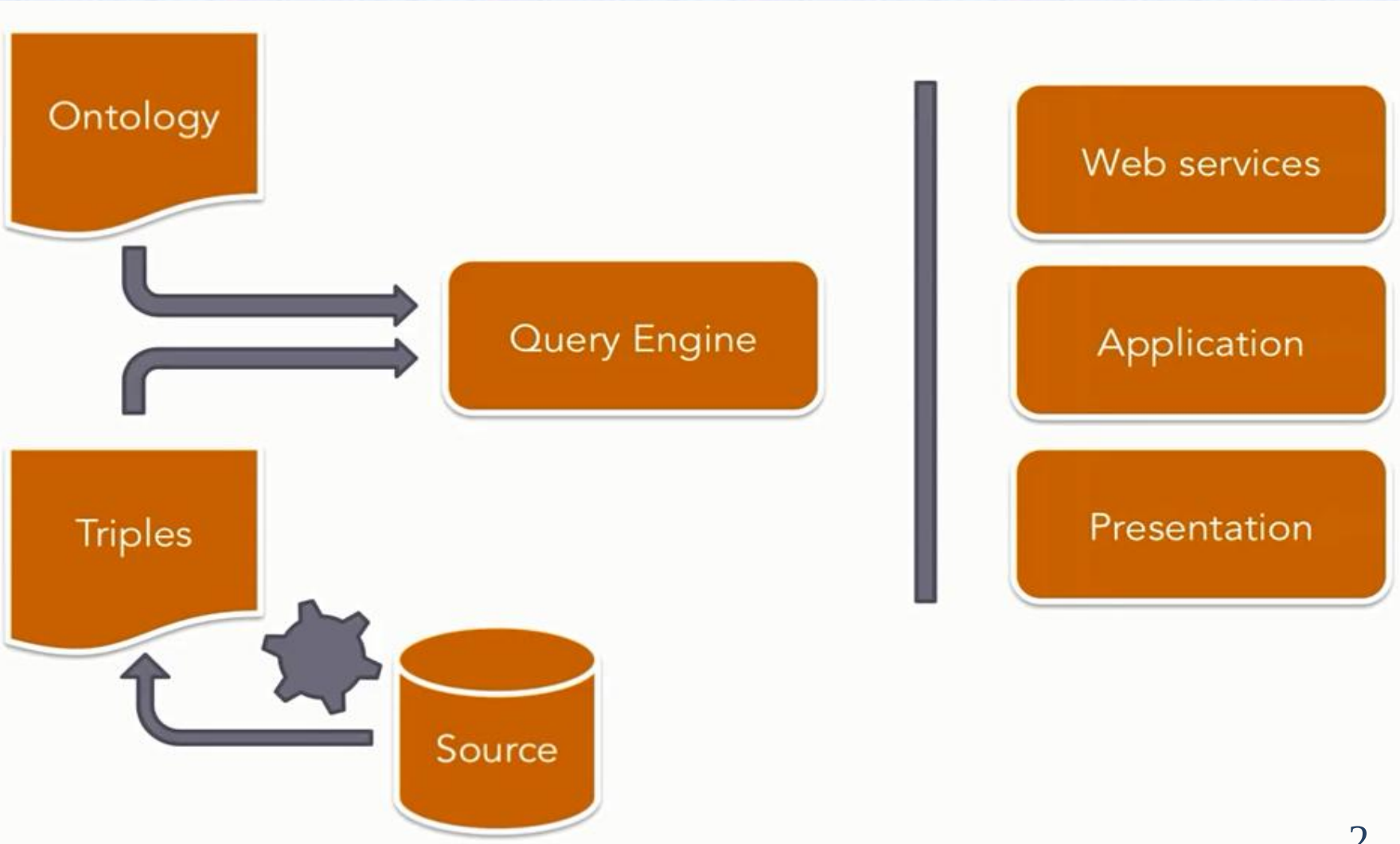
The Ontop Framework for Ontology Based Data Access

T. Bagosi, D. Calvanese, J. Hardi,
S. Komla-Ebri, D. Lanti, M. Rezk,
M. Rodríguez-Muro, M. Slusnys, G. Xiao

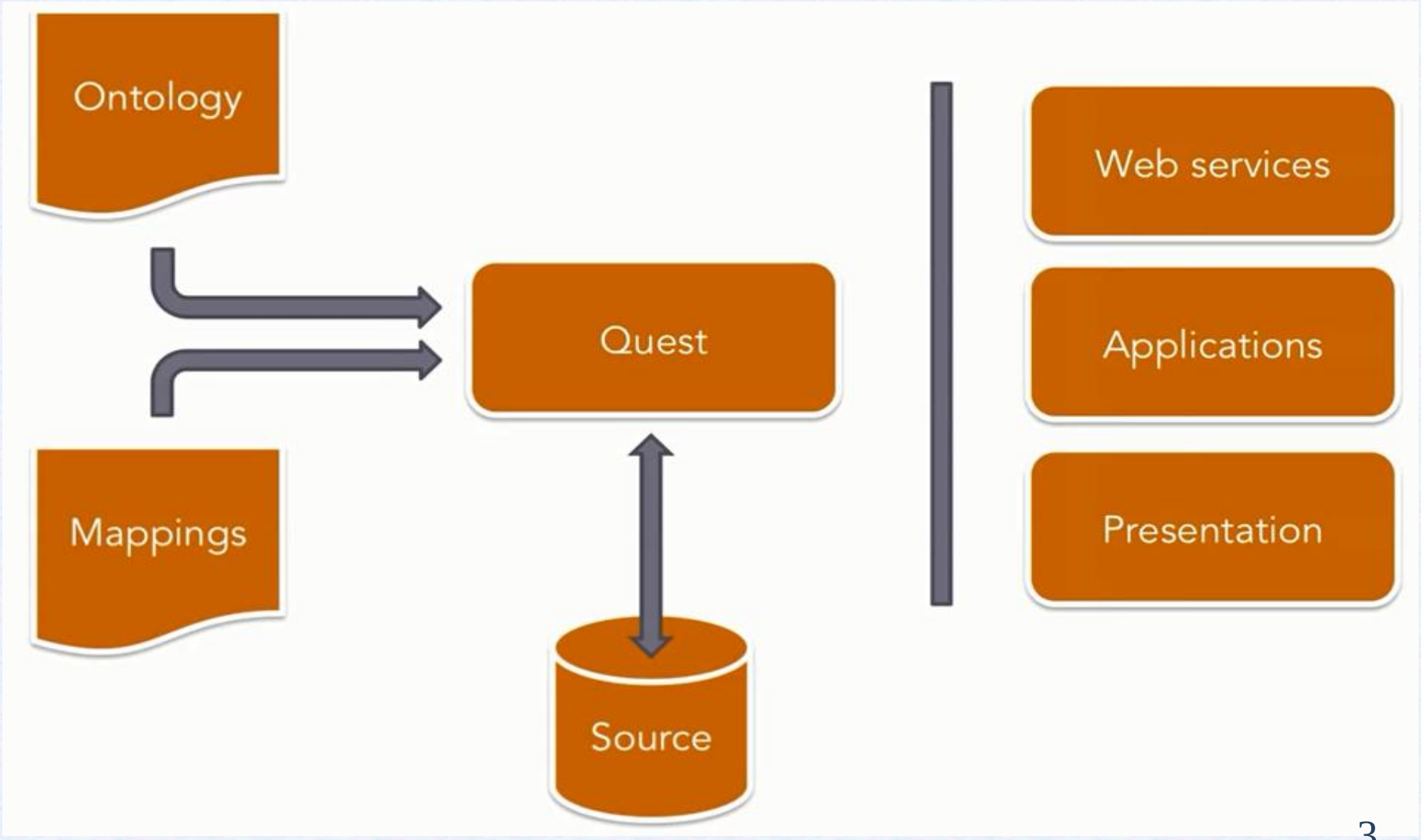
Presented by
Devin Rotondo

Ontology Based Data Access (OBDA)



https://www.youtube.com/watch?feature=player_embedded&v=KHtlARfex4c Quest ISWC Demo

Quest



https://www.youtube.com/watch?feature=player_embedded&v=KHtlARfex4c Quest ISWC Demo

Ontology Based Data Access

- OBDA systems are used for query answering
- The conceptual layer is expressed as RDF(S) or OWL ontology
- The data is stored inside relational databases

Ontology Based Data Access

An OBDA system is a triple $O = \langle \mathcal{T}, S, \mathcal{M} \rangle$

- $\mathcal{T}$ is the intensional level of an ontology (TBox)
- S is a relational database representing the sources
- $\mathcal{M}$ is a set of mapping assertions

Mappings

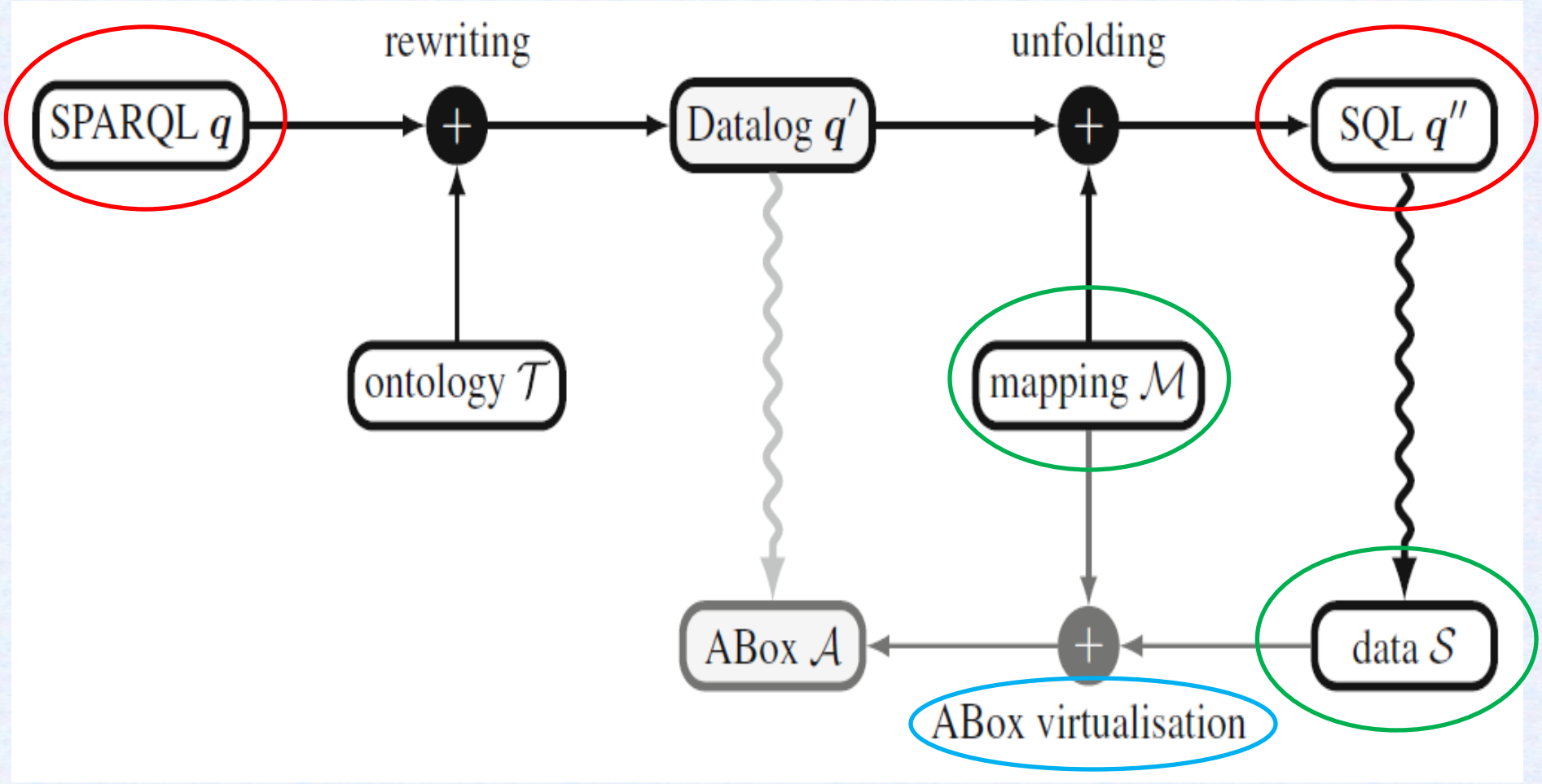
A mapping is a set of assertions of the form

$$\phi(x) \leftarrow \psi(x)$$

$\phi(x)$ is a query over S

$\psi(x)$ is a query over $\mathcal{T}$ with free variable x

Query Processing in OBDA



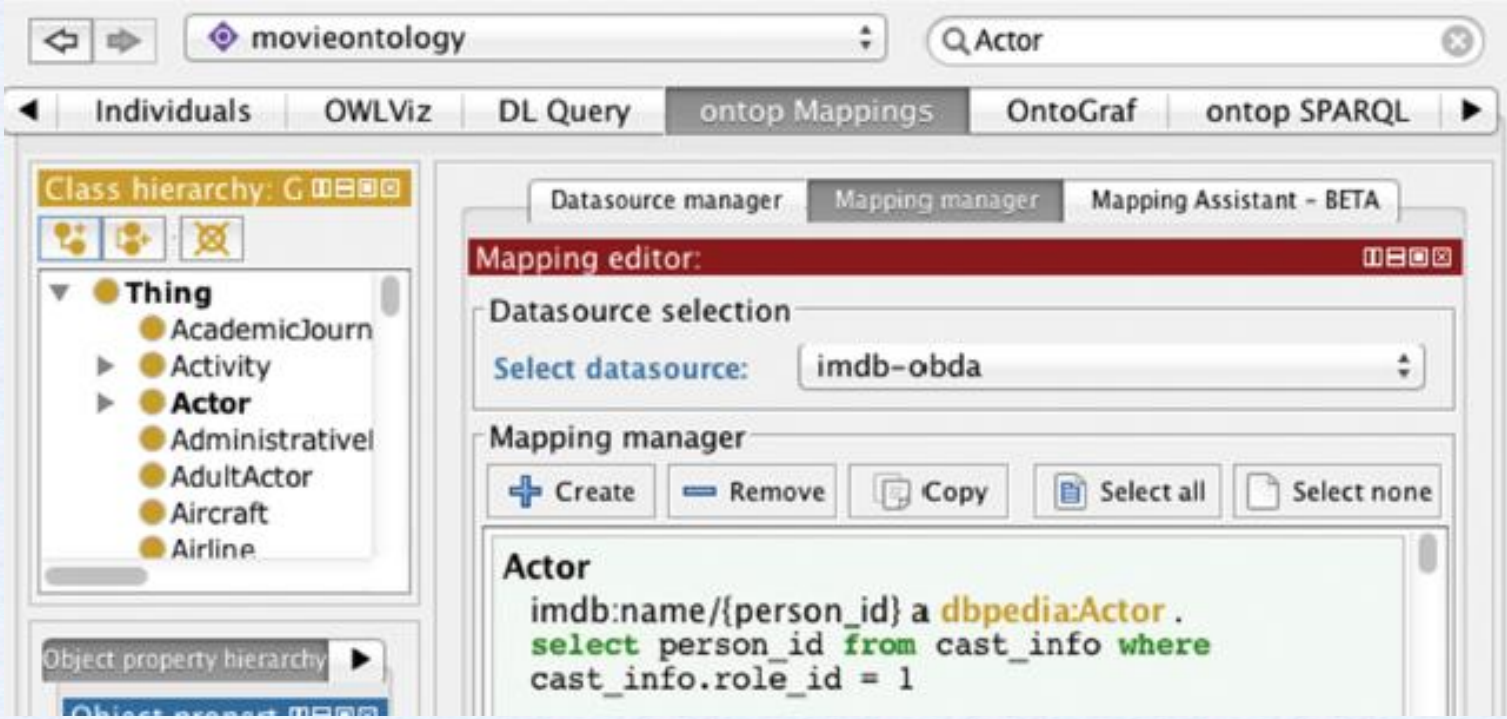
The Ontop Framework

- Ontop is an open-source OBDA framework developed at the Free University of Bolzen-Bolzano
- Ontop supports: OWL, SPARQL, SPARQL OWL 2 QL regime, R2RML, SWRL (W3C)
- Ontop supports all databases implementing SQL 99 (SQL 3) - including PostgreSQL, MySQL, etc.

??? SQL 2003, SQL 2006, SQL 2008, SQL 2011 ???

Supported in Ontop

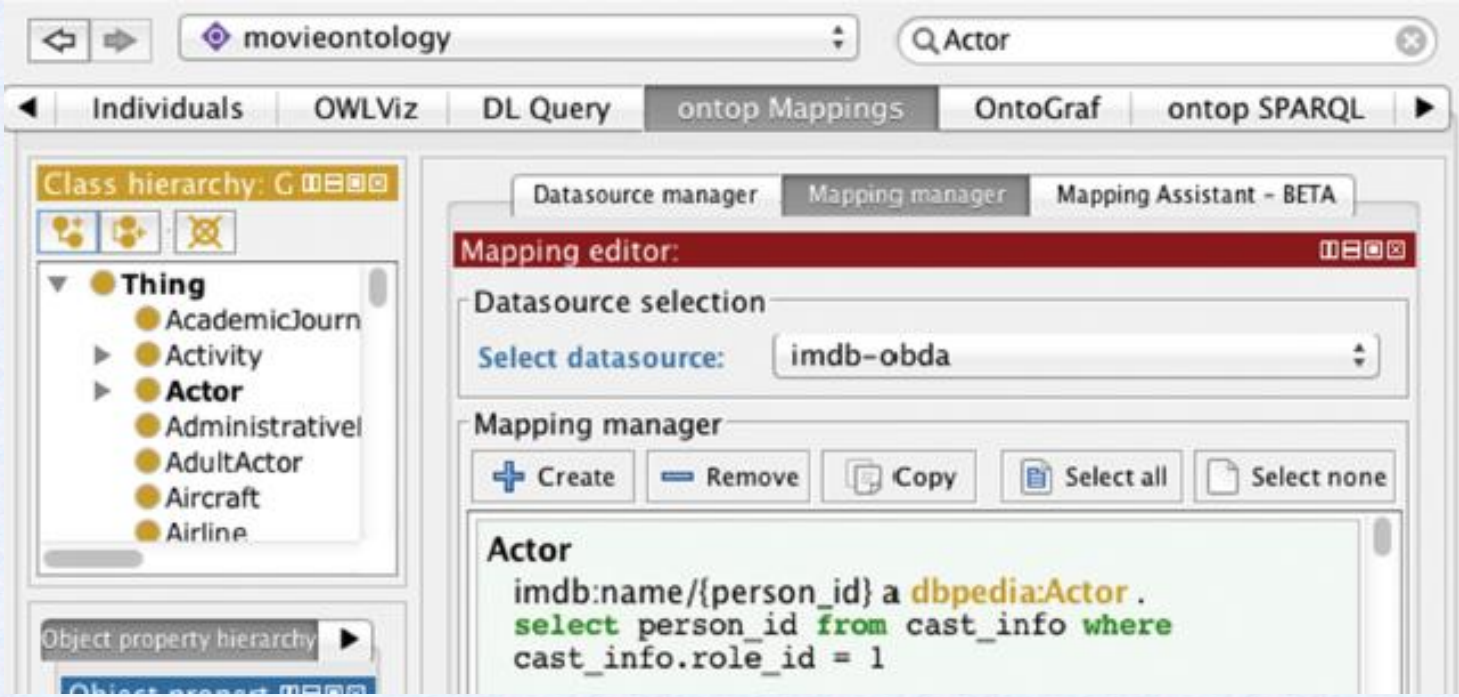
- **Mapping** supported:
  - The native Ontop mapping language
  - RDB2RDF Mapping Language (R2RML – W3C)



Mapping

- The **Ontop mapping language**

  Natural mappings that associate the data in the SQL database to the onlogy's vocabulary



Mapping

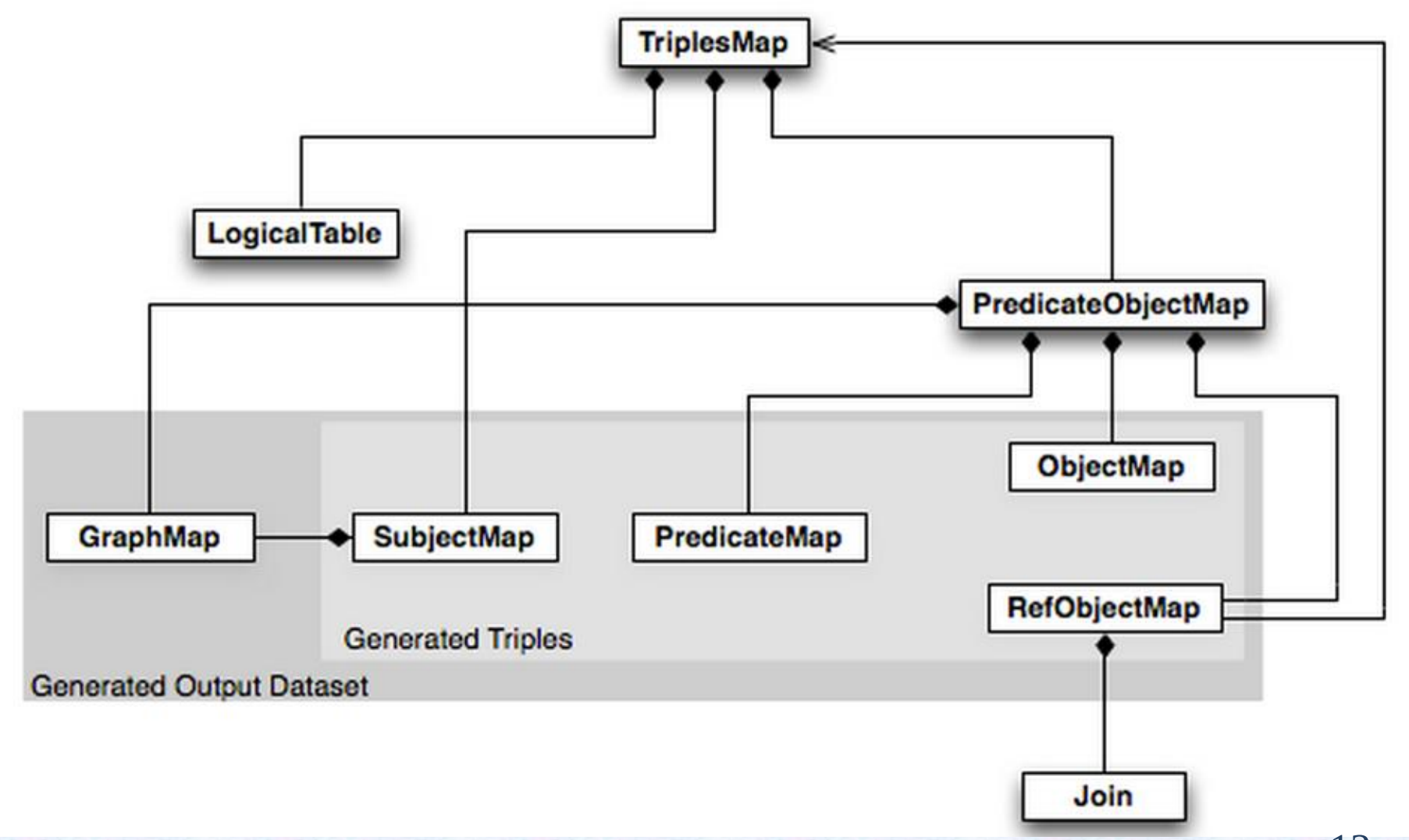
- The **RDB2RDF Mapping Language** (R2RML)
  - R2ML is a mapping language that allows expressing customized mapping from relational databases to RDF
  - Every R2RML mapping is tailored to a specific database schema and target vocabulary

Input: Relational database that conforms to the specified schema

Output: RDF dataset using predicates and types from the target vocabulary

http://www.w3.org/TR/r2rml/

R2RML Mapping Language



http://www.w3.org/TR/r2rml/

Supported in Ontop

- **Ontology**:
  - Fully support OWL QL ontology language
  - Supports linear recursive fragments of SWRL (semantic web language – W3C)

Ontology

- Fully supports **OWL QL** ontology language
  - It is based on the DL-Lite family of description logics
  - DL-Lite_R provides the logical support for OWL 2 QL
  - Query answering in LOGSPACE (AC^0) with respect to the size of the data
  - Can also have DL-Lite_A (unique name assumption) however this will also increase the complexity

http://www.w3.org/TR/owl2-profiles/

Ontology

- Supports **linear recursive fragments** of **SWRL**
  - SWRL is a widely used Semantic Web language
  - SWRL combines a DL ontology component with rules
  - Allows for only unary and binary predicates
  - Support only recursive predicates which are restricted to linear recursive

http://www.w3.org/Submission/SWRL/

Supported in Ontop

- **Query**:
  - Supports SPARQL 1.0
  - Supports SPARQL OWL QL Regime of SPARQL 1.1

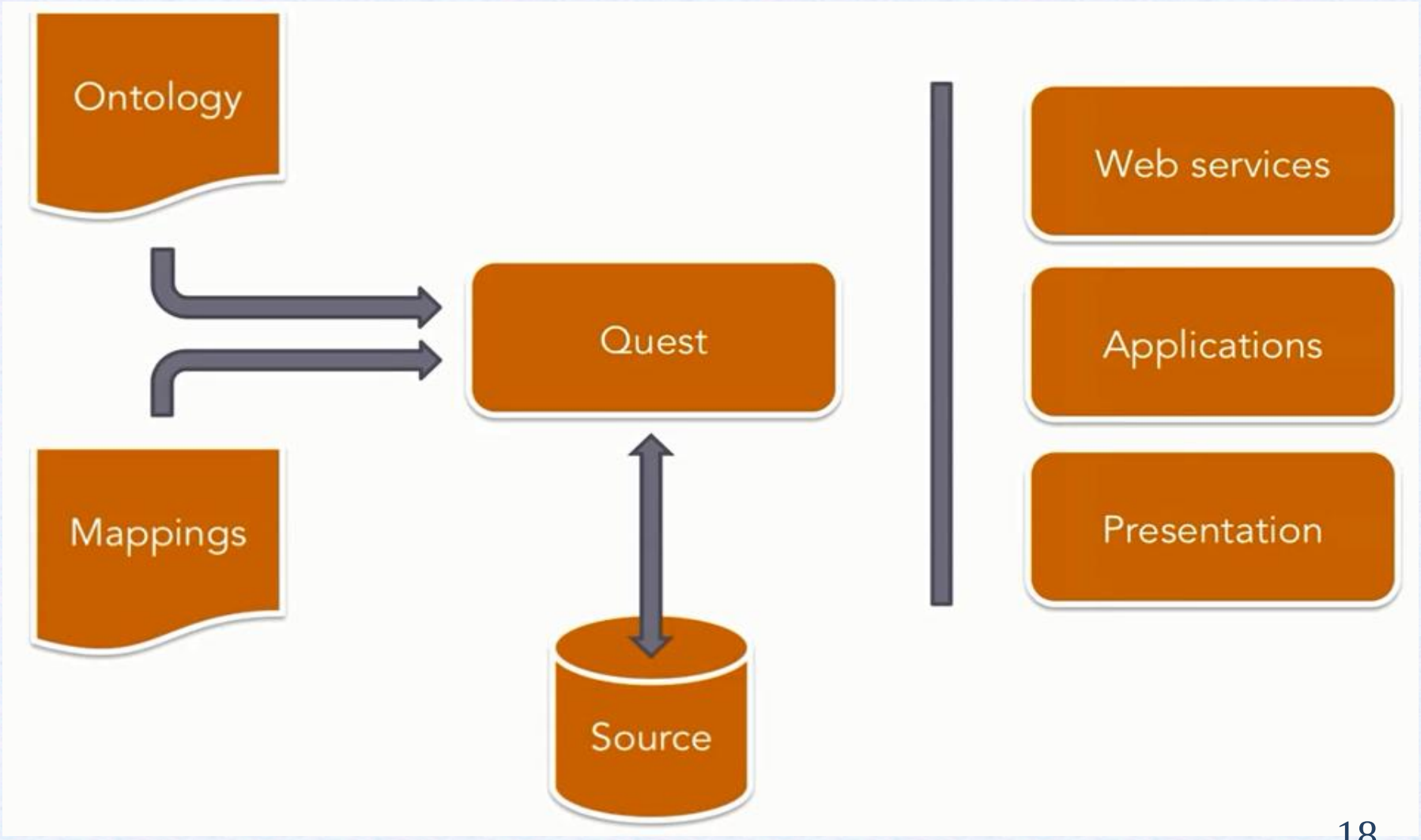
  We have already gone quite in depth about SPARQL in class

  Only OWL QL Regime is supported from OWL 1.1 - likely due to complexity

About Ontop

- The core of Ontop is the SPARQL engine Quest
- Quest supports RDFS and OWL 2 QL entailment regimes (rewriting SPARQL queries to SQL queries)
- Ontop can generate efficient (highly optimized) SQL queries

Quest



https://www.youtube.com/watch?feature=player_embedded&v=KHtlARfex4c Quest ISWC Demo

Components of Ontop

- Ontop uses Maven (Apache developed) for managing dependency (build automation - java)
- Ontop requires a JDBC driver to connect to a given database (java)
- Main component is the Ontop reasoner
  - Requires ontology file
  - OBDA file of mappings and database settings

Queries (java)

```
// Loading the query file
QueryController qc = new QueryController();
QueryIOManager qman = new QueryIOManager(qc);
qman.load("src/main/resources/example/movie/movieontology.q");

// Execute each query
for (QueryControllerGroup group : qc.getGroups()) {
    for (QueryControllerQuery query : group.getQueries()) {

        System.out.println("Executing query: " + query.getID());
        System.out.println("Query: \n" + query.getQuery());

        long start = System.nanoTime();
        QuestOWLResultSet res = st.executeTuple(query.getQuery());
        long end = System.nanoTime();
        double time = (end - start) / 1000;
        int count = 0;
        while (res.nextRow()) {
            count += 1;
        }

        System.out.println("Total result: " + count);
        System.out.println("Elapsed time: " + time + " ms");
    }
}
```

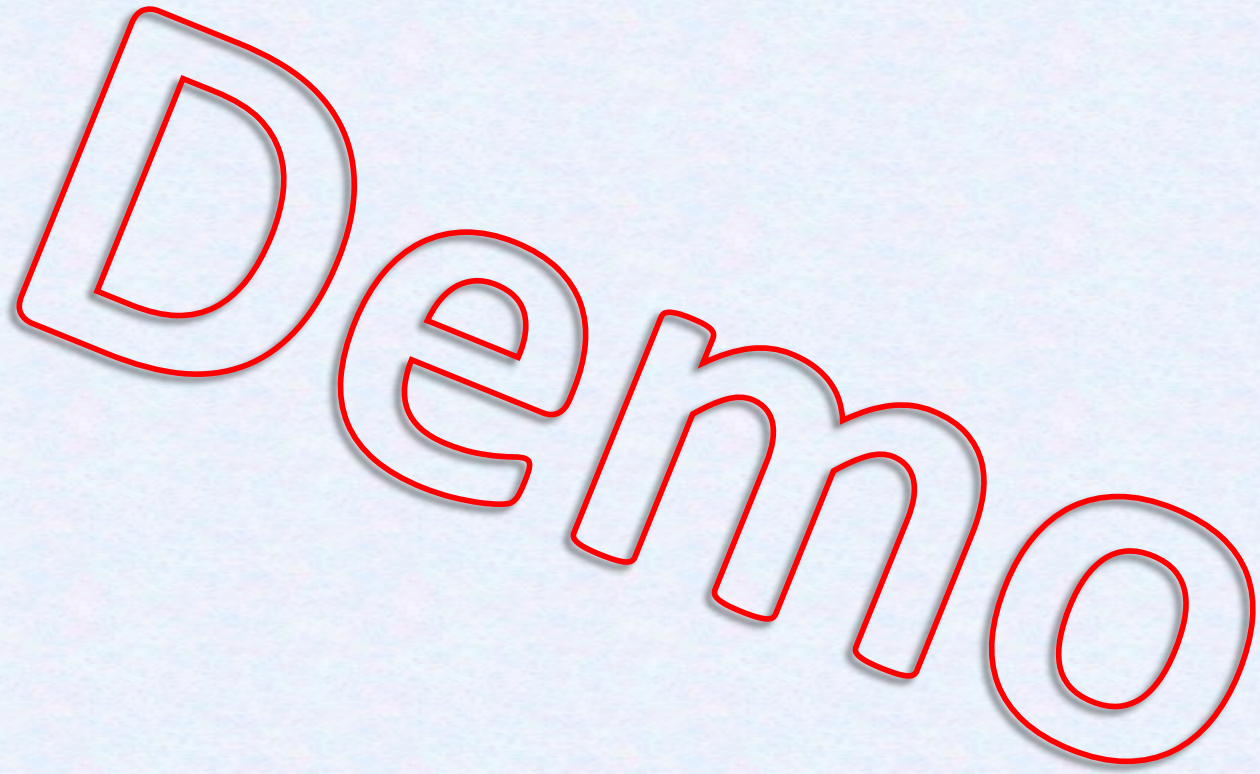
Queries (**Sesame**)

```
// Executing the query
Query query = conn.prepareQuery(QueryLanguage.SPARQL, queryString);

TupleQuery tq = (TupleQuery) query;

TupleQueryResult result = tq.evaluate();

while (result.hasNext()) {
    for (Binding binding : result.next()) {
            System.out.print(binding.getValue() + ", ");
    }
    System.out.println();
}
```


Demo

Conclusion

- Ontop in combination with Quest solve several of the previous problems with OBDA systems
- It allows for the use of SPARQL queries instead of SQL queries (Mastro)
- Seamless query rewriting and optimization (close to expert quality)

Discussion

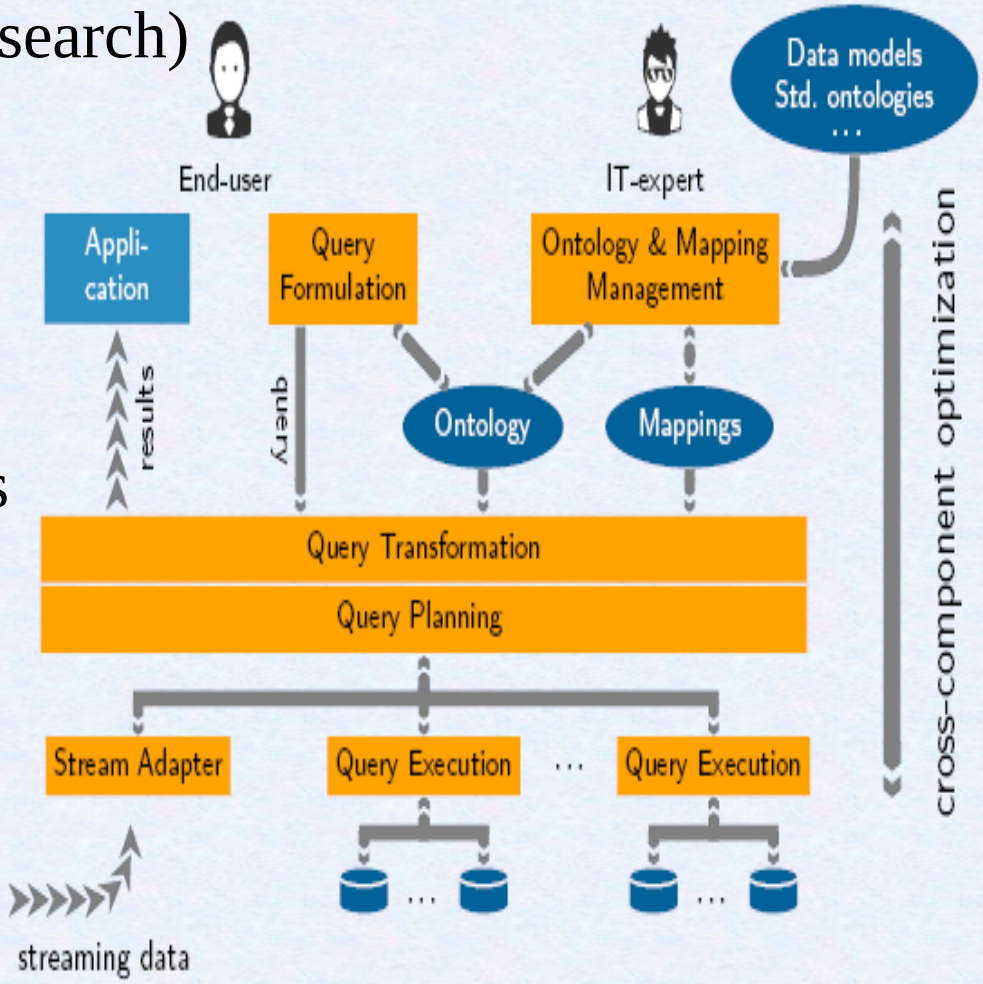
- Effect of the open world to closed world assumption in OBDA

- Can we consider Ontop expressive enough, or has its goal of remaining low computational complexity reduced its expressiveness too much

Discussion

Optique (Funded Ontop research)

- Development of novel and performance-critical key modules
- Provide early tool prototypes
- Maximize reusability of components developed in Optique



http://www.optique-project.eu/about-optique/research-topics-2/