Publishing OWL ontologies with Presto

Alexander De Leon[1] and Michel Dumontier[1,2]
1 School of Computer Science
2 Department of Biology
Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, K1S5B6 Canada

Presented @ **OWLED 2008**
Washington, DC.
April 2008

Motivation

An essential aspect of the Semantic Web is to ensure that the terminology defined in ontologies are web-accessible such that information about the ontological entity may be discovered and links with related entities explored. This idea is realized by the *Linked Data* architecture.

In practice, how can we make ontologies and their components (classes, properties, individuals) available on the web so that others can link to them and perform queries ?

Overview

Presto is a tool for publishing and querying OWL ontologies on the Semantic Web.

For a given ontology, Presto provides the following:

1. A self-referential namespace for all ontological documents and entities, so as to follow linked knowledge as a static ontological snapshot.

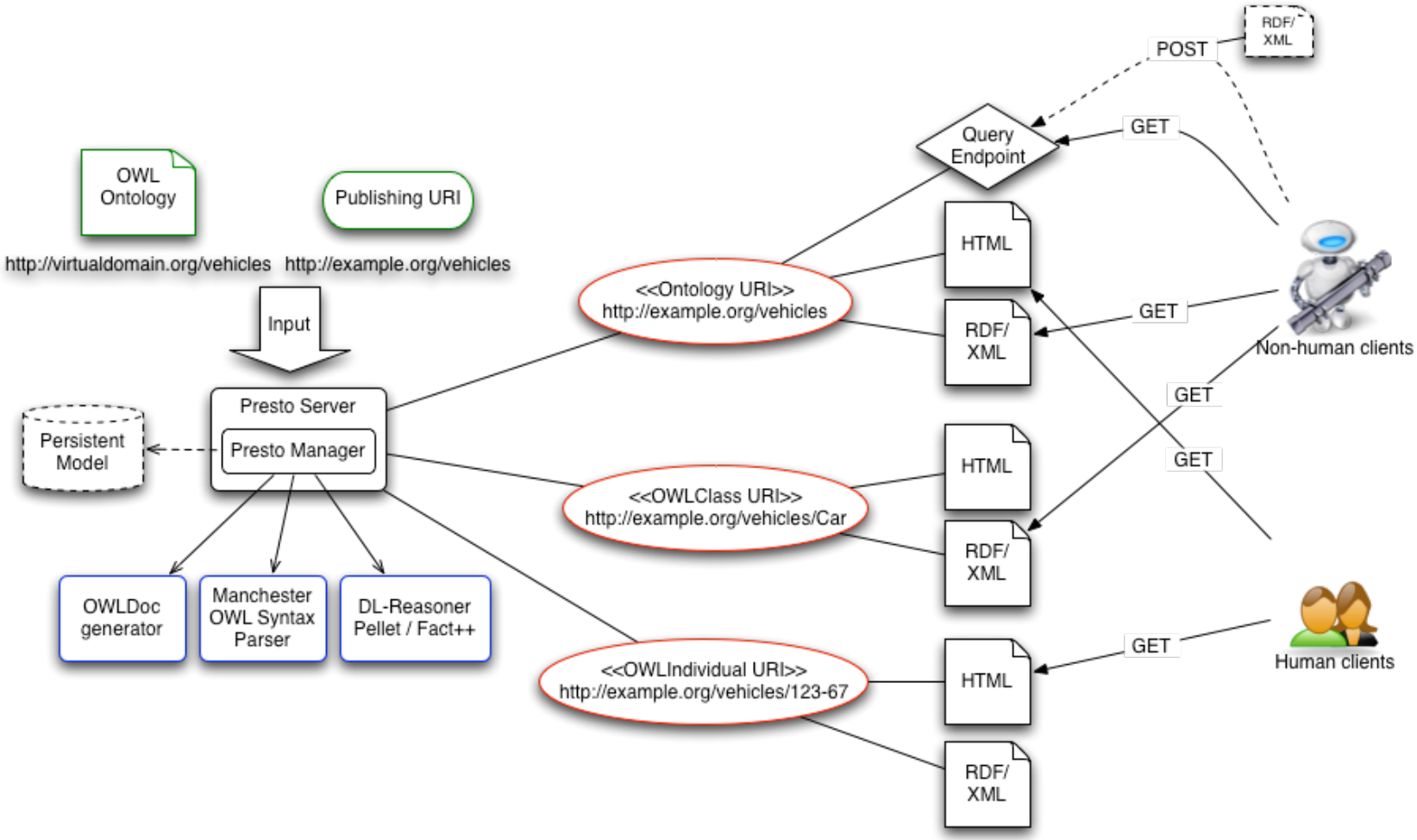
2. A RESTful service for DL and SPARQL queries that are identified by permanent HTTP URIs.

3. Content-negotiation capabilities to retrieve dynamically generated HTML or RDF/XML.

Presto

Michel Dumontier
& Alexander De Leon

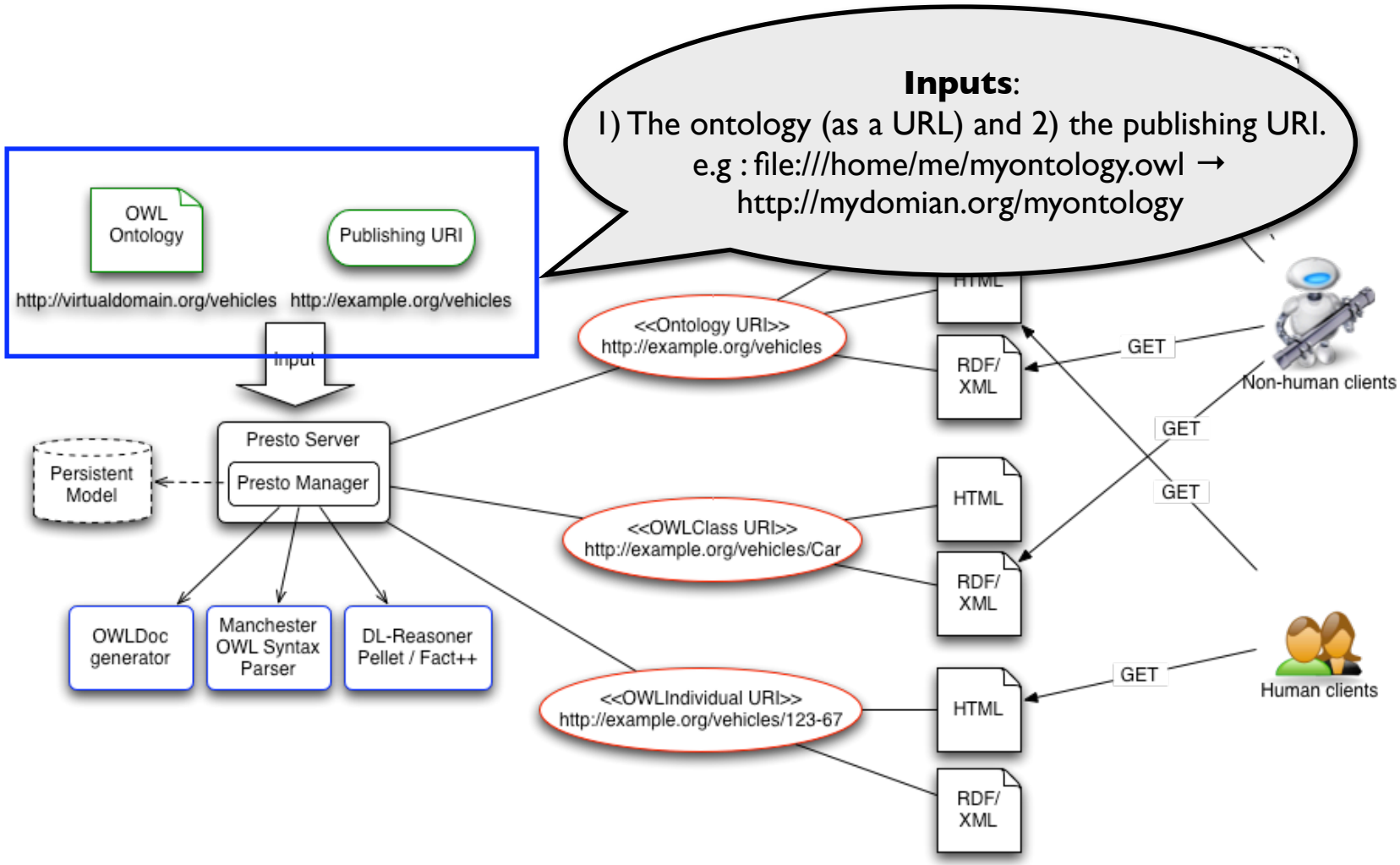
Overview



Presto

Michel Dumontier
& Alexander De Leon

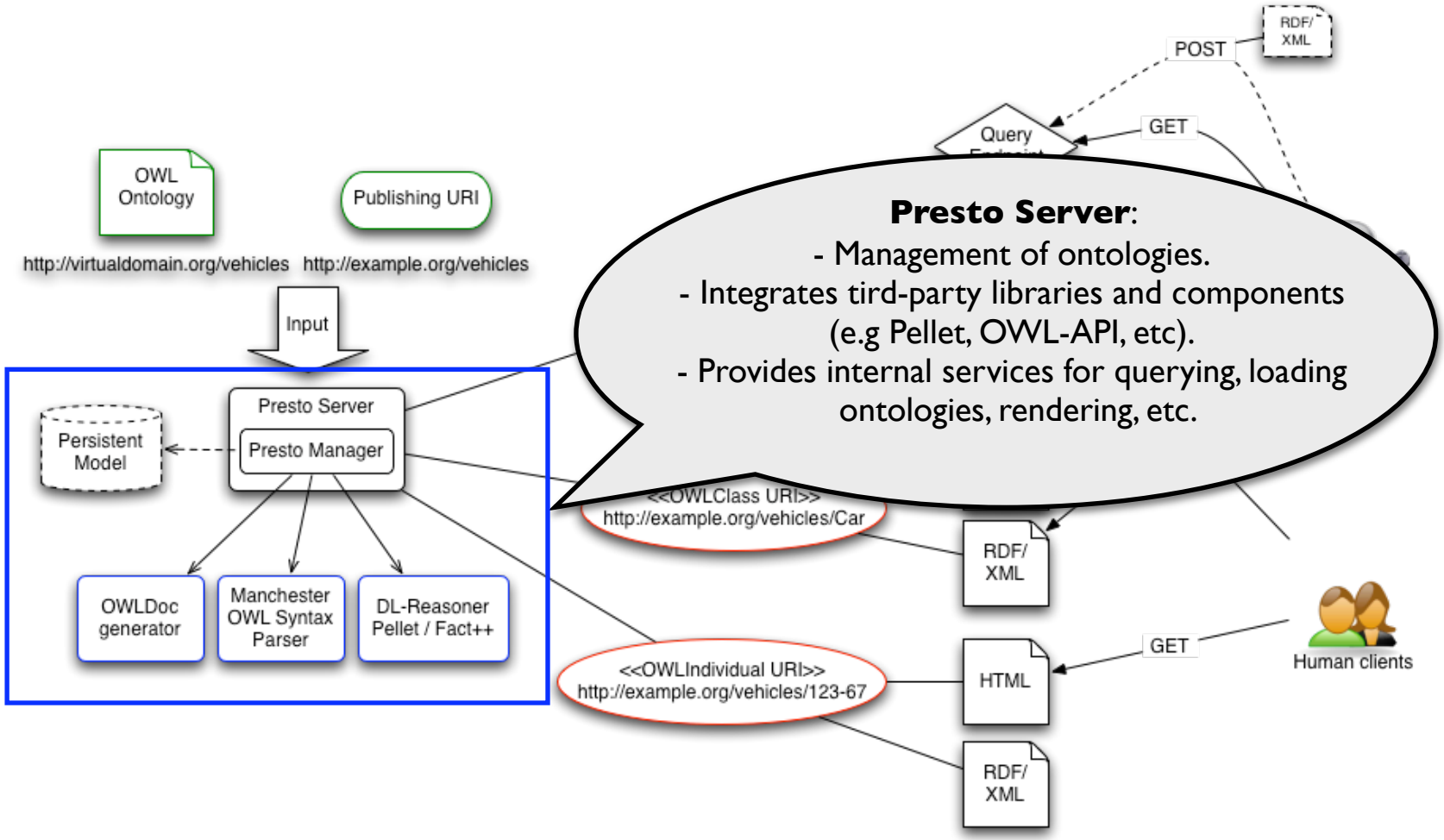
Overview



Presto

Michel Dumontier
& Alexander De Leon

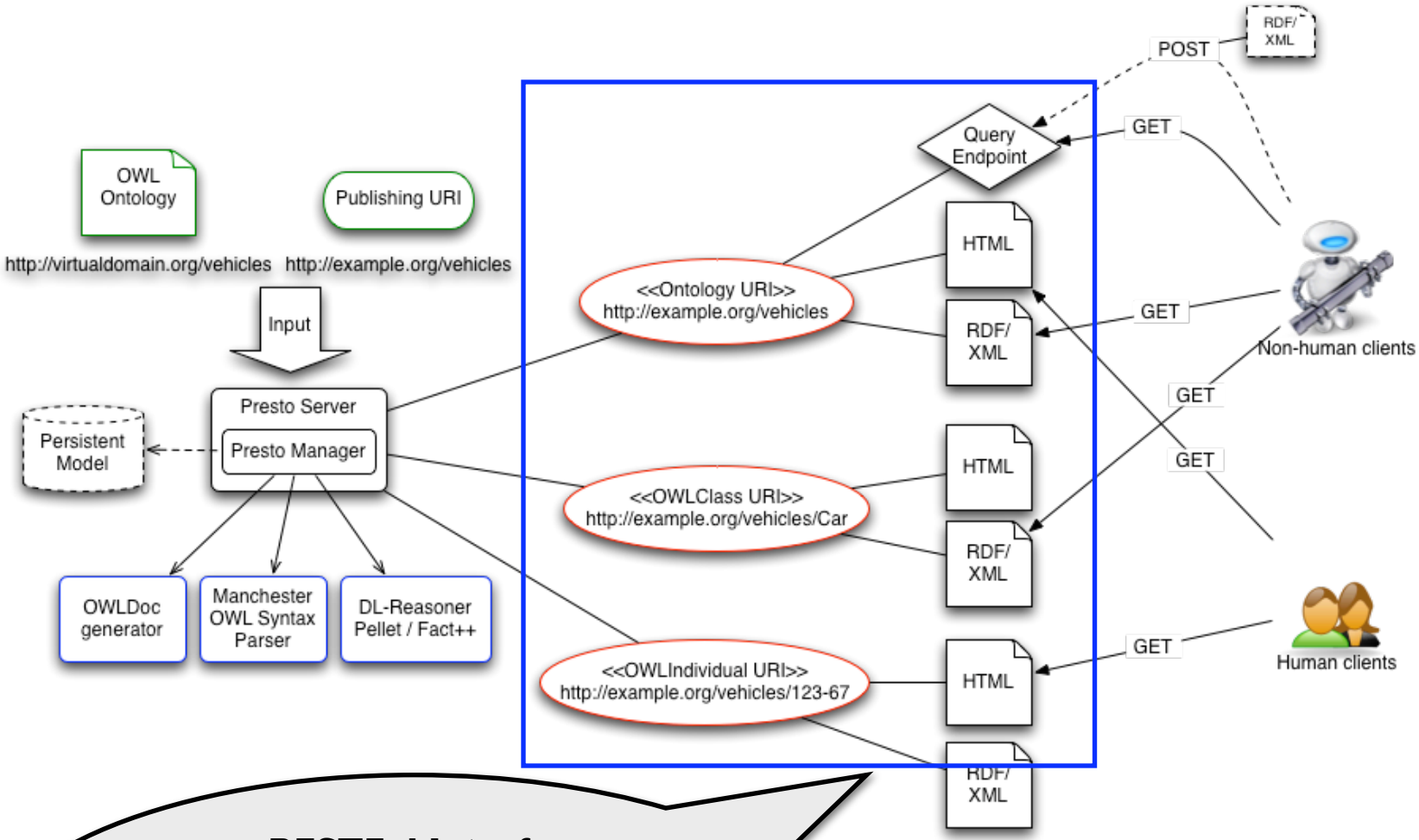
Overview



Presto

Michel Dumontier
& Alexander De Leon

Overview



RESTFul Interface

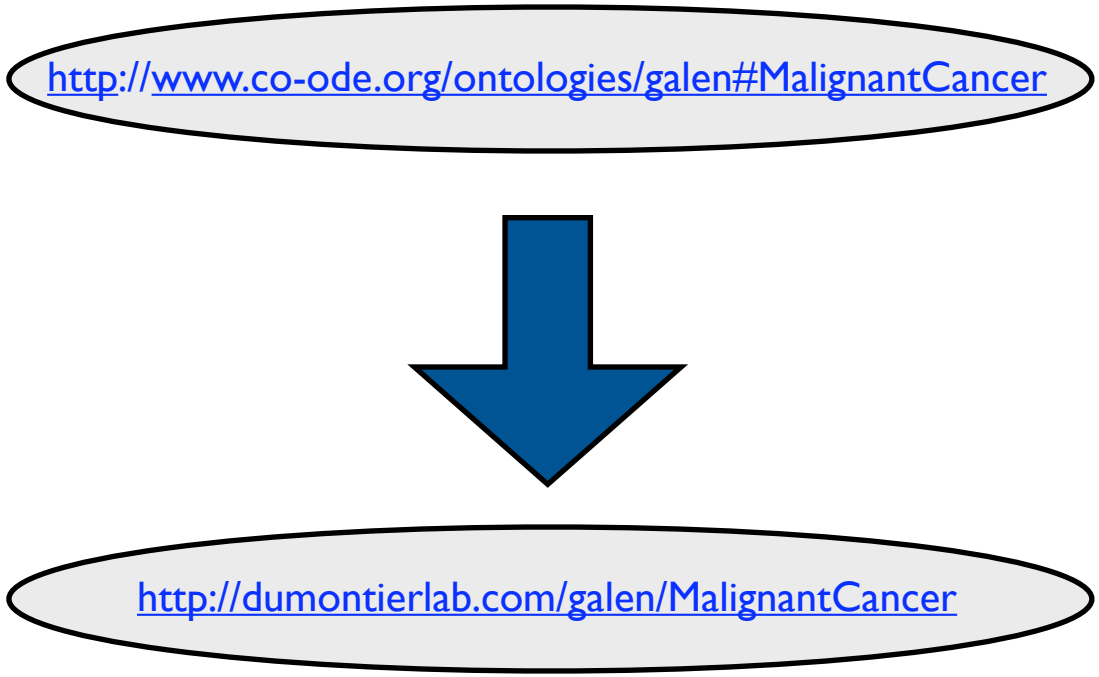
A representation of an ontology or any of its entities can be retrieved using a GET HTTP request on its deployed URI. Content negotiation determines the format of the representation (RDF/XML or HTML)

Carleton UNIVERSITY

Renaming

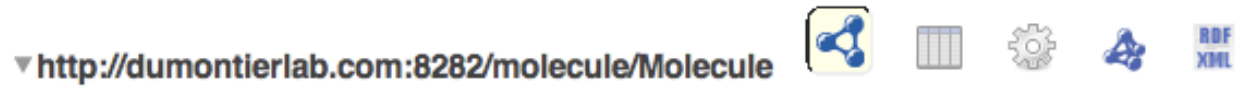
Presto replaces the URI of the input ontology, as well as, the URIs of all entities referred in the ontology.

Ontology : http://www.co-ode.org/galen/full-galen.owl
Publishing URI : http://dumontierlab.com/galen



Renaming

By rewriting the URIs we can assign HTTP resolvable URIs to each ontological entity.



http://dumontierlab.com:8282/molecule/Molecule

http://dumontierlab.com:8282/molecule/Molecule.owl	type	Ontology
	imports	Molecule Ontology (complex)
		http://dumontierlab.com:8282/molecule/Molecule.owl
		http://dumontierlab.com:8282/molecule/Source.owl
http://dumontierlab.com:8282/molecule/hasProperPart	type	ObjectProperty
http://dumontierlab.com:8282/molecule/Atom	type	Class
http://dumontierlab.com:8282/molecule/Molecule	type	Class
	subClassOf	http://dumontierlab.com:8282/molecule/Object
		type: Restriction
		onProperty: http://dumontierlab.com:8282/molecule/hasProperPart
		someValuesFrom: http://dumontierlab.com:8282/molecule/Atom
	disjointWith	http://dumontierlab.com:8282/molecule/Source
http://dumontierlab.com:8282/molecule/Object	type	Class
http://dumontierlab.com:8282/molecule/Source	type	Class

Querying

Ontologies can be queried by sending a HTTP GET request to the ontology URI with the mandatory ***query*** parameter. Queries can be formulated using SPARQL or Manchester OWL DL query.

Sample Query:

http://www.example.org/vehicles?query=Vehicle that hasPart some GasEngine

Querying

Sample Results:

```
<rdf:RDF ... >
<owl: Ontology>
    <owl:imports rdf:resource="http://www.example.org/vehicles" />
</owl:Ontology>

<!-- Query class expression -->
<owl:Class  rdf:about="&query;Vehicle%20and%20hasPart%20some%
20ManualTransmission" >
    <owl:intersectionOf rdf:parseType="Collection" >
        <rdf:Description rdf:about="Vehicle" />
        <owl:Restriction>
            <owl:onProperty rdf:resource="hasPart" />
            <owl:someValuesFrom rdf:resource="ManualTransmission" />
        </owl:Restriction>
    </owl:intersectionOf>
</owl:Class>
<!-- Individuals -->
<owl:Thing rdf:about="123-456" >
    <rdf:type rdf:resource="&query;Vehicle%20and%20hasPart%20some%
    20ManualTransmission" />
</owl:Thing>
...
</rdf:RDF>
```

Querying

Sample Results:

```
<rdf:RDF ... >
<owl: Ontology>
    <owl:imports rdf:resource="http://www.example.org/vehicl
</owl:Ontology>

<!-- Query class expression -->
<owl:Class  rdf:about="&query;Vehicle%20and%20hasPart%20some%
20ManualTransmission" >
    <owl:intersectionOf rdf:parseType="Collection" >
        <rdf:Description rdf:about="Vehicle" />
        <owl:Restriction>
            <owl:onProperty rdf:resource="hasPart" />
            <owl:someValuesFrom rdf:resource="ManualTransmission" />
        </owl:Restriction>
    </owl:intersectionOf>
</owl:Class>
<!-- Individuals -->
<owl:Thing rdf:about="123-456" >
    <rdf:type rdf:resource="&query;Vehicle%20and%20hasPart%20some%
    20ManualTransmission" />
</owl:Thing>
...
</rdf:RDF>
```

Query is express as a class.

Querying

Sample Results:

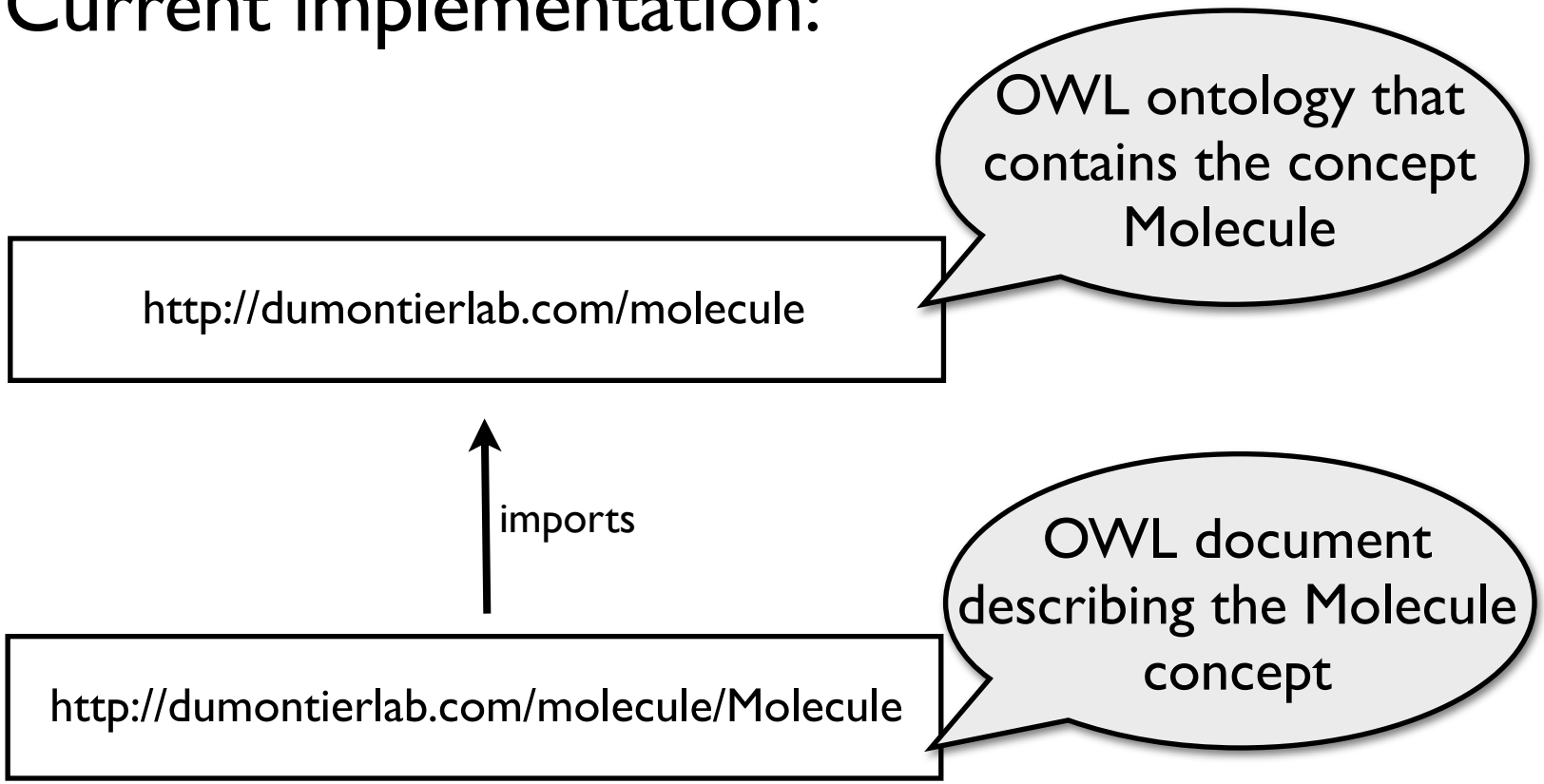
```
<rdf:RDF ... >
<owl: Ontology>
    <owl:imports rdf:resource="http://www.example.org/vehicles" />
</owl:Ontology>

<!-- Query class expression -->
<owl:Class  rdf:about="&query;Vehicle%20and%20hasPart%20some%
20ManualTransmission" >
    <owl:intersectionOf rdf:parseType="Collection" >
        <rdf:Description rdf:about="Vehicle" />
        <owl:Restriction>
            <owl:onProperty rdf:resource="hasPart" />
            <owl:someValuesFrom rdf:resource="ManualTr
        </owl:Restriction>
    </owl:intersectionOf>
</owl:Class>
<!-- Individuals -->
<owl:Thing rdf:about="123-456" >
    <rdf:type rdf:resource="&query;Vehicle%20and%20hasPart%20some%
    20ManualTransmission" />
</owl:Thing>
..
</rdf:RDF>
```

A-Box result asserted as instance of the query.

Future Directions: Modularization

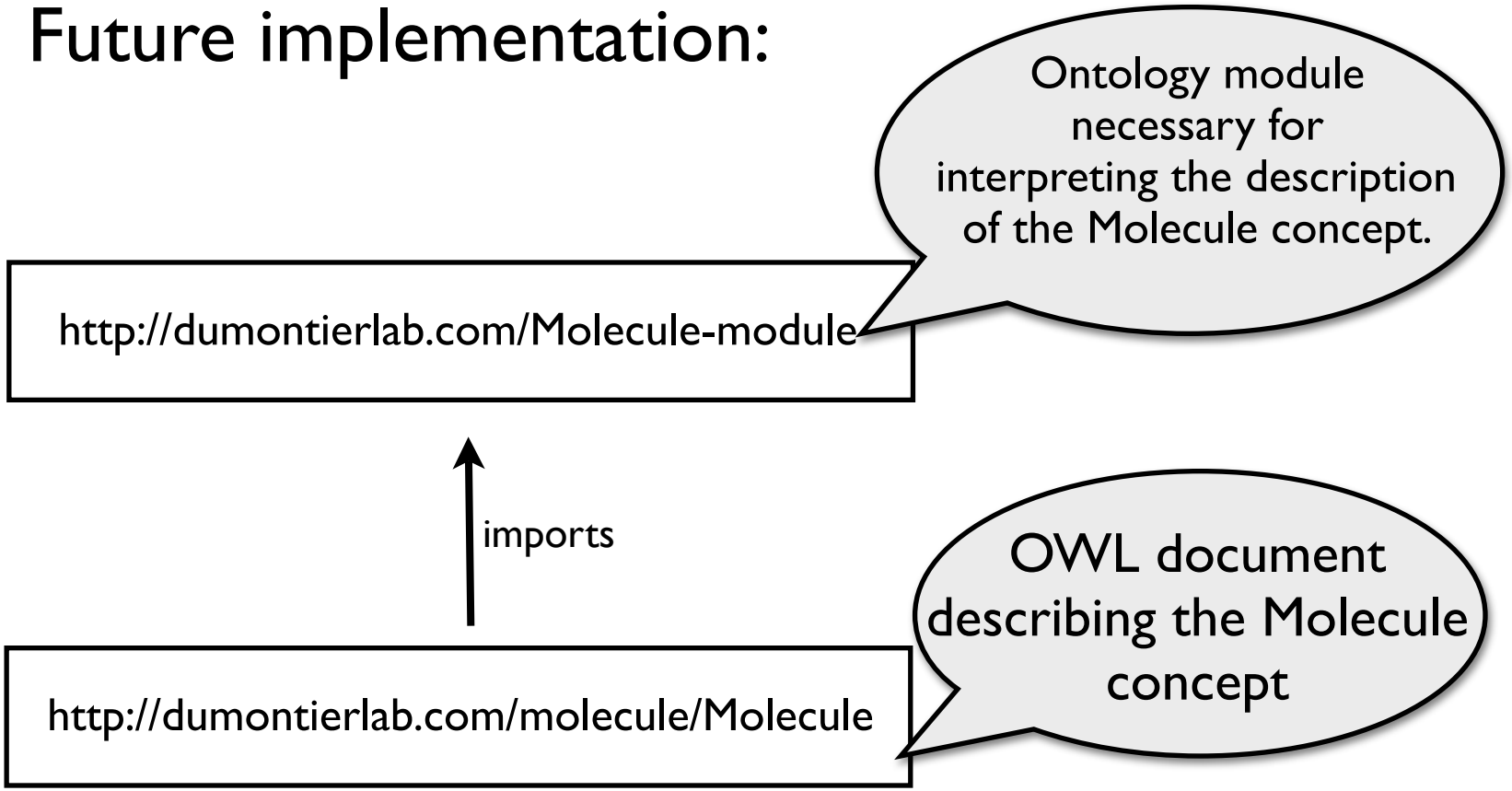
Current implementation:



Michel Dumontier
& Alexander De Leon

Future Directions: Modularization

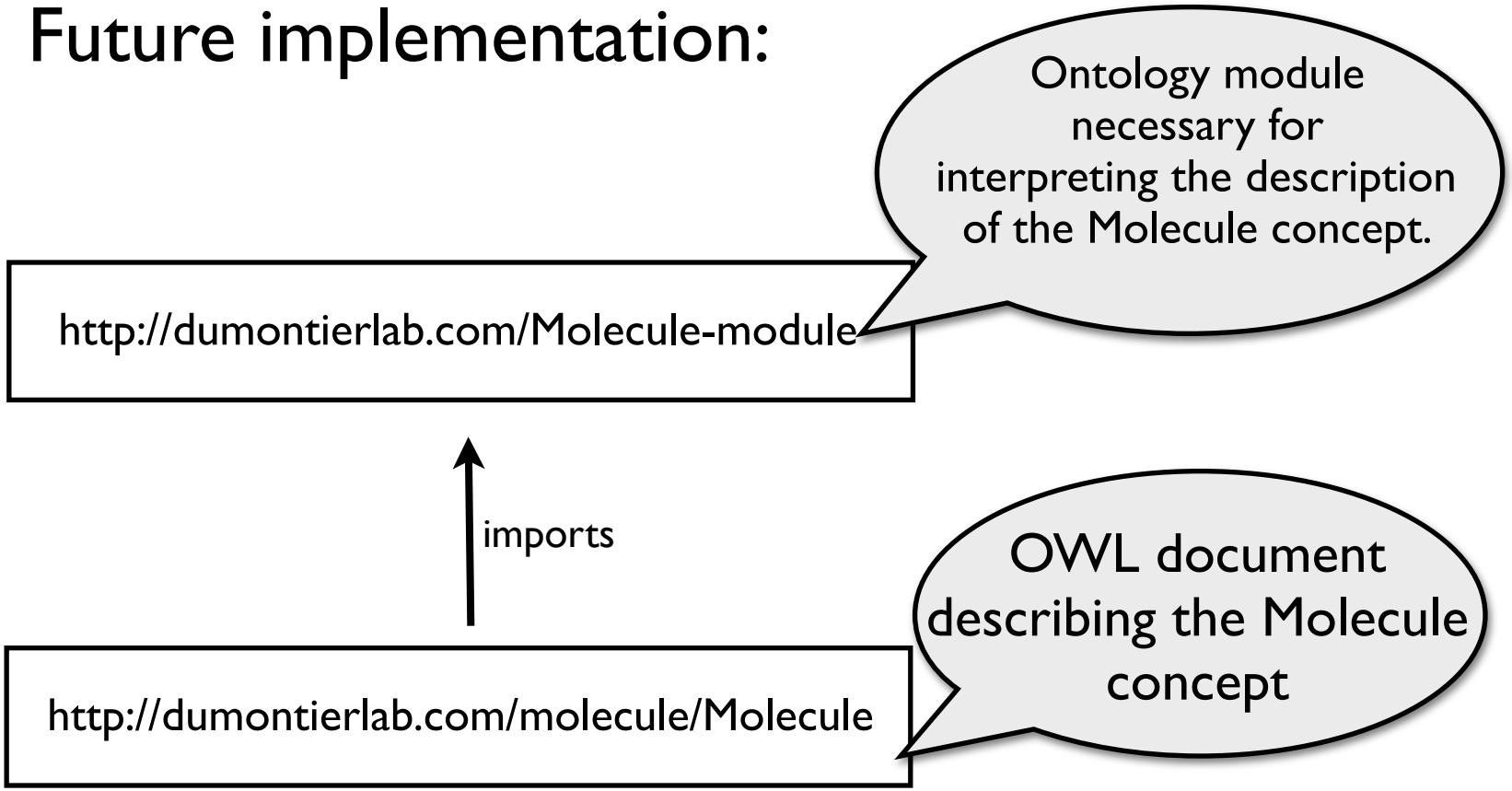
Future implementation:



Michel Dumontier
& Alexander De Leon

Future Directions: Modularization

Future implementation:



Presto

Michel Dumontier
& Alexander De Leon

- Motivation
- Overview
- Renaming
- Querying
- Future work:
  - Modularization
- Questions

Questions? ...