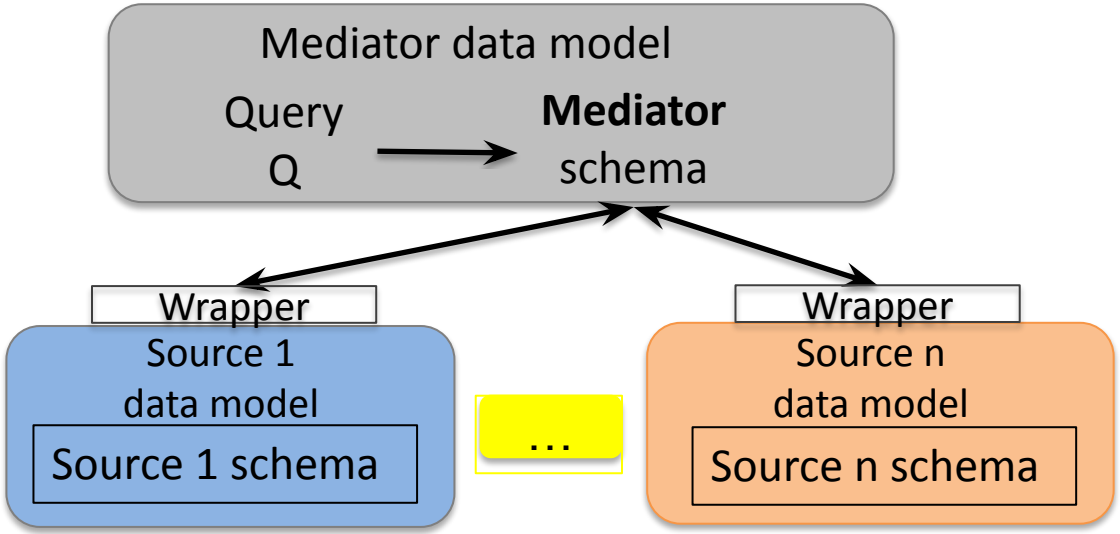
BIG GRAPH DATABASES
Lab: Obi-Wan

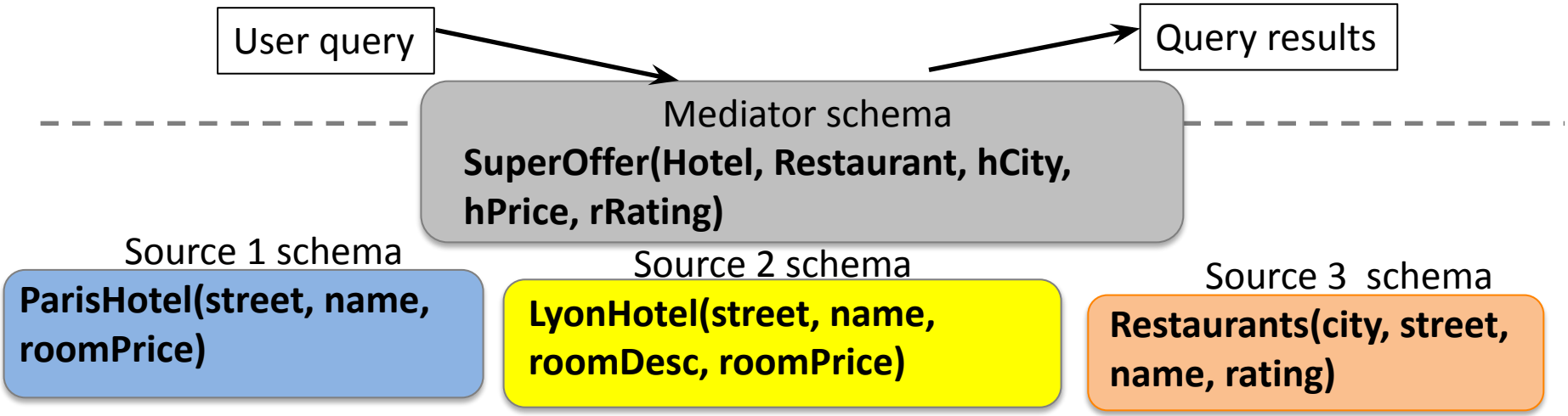
Garima Gaur
Inria and Ecole polytechnique
garima.gaur@inria.fr

Recap to Heterogenous DS Integration System (from Lecture)

- Key components of an integration system: data sources, mediator and wrappers



Global-Local-as-View: example (from the previous lecture)



New GLAV mapping:

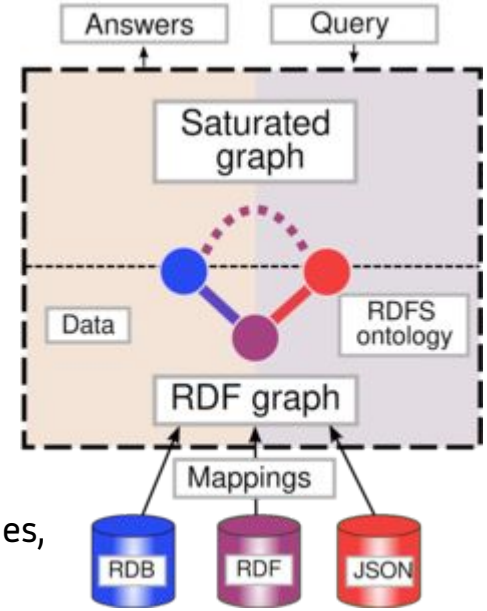
Q3Mediator: select * from SuperOffer where hCity='Lyon'

Q3Sources: select lh.name, r.name, lh.roomPrice * 0.5 as hPrice, r.rating as rRating
from LyonHotel lh, Restaurants r
where r.city='Lyon' and r.name='Lion d'Or' and r.street=lh.street

This mapping says: "each result of Q3Sources leads to a SuperOffer in Lyon".

Other mappings could define more SuperOffers in Lyon, or in other cities, or with rRating=3...

Obi-Wan: RDF Integration System



- An RDF integration systems that supports:
  - Global-Local-As-View mappings with an simple syntax
  - Supports RDFS ontologies
  - Supports BGP queries (SPARQL core)

- Each project contains the following files:
  - obi-wan.properties the file defines the query answering properties,
  - ris.json the file defines the RIS (mappings),
  - ontology.nt the RDFS ontology of the RIS in NT format,
  - querysession.properties the file needed for technical reasons related to the materialization approaches.

Mapping Format

- Each mapping contains a pair of query (q_m, q_s) where q_s is the query over the source data and q_m specifies the BGP that populates the integrated graph.
- In ris.json, each mapping is a key-value pair with 3 keys: name, head, body
- Head specifies the q_m and body specifies q_s.

```
{
  "name": "person",
  "head": [
    [ "$person", "<http://www.imdb.com/fullName>", "$fullName" ]
  ],
  "body": {
    "templates": {
      "fullName": "primaryName",
      "person": "<http://www.imdb.com/name/{nconst}>"
    },
    "query": "select nconst, primaryName from person",
    "datasource": "SW_IMDB"
  }
},
```

Ontology and Entailment

- Here, ontology specifies mainly 4 properties about classes and properties:
  - rdfs:domain, rdfs:range, rdfs:subproperty, rdfs:subclass
- Entailment: Using a set of entailment rules, to infer new data triples and ontology triples

	Rule name	Entailment rule	
$\mathcal{R}$	rdfs5	$(p_1, :\text{subproperty}, p_2), (p_2, :\text{subproperty}, p_3) \rightarrow (p_1, :\text{subproperty}, p_3)$	$\mathcal{R}_{\text{onto}}$
	rdfs11	$(s, :\text{subclass}, o), (o, :\text{subclass}, o_1) \rightarrow (s, :\text{subclass}, o_1)$	
	ext1	$(p, :\text{domain}, o), (o, :\text{subclass}, o_1) \rightarrow (p, :\text{domain}, o_1)$	
	ext2	$(p, :\text{range}, o), (o, :\text{subclass}, o_1) \rightarrow (p, :\text{range}, o_1)$	
	ext3	$(p, :\text{subproperty}, p_1), (p_1, :\text{domain}, o) \rightarrow (p, :\text{domain}, o)$	
	ext4	$(p, :\text{subproperty}, p_1), (p_1, :\text{range}, o) \rightarrow (p, :\text{range}, o)$	
	rdfs2	$(p, :\text{domain}, o), (s_1, p, o_1) \rightarrow (s_1, :\text{type}, o)$	$\mathcal{R}_{\text{data}}$
	rdfs3	$(p, :\text{range}, o), (s_1, p, o_1) \rightarrow (o_1, :\text{type}, o)$	
	rdfs7	$(p_1, :\text{subproperty}, p_2), (s, p_1, o) \rightarrow (s, p_2, o)$	
	rdfs9	$(s, :\text{subclass}, o), (s_1, :\text{type}, s) \rightarrow (s_1, :\text{type}, o)$	