CrEDIBLE

fédération de données et de ConnaissancEs Distribuées en Imagerie BiomédicaLE

Interrogation d'entrepôts distribués et hétérogènes

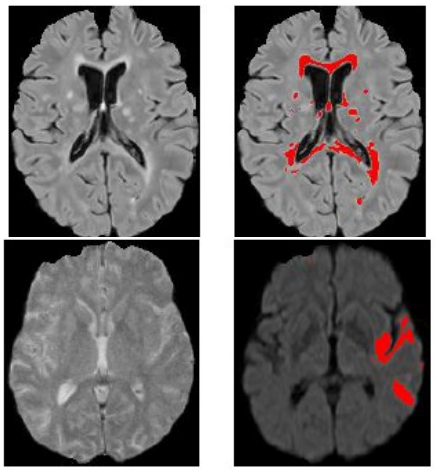
Johan Montagnat
Alban Gaignard

Contexte

- Equipe MODALIS, laboratoire I3S (Sophia Antipolis)
  - Département INS2I du CNRS
  - Génie logiciel et calcul distribué à grande échelle
- Projet CrEDIBLE
  - Mission pour l'Interdisciplinarité du CNRS
  - Appel MASTODONS 2011 (masses de données)
  - Projet reconductible sur 5 ans (d'année en année)
  - Animation de réseau scientifique (cf atelier 2013)
  - CNRS / U. Nice (I3S), INRIA (Sophia), INSERM (Rennes), U. Picardie (MIS), INSA / U. Lyon 1 (CREATIS)

Motivations

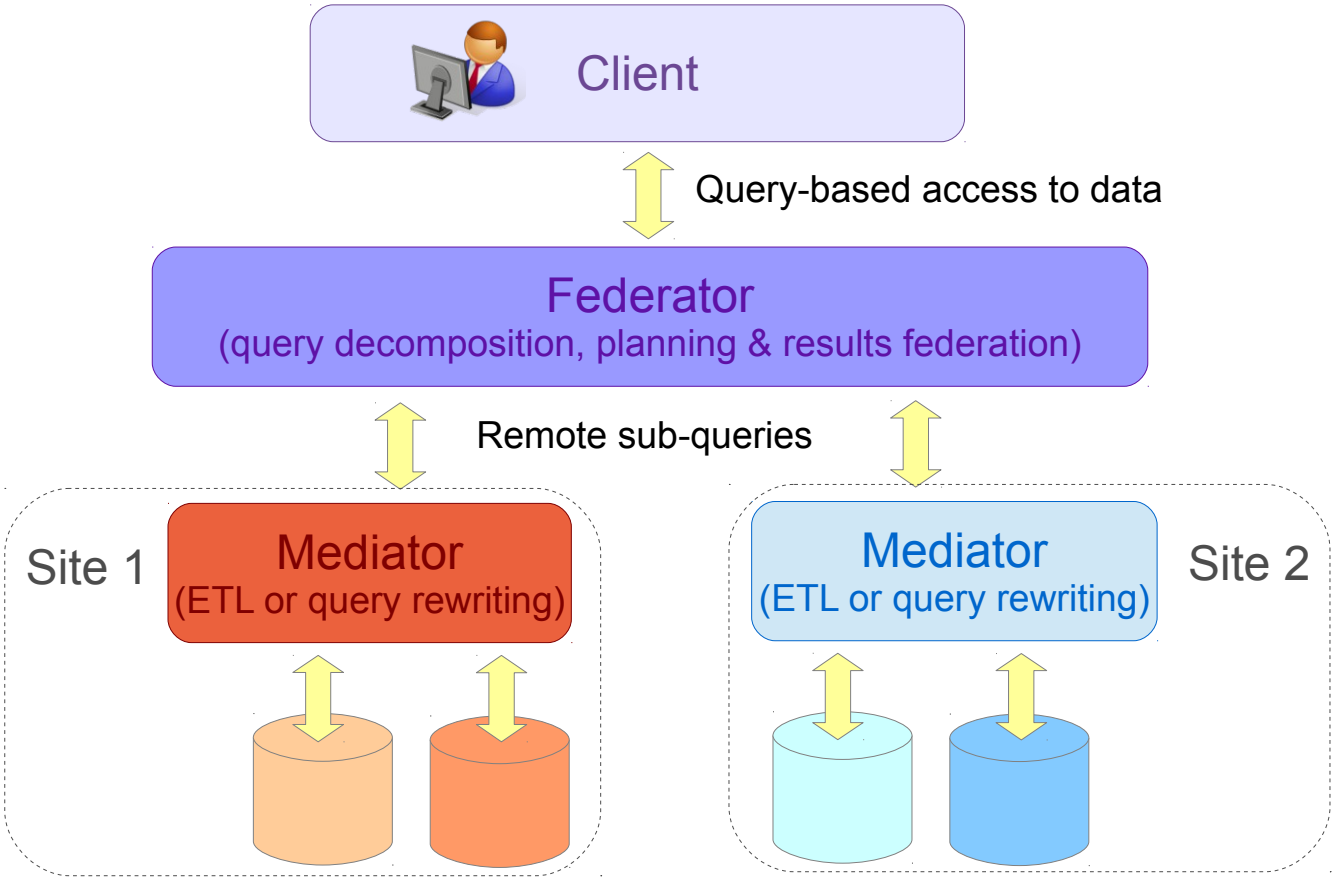
- Biomedical data
  - High heterogeneity: images, clinical data, biomarkers, biology....
  - Increasing amount / number of (open) sources – **Big Data**
    - Large-scale medical studies
      (statistical medical studies, epidemiology...)
  - Need for cross-factors analysis – **Linked Data**
    - Data (re)analysis opportunities
    - Translational research



- Centralized approaches encounter limitations
  - Large data volumes to transfer / archive / search
  - Sensitive patient data / complex access control policies
  - Need to adopt uniform data model & format
- Data is *de facto* distributed over acquisition centers

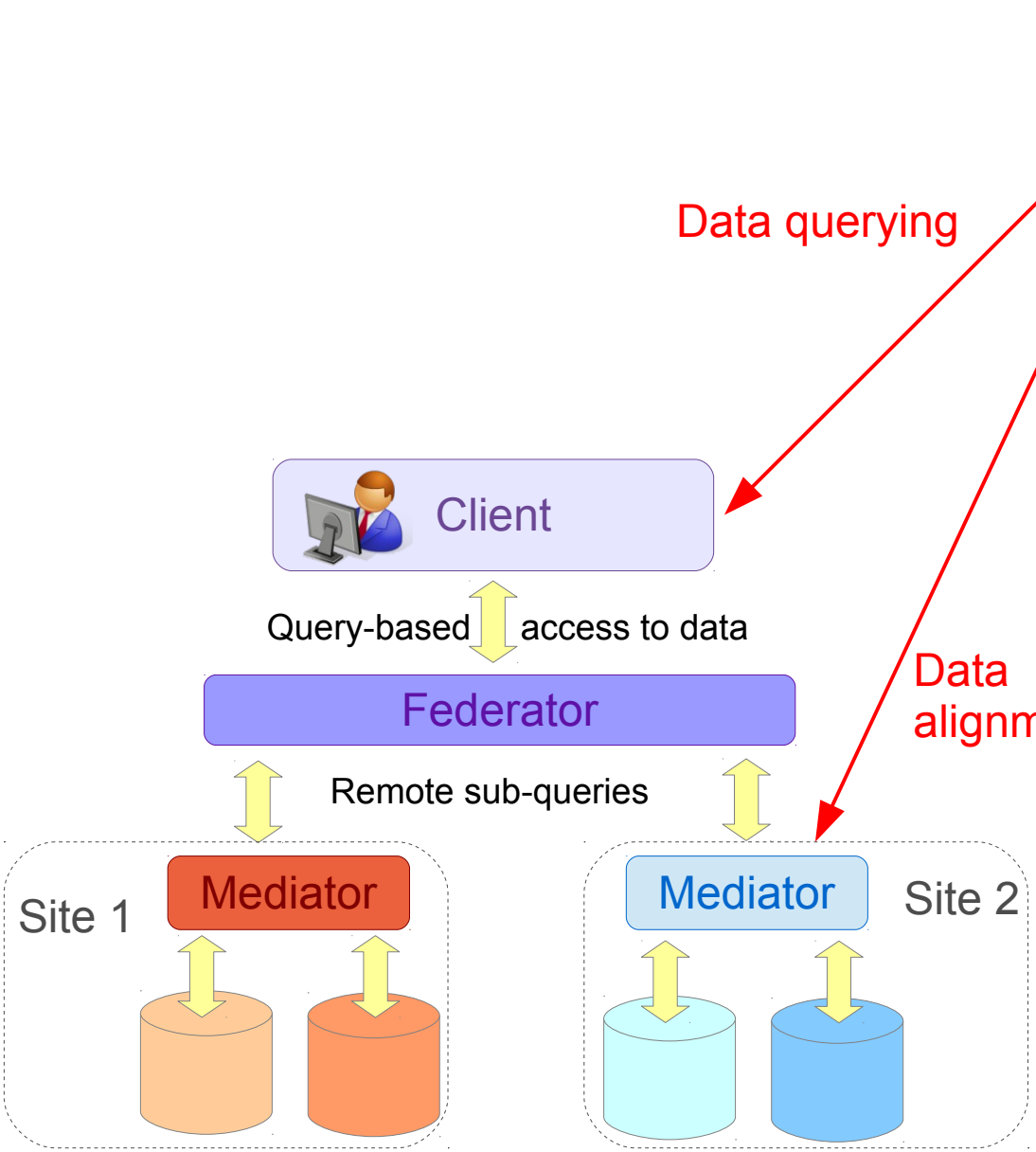
Biomedical data mediation & federation

- Data federation through distributed querying and query rewriting



- Heterogeneous databases schema mediation
- Medical data & metadata:
  - raw data + models + processing results + models + provenance...

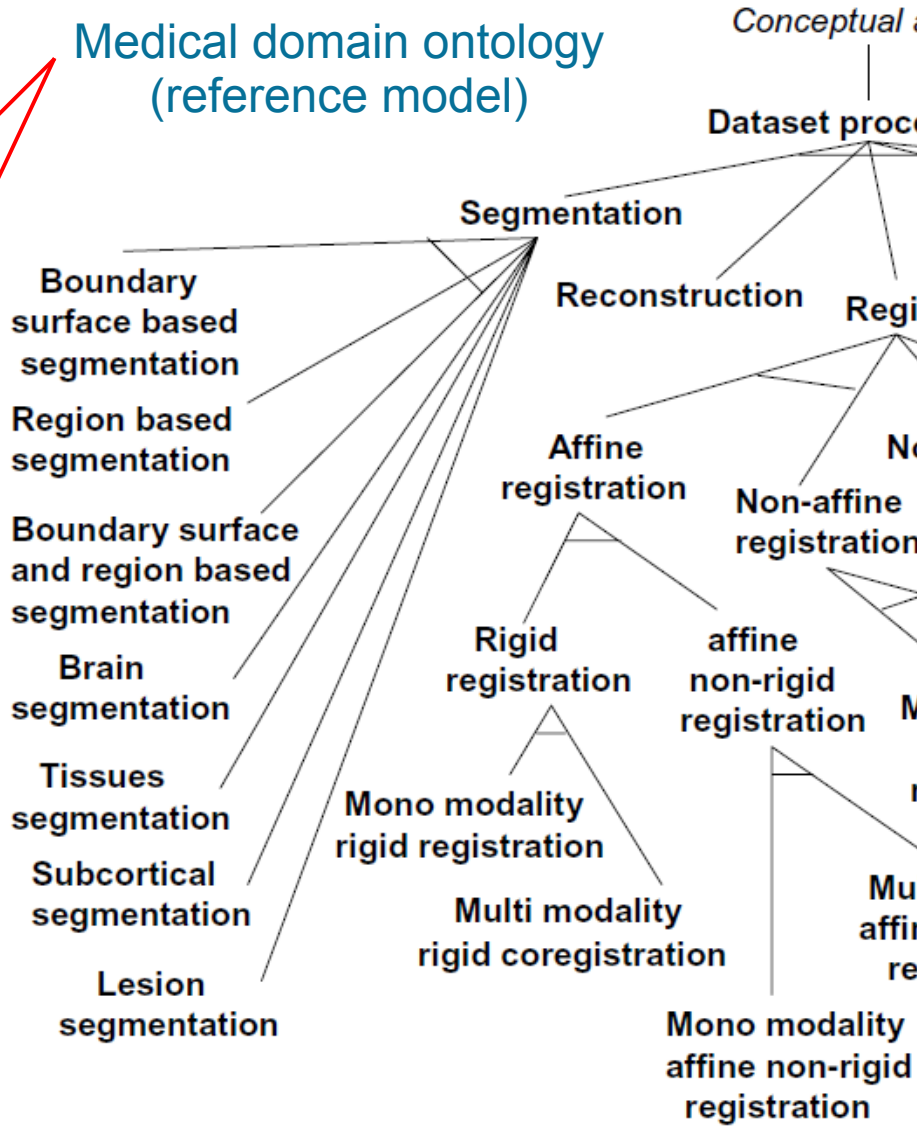
Domain ontology-based federation



Data querying

Data alignment

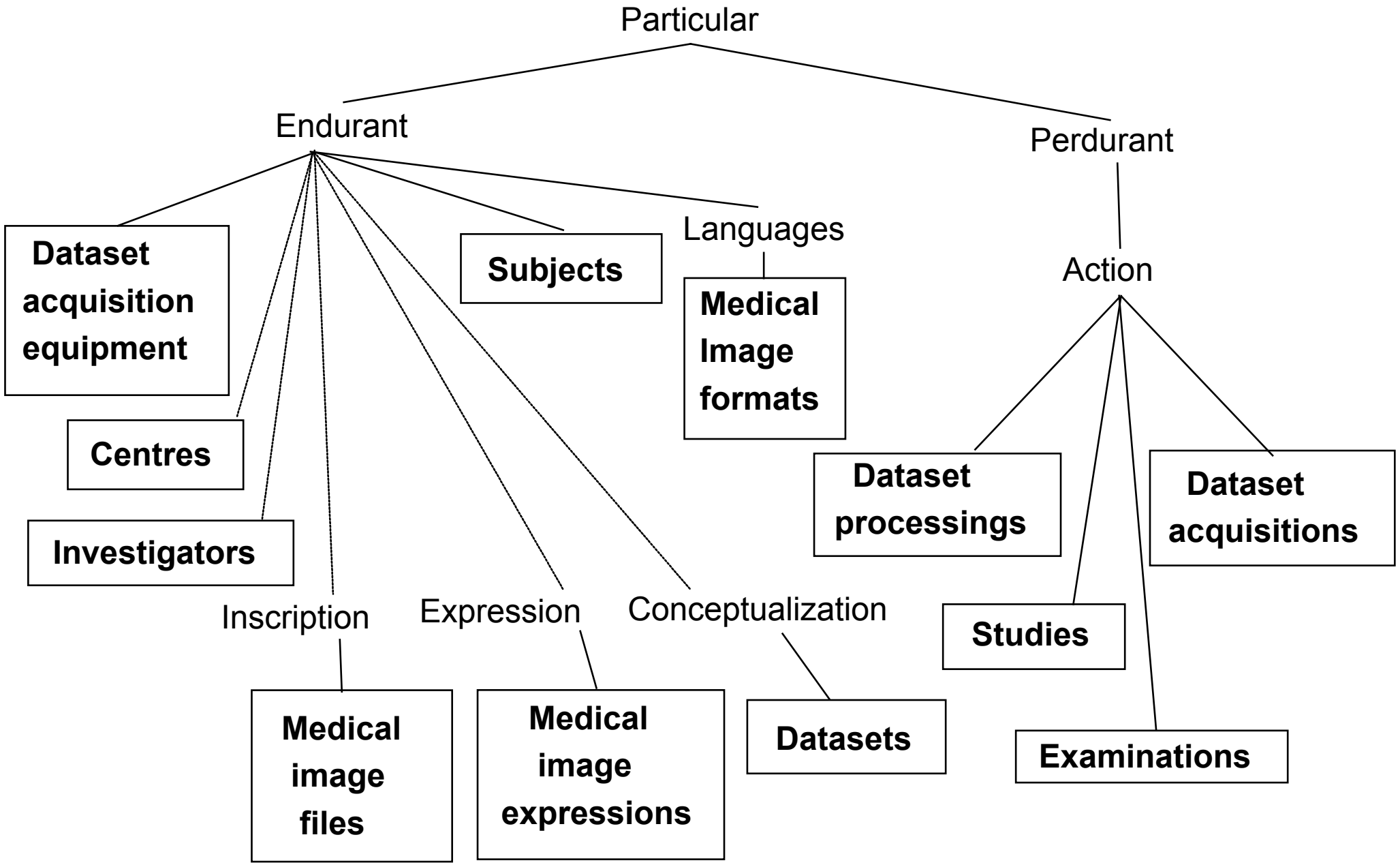
Medical domain ontology (reference model)



Reference ontology

- 3-levels structure
  - DOLCE foundational ontology, core ontologies, domain ontologies
- Covering
  - DataSets / Subjects / Studies
  - Data Processing Tools
  - ROIs and ROI Annotations
  - Scientific Measurements
  - Clinical Tests, Scores and instrument-based Assessment
  - Medical context
  - Data provenance
  - ...
- Domain-specific rules
  - Inference abilities
- Derived relational schema

Reference ontology



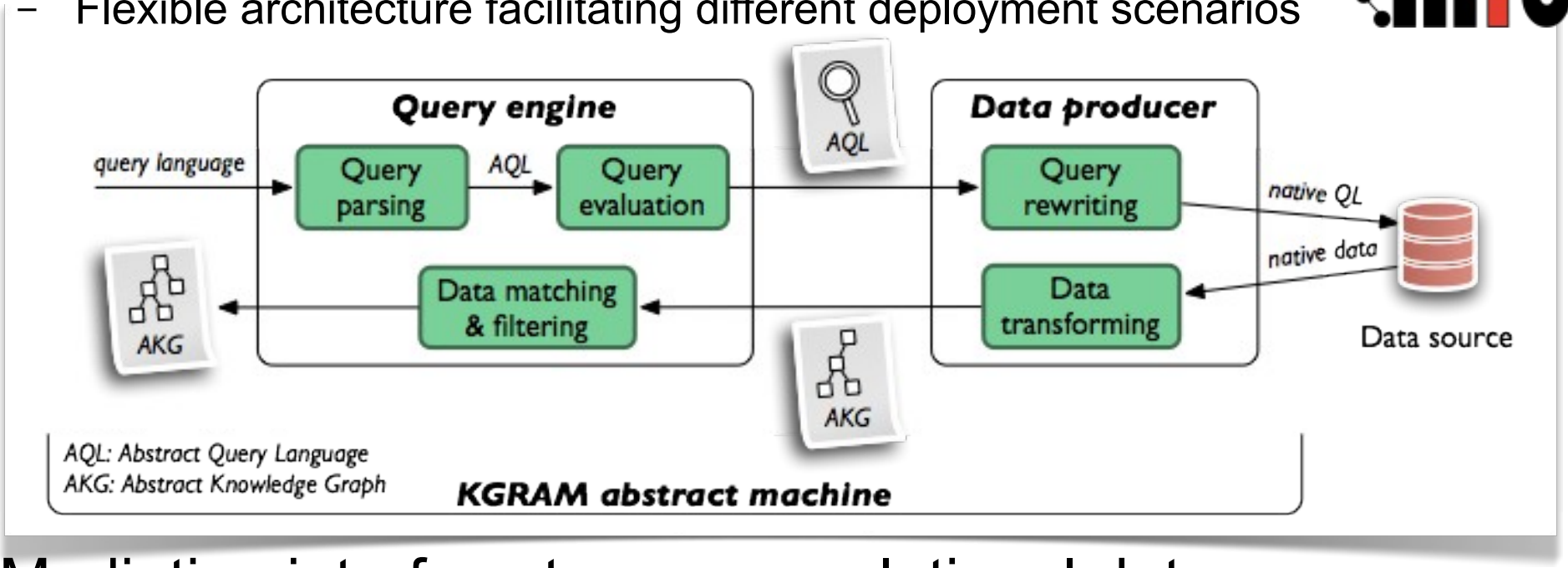
Ontology modules

- Modularized ontology to improve reuse and lightweightness
  - ONL-MR-DA: MR Dataset Acquisition
  - ONL-DP: Data Processing
  - ONL-MSA: Mental State Assessment
  - OntoVIP: Medical Image Simulation

http://bioportal.bioontology.org/ontologies

Data query and federation engine

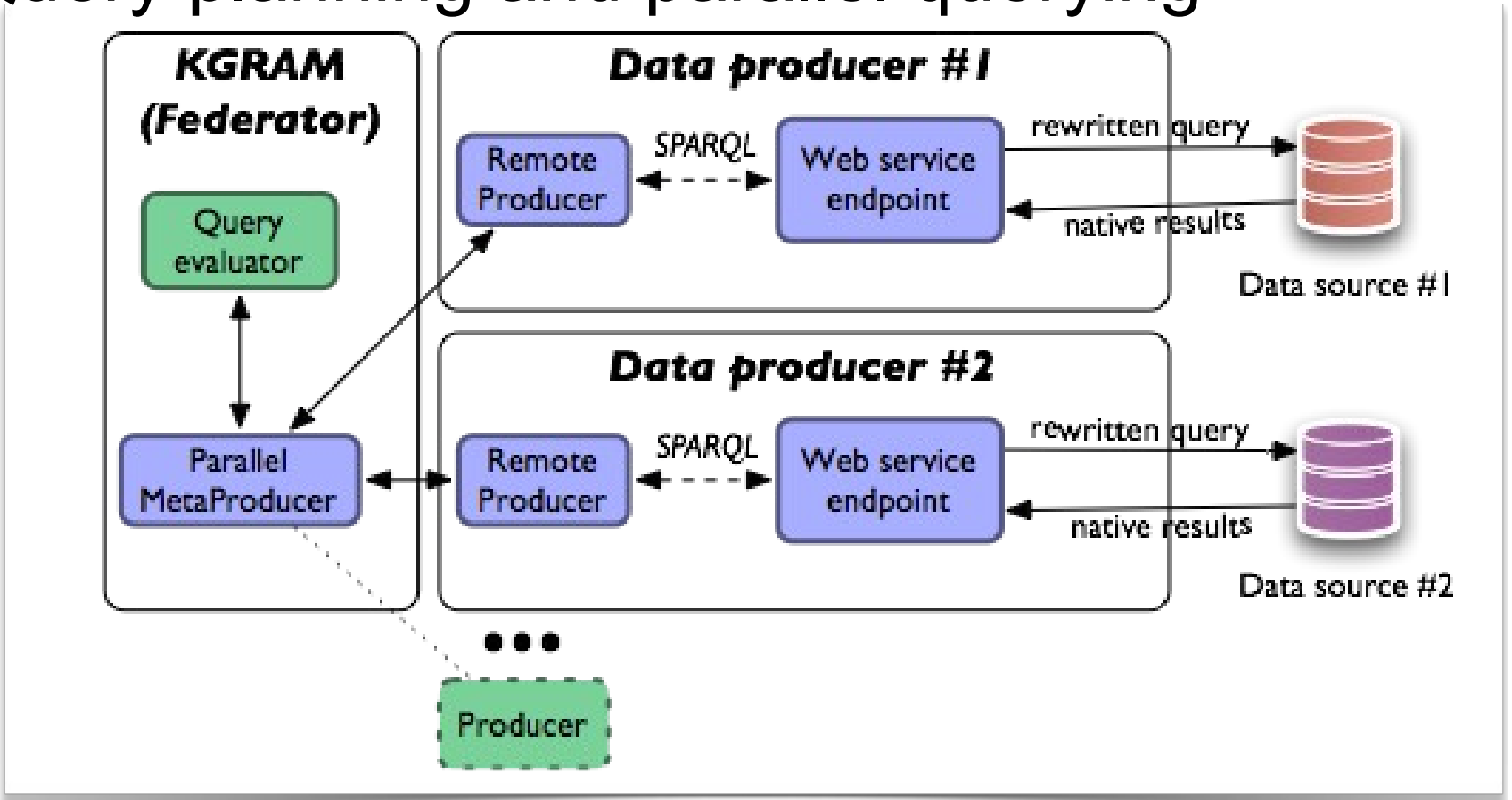
- KGRAM (Knowledge Graph Abstract Machine) Semantic query engine:
  - Full support of SPARQL1.1
  - Generic interface for heterogeneous backends
  - Flexible architecture facilitating different deployment scenarios



- Mediation interface to access relational data
  - Federated relational schema derived from the ontology

Distributed Query Processing

- Query federator decoupled from data sources
- Asynchronous querying of multiple data sources
- Query planning and parallel querying



Distributed Query Processing

- KGRAM query processing

Q

```
SELECT ?name ?date
WHERE { ?x foaf:name ?name . ?x dbpedia:birthDate ?date .
        FILTER (CONTAINS (?name, 'Bob')) }
```

- Asynchronous execution

Distributed Query Processing

- KGRAM query processing

```
Q   SELECT ?name ?date
    WHERE { ?x foaf:name ?name . ?x dbpedia:birthDate ?date .
      Q1    FILTER (CONTAINS (?name, 'Bob')) }        Q2
```

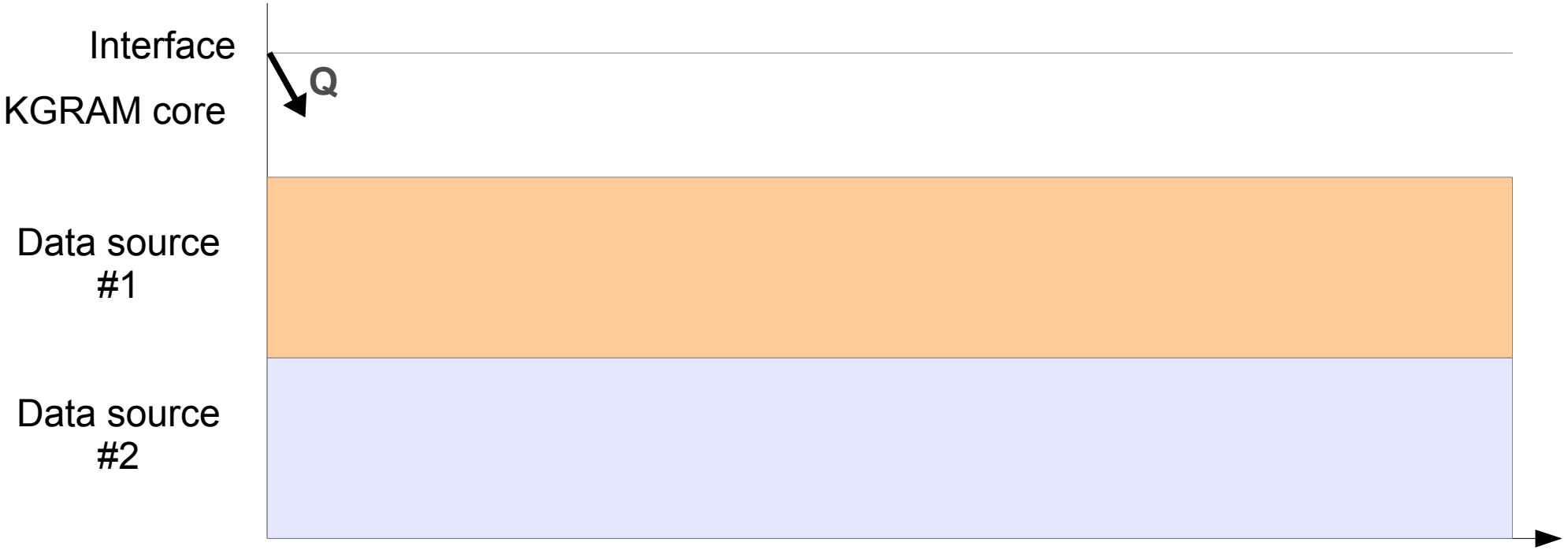
- Asynchronous execution

Distributed Query Processing

- KGRAM query processing

```
Q   SELECT ?name ?date
    WHERE { ?x foaf:name ?name . ?x dbpedia:birthDate ?date .
      Q1    FILTER (CONTAINS (?name, 'Bob')) }        Q2
```

- Asynchronous execution

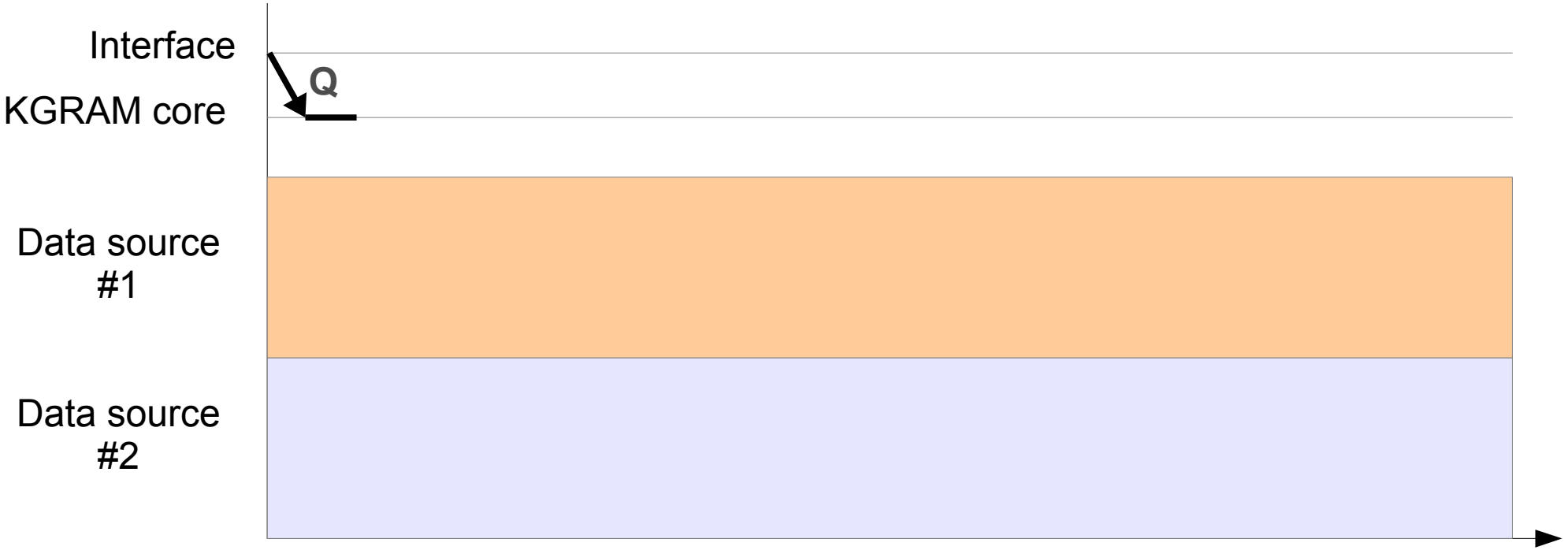


Distributed Query Processing

- KGRAM query processing

```
Q   SELECT ?name ?date
    WHERE { ?x foaf:name ?name . ?x dbpedia:birthDate ?date .
      Q1    FILTER (CONTAINS (?name, 'Bob')) }        Q2
```

- Asynchronous execution

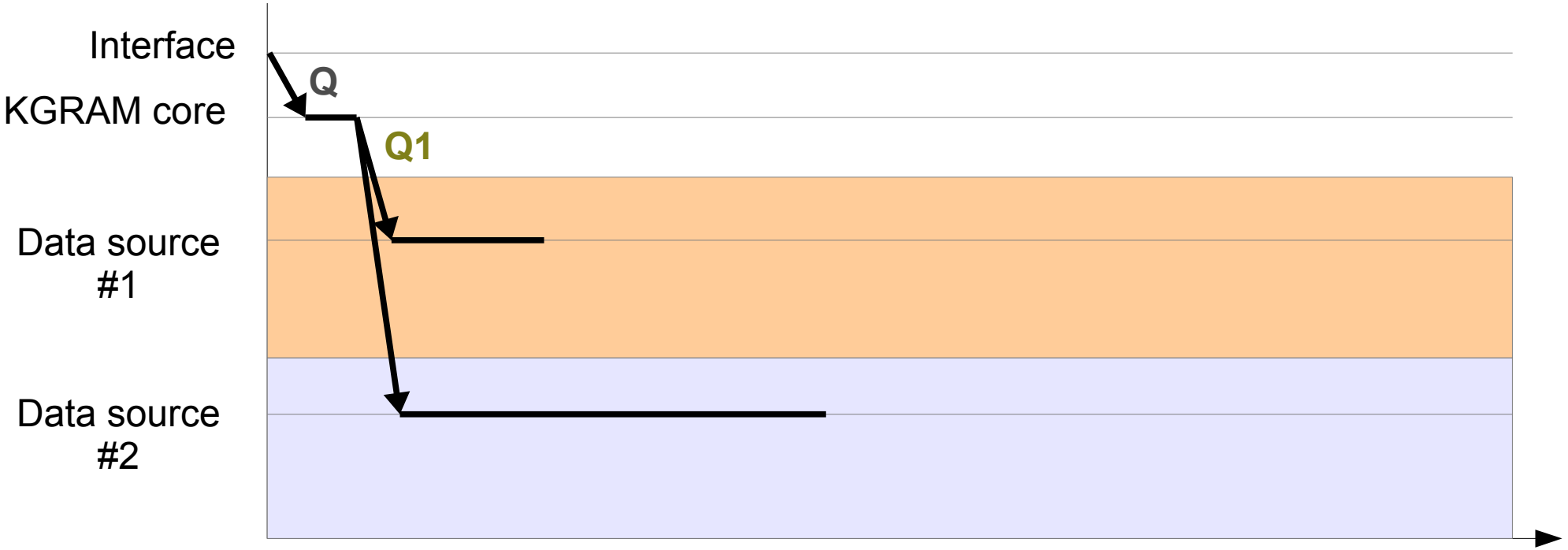


Distributed Query Processing

- KGRAM query processing

```
Q   SELECT ?name ?date
    WHERE { ?x foaf:name ?name . ?x dbpedia:birthDate ?date .
      Q1    FILTER (CONTAINS (?name, 'Bob')) }        Q2
```

- Asynchronous execution

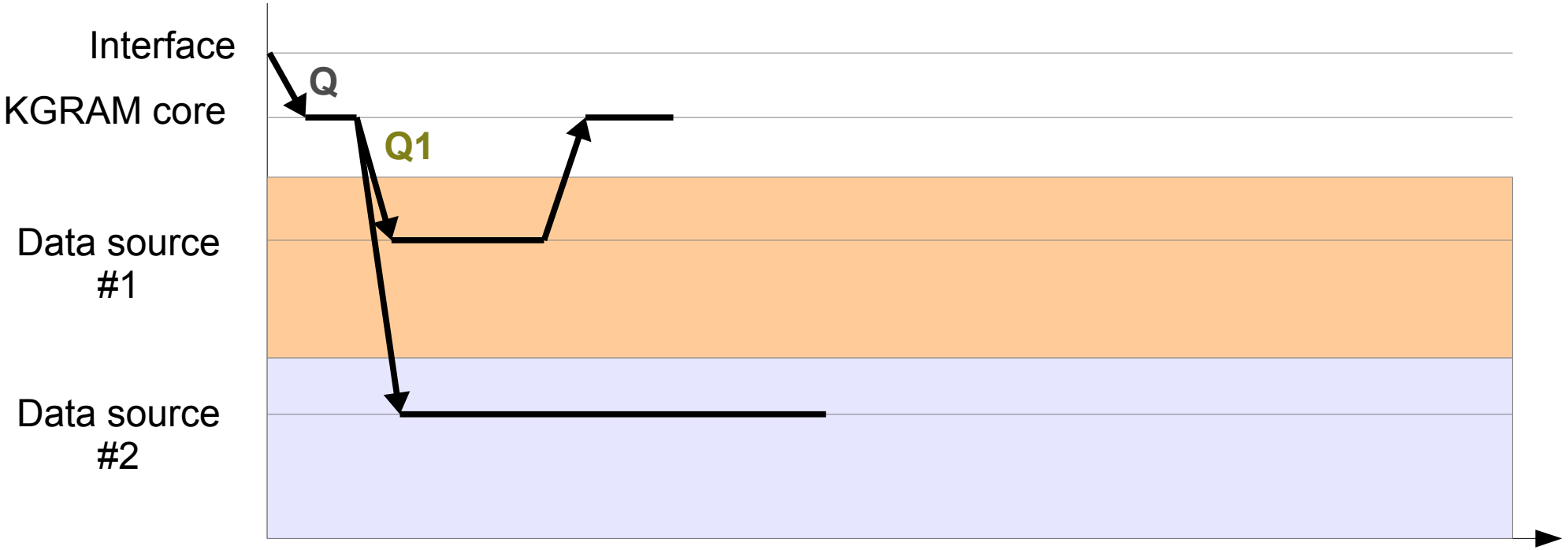


Distributed Query Processing

- KGRAM query processing

```
Q   SELECT ?name ?date
    WHERE { ?x foaf:name ?name . ?x dbpedia:birthDate ?date .
      Q1    FILTER (CONTAINS (?name, 'Bob')) }        Q2
```

- Asynchronous execution

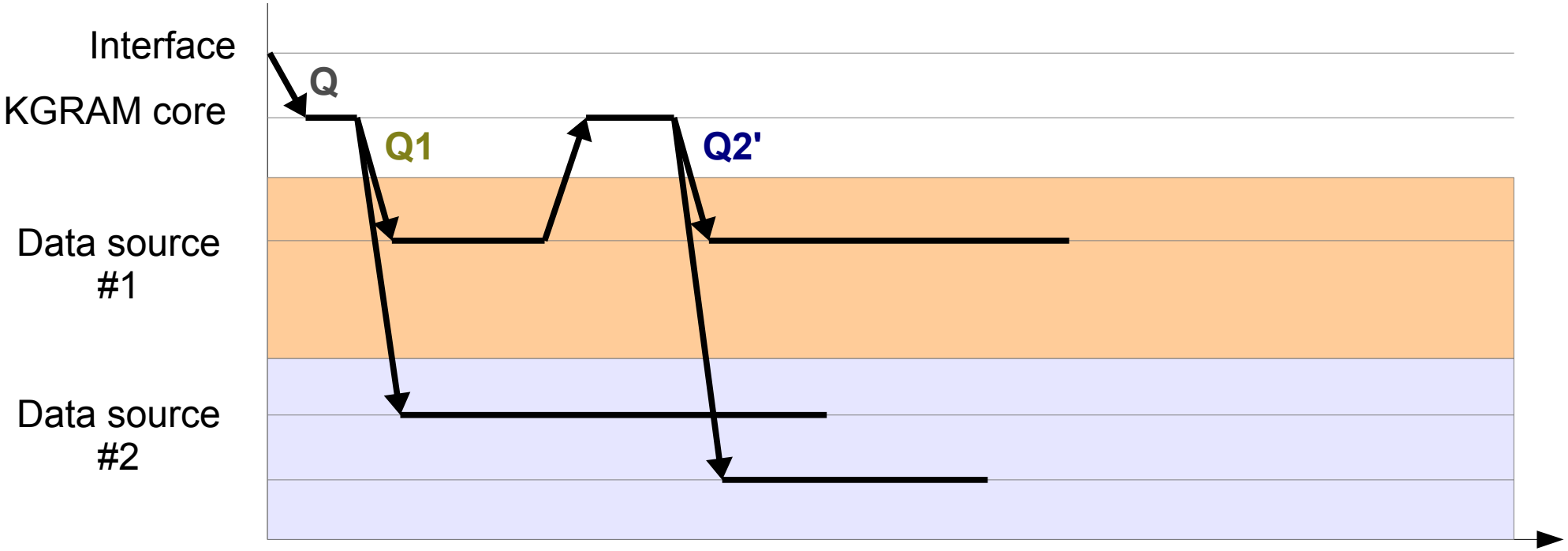


Distributed Query Processing

- KGRAM query processing

```
Q   SELECT ?name ?date
    WHERE { ?x foaf:name ?name . ?x dbpedia:birthDate ?date .
        Q1  FILTER (CONTAINS (?name, 'Bob')) }        Q2
```

- Asynchronous execution

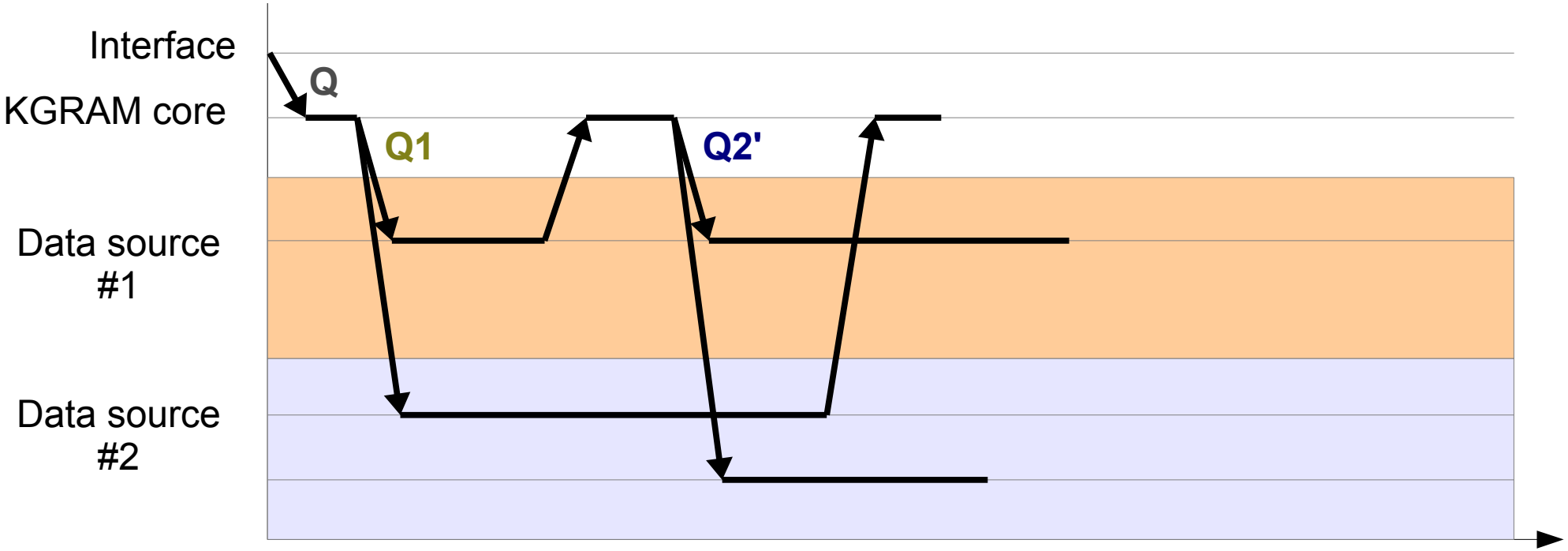


Distributed Query Processing

- KGRAM query processing

```
Q   SELECT ?name ?date
    WHERE { ?x foaf:name ?name . ?x dbpedia:birthDate ?date .
        Q1  FILTER (CONTAINS (?name, 'Bob')) }        Q2
```

- Asynchronous execution

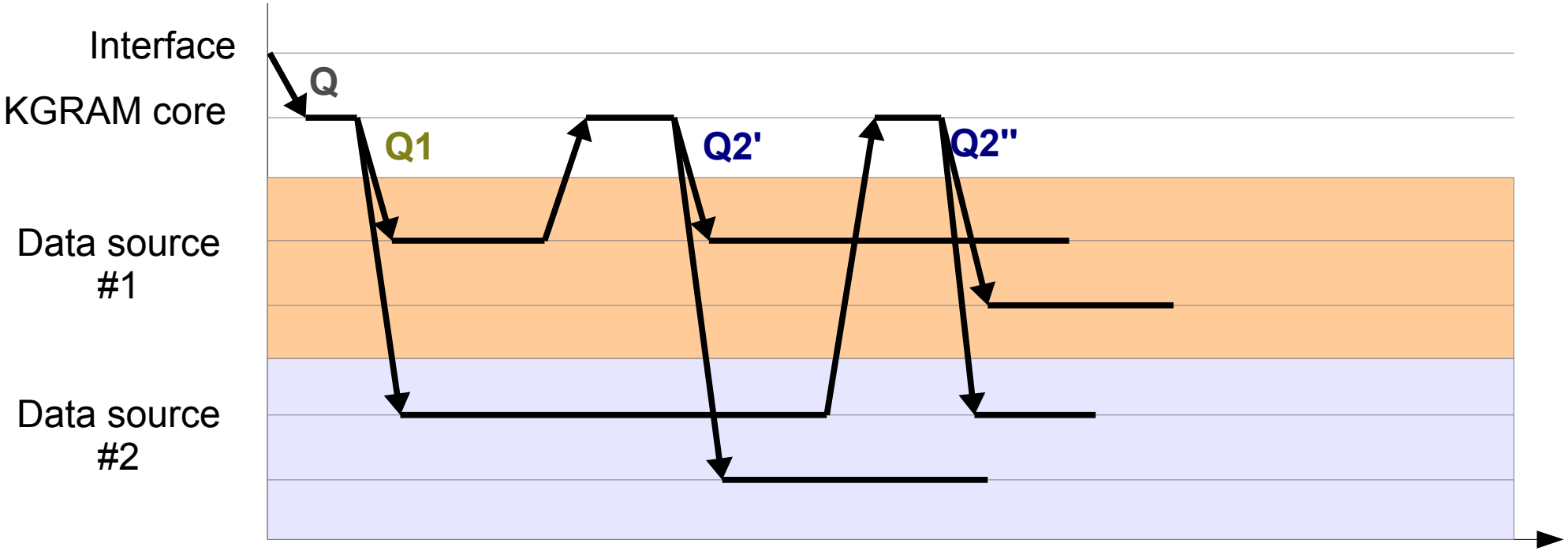


Distributed Query Processing

- KGRAM query processing

```
Q   SELECT ?name ?date
    WHERE { ?x foaf:name ?name . ?x dbpedia:birthDate ?date .
      Q1    FILTER (CONTAINS (?name, 'Bob')) }        Q2
```

- Asynchronous execution

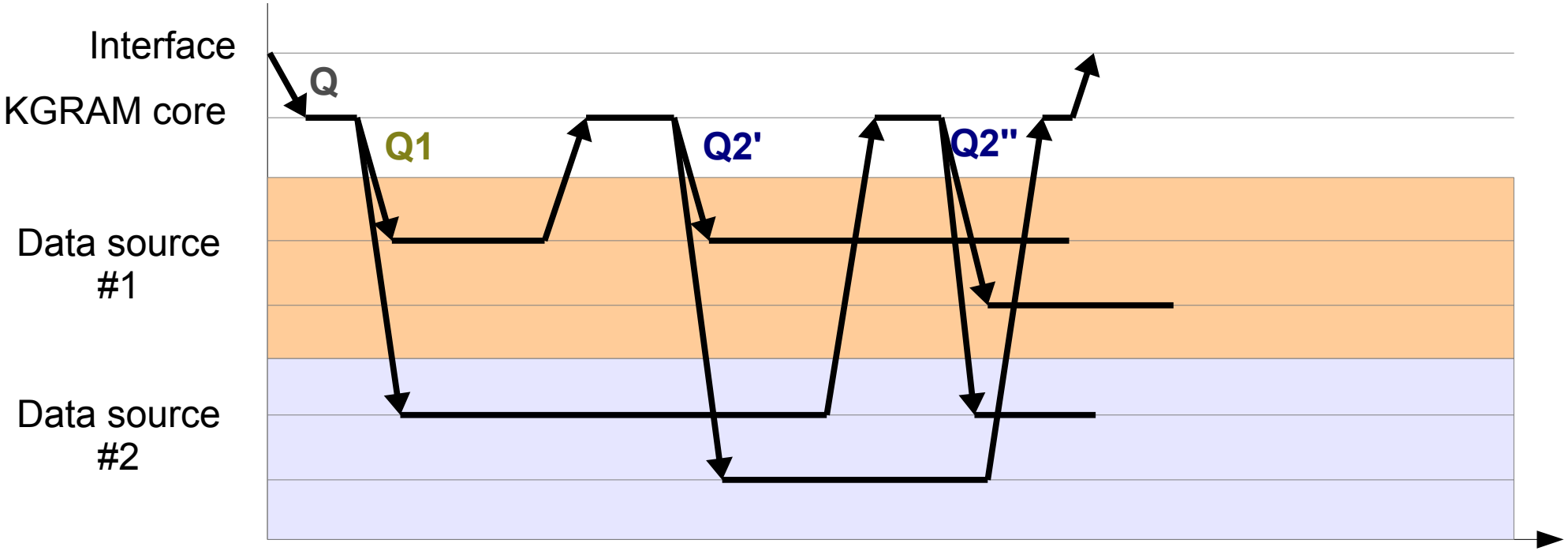


Distributed Query Processing

- KGRAM query processing

```
Q   SELECT ?name ?date
    WHERE { ?x foaf:name ?name . ?x dbpedia:birthDate ?date .
      Q1    FILTER (CONTAINS (?name, 'Bob')) }        Q2
```

- Asynchronous execution

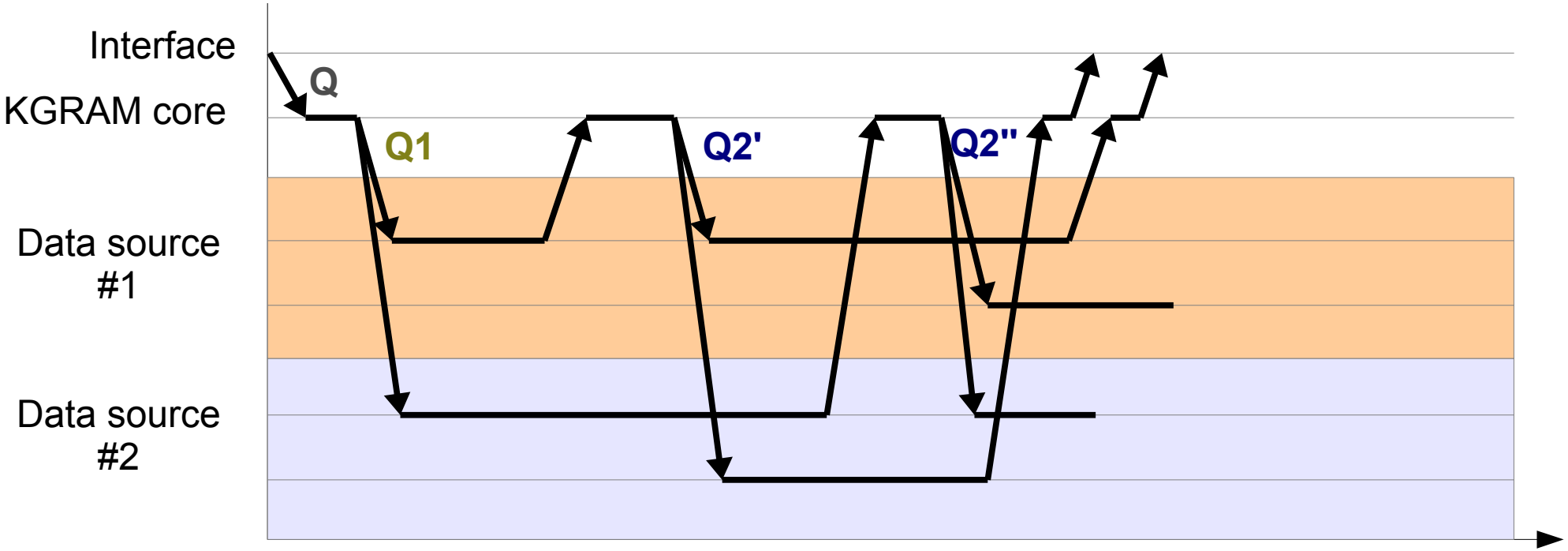


Distributed Query Processing

- KGRAM query processing

```
Q   SELECT ?name ?date
    WHERE { ?x foaf:name ?name . ?x dbpedia:birthDate ?date .
      Q1    FILTER (CONTAINS (?name, 'Bob')) }        Q2
```

- Asynchronous execution

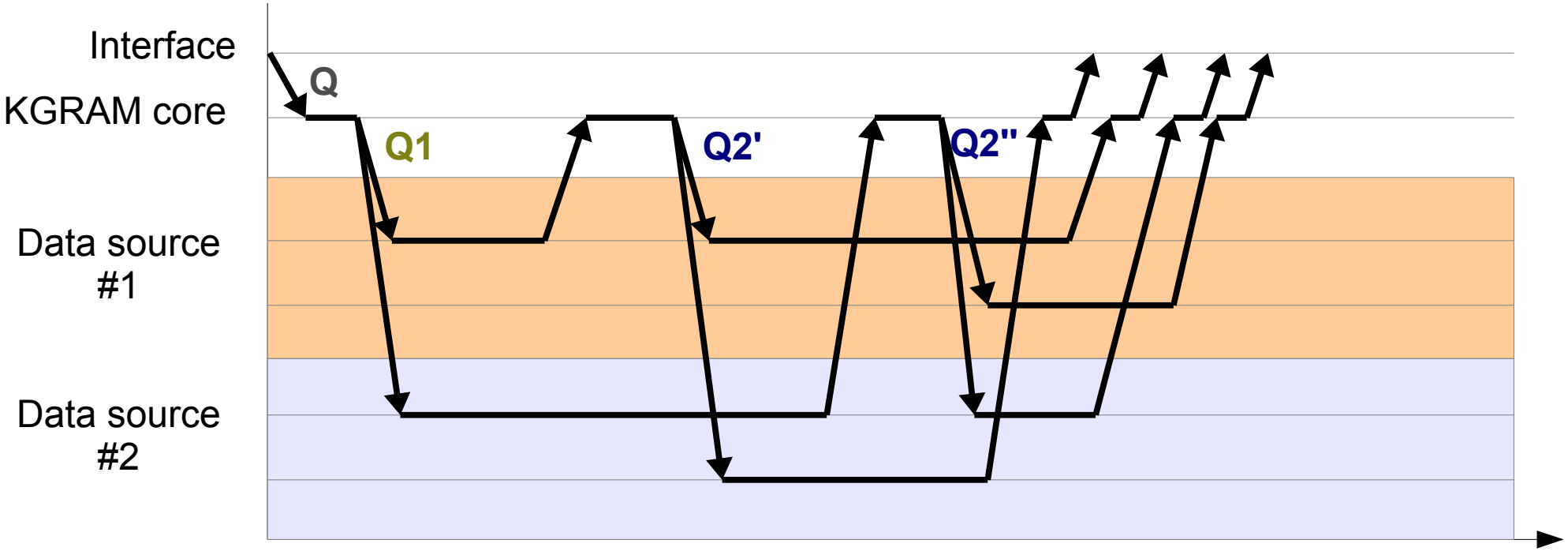


Distributed Query Processing

- KGRAM query processing

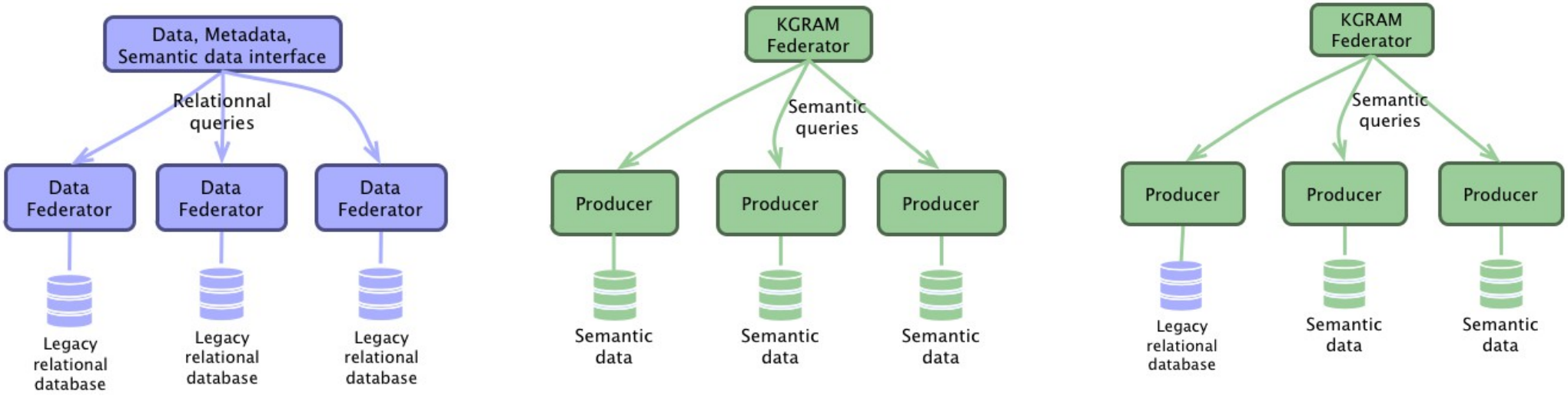
```
Q   SELECT ?name ?date
    WHERE { ?x foaf:name ?name . ?x dbpedia:birthDate ?date .
      Q1    FILTER (CONTAINS (?name, 'Bob')) }        Q2
```

- Asynchronous execution

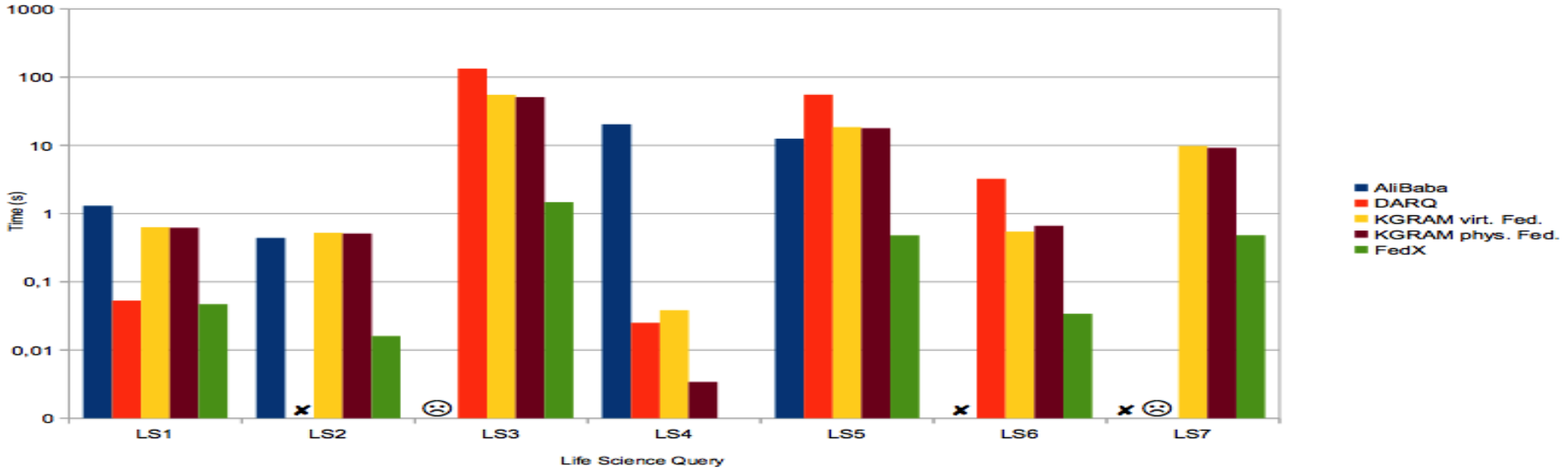


Performance results

- Heterogeneous (relational / semantic) stores querying

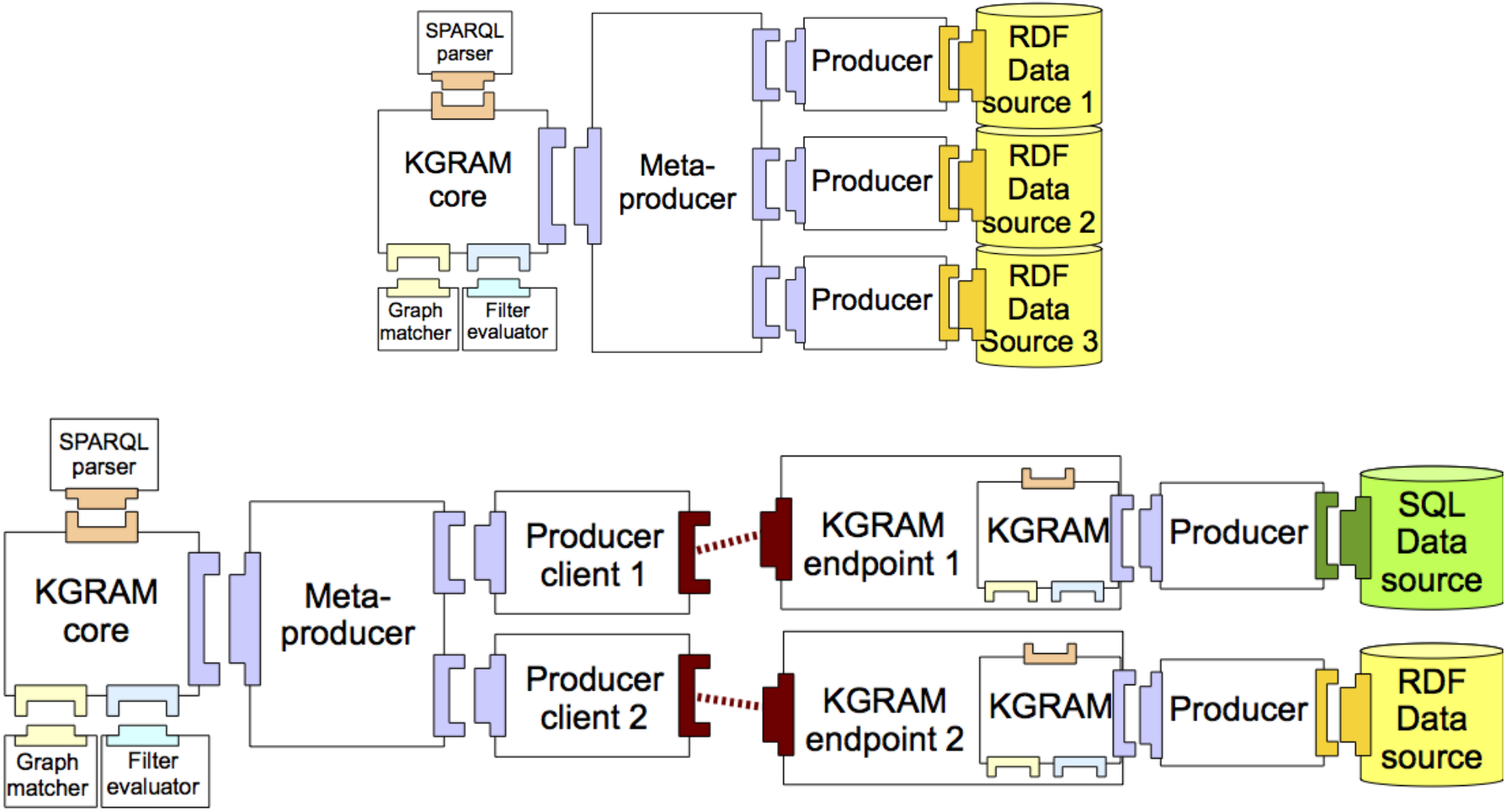


- FedBench standard benchmark



Deployment

- Customizable for different deployment scenarios



Conclusions

- Query-based data federation
- Using semantic web standards (SPARQL, RDF)
  - Emphasis on query language expressivity
- Ontology-based
  - Reference model for data alignment and query terms
- Currently deployed at medium scale
  - Broad applicability (standards compliance) given that ontologies are available
  - Query optimization work on-going