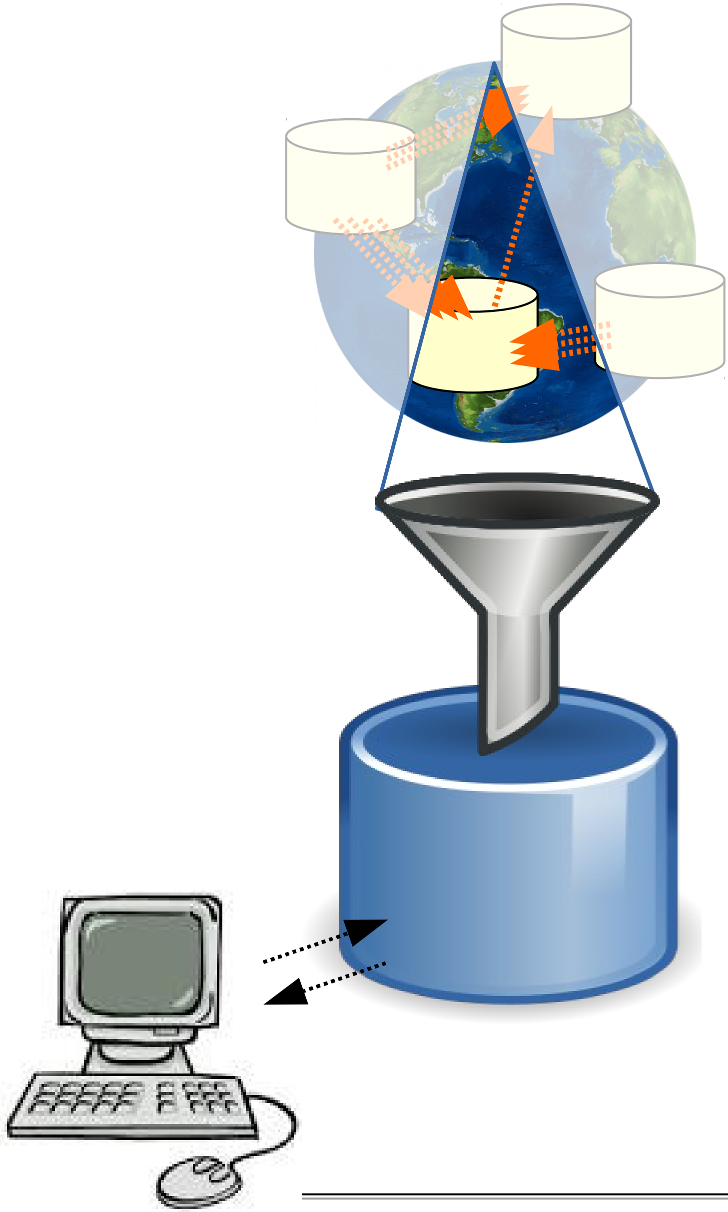
Semantic Web Technologies

Topic: Querying RDF Data Live on the Web

Olaf Hartig
olaf.hartig@liu.se

LINKÖPING UNIVERSITY

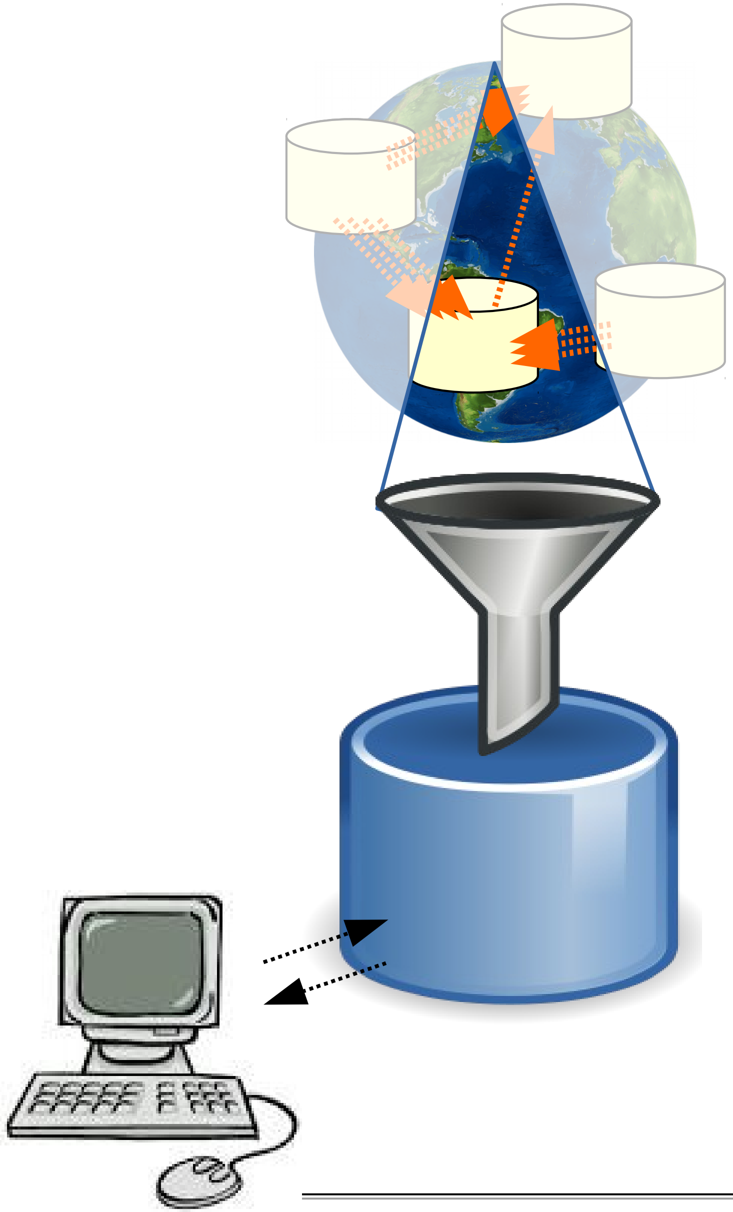
Prevalent Approach to Build Applications



- Set up a local database
- Copy data collected from the Web into this database
- Query this database

Picture source: https://pixabay.com/en/world-earth-planet-globe-map-1301744/

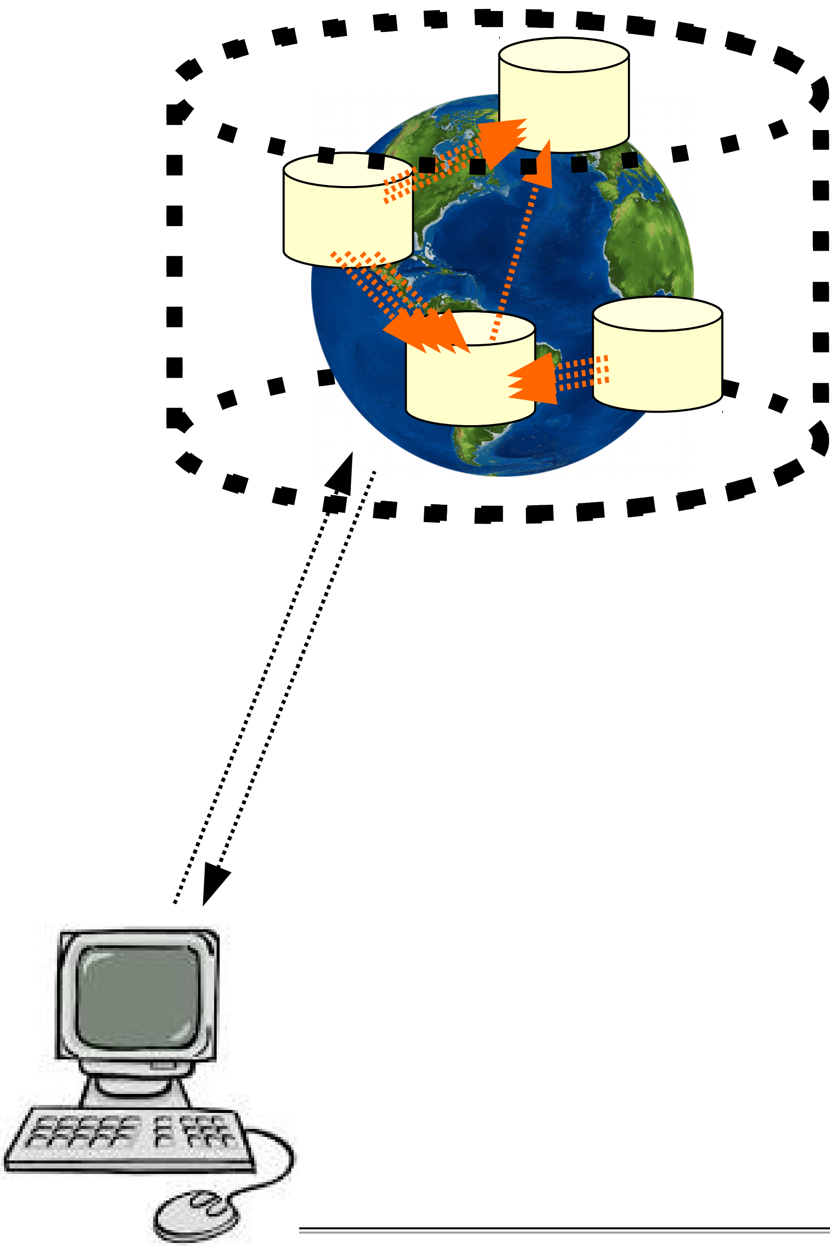
Limitations



- Setting up, populating, and maintaining the local DB *requires significant resources*
  - impractical for many small organizations and individuals
- *Legal issues* may prevent storing a copy of some of the data
- New data sources *not captured*
- Copied data becomes *outdated*

Picture source: https://pixabay.com/en/world-earth-planet-globe-map-1301744/

An Alternative View



- Conceive the Web itself as a *decentralized database system*

  … that can be queried in a *declarative* manner, where

  … queries over this database answered by accessing directly the *original data sources*

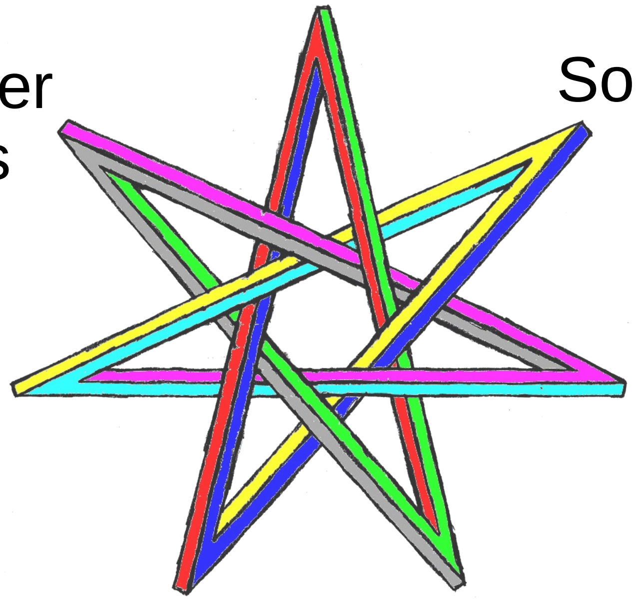
Picture source: https://pixabay.com/en/world-earth-planet-globe-map-1301744/

Ingredients

Formal foundations, including a declarative query language

Client-server interfaces

Source selection techniques



Query execution algorithms

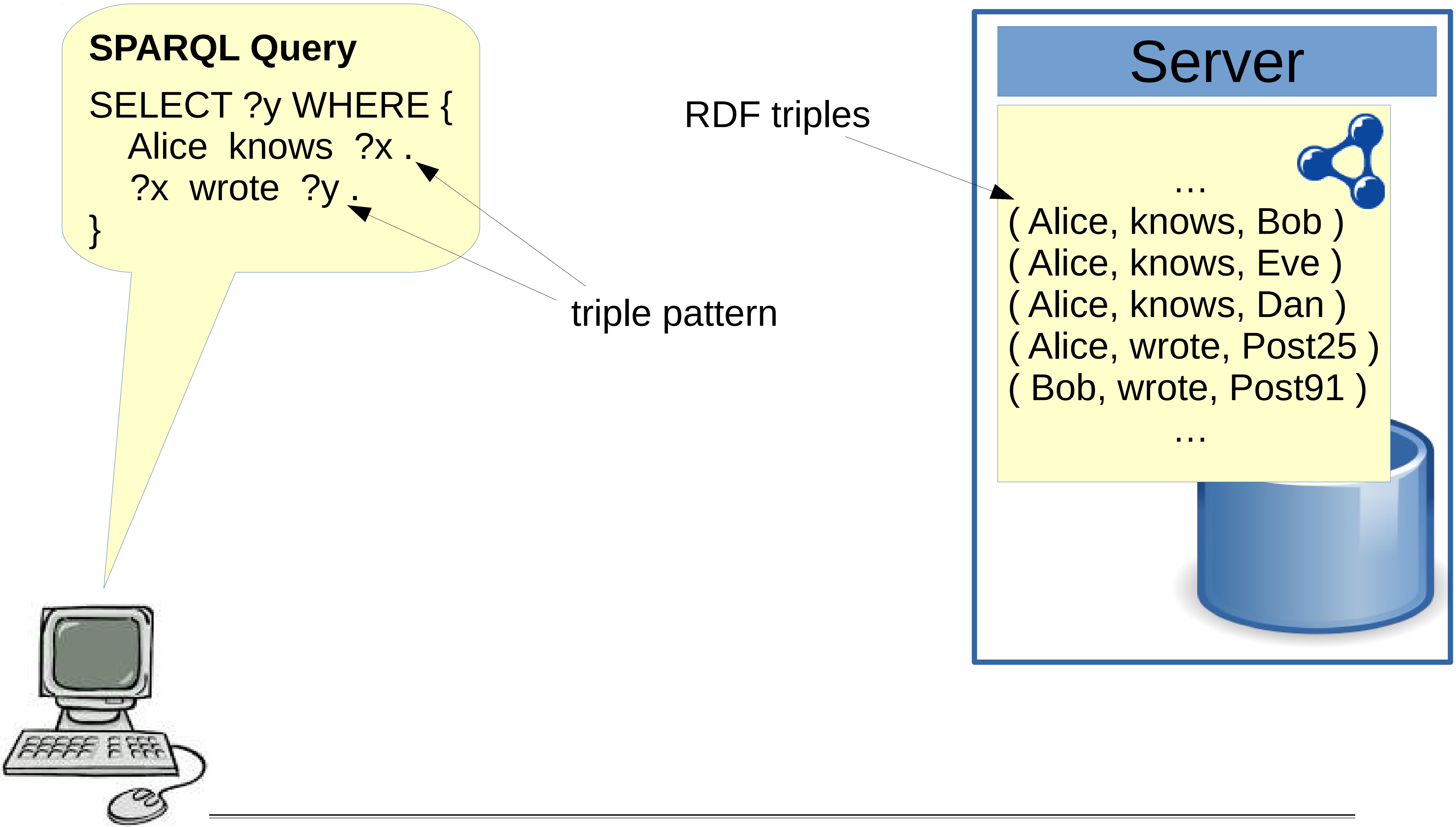
Query optimization techniques

Data provenance and data quality

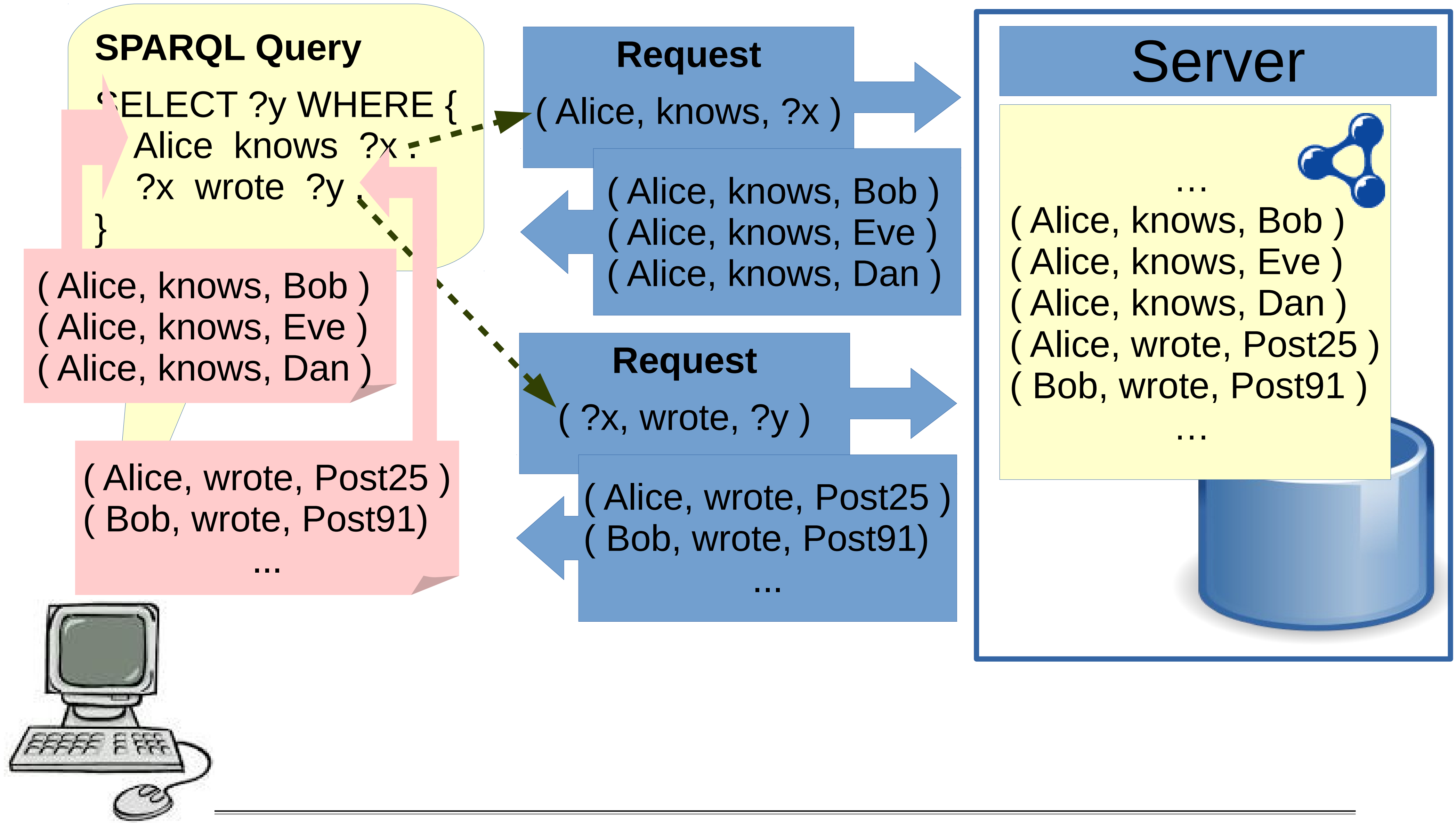
Data integration techniques

Picture source: https://www.deviantart.com/theceltícpoet/art/Seven-Pointed-Impossible-Star-189689250

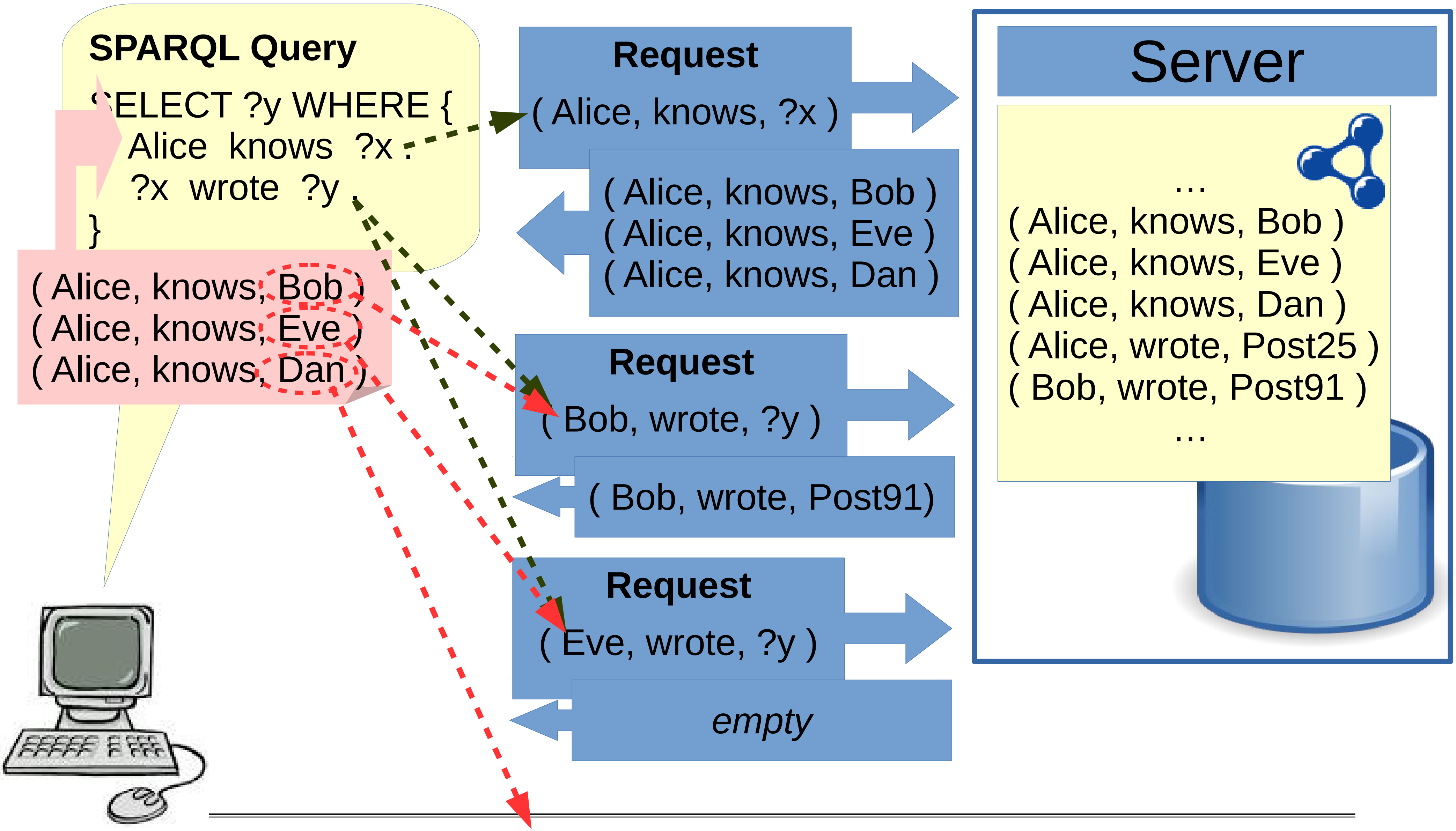
Example Scenario



Example Scenario



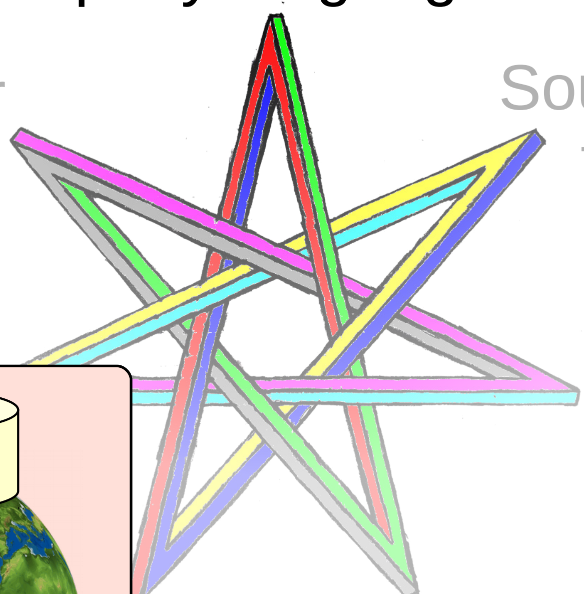
Example Scenario



Ingredients (cont’d)

Formal foundations, including a declarative query language

[ESWC’12], [HT’12], [WWW’14], [AMW’15], [ESWC’15]*, [ISWC’15], [JWS’16], [SWJ’17], [JWE’19]



Client-server interfaces

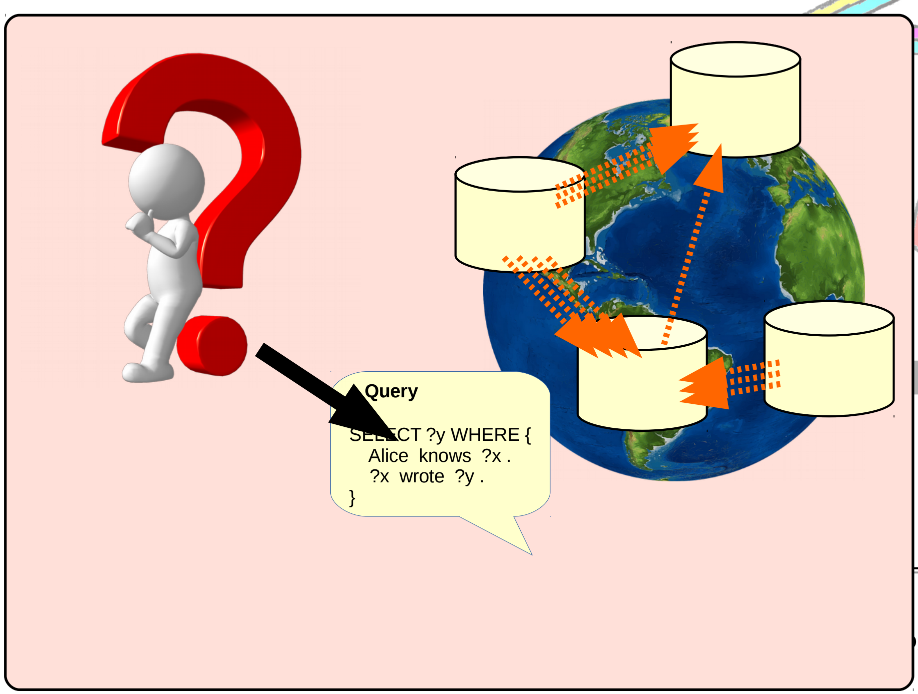
Source selection techniques

Query optimization techniques

Data integration techniques

ance

ality



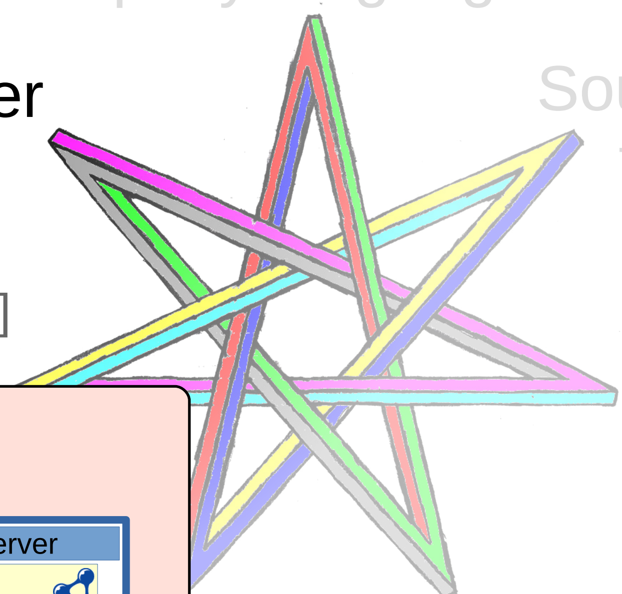
*Winner of Best Research Paper Award

Picture source: https://www.deviantart.com/thecelticpoet/art/Seven-Pointed-Impossible-Star-189689250
http://pngimg.com/download/38110

Ingredients (cont'd)

Formal foundations, including a declarative query language

[ESWC'12], [HT'12], [WWW'14], [AMW'15], [ESWC'15]*, [ISWC'15], [JWS'16], [SWJ'17], [JWE'19]



Source selection techniques

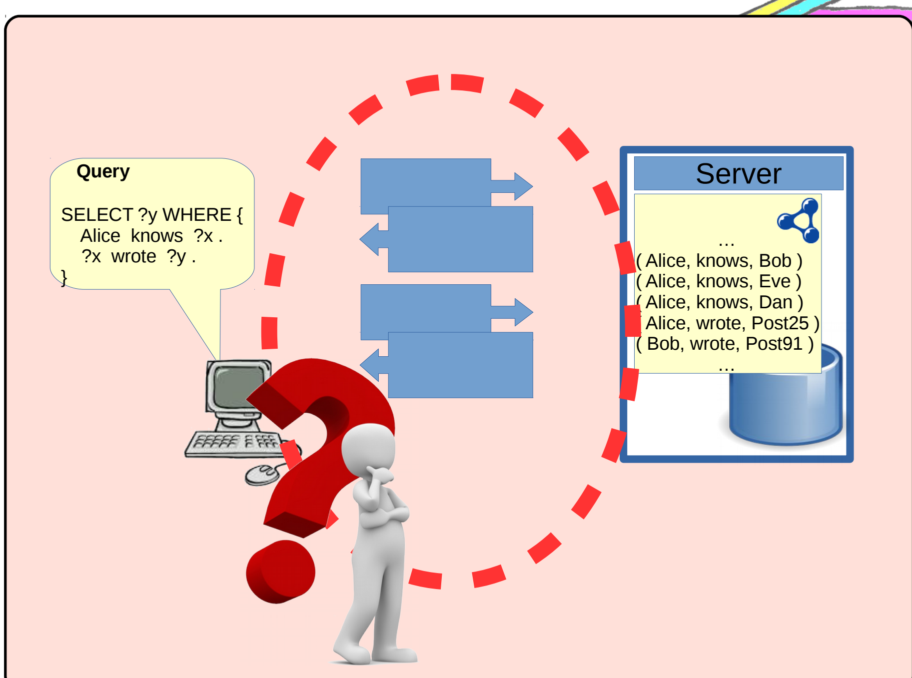
[ISWC'14], [ODBASE'16], [JWS'16], [AMW'17], [ISWC'17]*, [WWW'18], [IJCAI'18]. [iiWAS'20]

Client-server interfaces

Query optimization techniques

ance
ality

Data integration techniques



*Winner of Best Research Paper Award

Picture source: https://www.deviantart.com/thecelticpoet/art/Seven-Pointed-Impossible-Star-189689250
https://pixabay.com/sv/fr%C3%A5getecken-fr%C3%A5ga-svar-s%C3%B6kmotor-1019820/

Ingredients (cont'd)

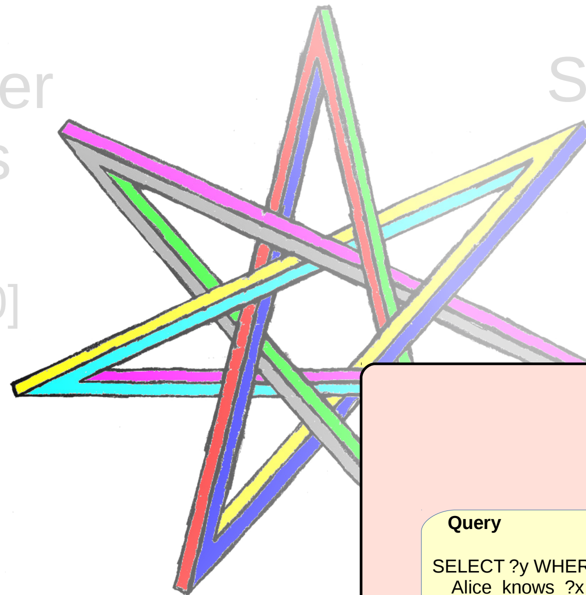
Formal foundations, including a declarative query language

[ESWC'12], [HT'12], [WWW'14], [AMW'15], [ESWC'15]*, [ISWC'15], [JWS'16], [SWJ'17], [JWE'19]

Client-server interfaces

[ISWC'14], [ODBASE'16], [JWS'16], [AMW'17], [ISWC'17]*, [WWW'18], [IJCAI'18], [iiWAS'20]

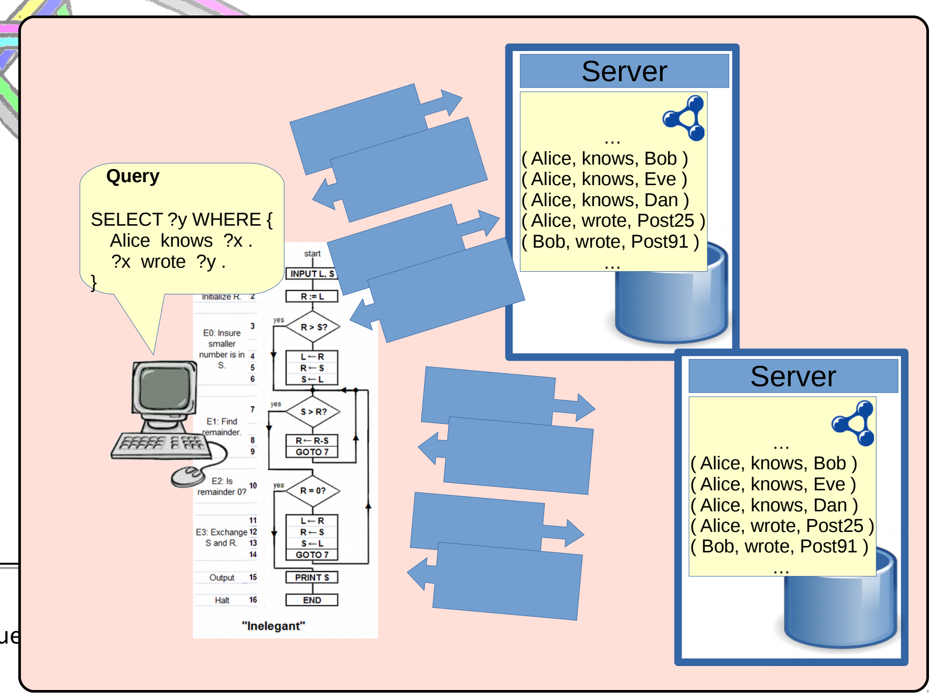
Source selection techniques



Query execution algorithms

[ISWC'09], [LDOW'11], [DBS'13], [SIGMOD'13]

Data provenance and data quality



Picture source: https://www.deviantart.com/theelticpoet/art/Seven-Pointed-Impossible-Star-189689250

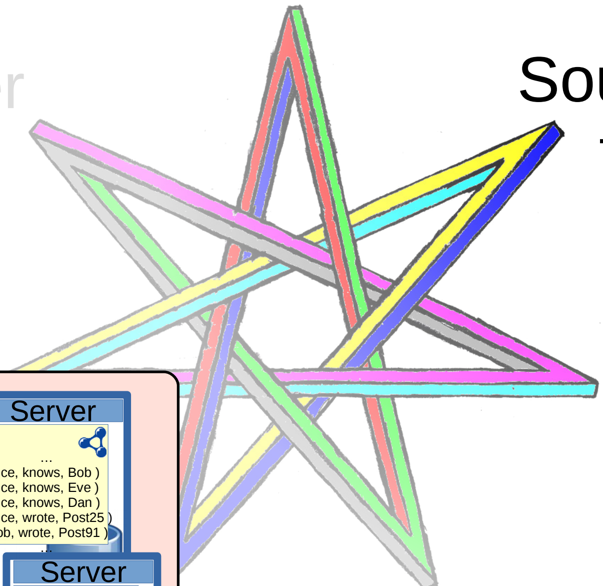
Ingredients (cont'd)

Formal foundations, including a declarative query language

[ESWC'12], [HT'12], [WWW'14], [AMW'15], [ESWC'15]*, [ISWC'15], [JWS'16], [SWJ'17], [JWE'19]

Client-server interfaces

[ISWC'14], [ODBASE'16], [JWS'16], [AMW'17], [ISWC'17]*, [WWW'18], [IJCAI'18], [iiWAS'20]



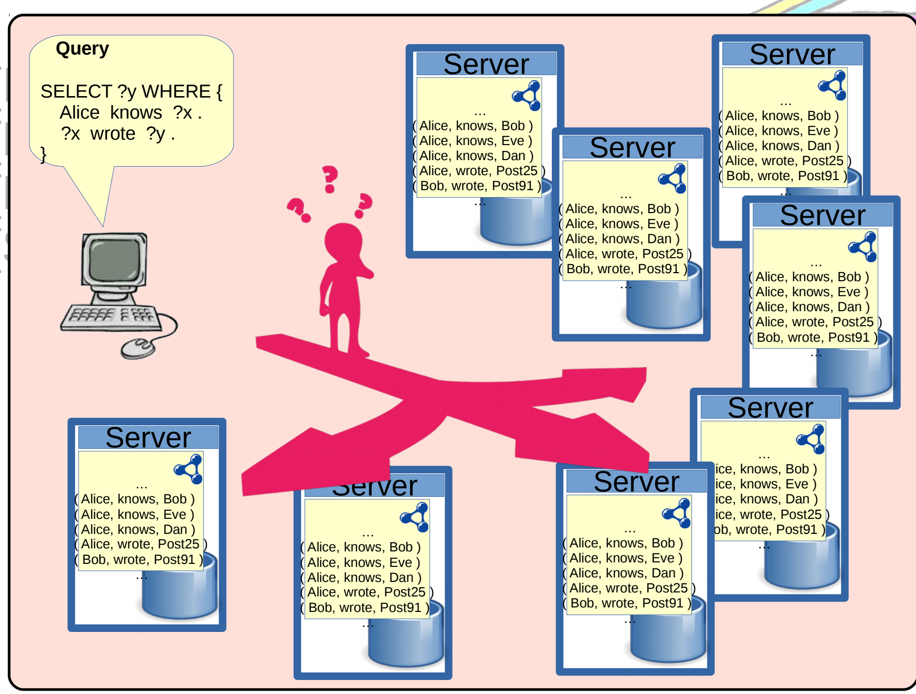
Source selection techniques

[ISWC'16], [Semantics'18]*

Query optimization techniques

…nce …ality

Data integration techniques



*Winner of Best Research Paper Award

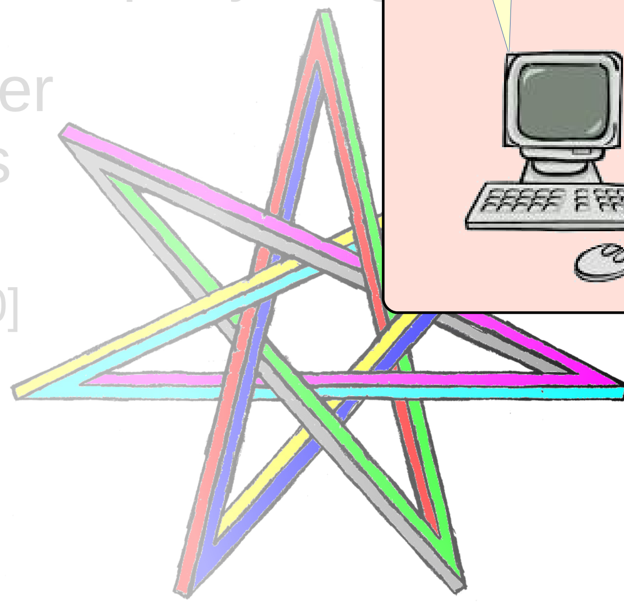
Picture source: https://www.deviantart.com/theceltticpoet/art/Seven-Pointed-Impossible-Star-189689250
https://svgsilh.com/03a9f4/image/3183317.html

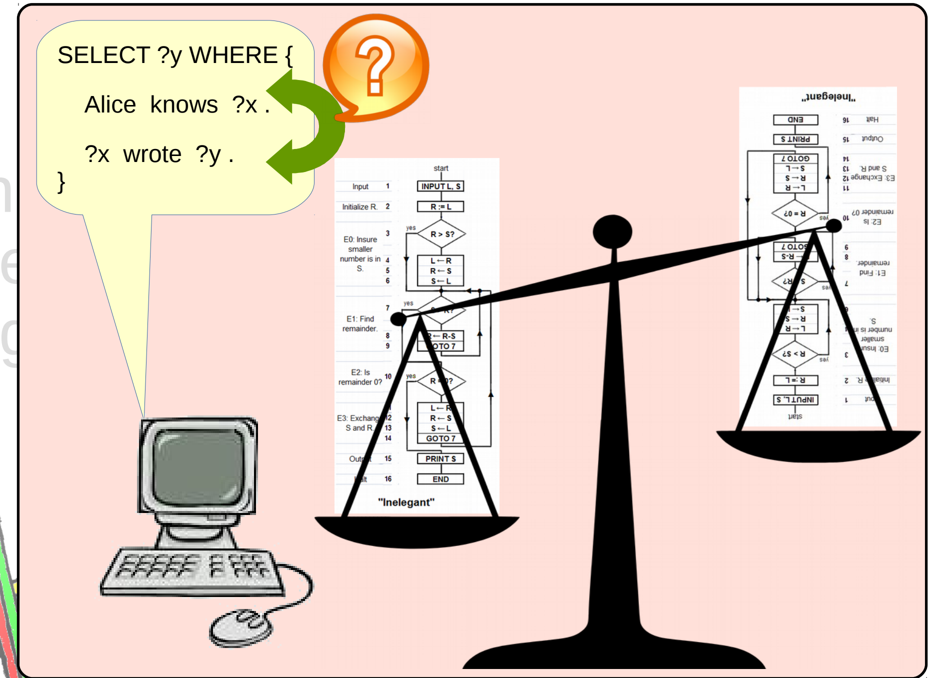
Ingredients (cont'd)

Formal foun
including a de
query lang

Client-server
interfaces

[ISWC'14],
[ODBASE'16],
[JWS'16],
[AMW'17], [ISWC'17]*,
[WWW'18], [IJCAI'18], [iiWAS'20]





Query optimization
techniques

[ESWC'11], [LDOW'11]

Query execution
algorithms

[ISWC'09],
[LDOW'11],
[DBS'13],
[SIGMOD'13]

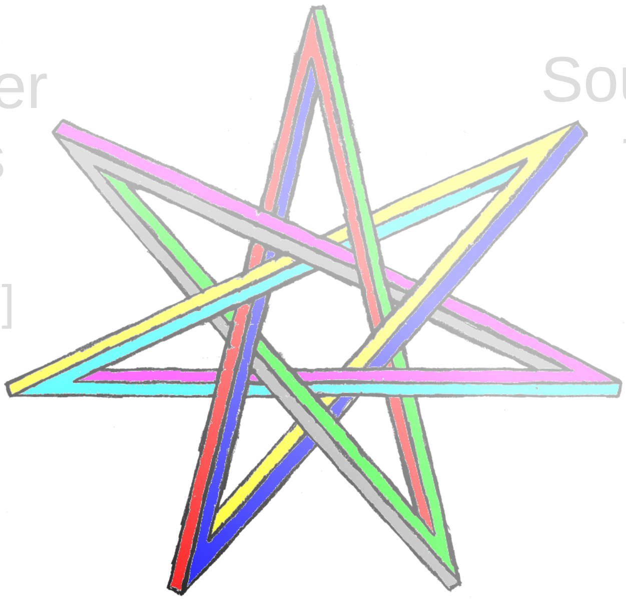
Data provenance
and data quality

Data integration
techniques

*Winner of Best Research Paper Award

Picture source: https://www.deviantart.com/thecelticpoet/art/Seven-Pointed-Impossible-Star-189689250

Ingredients (cont'd)



Formal foundations, including a declarative query language

[ESWC'12], [HT'12], [WWW'14], [AMW'15], [ESWC'15]*, [ISWC'15], [JWS'16], [SWJ'17], [JWE'19]

Client-server interfaces

[ISWC'14], [ODBASE'16], [JWS'16], [AMW'17], [ISWC'17]*, [WWW'18], [IJCAI'18], [iiWAS'20]

Source selection techniques

[ISWC'16], [Semantics'18]*

Query execution algorithms

[ISWC'09], [LDOW'11], [DBS'13], [SIGMOD'13]

Query optimization techniques

[ESWC'11], [LDOW'11]

Data provenance and data quality

[ESWC'09]*, [LDOW'09], [SWPM'09], [IPAW'10], [SPOT'10], [LDOW'12]

Data integration techniques

*Winner of Best Research Paper Award

Picture source: https://www.deviantart.com/thecelticpoet/art/Seven-Pointed-Impossible-Star-189689250

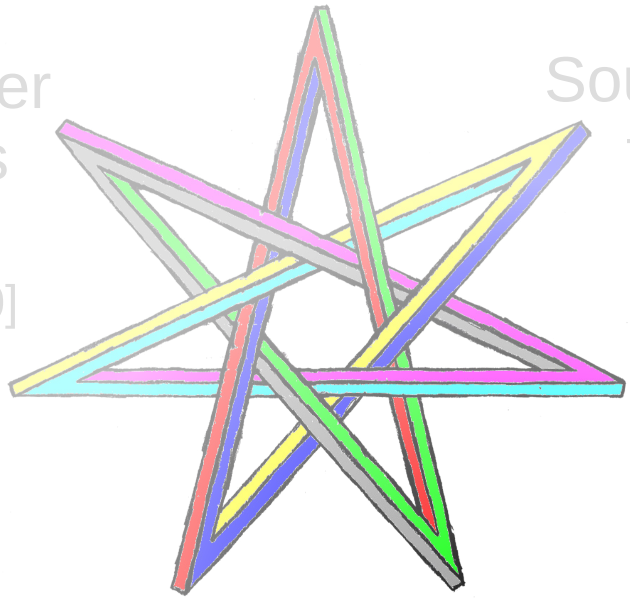
Ingredients (cont'd)

Formal foundations, including a declarative query language

[ESWC'12], [HT'12], [WWW'14], [AMW'15], [ESWC'15]*, [ISWC'15], [JWS'16], [SWJ'17], [JWE'19]

Client-server interfaces

[ISWC'14], [ODBASE'16], [JWS'16], [AMW'17], [ISWC'17]*, [WWW'18], [IJCAI'18], [iiWAS'20]



Source selection techniques

[ISWC'16], [Semantics'18]*

Query execution algorithms

[ISWC'09], [LDOW'11], [DBS'13], [SIGMOD'13]

Query optimization techniques

[ESWC'11], [LDOW'11]

Data provenance and data quality

[ESWC'09]*, [LDOW'09], [SWPM'09], [IPAW'10], [SPOT'10], [LDOW'12]

Data integration techniques [AMW'17]

*Winner of Best Research Paper Award

Picture source: https://www.deviantart.com/theceltiicpoet/art/Seven-Pointed-Impossible-Star-189689250

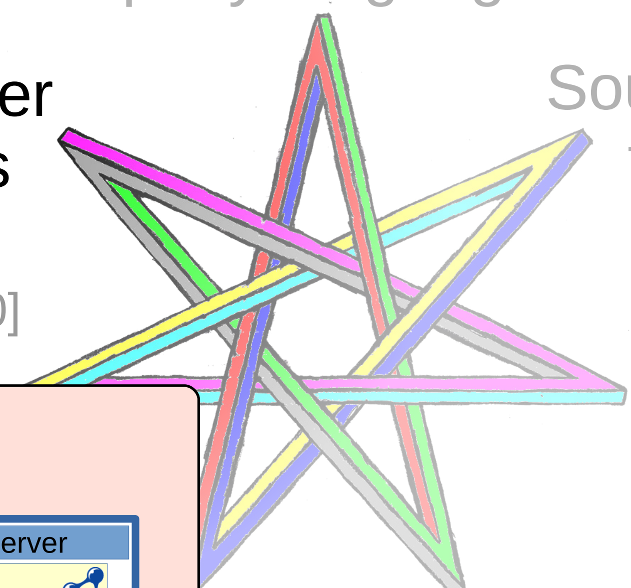
Ingredients (cont'd)

Formal foundations, including a declarative query language

[ESWC'12], [HT'12], [WWW'14], [AMW'15], [ESWC'15]*, [ISWC'15], [JWS'16], [SWJ'17], [JWE'19]

[ISWC'14], [ODBASE'16], [JWS'16], [AMW'17], [ISWC'17]*, [WWW'18], [IJCAI'18], [iiWAS'20]

Client-server interfaces



Source selection techniques

[ISWC'16], [Semantics'18]*

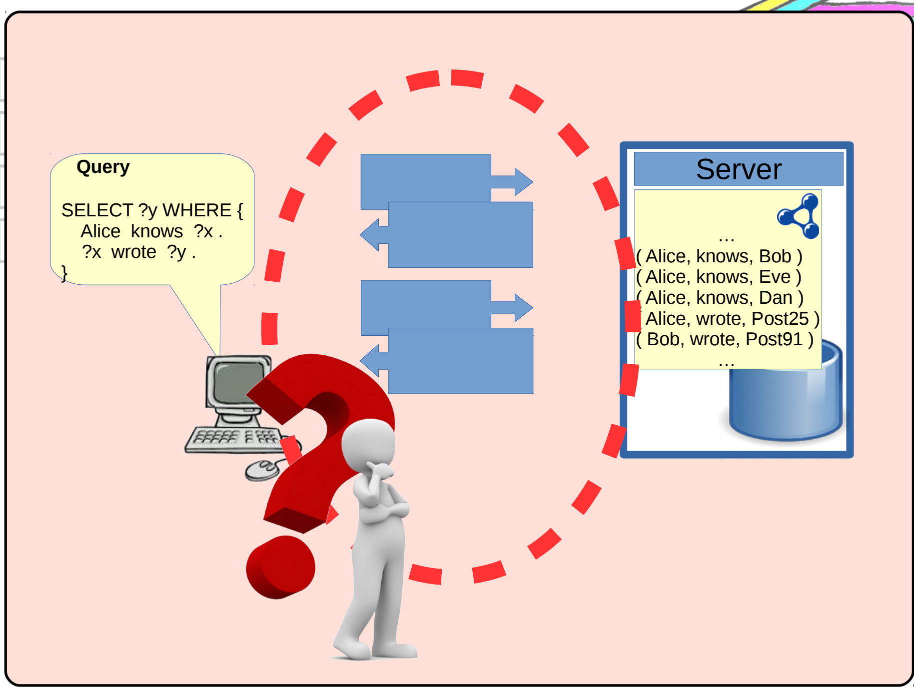
Query optimization techniques

[ESWC'11], [LDOW'11]

ance

ality

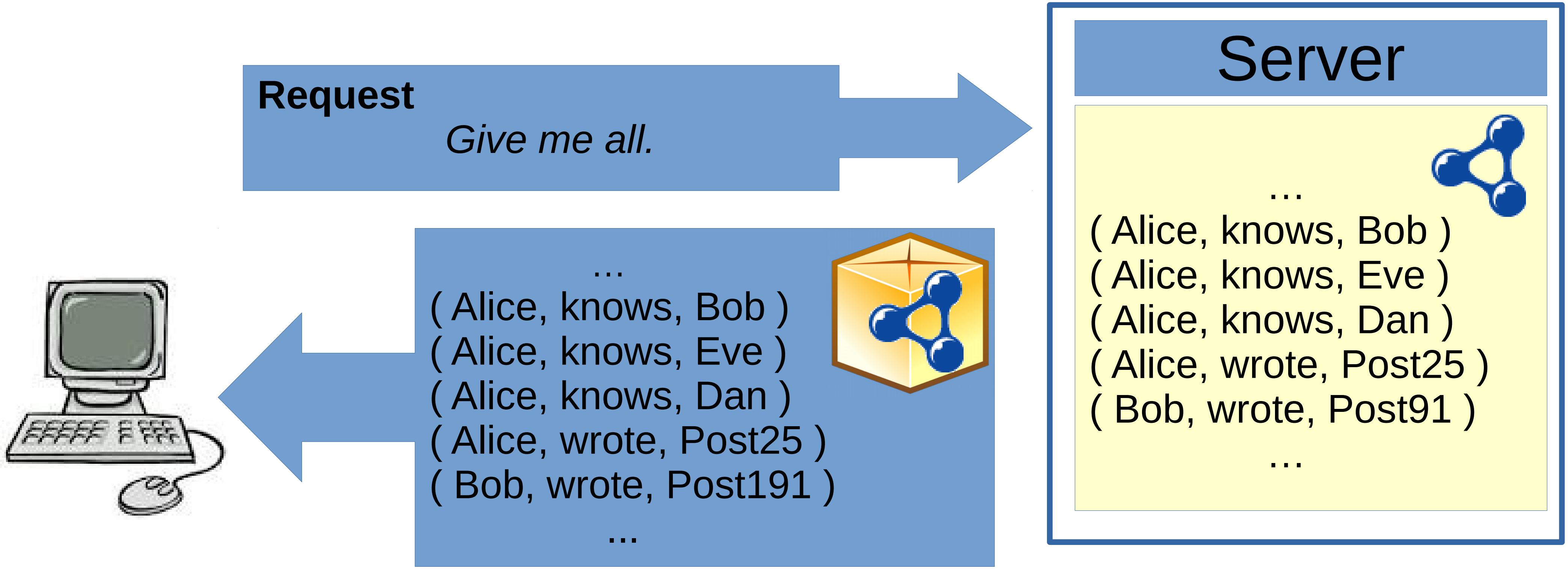
Data integration techniques [AMW'17]



*Winner of Best Research Paper Award

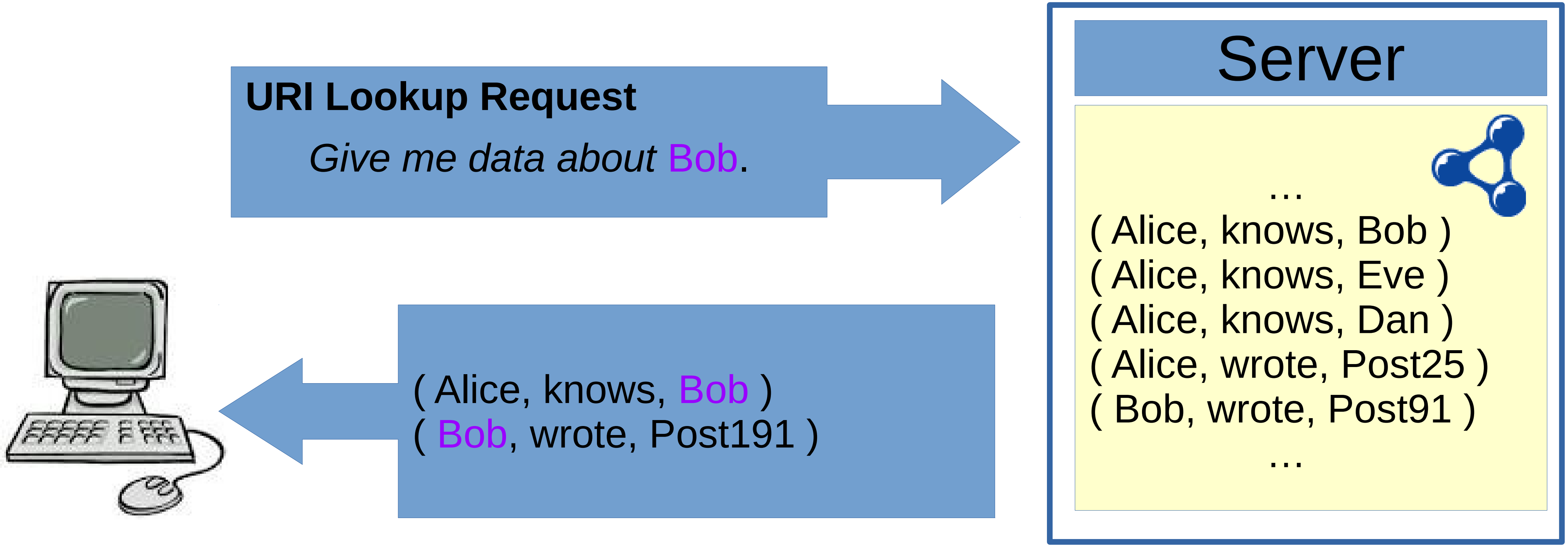
Picture source: https://www.deviantart.com/thecelticpoet/art/Seven-Pointed-Impossible-Star-189689250

Semantic Web Server Interfaces



RDF data dump

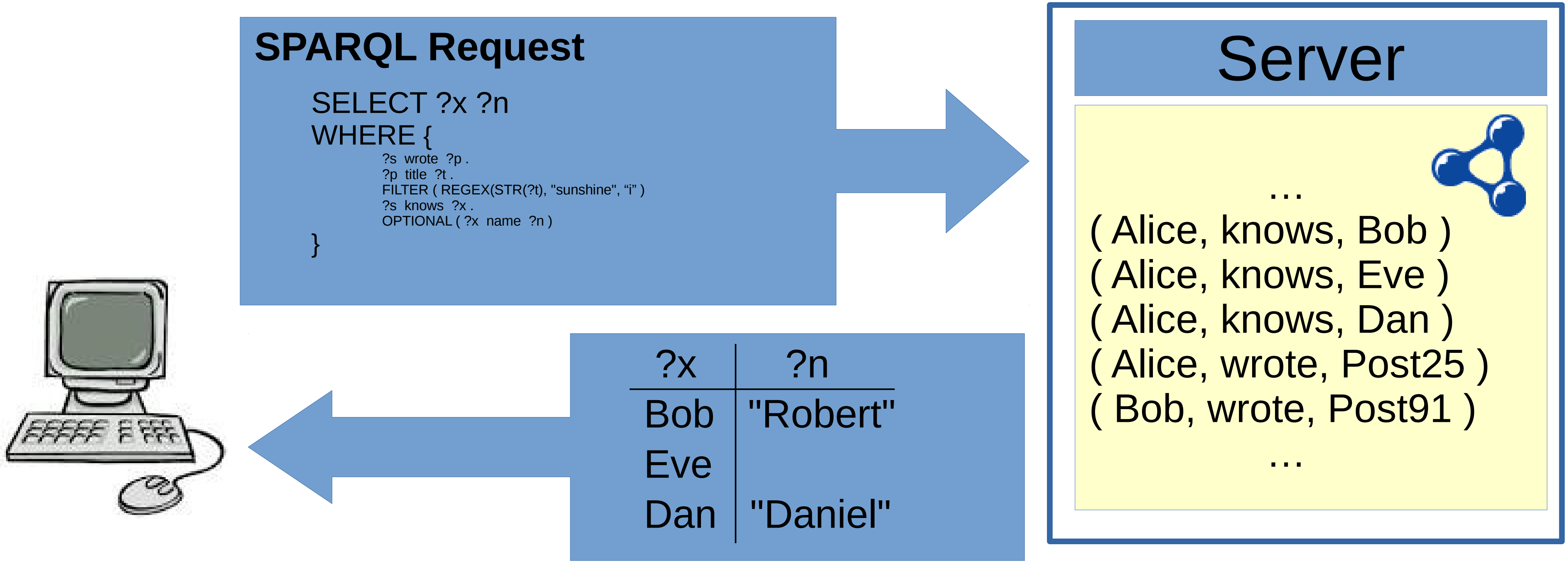
Semantic Web Server Interfaces



RDF data dump

Linked Data documents

Semantic Web Server Interfaces



RDF data dump

Linked Data documents

SPARQL endpoint

Semantic Web Server Interfaces

SPARQL Request

```
SELECT ?x ?n
WHERE {
        ?s  wrote  ?p .
        ?p  title  ?t .
        FILTER ( REGEX(STR(?t), "sunshine", “i” )
        ?s  knows  ?x .
        OPTIONAL ( ?x  name  ?n )
}
```

?x	?n
Bob	"Robert"
Eve	
Dan	"Daniel"

Server

…
(Alice, knows, Bob)
(Alice, knows, Eve)
(Alice, knows, Dan)
(Alice, wrote, Post25)
(Bob, wrote, Post91)
…

SPARQL endpoint

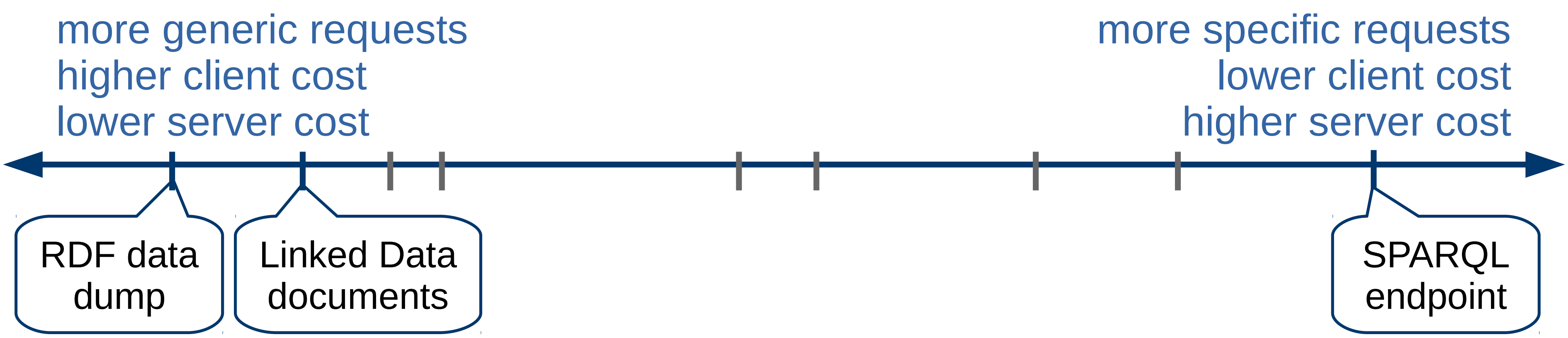
Out of 427 public SPARQL endpoints,
more than half had **<95% availability**.[1]

→ not available for at least 1.5 days each month

[1] C. Buil Aranda, A. Hogan, J. Umbrich, et al.: *SPARQL Web-Querying Infrastructure: Ready for Action?* ISWC 2013.

Linked Data Fragments Framework[1,2]

- Whole spectrum of trade-offs exists between these extremes
- Explore this spectrum and find interesting sweet spots



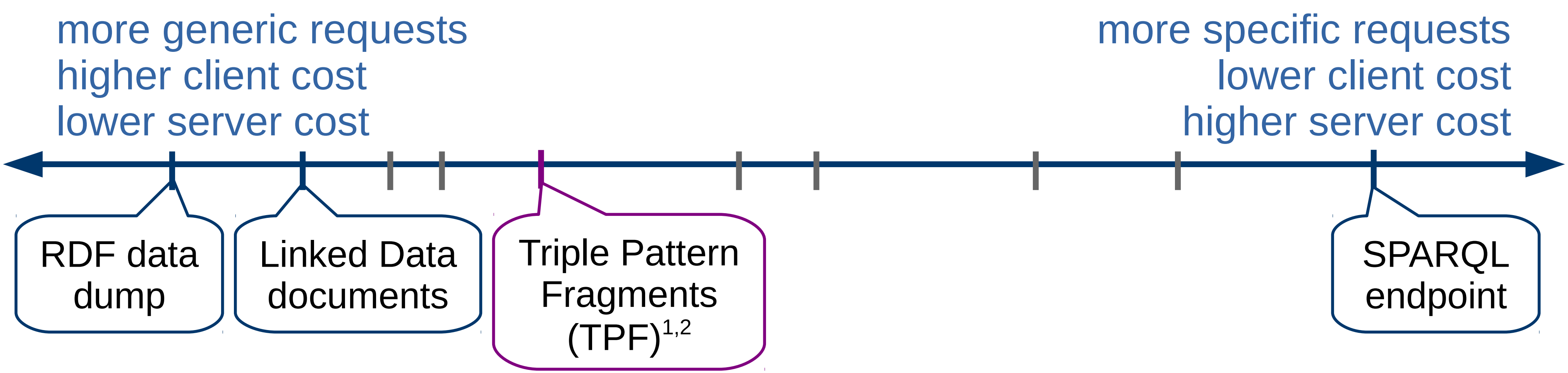
[1] R. Verborgh, **O. Hartig**, B. De Meester, et al.: *Querying Datasets on the Web with High Availability.* ISWC 2014.
[2] R. Verborgh, M. Vander Sande, **O. Hartig**, et al.: *Triple Pattern Fragments: a Low-cost Knowledge Graph Interface for the Web*. In Journal of Web Semantics 37-38, 2016

Triple Pattern Fragments (TPF)[1,2]

[1] R. Verborgh, **O. Hartig**, B. De Meester, et al.: *Querying Datasets on the Web with High Availability.* ISWC 2014.
[2] R. Verborgh, M. Vander Sande, **O. Hartig**, et al.: *Triple Pattern Fragments: a Low-cost Knowledge Graph Interface for the Web*. In Journal of Web Semantics 37-38, 2016

Linked Data Fragments Framework[1,2]

- Whole spectrum of trade-offs exists between these extremes
- Explore this spectrum and find interesting sweet spots

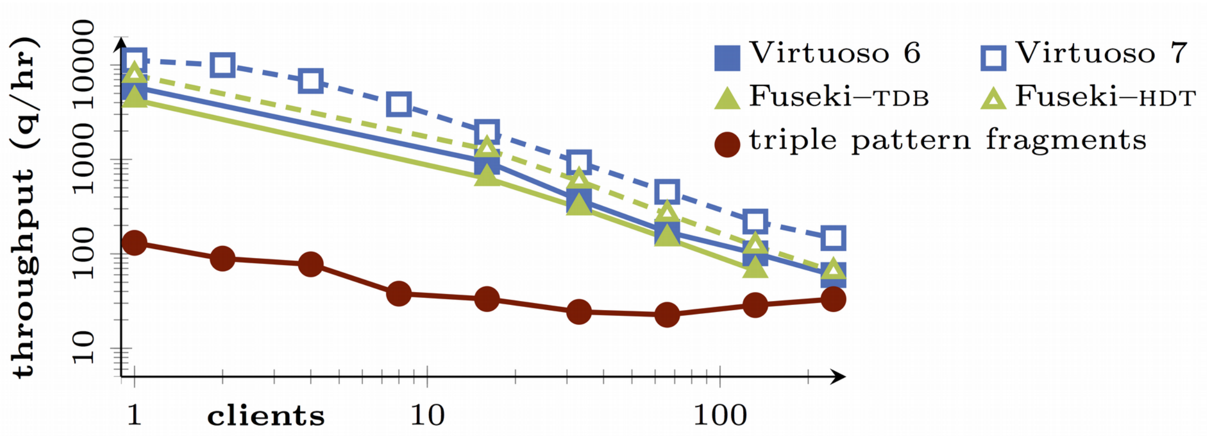


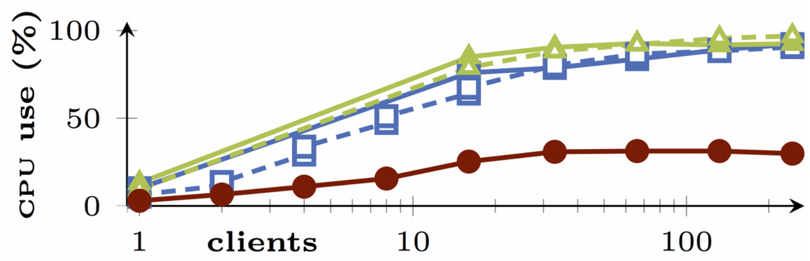
[1] R. Verborgh, **O. Hartig**, B. De Meester, et al.: *Querying Datasets on the Web with High Availability.* ISWC 2014.
[2] R. Verborgh, M. Vander Sande, **O. Hartig**, et al.: *Triple Pattern Fragments: a Low-cost Knowledge Graph Interface for the Web*. In Journal of Web Semantics 37-38, 2016

Summary of Experimental Results

Compared to SPARQL endpoints, query throughput is lower but ...

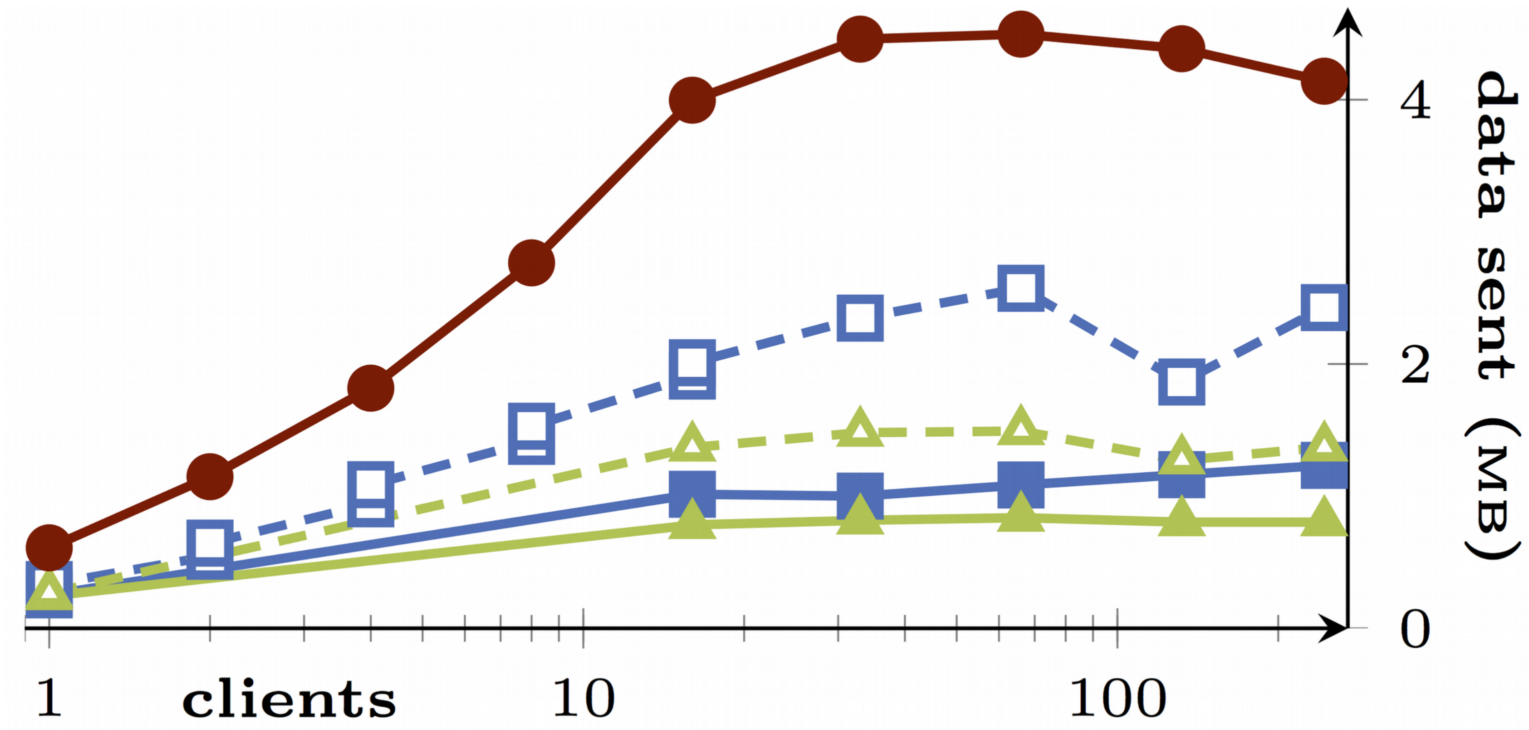
- ...resilient to high client numbers
- ...server-side load is much smaller and more regular (which allows for a higher availability, in particular on small, less expensive servers!)





[1] R. Verborgh, **O. Hartig**, B. De Meester, et al.: *Querying Datasets on the Web with High Availability.* ISWC 2014.
[2] R. Verborgh, M. Vander Sande, **O. Hartig**, et al.: *Triple Pattern Fragments: a Low-cost Knowledge Graph Interface for the Web*. In Journal of Web Semantics 37-38, 2016

Server Traffic (TPF vs. SPARQL Endpoints)



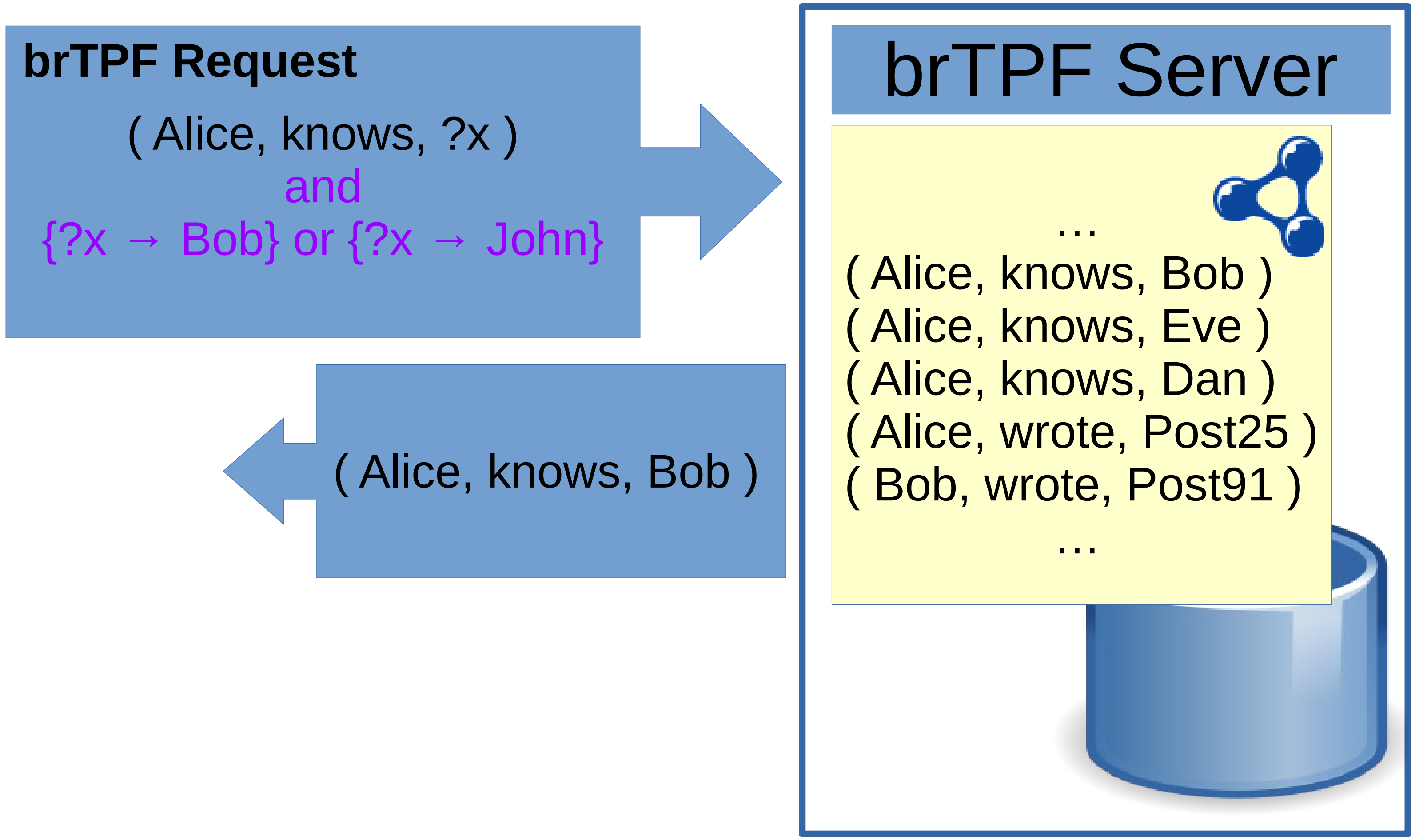
Observation: server traffic is higher

[1] R. Verborgh, **O. Hartig**, B. De Meester, et al.: *Querying Datasets on the Web with High Availability.* ISWC 2014.
[2] R. Verborgh, M. Vander Sande, **O. Hartig**, et al.: *Triple Pattern Fragments: a Low-cost Knowledge Graph Interface for the Web*. In Journal of Web Semantics 37-38, 2016

Bindings-Restricted Triple Pattern Fragments (brTPF)[1]

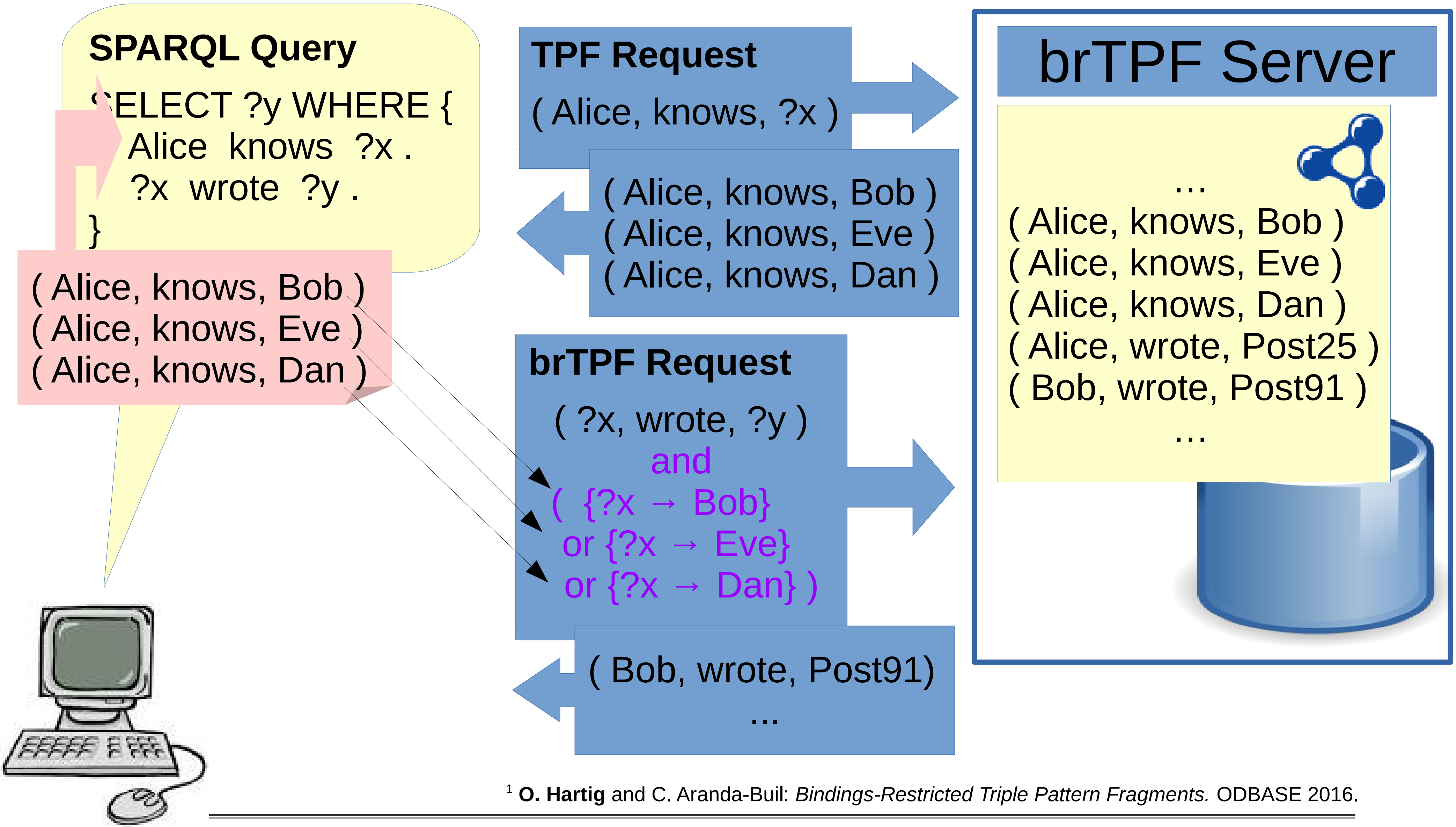
[1] **O. Hartig** and C. Aranda-Buil: *Bindings-Restricted Triple Pattern Fragments.* ODBASE 2016.

brTPF Interface



[1] **O. Hartig** and C. Aranda-Buil: *Bindings-Restricted Triple Pattern Fragments.* ODBASE 2016.

Example Scenario Revisited

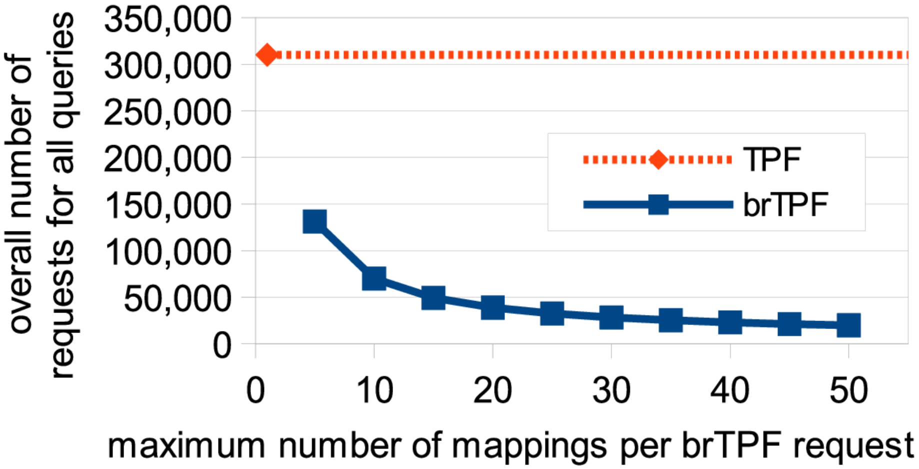


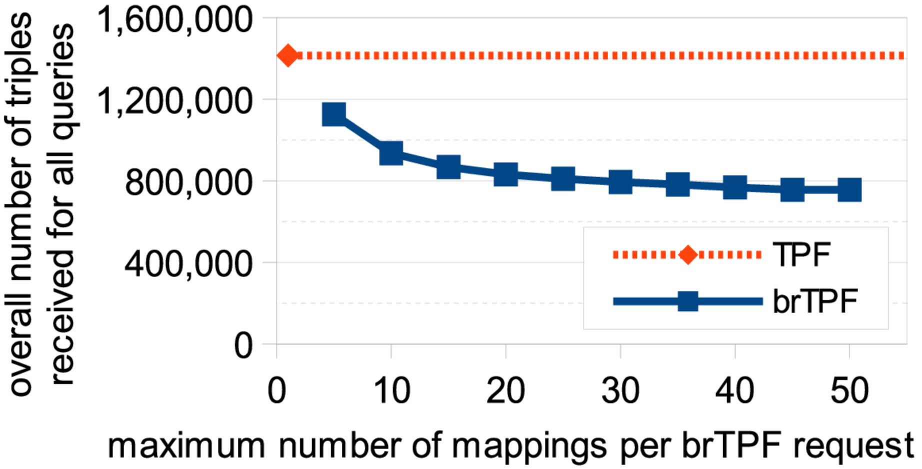
[1] **O. Hartig** and C. Aranda-Buil: *Bindings-Restricted Triple Pattern Fragments.* ODBASE 2016.

Experimental Results

Network load

- brTPF achieves significantly smaller number of requests than TPF, and less data is transferred





[1] **O. Hartig** and C. Aranda-Buil: *Bindings-Restricted Triple Pattern Fragments.* ODBASE 2016.

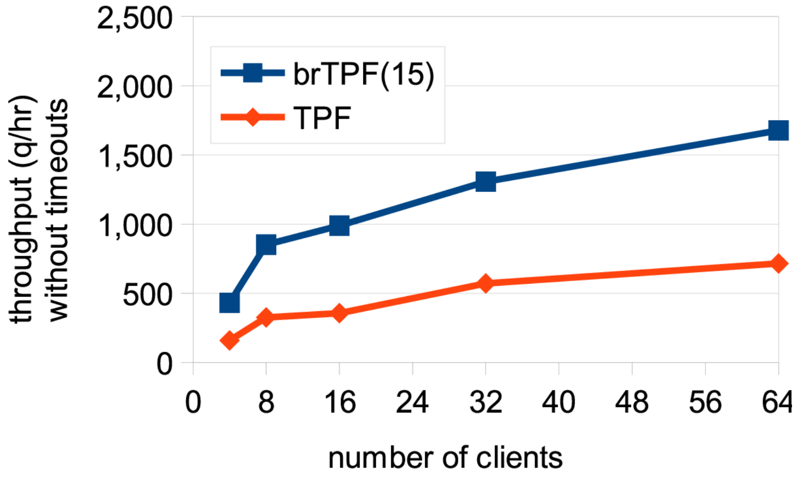
Experimental Results

Network load

- brTPF achieves significantly smaller number of requests than TPF, and less data is transferred

Performance under load

- Both approaches scale to an increasing number of clients, brTPF has a superior scaling behavior
- brTPF achieves a greater throughput than TPF



[1] **O. Hartig** and C. Aranda-Buil: *Bindings-Restricted Triple Pattern Fragments.* ODBASE 2016.

Experimental Results

Network load

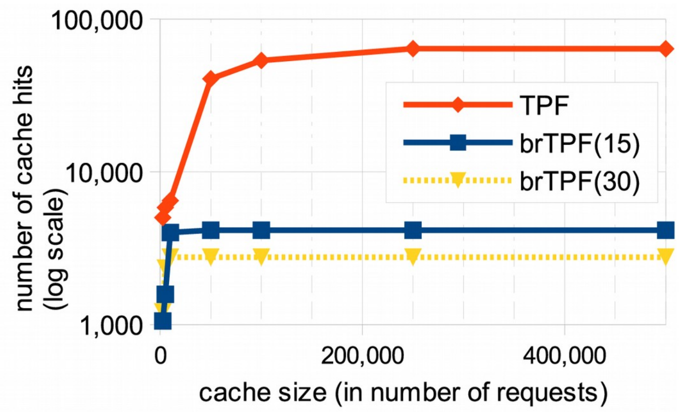
- brTPF achieves significantly smaller number of requests than TPF, and less data is transferred

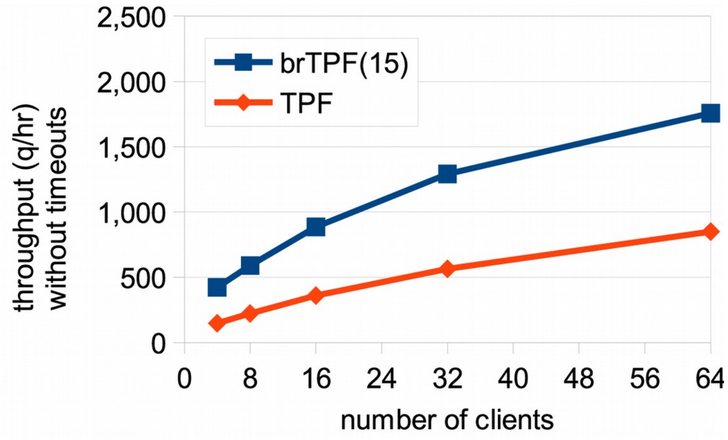
Performance under load

- Both approaches scale to an increasing number of clients, brTPF has a superior scaling behavior
- brTPF achieves a greater throughput than TPF

Impact of HTTP caching

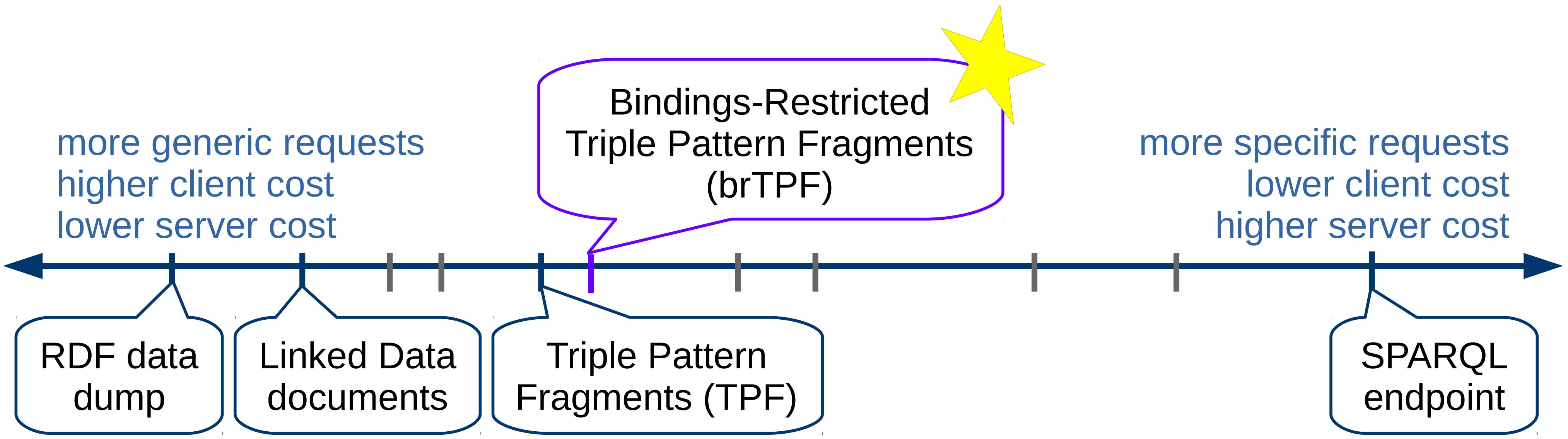
- TPF requests have a higher cache-hit likelihood
- Caching does not help TPF to outperform brTPF





Comparison

Compared to TPF-based executions of SPARQL queries, by using the brTPF interface instead, we can achieve a *reduced network load* as well as an *increased throughput*.



A More Fundamental Understanding

- Server: Can we analyze an interface before actually implementing it?
- Client: Can a given query be executed at all by using a specific interface?

image source: https://www.maxpixel.net/Characters-Question-Response-Question-Mark-Symbol-1015308

A Model of Distributed Query Computation in Client-Server Scenarios

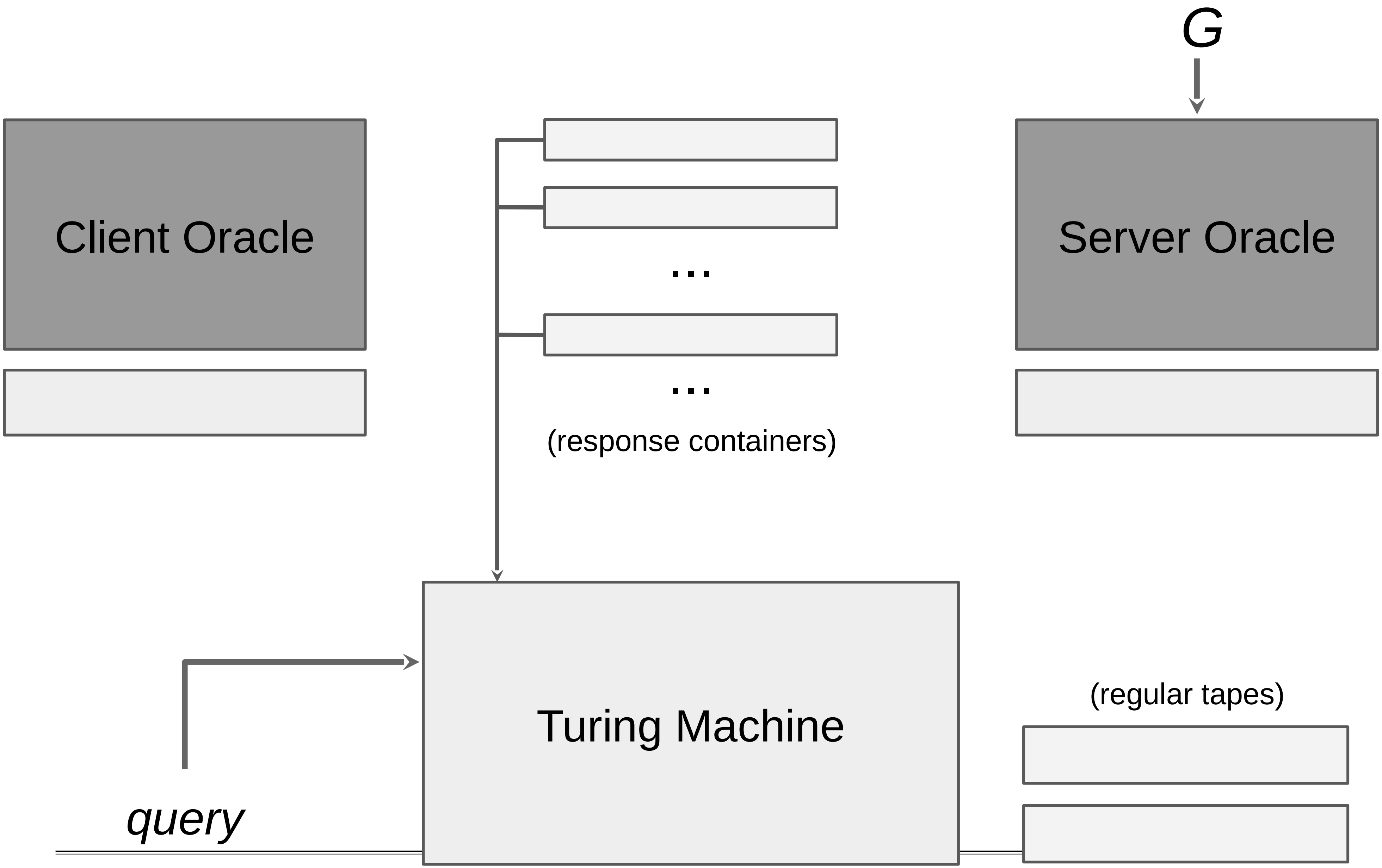
O. Hartig, I. Letter, J. Pérez: *A Formal Framework for Comparing Linked Data Fragments.* ISWC 2017. (Best Research Paper Award winner)
O. Hartig, I. Letter, J. Pérez: *A Model of Distributed Query Computation in Client-Server Scenarios.* IJCAI 2018. (invited paper)

Main Contributions

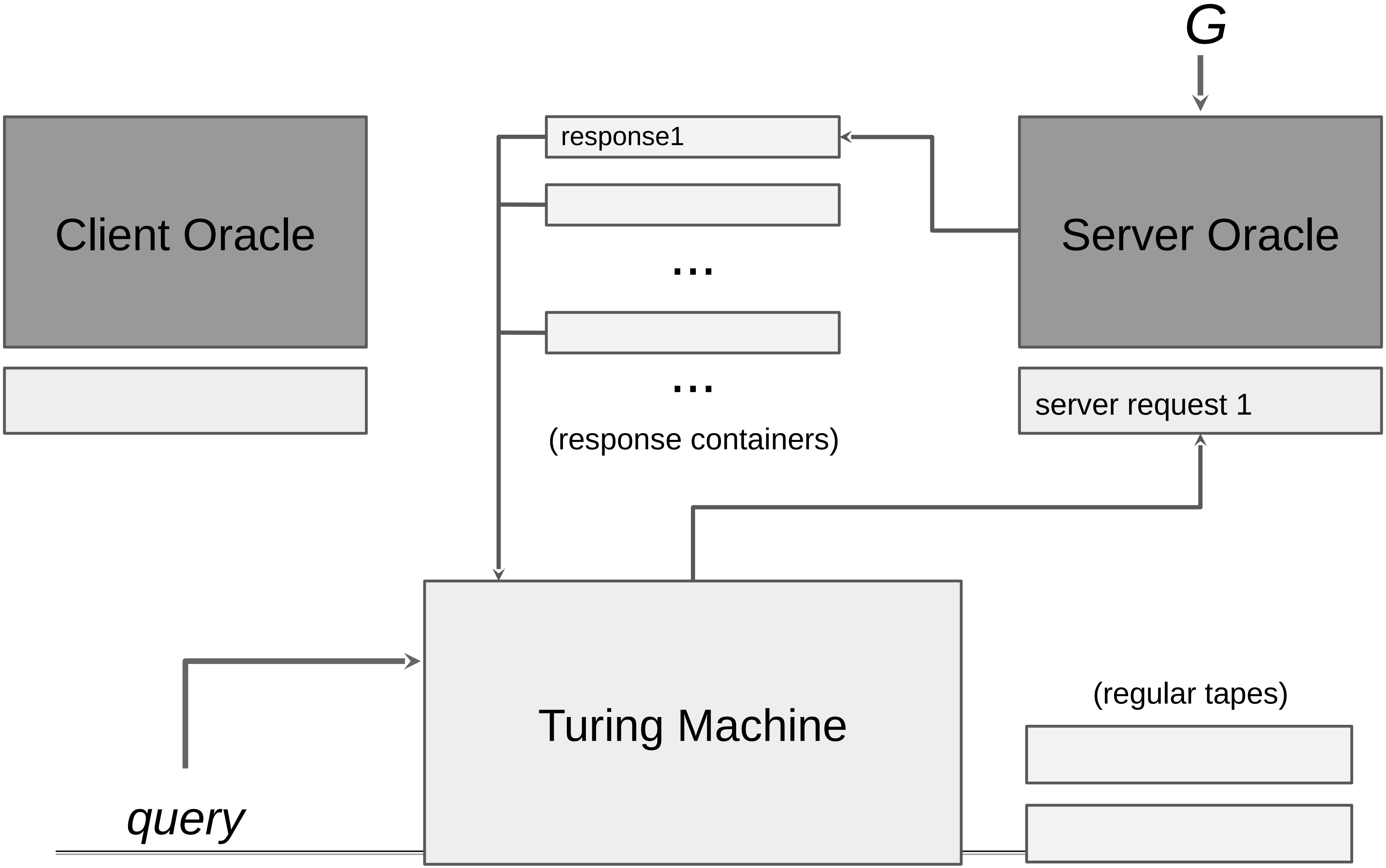
- **Formal machine model for LDF settings**
  based on Turing Machines
- **Complete expressiveness lattice**
  considering several combinations of interfaces
- **Fine-grained complexity analysis**
  classical complexity, # of requests, data transferred

O. Hartig, I. Letter, J. Pérez: *A Formal Framework for Comparing Linked Data Fragments.* ISWC 2017. (Best Research Paper Award winner)
O. Hartig, I. Letter, J. Pérez: *A Model of Distributed Query Computation in Client-Server Scenarios.* IJCAI 2018. (invited paper)

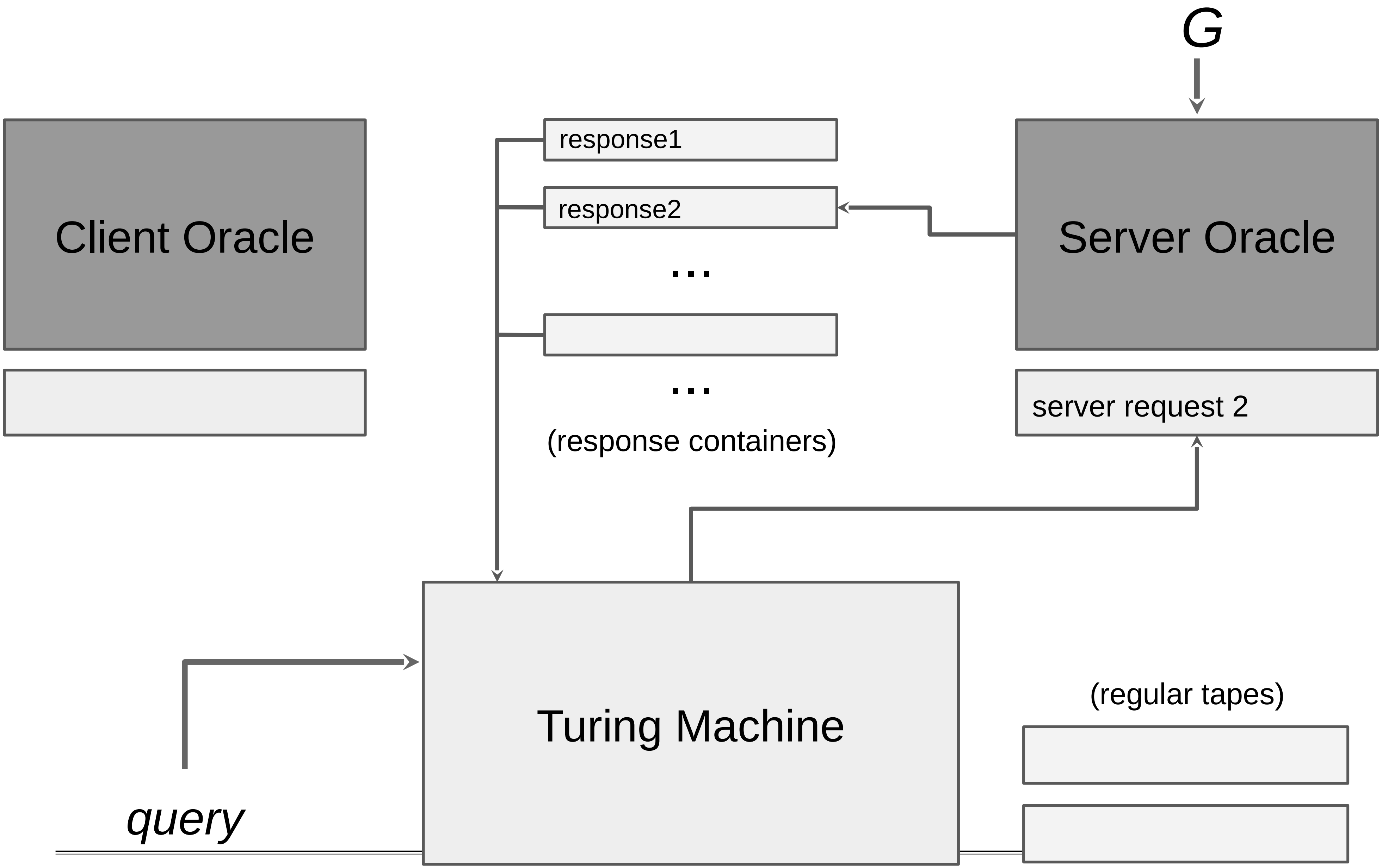
Linked Data Fragments Machine (LDFM)



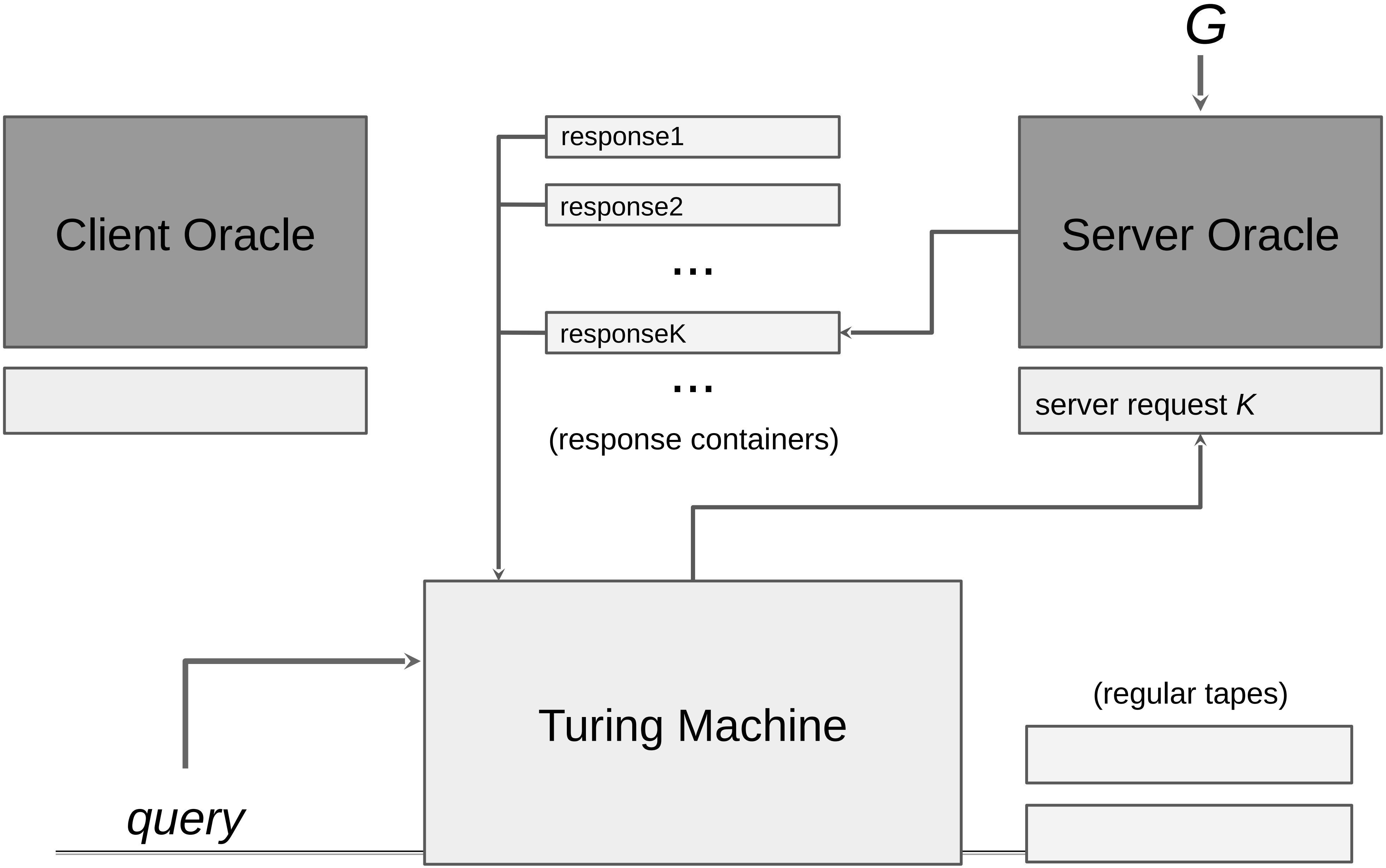
Linked Data Fragments Machine (LDFM)



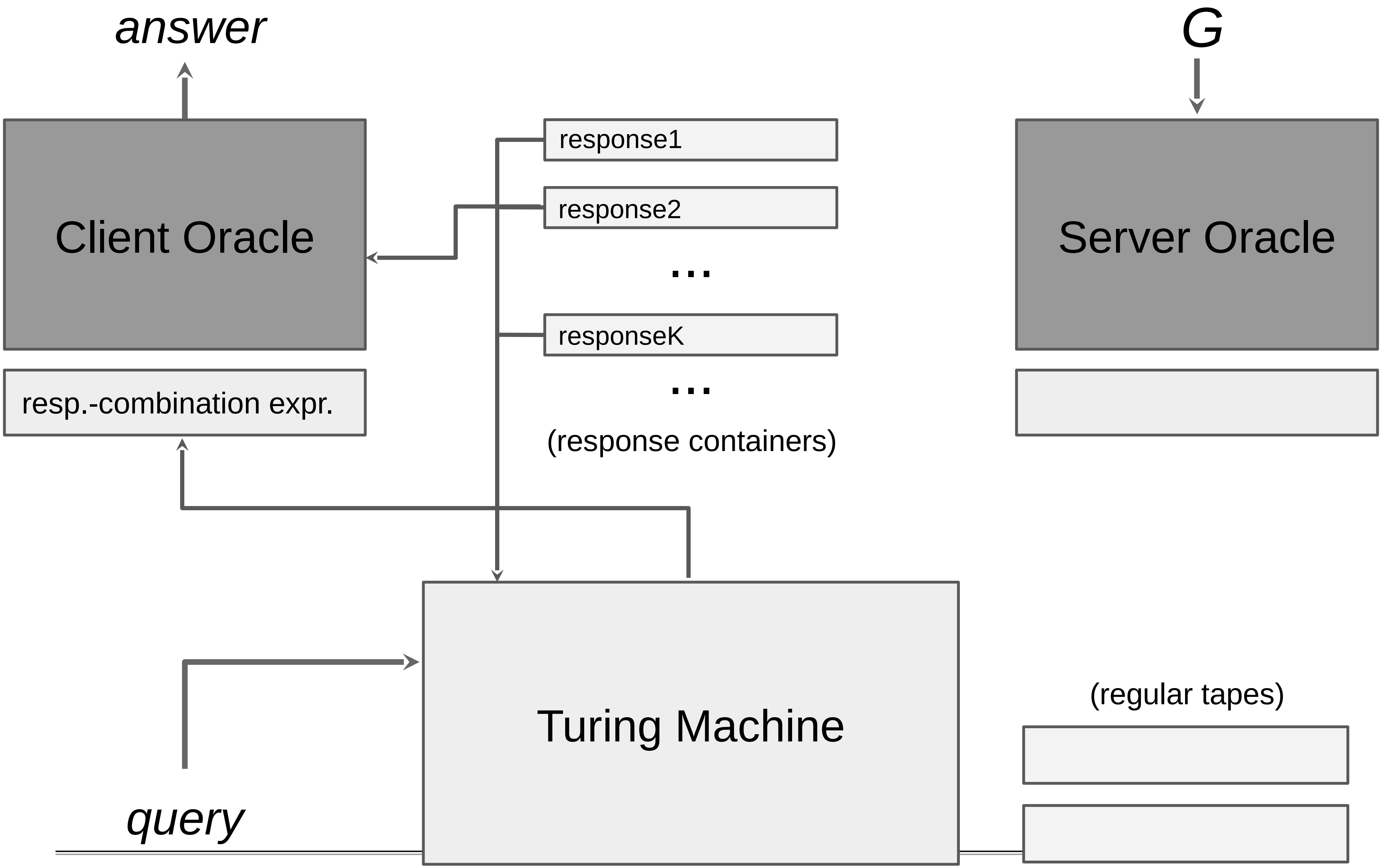
Linked Data Fragments Machine (LDFM)



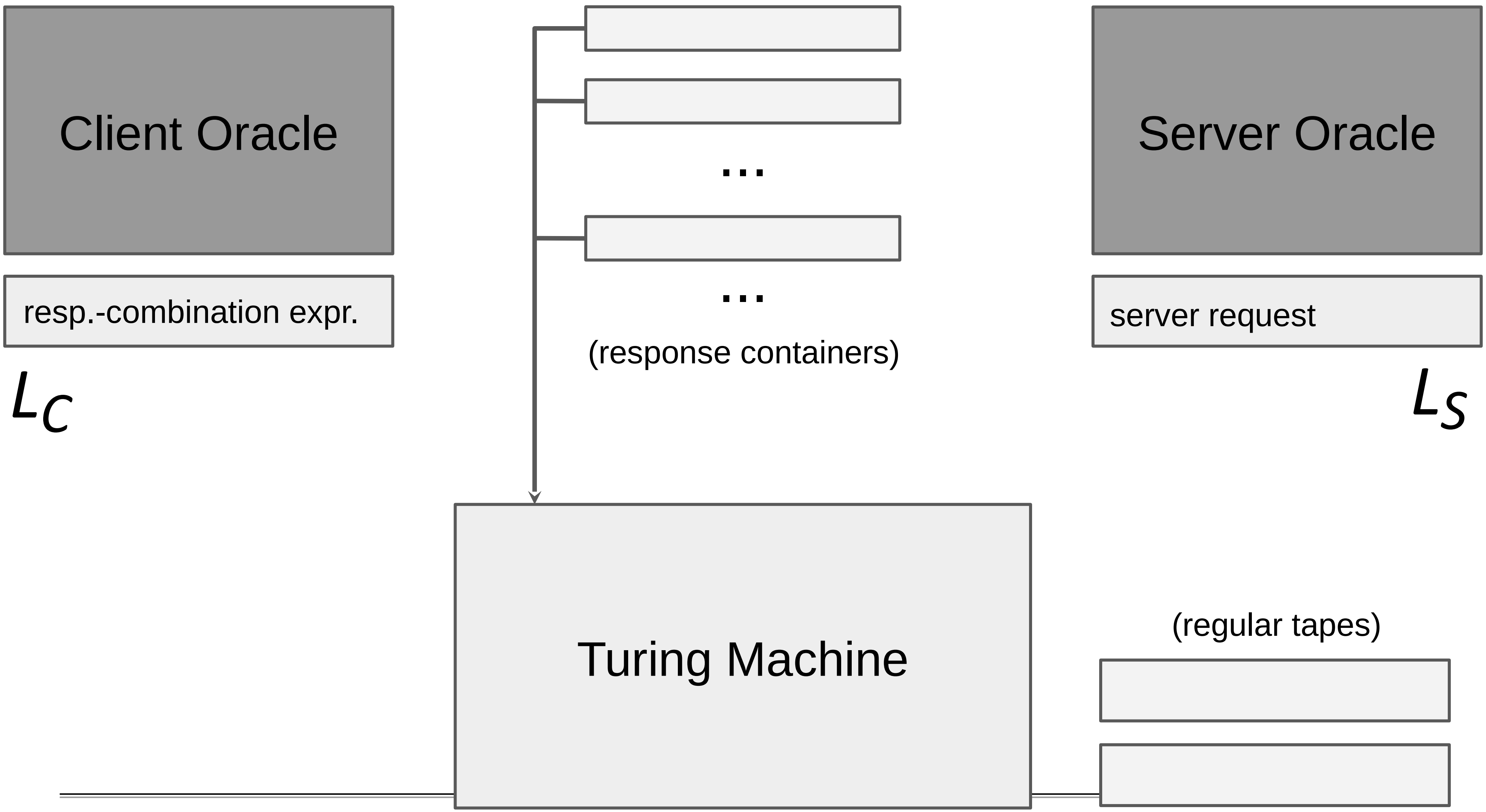
Linked Data Fragments Machine (LDFM)



Linked Data Fragments Machine (LDFM)

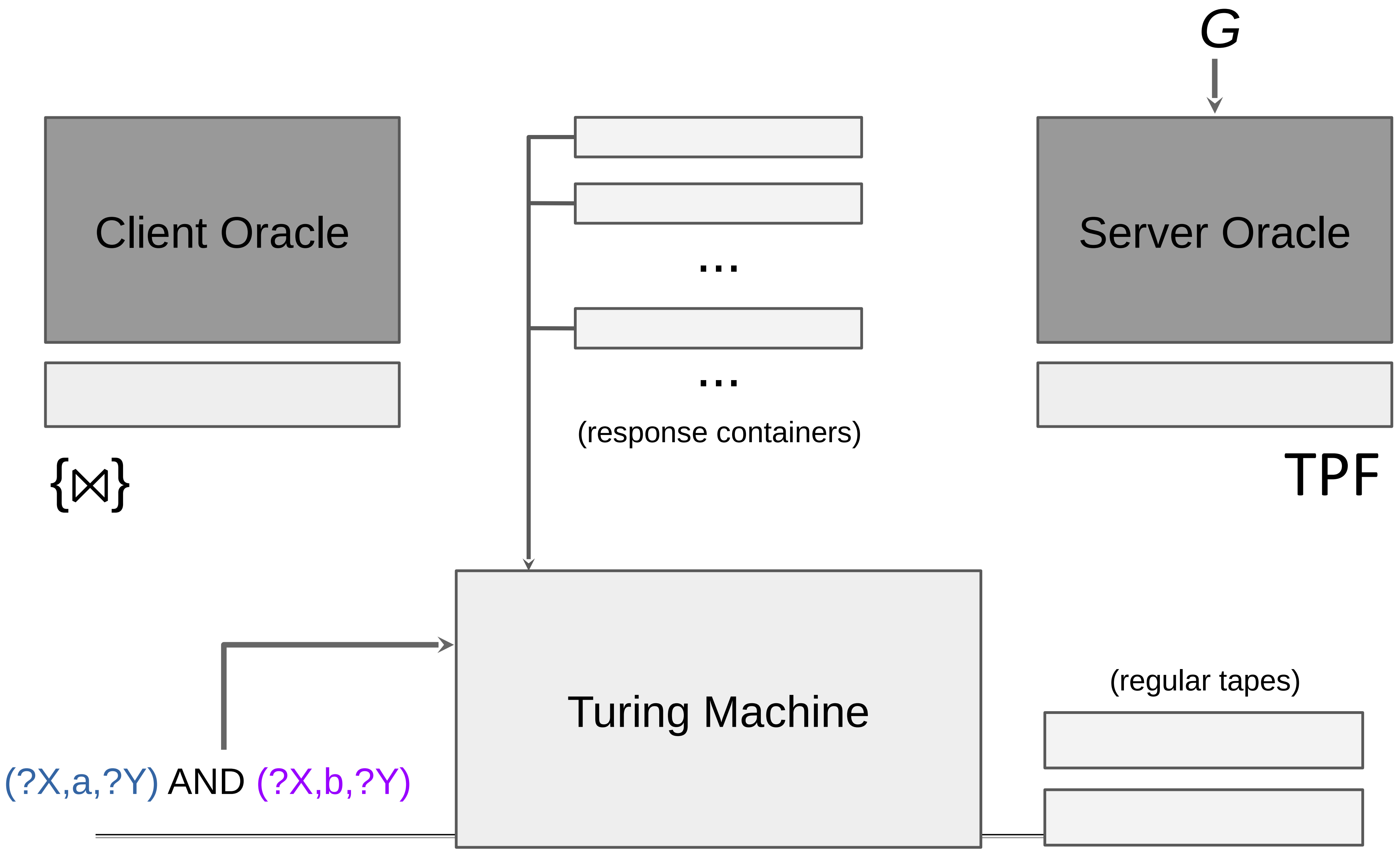


(L_C, L_S)-LDFM



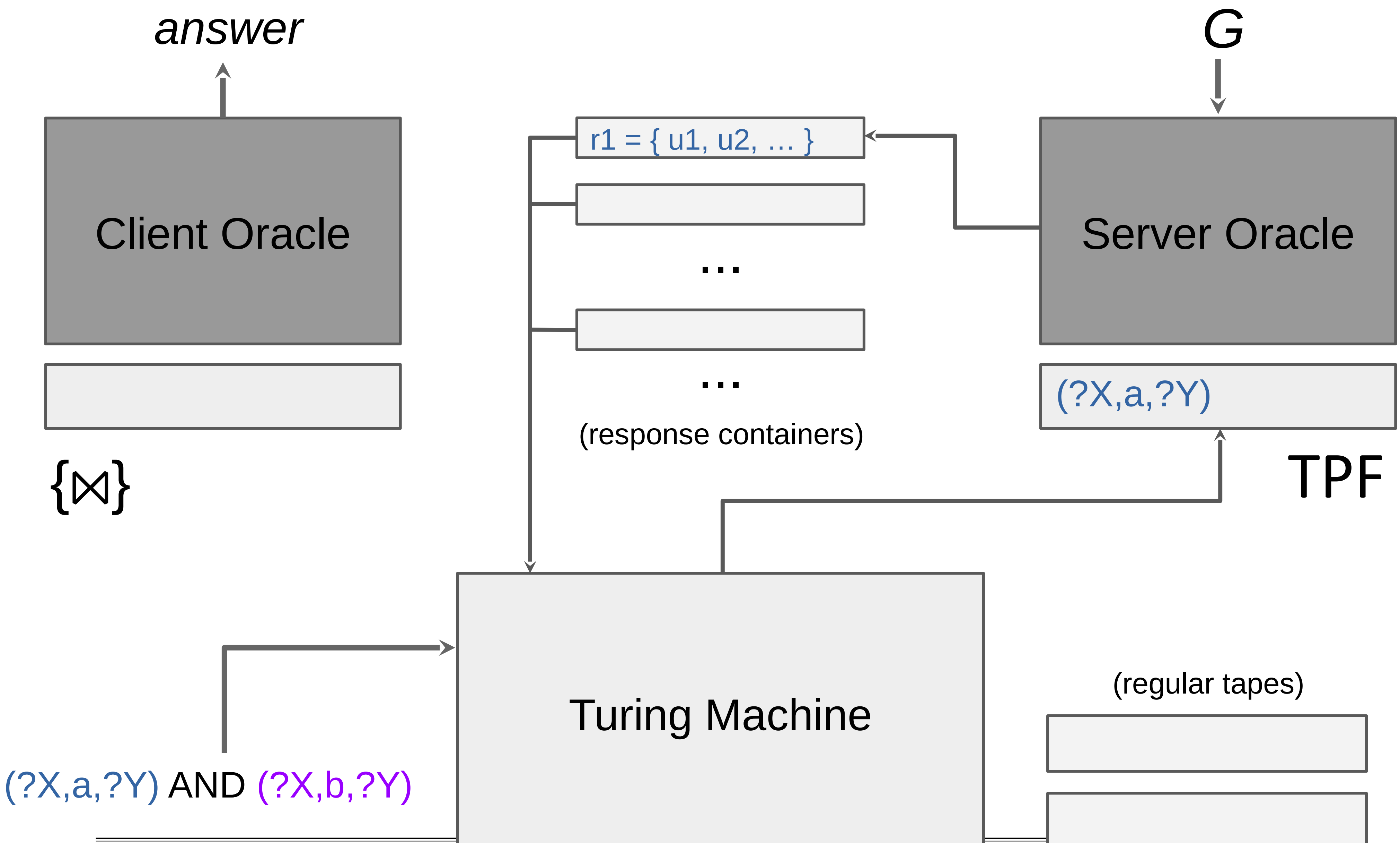
Client Oracle
resp.-combination expr.
L_C
…
…
(response containers)
Server Oracle
server request
L_S
Turing Machine
(regular tapes)

({⋈},TPF)-LDFM

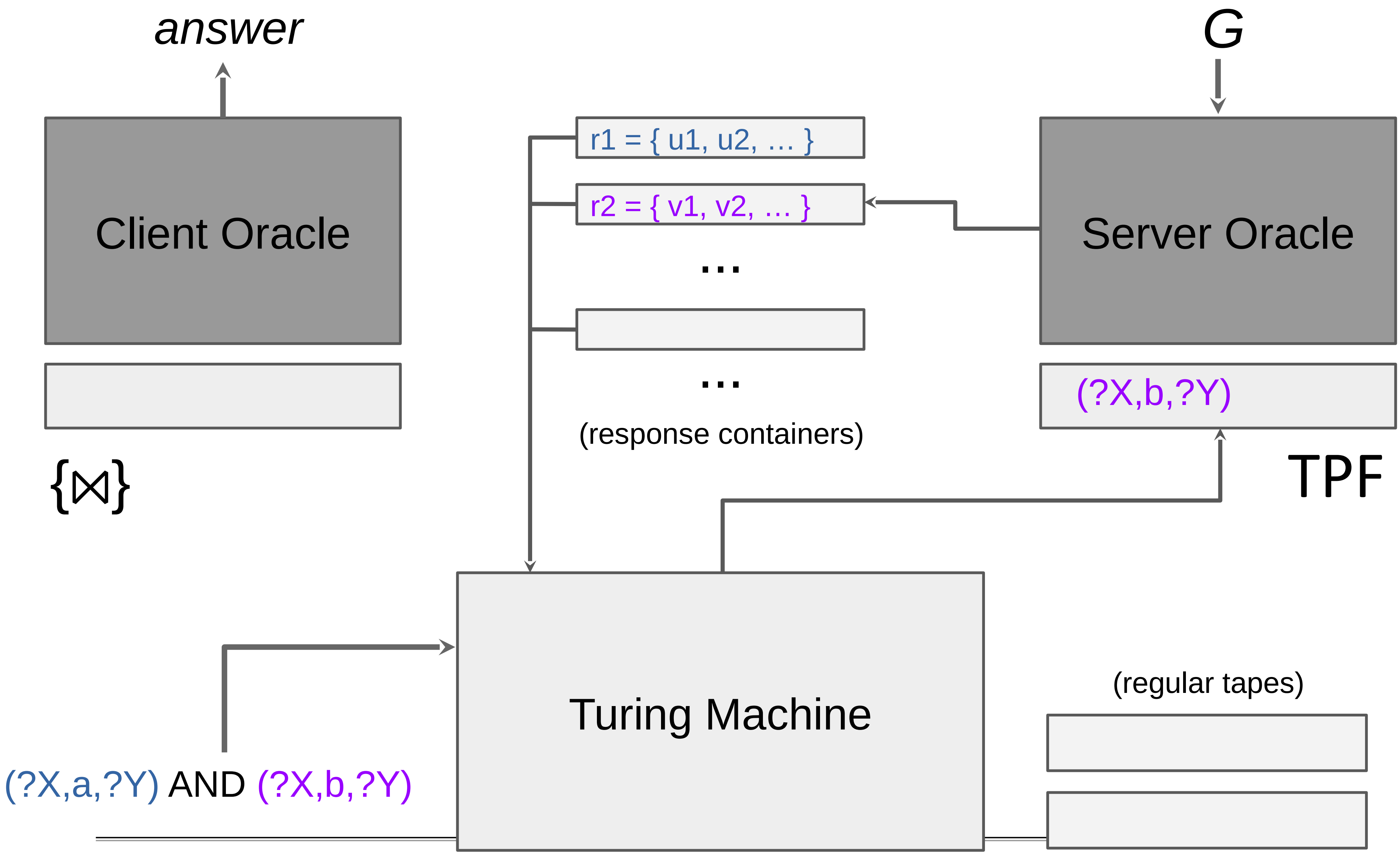


G
Client Oracle
…
…
(response containers)
Server Oracle
{⋈}
TPF
Turing Machine
(regular tapes)
(?X,a,?Y) AND (?X,b,?Y)

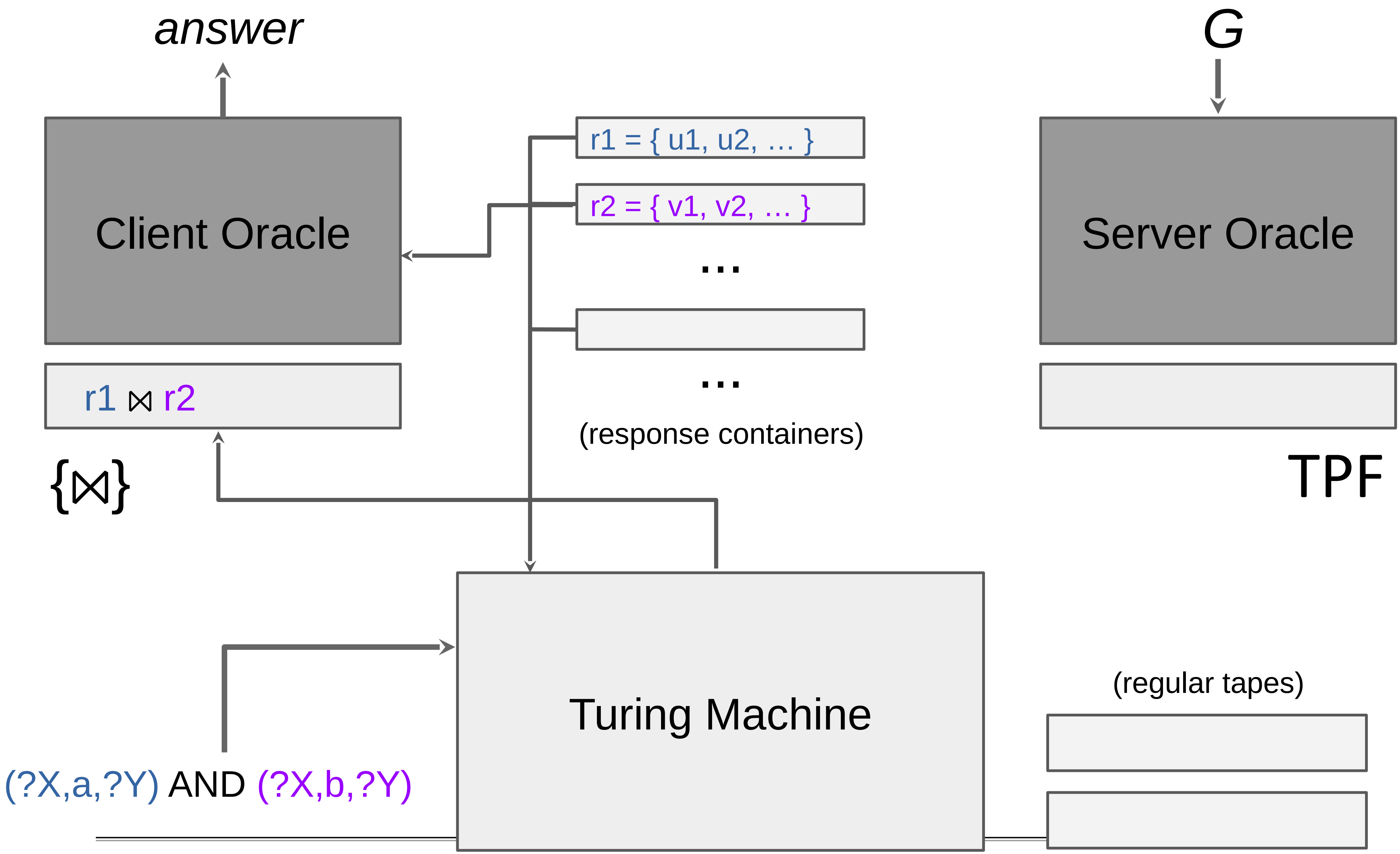
({⋈},TPF)-LDFM


answer
G
Client Oracle
r1 = { u1, u2, … }
Server Oracle
…
…
(response containers)
(?X,a,?Y)
{⋈}
TPF
Turing Machine
(regular tapes)
(?X,a,?Y) AND (?X,b,?Y)

({⋈},TPF)-LDFM


answer
G
Client Oracle
Server Oracle
r1 = { u1, u2, … }
r2 = { v1, v2, … }
…
…
(response containers)
(?X,b,?Y)
{⋈}
TPF
Turing Machine
(regular tapes)
(?X,a,?Y) AND (?X,b,?Y)

({⋈},TPF)-LDFM

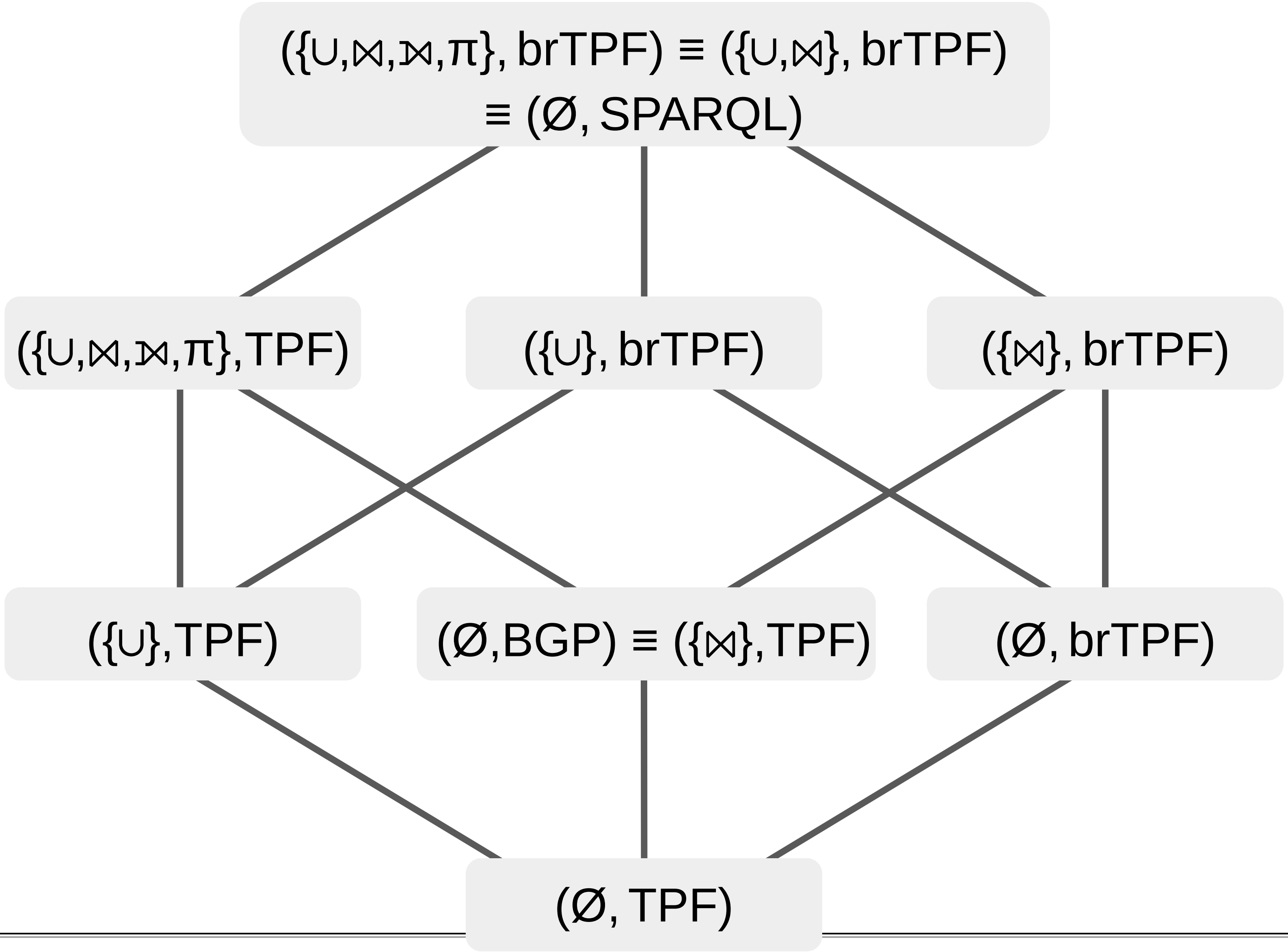

answer
G
Client Oracle
r1 = { u1, u2, … }
r2 = { v1, v2, … }
…
Server Oracle
r1 ⋈ r2
…
(response containers)
{⋈}
TPF
Turing Machine
(regular tapes)
(?X,a,?Y) AND (?X,b,?Y)

What are the queries that can be computed by (L_C,L_S)-LDFMs?

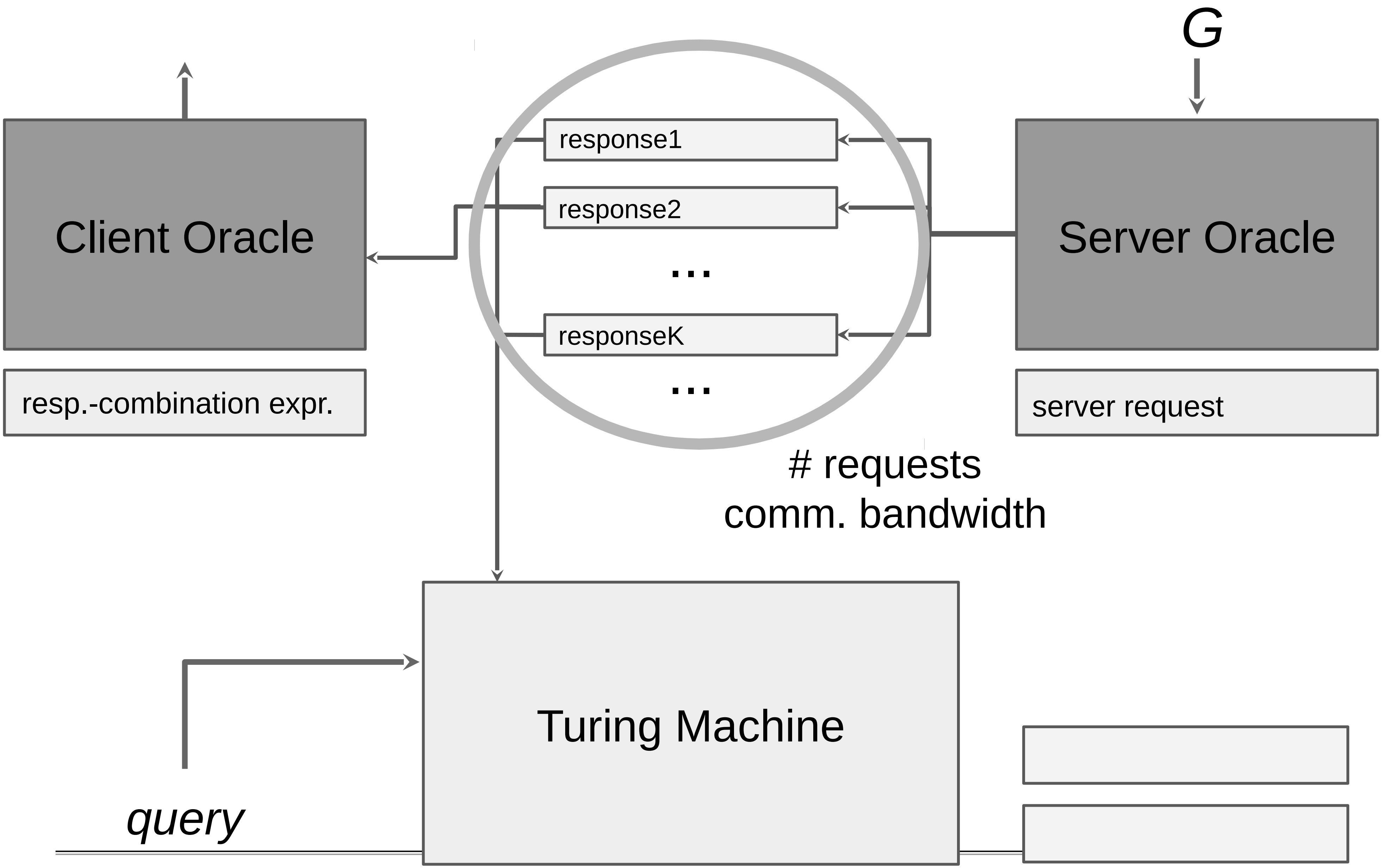
$$(L_C, L_S) \equiv_e (R_C, R_S)$$

$$(L_C, L_S) \prec_e (R_C, R_S)$$

Expressiveness Lattice



Additional Complexity Measures



How do different language combinations compare?

$$(L_C, L_S) \prec_T (R_C, R_S)$$

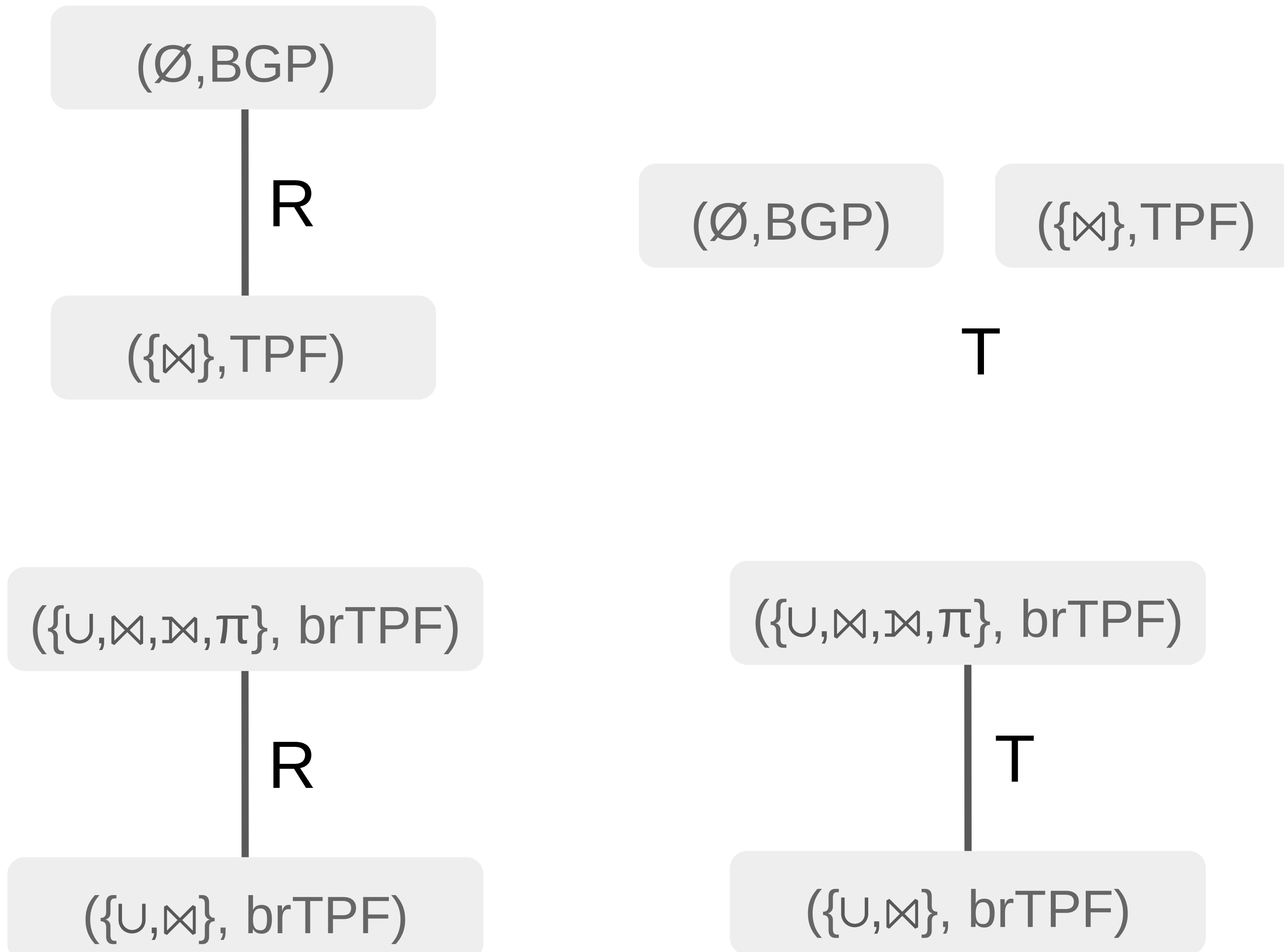
in terms of

|resp1| + |resp2| + … + |respK|

$$(L_C, L_S) \prec_R (R_C, R_S)$$

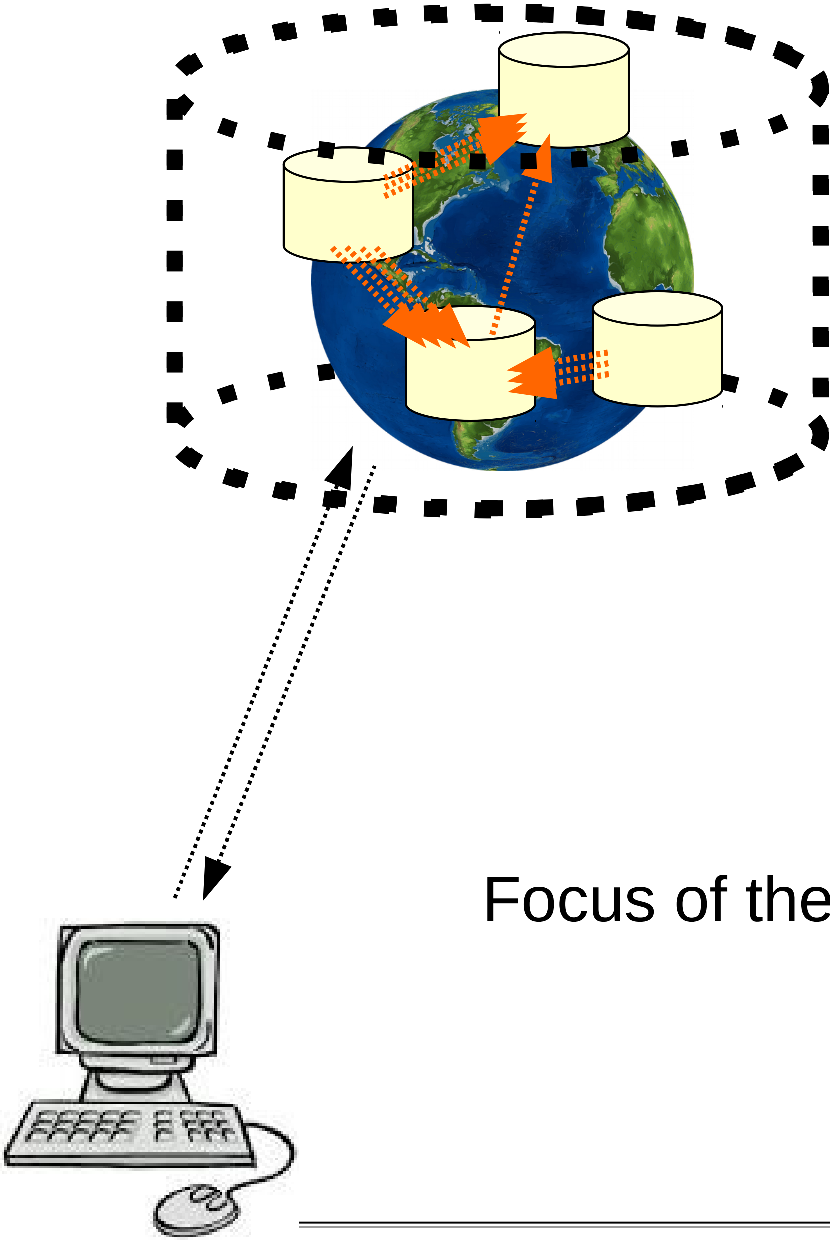
in terms of K

Comparison



What we have seen so far ...

Options we have seen so far



- SPARQL endpoints
- SPARQL over other types of query-based data access interfaces
  - Triple Pattern Fragment (TPF) interfaces
  - Bindings-Restricted TPF (brTPF) interfaces

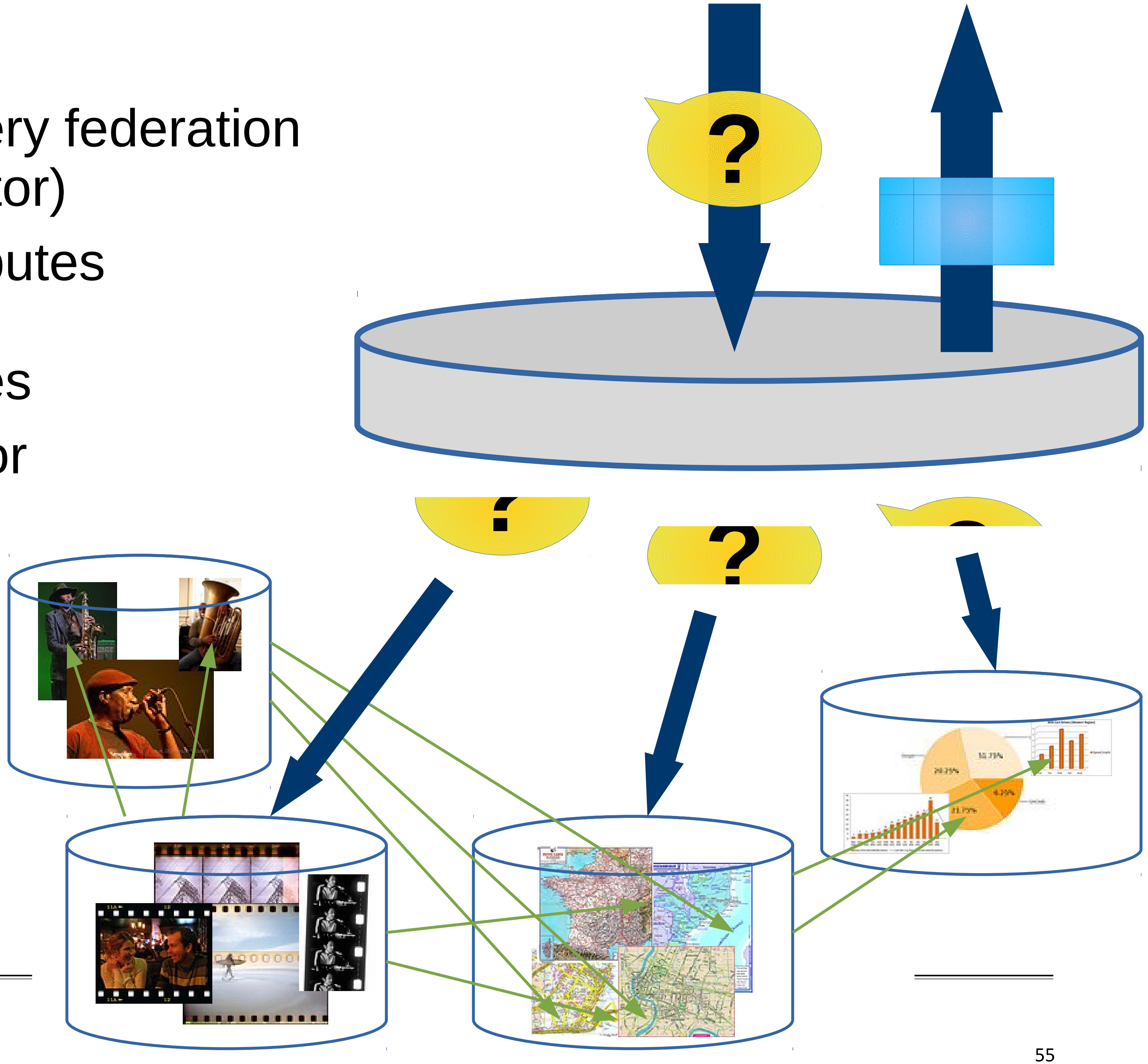
Focus of these: queries over a single dataset

Picture source: https://pixabay.com/en/world-earth-planet-globe-map-1301744/

Federated Query Processing

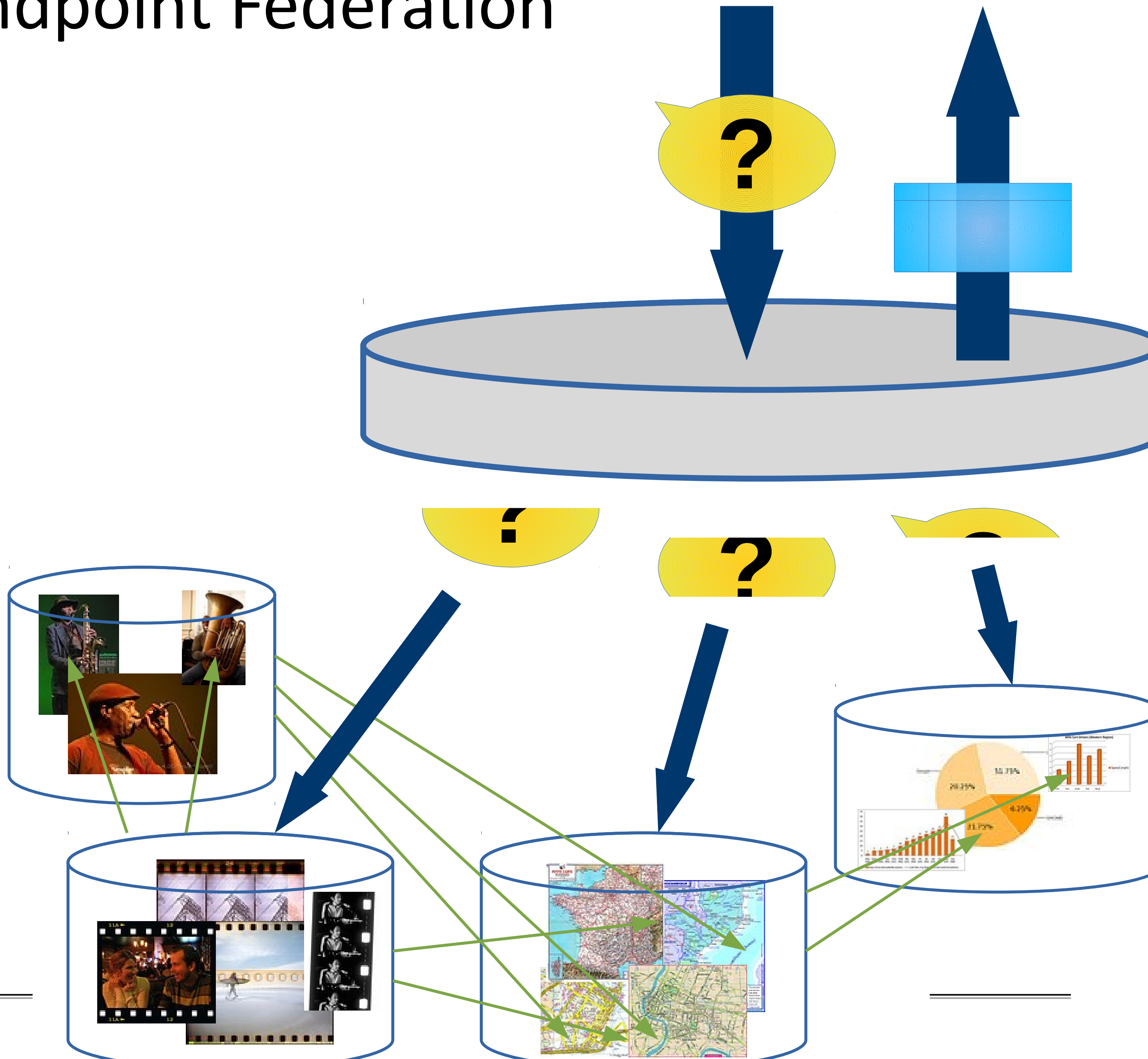
Idea

- Querying a query federation service (mediator)
- Mediator distributes sub-queries to relevant sources
- Finally, mediator combines sub-results



SPARQL Endpoint Federation

- Prototypes:
  - FedX
  - SPLENDID
  - ANAPSID
  - CostFed
  - etc.



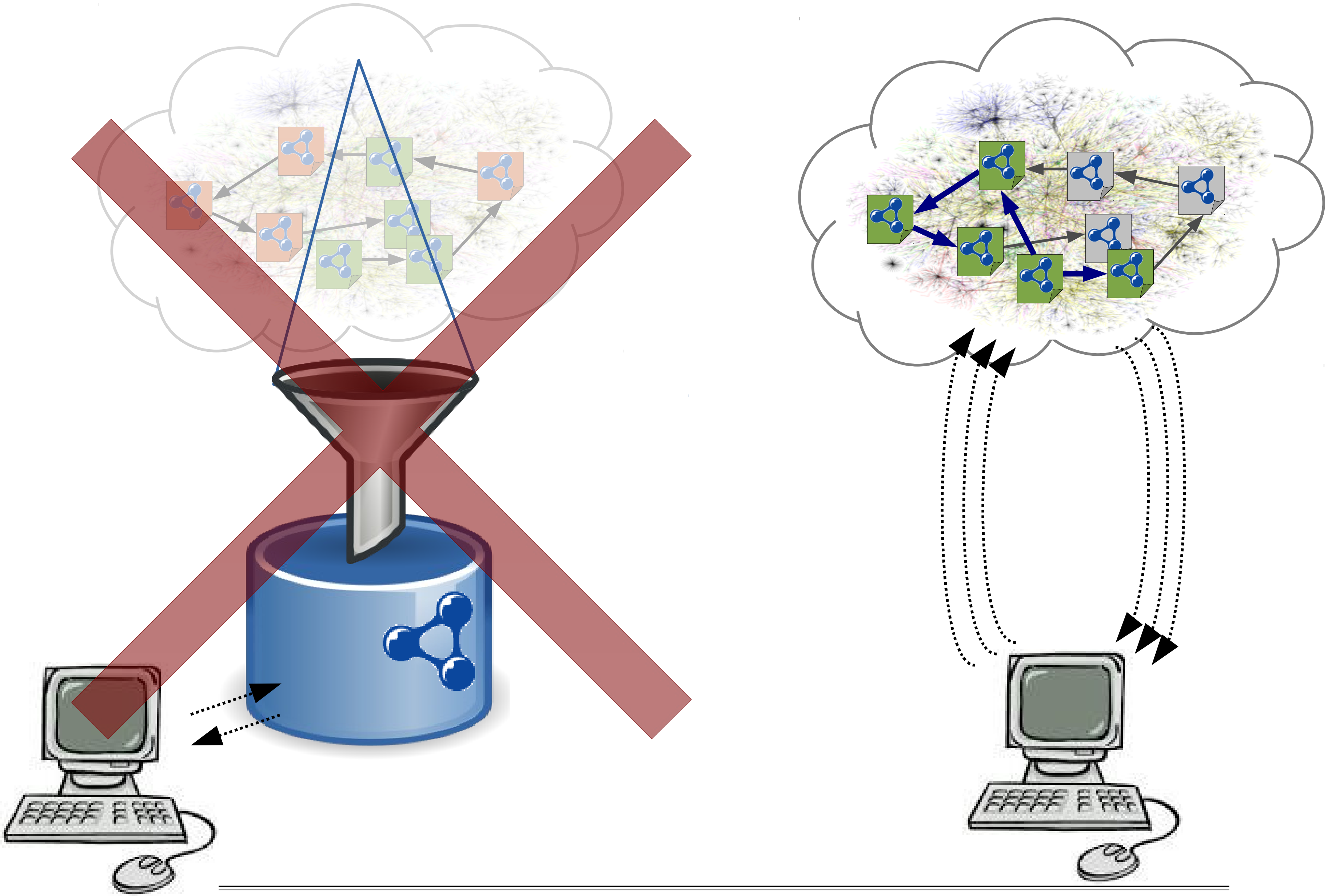
SPARQL 1.1 Federation Extension

- SERVICE pattern in SPARQL 1.1
  - Explicitly specify query patterns whose execution must be distributed to a remote SPARQL endpoint

```
SELECT ?v ?ve WHERE
{
  ?v rdf:type umbel-sc:Volcano ;
     p:location dbpedia:Italy .
  SERVICE <http://volcanos.example.org/query> {
                              ?v p:lastEruption ?ve }
}
```

Linked Data Query Processing

Linked Data Query Processing

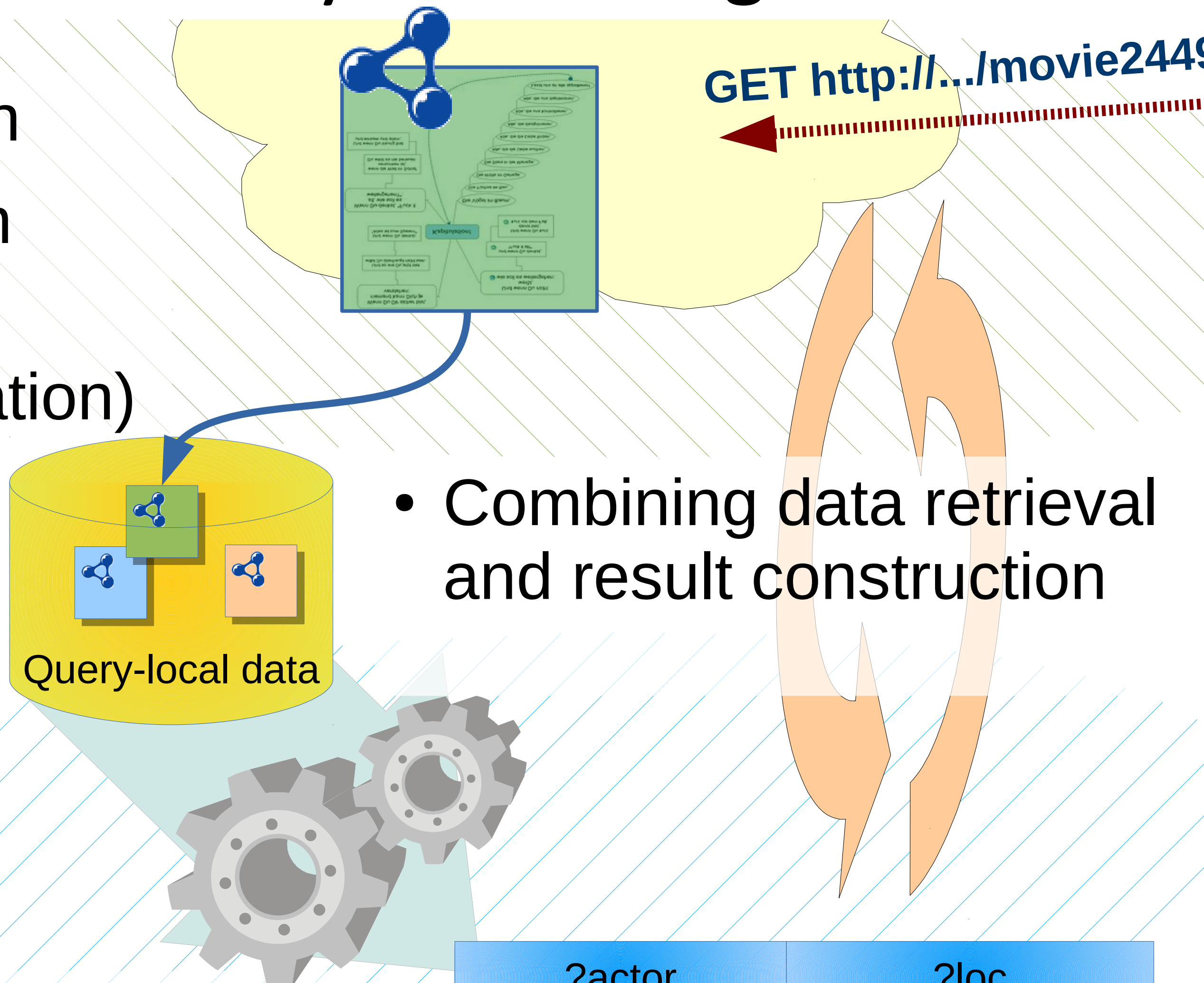


Query Languages

- SPARQL 1.0 [Hartig 2012]
  - Query semantics adapted to the Linked Data setting
- SPARQL Property Paths Patterns [Hartig and Pirrò 2015, 2017]
  - Query semantics adapted to the Linked Data setting
- NautiLOD [Fionda, Gutierrez, and Pirrò 2012]
- LDPath (no formal semantics) [Schaffert et al. 2012]
- LDQL [Hartig and Perez 2015, 2016]
  - Strictly more expressive than any of the above
  - Most basic type of expressions: (*N*,*P*) where *N* is a "link path expression" to specify the query-relevant region of the Web and *P* is a query to be evaluated over the data in the region

"Ingredients" of LD Query Processing

- Data retrieval approach
  - ➢ Data source selection
  - ➢ Data source ranking (optional, for optimization)



- Combining data retrieval and result construction

- Result construction approach

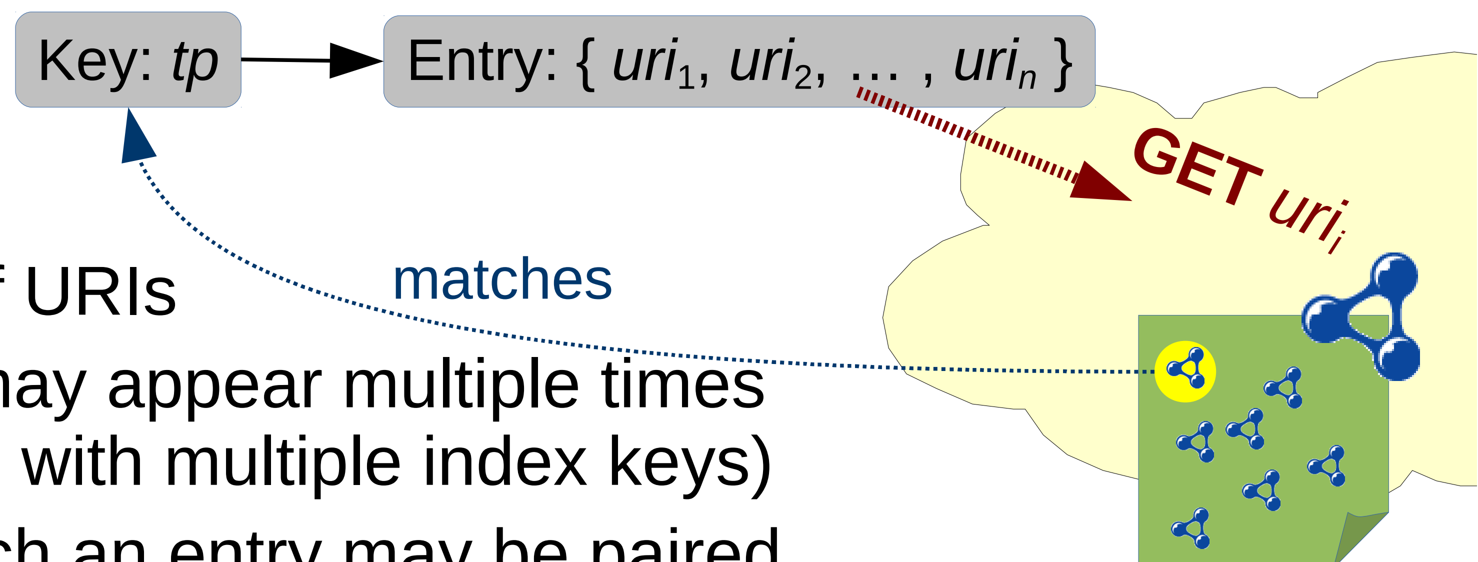
?actor	?loc
http://mdb.../Paul	http://geo.../Berlin
http://mdb.../Ric	http://geo.../Rome

Objective of Source Selection

- Source selection: Given a Linked Data query,
  determine a set of URIs to look up
- Ideal source selection approach:
  - For any query, selects all relevant URIs
  - For any query, selects relevant URIs only
- Irrelevant URIs are not required to answer the query
  - Avoiding their lookup reduces cost of query executions significantly!
- Caveat:
  - What URIs are relevant (resp. irrelevant) is unknown before the query execution has been completed.

Index-Based Source Selection

- **Idea:** Use pre-populated index to determine relevant URIs (and to avoid as many irrelevant ones as possible) [Harth et al. 2010]
- **Index keys:**
  - Different approaches possible [Umbrich et al. 2011]
  - e.g., triple patterns [Ladwig and Tran 2010]



- **Index entries:**
  - Usually, a set of URIs
  - Indexed URIs may appear multiple times (i.e., associated with multiple index keys)
  - Each URI in such an entry may be paired with a cardinality (utilized for source ranking)

Index Construction and Maintenance

- Construction:
  - Given a set of URIs, each of these URIs needs to be looked up and its data needs to be retrieved
  - Alternative: crawl the Web to obtain URIs and their data
  - Alternative: populate index as by-product of query execution
- Maintenance:
  - Web of Linked Data expands and changes over time
  - Add new URIs to the index
  - Keep index in sync with original data
- None of this has been studied yet!

Source Selection by Live Exploration

- **Idea:** Discover relevant URIs recursively by traversing (specific) data links at query execution runtime [Hartig et al. 2009]
  - Natural support of reachability-based query semantics [Hartig and Freytag 2012]

- **Retrieved data serves two purposes:**

(1) Discover further URIs

(2) Construct query result

Live Exploration versus Index-Based

- **Possibilities for parallelized data retrieval are limited**
  - Data retrieval adds to query execution time significantly
- **Usable immediately**
  - Most suitable for “on-demand” querying scenario
- **Depends on the structure of the network of data links**

- **Data retrieval can be fully parallelized**
  - Reduces the impact of data retrieval on query exec. time
- **Usable only after initialization phase**
- **Depends on what has been selected for the index**
- **May miss new data sources**

None of both strategies is superior over the other w.r.t. result completeness (under full-Web query semantics).

- Both strategies may miss (different) solutions for a query

Hybrid Source Selection

Why not get the best of both strategies by combining them?

- Interesting direction of future work
- Ideas:
  - Use index to obtain seed URIs for live exploration (a first approach: "mixed strategy" [Ladwig and Tran 2010])
  - Feed back information obtained by live exploration to update, to expand, or to reorganize the index
  - Use data summary for controlling a live exploration process (e.g., by prioritizing the URIs scheduled for lookup)

Separated Execution Approaches

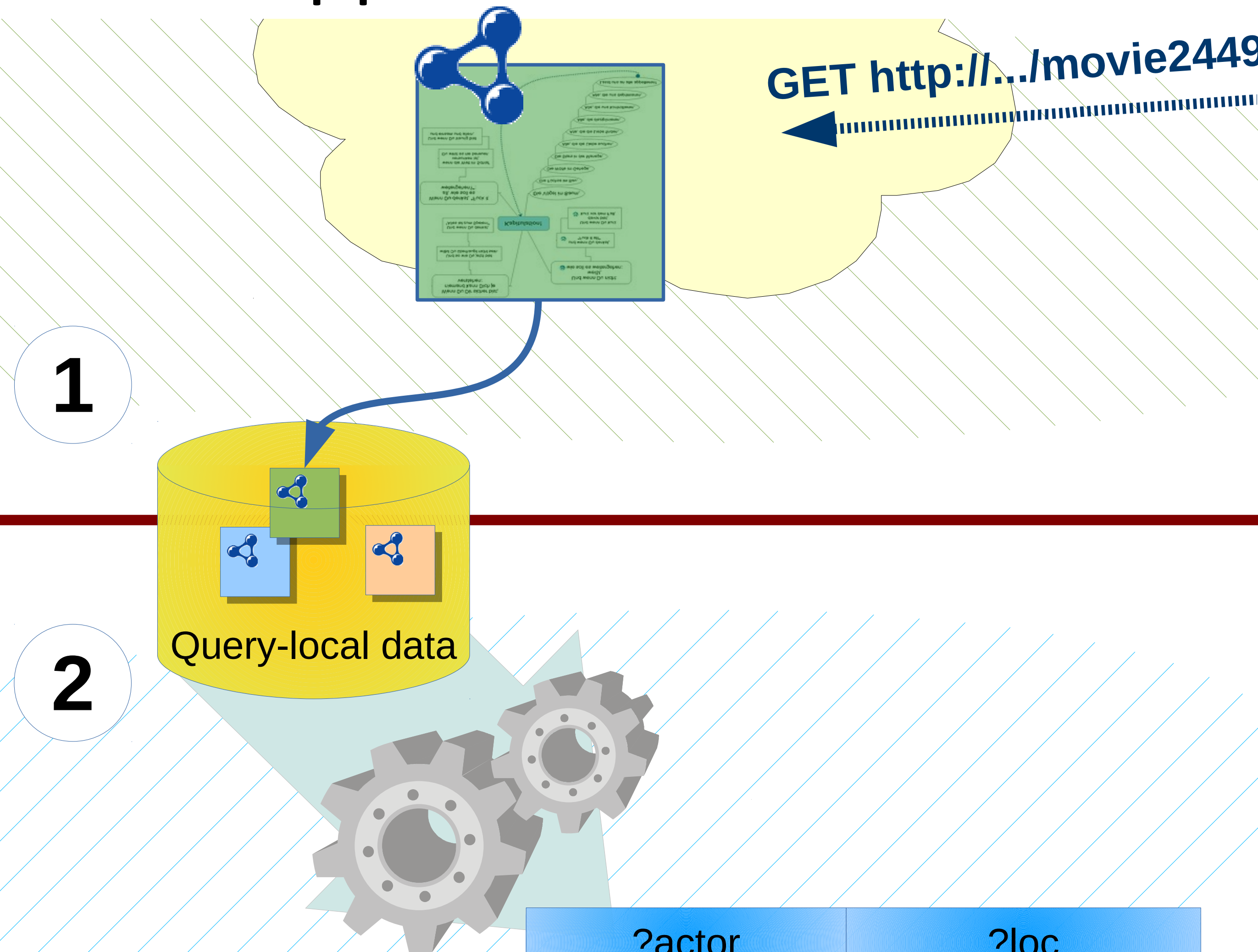
… clearly separate

data retrieval

and

result construction

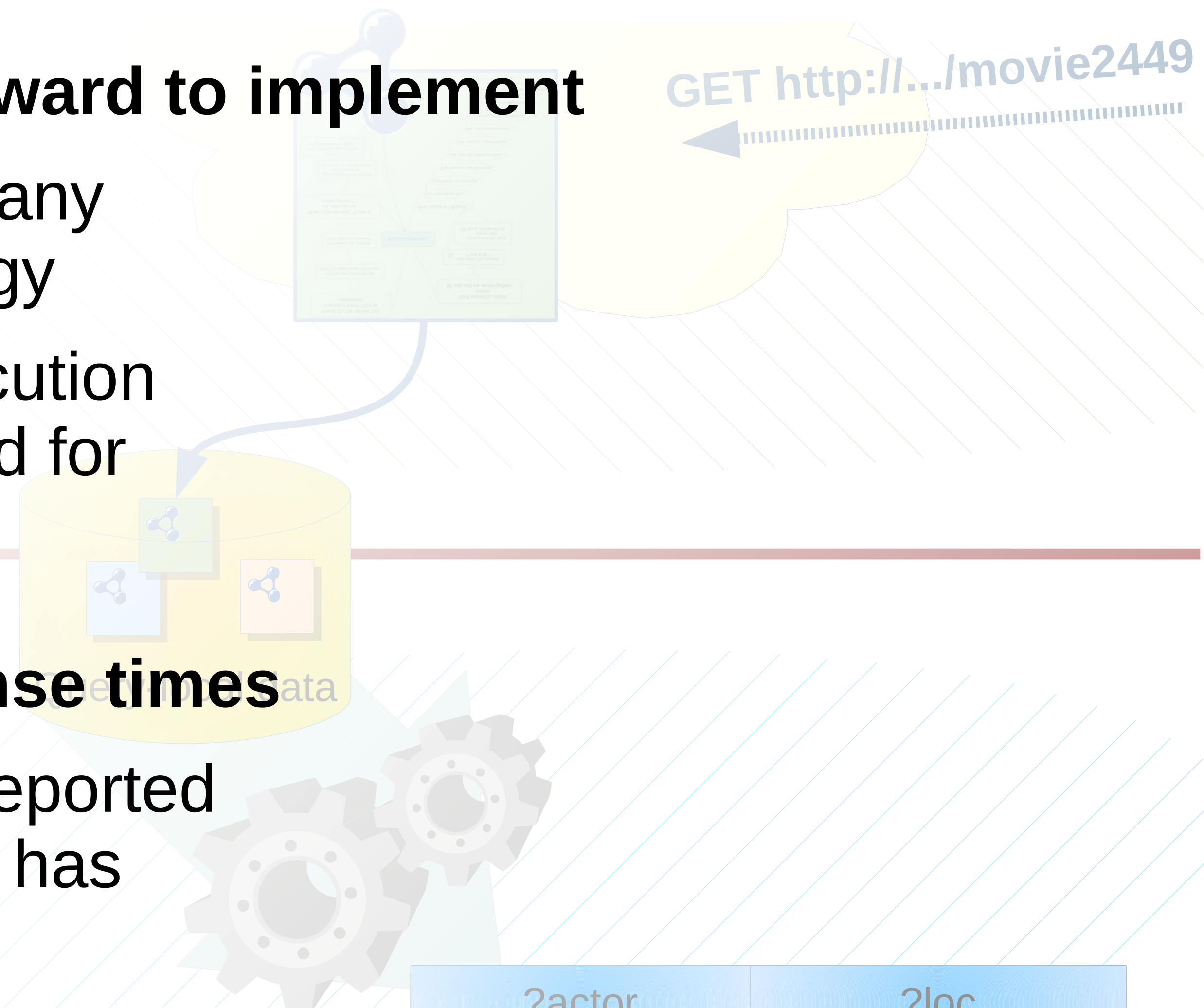
into two consecutive phases



?actor	?loc
http://mdb.../Paul	http://geo.../Berlin
http://mdb.../Ric	http://geo.../Rome

Properties of Separated Execution

- **Advantage: straightforward to implement**
  - Can be combined with any source selection strategy
  - A traditional query execution plan might then be used for constructing the result
- **Downside: long response times**
  - First solutions can be reported only after data retrieval has been completed

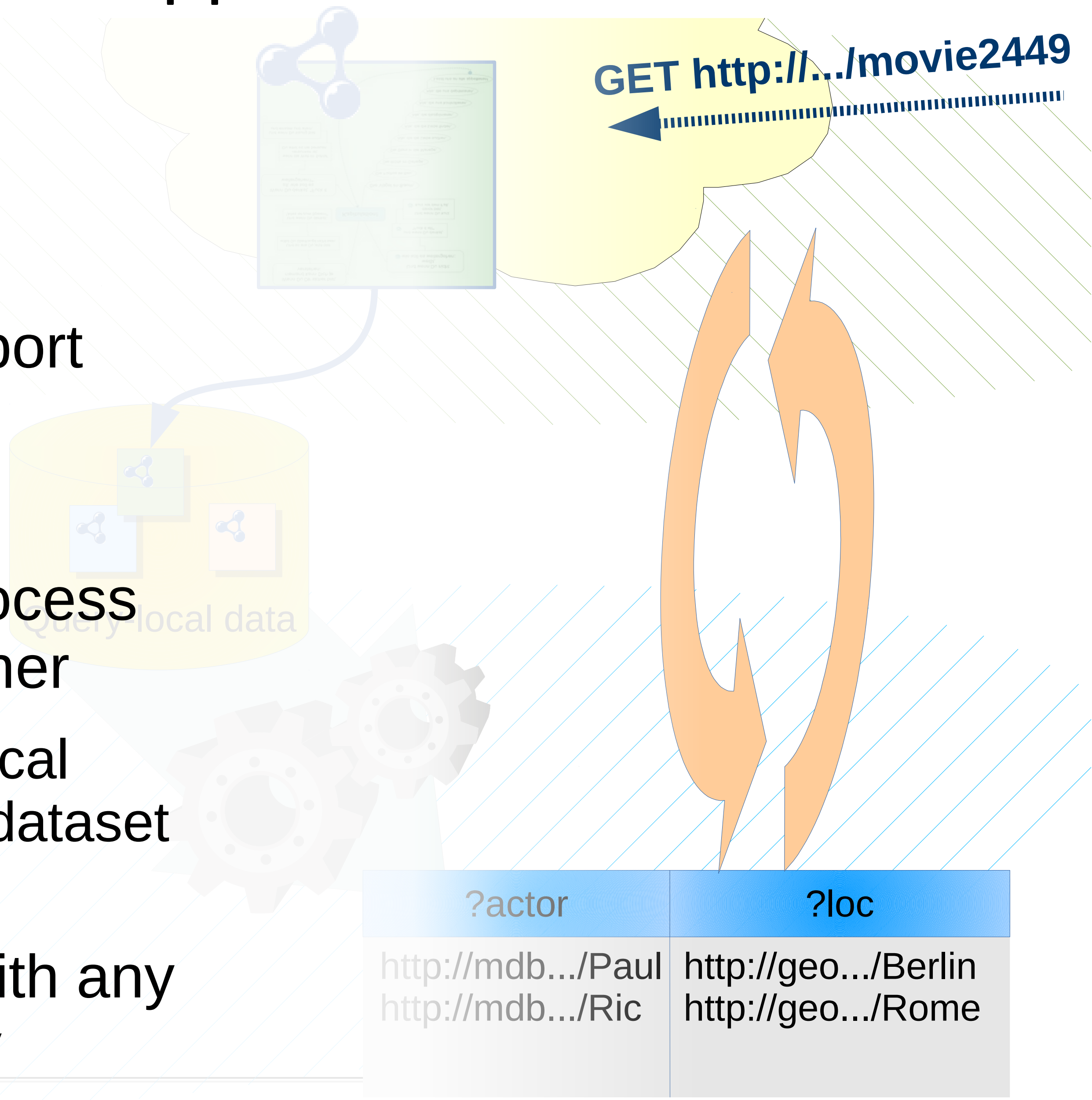


?actor	?loc
http://mdb.../Paul	http://geo.../Berlin
http://mdb.../Ric	http://geo.../Rome

Integrated Execution Approaches

… intertwine
data retrieval and
result construction

- Implementations may report first solutions early
  - For monotonic queries
- Implementations may process data in a streaming manner
  - May require less query-local memory (b/c query-local dataset need not be materialized)
- Can also be combined with any source selection strategy



Traversal-Based Query Execution

… is the combination of integrated execution and live exploration

- **Implementation techniques:**
  - Pipelined iterators [Hartig et al. 2009], [Hartig 2011]
  - Symmetric hash join [Ladwig and Tran 2011]
  - Rete match algorithm [Miranker et al. 2012]
  - Eddies-based network of operators [Hartig and Özsu 2016]

Open Challenges

Open Challenges

- In the context of Linked Data query processing:
  - Execution techniques that go beyond BGPs
  - Comprehensive experimental comparison of approaches (plus: benchmark)
  - Query optimization
- Heterogeneity in terms of data access interfaces

[Cheng and Hartig 2020]

- Heterogeneity in the data
  - Different vocabularies/ontologies
  - Different URIs for the same thing
  - Different data models

www.liu.se

LINKÖPING UNIVERSITY