Querying Decentralized Knowledge Graphs

Hala Skaf-Molli
Hala.Skaf@univ-nantes.fr
http://pagesperso.ls2n.fr/~skaf-h

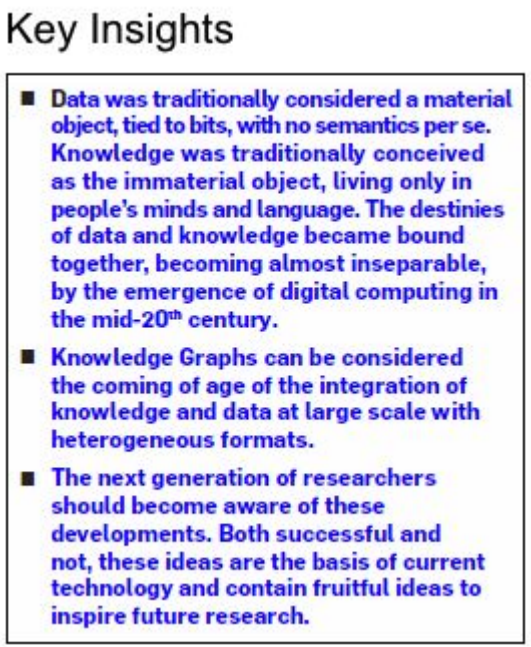
University of Nantes, LS2N, France
(Distributed Data Management) GDD Team

UNIVERSITÉ DE NANTES

Keynote 06-07

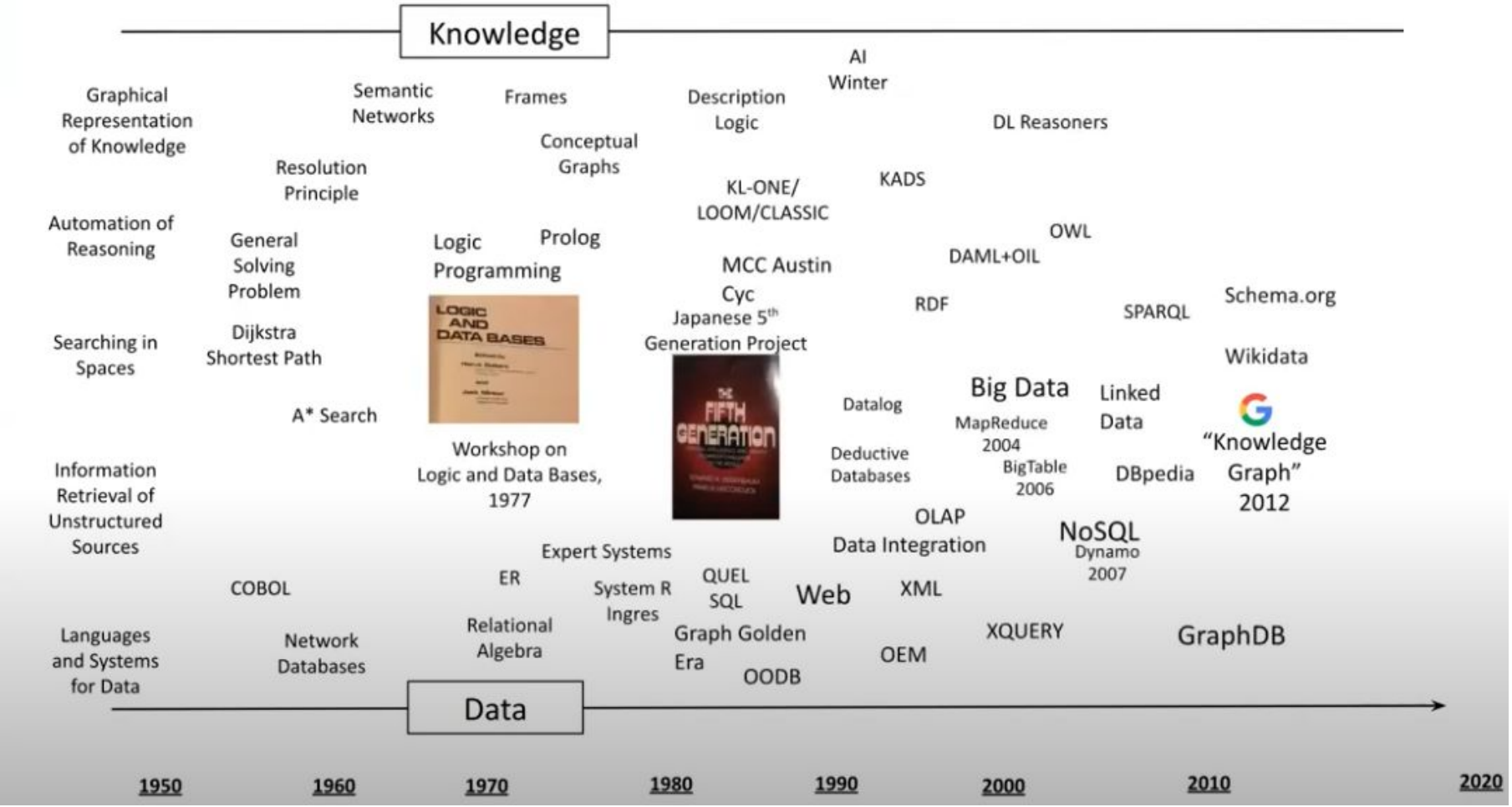
What is Data?
What is knowledge?

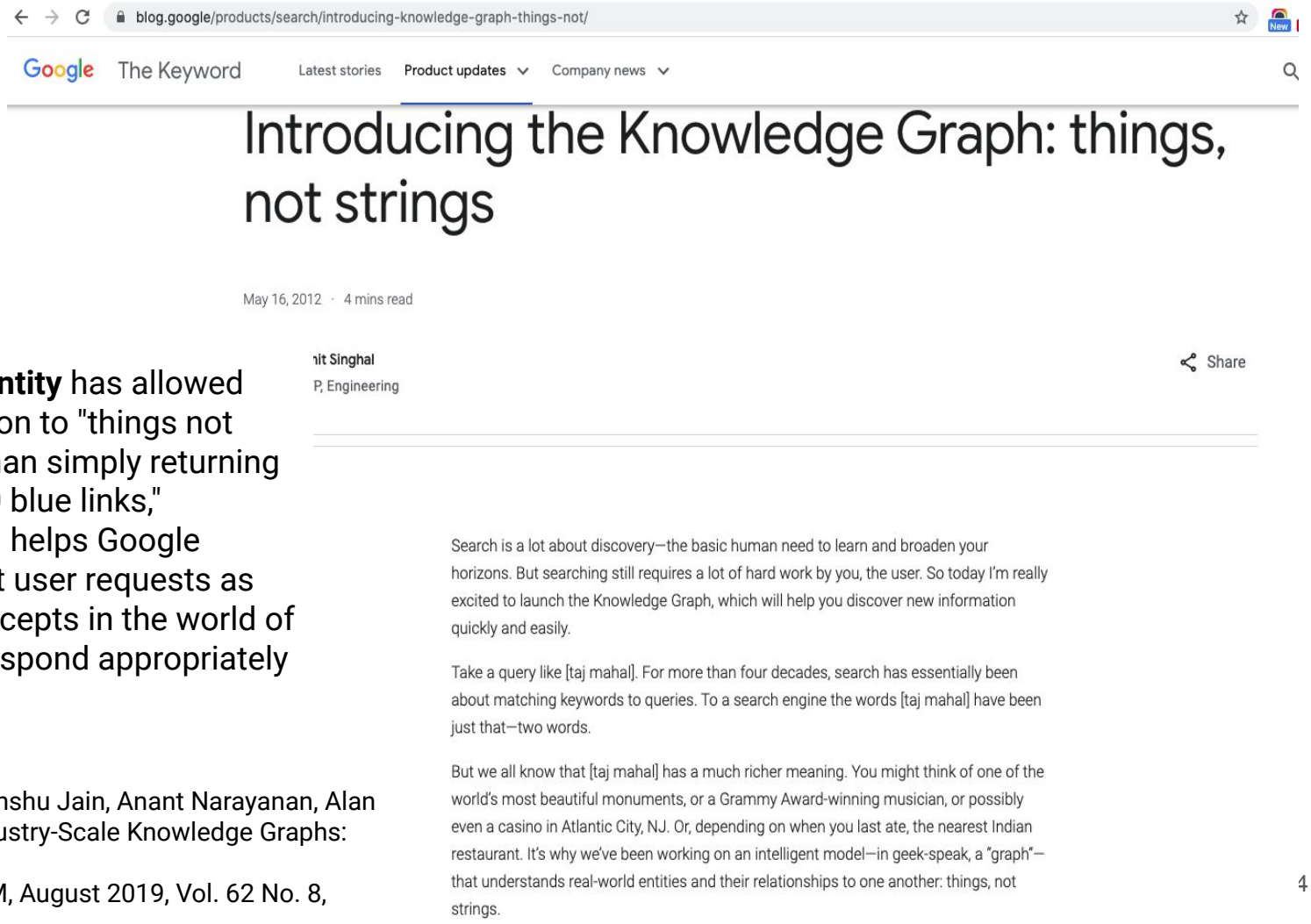
- **Data**: is any sequence of one or more symbols, potentially meaningful (needs knowledge)
  - 8 February 1828, Nantes, ...
- **Knowledge**: Potential meaning (needs Data)
- Data+Knowledge = Meaning
  - Assertions:
    - Nantes is the birthplace of Jules Verne.
    - Jules Verne was born on February 8, 1828.



Claudio Gutierrez, Juan F. Sequeda. Knowledge Graphs. Communications of the ACM, March 2021, Vol. 64 No. 3,

Source: [EDBT 2021 Tutorial on the History of Knowledge Graph's]





This focus on **identity** has allowed Google to transition to "things not strings." Rather than simply returning the traditional "10 blue links," Knowledge Graph helps Google products interpret user requests as references to concepts in the world of the user and to respond appropriately

Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, Jamie Taylor. Industry-Scale Knowledge Graphs: Lessons and Challenges
Communications of the ACM, August 2019, Vol. 62 No. 8,

jules verne

Tous | Images | Actualités | Vidéos | Maps | Plus | Paramètres | Outils

Environ 38 100 000 résultats (0,75 secondes)

https://fr.wikipedia.org › wiki › Jules_Verne

Jules Verne — Wikipédia

Jules Verne, né le 8 février 1828 à Nantes et mort le 24 mars 1905 à Amiens, est un écrivain français dont l'œuvre est, pour la plus grande partie, constituée de ...

Période d'activité ... **Nationalité :** Français
Nom de naissanc... **Formation :** Lycée Georges-Clemenceau (1844...

Maison de Jules Verne · Portail:Jules Verne · Revue Jules Verne · Michel Verne

https://julesverne.nantesmetropole.fr › approfondir › jul...

Jules Verne et Nantes - Musée Jules Verne

Tous ces éléments ont durablement marqué **Jules Verne**. Nantes a été la source du rêve et le creuset de l'inspiration des Voyages extraordinaires. Extrait des ...

https://julesverne.nantesmetropole.fr › approfondir › la-...

La vie et l'œuvre de Jules Verne - Musée Jules Verne

Jules Verne s'est toujours considéré comme un auteur dramatique. A 17 ans, il écrivait des drames romantiques imités de Victor Hugo, mais c'est plutôt avec le ...

https://www.linternaute.fr › ... › XIXème siècle

Jules Verne : biographie de l'auteur des Voyages ...

16 avr. 2020 — BIOGRAPHIE **JULES VERNE** - **Jules Verne** est considéré comme le père français de la Science-Fiction. C'est l'auteur des Voyages ...

Sa biographie · Voyages extraordinaires · Voyage au centre de la Terre · Sa mort

Autres questions posées

Quel est le mouvement littéraire de Jules Verne ?

Quel est le premier succès de Jules Verne ?

Quels sont les œuvre les plus connus de Jules Verne ?

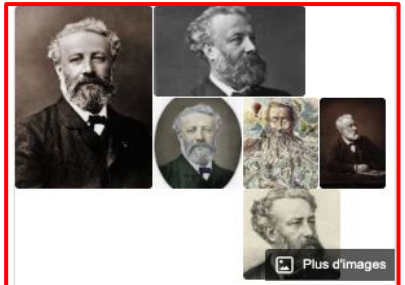
Quels progrès scientifiques évoqué Jules Verne dans ses romans ?

Commentaires

https://www.babelio.com › auteur › Jules-Verne

Jules Verne - Babelio

Knowledge Panel



Plus d'images

Jules Verne

Écrivain

Jules Verne, né le 8 février 1828 à Nantes et mort le 24 mars 1905 à Amiens, est un écrivain français dont l'œuvre est, pour la plus grande partie, constituée de romans d'aventures évoquant les progrès scientifiques du XIXe siècle. Wikipédia

Date/Lieu de naissance : 8 février 1828, Nantes

Date de décès : 24 mars 1905, Amiens

Films : Voyage au centre de la Terre, PLUS

Pièces de théâtre : Un neveu d'Amérique, Michel Strogoff

Enfants : Michel Verne, Valentine Morel, Suzanne Morel

Livres — Voir d'autres éléments (plus de 45)

Vingt Mille Lieues sous les... 1870
Voyage au centre de la Ter... 1864
Le Tour du monde en quatr... 1873
L'Île mystérie... 1875

roducts/search/introducing-knowledge-graph-things-not/

rd Latest stories Product updates Company news

Introducing the Knowled not strings

May 16, 2012 · 4 mins read

A Amit Singhal
SVP, Engineering

Search is a lot about discovery—the basic human need horizons. But searching still requires a lot of hard work excited to launch the Knowledge Graph, which will help quickly and easily.

Take a query like [taj mahal]. For more than four decad about matching keywords to queries. To a search engi just that—two words.

But we all know that [taj mahal] has a much richer mea world's most beautiful monuments, or a Grammy Awa even a casino in Atlantic City, NJ. Or, depending on wh restaurant. It's why we've been working on an intelliger that understands real-world entities and their relations strings.

The Knowledge Graph enables you to search for things

Entreprise Knowledge Graphs

	Data model	Size of the graph	Development stage
Microsoft	The types of entities, relations, and attributes in the graph are defined in an ontology.	~2 billion primary entities, ~55 billion facts	Actively used in products
Google	Strongly typed entities, relations with domain and range inference	1 billion entities, 70 billion assertions	Actively used in products
Facebook	All of the attributes and relations are structured and strongly typed, and optionally indexed to enable efficient retrieval, search, and traversal.	~50 million primary entities, ~500 million assertions	Actively used in products
eBay	Entities and relation, well-structured and strongly typed	Expect around 100 million products, >1 billion triples	Early stages of development and deployment
IBM	Entities and relations with evidence information associated with them.	Various sizes. Proven on scales documents >100 million, relationships >5 billion, entities >100 million	Actively used in products and by clients

Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, Jamie Taylor. Industry-Scale Knowledge Graphs: Lessons and Challenges Communications of the ACM, August 2019, Vol. 62 No. 8,

Open Knowledge Graphs

DBpedia is a knowledge graph extracted from structured data in Wikipedia.

Wikidata is a collaboratively edited knowledge graph, operated by the Wikimedia foundation (hosting Wikipedia)

	Instances/Entities	Assertions	Classes	Relations
DBpedia	5,044,223	854,294,312	760	1,355
CYC	122,441	2,229,266	116,821	116,821
Wikidata	52,252,549	732,420,508	2,356,259	6,236
NELL	5,120,688	60,594,443	1,187	440

Nicolas Heist et al. Knowledge Graphs on the Web - An Overview, Knowledge Graphs for Explainable Artificial Intelligence: Foundations, Applications and Challenges. Vol. 47. 2020

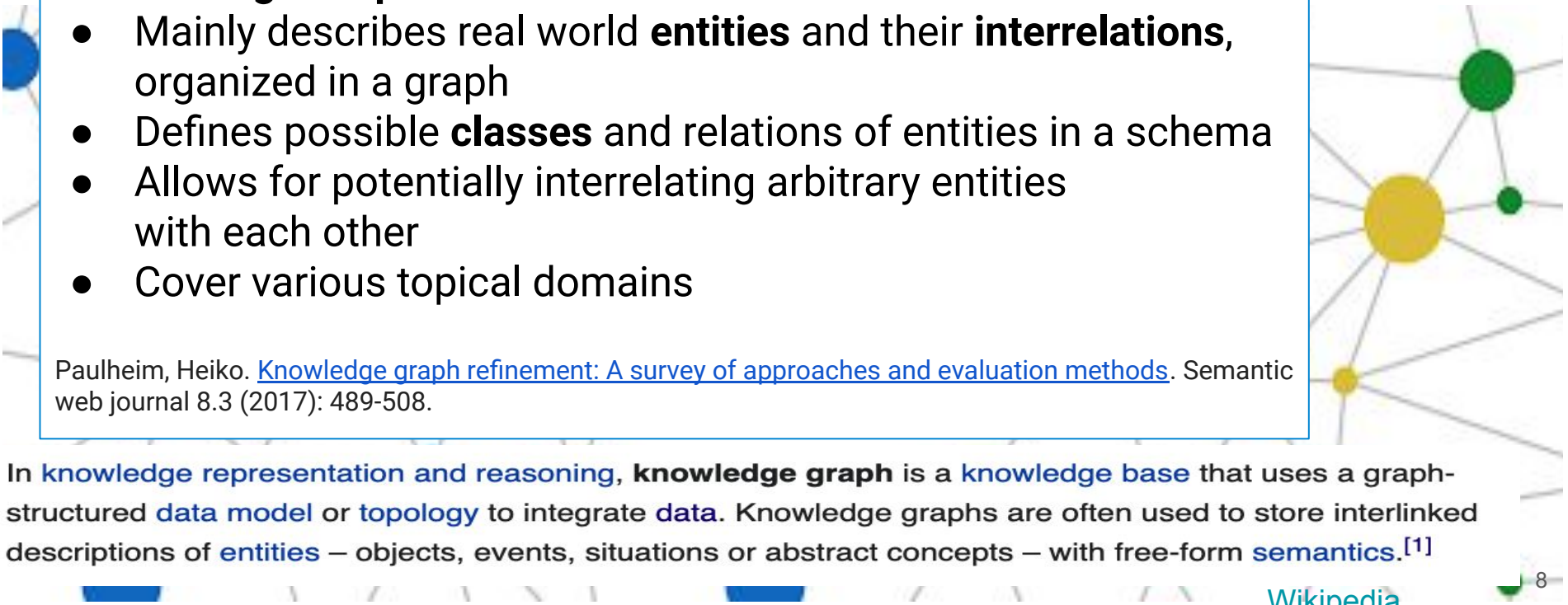
Lehmann, Jens, et al. "**Dbpedia–a large-scale**, multilingual knowledge base extracted from wikipedia." Semantic web 6.2 (2015): 167-195.
Malyshev, Stanislav, et al. "Getting the most out of **wikidata**: Semantic technology usage in wikipedia's knowledge graph." International Semantic Web Conference. 2018.

Some definitions of a Knowledge Graph

A **Knowledge Graph:**

- Mainly describes real world **entities** and their **interrelations**, organized in a graph
- Defines possible **classes** and relations of entities in a schema
- Allows for potentially interrelating arbitrary entities with each other
- Cover various topical domains

Paulheim, Heiko. Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic web journal 8.3 (2017): 489-508.



In knowledge representation and reasoning, **knowledge graph** is a knowledge base that uses a graph-structured data model or topology to integrate data. Knowledge graphs are often used to store interlinked descriptions of entities – objects, events, situations or abstract concepts – with free-form semantics.[1]

Wikipedia

A KG describes real-world entities and their relation, organized in a graph

Schema.org

Welcome to Schema.org

Schema.org is a collaborative, community activity with a mission to create, maintain, and promote schemas for structured data on the Internet, on web pages, in email messages, and beyond.

Schema.org vocabulary can be used with many different encodings, including RDFa, Microdata and JSON-LD. These vocabularies cover entities, relationships between entities and actions, and can easily be extended through a well-documented extension model. Over 10 million sites use Schema.org to markup their web pages and email messages. Many applications from Google, Microsoft, Pinterest, Yandex and others already use these vocabularies to power rich, extensible experiences.

Founded by Google, Microsoft, Yahoo and Yandex, Schema.org vocabularies are developed by an open community process, using the public-schemaorg@w3.org mailing list and through GitHub.

A shared vocabulary makes it easier for webmasters and developers to decide on a schema and get the maximum benefit for their efforts. It is in this spirit that the founders, together with the larger community have come together - to provide a shared collection of schemas.

We invite you to get started!

View our blog at blog.schema.org or see release history for version 12.0.

schema.org

Schema.org

Welcome to Schema.org

Schema.org is a collaborative, community activit
promote schemas for structured data on the Inte
beyond.

Schema.org vocabulary can be used with many
and JSON-LD. These vocabularies cover entities,
can easily be extended through a well-document
Schema.org to markup their web pages and ema
Microsoft, Pinterest, Yandex and others already
experiences.

Founded by Google, Microsoft, Yahoo and Yande
an open community process, using the public-sc
GitHub.

A shared vocabulary makes it easier for webmas
get the maximum benefit for their efforts. It is i
larger community have come together - to provi

We invite you to get started!

View our blog at blog.schema.org or see release

Schema.org Documentation Schemas About

Organization of Schemas

The schemas are a set of 'types', each associated with a set of properties. The types are arranged in a hierarchy. The vocabulary currently consists of 779 Types, 1390 Properties 15 Datatypes, 81 Enumerations and 437 Enumeration members.

Browse the full hierarchy in HTML:

- One page per type
- Full list of types, shown on one page

Or you can jump directly to a commonly used type:

- Creative works: CreativeWork, Book, Movie, MusicRecording, Recipe, TVSeries ...
- Embedded non-text objects: AudioObject, ImageObject, VideoObject
- Event
- Health and medical types: notes on the health and medical types under MedicalEntity.
- Organization
- Person
- Place, LocalBusiness, Restaurant ...
- Product, Offer, AggregateOffer
- Review, AggregateRating
- Action

See also the releases page for recent updates and project history.

We also have a small set of primitive data types for numbers, text, etc. More details about the data model, etc. are available here.

In this talk

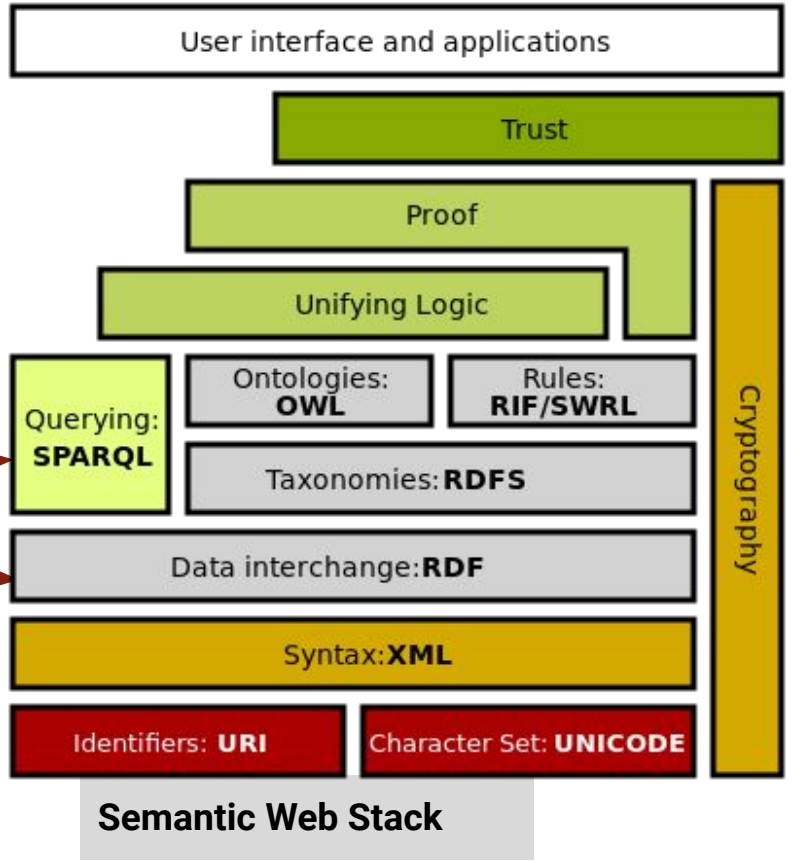
Our research

- Accessibility of open knowledge graphs
  - Anyone can ask any query and get complete answer in a “reasonable time”
- Web preemption allows accessibility

Open Knowledge Graphs and Semantic Web

Knowledge representation model

Query language to retrieve information from KGs.



Semantic Web Stack

Resource Description Framework for Knowledge Representation

w3.org/TR/rdf11-concepts/

1.1 Graph-based Data Model

The core structure of the abstract syntax is a set of triples, each consisting of a subject, a predicate and an object. A set of such triples is called an RDF graph. An RDF graph can be visualized as a node and directed-arc diagram, in which each triple is represented as a node-arc-node link.

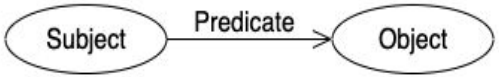
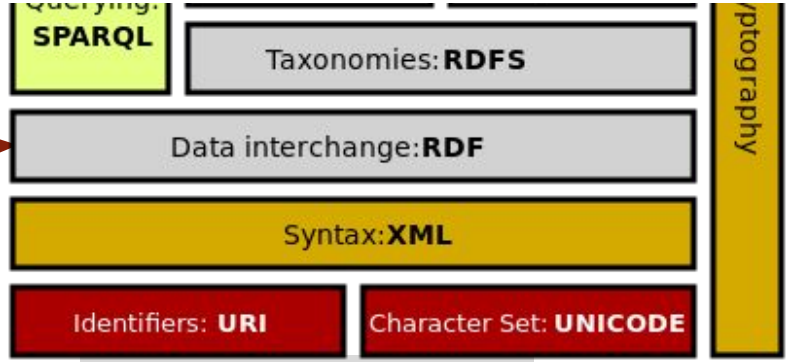


Fig. 1 An RDF graph with two nodes (Subject and Object) and a triple connecting them (Predicate)

There can be three kinds of nodes in an RDF graph: IRIs, literals, and blank nodes.

Facts are stored in triple format (subject predicate object).



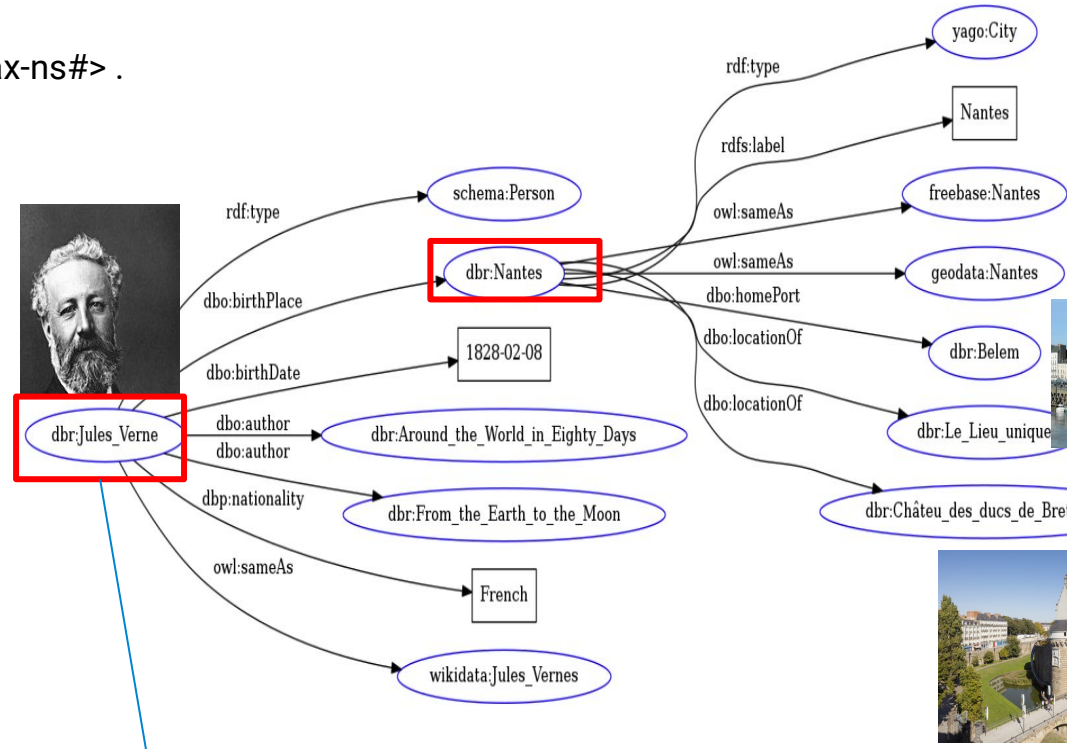
Semantic Web Stack

RDF is flexible, schema-free to represent knowledge as triples (subject predicate object)

@prefix **dbr**:<http://dbpedia.org/resource/> .
@prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix schema: <http://www.schema.org/>.
@prefix dbo:<http://dbpedia.org/ontology/>.
@prefix owl: <http://www.w3.org/2002/07/owl#> .

dbr:Jules_Verne rdf:type schema:Person;
dbo:birthPlace **dbr:Nantes;**
dbo:birthDate "1828-02-08";
dbo:author
dbr:Around_the_World_in_Eighty_Days,
dbr:From_the_Earth_to_the_Moon;
dbp:nationality "French";
owl:sameAs wikidata:Jules_Verne.

.....



URI: <http://dbpedia.org/resource/Jules_Verne>

SPARQL queries for querying RDF knowledge Graphs

SPARQL 1.1

Federated

```
SELECT ?s
WHERE {
  ?s a foaf:Person .
  SERVICE
<http://dbpedia.org/sparql>
{?s foaf:knows ?o }
}
```

Basic Graph Pattern

```
SELECT ?actor ?city
 WHERE {
     ?s  rdf:type  dbo:Actor ;
          dbo:birthPlace ?city.
}
```

Aggregate

```
SELECT ?c  (count(?i) as ?card)
 WHERE {
     ?i  rdf:type  ?c.
       GROUP BY ?c
}
```

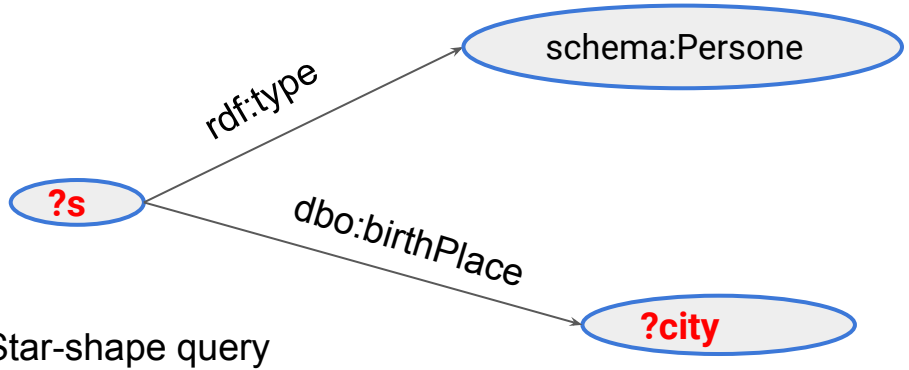
Property Path

```
SELECT * WHERE{
?o dbo:basedOn ?i .
?i  dbo:genre dbr:Fiction .
?o rdf:type rdfs:subClassOf* dbo:Work
}
```

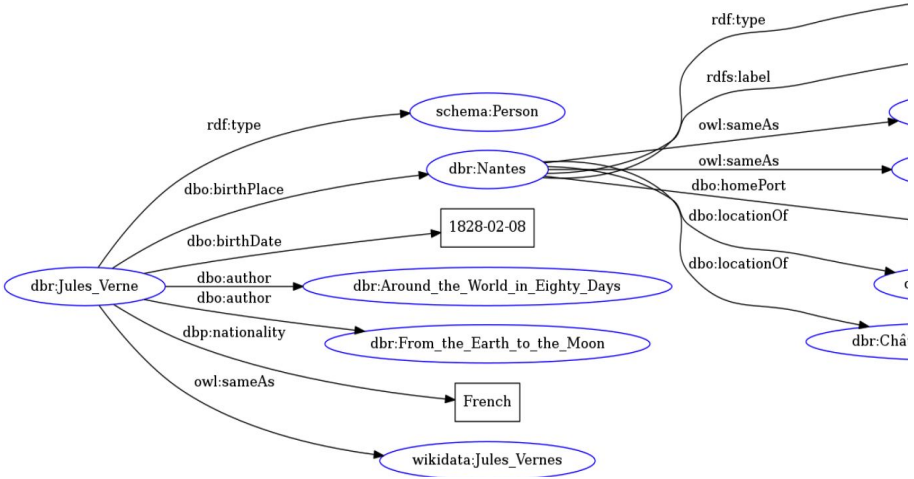
https://www.w3.org/TR/sparql11-query/

Example of a SPARQL query

```
SELECT ?s  ?city
 WHERE {
     ?s  rdf:type  schema:Person ;
         dbo:birthPlace ?city.
}
```



Star-shape query



?s	?city
dbr:Jules_Vernes	dbr:Nantes

What are the birthplaces of all movie actors?

Public SPARQL Endpoint of dbpedia

Virtuoso SPARQL Query Editor

Default Data Set Name (Graph IRI)

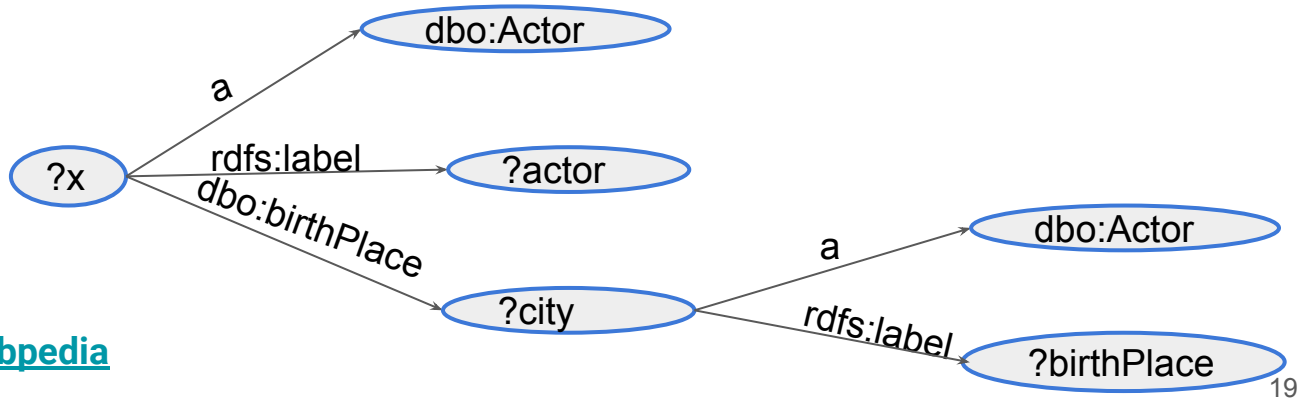
http://dbpedia.org

Query Text

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?actor ?birthPlace
WHERE {
        ?x a dbo:Actor;
        rdfs:label ?actor;
        dbo:birthPlace ?city.
  ?city a dbo:City;
        rdfs:label ?birthPlace.
}
```

actor	birthPlace
"منغ جيا"@ar	"Loudi"@en
"منغ جيا"@ar	"Loudi"@de
"منغ جيا"@ar	"Loudi"@es
"منغ جيا"@ar	"Loudi"@fr
"منغ جيا"@ar	"Loudi"@it
"منغ جيا"@ar	"婁底市"@ja
"منغ جيا"@ar	"Loudi"@nl
"منغ جيا"@ar	"Loudi"@pl
"منغ جيا"@ar	"Loudi"@pt
"منغ جيا"@ar	"Лоуди"@ru
"منغ جيا"@ar	"娄底市"@zh
"Meng Jia"@es	"Loudi"@en



Try it yourself: [Actor and city on dbpedia]

Interlinked Open Knowledge Graphs

Tim Berners-Lee
Date: 2006-07-27, last change: $Date: 2009/06/18 18:24:33 $
Status: personal view only. Editing status: imperfect but published.
Up to Design Issues

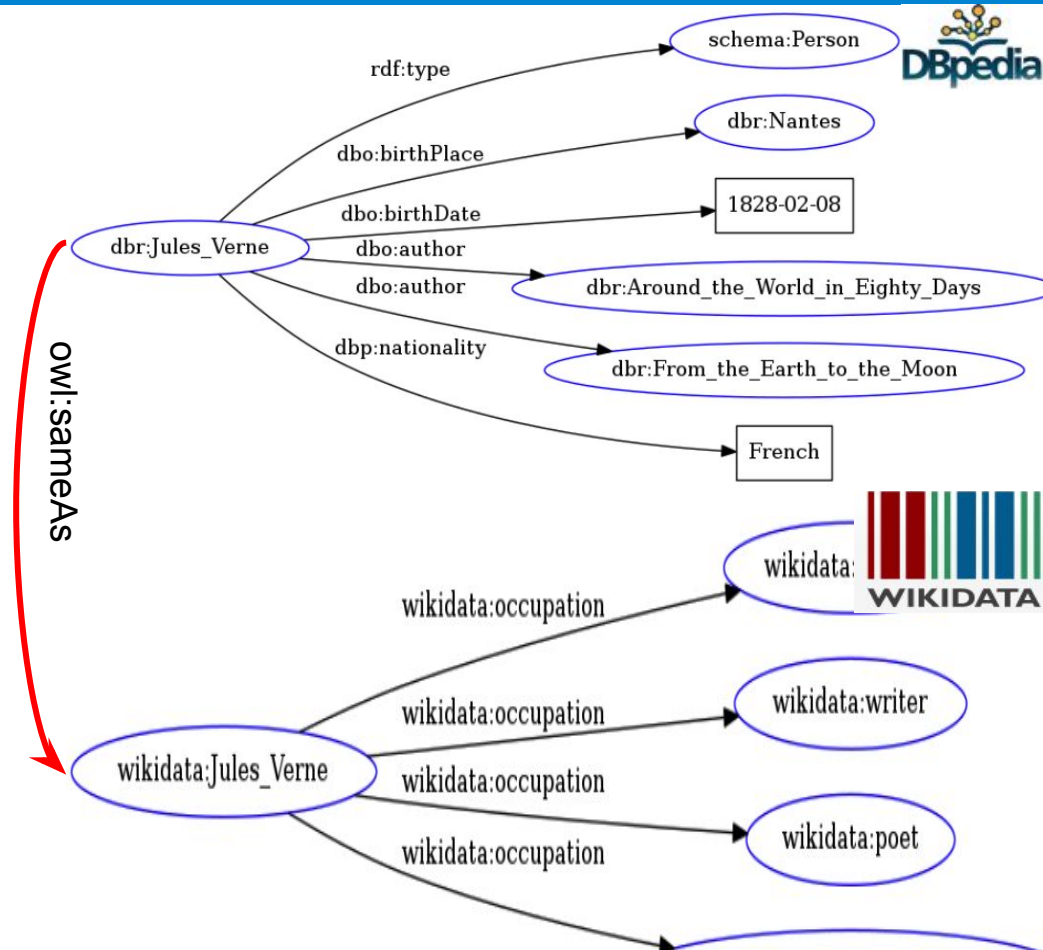
Linked Data

The Semantic Web isn't just about putting data on the web. It is about making links, so that a person or machine can explore the web of data. With linked data, when you have some of it, you can find other, related, data.

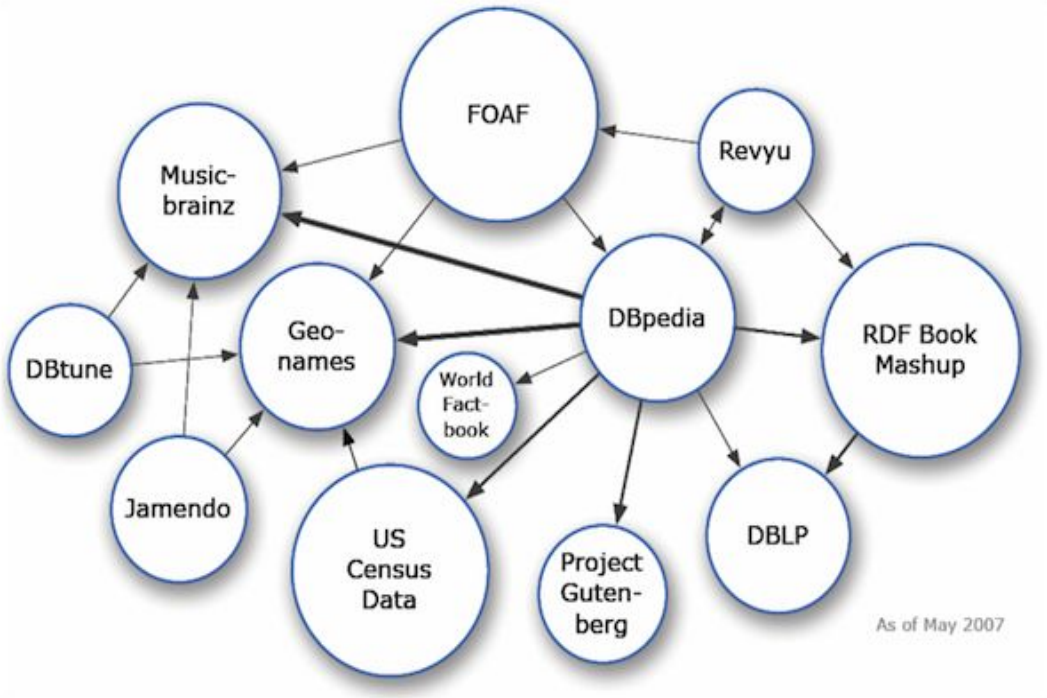
Like the web of hypertext, the web of data is constructed with documents on the web. However, unlike the web of hypertext, where links are relationships anchors in hypertext documents written in HTML, for data they links between arbitrary things described by RDF,. The URIs identify any kind of object or concept. But for HTML or RDF, the same expectations apply to make the web grow:

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)
4. Include links to other URIs. so that they can discover more things.

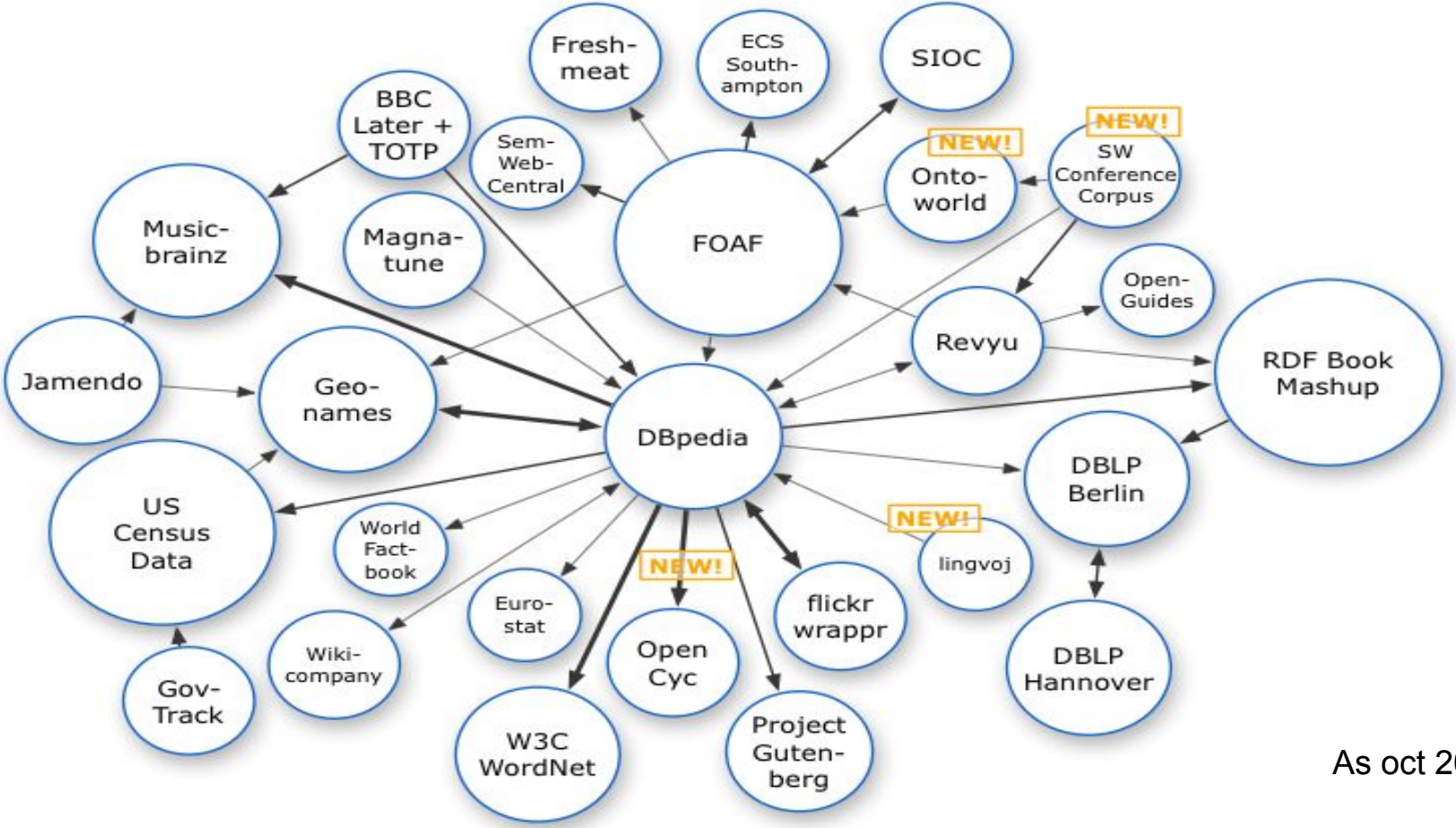
https://www.w3.org/DesignIssues/LinkedData.html



Linked Open Data

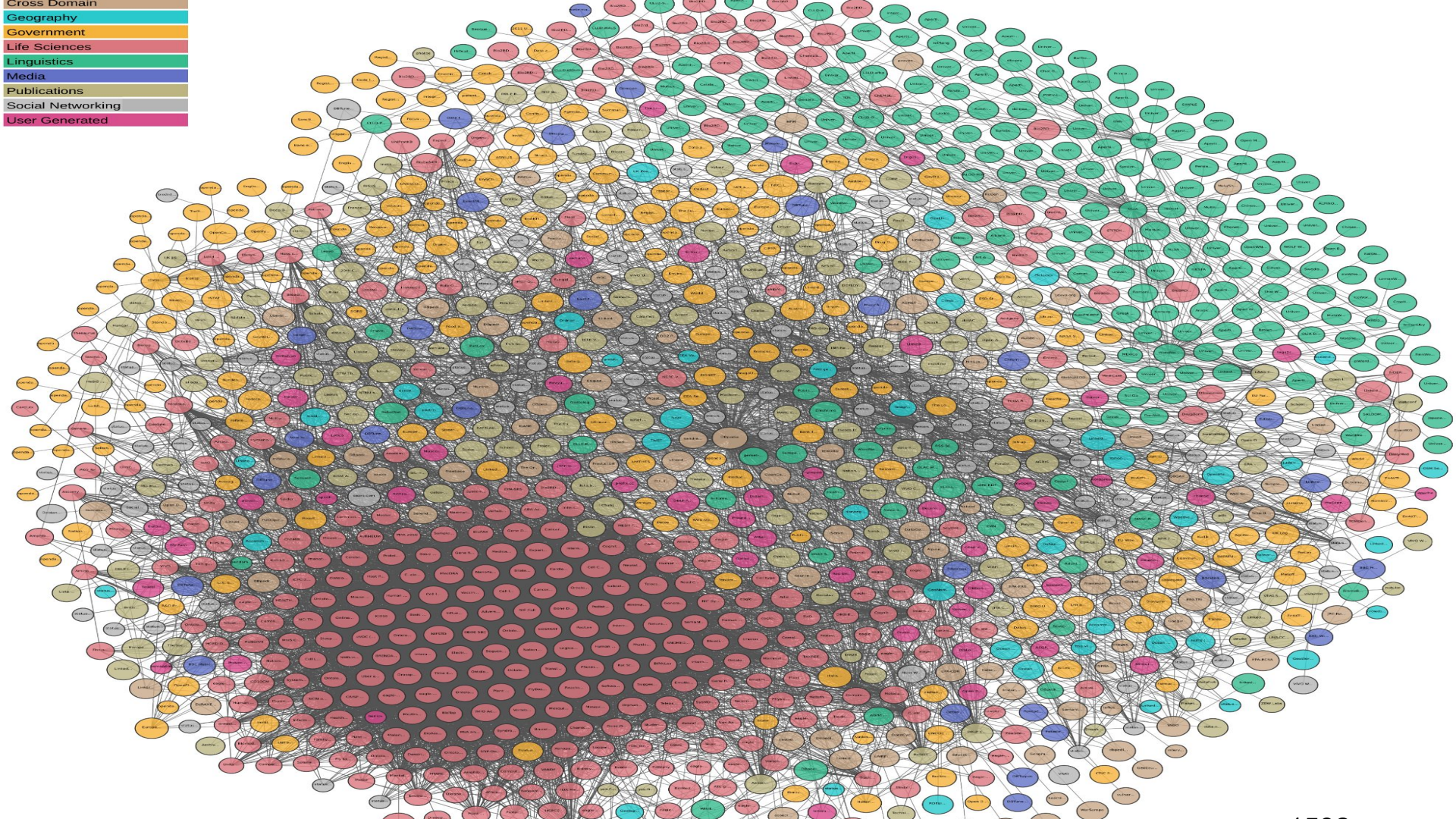


https://lod-cloud.net/



As oct 2007 (25)

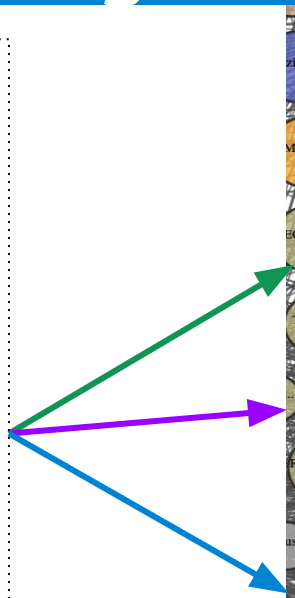
https://lod-cloud.net/

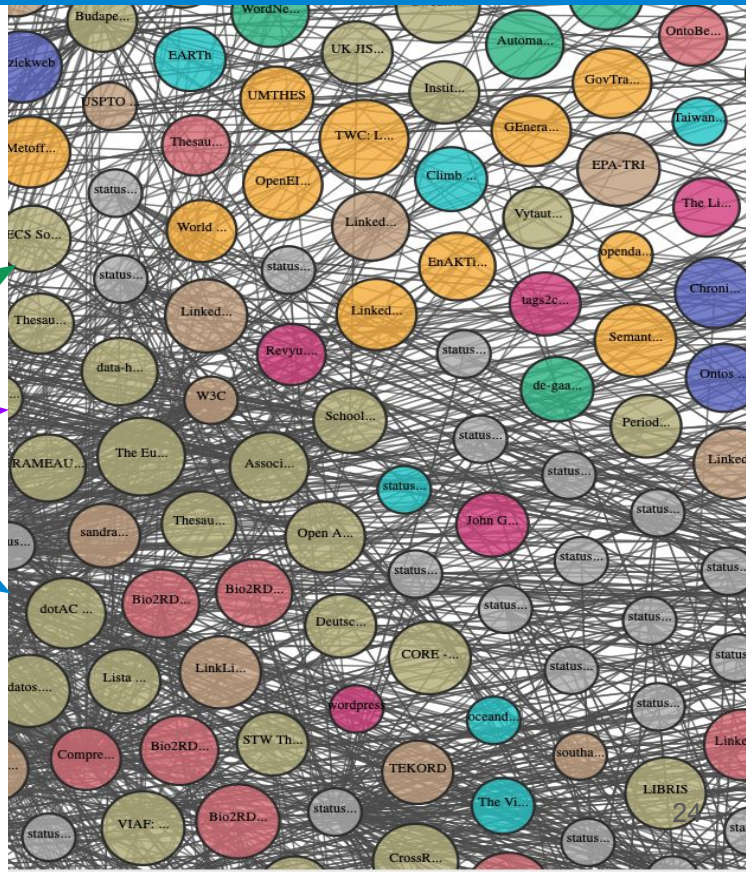

Cross Domain
Geography
Government
Life Sciences
Linguistics
Media
Publications
Social Networking
User Generated

Online querying: Write a federated SPARQL query using SERVICE Clause

```
SELECT ?s
WHERE {
  ?s a foaf:Person .
  SERVICE <http://dbpedia.org/sparql>
{?s foaf:knows ?o }
SERVICE
<http://wikidata.org/sparql>
{?s foaf:knows ?o }
SERVICE
<http://LinkedMDB.org/sparql>
{?s foaf:knows ?o }

}
```



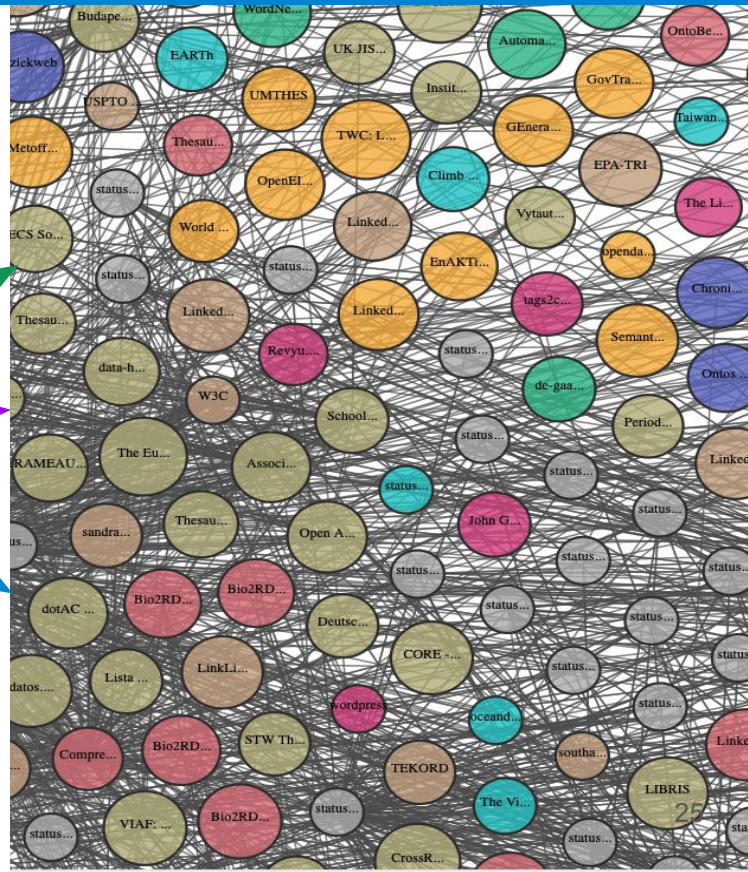


Write a federated SPARQL query using SERVICE Clause

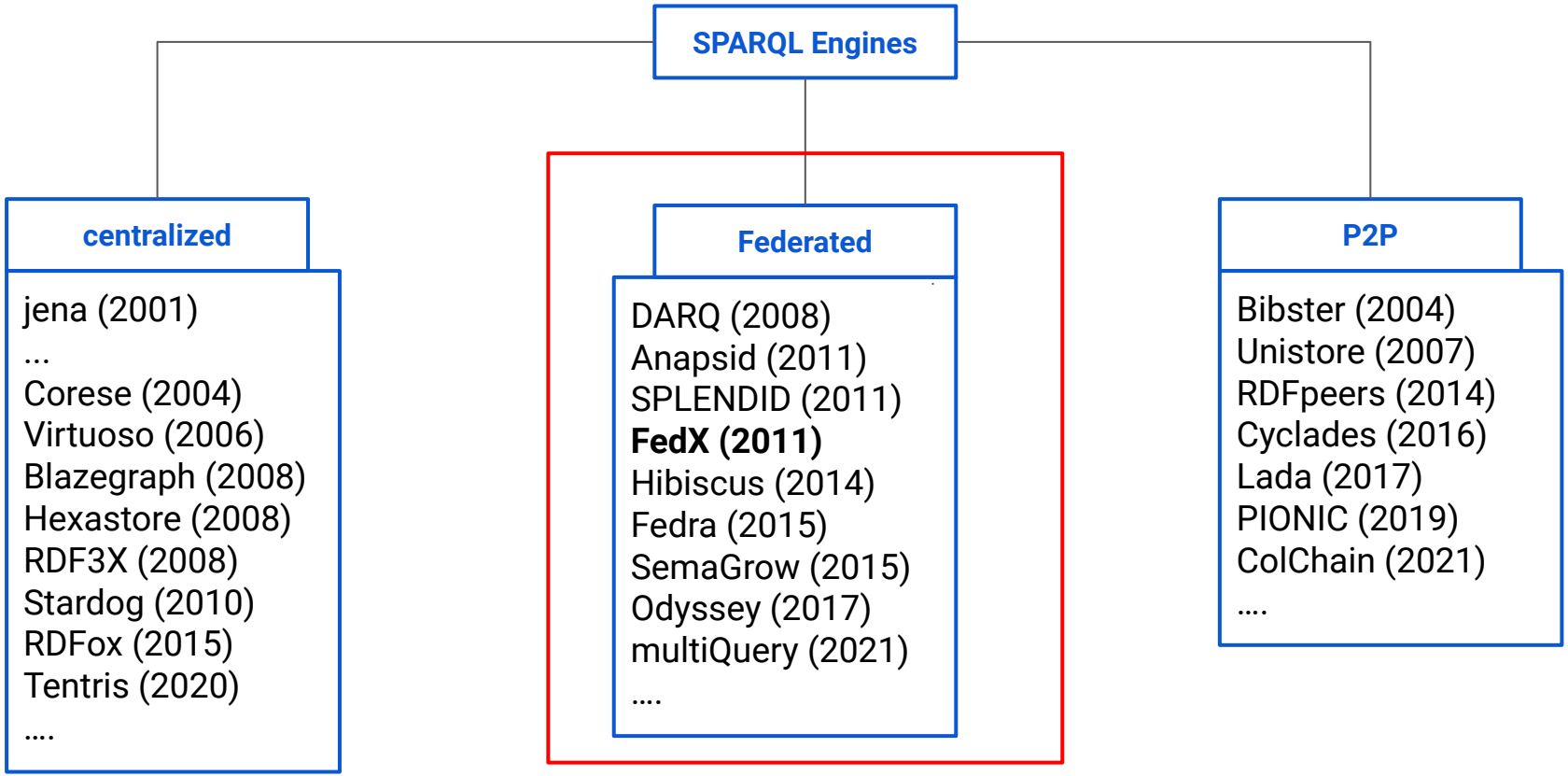
```
SELECT ?s
WHERE {
  ?s a foaf:Person .
  SERVICE <http://dbpedia.org/sparql>
{?s foaf:knows ?o }
SERVICE
<http://wikidata.org/sparql>
{?s foaf:knows ?o }
SERVICE
<http://LinkedMDB.org/sparql>
{?s foaf:knows ?o }

}
```

Do not scale

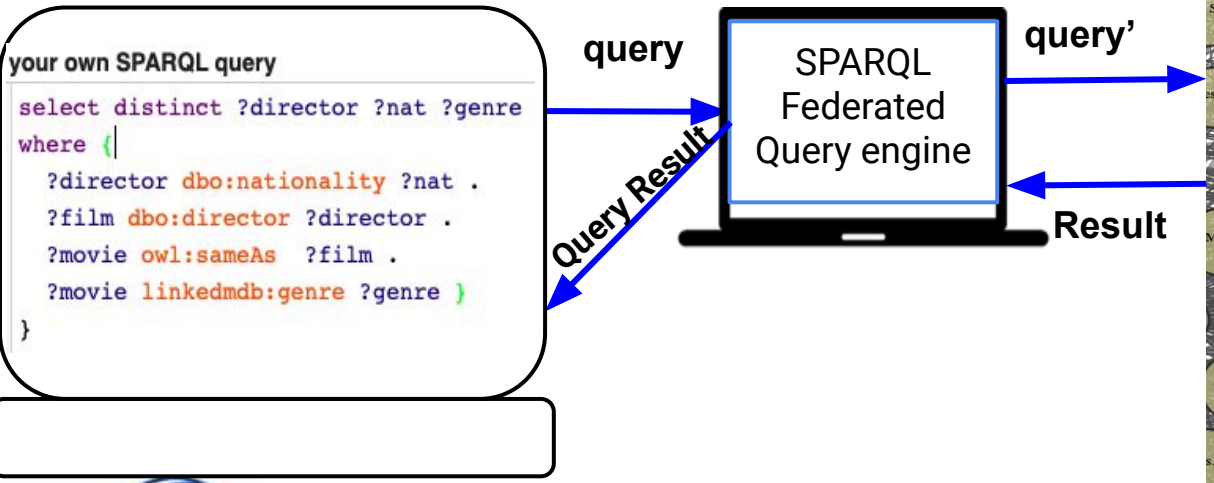


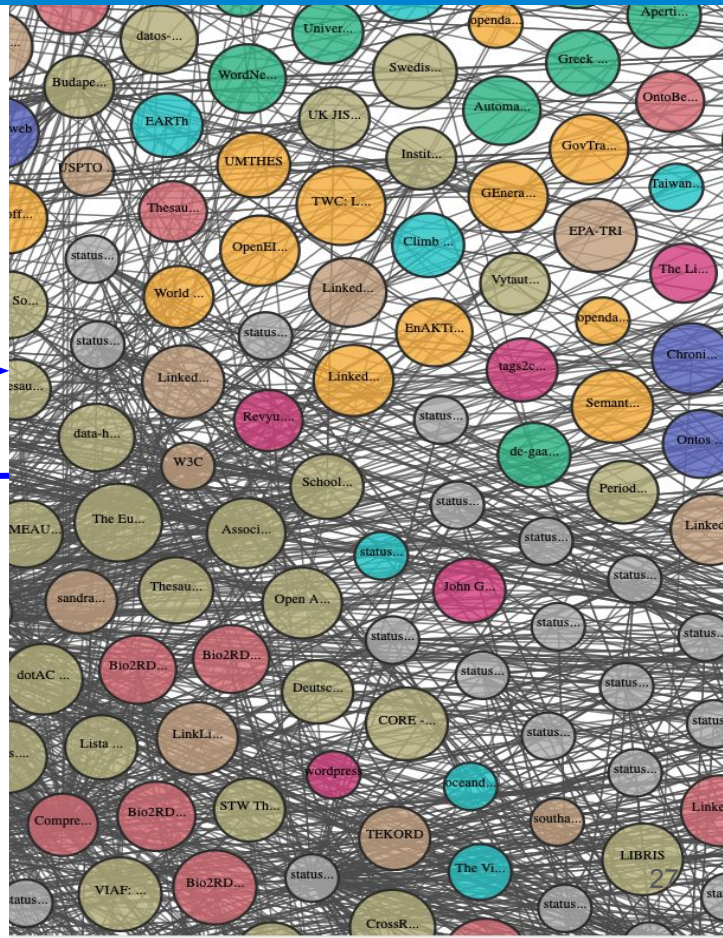
Storing and querying knowledge Graphs



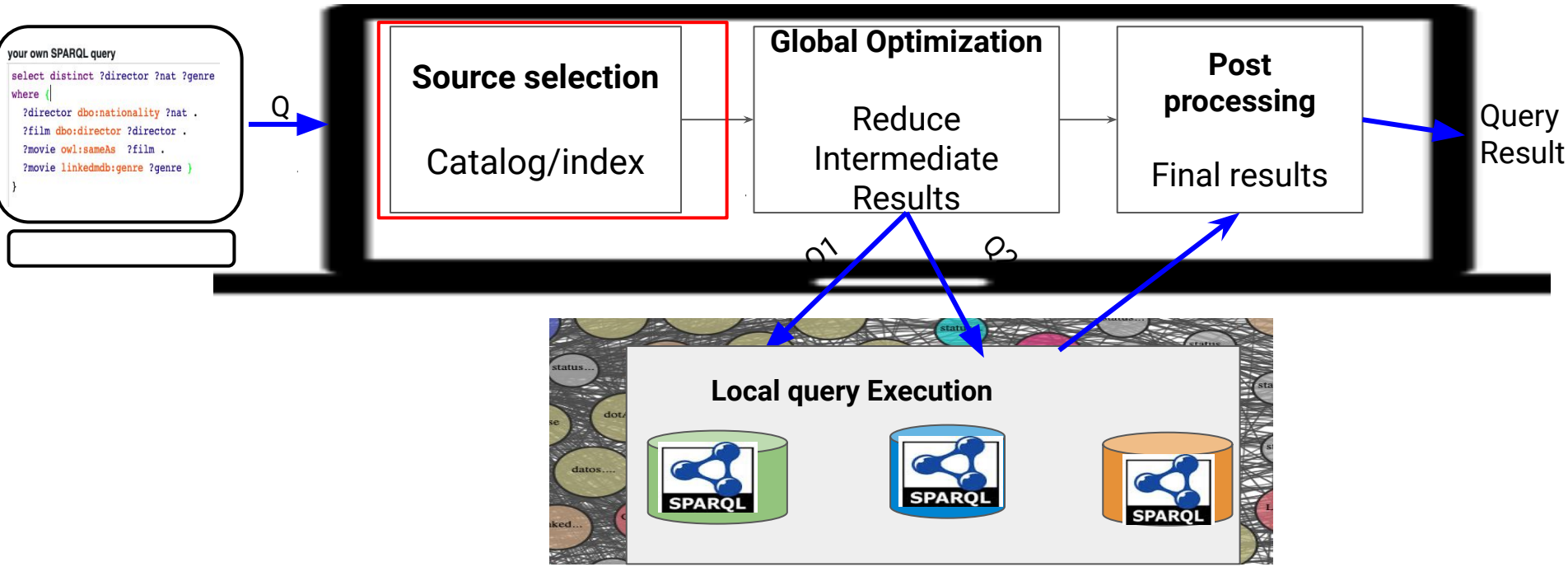
Federated SPARQL Query Engines

Enable efficient SPARQL query processing on virtually integrated Linked Data sources.





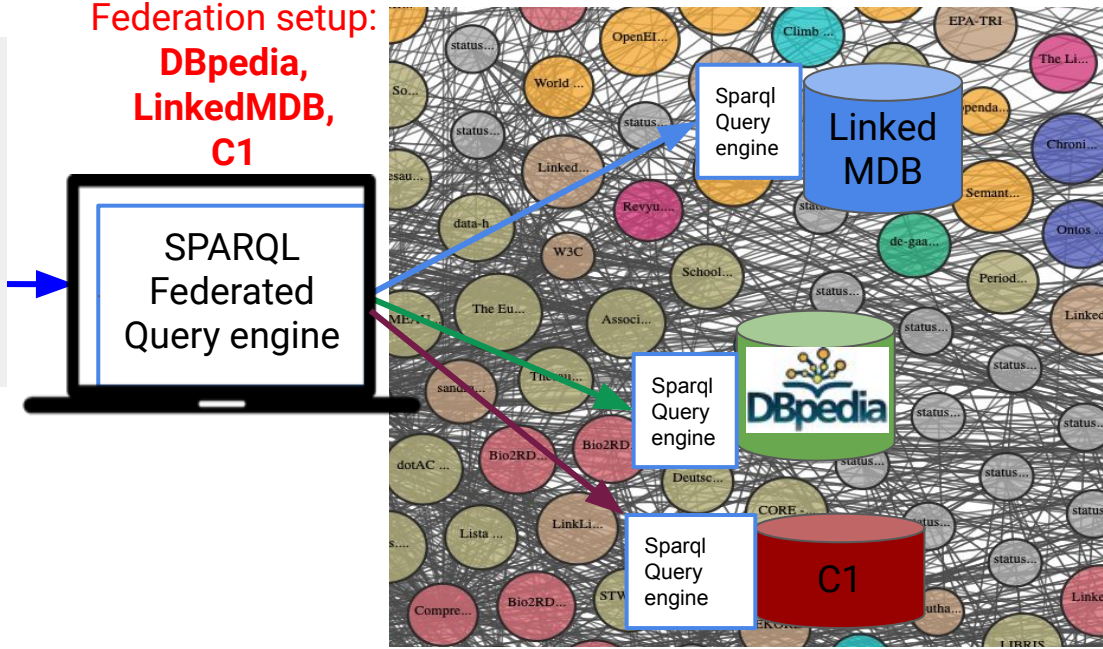
Inside Federated SPARQL Query Engines



Basic Source Selection

```
select distinct ?director ?nat   ?genre
where {
  ?director dbo:nationality ?nat .      TP1
  ?film dbo:director  ?director .       TP2
  ?movie owl:sameAs  ?film .            TP3
  ?movie linkedmdb:genre ?genre     TP4
}
```

Federation setup:
DBpedia, LinkedMDB, C1

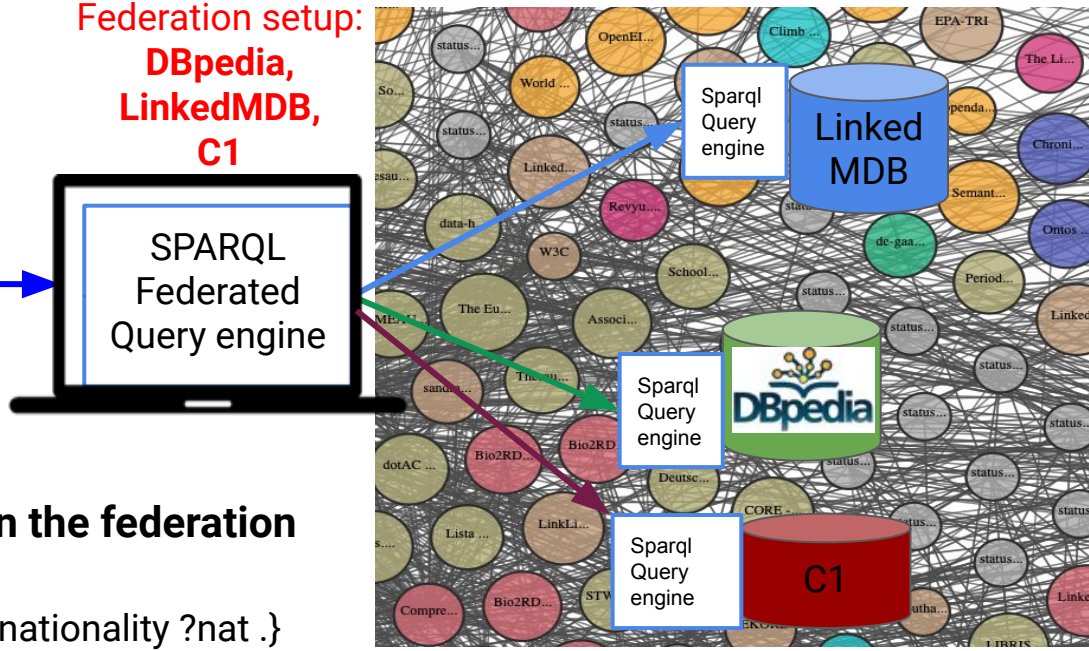


Andreas Schwarte et al. "Fedx: Optimization techniques for federated query processing on linked data."
International Semantic Web Conference, ISWC 2011

Basic Source Selection

```
select distinct ?director ?nat   ?genre
where {
  ?director dbo:nationality ?nat .      TP1
  ?film dbo:director  ?director .       TP2
  ?movie owl:sameAs  ?film .            TP3
  ?movie linkedmdb:genre ?genre         TP4
}
```

Federation setup:
DBpedia, LinkedMDB, C1



Send SPARQL ASK to each endpoint in the federation

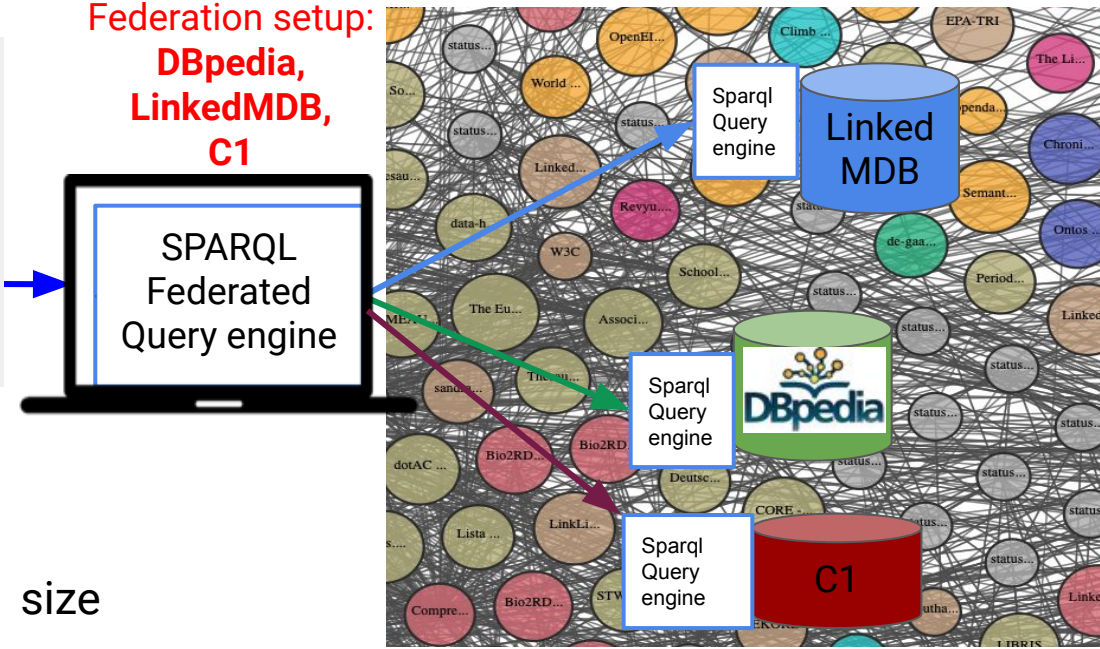
SPARQL ASK to dbpedia {?director dbo:nationality ?nat .}
SPARQL ASK to wikidata {?director dbo:nationality ?nat .}
…….

Andreas Schwarte et al. "Fedx: Optimization techniques for federated query processing on linked data."
International Semantic Web Conference, ISWC 2011

Basic Source Selection

```
select distinct ?director ?nat   ?genre
where {
  ?director dbo:nationality ?nat .      TP1
  ?film dbo:director  ?director .       TP2
  ?movie owl:sameAs  ?film .            TP3
  ?movie linkedmdb:genre ?genre         TP4
}
```

Federation setup:
DBpedia, LinkedMDB, C1



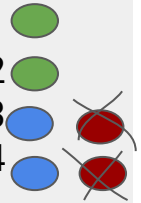
Hardly scale with number of sources, size of queries and unbounded predicate

Andreas Schwarte et al. "Fedx: Optimization techniques for federated query processing on linked data."
International Semantic Web Conference, ISWC 2011

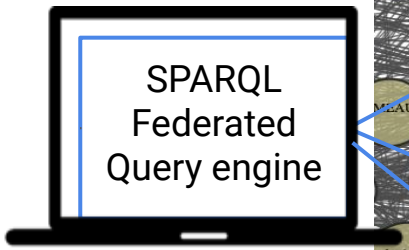
Complex Source selection

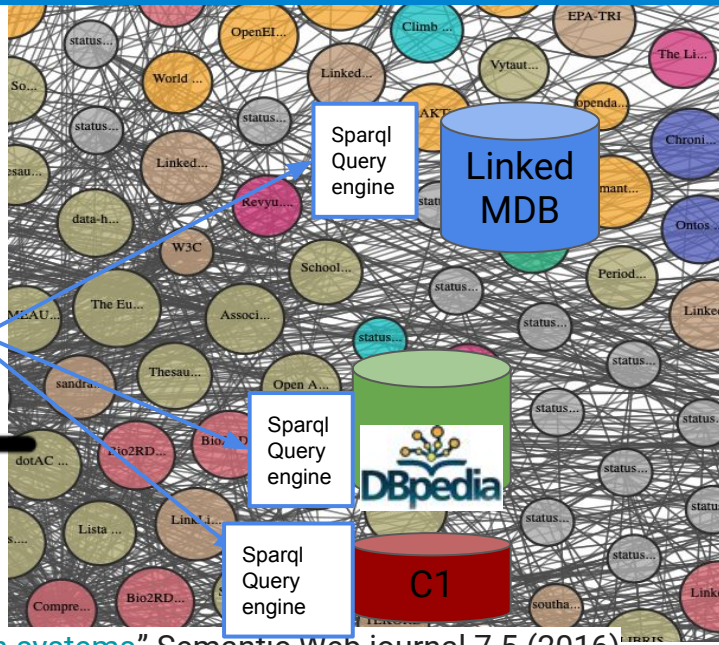
Use prefix +ASK

```
select distinct ?director ?nat   ?genre
where {
  ?director dbo:nationality ?nat .      TP1
  ?film dbo:director  ?director .       TP2
  ?movie owl:sameAs  ?film .            TP3
  ?movie linkedmdb:genre ?genre         TP4
}
```



Federation setup:
DBpedia, LinkedMDB, C1





Better but still we need ASK ..

Saleem, Muhammad, et al. “A fine-grained evaluation of SPARQL endpoint federation systems”.Semantic Web journal 7.5 (2016)
G Montoya, H Skaf-Molli, P Molli, ME Vidal. “Decomposing federated queries in presence of replicated fragments”. Journal of Web Semantics, Elsevier 2017.
G Montoya, H Skaf-Molli, K Hose. “The Odyssey approach for optimizing federated SPARQL queries” International Semantic Web Conference, 2017.
P. Peng, Q. Ge, L. Zou, M. T. Özsu, Z. Xu and D. Zhao, "Optimizing Multi-Query Evaluation in Federated RDF Systems," in IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 4, 2021

Everything seems to be great knowledge is accessible.
What is the problem ?

The Real Story

Public SPARQL Endpoints are not really accessible.

They do not allow to execute any SPARQL query and get complete results.

What are the birthplaces of all movie actors?

DBpedia endpoint yields partial results...

Virtuoso SPARQL Query Editor

Default Data Set Name (Graph IRI)

http://dbpedia.org

Query Text

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?actor ?birthPlace
WHERE {
        ?x a dbo:Actor;
        rdfs:label ?actor;
        dbo:birthPlace ?city.
  ?city a dbo:City;
        rdfs:label ?birthPlace.
}
```

actor	birthPlace
"منغ جيا"@ar	"Loudi"@en
"منغ جيا"@ar	"Loudi"@de
"منغ جيا"@ar	"Loudi"@es
"منغ جيا"@ar	"Loudi"@fr
"منغ جيا"@ar	"Loudi"@it
"منغ جيا"@ar	"婁底市"@ja
"منغ جيا"@ar	"Loudi"@nl
"منغ جيا"@ar	"Loudi"@pl
"منغ جيا"@ar	"Loudi"@pt
"منغ جيا"@ar	"Лоуди"@ru
"منغ جيا"@ar	"娄底市"@zh
"Meng Jia"@es	"Loudi"@en

X-SPARQL-MaxRows: 10000

Only **10 000** results out of **35 215** expected !!

Try it yourself: [Actor and city on dbpedia]

Aggregate queries online on public SPARQL endpoints

Ex: Number of objects per class

```
1 select (count (?o) as ?x) ?c where {
2   ?s a ?c ; ?p ?o
3   } group by ?c
```

On Dbpedia: Partial Results

Virtuoso SPARQL Query Editor

Default Data Set Name (Graph IRI)

http://dbpedia.org

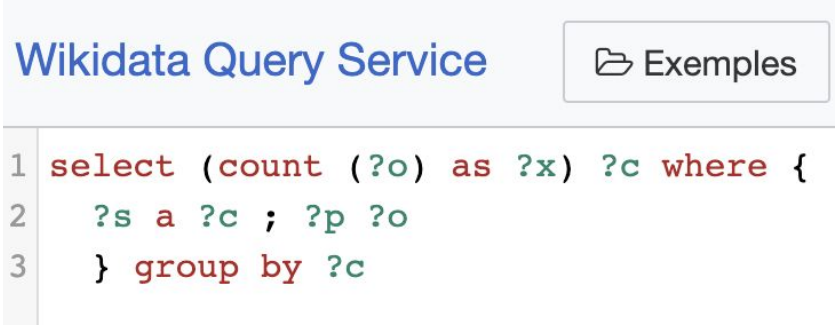
Query Text

```
 SELECT (COUNT(?o) AS ?x) ?c WHERE {
?s a ?c ; ?p ?o
} GROUP BY ?c
```

x	c
1	http://dbpedia.org/class/yago/WikicatMinesweepersOfTheFijianNavy
6	http://dbpedia.org/class/yago/WikicatParalympicBronzeMedalistsForLatvia
1	http://dbpedia.org/class/yago/WikicatFoundationsBasedInRussia
1	htt

X-SPARQL-MaxRows: 10000

On Wikidata: Timeout




```
SPARQL-QUERY: queryStr=select (count (?o) as ?x) ?c where {
  ?s a ?c ; ?p ?o
  } group by ?c
java.util.concurrent.TimeoutException
        at java.util.concurrent.FutureTask.get(FutureTask.java
```

Retrieve creative works and the list of fictional works that inspired them

Wikidata Query Service

Exemples

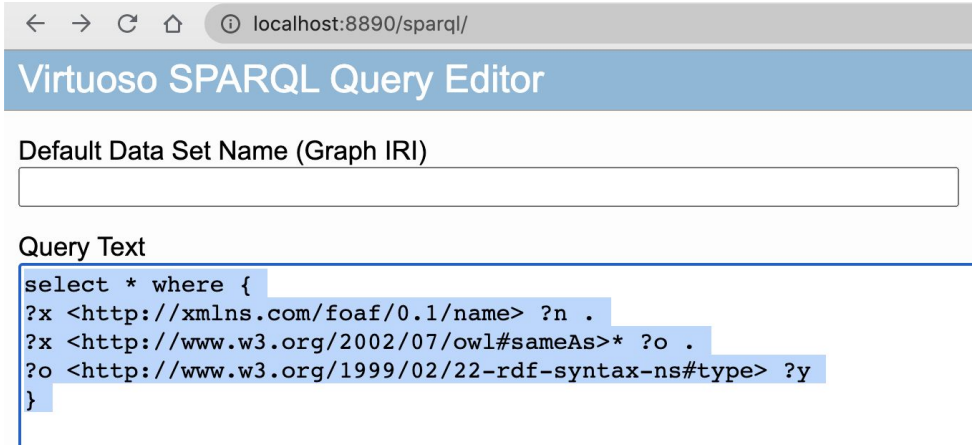
```
SELECT ?oeuvre ?inspiration
WHERE {
  ?inspiration wdt:P136 wd:Q8253 .
  ?oeuvre wdt:P144 ?inspiration .
  ?oeuvre wdt:P31/wdt:P279* wd:Q17537576 .
}
```

Property Path

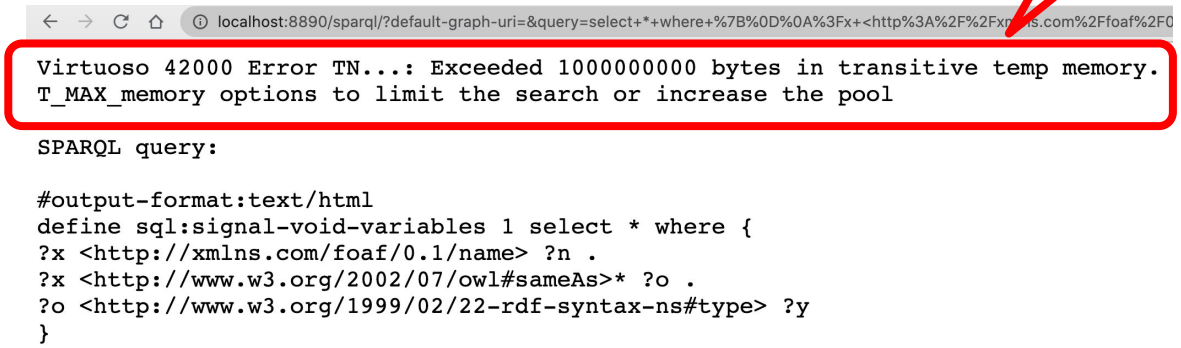
Match path of arbitrary length !

On Wikidata: No Results

On DBpedia: Partial results



The query is killed by quotas



Fair Usage policy

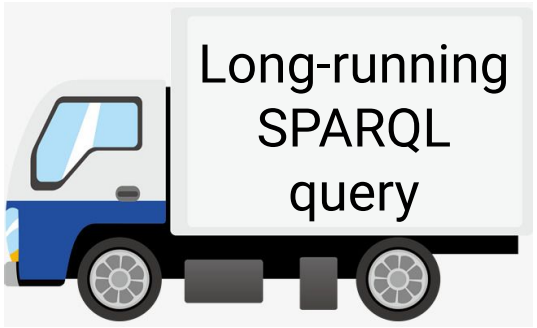
“**A Fair Use Policy** is in place in order to provide a stable and responsive endpoint for the community”

- Communication Quotas: Limit the arrival rate of queries per IP
- Space Quotas: Prevent one query to consume all the memory of the server
- **Time Quotas**: Avoid **convoy effect**

https://wiki.dbpedia.org/public-sparql-endpoint

Convoy effect

- **Convoy effect:** A long-running SPARQL query slows down short-running ones [1]
- **All threads** of the Web server can be **busy** with long queries

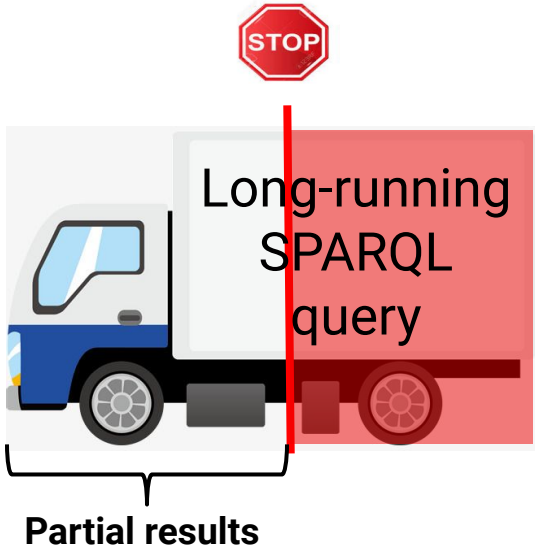


Short SPARQL
queries

[1] M.W. Blasgen, et al. *"The convoy phenomenon"*, In Operating Systems Review 13 (2) (1979)

Quotas prevent convoy effects

- Long-running queries are **interrupted** by quotas
- However, quotas also **deteriorate answer completeness!**
  - Interrupted queries only deliver **partial results**



Issues

Without Quotas

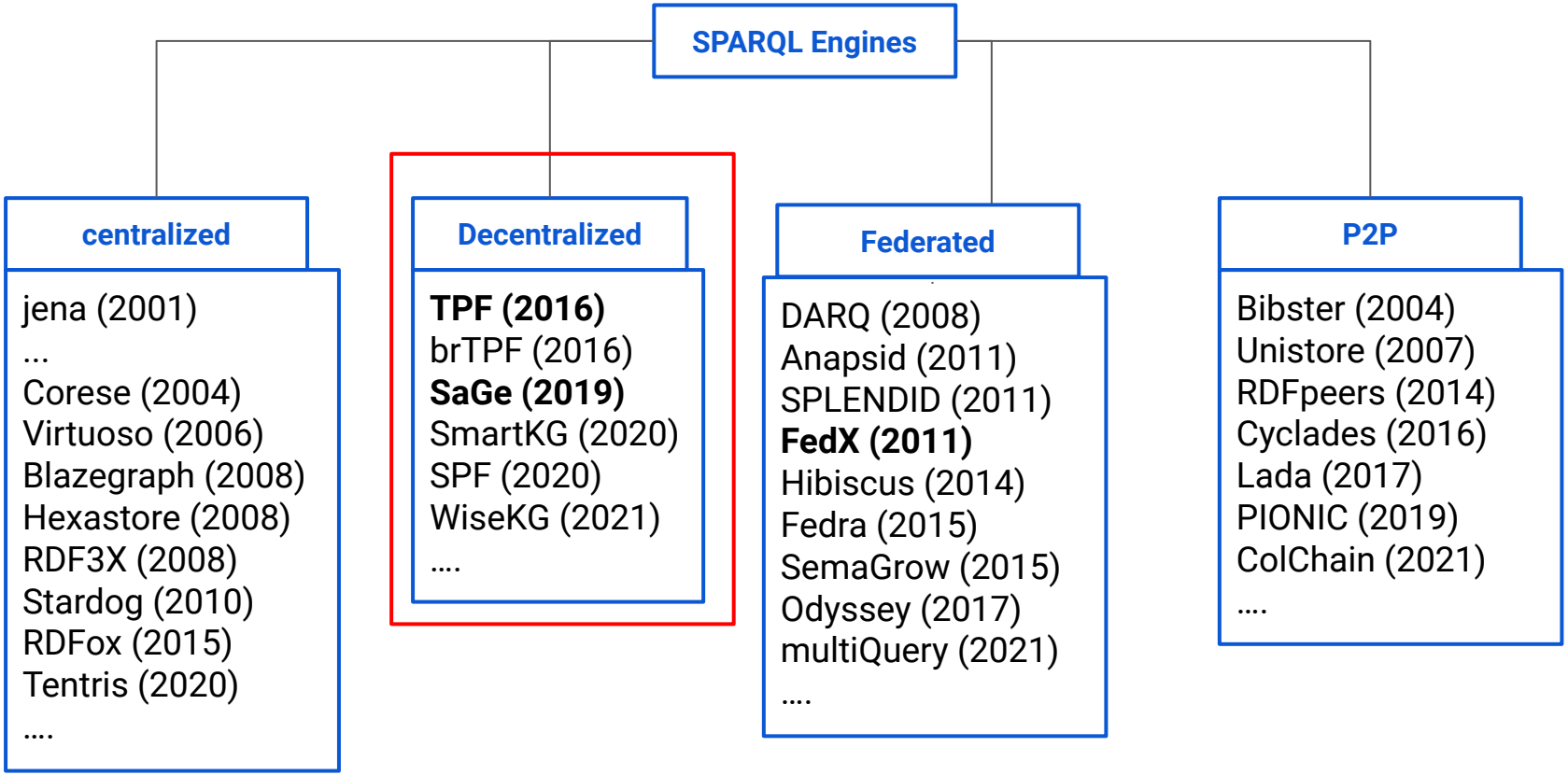
- Complete results
- server congestion
- Server not responsive

With Quotas

- Partial results
- Responsive server

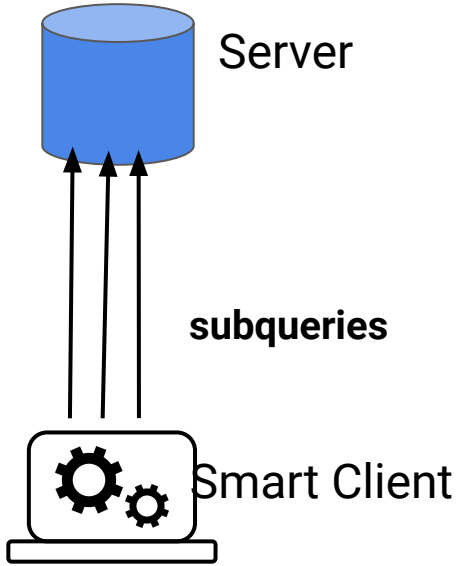
To be useful, Endpoints have to be responsive *and* deliver complete results !!

Storing and querying knowledge Graphs



Decentralization SPARQL query processing

- Share the load between servers and clients
  - Building simpler servers with restricted expressivity
  - Building more intelligent clients that contribute to the execution of the query

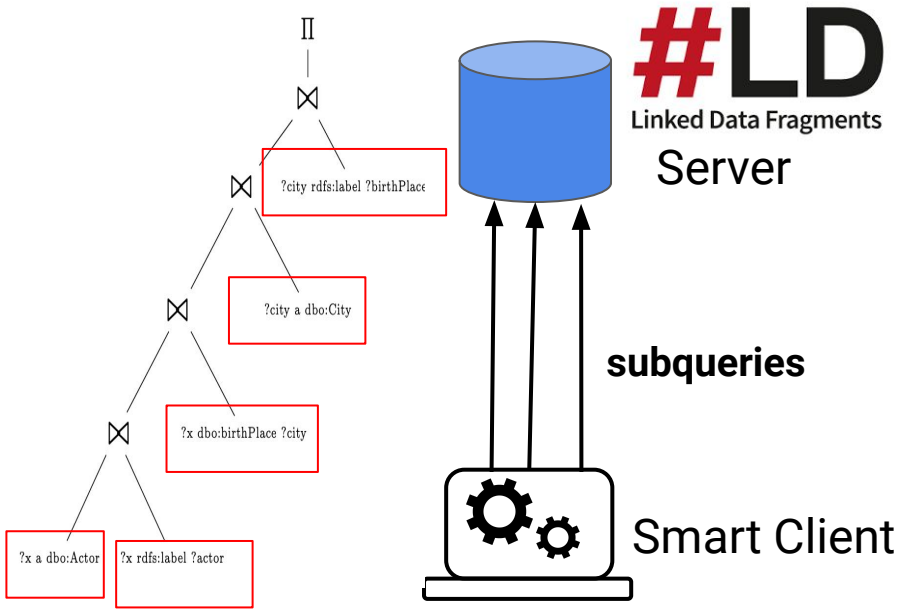


R. Verborgh, et al. "*Triple pattern fragments: A low-cost knowledge graph interface for the web*", Journal of Web Semantics (2016)

Restrict the server expressivity

Triple Pattern Fragments (TPF)

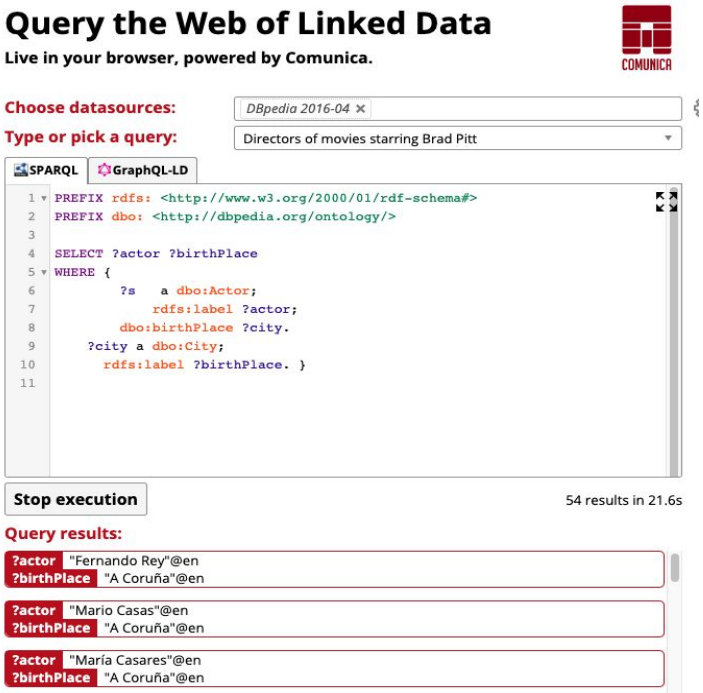
- The server supports **paginated triple patterns**
- A Smart client executes join, OPTIONAL, aggregate, property paths, etc



R. Verborgh, et al. "*Triple pattern fragments: A low-cost knowledge graph interface for the web*", Journal of Web Semantics (2016)

TPF with restricted web servers terminates but

- **Poor performance:**
  - What are the birthplaces of all movie actors?
  - **2h** query execution time
  - Large number of HTTP calls:
    - 507 156 HTTP requests sent
  - Huge Data transfer:
    - 2GB of Data Transfers
- **Too much calls and data transfer.**

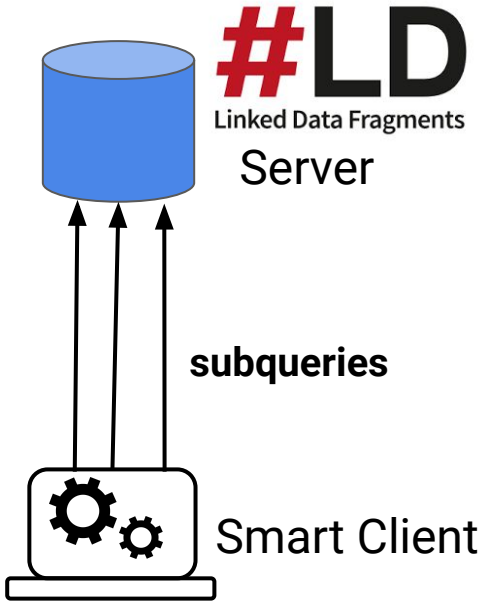


http://query.linkeddatafragments.org/

Optimizations but still restricted expressivity

Cyclade: Collaborative Cache
Ladda: Collaborative query processing

brTPF: allows clients to attach intermediate results to TPF requests (bind join strategy)
SMART-KG: The server ships graph partitions per star-shaped patterns to the client
WiseKG: SMART-KG with cost model to balance star-shaped patterns between server-client



P. Folz, H. Skaf-Molli et P. Molli (2016). CyClaDEs: A Decentralized Cache for Triple Pattern Fragments. 13th Extended Semantic Web Conference 2016
A. Grall, P. Folz, G. Montoya, H. Skaf-Molli, P. Molli, M. V. Sande, and R. Verborgh. Ladda: SPARQL Queries in the Fog of Browsers. Demo ESWC 2017

Hartig, Olaf, et al. "Bindings-restricted triple pattern fragments." *OTM - On the Move to Meaningful Internet Systems*". 2016.
Azzam, Amr, et al. "SMART-KG: hybrid shipping for SPARQL querying on the web." *Proceedings of The Web Conference 2020*
Azzam, Amr, et al. "WiseKG: Balanced Access to Web Knowledge Graphs." *Proceedings of the Web Conference 2021*.

Endpoints vs Restricted Interfaces

SPARQL Endpoints with quotas

- Fast
- But, partial results

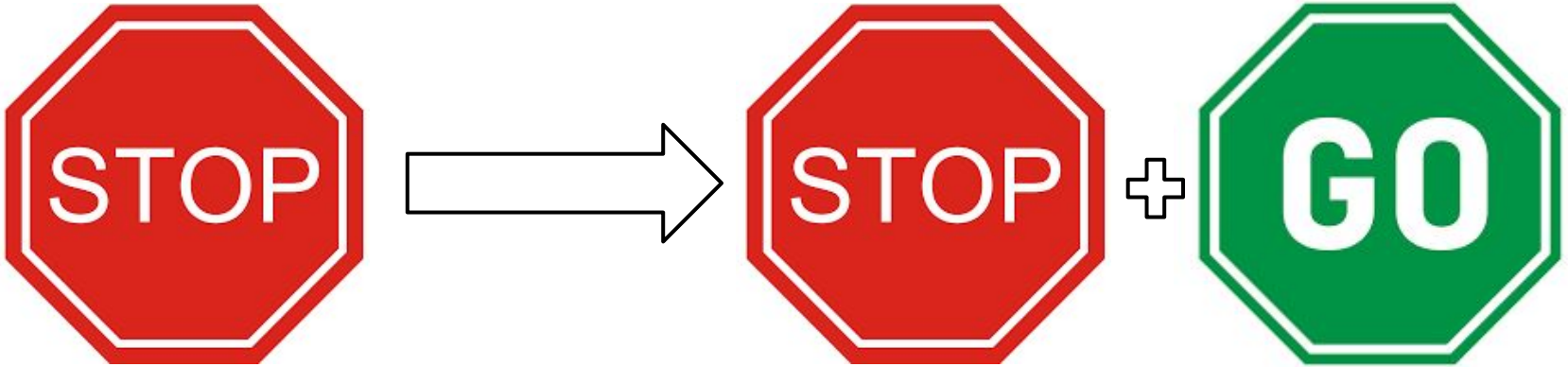
Restricted Interfaces

- Complete results...
- But, huge data transfer and poor performances

We need completeness and performances !

Our Approach: Web Preemption

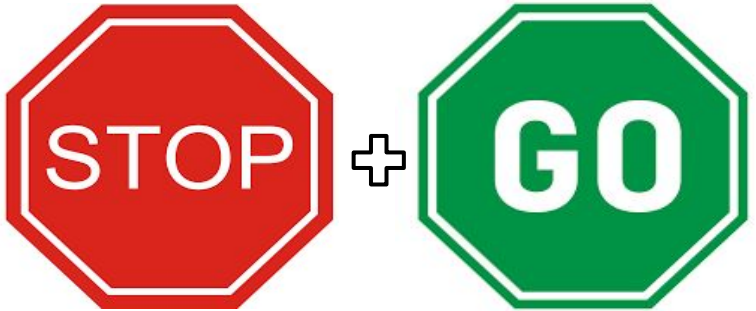
Query interruption is not an issue if the query can be resumed later



Web Preemption

We define **Web preemption** as:

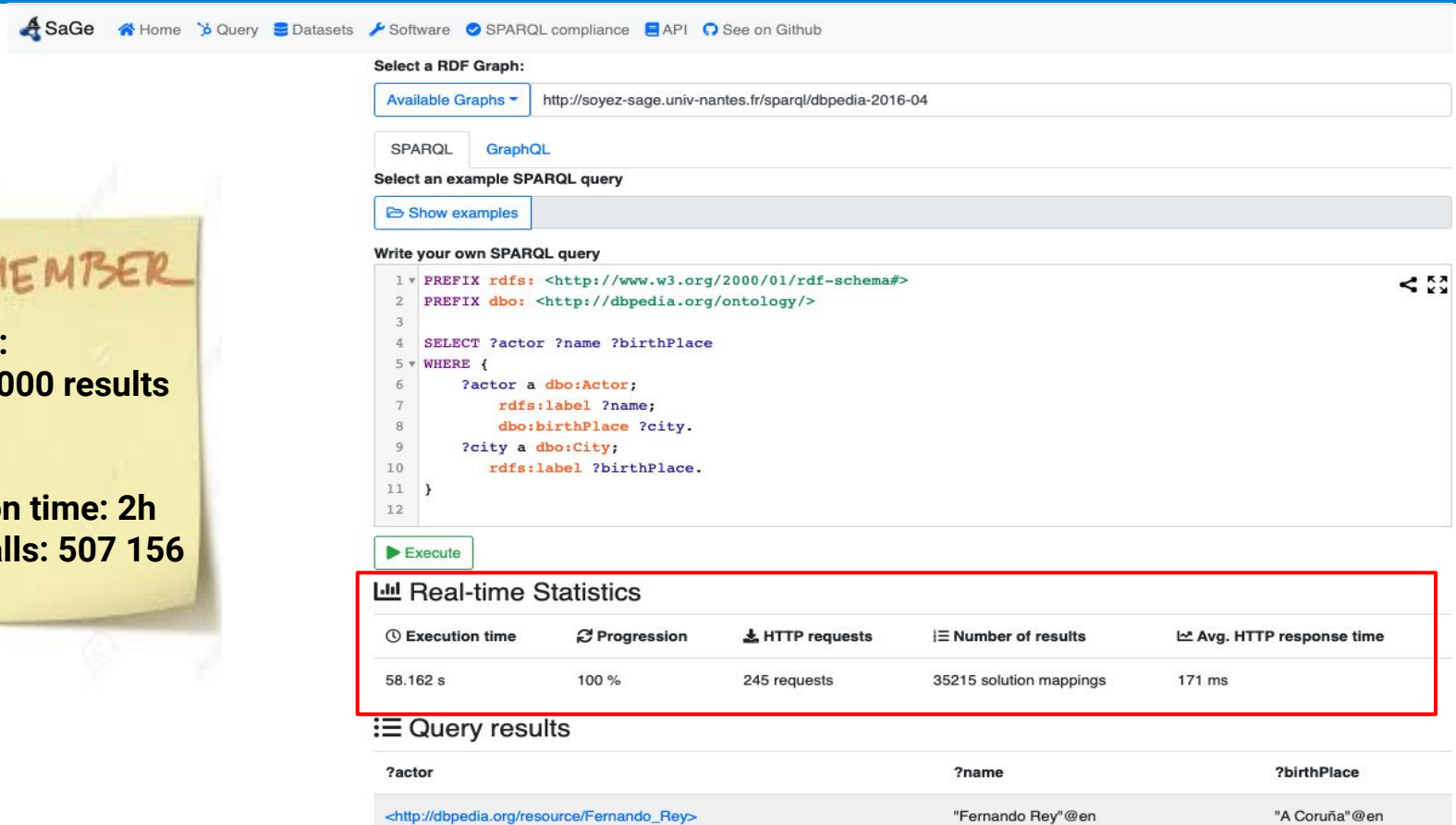
*"The capacity of a Web server to **suspend** a running query after a **time quantum** with the intention to **resume** it later."*



Thomas Minier, Hala Skaf-Molli and Pascal Molli. "SaGe: Web Preemption for Public SPARQL Query services" in Proceedings of the 2019 World Wide Web Conference (WWW') 2019.

What are the birthplaces of all movie actors?

http://sage.univ-nantes.fr

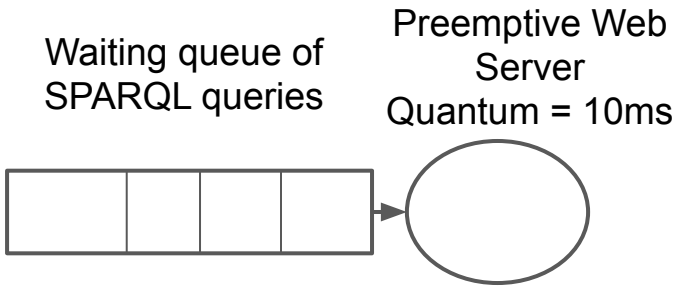


REMEMBER

DBpedia:
Only 10 000 results

TPF:
Execution time: 2h
HTTP calls: 507 156

Web Preemption in action

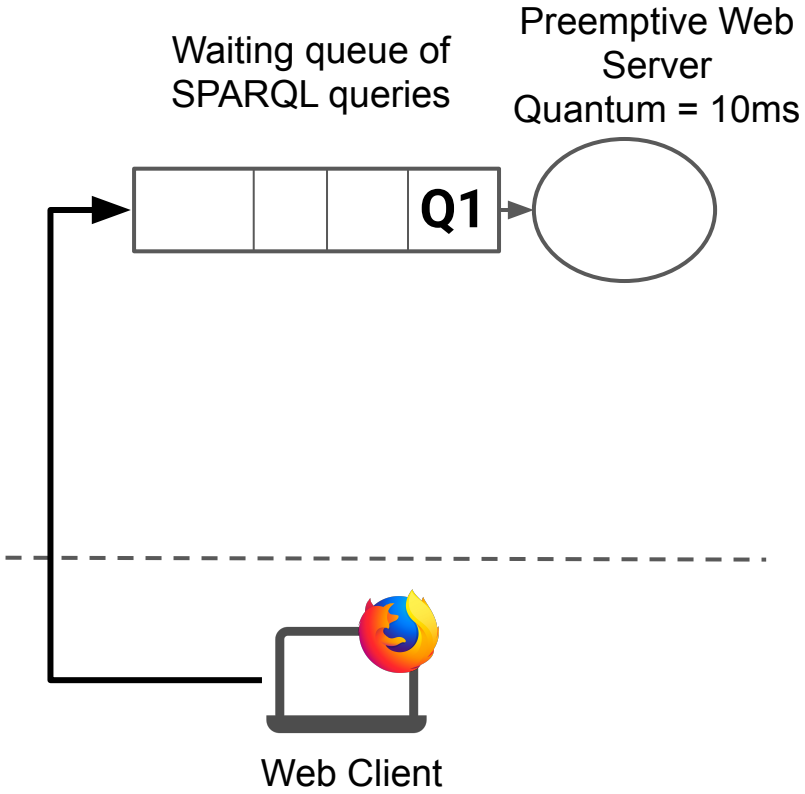


Q1

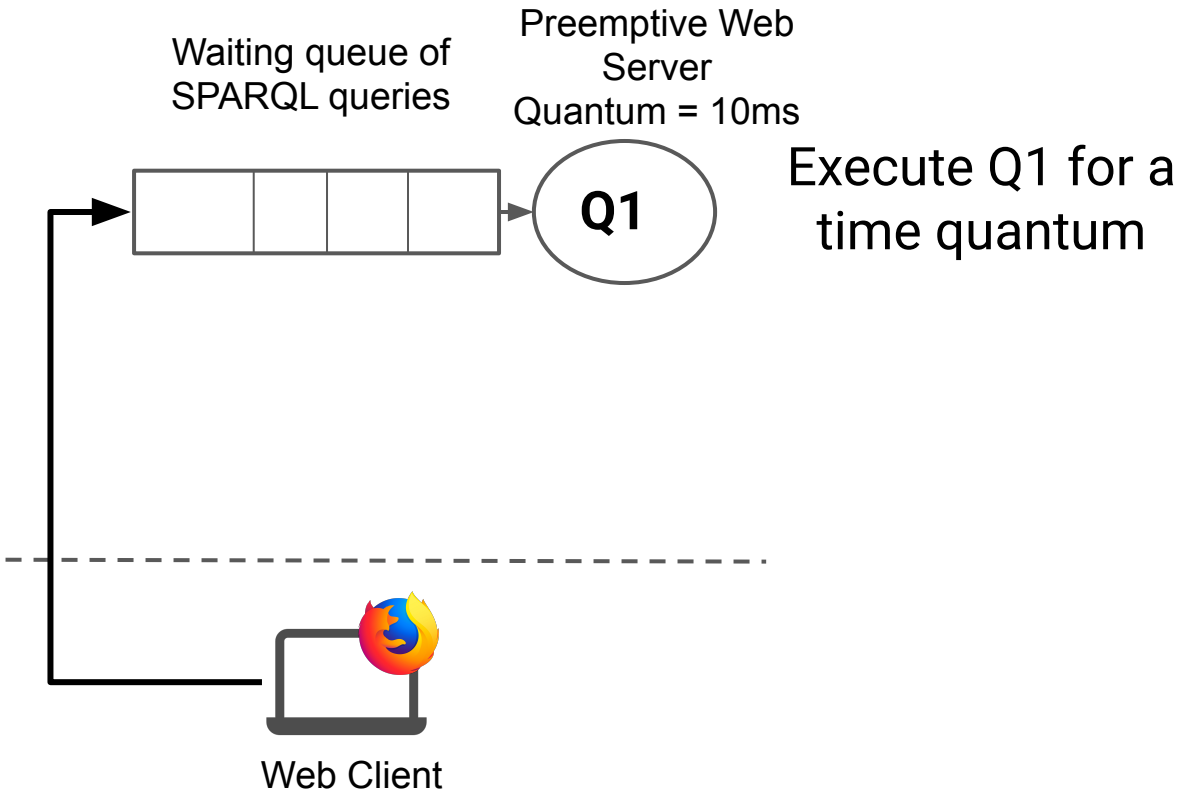
```
SELECT ?actor ?name ?birthPlace WHERE { ------------------------.
  ?actor a dbo:Actor;
         rdfs:label ?name;
         dbo:birthPlace ?city.
  ?city a dbo:City;
        rdfs:label ?birthPlace.
}
```

Web Client

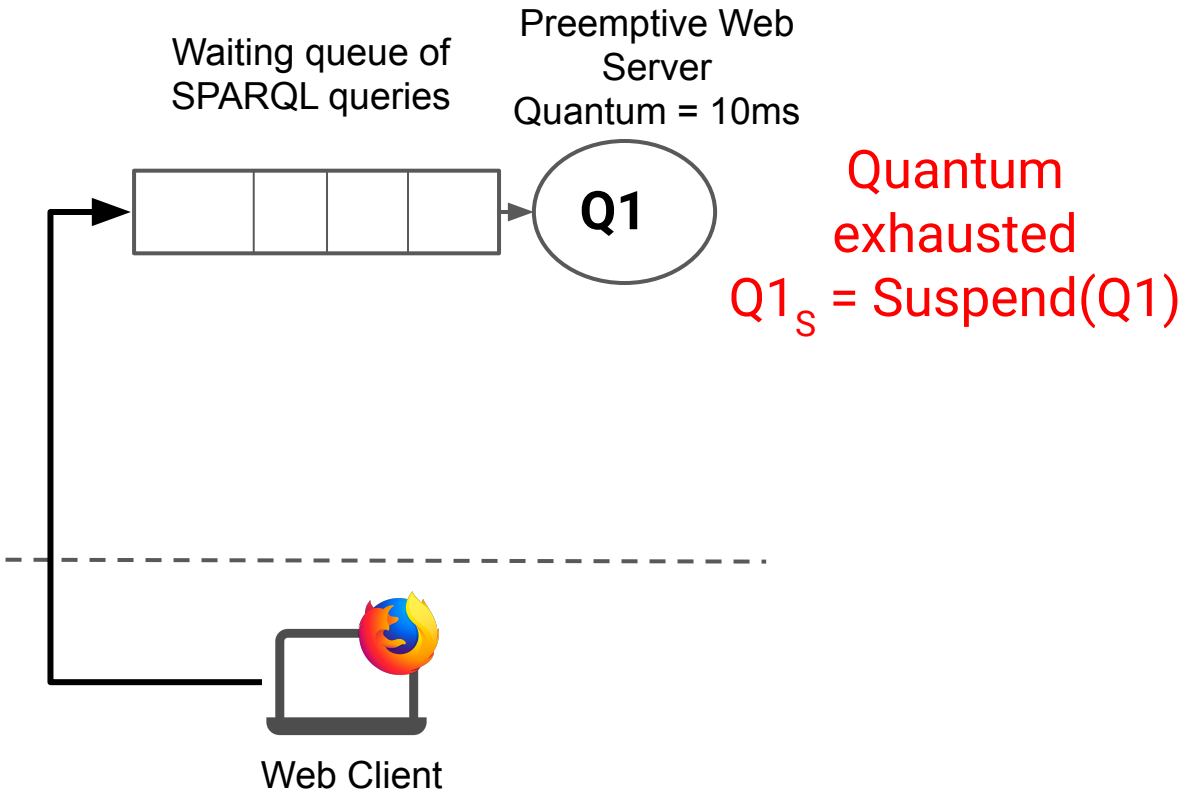
Web Preemption in action



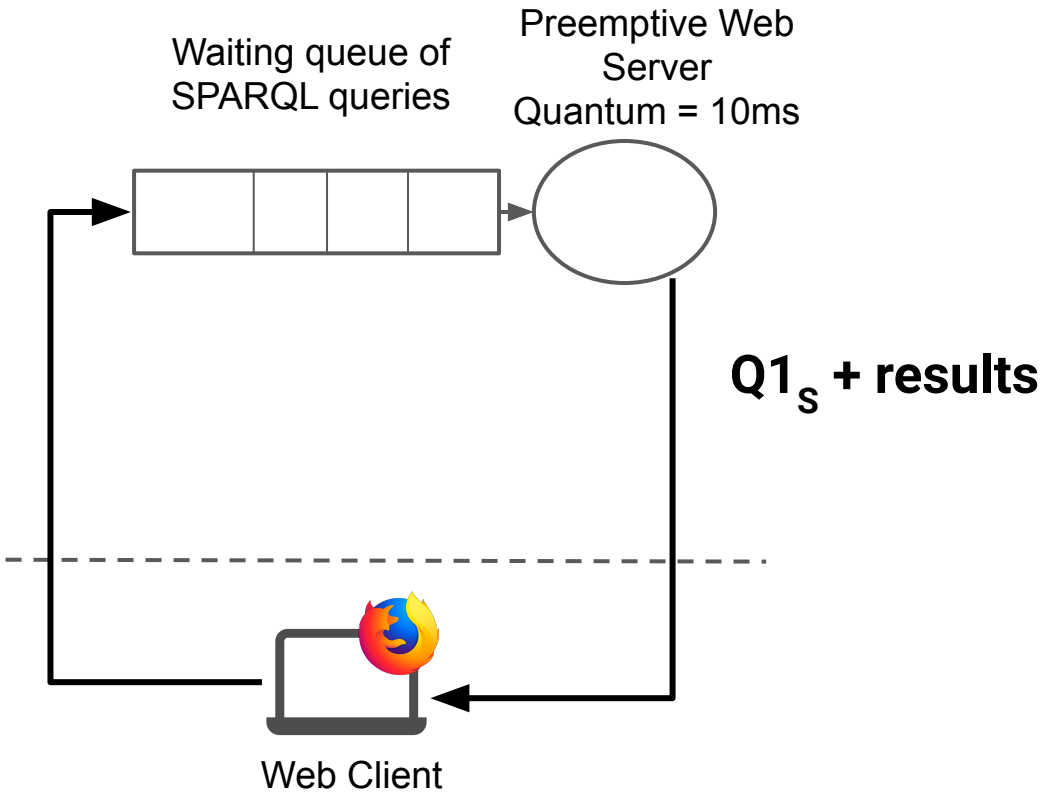
Web Preemption in action



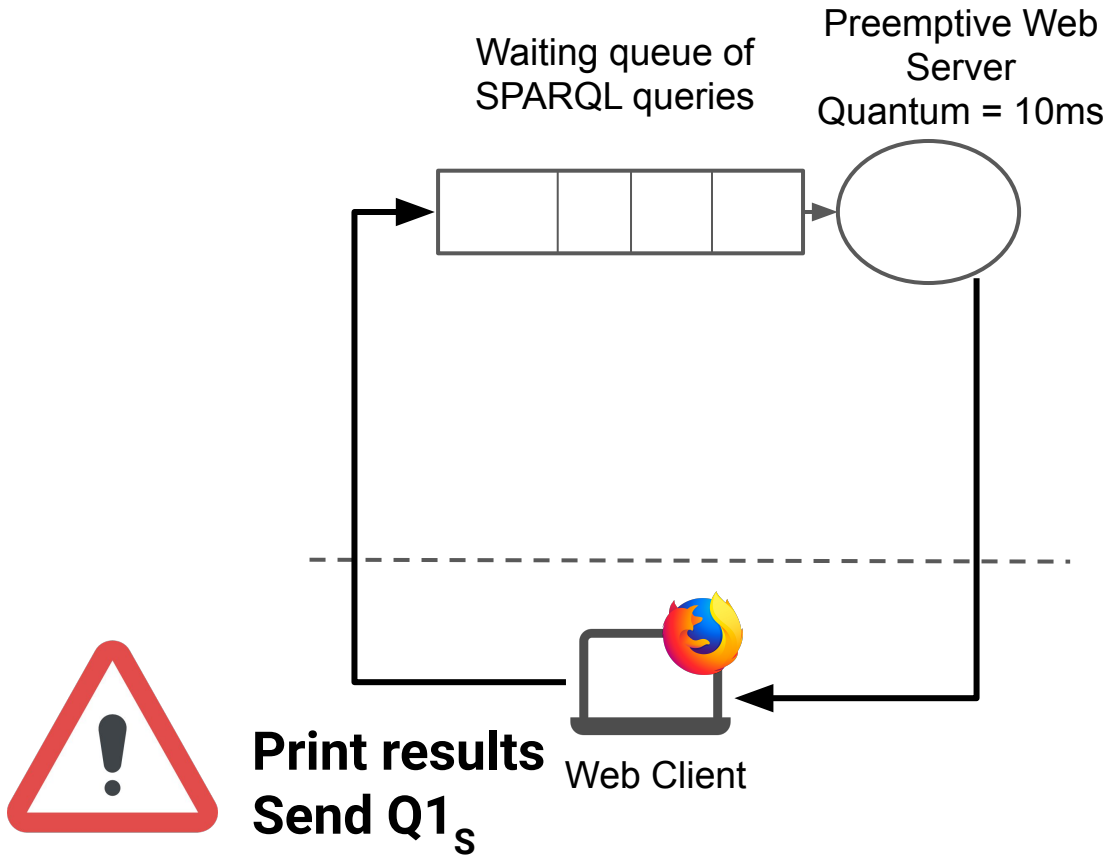
Web Preemption in action



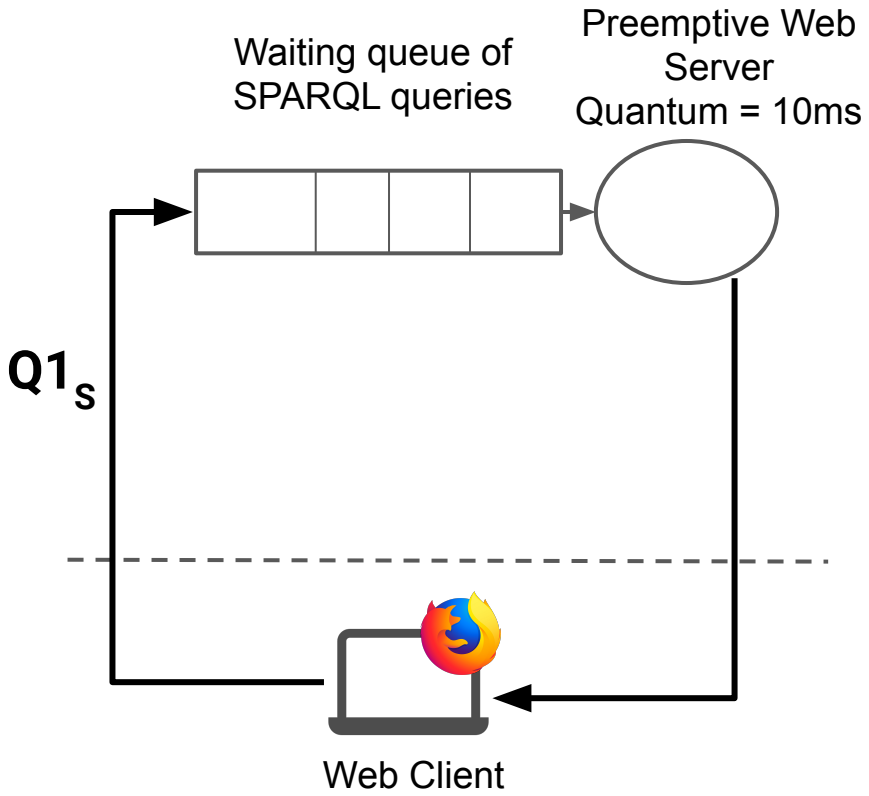
Web Preemption in action



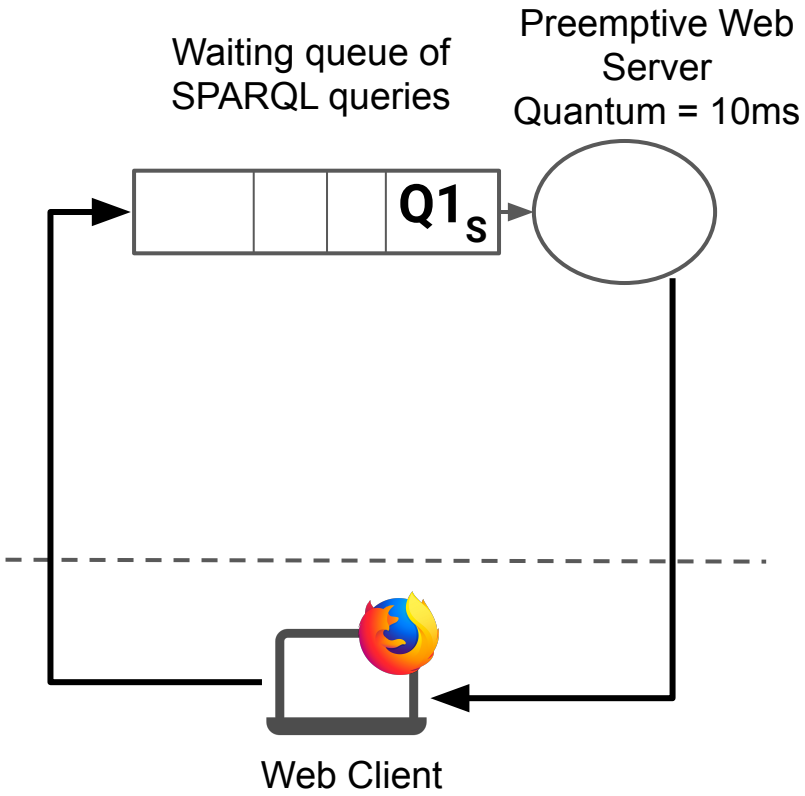
Web Preemption in action



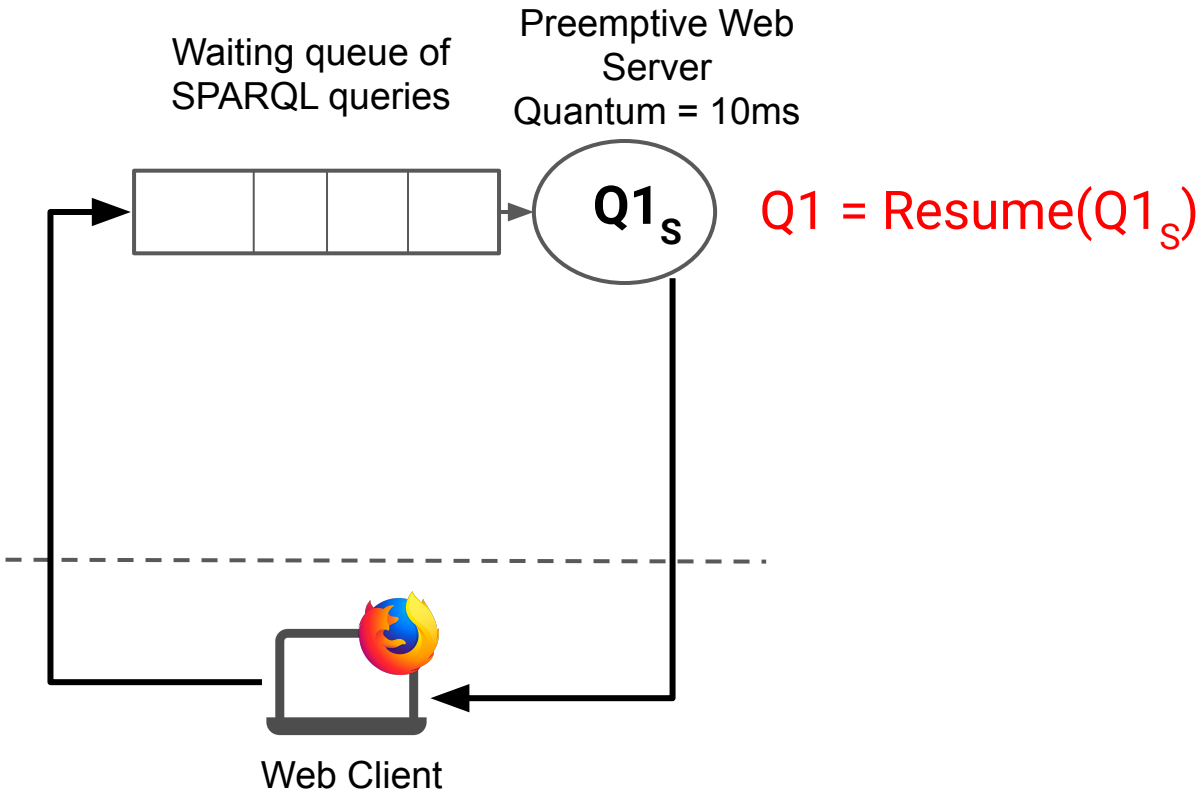
Web Preemption in action



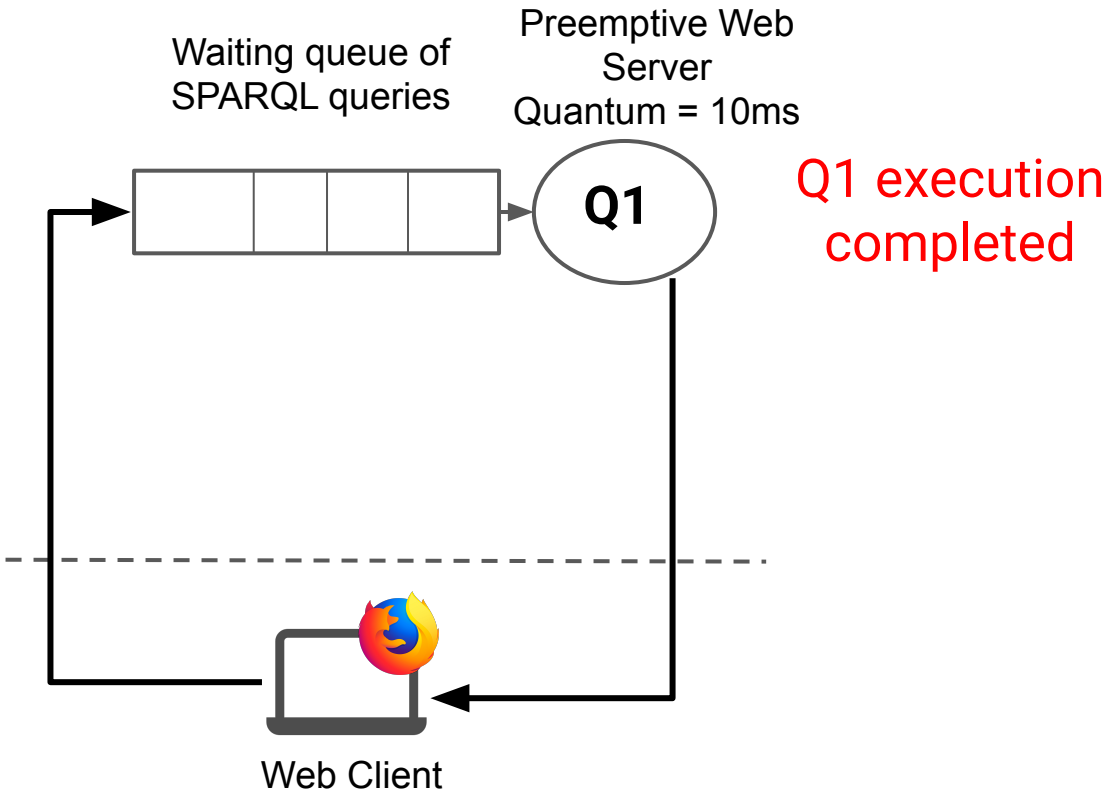
Web Preemption in action



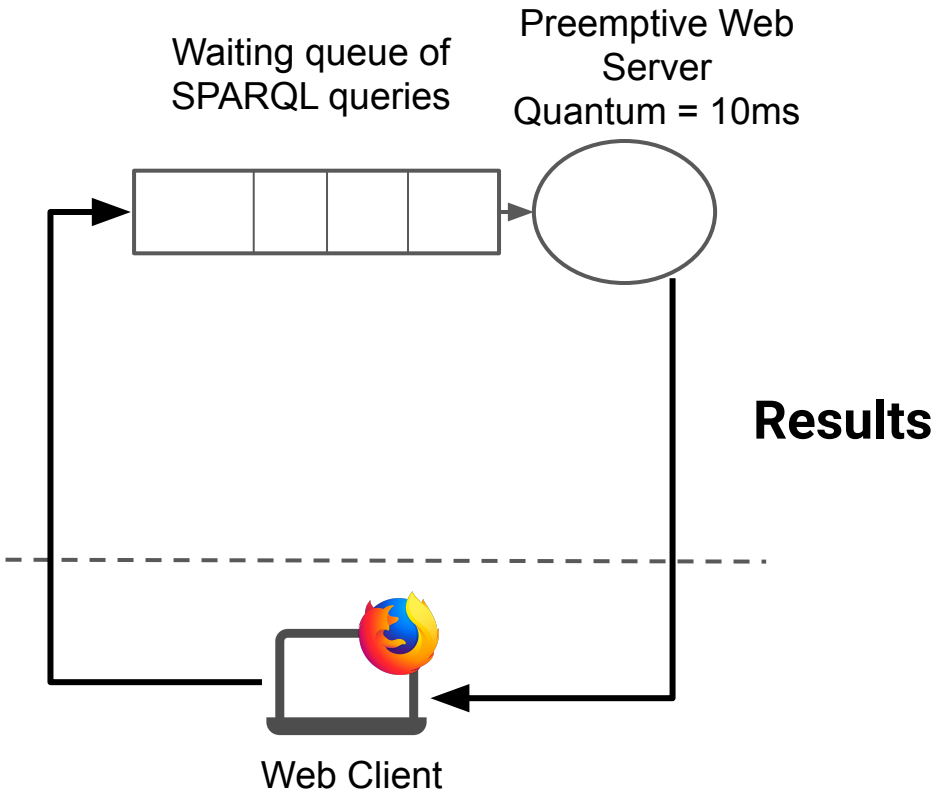
Web Preemption in action



Web Preemption in action



Web Preemption in action



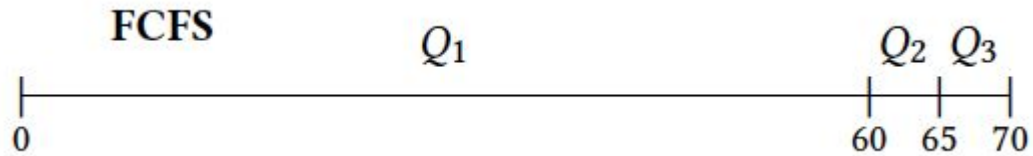
Advantages of Web Preemption

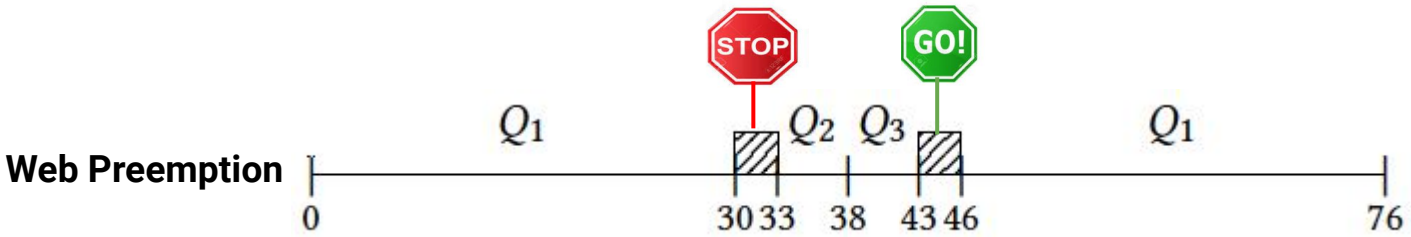
- Fair Sparql Endpoint by design
  - No quotas, no time-out.
- Better **average completion time**
- Better **time for first results**

First-Come First-Served (FCFS) vs Web Preemption

Queries execution times

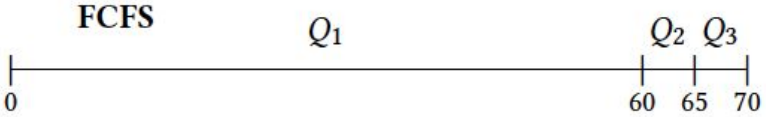
- Q1: 60s
- Q2: 5s
- Q3: 5s

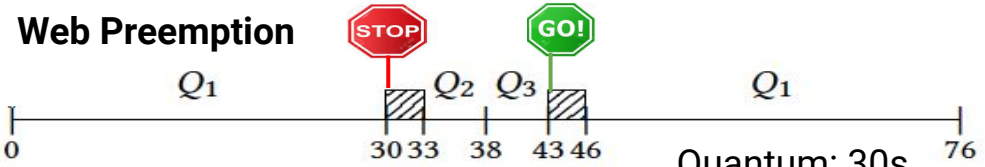




Time Quantum: 30s
Preemption overhead: 3s for Suspend/Resume

First-Come First-Served (FCFS) vs Web Preemption





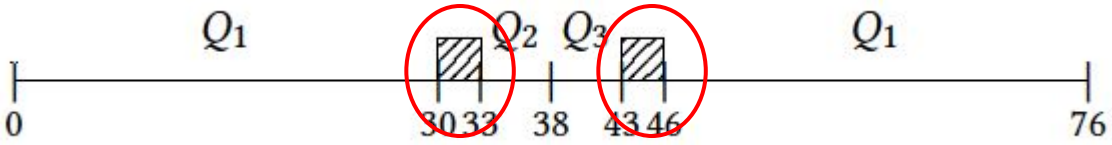
Quantum: 30s
Overhead: 3s

	FCFS	Web Preemption
Throughput (query/second)	$\frac{3}{70} = 0,042$	$\frac{3}{76} = 0.039$
Average query completion time (s)	$\frac{60+65+70}{3} = 65s$	$\frac{76+38+43}{3} = 52.3s$
Average time for first results (s)	$\frac{60+65+70}{3} = 65s$	$\frac{30+38+43}{3} = 37s.$

Preemption overhead

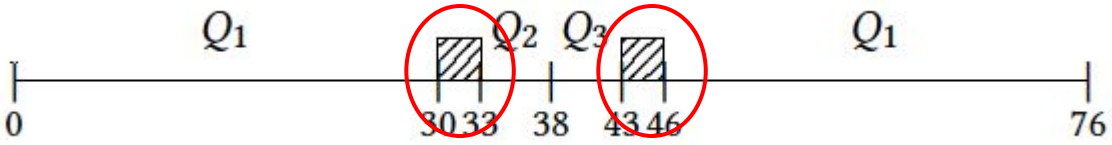
The **preemption overhead** is the time to suspend a running query and to resume the next waiting query

Objective: Minimize the preemption overhead



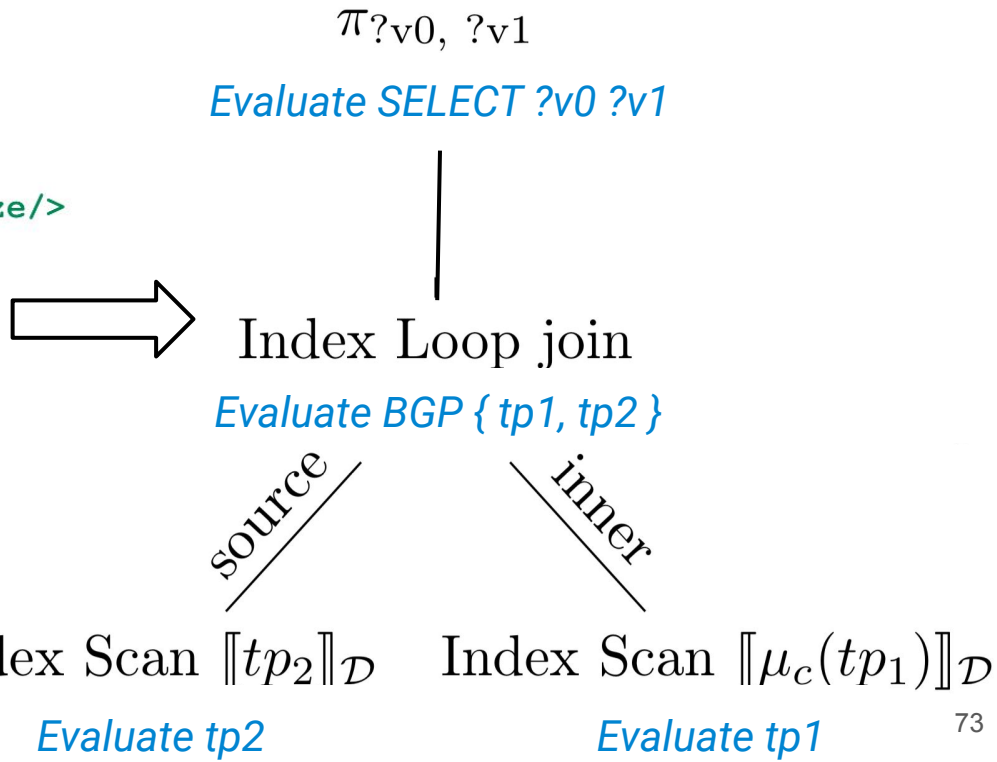
Minimizing the overhead

- Minimize the complexity in **time** of Suspend and Resume
- Minimize the complexity in **space** of Suspend and Resume



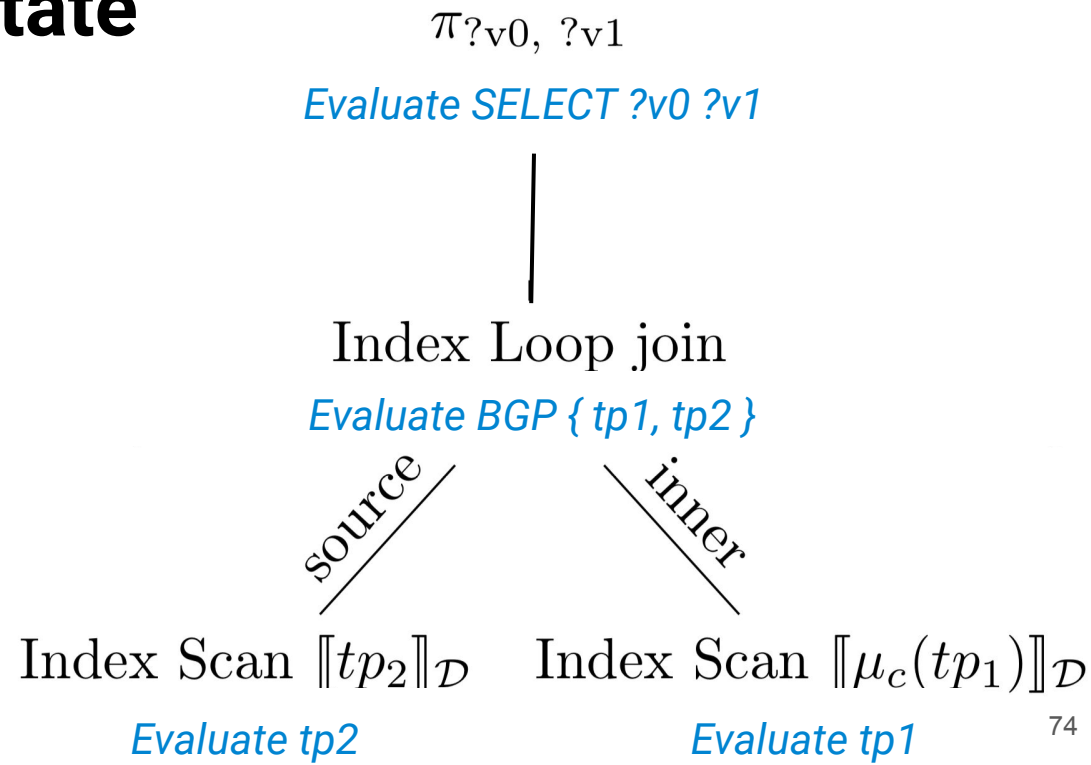
We suspend/resume a running physical query execution plan

```
PREFIX schema: <http://schema.org/contentSize/>
PREFIX gr: <http://purl.org/goodrelations/>
SELECT DISTINCT ?vo ?v1 WHERE {
 ?v0 gr:includes ?v1. # tp1
 ?v1 schema:contentSize ?v3. # tp2
}
```



Suspending a physical query plan

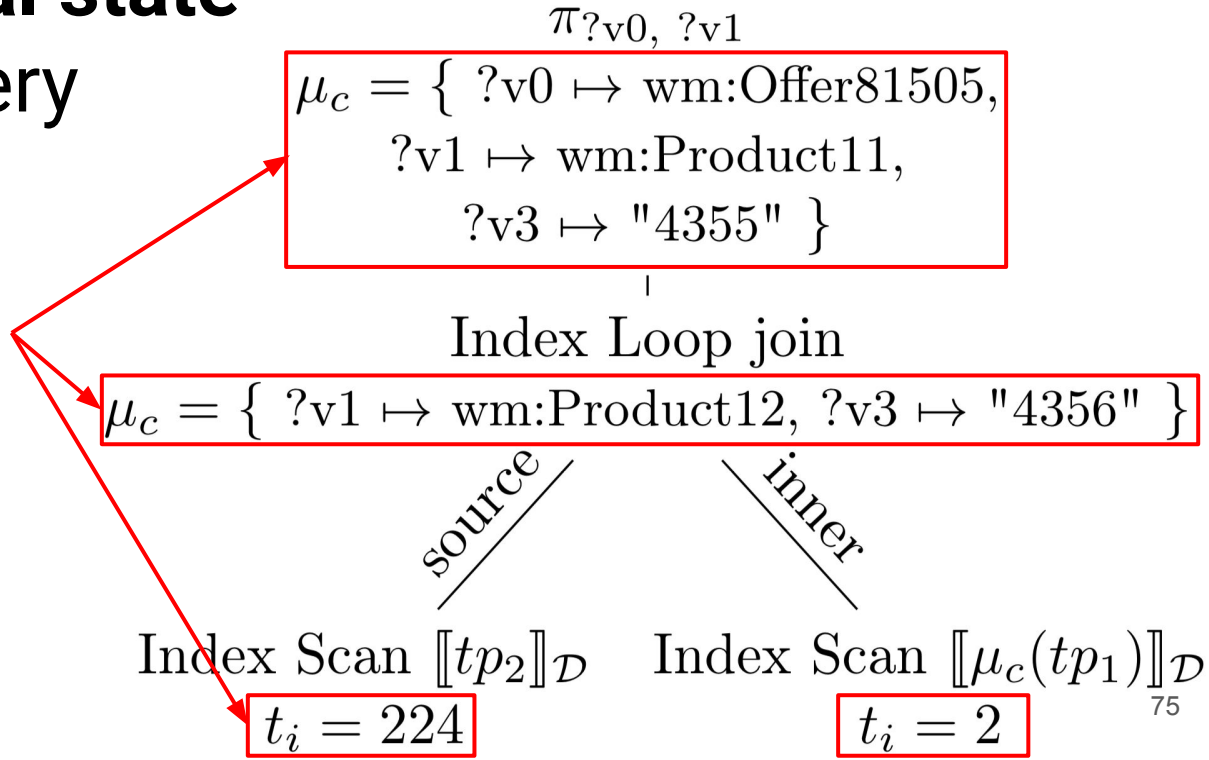
Saving the **internal state** of all physical query operators



Suspending a physical query plan

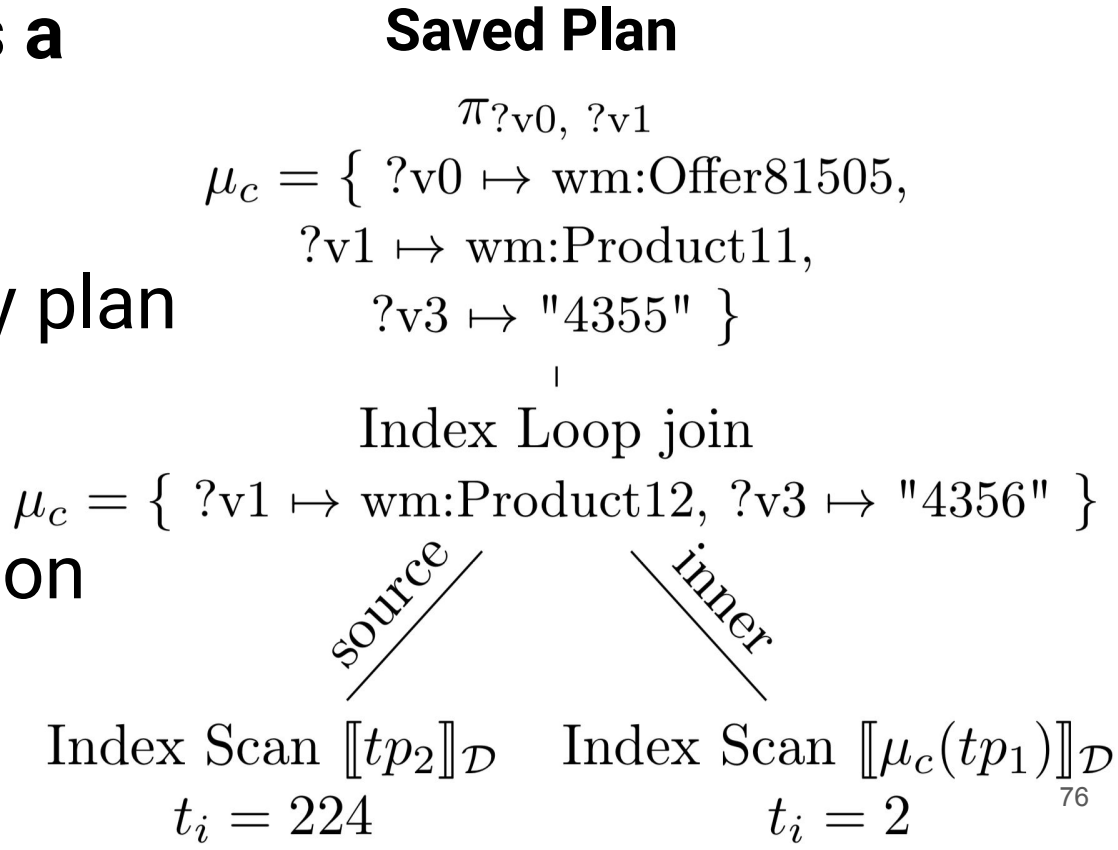
Saving the **internal state** of all physical query operators

Operators internal states



Resuming a physical query plan

1. The server **receives a saved plan**
2. It **rebuilds** the query plan from the saved one
3. It continues execution for a time quantum



SaGe: A preemptive SPARQL query engine

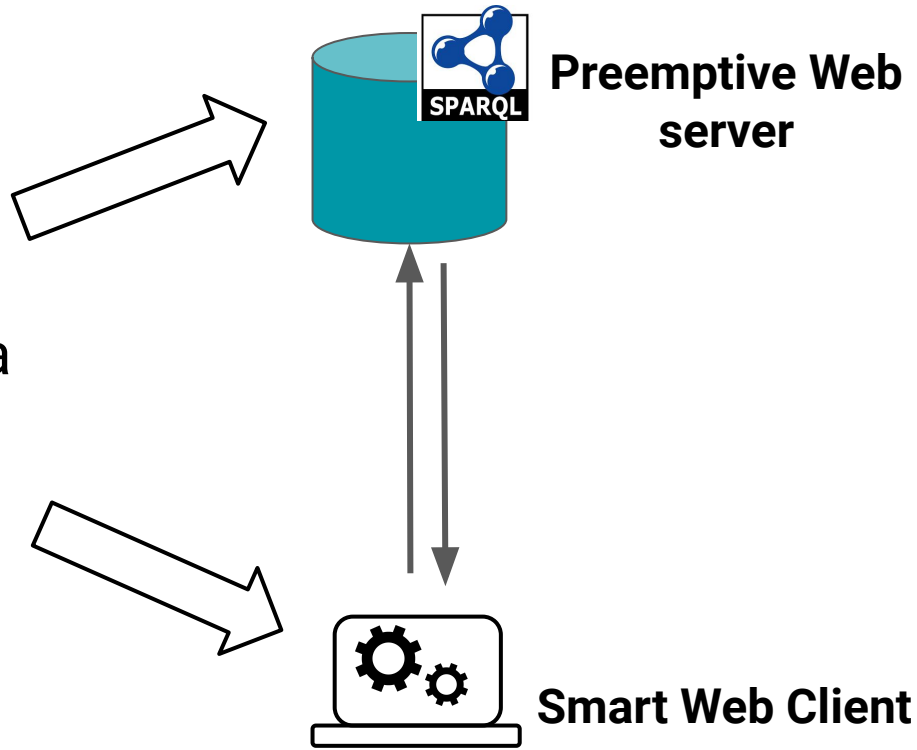
SPARQL Physical Query Operators

- **One mapping-at-a-time:**
  - Operator has one bag of mappings as input
  - Ex: Projection

- **Full-mappings** operators
  - Need *n* bags of mappings as input
  - Ex: Order By

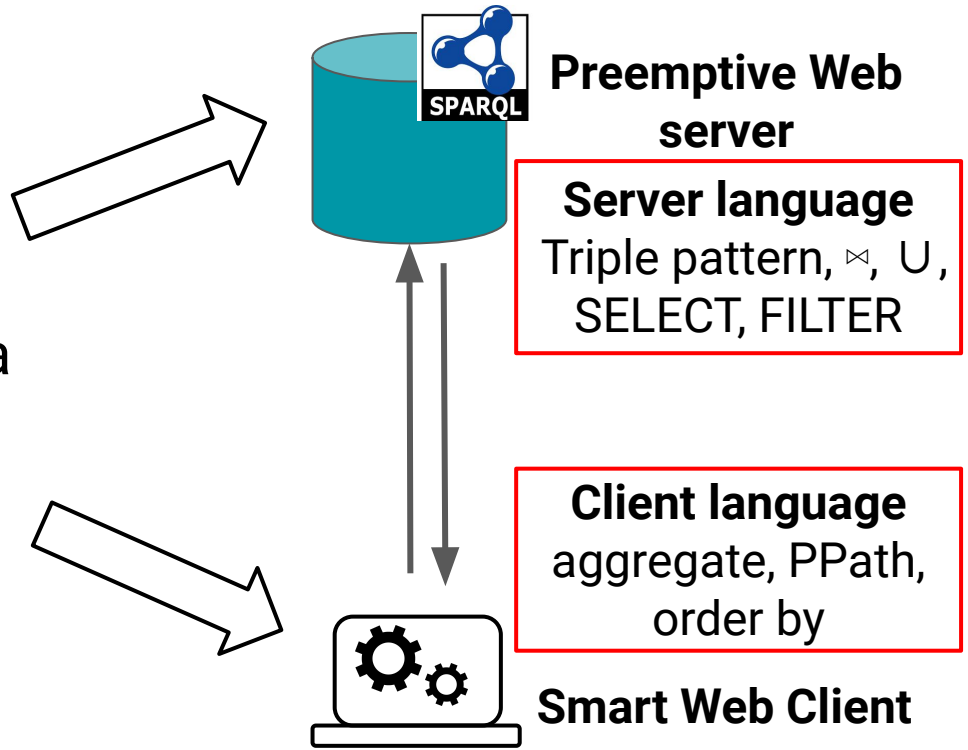
SaGe distributes Physical Query Operators between Server and Client

- **One mapping-at-a-time** operators
  - No need to materialize data
- **Full-mappings** operators
  - Need to materialize data



SaGe distributes Physical Query Operators between Server and Client

- **One mapping-at-a-time** operators
  - No need to materialize data
- **Full-mappings** operators
  - Need to materialize data



SaGe distributes Physical Query Operators between Server and Client

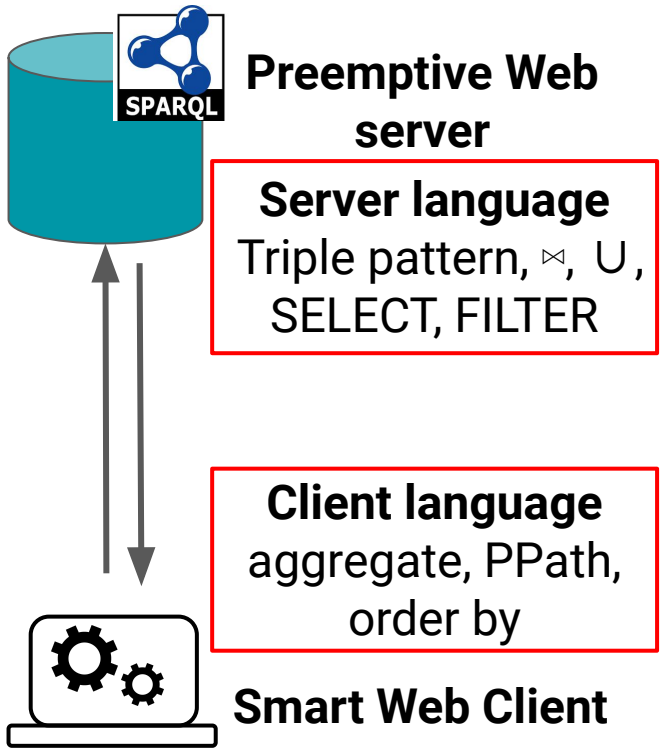
SaGe Preemptive Web **server**

+

SaGe Smart Web **Client**

=

Full SPARQL queries



Complexity of Preemptable Operators

	Suspend	Resume	
Preemptable operator	**Space complexity of local state**	**Time complexity of loading local state**	**Remarks**
$\pi_{v_1,\ldots,v_k}(P)$	$\mathcal{O}(k + \lvert var\ (P)\rvert)$	$\mathcal{O}(1)$	
Index Scan tp	$\mathcal{O}(\lvert tp\rvert + \lvert t\rvert)$	$\mathcal{O}(\log_b(\lvert\mathcal{D}\rvert))$	Require indexes on all kinds of triple patterns
Merge Join $P_1 \bowtie P_2$	$\mathcal{O}(\lvert var\ (P_1)\rvert + \lvert var(P_2)\rvert)$	$\mathcal{O}(1)$	
Index Loop Join $P \bowtie tp$	$\mathcal{O}(\lvert var\ (P)\rvert + \lvert tp\rvert + \lvert id\rvert)$	$\mathcal{O}(\log_b(\lvert\mathcal{D}\rvert))$	
P_1 UNION P_2	$\mathcal{O}(1)$	$\mathcal{O}(1)$	Multi-set Union
P Filter $\mathcal{R}$	$\mathcal{O}(\lvert var\ (P)\rvert + \lvert\mathcal{R}\rvert)$	$\mathcal{O}(1)$	Pure logical expression only
Server physical plan	$\mathcal{O}(\lvert Q\rvert \times \lvert t\rvert)$	$\mathcal{O}(\lvert Q\rvert \times \log_b(\lvert\mathcal{D}\rvert))$	

|Q| : the number of operators in the physical query execution plan
|t| and |tp| : the size of encoding a **RDF triple** and a **triple pattern**, respectively.

Complexity of Preemptable Operators

Suspend Resume

Preemptable operator	Space complexity of local state	Time complexity of loading local state	Remarks
π			
Inc			on all erns
Merge			
Index L			
P_1			
P			pression
Server physical plan	$\mathcal{O}(\|Q\| \times \|t\|)$	$\mathcal{O}(\|Q\| \times \log_b(\|\mathcal{D}\|))$	

- Minimize the complexity in **time** of Suspend and Resume
  - Bounded by **the size of the plan** and **log(|D|)**
- Minimize the complexity in **space** of Suspend and Resume
  - Bounded by **the size of the plan**
- Mainly depends on the size of the plan, which is generally **small**

|Q| : the number of operators in the physical query execution plan
|t| and |tp| : the size of encoding a **RDF triple** and a **triple pattern**, respectively.

Experimental Study

Experimental Study

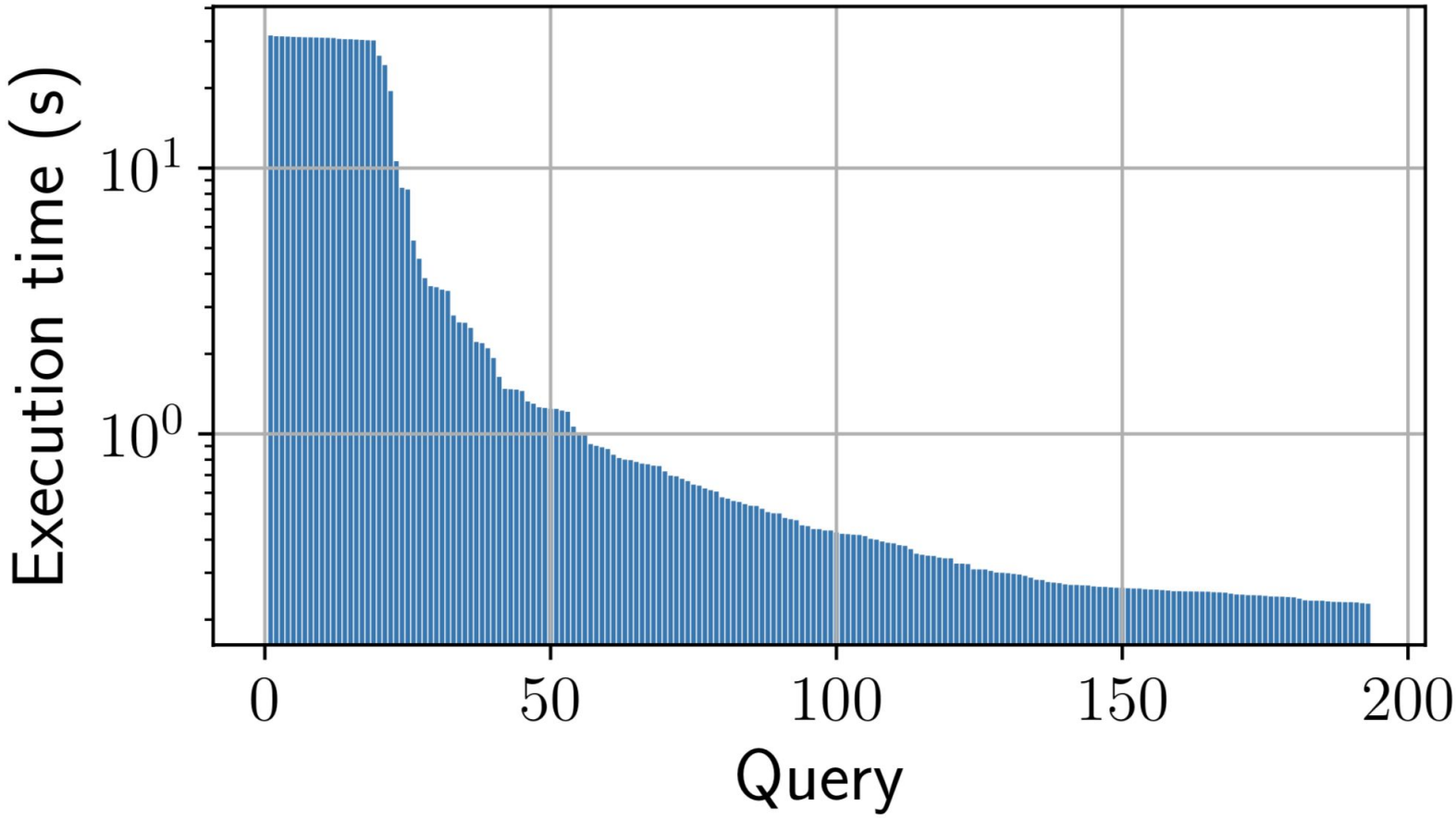
1. What is the **overhead** of Web preemption in time and space?
2. Does Web preemption improve the **average workload completion time**?
3. Does Web preemption improve the **time for first results**?

Experimental Setup

Dataset and Queries

- Waterloo SPARQL Diversity Test suite [1]
  - 10^7 triples, stored using HDT [2]
- Generate 50 workloads of 193 queries
  - 20% of queries require more than 30s to execute
- Same as BrTPF experiments [3]

[1] G. Aluç et al., *"Diversified stress testing of RDF data management systems"*, ISWC 2014
[2] J. D. Fernández et al., *"RDF representation for publication and exchange (HDT)"*, Journal of Web Semantic (2013)
[3] O. Hartig et al. *"Bindings-restricted triple pattern fragments"*, in ODBASE 2016



Distribution of query execution time for each workload

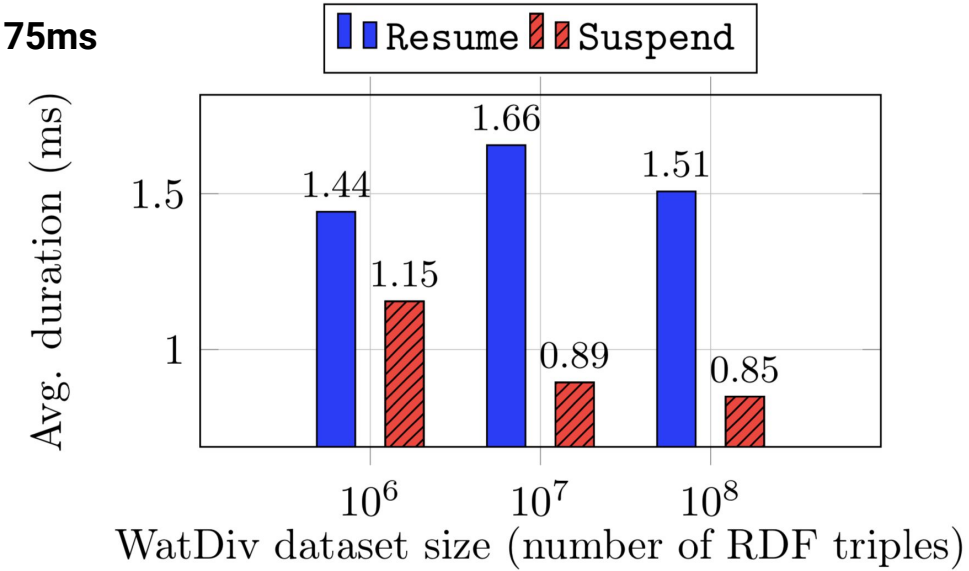
Compared approaches

- **SaGe**
  - Quantums of 75ms (**SaGe-75ms)** & 1s (**SaGe-1s)**
- **Virtuoso** [1] for SPARQL endpoints
  - **Without quotas**
- Triple Pattern Fragments [2] (**TPF**)
- Bindings-restricted Triple Pattern Fragments [3] **(BrTPF)**

[2] R. Verborgh et al. "*Triple pattern fragments: A low-cost knowledge graph interface for the web*", Journal of Web Semantics (2016)
[3] O. Hartig et al. *"Bindings-restricted triple pattern fragments"*, in ODBASE 2016
[1] O.Erling et al. *"RDF support in the Virtuoso DBMS"*, In Networked Knowledge - Networked Media, 2009

What is the overhead in time of Web preemption?

Time quantum: 75ms



Average preemption overhead

The size of the dataset does not impact the overhead, around **1ms for Suspend** and **1,5ms for Resume** (3% of time quantum)

What is the overhead in space of Web preemption?

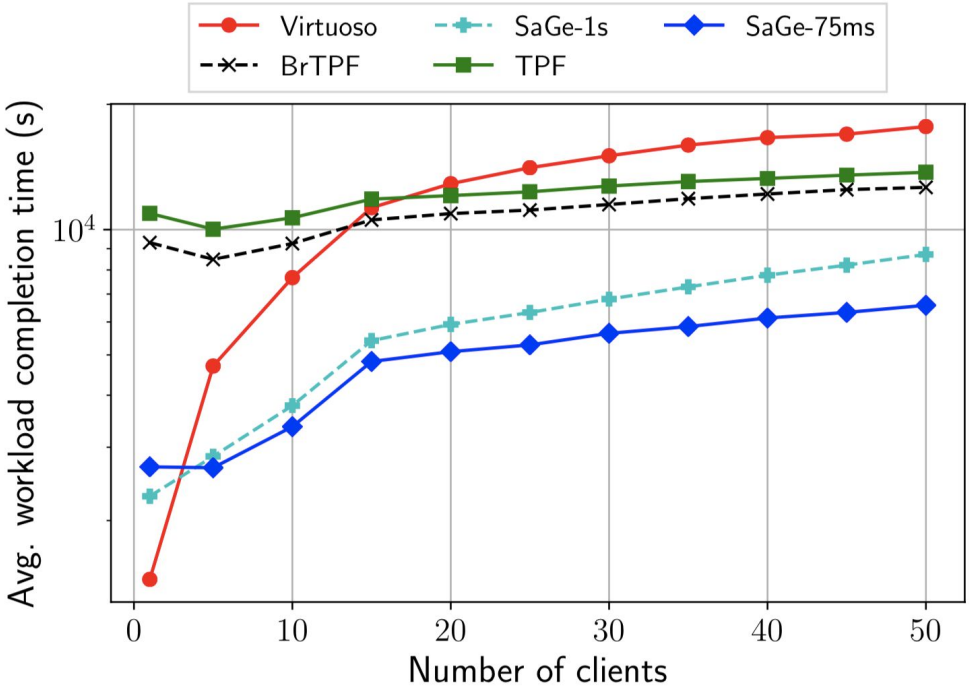
Sizes of saved physical query execution plans

Mean	Min	Max	Standard deviation
1.716 kb	0.276 kb	6.212 kb	1.337 kb

- Space is proportional to the number of operators in the plan
- The size of a saved plan remains small, no more **6.2Kb** for a query with **ten joins**

Does Web preemption improve the average workload completion time?

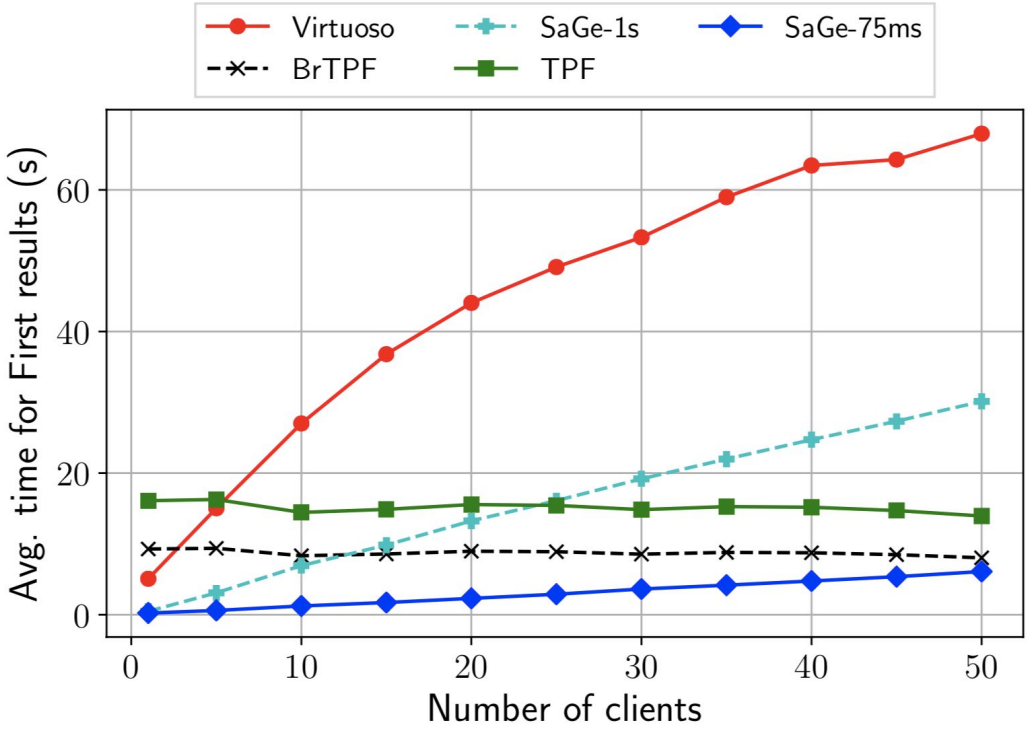
- Virtuoso is impacted by convoy effect
- BrTPF & TPF avoid convoy effect, but they are **slow**
- SaGe-75ms avoids convoy effect and performs better than others



Average workload completion time per client, with up to 50 concurrent clients (logarithmic scale, one worker)

Does Web preemption improve the time for first results (*TFR*)?

- Virtuoso is impacted by convoy effect
- *TFR* for BrTPF & TPF increase slowly
- *TFR* for SaGe is **proportional** to the time quantum



Average time for first results (over all queries), with up to 50 concurrent clients (linear scale)

Which Operators are Preemptable?

Preemptable	??	Not preemptable
• Triple pattern, • Projection, • Join, • Union, • Bind, Group, • Most Filters	• Optional • Filter not exist • Minus	• **Aggregation:** requires to store group keys • **Order-by:** requires all results before sorting • **Nested queries:** storing results of inner query • **Property paths:** need to remember visited nodes...

If we consider only these operators: A preemptive SPARQL endpoint behaves as a SPARQL endpoint.

Processing SPARQL Aggregate Queries with Web Preemption

- Compute **partial Aggregates** per quantum
- Client merge partial aggregate
- Correct because aggregation functions are decomposable

Table 1: Decomposition of SPARQL aggregation functions

SPARQL Aggregations functions								
	COUNT	SUM	MIN	MAX	AVG	COUNT_D	SUM_D	AVG_D
f_1	COUNT	SUM	MIN	MAX	SaC	CT		
$v \diamond v'$	$v + v'$		$min(v, v')$	$max(v, v')$	$v \oplus v'$	$v \cup v'$		
h	Id				$(x, y) \mapsto x/y$	COUNT	SUM	AVG

SaGe-Agg: Processing SPARQL Aggregate Queries with Web Preemption. A Grall, T Minier, H Skaf-Molli, P Molli. ESWC 2020

Retrieve creative works and the list of fictional works that inspired them

Wikidata Query Service

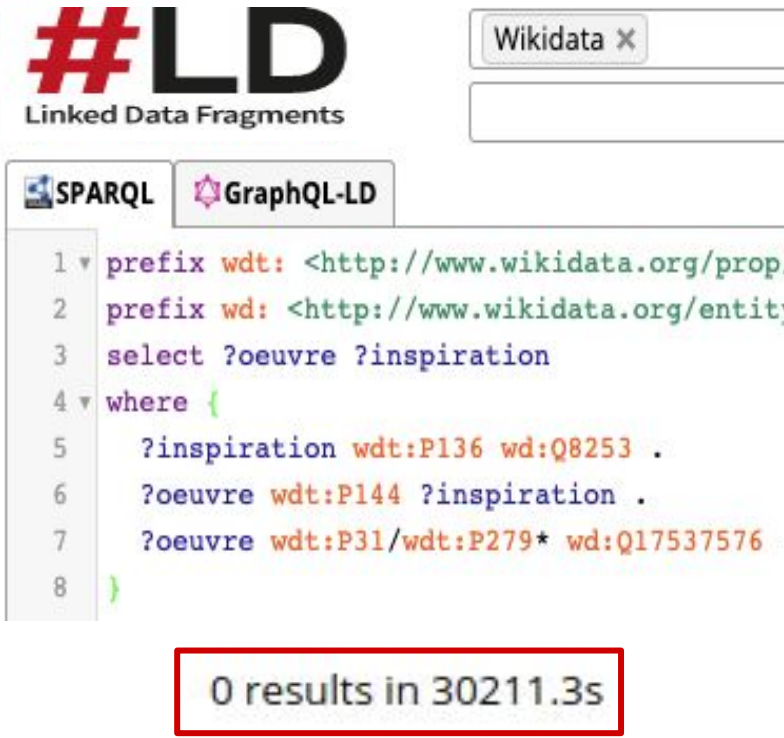
Exemples

```
1 SELECT ?oeuvre ?inspiration
2 WHERE {
3   ?inspiration wdt:P136 wd:Q8253 .
4   ?oeuvre wdt:P144 ?inspiration .
5   ?oeuvre wdt:P31/wdt:P279* wd:Q17537576 .
6 }
```

Killed after 60 s!

TPF with restricted web servers terminates but

- After **8 hours**, the query is still running
- Why ?
  - No support for BGP
  - No support for transitive closures on server-side
- **Too much calls and data transfer. Not realistic !**



With the preemptive server SaGe in ~9 sec !

Query editor

```
1 PREFIX wdt: <http://www.wikidata.org/prop/direct/>
2 PREFIX wd: <http://www.wikidata.org/entity/>
3 SELECT ?artWork ?otherWork WHERE {
4   ?artWork wdt:P144 ?otherWork .
5   ?artWork wdt:P31 ?v . ?v wdt:P279* wd:Q17537576 .
6   ?otherWork wdt:P136 wd:Q8253 .
7 }
```

Execution time	HTTP requests	Data transfer	Numbe
9374 ms	97 requests	66677 bytes	62 solution mappings

REMEMBER

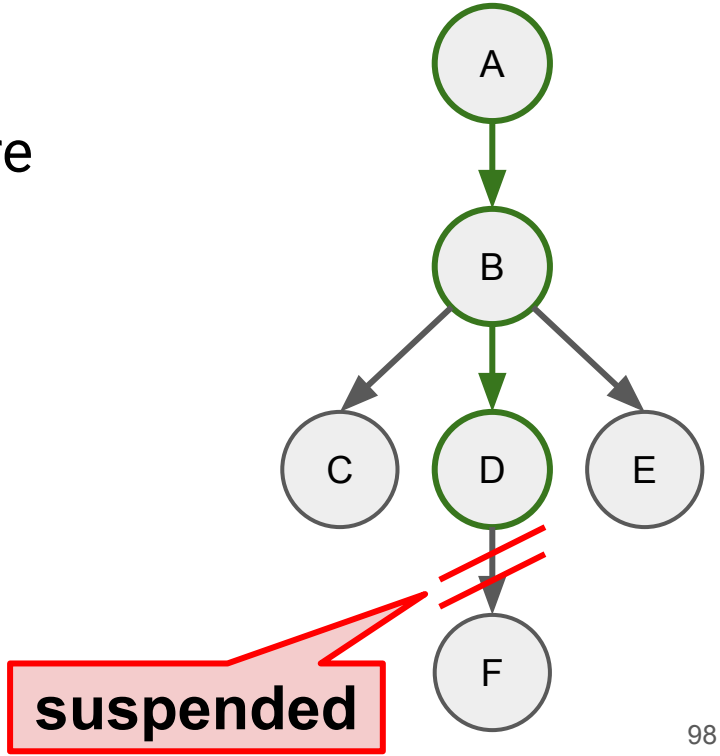
DBpedia: Killed
Wikidata: Killed

TPF:
I stopped query after 8 hours, no results

Julien AIMONIER-DAVAT, Hala-Skaf Molli, Pascal Molli.SaGe-Path: Pay-as-you-go SPARQL Property Path Queries Processing using Web Preemption. Demo Extended Semantic Web Conference, ESWC 2021.

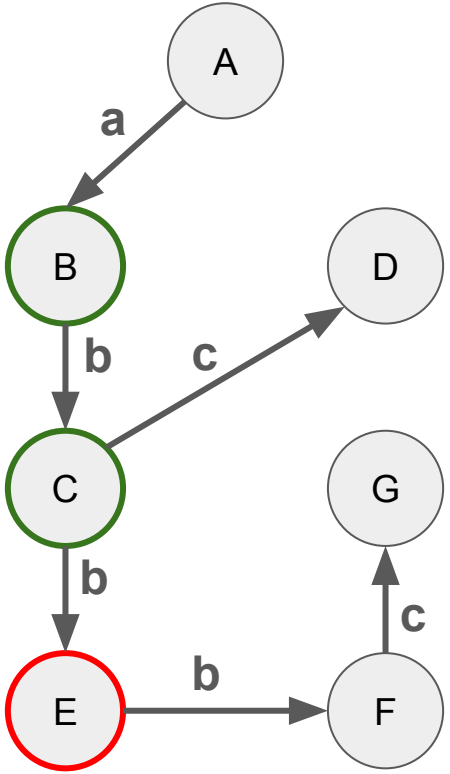
Transitive closure (*) is not preemptive

- To suspend/resume a transitive closure
  - Need to keep at least the current path ~ O(graph)
- O(suspend/resume) ~ O(graph)



Key idea: Server & Client collaborate

- Partial Transitive Closure (PTC) on the server
  - Exploration depth is limited by **k**
  - O(suspend/resume) = O(k)
- PTC is **preemptable** but **not complete**
  - Return **frontier nodes** = nodes visited at distance k
- The client **expands frontier nodes** by generating new queries
  - Ensure **complete results**



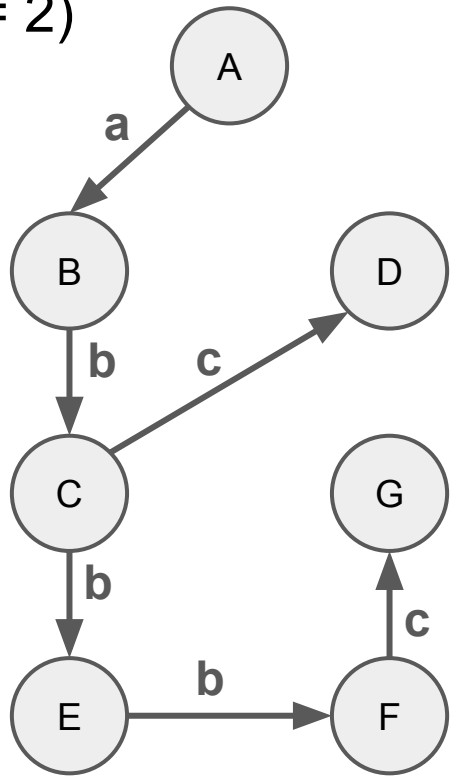
PTC(B, b, 2)

Julien AIMONIER-DAVAT, Hala-Skaf Molli, Pascal Molli. Processing SPARQL Property Path Queries Online with Web Preemption. Extended Semantic Web Conference, ESWC 2021.

SELECT * WHERE { A a ?x . ?x b+ ?y . ?y c ?o }

Client

Server (k = 2)

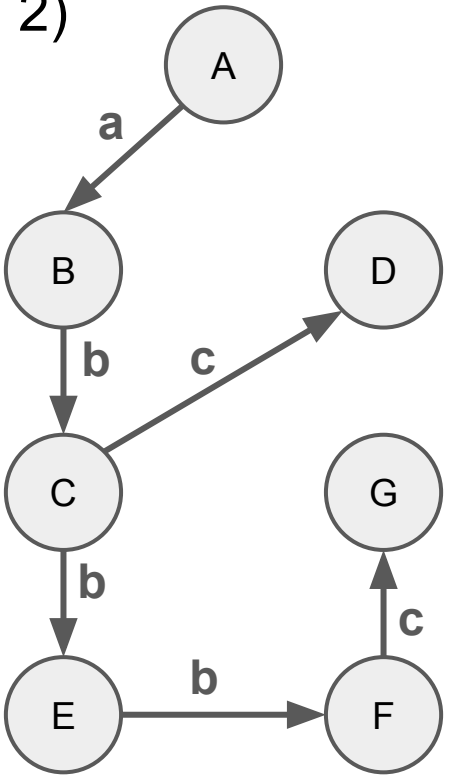


PTC in action

Client

Server (k = 2)

SELECT ?x ?y ?o
WHERE { A a ?x . ?x b+ ?y . ?y c ?o **}**

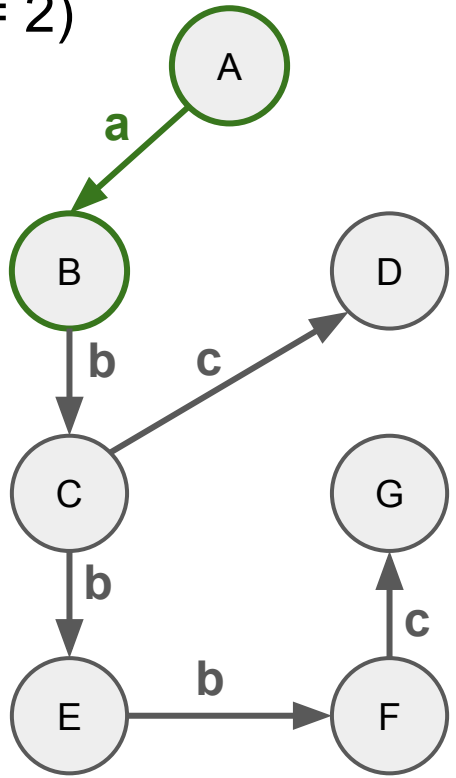


PTC in action

Client

Server (k = 2)

SELECT ?x ?y ?o
WHERE { A a ?x . ?x b+ ?y . ?y c ?o **}**

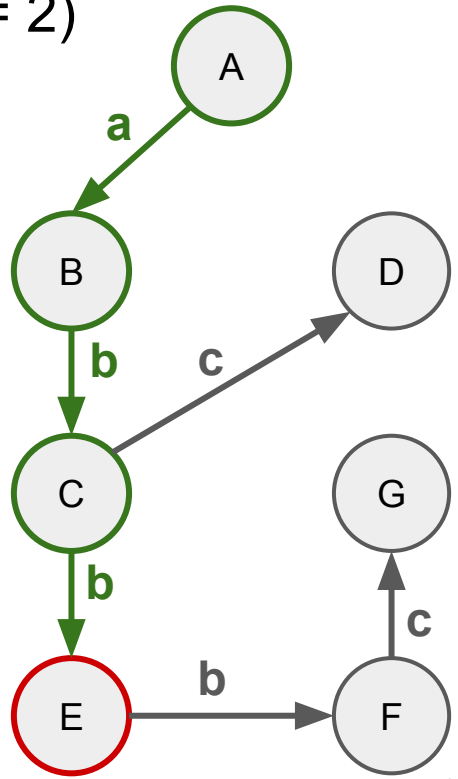


PTC in action

Client

Server (k = 2)

SELECT ?x ?y ?o
WHERE { A a ?x . ?x b+ ?y . ?y c ?o **}**

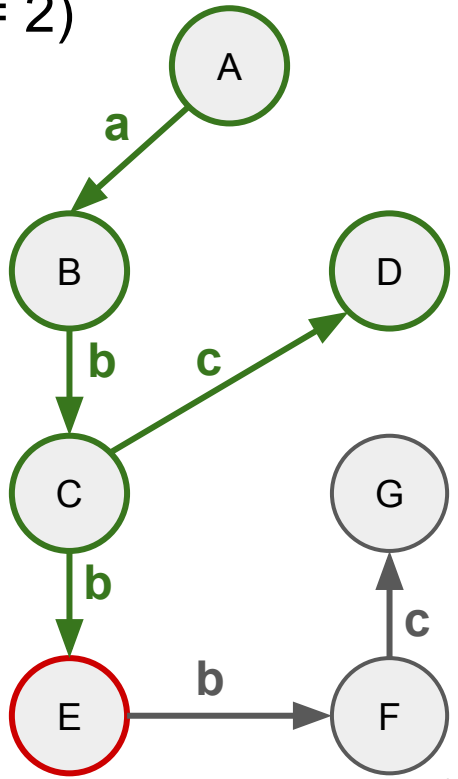


PTC in action

Client

Server (k = 2)

```
SELECT ?x ?y ?o
WHERE { A a ?x . ?x b+ ?y . ?y c ?o }
```



PTC in action

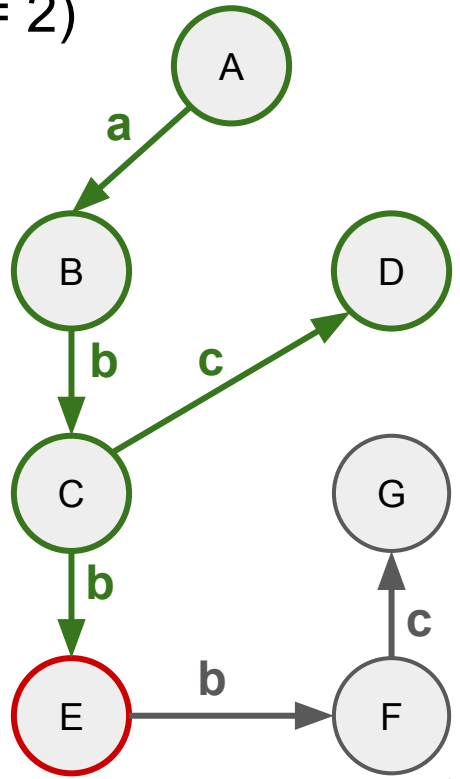
Client

Server (k = 2)

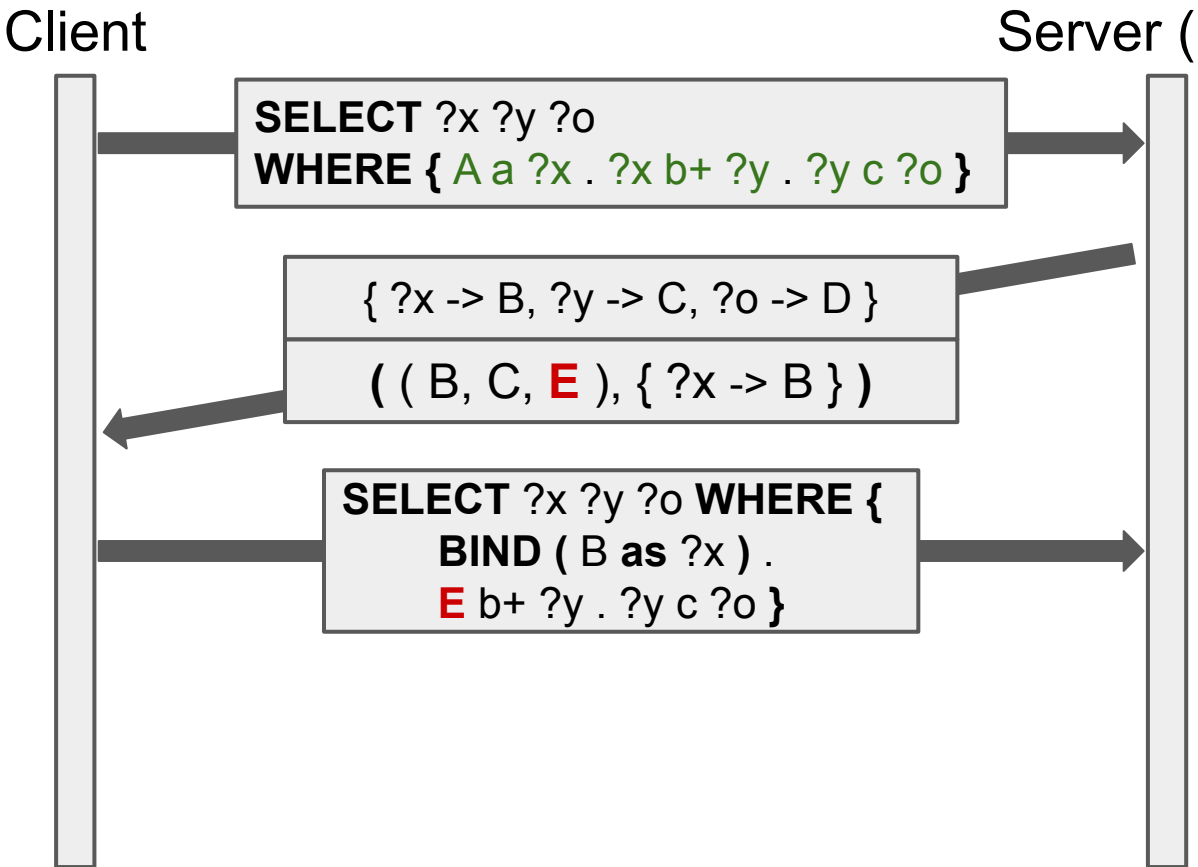
SELECT ?x ?y ?o
WHERE { A a ?x . ?x b+ ?y . ?y c ?o **}**

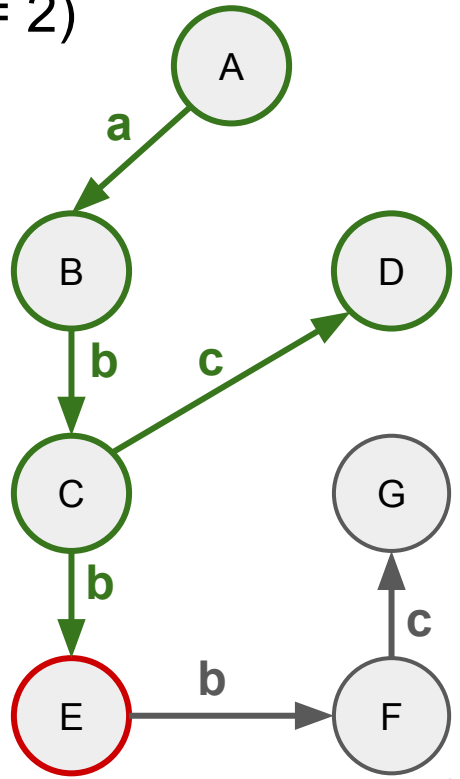
{ ?x -> B, ?y -> C, ?o -> D }

((B, C, **E**), { ?x -> B } **)**

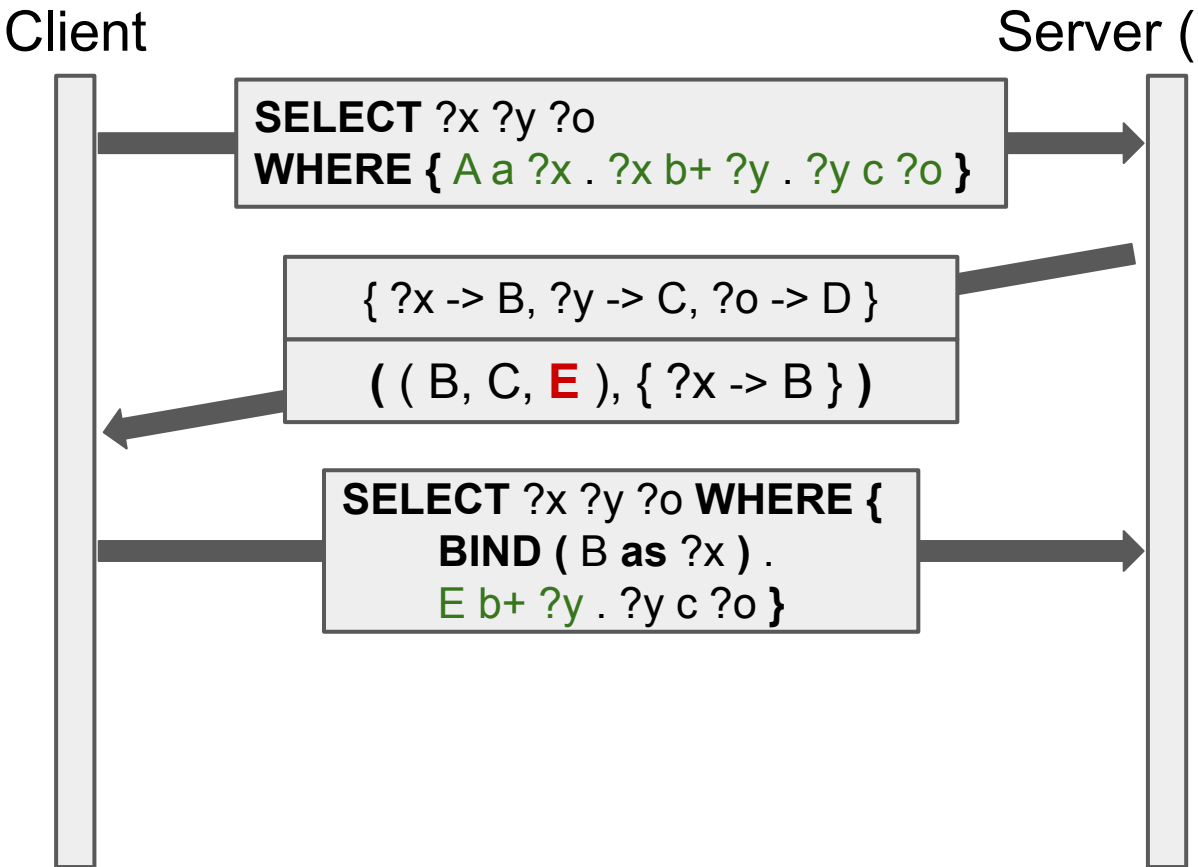


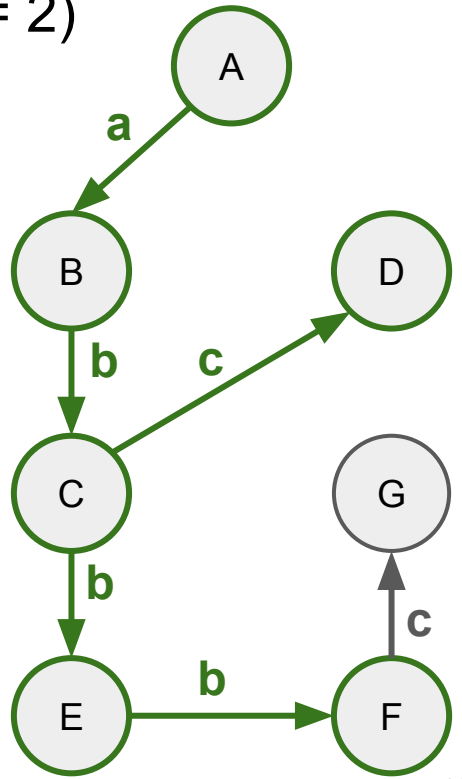
PTC in action



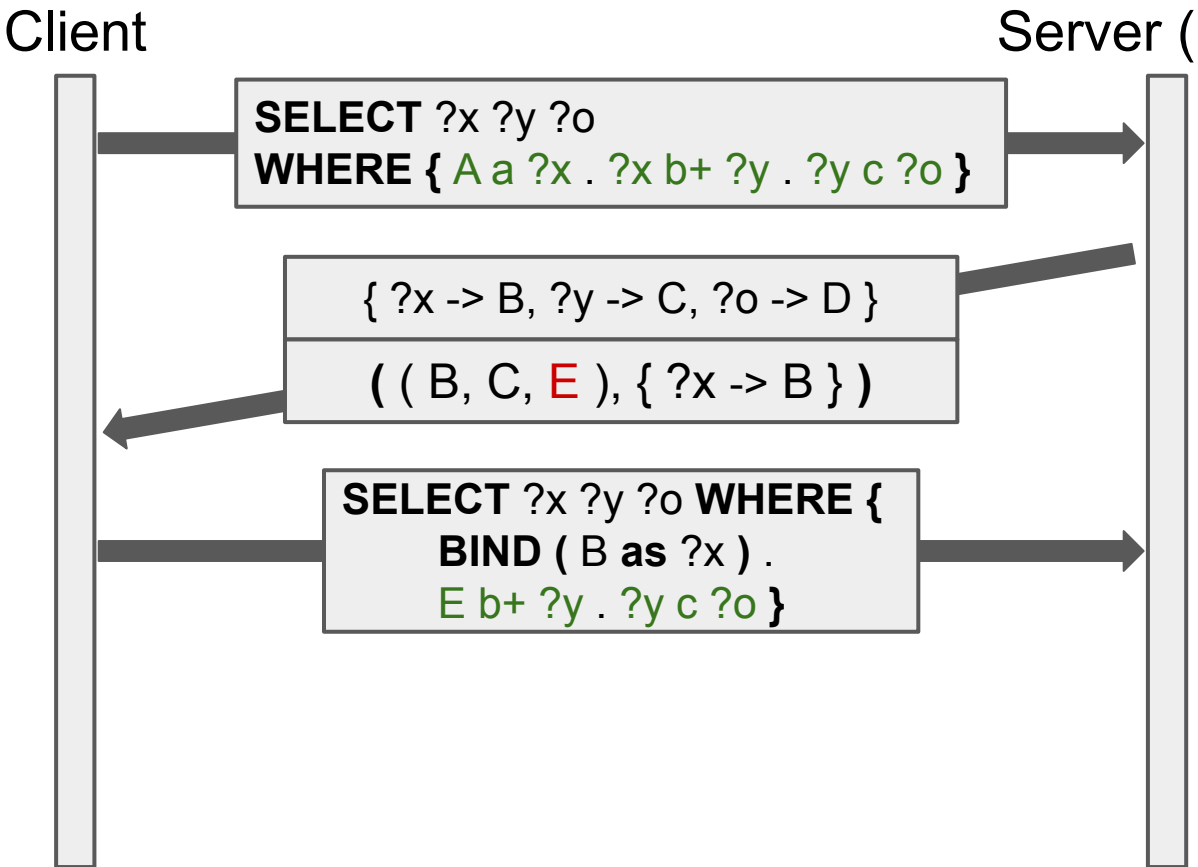


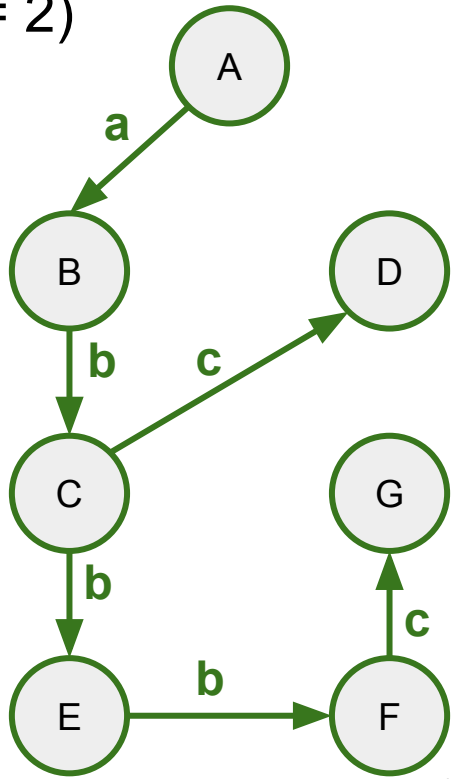
PTC in action



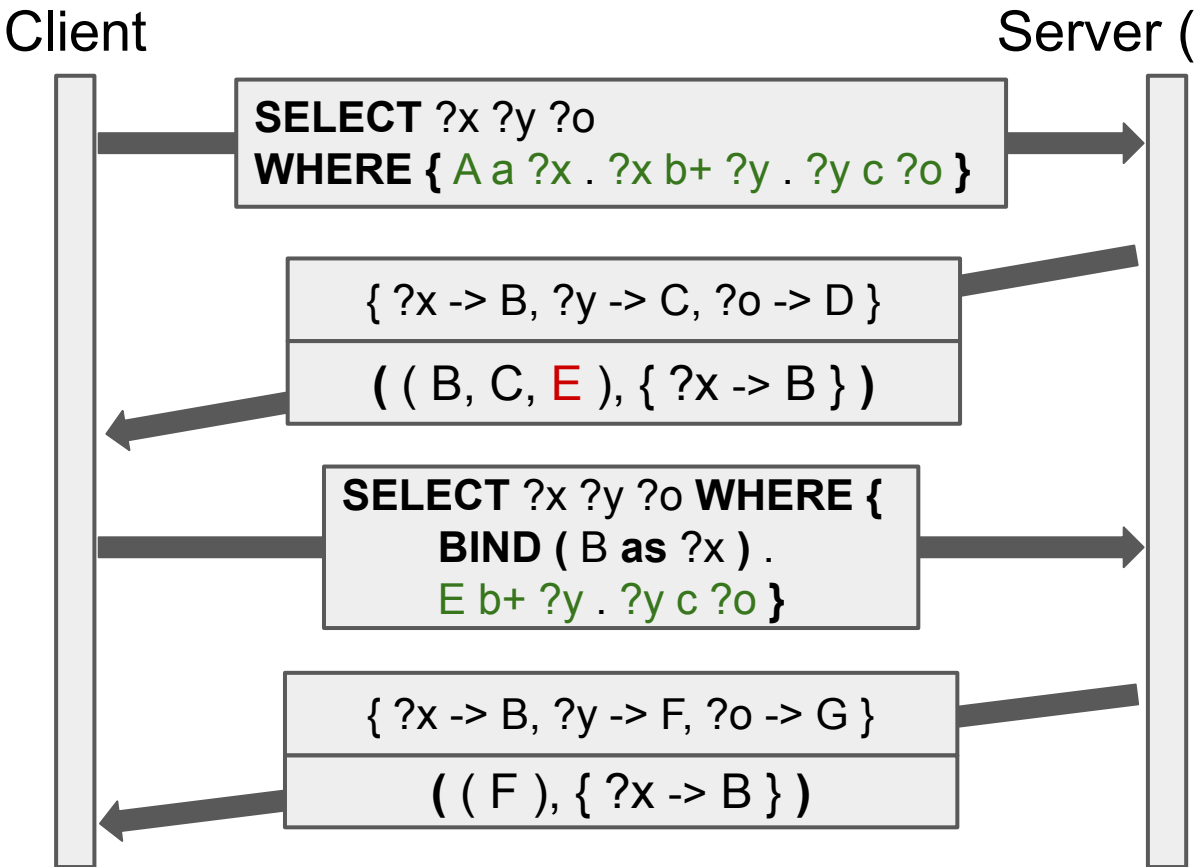


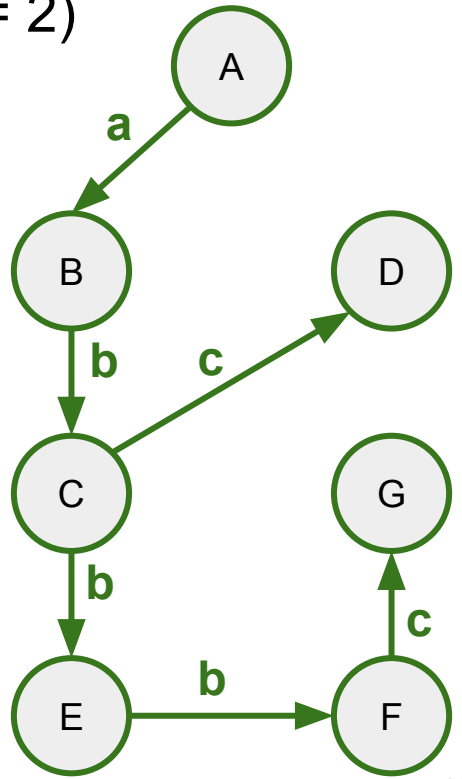
PTC in action



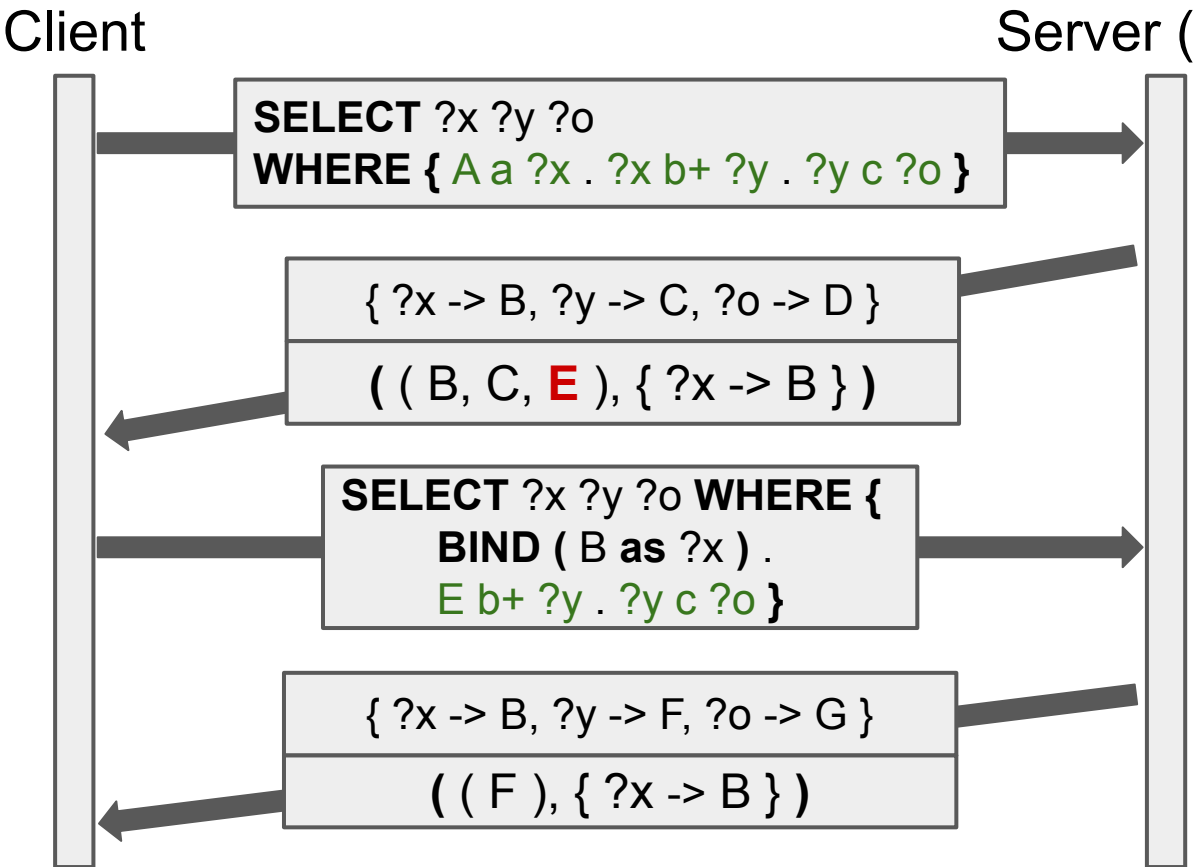


PTC in action





PTC in action

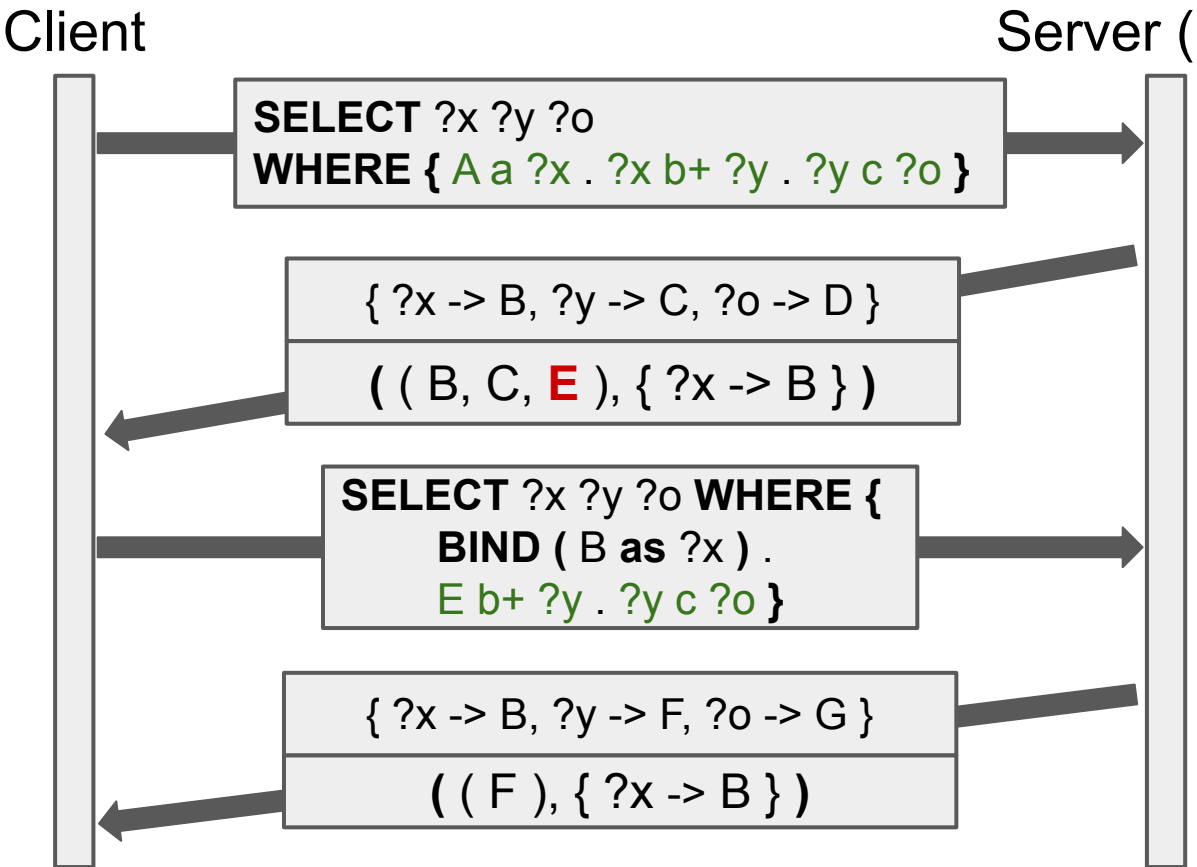


All joins are performed on the server !

- No intermediate results transferred to the client

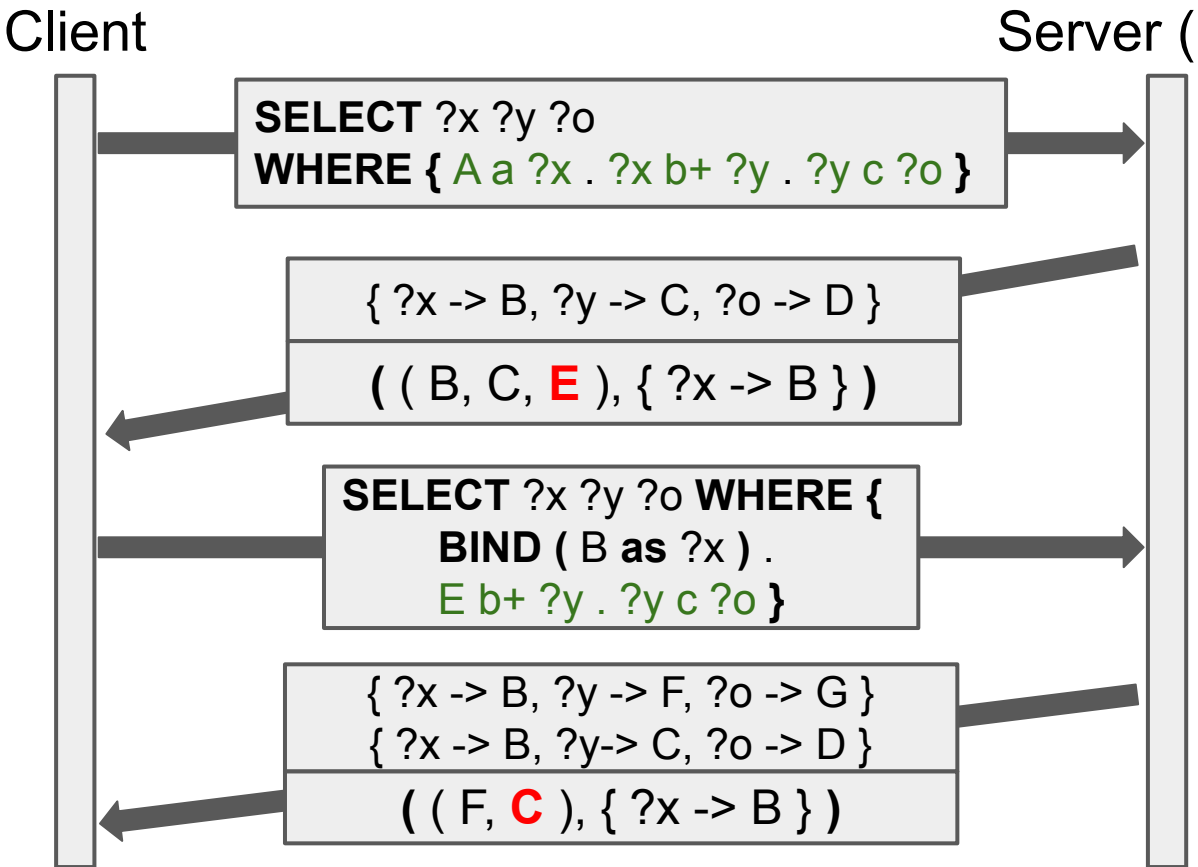
Only visited nodes and final results are transferred

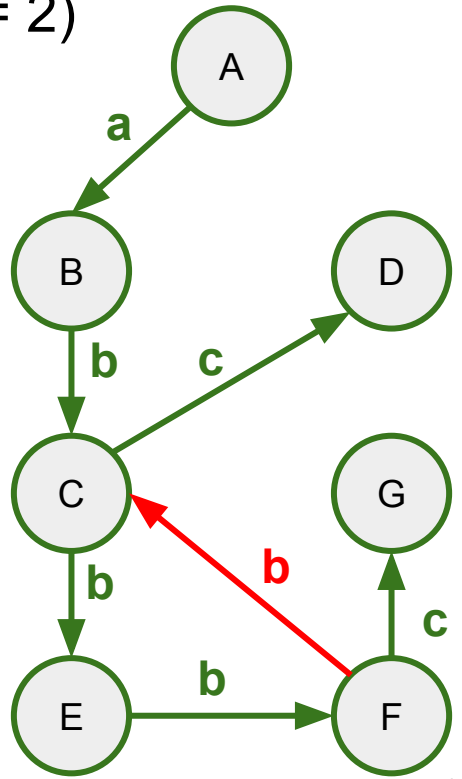
PTC in action



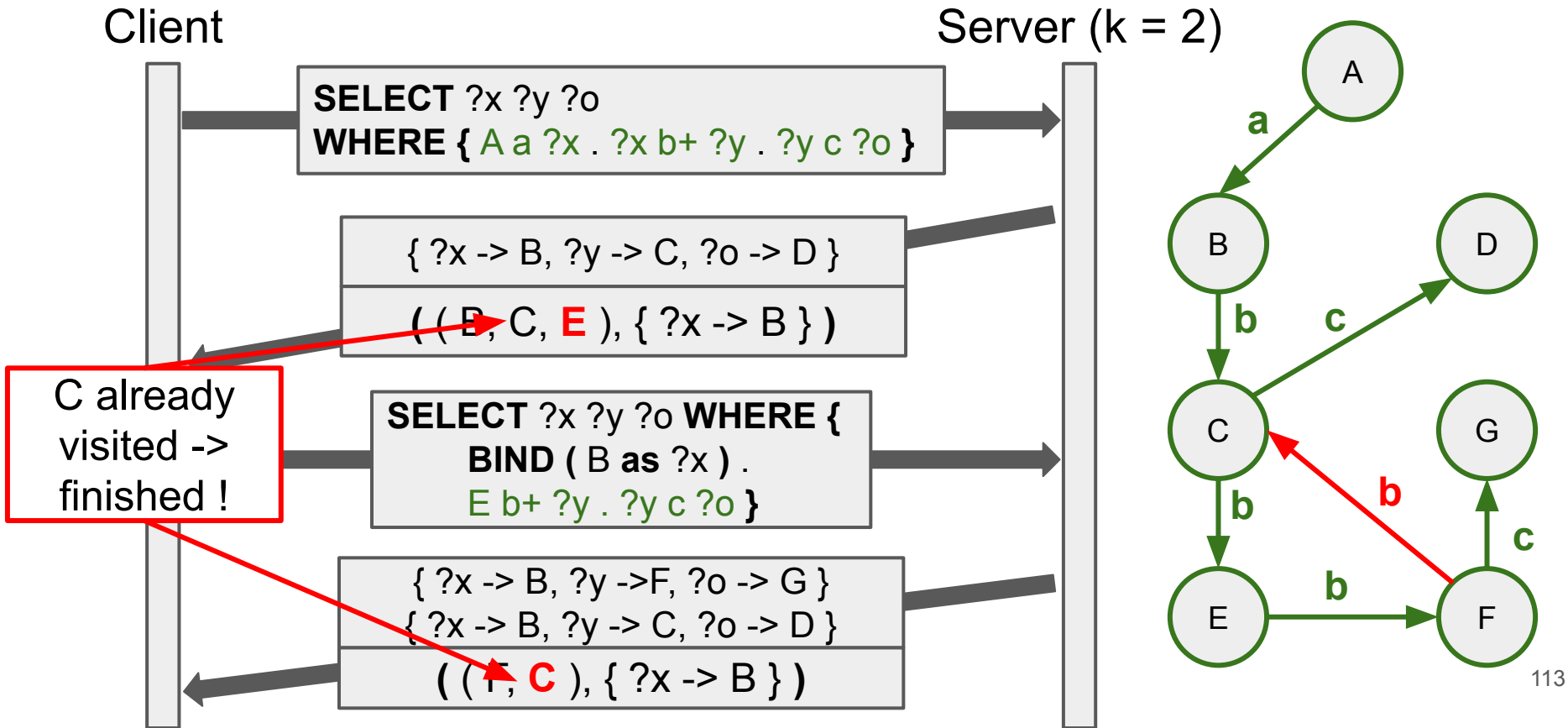
All results are returned ranked by depth
First answers quickly

What happens with cycles ?

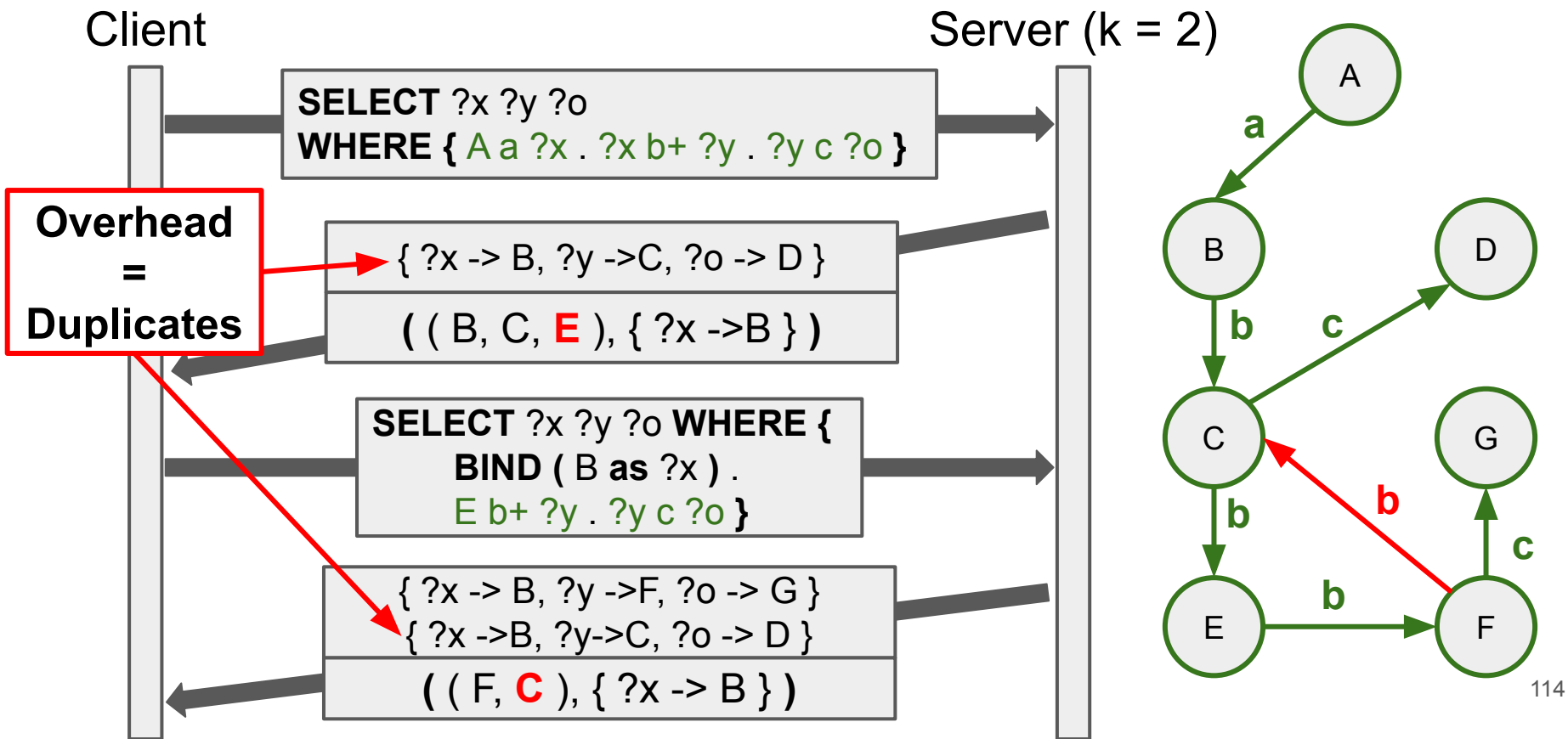




What happen with cycles ?



What happen with cycles ?



Experimental study

Experimental study

1. What is the performance of SaGe-PTC compared to the baseline SaGe?
2. **What is the performance of SaGe-PTC compared to optimal solutions as Fuseki/Virtuoso, i.e. What is the price to pay for complete results with no quota?**
3. What is the impact of the PTC limit **k** on data transfer, execution time and number of calls ?
4. What is the impact of the quantum on data transfer, execution time and number of calls ?

Experimental setup

- Dataset:
  - Synthetic gMark Benchmark [1] of 10M triples with cycles.
- Queries:
  - 30 queries with maxPath of 20, timeout of 30m
  - 1 to 3 transitive closures within a BGP

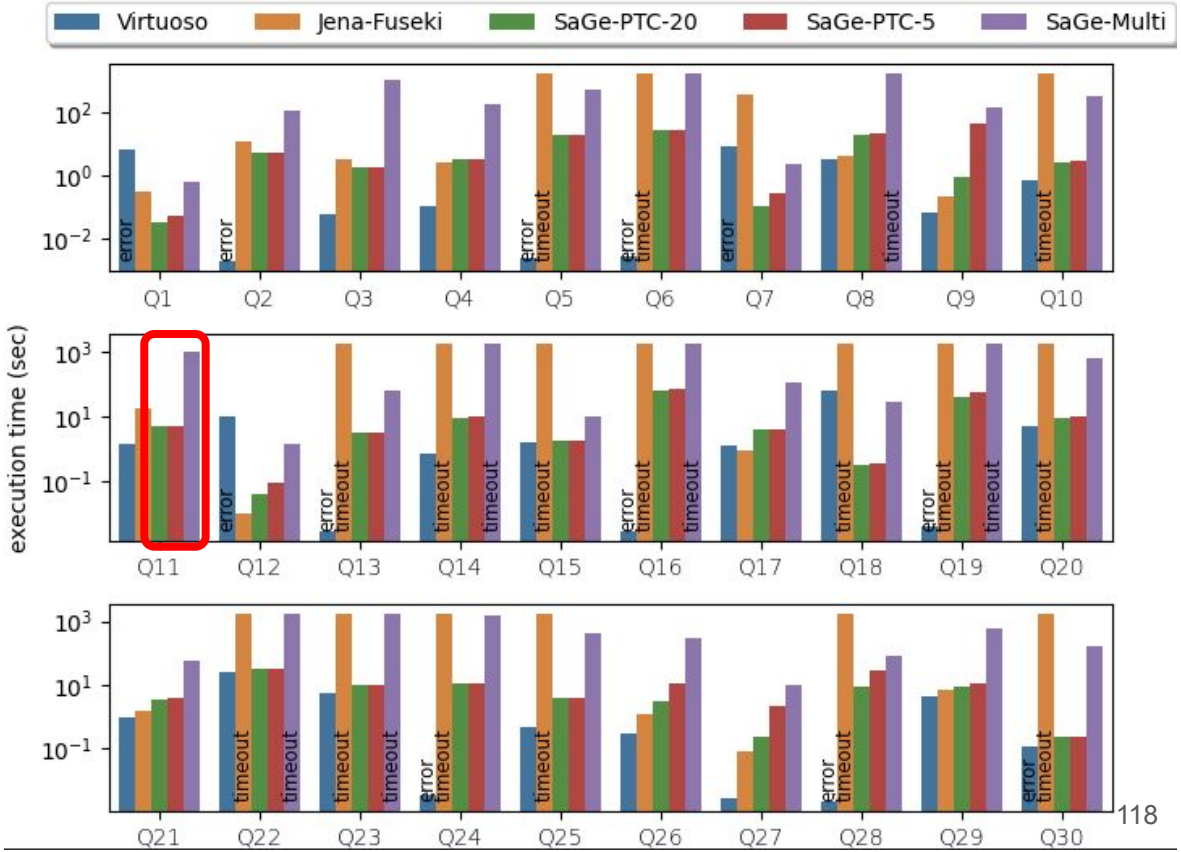
```
PREFIX : <http://example.org/gmark/>
SELECT ?x0
WHERE {
  ?x0 (:peditor) ?v0 . ?v0 (:plike) ?v1 . ?v1 (:phasReview) ?x1 .
  ?x1 (^:phasReview) ?v2 . ?v2 (:partist) ?x2 .
  ?x2 (:pfriendOf)+ ?x3 .
}
```

```
PREFIX : <http://example.org/gmark/>
SELECT ?x0 ?x4
WHERE {
  ?x0 (^:plocation) ?v0 . ?v0 (^:peditor) ?x1 .
  ?x1 ((:pauthor/^:pauthor))+ ?x2 .
  ?x2 ((:phomepage/^:phomepage)|(^:pincludes/:pincludes))+ ?x3 .
  ?x3 (:peditor) ?x4 .
}
```

[1] Bagan, G., Bonifati, A., Ciucanu, R., Fletcher, G. H., Lemay, A., & Advokaat, N. (2016). gMark: Schema-driven generation of graphs and queries. *IEEE Transactions on Knowledge and Data Engineering*, *29*(4), 856-869.

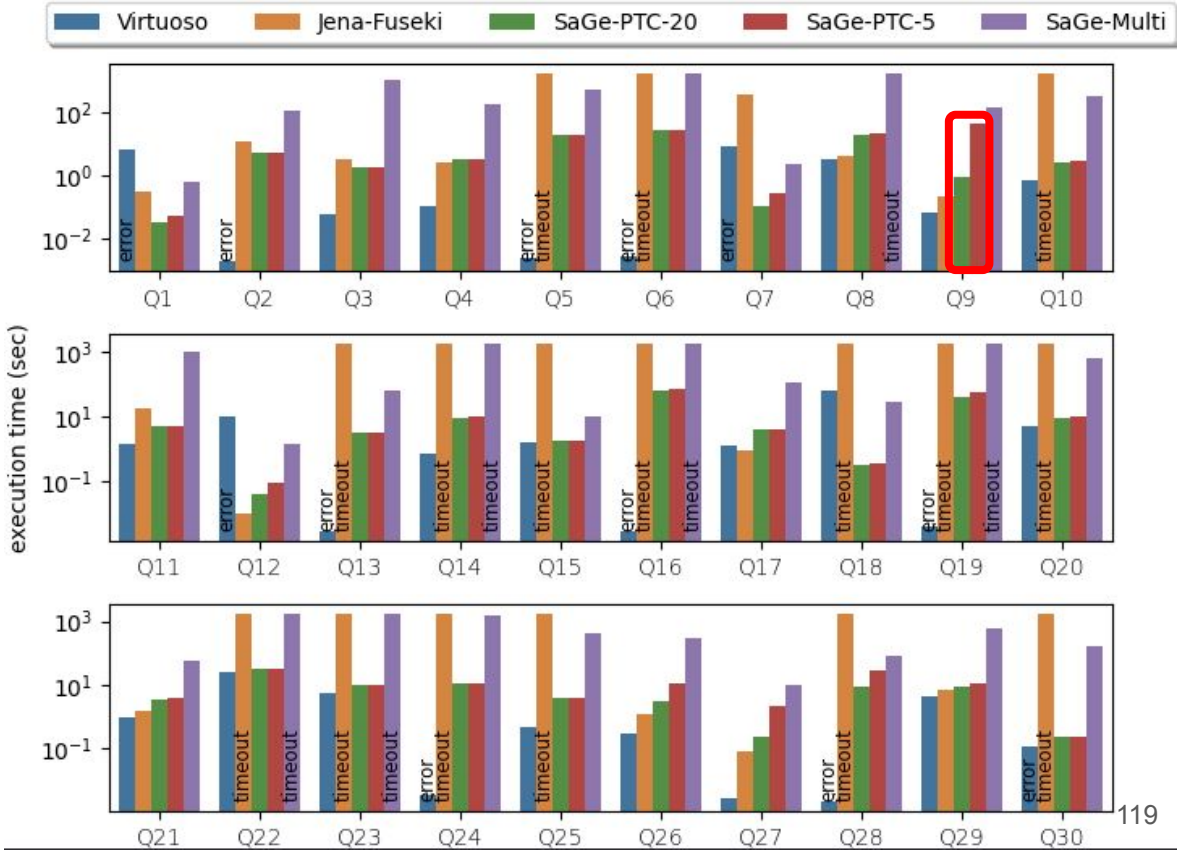
Execution time with a Quantum of 60s

- SaGe-PTC outperforms the baseline SaGe
- Not transferring intermediate results is really effective !



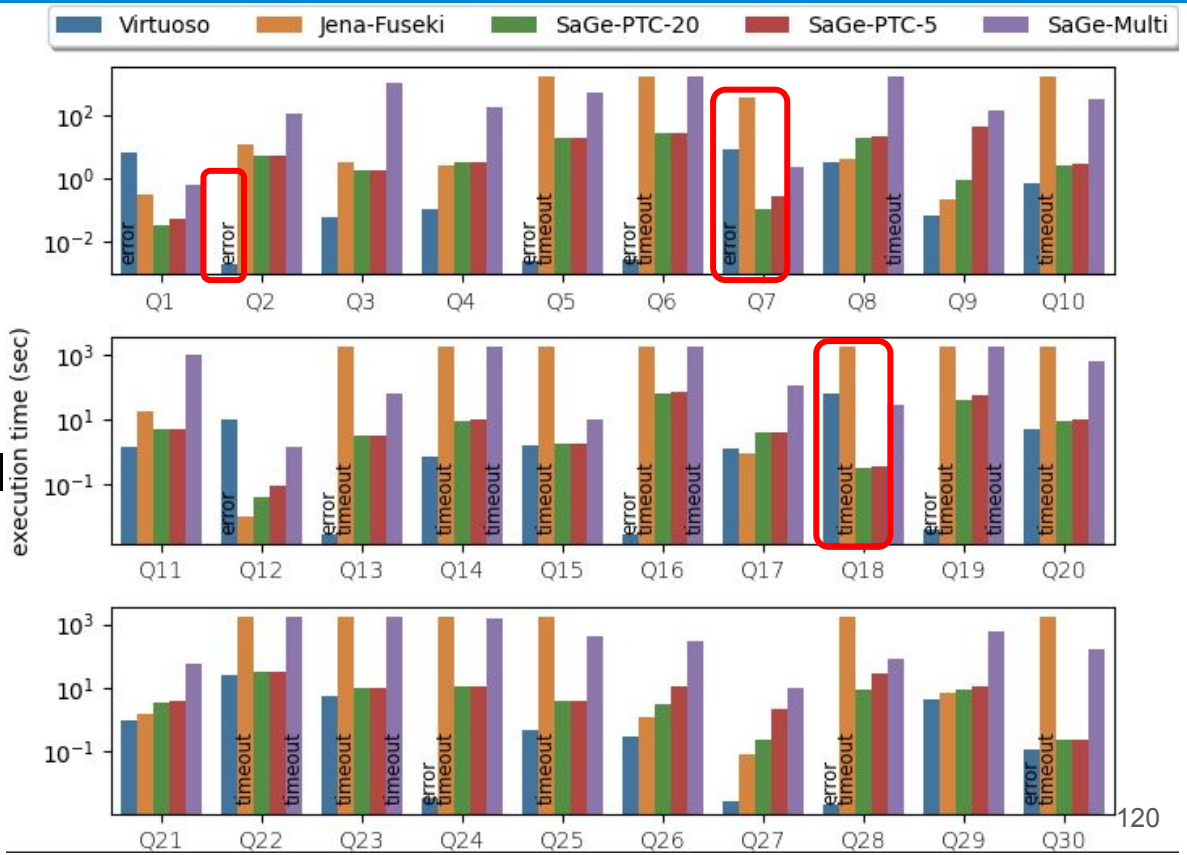
Execution time with a Quantum of 60s

- SaGe-PTC20 always better than SaGe-PTC5
- As expected, k=20 is the best case for SaGe-PTC.



Execution time with a Quantum of 60s

- Compared to Virtuoso SaGe-PTC better supports standard
  - Virtuoso rejects 12 of the 30 queries
- Surprisingly, SaGe-PTC can be better than optimal solutions
  - Ex: Q7, Q1, Q18
- SaGe is competitive vs Fuseki and always terminate



Which Operators are Preemptable?

Preemptable	??	Not preemptable
• Triple pattern, • Projection, • Join, • Union, • Bind, Group, • Most Filters • ✅ Partial aggregation • ✅ Partial property paths	• Optional • Filter not exist • Minus	• **Aggregation:** requires to store group keys • **Order-by:** requires all results before sorting • **Nested queries:** storing results of inner query • **Property paths:** need to remember visited nodes...

Takeaways

- Knowledge graphs are everywhere, they cover all domains
- Public SPARQL servers allow online querying of knowledge graphs
  - Quotas ensure fair usage policies but prevent to build applications
- Restricted server interfaces (TPF) ensures complete results
  - Suffer of large number of HTTP calls and large data transfer

Takeaways

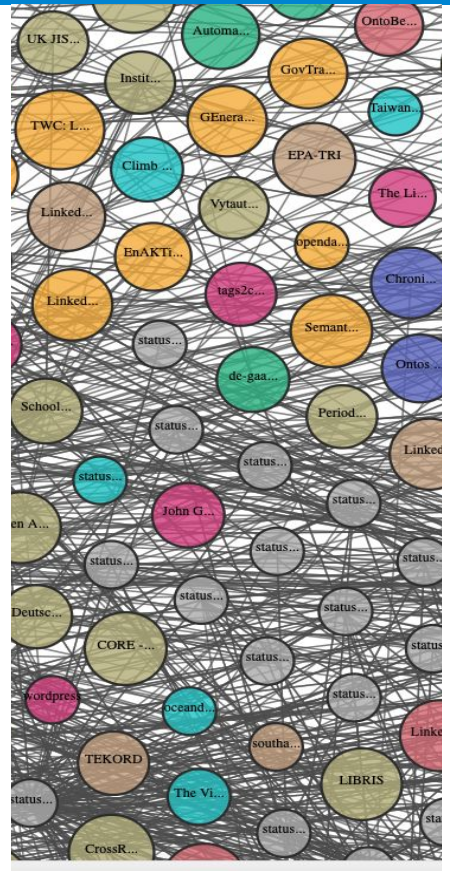
- With web preemption, a SPARQL server is scalable and fair by design
- The more we do on the server, the faster it is… (it is not a surprise…)
  - Data shipping should be avoided if possible because it is bad for performances, it is also bad for the planet.

Challenges and Opportunities

- Web preemption and updates
  - How to ensure snapshot isolation with web preemption?
- Playing with Quantum
  - How to adjust quantums to workload and resources?
- Web preemption and parallelism
  - Take advantage of web preemption to split a running query into multiple queries...

Challenges and Opportunities

- Privacy preservation
  - Personal Knowledge Graph:
    - Take back control of your data Solid project initiated by Tim Berners-Lee
  - What happens when I query private and public knowledge graphs?

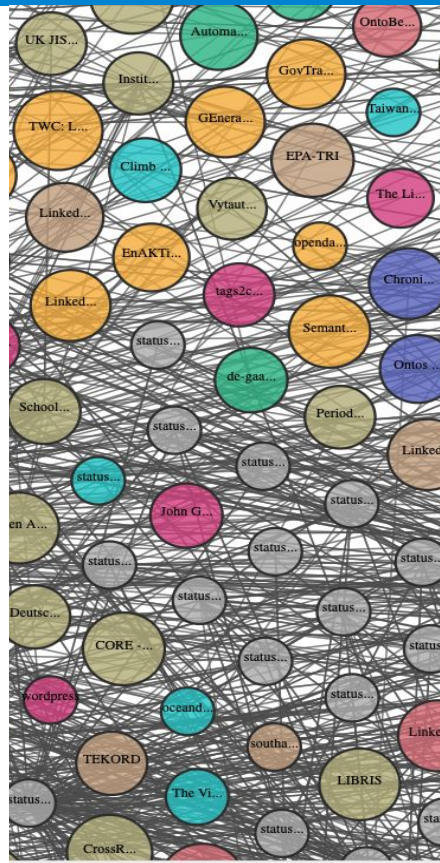


https://solidproject.org/

Challenges and Opportunities

- **Findability** of Knowledge graphs
  - Source selection services are local for a federated query engine and do not scale

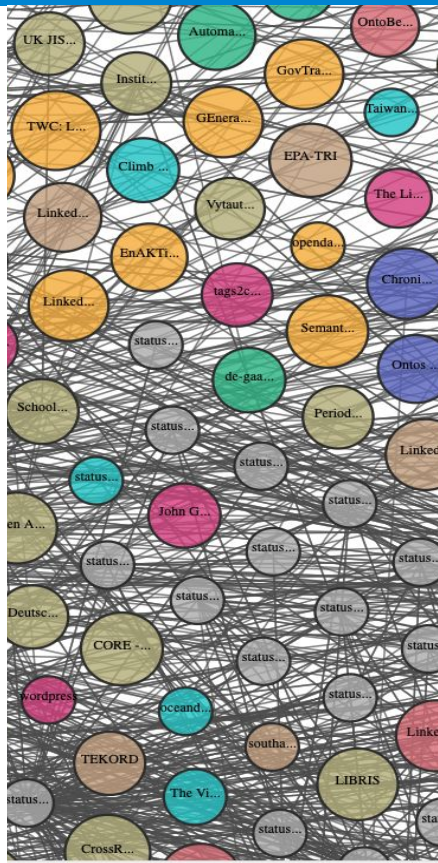
- Define a global source selection service that scales to the web
  - Findiability of Knowledge Graphs (à la Google)



Challenges and Opportunities

- Decentralized Knowledge Graphs (DeKaloG: ANR project at https://dekalog.univ-nantes.fr/)
  - Accessibility provided by the web preemption is a good starting point
    - Findinbaility ..
- RDF*/SPARQL*/Property Graph context for statements

  <<dbr:JulesVerne sch:educatedAt sch:Lycée-Clemenceau>> **sch:startTime** “1844”
    - Transparency…
- Query optimisation and machine learning



Olaf Hartig. Foundations to query labeled property graphs using sparql*. Technical report, 2019
Working group: https://w3c.github.io/rdf-star/

Querying Decentralized Knowledge Graphs

Live Demonstration

http://sage.univ-nantes.fr

Hala.Skaf@univ-nantes.fr

UNIVERSITÉ DE NANTES

Keynote 06-07