FedX:

A framework for efficiently evaluating SPARQL queries in a federated environment

CrEDIBLE working days, October 2013

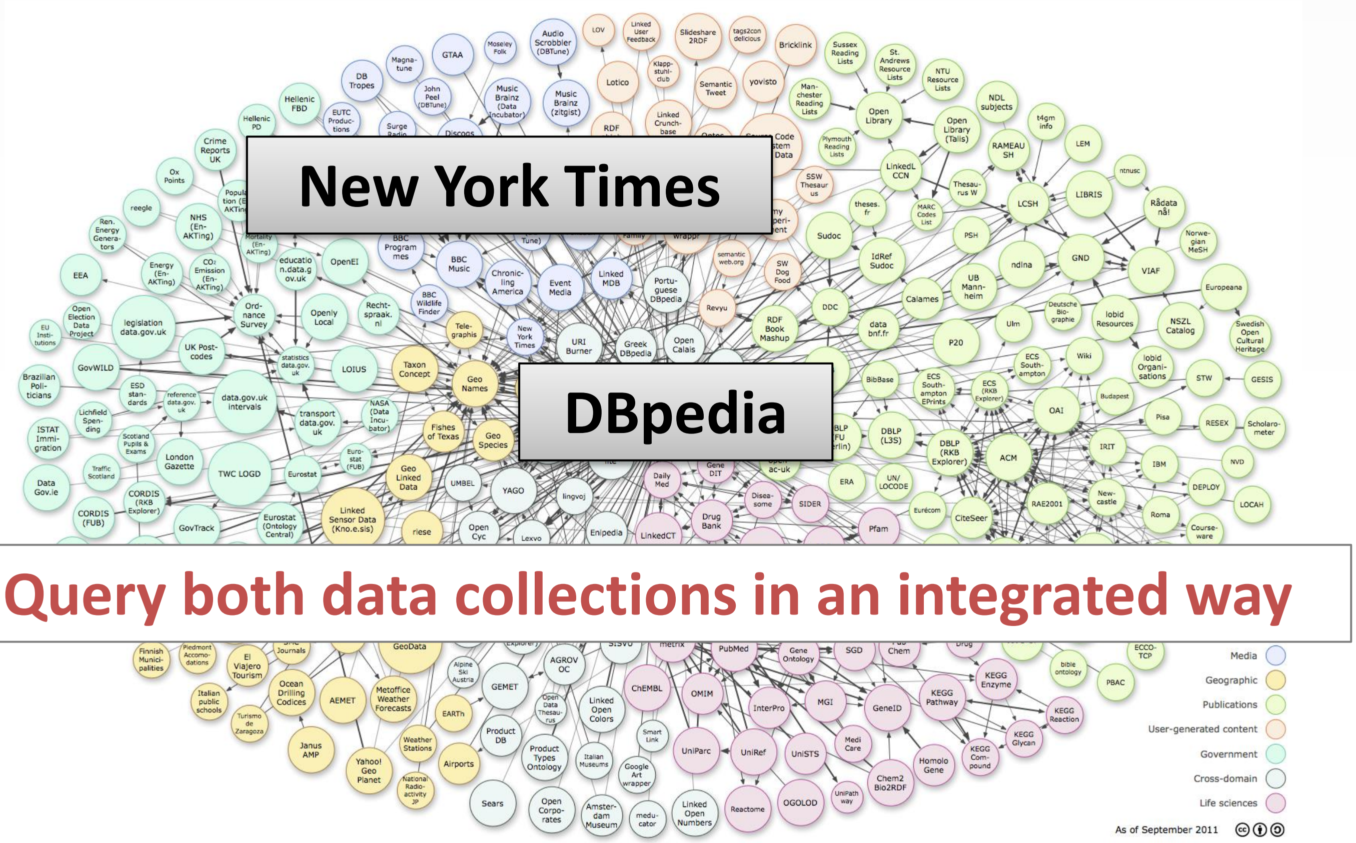
Andreas Schwarte, fluid Operations AG

Outline

- Introduction
- Federated Query Processing
- Optimization techniques in FedX
- Experiments
- Application scenarios
- Experiences & Outlook

Motivation

Query processing involving multiple distributed data sources, e.g. Linked Open Data cloud



New York Times

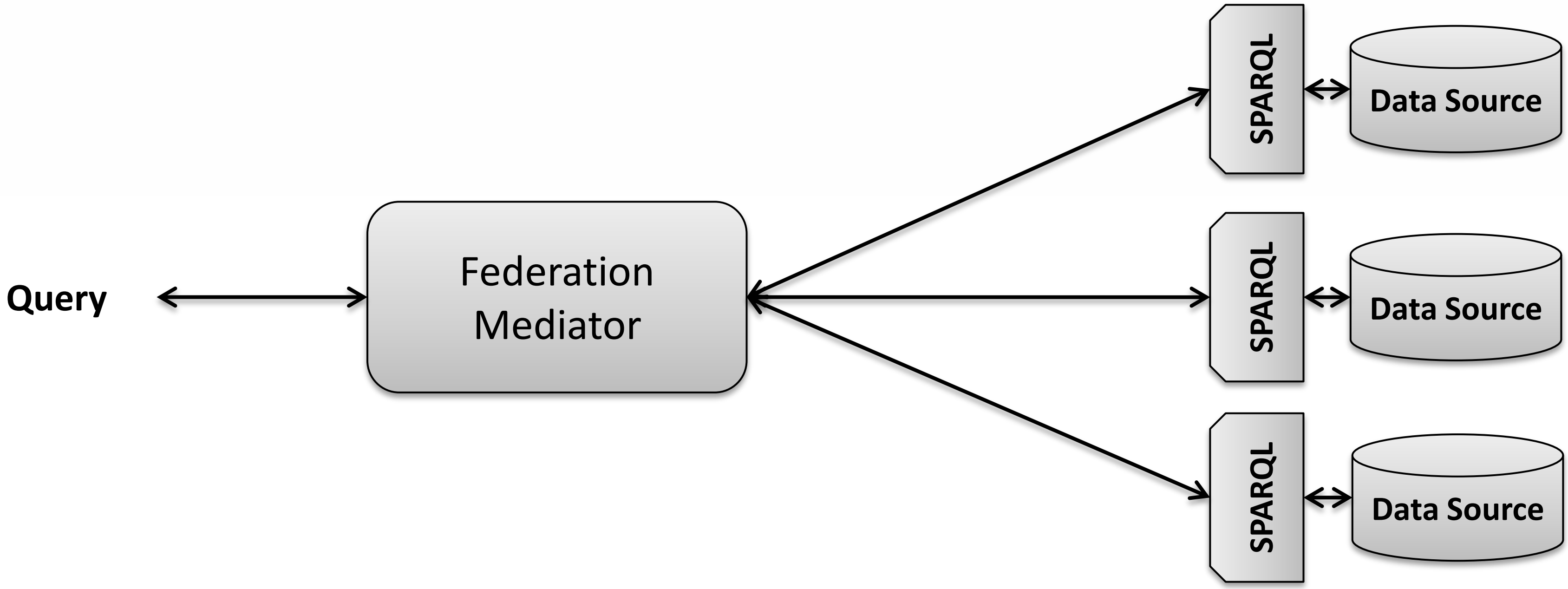
DBpedia

Query both data collections in an integrated way

Federated Query Processing

Federation mediator at the server

- ➔ Virtual integration of (remote) data sources
- ➔ Communication via SPARQL protocol



Federated Query Processing

Example Query from a General domain

Find US presidents and associated news articles

```
SELECT ?President ?Party ?TopicPage WHERE {
   ?President rdf:type dbpedia-yago:PresidentsOfTheUnitedStates .
   ?nytPresident owl:sameAs ?President .
   ?President dbpedia:party ?Party .
   ?nytPresident nytimes:topicPage ?TopicPage .
}
```

Federated Query Processing

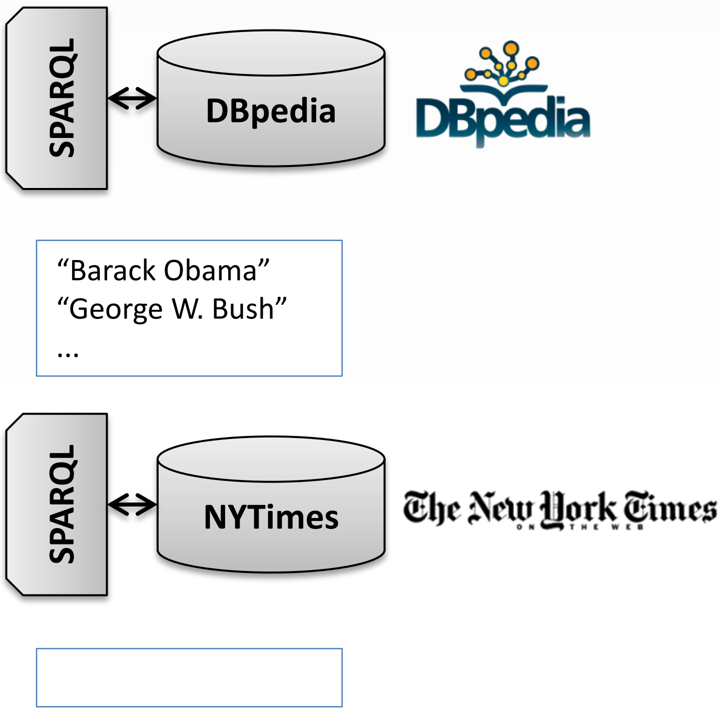
Query:

```
SELECT ?President ?Party ?TopicPage WHERE {
   ?President rdf:type dbpedia-yago:PresidentsOfTheUnitedStates .
   ?nytPresident owl:sameAs ?President .
   ...
}
```

Federation Mediator

?President rdf:type dbpedia-yago:PresidentsOfTheUnitedStates .

"Barack Obama"
"George W. Bush"
…



Federated Query Processing

Query:

```
SELECT ?President ?Party ?TopicPage WHERE {
   ?President rdf:type dbpedia-yago:PresidentsOfTheUnitedStates .
   ?nytPresident owl:sameAs ?President .
   …
}
```

Federation Mediator

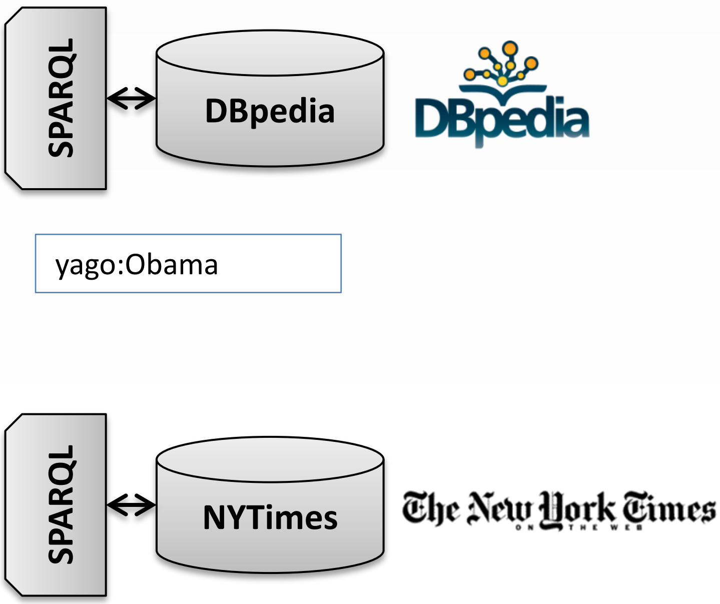
?nytPresident owl:sameAs “Barack Obama” .

Input:

“Barack Obama”
“George W. Bush”
…

Output:

“Barack Obama”, yago:Obama
“Barack Obama”, nyt:Obama



yago:Obama

nyt:Obama

Federated Query Processing

Query:

```
SELECT ?President ?Party ?TopicPage WHERE {
   ?President rdf:type dbpedia-yago:PresidentsOfTheUnitedStates .

   ...
}
```

Federation Mediator

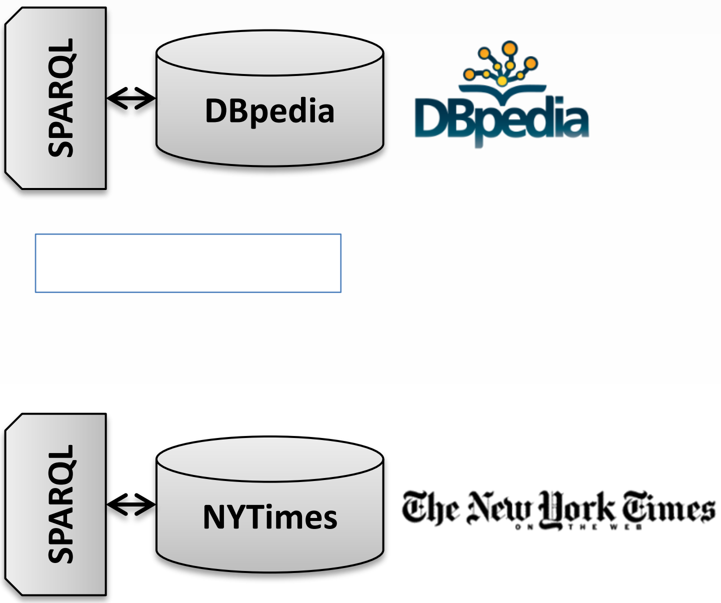
?nytPresident owl:sameAs “George W. Bush” .

Input:

“Barack Obama”
“George W. Bush”
…

Output:

“Barack Obama”, yago:Obama
“Barack Obama”, nyt:Obama
“George W. Bush”, nyt:Bush



nyt:Bush

… and so on for the other intermediate mappings and triple patterns …

FedX Query Processing Model

Scenario:

- Efficient SPARQL query processing on multiple distributed sources
- Full SPARQL 1.1 support
- Data sources are known and accessible as SPARQL endpoints
  - FedX is designed to be fully compatible with SPARQL 1.0 endpoints
- No a-priori knowledge about data sources
  - No local preprocessing of the data sources required
  - No need for pre-computed statistics
- On-demand federation setup
- Read-Only scenarios

Challenges to Federated Query Processing

1) Involve only relevant sources in the evaluation

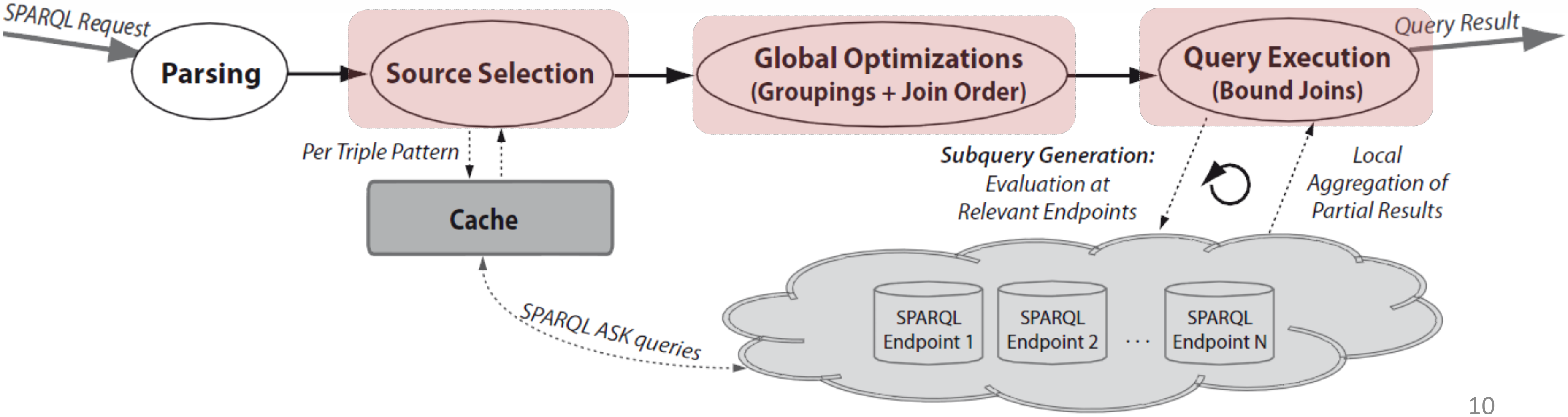
Avoid: Subqueries are sent to all sources, although potentially irrelevant

2) Compute joins close to the data

Avoid: All joins are executed locally in a nested loop fashion

3) Reduce remote communication

Avoid: Nested loop join that causes many remote requests



Optimization Techniques

1. Source Selection:

Idea:

Triple patterns are annotated with relevant sources

- Sources that can contribute information for a particular triple pattern
- SPARQL ASK requests in conjunction with a local cache
  - After a warm-up period the cache learns the capabilities of the data sources

    ➔ During Source Selection remote requests can be avoided

2. Exclusive Groups:

Idea:

Group triple patterns with the same single relevant source

- Evaluation in a single (remote) subquery
- Push join to the endpoint

Optimization Techniques (2)

Example: Source Selection + Exclusive Groups

```
SELECT ?President ?Party ?TopicPage WHERE {
 ?President rdf:type dbpedia-yago:PresidentsOfTheUnitedStates .
 ?President dbpedia:party ?Party .
 ?nytPresident owl:sameAs ?President .
 ?nytPresident nytimes:topicPage ?TopicPage .
}
```

Source Selection

@ DBpedia
@ DBpedia

Exclusive Group

@ DBpedia, NYTimes

@ NYTimes

Advantages:

- ➔ **Avoid sending subqueries to sources that are not relevant**
- ➔ **Delegate joins to the endpoint by forming exclusive groups (i.e. executing the respective patterns in a single subquery)**

Optimization Techniques (3)

3. Join Order:

Idea:

Iteratively determine the join order based on count-heuristic:

- Count free variables of triple patterns and groups
- Consider "resolved" variable mappings from earlier iteration

4. Bind Joins:

Idea:

Compute joins in a block nested loop fashion:

- Reduce the number of requests by "vectored" evaluation of a set of input bindings
- Renaming and Post-Processing technique for compliance with SPARQL 1.0
- Optional SPARQL 1.1 implementation using VALUES clause

Optimization Techniques (4)

Example: Bind Joins

```
SELECT ?President ?Party ?TopicPage WHERE {
   ?President rdf:type dbpedia:PresidentsOfTheUnitedStates .
   ?President dbpedia:party ?Party .
   ?nytPresident owl:sameAs ?President .
   ?nytPresident nytimes:topicPage ?TopicPage .
}
```

Assume that the following intermediate results have been computed as input for the last triple pattern

Block Input

"Barack Obama"
"George W. Bush"
…

Before (NLJ)

SELECT ?TopicPage WHERE { "Barack Obama" nytimes:topicPage ?TopicPage }
SELECT ?TopicPage WHERE { "George W. Bush" nytimes:topicPage ?TopicPage }
…

Now: Evaluation in a single remote request
construct + local post

SPARQL 1.1: VALUES clause implemented, but experiments show that UNION is more efficient

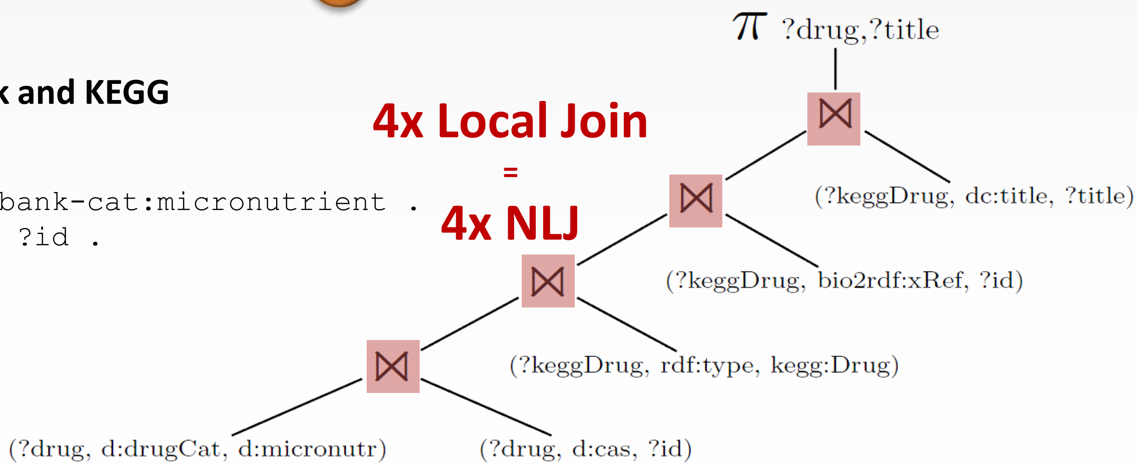
Optimization Example

1 SPARQL Query

Compute *Micronutrients* using Drugbank and KEGG

```
SELECT ?drug ?title WHERE {
   ?drug drugbank:drugCategory drugbank-cat:micronutrient .
   ?drug drugbank:casRegistryNumber ?id .
   ?keggDrug rdf:type kegg:Drug .
   ?keggDrug bio2rdf:xRef ?id .
   ?keggDrug dc:title ?title .
}
```

2 Unoptimized Internal Representation


π ?drug,?title
⋈
⋈
⋈
⋈
(?keggDrug, dc:title, ?title)
(?keggDrug, bio2rdf:xRef, ?id)
(?keggDrug, rdf:type, kegg:Drug)
(?drug, d:drugCat, d:micronutr)
(?drug, d:cas, ?id)

4x Local Join

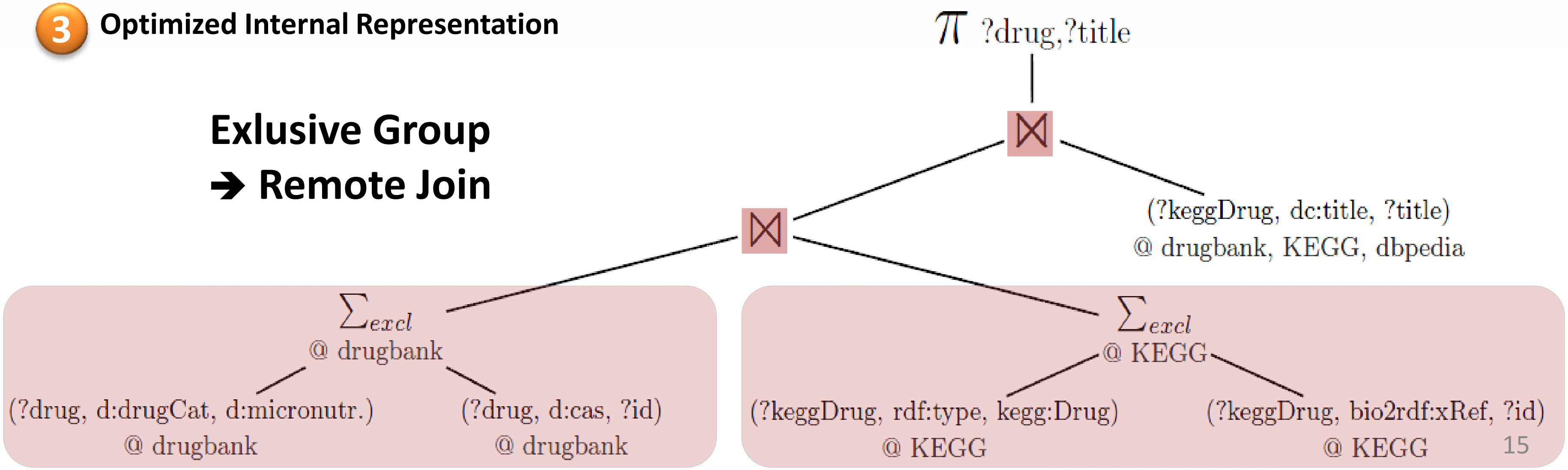
=

4x NLJ

3 Optimized Internal Representation

Exlusive Group

➔ Remote Join


π ?drug,?title
⋈
⋈
(?keggDrug, dc:title, ?title)
@ drugbank, KEGG, dbpedia
Σexcl
@ drugbank
(?drug, d:drugCat, d:micronutr.)
@ drugbank
(?drug, d:cas, ?id)
@ drugbank
Σexcl
@ KEGG
(?keggDrug, rdf:type, kegg:Drug)
@ KEGG
(?keggDrug, bio2rdf:xRef, ?id)
@ KEGG

Experiments

Based on FedBench benchmark suite

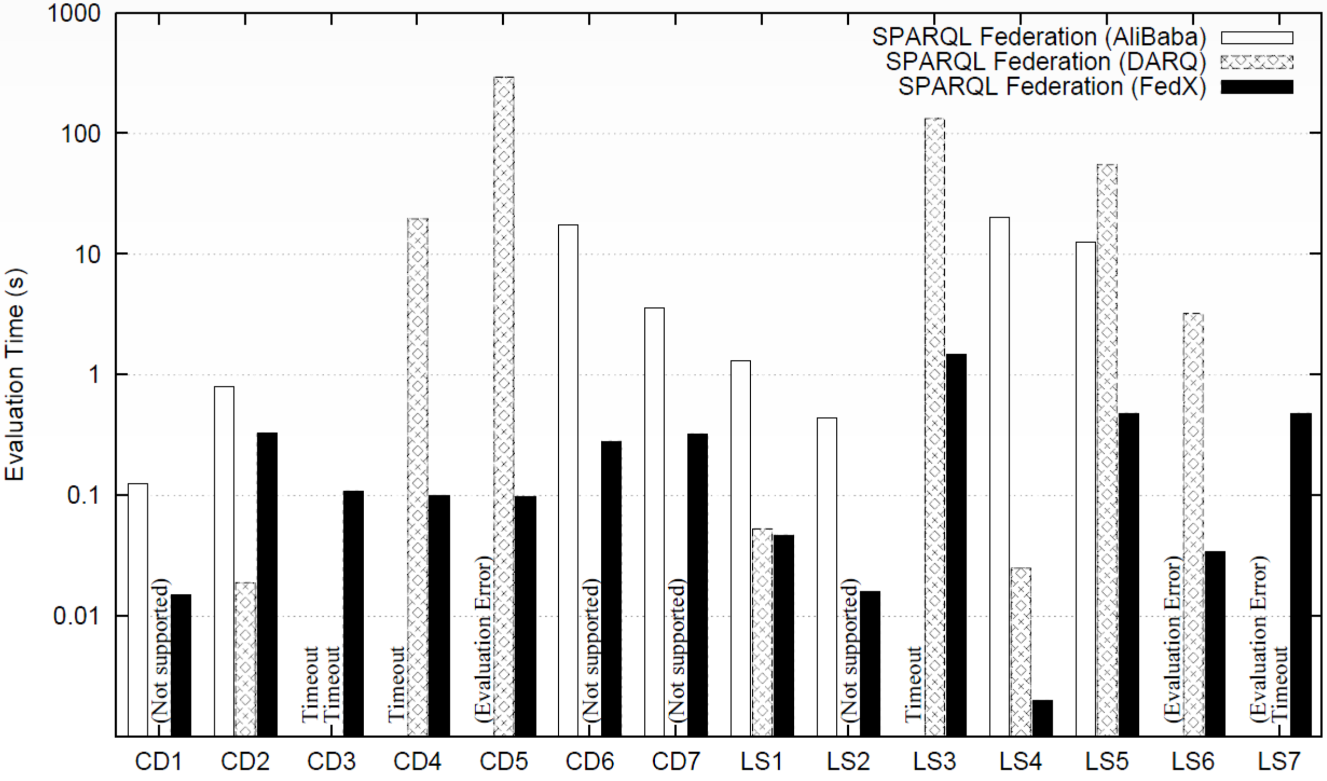
- 14 queries from the *Cross Domain* (CD) and *Life Science* (LS) collections
- Real-World Data from the Linked Open Data cloud
- Federation with 5 (CD) and 4 (LS) data sources
- Queries vary in complexity, size, structure, and sources involved

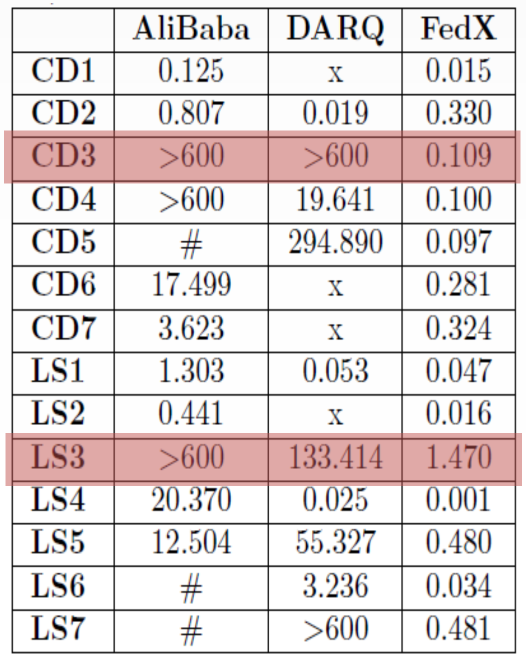
Benchmark environment

- HP Proliant 2GHz 4Core, 32GB RAM
- 20GB RAM for server (federation mediator)
- Local copies of the SPARQL endpoint to ensure reproducibility and reliability of the service
  - Provided by the FedBench Framework

Experiments (2)

a) Evaluation times of Cross Domain (CD) and Life Science (LS) queries

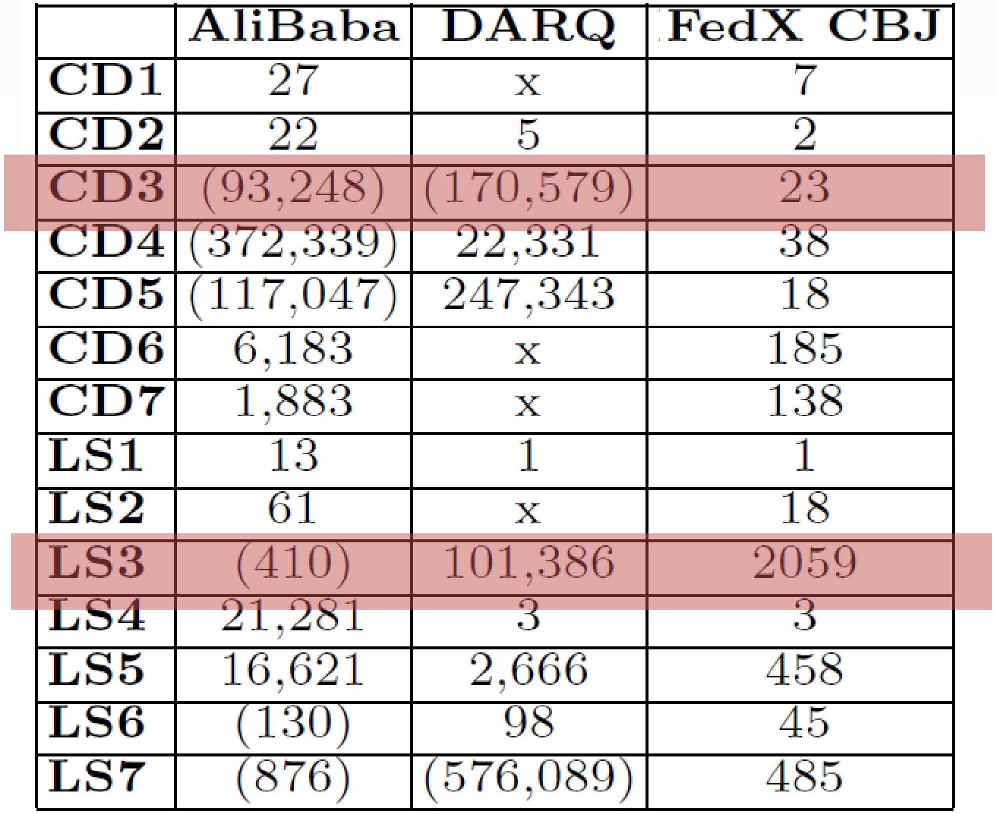




	AliBaba	DARQ	FedX
CD1	0.125	x	0.015
CD2	0.807	0.019	0.330
CD3	>600	>600	0.109
CD4	>600	19.641	0.100
CD5	#	294.890	0.097
CD6	17.499	x	0.281
CD7	3.623	x	0.324
LS1	1.303	0.053	0.047
LS2	0.441	x	0.016
LS3	>600	133.414	1.470
LS4	20.370	0.025	0.001
LS5	12.504	55.327	0.480
LS6	#	3.236	0.034
LS7	#	>600	0.481

Experiments (3)

b) Number of requests sent to the endpoints



	AliBaba	DARQ	FedX CBJ
CD1	27	x	7
CD2	22	5	2
CD3	(93,248)	(170,579)	23
CD4	(372,339)	22,331	38
CD5	(117,047)	247,343	18
CD6	6,183	x	185
CD7	1,883	x	138
LS1	13	1	1
LS2	61	x	18
LS3	(410)	101,386	2059
LS4	21,281	3	3
LS5	16,621	2,666	458
LS6	(130)	98	45
LS7	(876)	(576,089)	485

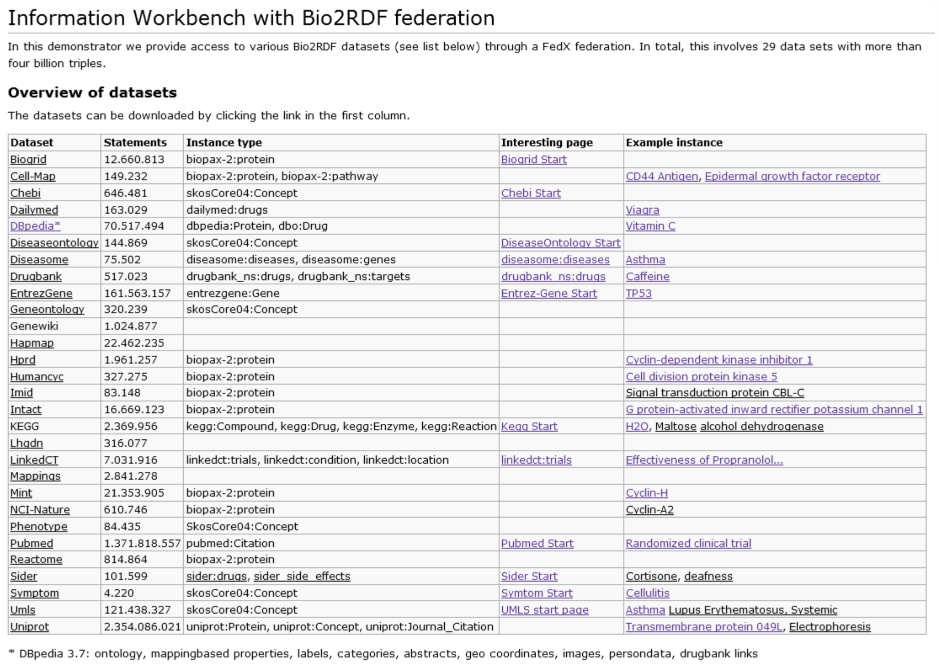
Runtimes
AliBaba: >600s
DARQ: >600s
FedX: 0.109s

Runtimes
AliBaba: >600s
DARQ: 133s
FedX: 1.4s

Application Scenarios

Bio2RDF scenario:

- 29 datasets with more than 4 billion triples integrated in the Information Workbench
  - Structured queries, instance pages, and dashboards
  - Example: PubMed publications, Trials, Diseases, etc.



Information Workbench with Bio2RDF federation

In this demonstrator we provide access to various Bio2RDF datasets (see list below) through a FedX federation. In total, this involves 29 data sets with more than four billion triples.

Overview of datasets

The datasets can be downloaded by clicking the link in the first column.

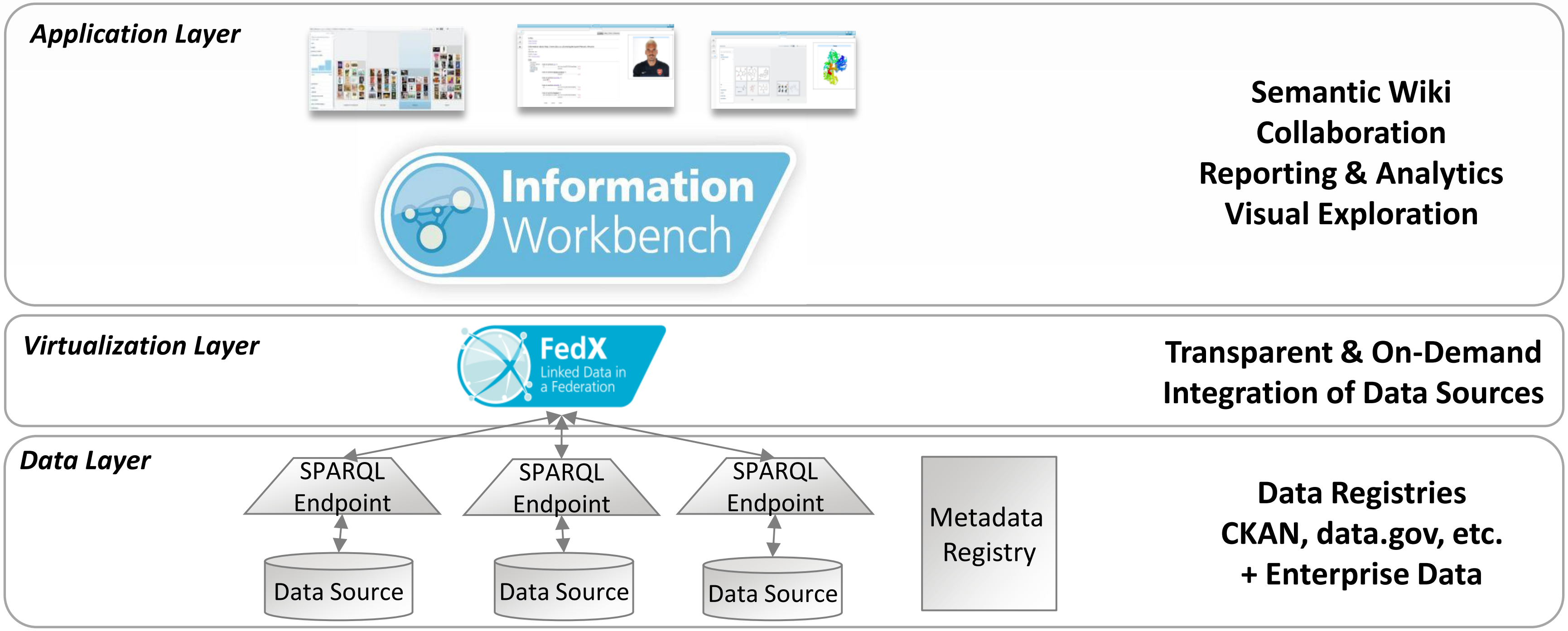
Dataset	Statements	Instance type	Interesting page	Example instance
Biogrid	12.660.813	biopax-2:protein	Biogrid Start	
Cell-Map	149.232	biopax-2:protein, biopax-2:pathway		CD44 Antigen, Epidermal growth factor receptor
Chebi	646.481	skosCore04:Concept	Chebi Start	
Dailymed	163.029	dailymed:drugs		Viagra
DBpedia*	70.517.494	dbpedia:Protein, dbo:Drug		Vitamin C
Diseaseontology	144.869	skosCore04:Concept	DiseaseOntology Start	
Diseasome	75.502	diseasome:diseases, diseasome:genes	diseasome:diseases	Asthma
Drugbank	517.023	drugbank_ns:drugs, drugbank_ns:targets	drugbank_ns:drugs	Caffeine
EntrezGene	161.563.157	entrezgene:Gene	Entrez-Gene Start	TP53
Geneontology	320.239	skosCore04:Concept		
Genewiki	1.024.877			
Hapmap	22.462.235			
Hprd	1.961.257	biopax-2:protein		Cyclin-dependent kinase inhibitor 1
Humancyc	327.275	biopax-2:protein		Cell division protein kinase 5
Imid	83.148	biopax-2:protein		Signal transduction protein CBL-C
Intact	16.669.123	biopax-2:protein		G protein-activated inward rectifier potassium channel 1
KEGG	2.369.956	kegg:Compound, kegg:Drug, kegg:Enzyme, kegg:Reaction	Kegg Start	H2O, Maltose alcohol dehydrogenase
Lhgdn	316.077			
LinkedCT	7.031.916	linkedct:trials, linkedct:condition, linkedct:location	linkedct:trials	Effectiveness of Propranolol...
Mappings	2.841.278			
Mint	21.353.905	biopax-2:protein		Cyclin-H
NCI-Nature	610.746	biopax-2:protein		Cyclin-A2
Phenotype	84.435	SkosCore04:Concept		
Pubmed	1.371.818.557	pubmed:Citation	Pubmed Start	Randomized clinical trial
Reactome	814.864	biopax-2:protein		
Sider	101.599	sider:drugs, sider_side_effects	Sider Start	Cortisone, deafness
Symptom	4.220	skosCore04:Concept	Symtom Start	Cellulitis
Umls	121.438.327	skosCore04:Concept	UMLS start page	Asthma Lupus Erythematosus, Systemic
Uniprot	2.354.086.021	uniprot:Protein, uniprot:Concept, uniprot:Journal_Citation		Transmembrane protein 049L, Electrophoresis

* DBpedia 3.7: ontology, mappingbased properties, labels, categories, abstracts, geo coordinates, images, persondata, drugbank links

See “An Experience Report of Large Scale Federations” (http://arxiv.org/abs/1210.5403)

FedX – The Bigger Picture

Information Workbench:
Integration of Virtualized Data Sources as a Service
(incl. Enterprise data sources)



Application Layer

Semantic Wiki
Collaboration
Reporting & Analytics
Visual Exploration

Virtualization Layer

Transparent & On-Demand
Integration of Data Sources

Data Layer

Data Registries
CKAN, data.gov, etc.
+ Enterprise Data

Experiences & Outlook

Federation in practice

- Requires reliable federation members
  - SPARQL endpoints in controlled environments (local intranet)
  - Hard to deal with unreachable / broken endpoints
- Works best for queries with clearly separated vocabulary / namespaces
- Linking between datasets needs to be improved
- Query performance quite efficient and good for static applications (e.g. dashboarding)
  - Not yet suitable for highly interactive applications

Outlook

- Statistics layer to improve source selection and join ordering
- Support for write scenarios
- New join strategies (Hash Join instead of BNLJ)
- Component to prune subqueries by namespace

Thank you!

Contact

fluid Operations AG
Altrottstr. 31
Walldorf
Germany

+49 (0) 6227 358087-0
www.fluidops.com
contact@fluidOps.com

Further information on FedX
http://www.fluidops.com/fedx

References

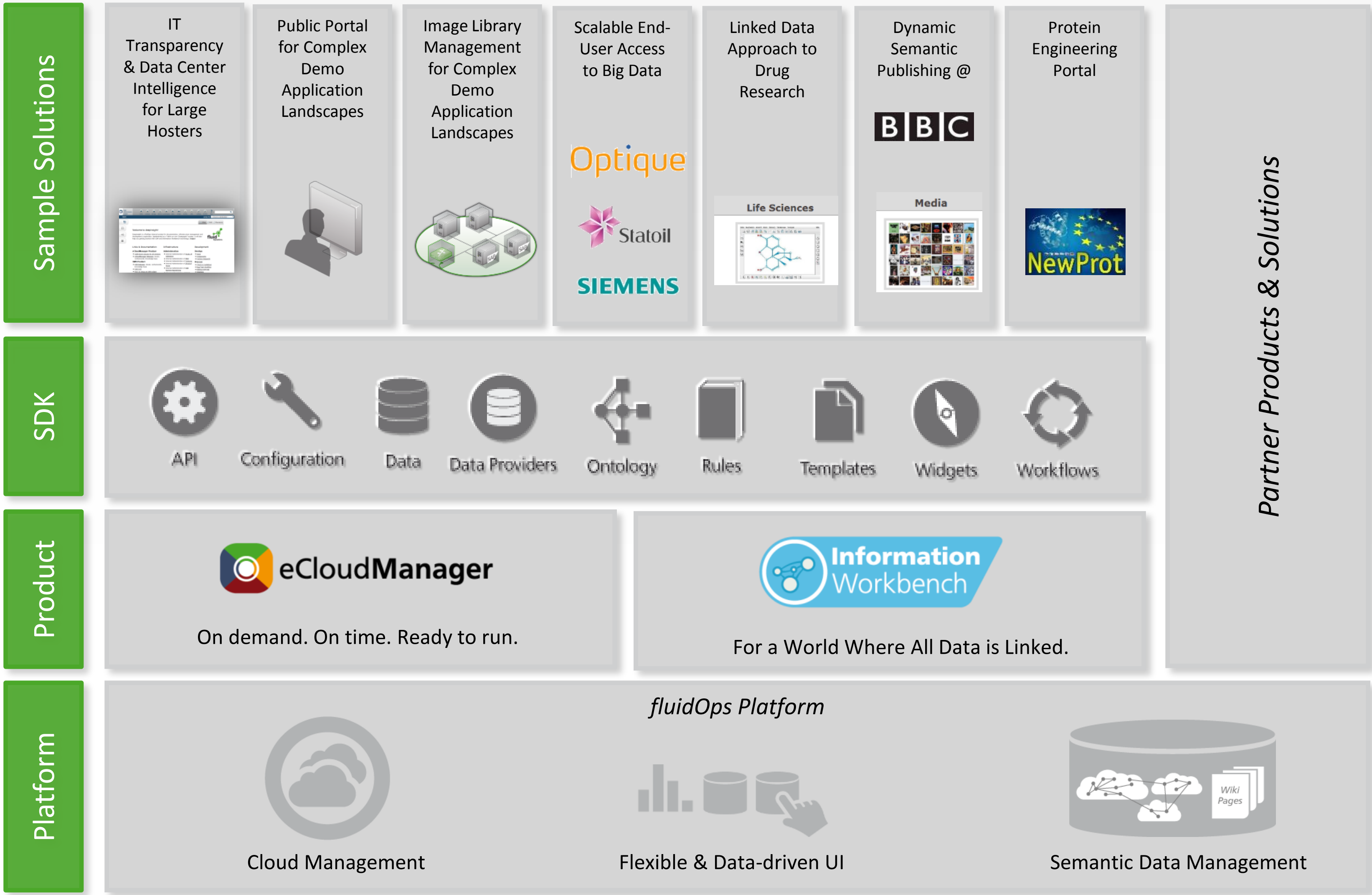
FedX: Optimization Techniques for Federated Query Processing on Linked Data
Andreas Schwarte, Peter Haase, Katja Hose, Ralf Schenkel, Michael Schmidt.
In Proc. ISWC 2011, Bonn (Germany).

FedBench: A Benchmark Suite for Federated Semantic Data Query Processing
Michael Schmidt, Olaf Görlitz, Peter Haase, Günter Ladwig, Andreas Schwarte, Thanh Tran.
In Proc. ISWC 2011, Bonn (Germany).

An Experience Report of Large Scale Federations
Andreas Schwarte, Peter Haase, Michael Schmidt, Katja Hose, Ralf Schenkel
http://arxiv.org/abs/1210.5403

FedSearch: efficiently combining structured queries and full-text search in a SPARQL federation
Andriy Nikolov, Andreas Schwarte, Christian Hütter
ISWC 2013, Sidney (Australia).

The fluidOps Platform



Sample Solutions

IT Transparency & Data Center Intelligence for Large Hosters

Public Portal for Complex Demo Application Landscapes

Image Library Management for Complex Demo Application Landscapes

Scalable End-User Access to Big Data

Linked Data Approach to Drug Research

Dynamic Semantic Publishing @

Protein Engineering Portal

Partner Products & Solutions

SDK

API

Configuration

Data

Data Providers

Ontology

Rules

Templates

Widgets

Workflows

Product

eCloudManager

On demand. On time. Ready to run.

Information Workbench

For a World Where All Data is Linked.

Platform

fluidOps Platform

Cloud Management

Flexible & Data-driven UI

Semantic Data Management