UPPSALA UNIVERSITET

Scientific Analysis by Queries in Extended SPARQL over a Scalable e-Science Data Store

Andrej Andrejev, Salman Toor, Andreas Hellander*, Sverker Holmgren, Tore Risch

Department of Information Technology, Uppsala University
** Department of Computer Science, University of California Santa Barbara*

andrej.andrejev@it.uu.se

UPPSALA UNIVERSITET

- Introduction
- SciSPARQL overview
- Evaluation
- RDF views over external storage systems
- Related approaches
- Summary

Motivation

Big data needs

- *scalable* data management
- good *documentation*
- *easy access*
- *reuse* of existing software packages

UPPSALA UNIVERSITET

Motivation

Big data needs

- *scalable* data management
  - **standard** relational database management systems,
  - specialized e-Science data stores
- good *documentation*
- *easy access*
- *reuse* of existing software packages

Motivation

Big data needs

- *scalable* data management
  - **standard** relational database management systems,
  - specialized e-Science data stores
- good *documentation*
  - **standard** W3C representation for metadata: RDF
- *easy access*
- *reuse* of existing software packages

Motivation

Big data needs

- *scalable* data management
  - **standard** relational database management systems,
  - specialized e-Science data stores
- good *documentation*
  - **standard** W3C representation for metadata: RDF
- *easy access*
  - **standard** W3C query language for searching RDF databases: SPARQL
- *reuse* of existing software packages

UPPSALA UNIVERSITET

Motivation

Big data needs

- *scalable* data management
  - **standard** relational database management systems,
  - specialized e-Science data stores
- good *documentation*
  - **standard** W3C representation for metadata: RDF
- *easy access*
  - **standard** W3C query language for searching RDF databases: SPARQL
- *reuse* of existing software packages
  - calling **standard** and custom libraries from queries

Problem with RDF databases

RDF (**R**esource **D**escription **F**ramework) – a W3C standard
"metadata data model"

- RDF is very suitable for describing properties about scientific experiments (metadata) **but**:
  - Scientific data usually involves numerical arrays
  - Arrays are represented in a very inefficient way in RDF

- **Our approach**: Extend RDF with compact numerical array representation called *Numeric Multidimensional Arrays (NMA)*

Problem with SPARQL queries

SPARQL (**S**PARQL **P**rotocol **a**nd **R**DF **Q**uery **L**anguage) – a W3C standard language for querying RDF

- SPARQL is very suitable for searching scientific RDF-based metadata, **but**:
  - SPARQL has no support for queries involving array operations

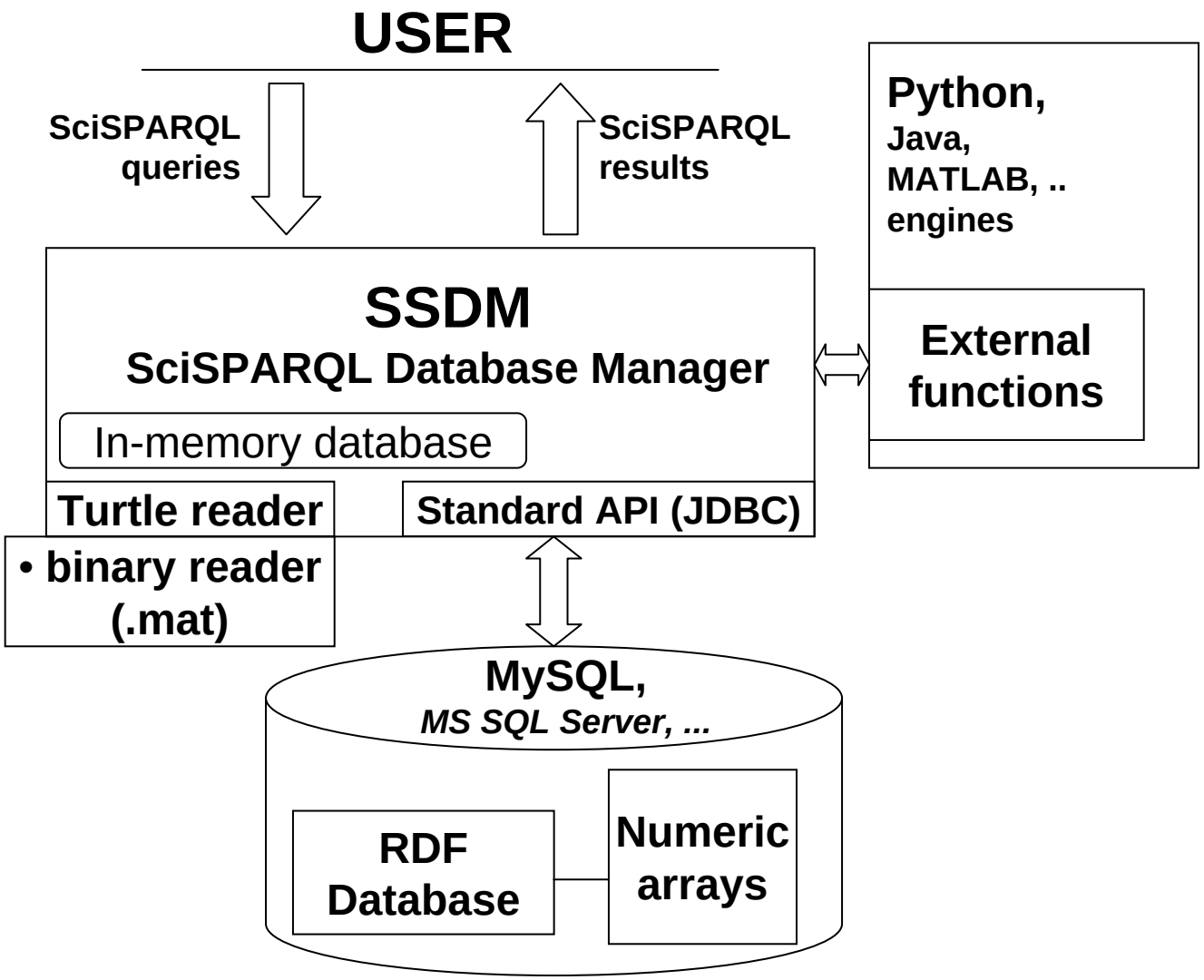
- **Our approach**: Extent SPARQL with common array operators => SciSPARQL

Reusing program libraries

- Often need for using existing program libraries when processing experiments data, **but**:
  - SPARQL has no standard way of plugging in external program libraries and algorithms

- **Our approach**: SciSPARQL provides a general mechanism to call functions in C, Java, Python, or MATLAB

Our System Architecture



UPPSALA UNIVERSITET

- Introduction
- SciSPARQL overview
- Evaluation
- RDF views over external storage systems
- Related approaches
- Summary

Basic RDF experimental metadata

EXAMPLE

Basic RDF experimental metadata

EXAMPLE

Basic SPARQL metadata query

EXAMPLE

- *RDF database of triples:*

```
prefix : <http://udbl.it.uu.se/bistab#>

:Task1 :inExperiment :Experiment1 .
:Experiment1 :responsibleName ”Andrej” .
```

Basic SPARQL metadata query

EXAMPLE

- *RDF database of triples:*

```
prefix : <http://udbl.it.uu.se/bistab#>

:Task1 :inExperiment :Experiment1 .
:Experiment1 :responsibleName ”Andrej” .
```

- *Select all the tasks that Andrej is responsible for*

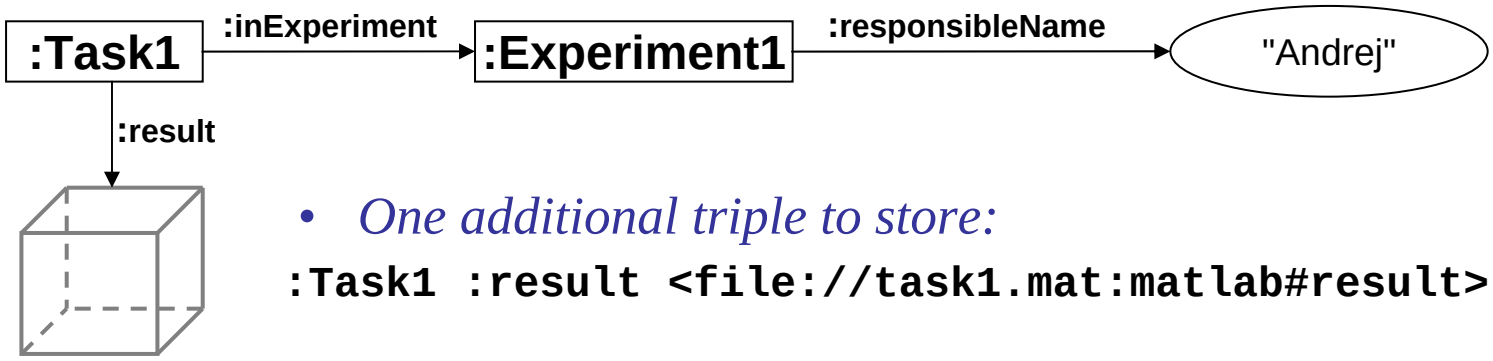
```
SELECT ?task
 WHERE { ?task :inExperiment ?experiment .
         ?experiment :responsibleName "Andrej" }
```

SSDM extends RDF with arrays

EXAMPLE

SciSPARQL extends SPARQL with array access

EXAMPLE

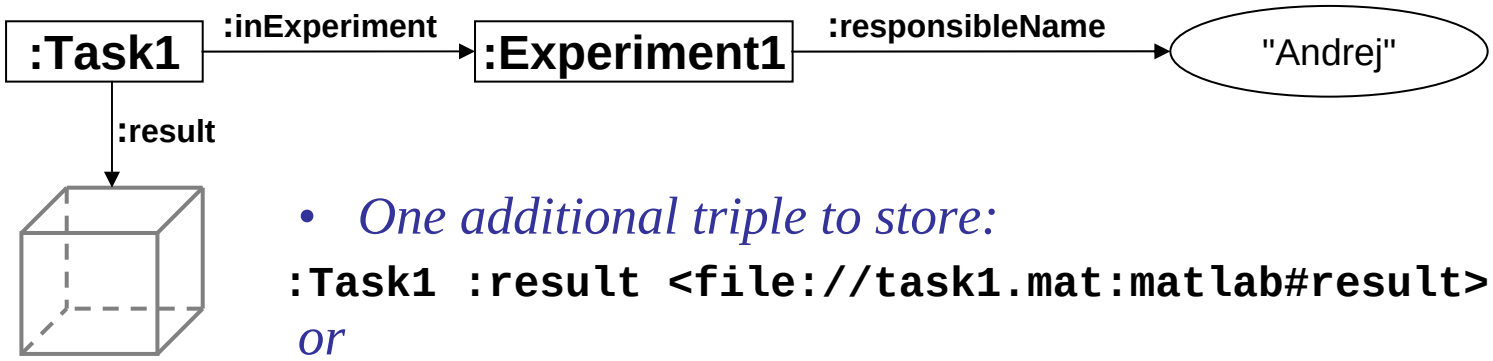


- *One additional triple to store:*

```
:Task1 :result <file://task1.mat:matlab#result> .
```

SciSPARQL extends SPARQL with array access

EXAMPLE



- *One additional triple to store:*

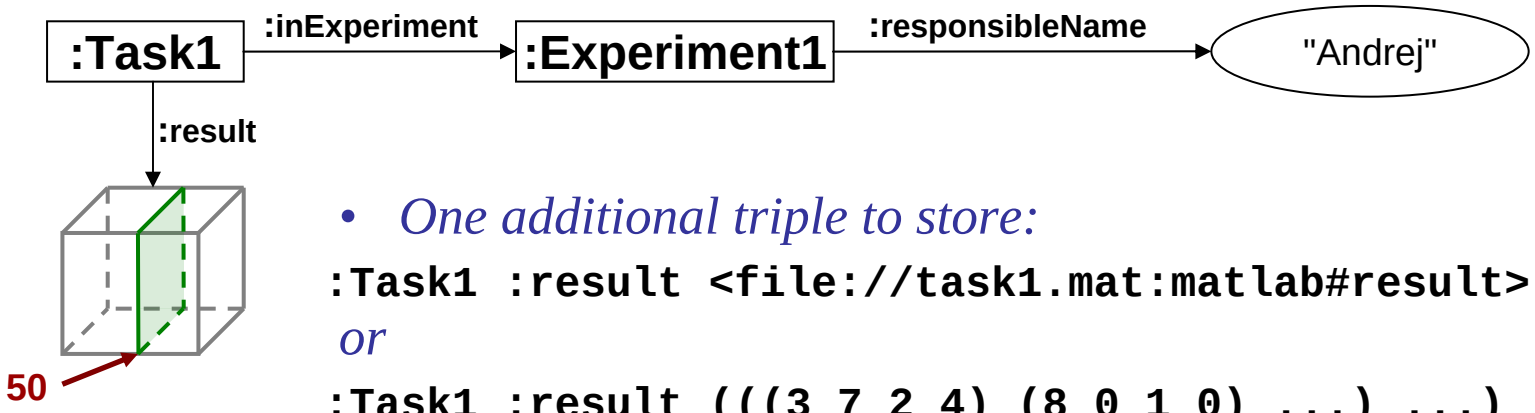
```
:Task1 :result <file://task1.mat:matlab#result> .
```

or

```
:Task1 :result (((3 7 2 4) (8 0 1 0) ...) ...) ...) .
```

SciSPARQL extends SPARQL with array access

EXAMPLE



- *One additional triple to store:*

```
:Task1 :result <file://task1.mat:matlab#result> .
```

or

```
:Task1 :result (((3 7 2 4) (8 0 1 0) ...) ...) ...) .
```

- *Select 50-slice of "result" arrays of all tasks that Andrej is responsible for*

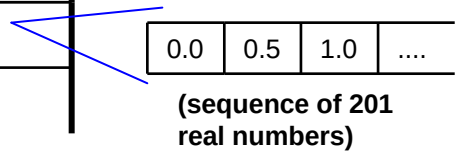
```
SELECT (?result[50,:,:] AS ?slice50)
 WHERE { ?task :result ?result ;
               :inExperiment ?experiment .
         ?experiment :responsibleName "Andrej" }
```

Relational Representation

EXAMPLE
of a relational database describing a BISTAB scientific experiments

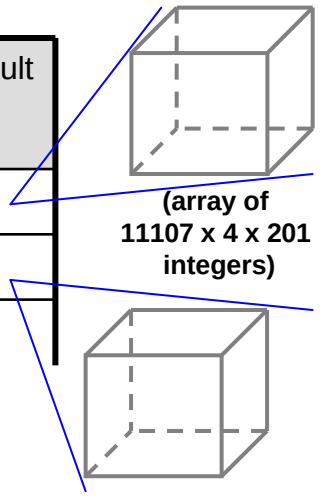
Experiment

id	mesh	simulation algorithm	# cells	# species	specie ids				time points
					A	B	E_A	E_B	
1	triangular #1	nsm	11107	4	0	1	2	3	

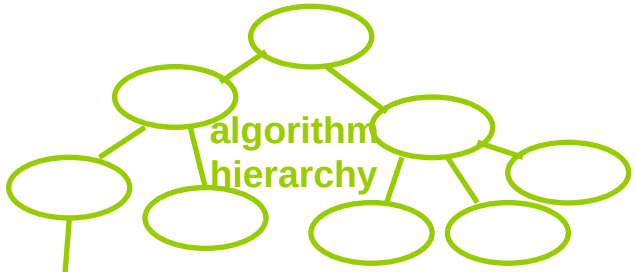


Task

id	experiment id	parameters				realization	result
		k_1	k_a	k_d	k_4		
1	1	32.159	79.279	782750669.857	53.286	1	
2	1	19.151	39.044	300035857.676	73.445	1	



Relational Representation



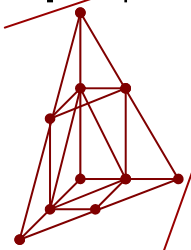
more types of species

Experiment

id	mesh	simulation algorithm	# cells	# species	specie ids				time points
					A	B	E_A	E_B	
1	triangular #1	nsm	11107	4	0	1	2	3	

0.0	0.5	1.0	....

(sequence of 201 real numbers)

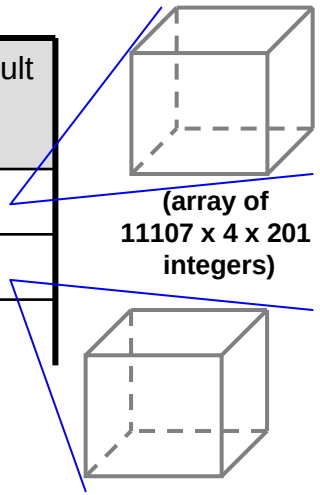


vertex graph with coordinates

more parameters

Task

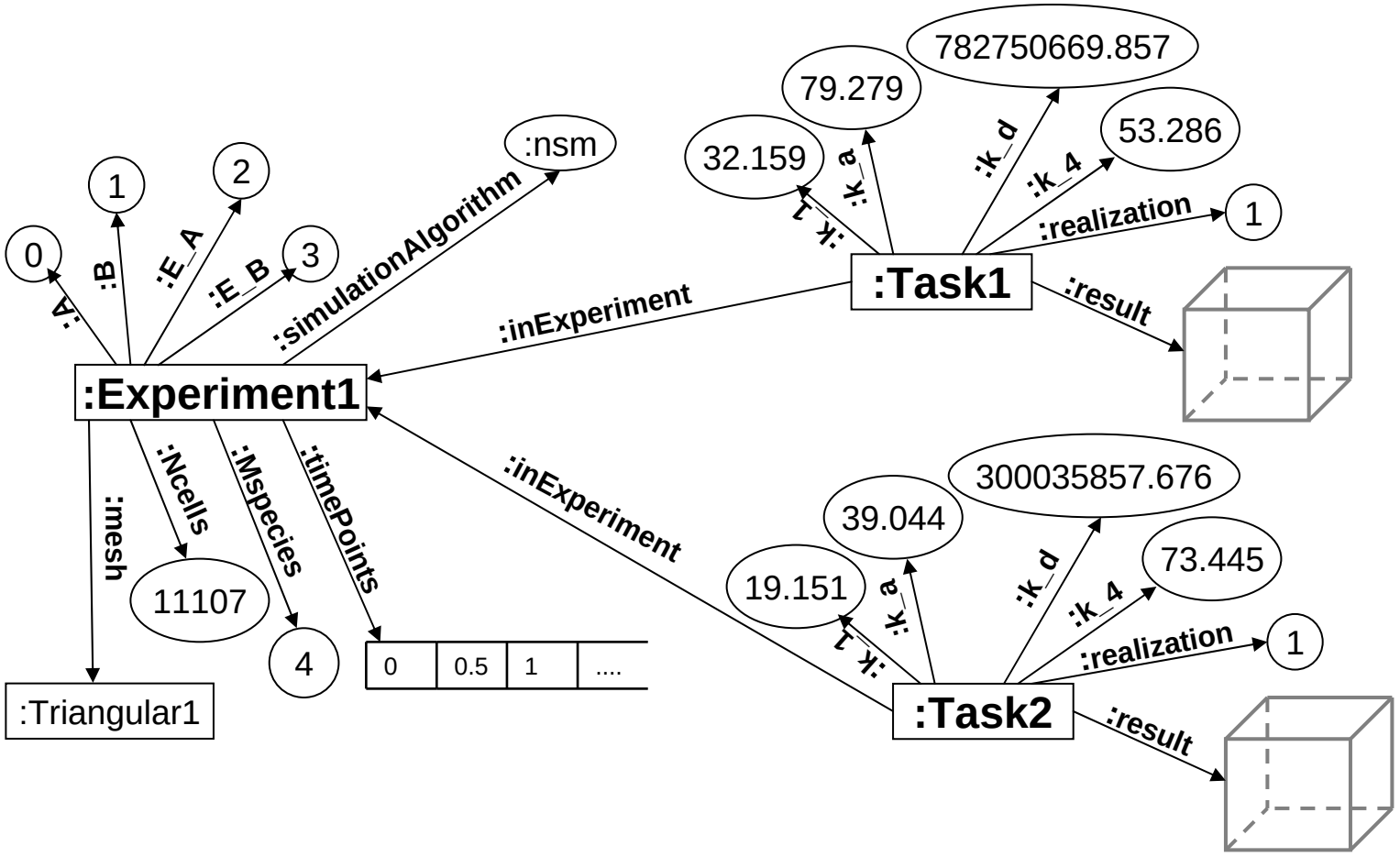
id	experiment id	parameters				realization	result
		k_1	k_a	k_d	k_4		
1	1	32.159	79.279	782750669.857	53.286	1	
2	1	19.151	39.044	300035857.676	73.445	1	



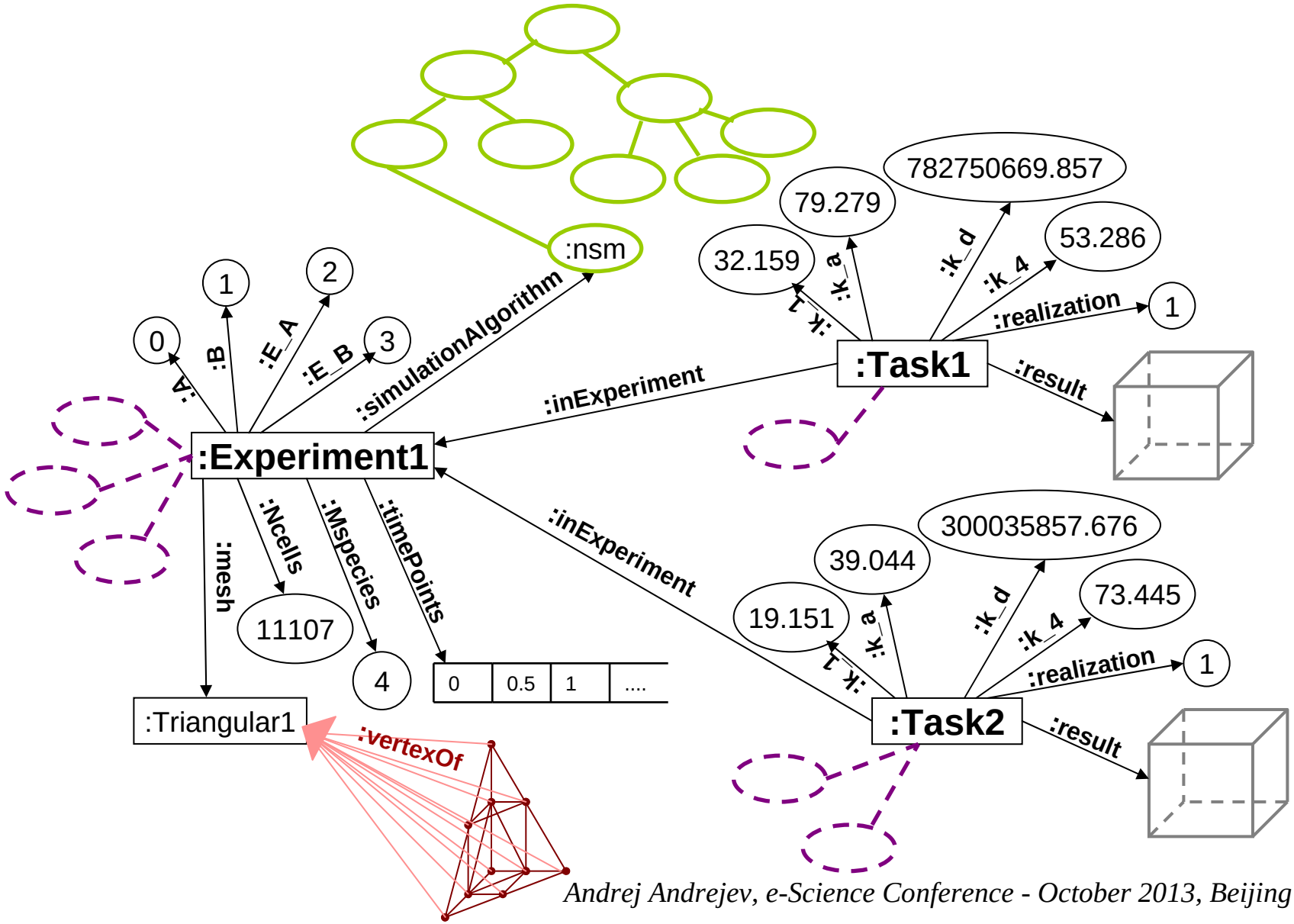
(array of 11107 x 4 x 201 integers)

SSDM Representation: RDF with Arrays

EXAMPLE
of an RDF database describing BISTAB scientific experiments



SSDM Representation: RDF with Arrays



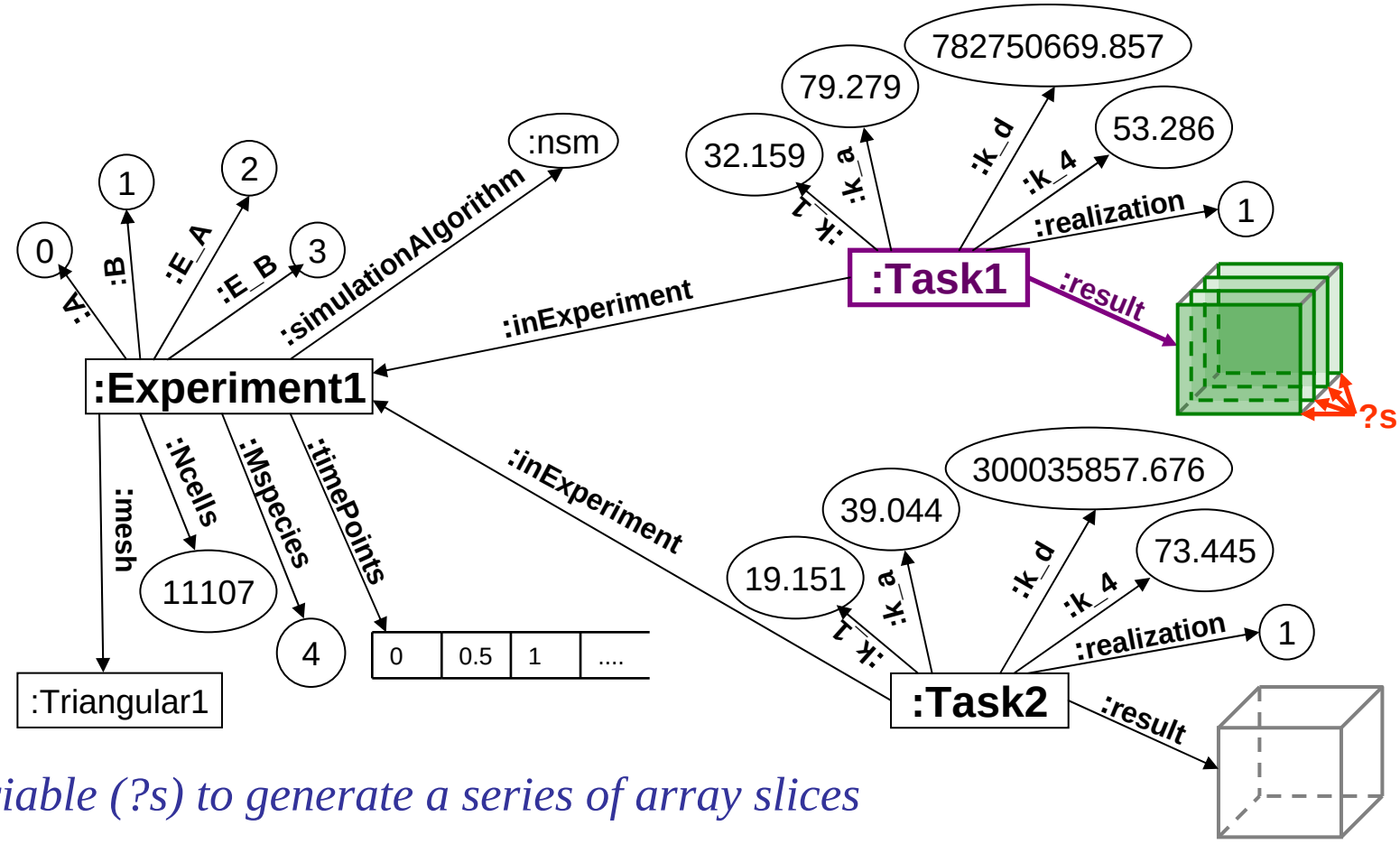
More

Scientifc SPARQL

examples

SciSPARQL Query Language

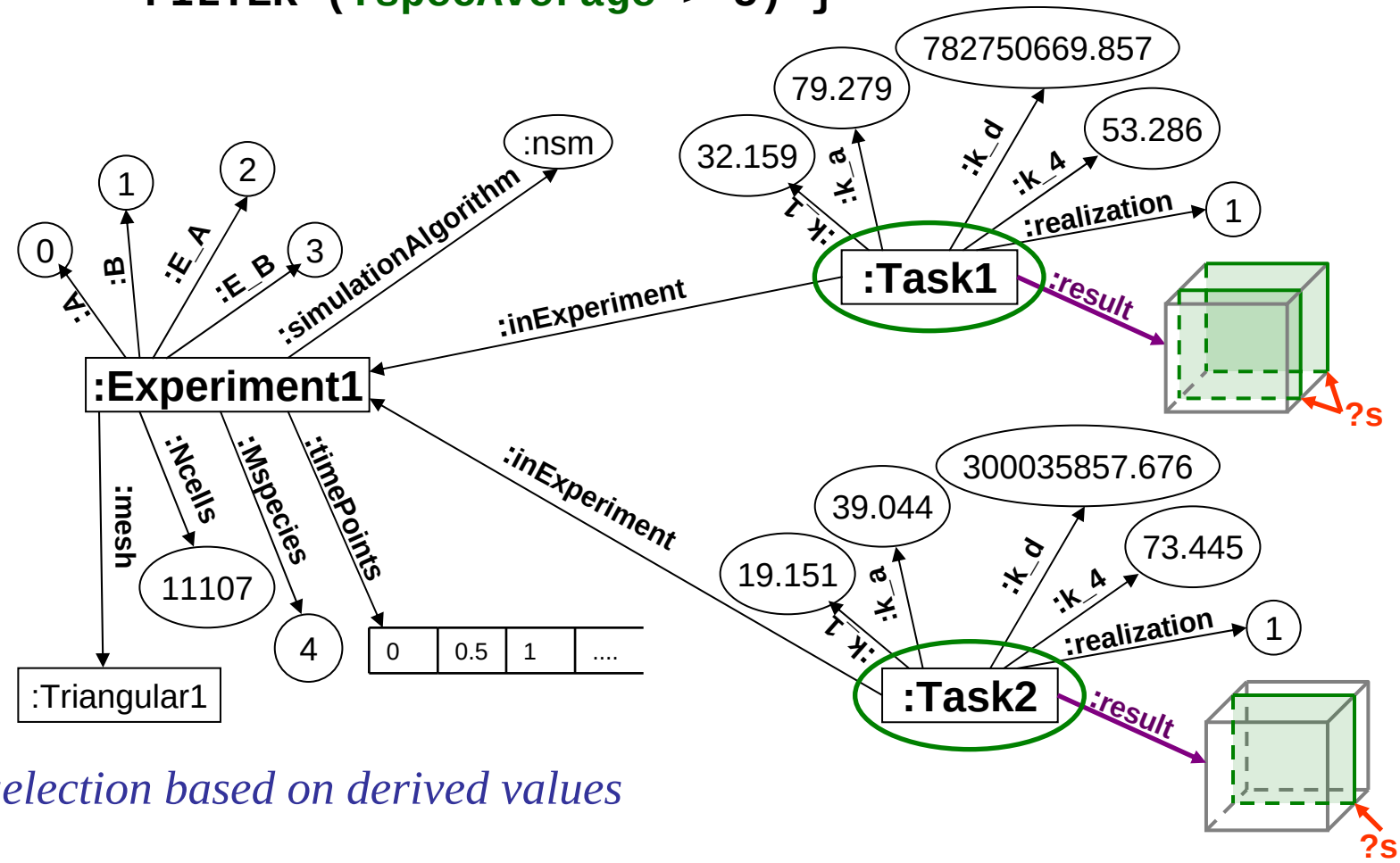
```
SELECT (AVG(?result[:,:,?s]) AS ?specAvarage)
 WHERE { :Task1 :result ?result }
```



- *Use free variable (?s) to generate a series of array slices*

SciSPARQL Query Language

```
SELECT ?task ?s ?specAverage
 WHERE { ?task :result ?result .
         BIND (AVG(?result[:,:,?s]) AS ?specAvarage) .
         FILTER (?specAverage > 5) }
```



- *Filter data selection based on derived values*

- Introduction
- SciSPARQL overview
- Evaluation
- RDF views over external storage systems
- Related approaches
- Summary

UPPSALA UNIVERSITET

Our Contribution

SSDM shows performance on par with MATLAB, with added value of

MATLAB	SciSPARQL
Programs implementing analysis algorithms	High-level queries
No metadata management user manually manages files	Uniform management of both data and metadata

Our Contribution

SSDM shows performance on par with MATLAB, with added value of

MATLAB

```
sum_of_A = [];
load('input.mat'); % parameters, tspan 'metadata'
t = find(tspan==10);
a = 1; % this 'metadata' is not stored anywhere
mspecies = 8;
for ii=1:100 % amount of files should be known!
    if parameters(1,ii) >= 50
       && parameters(1,ii) <= 90
       && parameters(3,ii) >= 1.0E8
       && parameters(3,ii) <= 1.0E9
      realization = strcat( % consruct filenames
        'C:/DATA/bistab2f/realization_',
        int2str(ii),'_1.mat');
      load(realization); % load matrices 1-by-1
      sum_of_A = [sum_of_A
                  sum(UU(a:mspecies:end, t))];
    end
end
sum_of_A;
```

SciSPARQL Q2

```
SELECT (array_sum(?U[?a-1::?mspecies,?j]) AS ?res)
 WHERE { ?task :U ?U ; # retrieve data
               :k_a ?k_a ; # retrieve metadata
               :k_d ?k_d ;
               :inExperiment ?experiment .
         ?experiment :A ?a ;
                     :MSpecies ?mspecies ;
                     :tspan ?tspan .
         FILTER (?tspan[?j] = 10 &&
                 1.0E8 <= ?k_d && ?k_d <= 1.0E9 &&
                 50 <= ?k_a && ?k_a <= 90 ) };
```

Our Contribution

```
sum_of_A = [];
load('input.mat'); % parameters, tspan 'metadata'
t = find(tspan==10);
a = 1; % this 'metadata' is not stored anywhere
mspecies = 8;
for ii=1:100 % amount of files should be known!
    if parameters(1,ii) >= 50
       && parameters(1,ii) <= 90
       && parameters(3,ii) >= 1.0E8
       && parameters(3,ii) <= 1.0E9
      realization = strcat( % consruct filenames
        'C:/DATA/bistab2f/realization_',
        int2str(ii),'_1.mat');
      load(realization); % load matrices 1-by-1
      sum_of_A = [sum_of_A
                  sum(UU(a:mspecies:end, t))];
    end
end
sum_of_A;
```

Our Contribution

SSDM shows performance on par with MATLAB, with added value of

```
SELECT (array_sum(?U[?a-1::?mspecies,?j]) AS ?res)
 WHERE { ?task :U ?U ; # retrieve data
               :k_a ?k_a ; # retrieve metadata
               :k_d ?k_d ;
               :inExperiment ?experiment .
         ?experiment :A ?a ;
                     :MSpecies ?mspecies ;
                     :tspan ?tspan .
         FILTER (?tspan[?j] = 10 &&
                 1.0E8 <= ?k_d && ?k_d <= 1.0E9 &&
                 50 <= ?k_a && ?k_a <= 90 ) };
```

```
sum_of_A =
load('input
t = find(ts
a = 1; % th
mspecies =
for ii=1:10
    if para
       && p
       && p
       && p
      reali
        'C:
        int2str(ii), '.mat' );
      load(realization); % load matrices 1-by-1
      sum_of_A = [sum_of_A
                  sum(UU(a:mspecies:end, t))];
    end
end
sum_of_A;
```

SSDM Performance

7GB database, query execution times (in seconds) with all data on disk

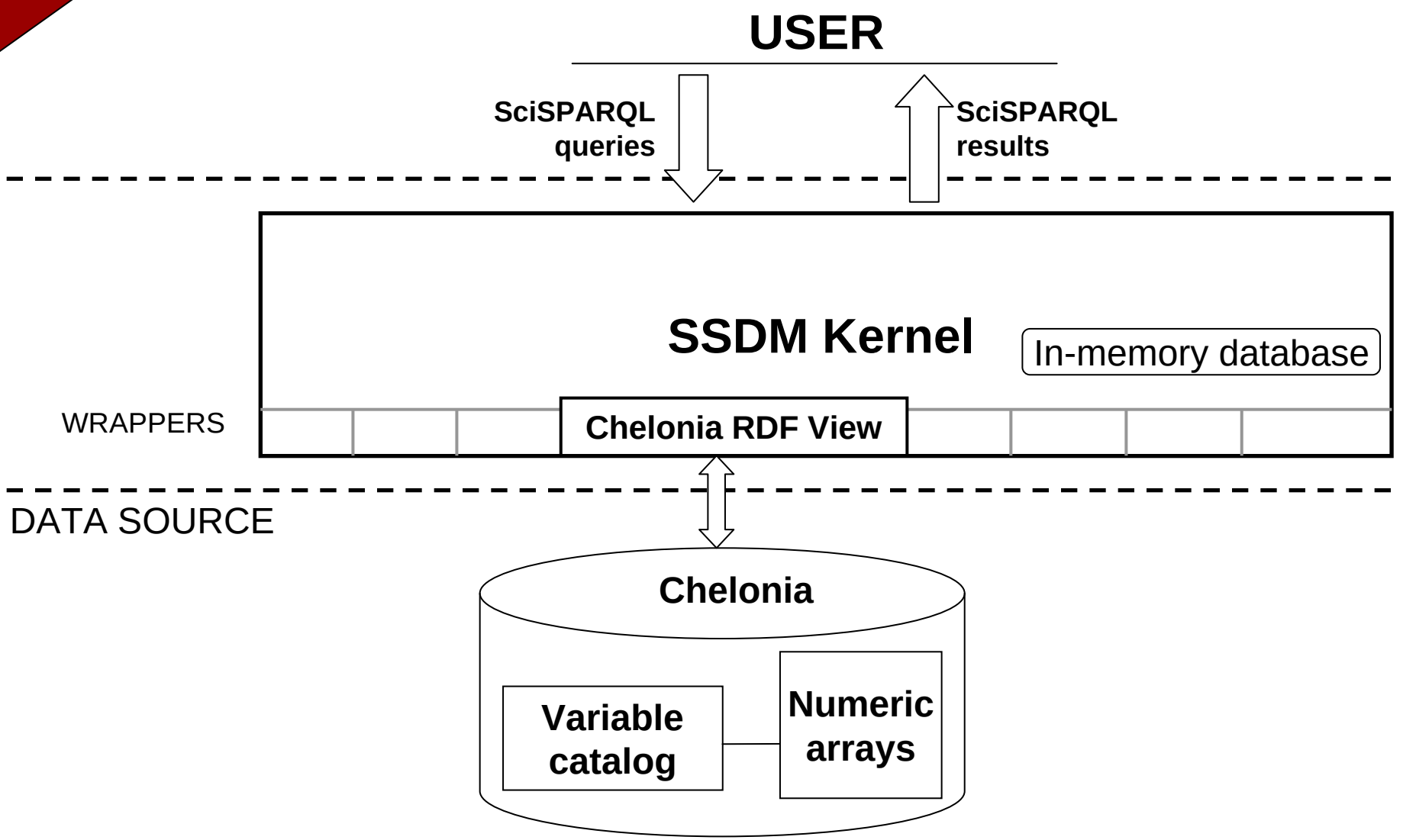
Task	Data retrieved	SSDM with back-end		MATLAB script
		MySQL	MS SQL Server	
Q1: (selective query) Compute an aggregate value over 1 big matrix, every 8th row	**18MB**	**1.748**	**2.15**	**1.826**
Q2: (SSDM worst case) Select 36 matrices, access one column × every 8th row	**642MB**	**80.703**	**44.512**	**30.042**
Q3: (database scan) Compute AGRMAX of Q1 across all matrices, 25% rows	**1785MB**	**187.073**	**192.365**	**133.279**

=> SSDM provides desired functionality with comptetitive performance

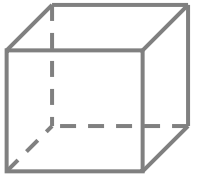
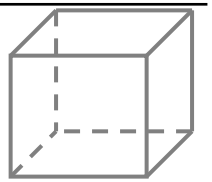
- Introduction
- SciSPARQL overview
- Evaluation
- RDF views over external storage systems
- Related approaches
- Summary

UPPSALA UNIVERSITET

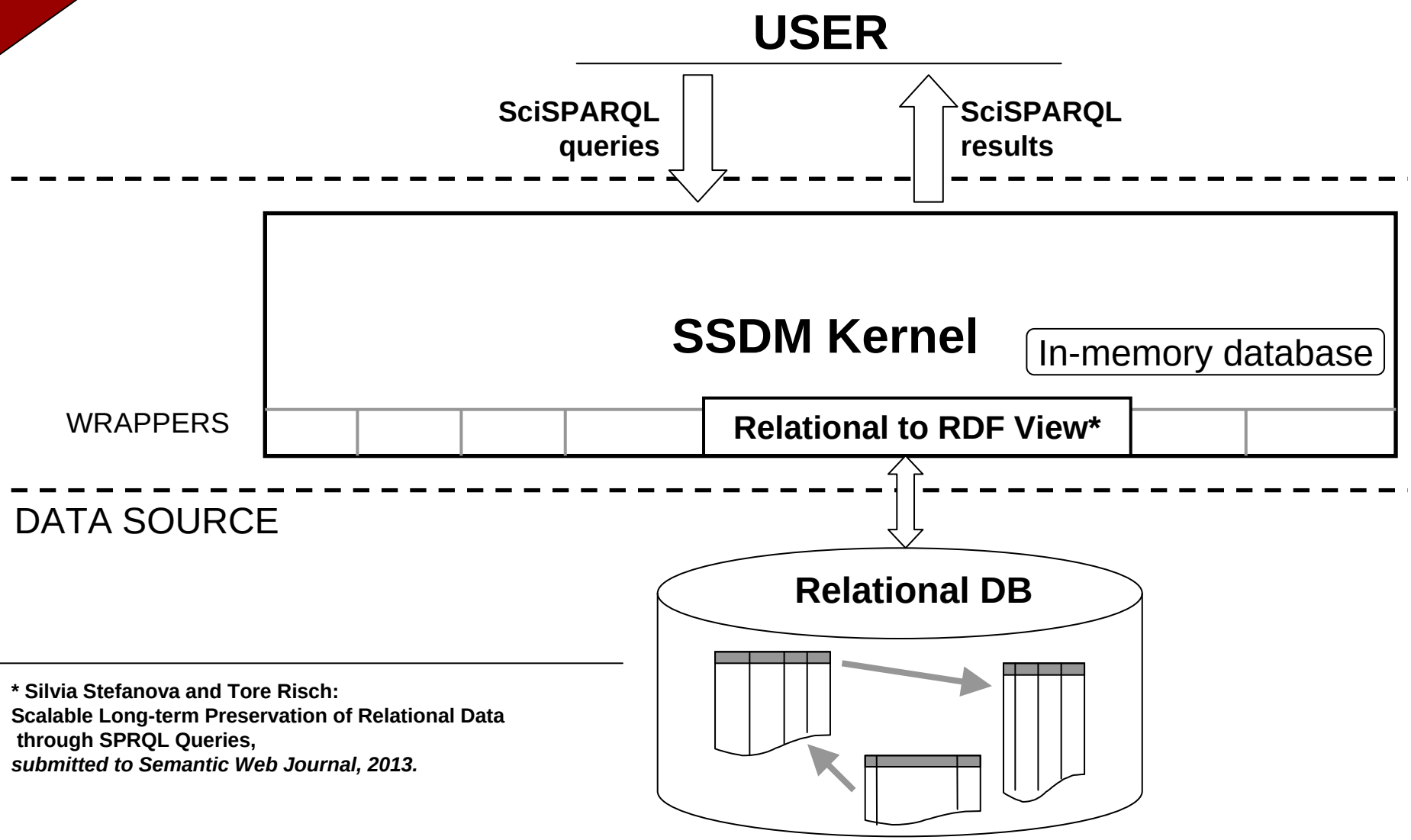
RDF views over external storage systems



Chelonia Native Schema

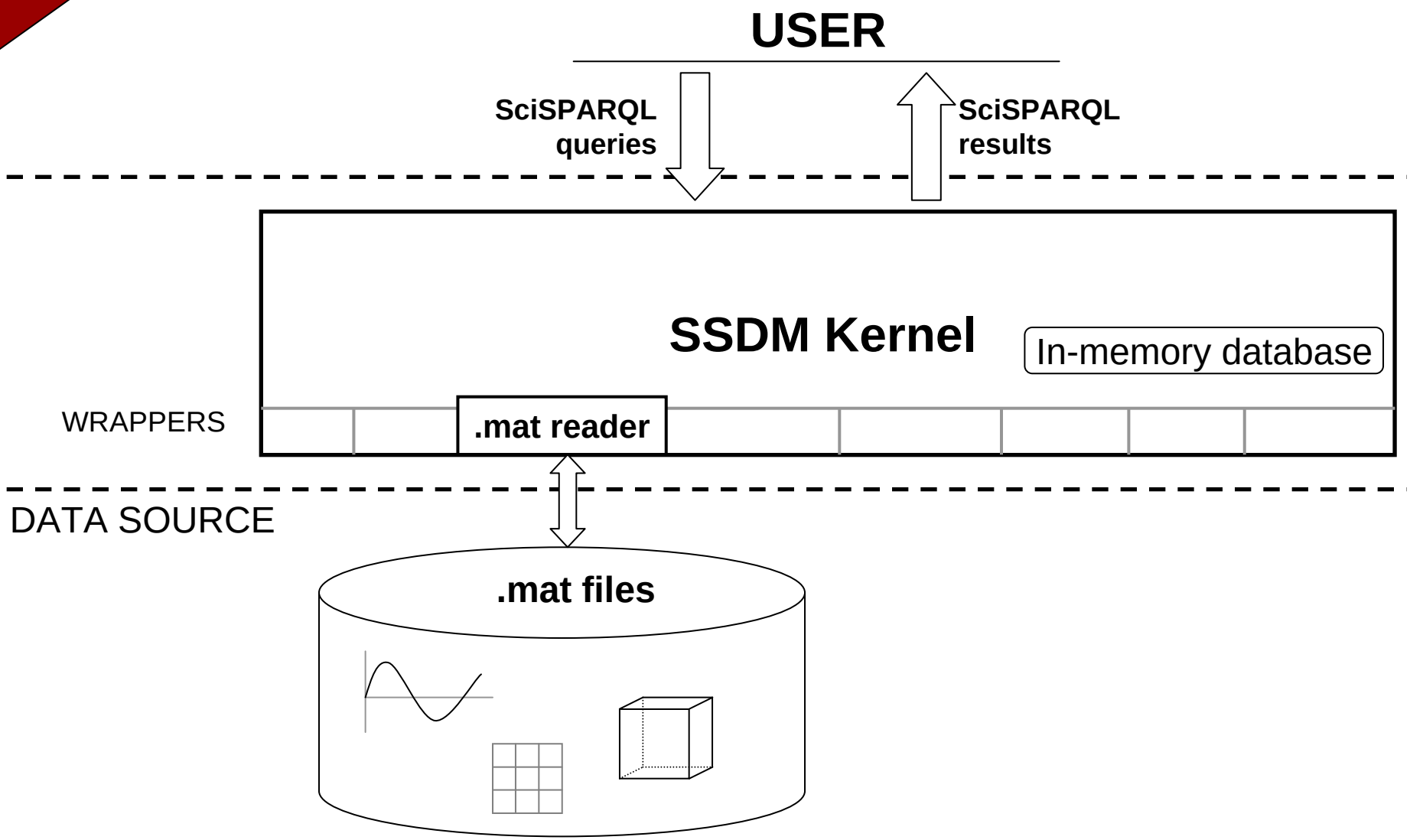
var / task id	k_1	k_a	k_d	k_4	realization	result
1	32.159	79.279	782750669.857	53.286	1	
2	19.151	39.044	300035857.676	73.445	1	

RDF views over external storage systems

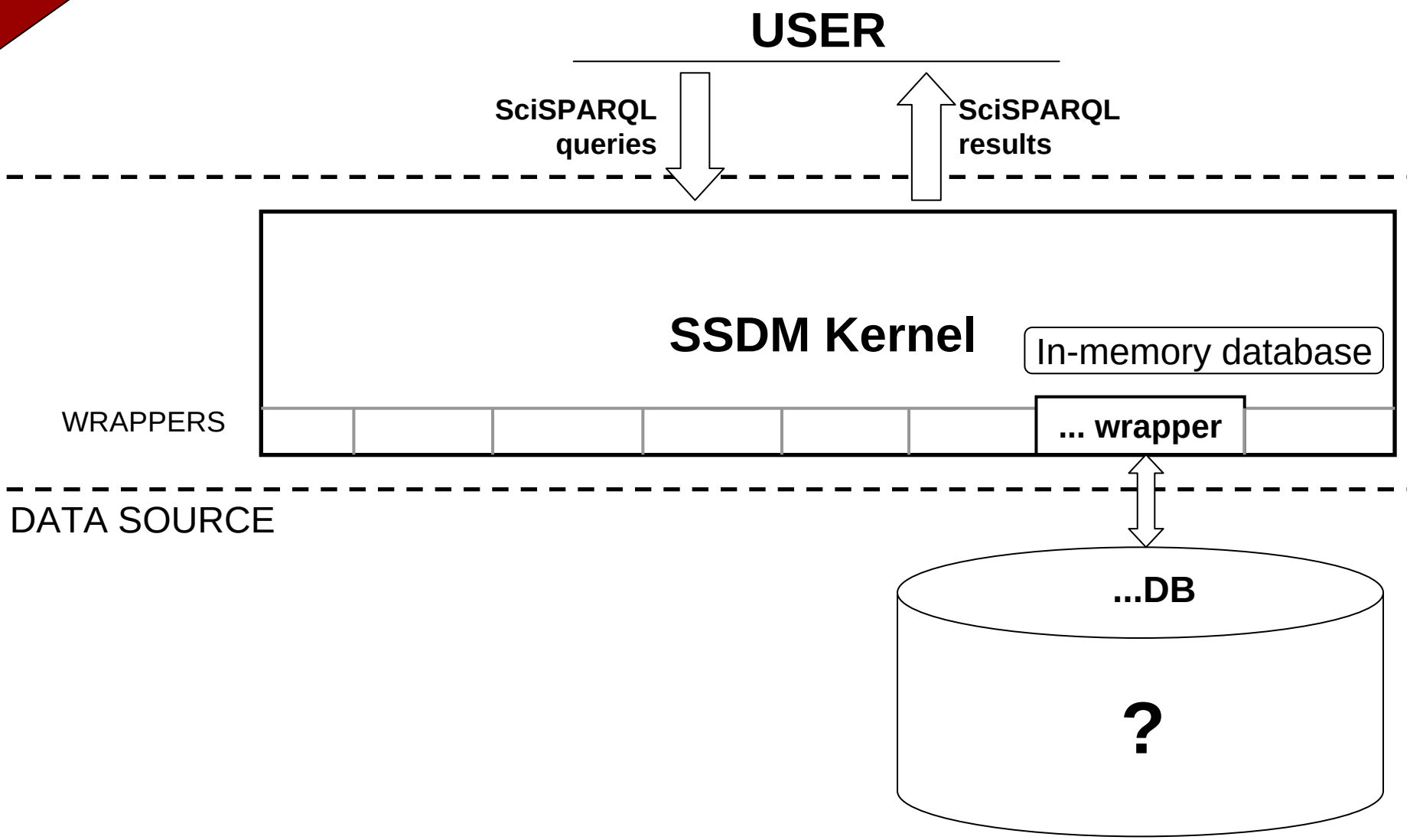


*** Silvia Stefanova and Tore Risch:**
Scalable Long-term Preservation of Relational Data through SPRQL Queries,
submitted to Semantic Web Journal, 2013.

RDF views over external storage systems



RDF views over external storage systems



UPPSALA UNIVERSITET

- Introduction
- SciSPARQL overview
- Evaluation
- RDF views over external storage systems
- Related approaches
- Summary

Related approaches

Databases

- High-level metadata descriptions (schemas)
- Scalable data representation
- High-level query languages

RDF

- Designed for metadata in general
- Voluntary schema
- Weak support numeric applications

Files and programs

- No explicit metadata
- Many storage formats and APIs
- Numerical libraries maintained since 1960:s
- Extensively used in scientific computing

UPPSALA UNIVERSITET

Related approaches

Databases

- High-level metadata descriptions (schemas)
- Scalable data representation
- High-level query languages

RDF

- Designed for metadata in general
- Voluntary schema
- Weak support numeric applications

Files and programs

- No explicit metadata
- Many storage formats and APIs
- Numerical libraries maintained since 1960:s
- Extensively used in scientific computing

SSDM and SciSPARQL

- Full database support
- Flexibility of RDF
- Reuse of existing libraries

Related query languages

- Extending RDBMS with array semantics
  - AQuery [Lerner & Shasha, 2003]
  - SciQL [Kersten et.al, 2011]
- Storing arrays as BLOBs
  - RasQL [Furtado & Baumann, 1999]
  - UDFs in T-SQL [Dobos et.al., 2011]
- Specialized array databases and languages
  - AQL [Libkin et.al. 1996]
  - SciDB [Cudre-Mauroux et.al. 2009]
- Foreign Functions in SPARQL
  - SESAME
  - CORESE

UPPSALA UNIVERSITET

- Introduction
- SciSPARQL overview
- Evaluation
- RDF views over external storage systems
- Related approaches
- Summary

UPPSALA UNIVERSITET

Summary

SSDM (SciSPARQL Database Manager) provides

- Efficient storage of RDF with arrays
- Back-end relational database storage and various data file formats
- Access to external databases

SciSPARQL provides

- support of numeric multidimensional arrays and operations
- extensibility with foreign functions in C, Java, Python, and MATLAB

UPPSALA UNIVERSITET

The software, documentation, and examples are available at

http://www.it.uu.se/research/group/udbl/SciSPARQL

This work was supported by

eSSENCE THE e-SCIENCE COLLABORATION

UPPSALA UNIVERSITET · UMEÅ UNIVERSITET · LUND UNIVERSITY