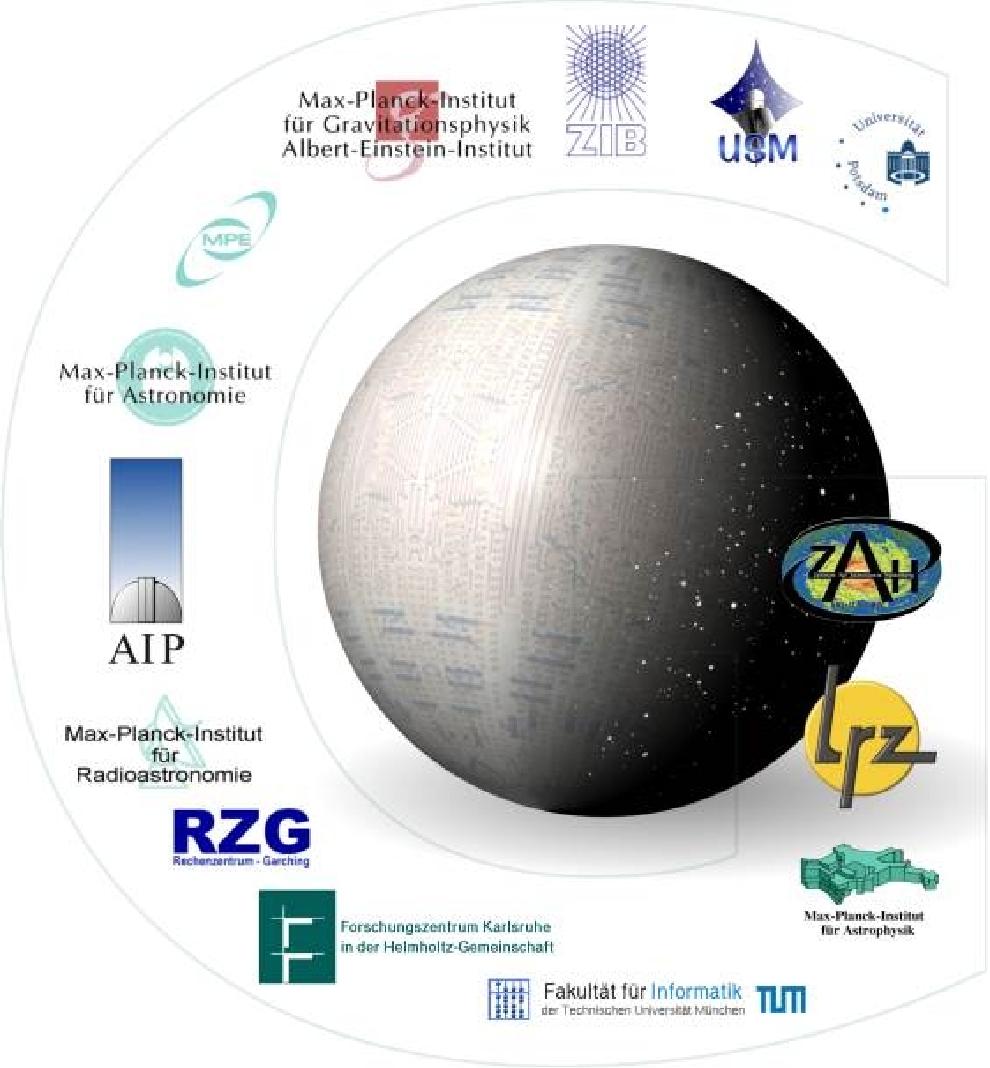
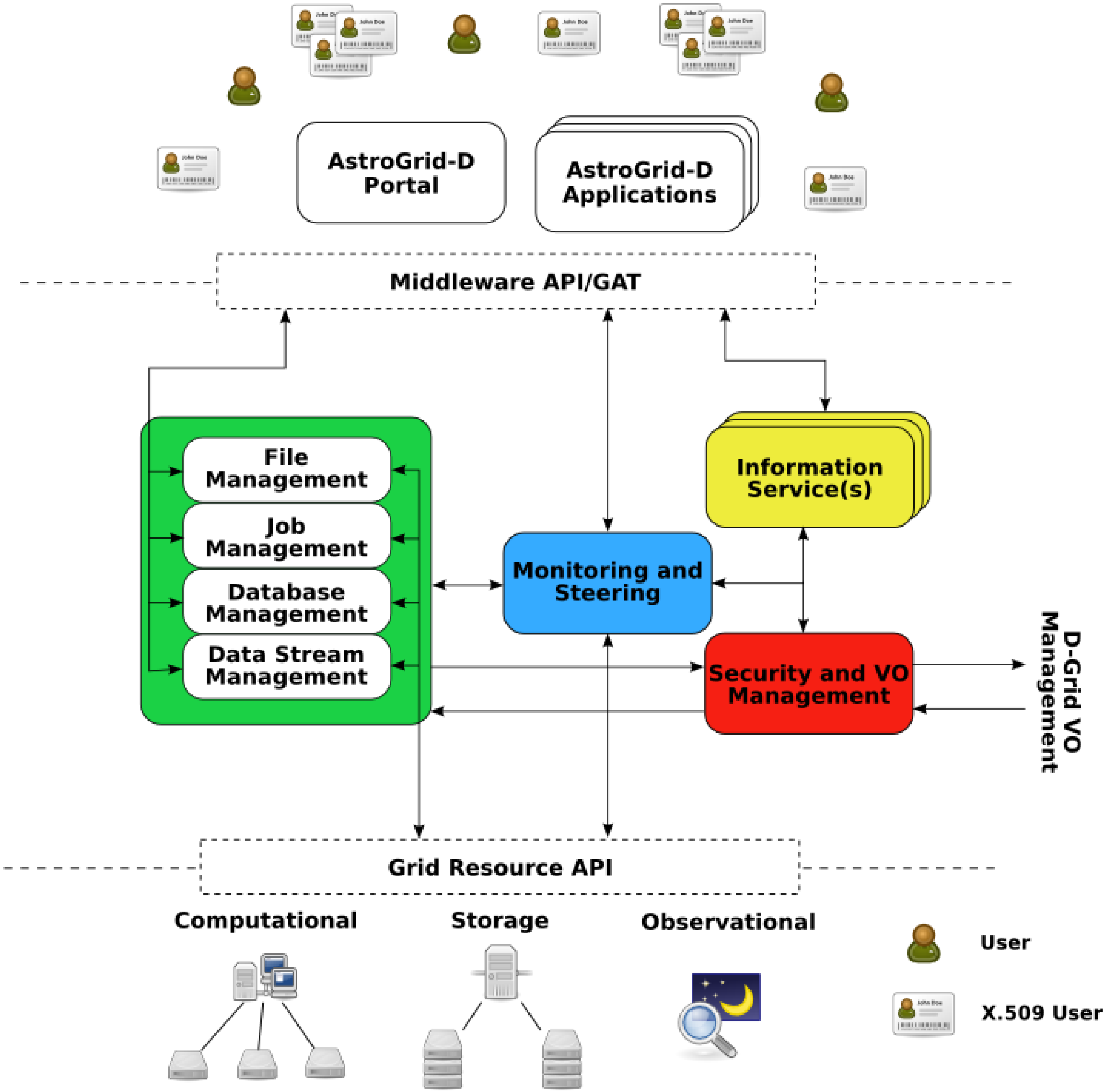
The AstroGrid-D Information Service

Mikael Högqvist
hoegqvist@zib.de



Introduction

- Roadmap
  - AstroGrid-D Introduction
  - Requirements and Approach
  - Applications



AstroGrid-D Portal

AstroGrid-D Applications

Middleware API/GAT

File Management

Job Management

Database Management

Data Stream Management

Information Service(s)

Monitoring and Steering

Security and VO Management

D-Grid VO Management

Grid Resource API

Computational

Storage

Observational

User

X.509 User

Motivation

- Why do we need metadata?
  - Describe/organize resources to find them later
- Answer questions like:
  - Where is the data-set used for the creation of the data in this graph?
  - Why did my last grid-job fail?
  - Find any existing data-set produced with my application using these parameters.
  - What resources match these job requirements?

AstroGrid-D Metadata

- Virtualized Resources
  - GLUE schema
  - RTML
- Activity of grid services (jobs, files, data stream)
- Application-specific metadata (job history, simulation progress, ...)
- Scientific metadata (domain-specific description of data sets, provenance, ...)

Requirements

- Extensible/flexible data model
- Integration of different metadata sources
- Easy to extract and export metadata
- Restrict access for unauthorized users
- Handle different metadata characteristics

Approach

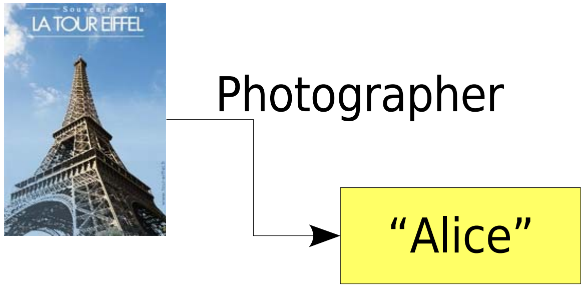
- Uniform interface for metadata management and query
- Common information model using RDF
  - Schema != database structure
  - An RDF entry is a triple (subject, predicate, object)
  - A set of triples form a graph
- Queries are defined using SPARQL
  - Query language for RDF
  - Queries over multiple databases possible

Approach

- Uniform interface for metadata management and query
- Common information model using RDF
  - Schema != database structure
  - An RDF entry is a triple (subject, predicate, object)
  - A set of triples form a graph
- Queries are defined using SPARQL
  - Query language for RDF
  - Queries over multiple databases possible

RDF Example

“A picture of the Eiffel tower has a photographer with value Alice”



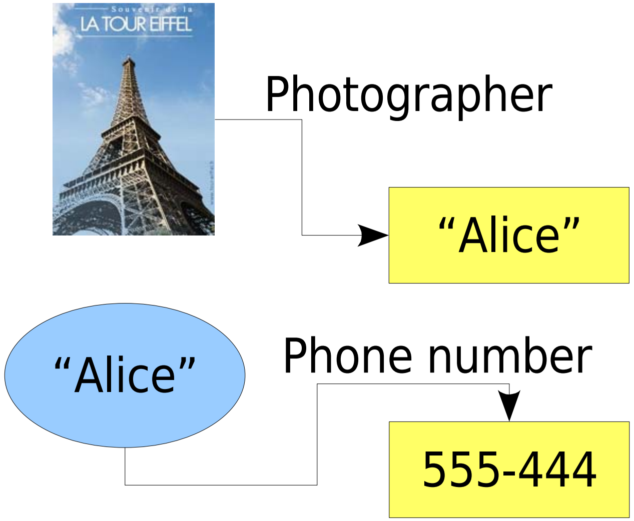
Photographer

“Alice”

RDF Example

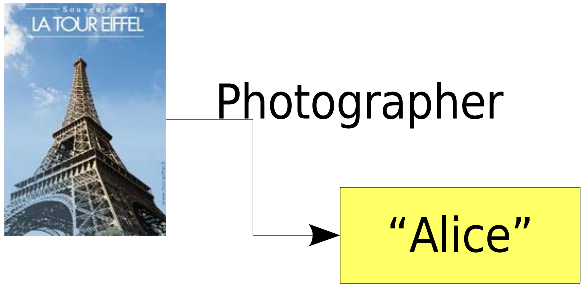
“A picture of the Eiffel tower has a photographer with value Alice”

“Alice has a phone number with value 555-444”



RDF Example

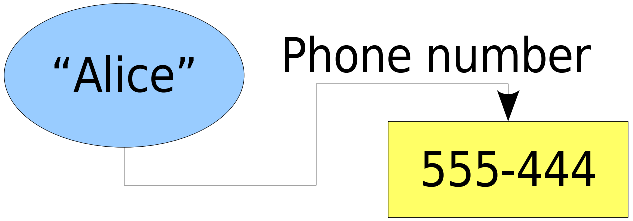
“A picture of the Eiffel tower has a photographer with value Alice”



Photographer

“Alice”

“Alice has a phone number with value 555-444”

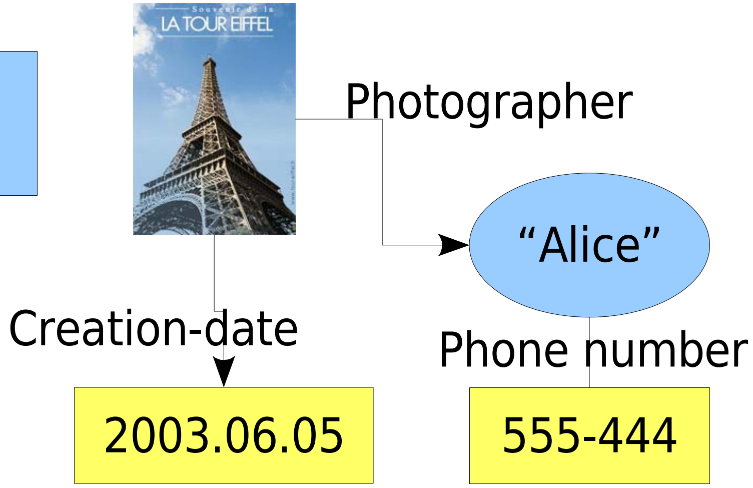


“Alice”

Phone number

555-444

“A picture of the Eiffel tower has a creation-date with value 2003.06.05”



Photographer

“Alice”

Creation-date

Phone number

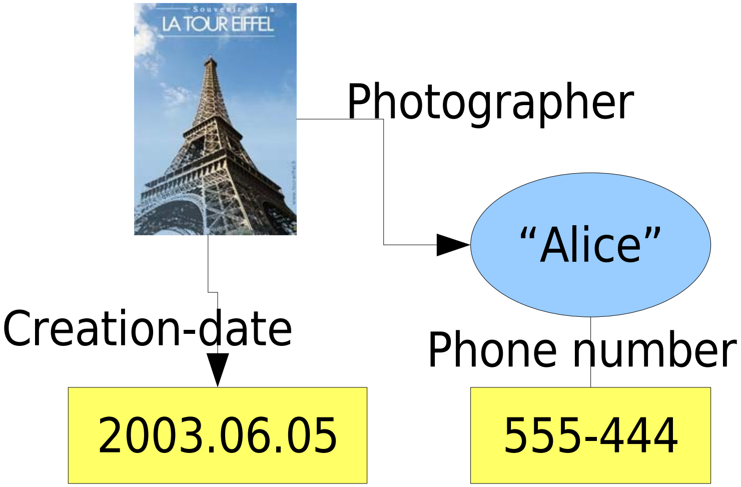
2003.06.05

555-444

Approach

- Uniform interface for metadata management and query
- Common information model using RDF
  - Schema != database structure
  - An RDF entry is a triple (subject, predicate, object)
  - A set of triples form a graph
- Queries are defined using SPARQL
  - Query language for RDF
  - Queries over multiple databases possible

SPARQL Example



Photographer

“Alice”

Creation-date

Phone number

2003.06.05

555-444

“What is the name and phone number of the photographer who took the picture of the Eiffel tower?”

Input graph

```
SELECT ?phone_number ?name WHERE
{“Picture of Eiffel tower” “Photographer” ?name .
  ?name “Phone number” ?phone_number }
```

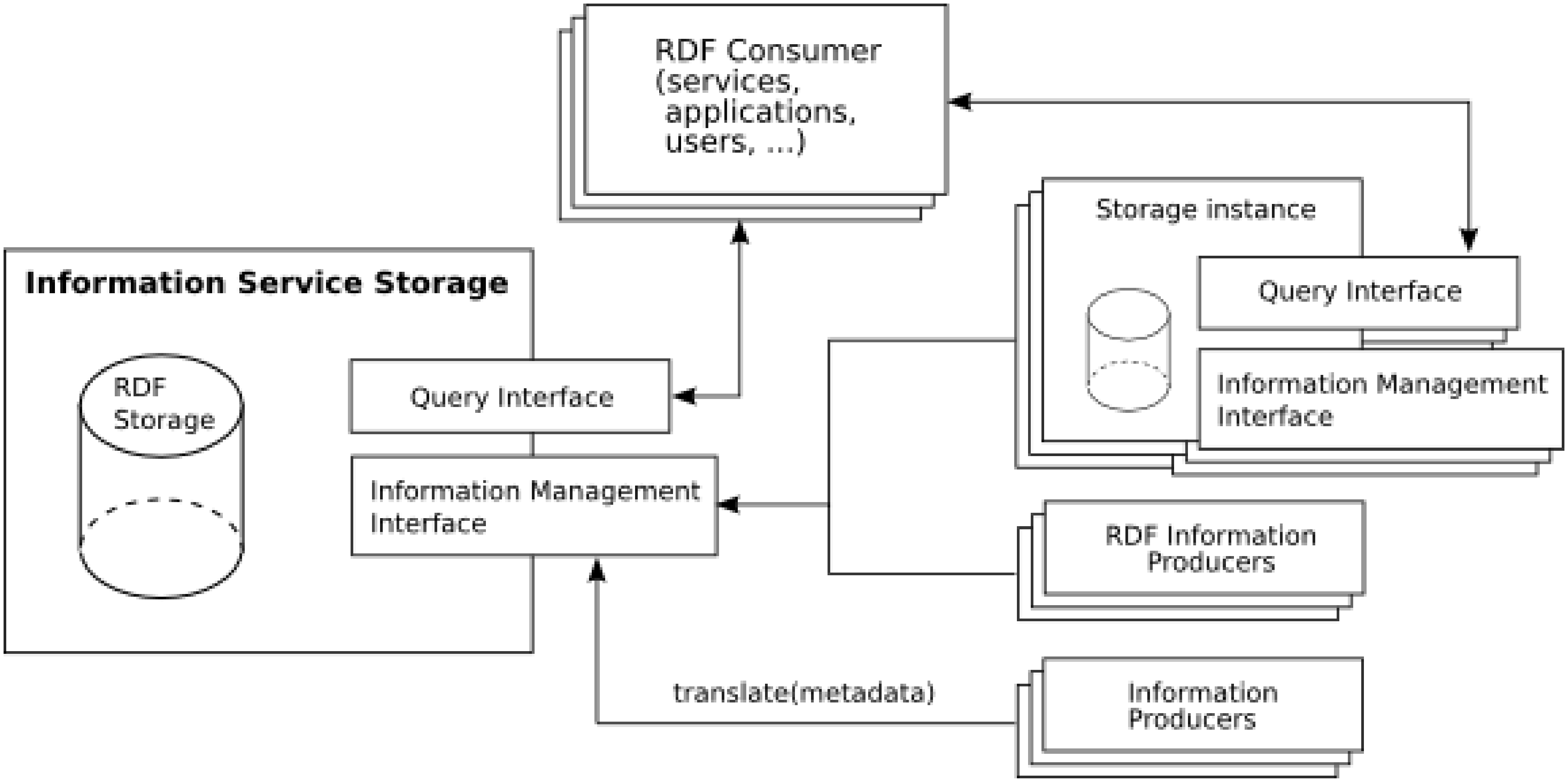
Number	Name
555-444	Alice

Output results

Platform

- RDF-store for metadata management and query
- External interface based on HTTP
- GUI for introspection and query
- X.509 proxy certificates and VOMRS for VO-management
- Use the tools and vocabularies available!

Framework


RDF Consumer (services, applications, users, ...)
Information Service Storage
RDF Storage
Query Interface
Information Management Interface
Storage instance
Query Interface
Information Management Interface
RDF Information Producers
translate(metadata)
Information Producers

Current applications

- Cactus simulations (integration test suite)
- MDS
- Robotic telescopes
- Job submission
- Data stream management
- Demos
  - Resource map
  - Timeline

Conclusions

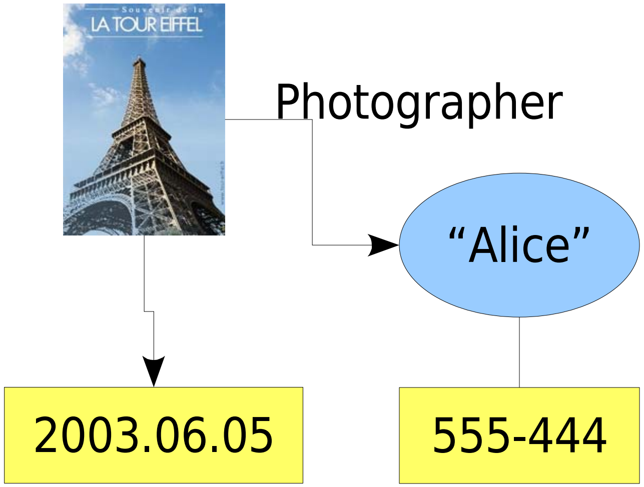
- Heterogeneous environment with many use cases and resource types
- RDF used as a common data model
- SPARQL for queries
- RDF-database for metadata management
  - X.509 proxy certificates
  - HTTP interface

References

- F. Manola and E. Miller. RDF primer. http://www.w3.org/TR/rdf-primer/, February 2004.
- E. Prud’hommeaux and A. Seaborne. SPARQL query language for RDF http://www.w3.org/TR/rdf-sparql-query/, April 2006.
- J. J. Carroll, C. Bizer, P. Hayes and P. Stickler. Named graphs, provenance and trust. In Proc. of WWW 2005.
- F.V. Hessman, C. Pennypacker, E. Romero-Colmenero and G. Tuparev. Telescope networking and user support via remote telescope markup language. In Proc. of SPIE 2004.
- S. Andreozzi, S. Burke, L. Field, S. Fisher, B. Konya, M. Mambelli, J. M. Schopf, M. Viljoen, and A. Wilson. Glue schema specification version 1.2. Technical report, December 2005.
- M. Högqvist and T. Röblitz. AstroGrid-D Metadata Management and Discovery, Deliverable D2.1

Storage concepts

- Context, name of an RDF-graph
- Collection, set of contexts
- Security
  - ACLs and levels
- Garbage collection



Photographer

“Alice”

2003.06.05

555-444

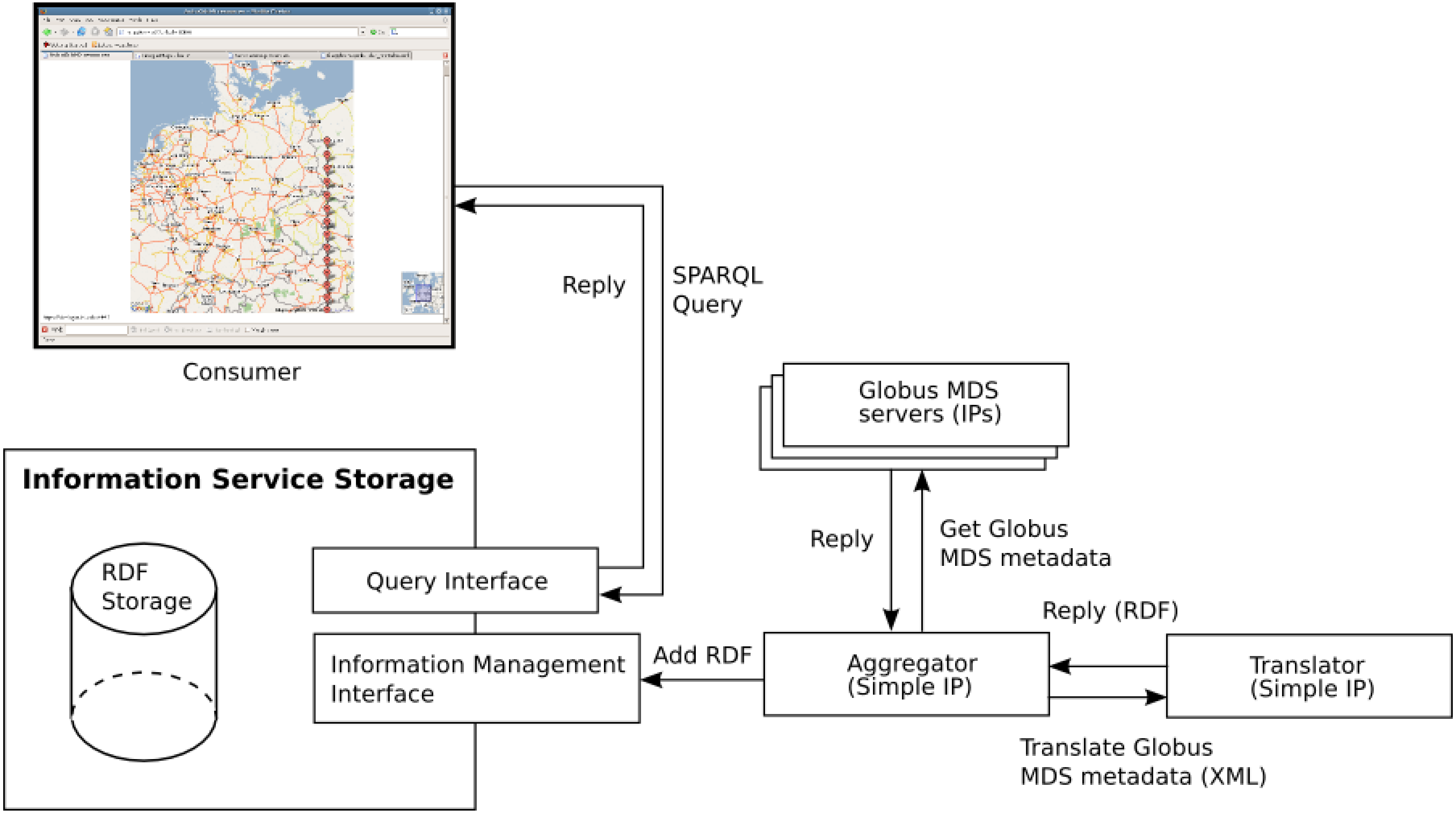
Writing information producers and consumers

- Developing an information producer
  - Analyze application and design a vocabulary
  - Generate metadata
  - Interface with the information service
- Developing a consumer
  - Define the query in SPARQL -> execute query
  - Handle query results

Demo

- Idea: use Google's map API to present grid resources using RDF metadata provided by the information services
- Tools
  - MDS4 WebMDS produces an XML representation of resource information
  - Template language for translating to RDF
  - An RDF store
  - Web service interface to add and query the RDF store

Demo: Component interaction



Consumer

Reply

SPARQL Query

Globus MDS servers (IPs)

Information Service Storage

RDF Storage

Query Interface

Information Management Interface

Reply

Get Globus MDS metadata

Reply (RDF)

Add RDF

Aggregator (Simple IP)

Translator (Simple IP)

Translate Globus MDS metadata (XML)

Demo: SPARQL queries

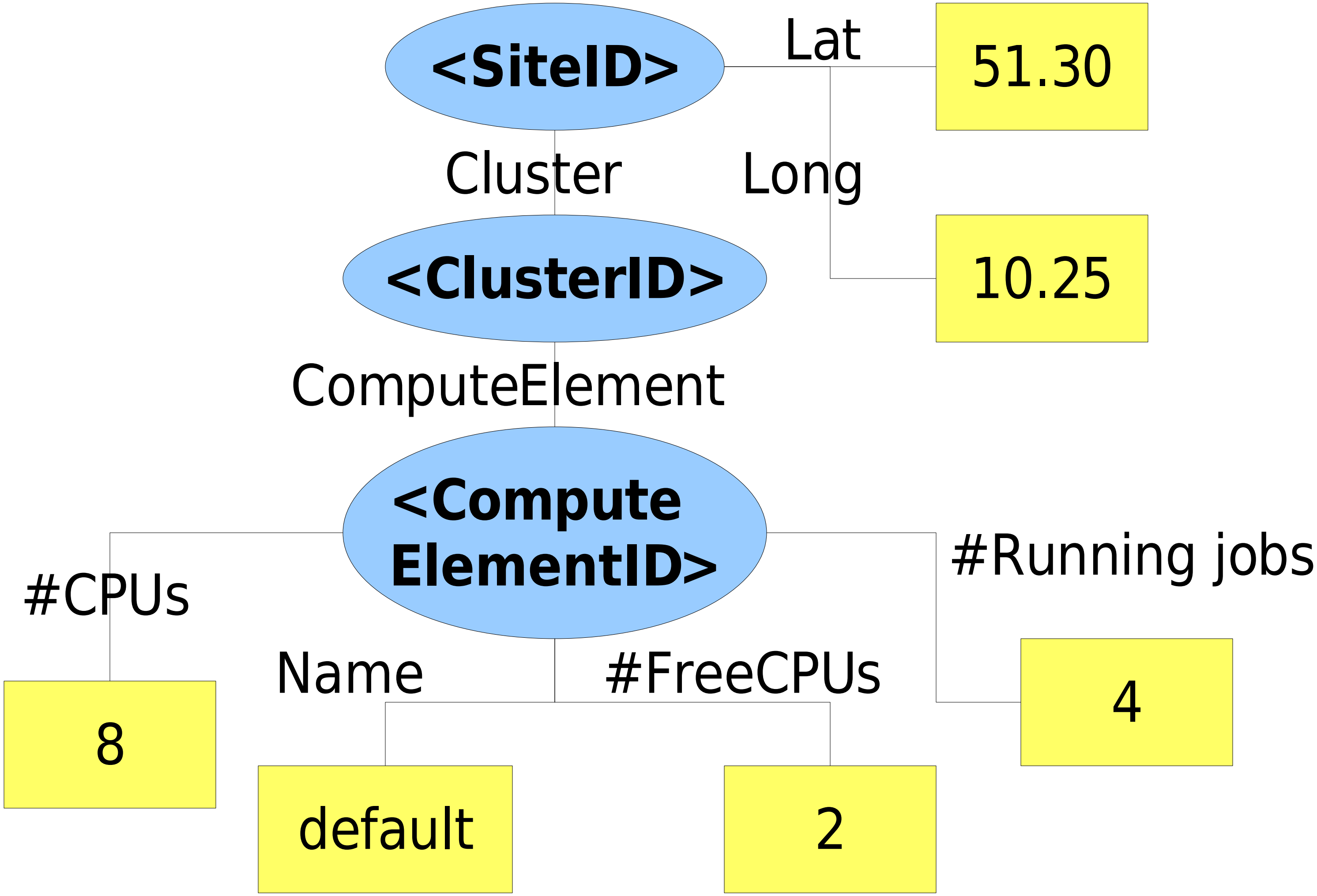
“Get all computing elements from site S”

```
SELECT ?ce WHERE {S “ComputeElement” ?ce}
```

“Get all sites and their longitude and latitude if available”

```
SELECT ?site, ?lat, ?long WHERE
{?site rdf:type “Site” .
 OPTIONAL {?site geo:lat ?lat .
           ?site geo:long ?long }
}
```

Demo: RDF graph (example)


<SiteID>
Lat
51.30
Cluster
Long
<ClusterID>
10.25
ComputeElement
<Compute ElementID>
#Running jobs
#CPUs
Name
#FreeCPUs
4
8
default
2