

XML Data Integration in OGSA Grids

Carmela Comito and Domenico Talia
University of Calabria – Italy

comito@si.deis.unical.it



Outline

❖ Introduction

❖ Data Integration and Grids

❖ The XMAP Data Integration Framework

- Integration Model
- XPath Query Reformulation Algorithm

❖ The Grid Data Integration System

❖ Conclusions

The Problem...(1)

- Grid applications can access distributed heterogeneous data sources
 - Managed by different software system
 - Accessible through different protocols and interfaces
 - Modeled through different data models
- Data Sources are autonomous and highly dynamic
- The case for high-level services:
 - Assist users to access several databases
 - Exploit the variety and dynamic nature of resources offered by the Grid

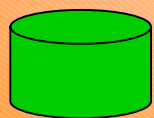
The Problem...(2)

...I'd like to find all the places holding works of art of Impressionists

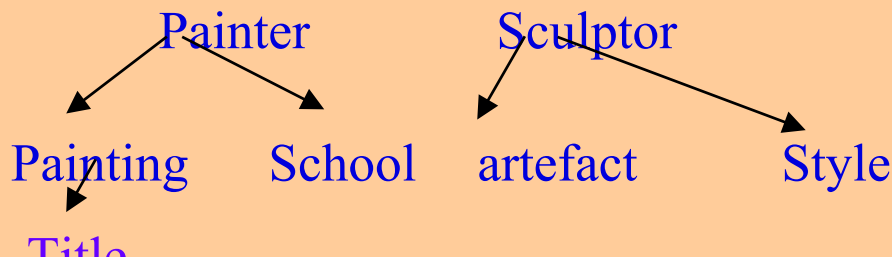
NG

Artist

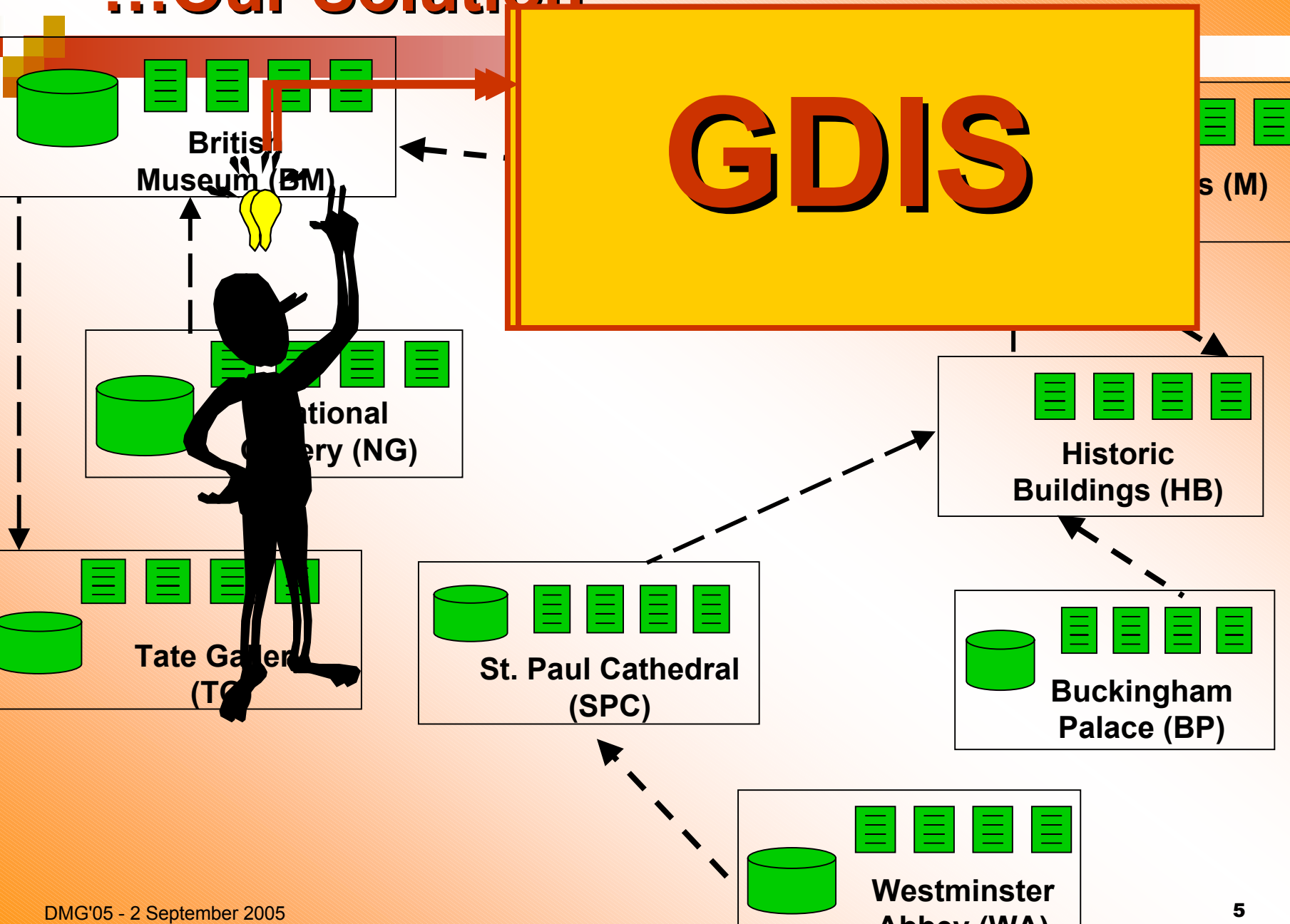
- ❖ Data Integration is a key issue for exploiting the availability of large, heterogeneous, distributed data volumes on Grids
- ❖ Integration formalisms can benefit from OGSA-based Grids
 - ❑ Dynamic discovery, allocation, access and use of resources



**British
Museum (BM)**



...Our Solution



Goals

- Develop a decentralized framework for integrating heterogeneous XML data source
 - Addressing challenges arisen from autonomous, dynamic data sources across unpredictable network
 - Meeting the requirements of scalability, robustness, autonomy
- Deploy the integration framework in a service-based Grid architecture
 - Expose data integration utilities as Grid services
- Exploit the middleware provided by OGSA-DAI, OGSA-DQP and Globus Toolkit


Data Integration and Grids

A data integration system provides a uniform query interface across autonomous, heterogeneous networked or local data sources

- ❑ Federated Database Management Systems (FDBMSs)
- ❑ Mediator/Wrapper based Integration System

In the Grid

- ❑ Multiple, autonomous, unpredictable sites
- ❑ Huge, highly dynamic, data volumes
- ❑ Heterogeneity and Distribution of data resources
- ❑ Sites both clients and servers



Traditional approaches to data integration are not suitable in Grid settings

Challenges of Grid-based Data Integration Architectures

The Grid raises new challenges in data integration systems:

No need for a central mediated schema (*Decentralization*)

Ability to map data as is most convenient (*Flexibility, Dynamism*)

Wide-scale, ad-hoc nature (*Scalability*)

Queries are posed using the node's schema. Answers come from anywhere in the system (*Sharing and Cooperation*)

Peer-to-Peer and Grids

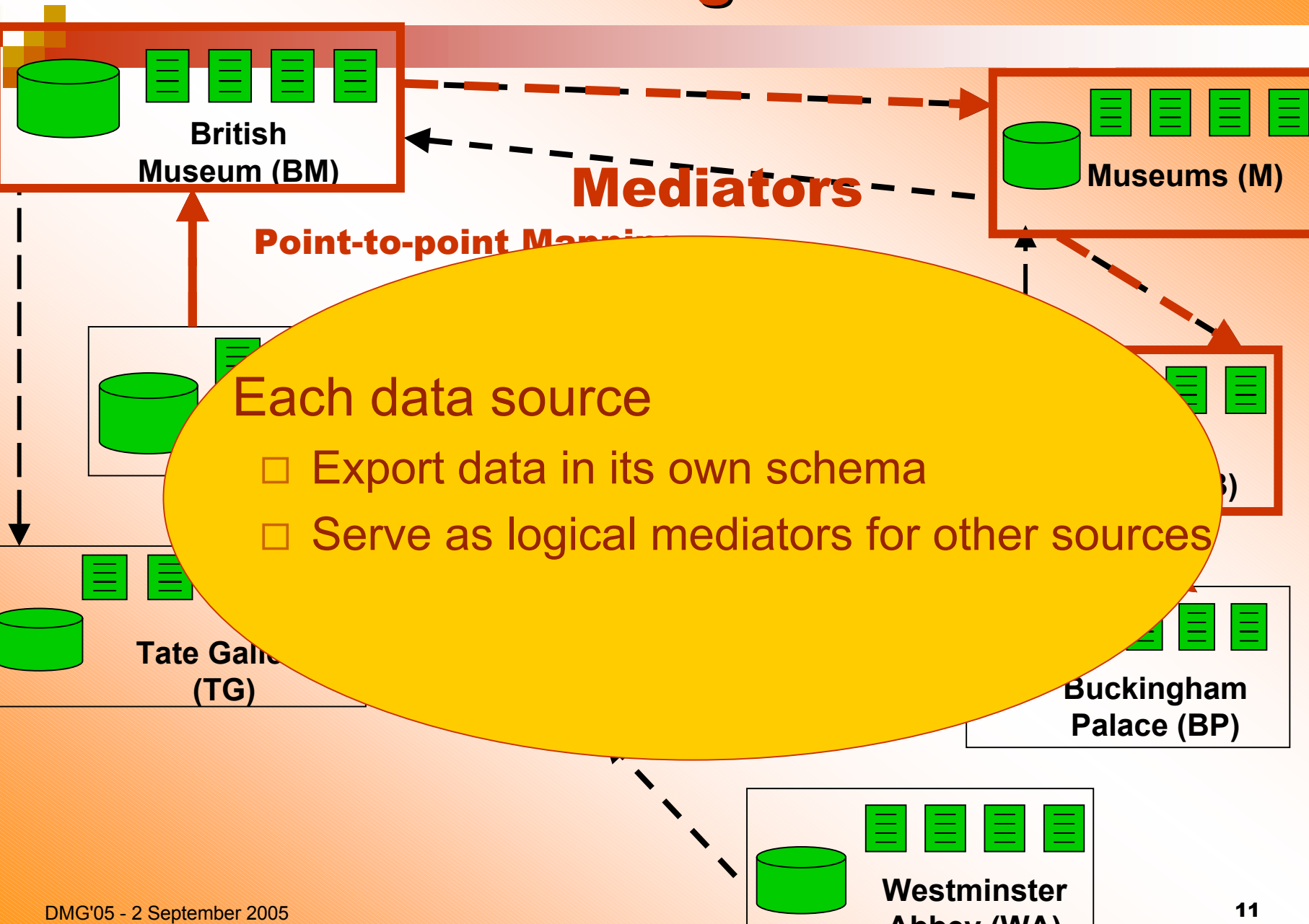
- ▶ Recent works on data management in peer-to-peer systems
 - Lacks a global schema
 - Each peer represents an autonomous information system
 - Semantic mappings are established directly among peers
- ▶ Peer-to-peer based data management architectures present similar features with respect to Grid-based ones
 - OGSA Grids can provide a suitable and reliable infrastructure for P2P systems
 - P2P architectures address issues and problems common to several Grid applications
- ▶ The proposed integration model is inspired from recent approaches in P2P data integration

XMAP: XML Data Integration Framework

A decentralized network of semantically related XML data sources

- A set of distributed, heterogeneous, autonomous XML data sources
 - Different data sources have their own schema
 - Mapping is a key issue to any data sharing architecture
- XMAP integration model is based on *schema mappings*
- Mapping specification is flexible and scalable not resorting to any hierarchical structure
 - Each source schema is directly connected to only a small number of other schemas (*point-to-point mapping*)
 - Each source schema is reachable from all other schemas belonging to its transitive closure (*transitive mapping*)

XMAP: XML Data Integration Framework



Schema Mapping in XMAP

- Schema mapping in XMAP associates paths in different schemas (*path-to-path mapping*)
- A Mapping M over a source schema S is a set of mapping rules $R^M = \{R^M_1, R^M_2, \dots, R^M_k\}$
 - A mapping rule R^M_i relates a pair of schemas by associating paths on the basis of mappings cardinality constraints
- Mapping rules are specified in XML documents called **XMAP documents**
 - Each source schema is associated to an XMAP document containing all the mapping rules related to it.

XMAP Document Structure

```
<schema targetNamespace="http://XMAP/XMAPDocument"
  xmlns="http://www.w3.org/2001/XMLSchema" ...>
  <element name="Mapping">
    <complexType>
      <sequence>
        <element name="sourceSchema" type="string" minOccurs="1"
maxOccurs="1"/>
        <element name="Rule" minOccurs="1">
          <complexType>
            <sequence>
              <attribute name="Cardinality" type="string" minOccurs="1"
maxOccurs="1"/>
              <element name="sourcePath" type="string" minOccurs="1"/>
              <element name="destSchema" type="string" minOccurs="1"
maxOccurs="1"/>
              <element name="destPath" type="string" minOccurs="1"/>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </complexType>
  </element>
</schema>
```

XMAP Reformulation Algorithm

- *XMAP Reformulation Algorithm* reformulates an XPath query Q over all the schemas related to Q
 - Query is answered by chaining of mapped sources using the mapping rules defined in *XMAP* documents
 - Direct reformulations of Q by using the mapping of S (point-to-point mapping)
 - Transitive reformulations are obtained by recursively invoking the algorithm over each reformulated query (transitive mappings).

➤ INPUT

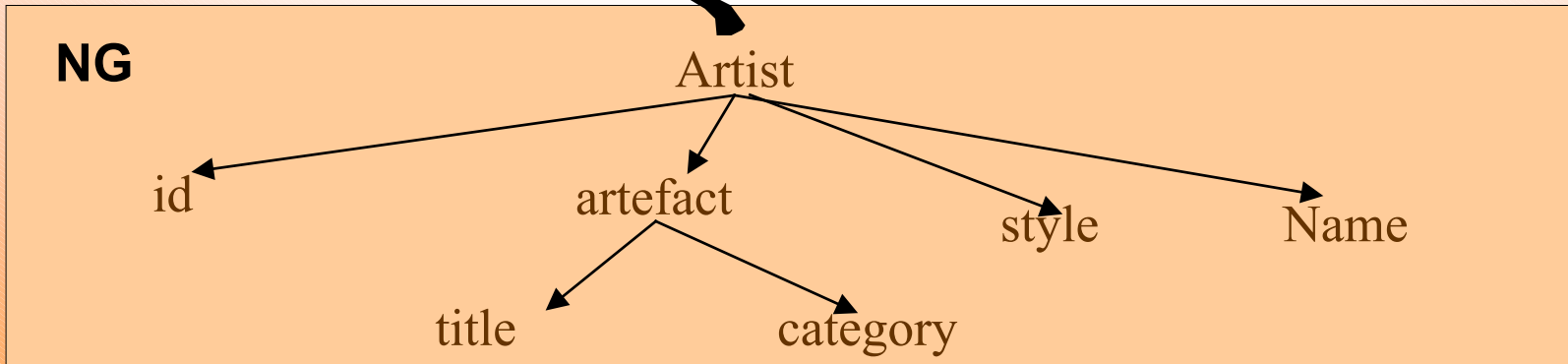
- A query Q over the schema NG
- *XMAP* document associated with NG , $XMAP_{NG}$

➤ OUTPUT

- A set of reformulated queries Q_{R_i}

XMAP Reformulation Example

...I'd like to find all the places holding works of art of Impressionists...



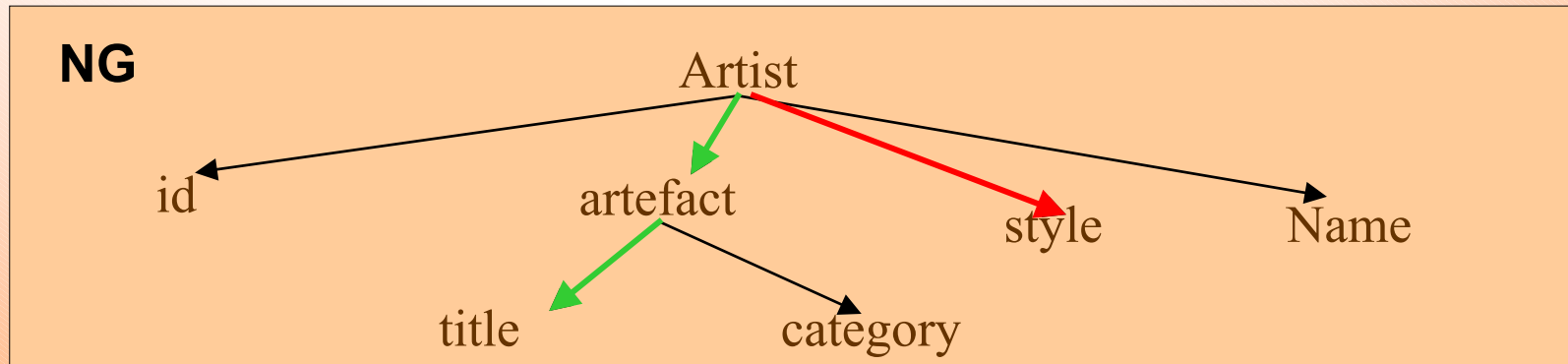
Q=/Artist[style="Impressionism"]/artefact/title

XMAP Reformulation Algorithm - Steps

1. *Identifying the paths in Q .*

XMAP Reformulation Algorithm - Example

Q=/Artist[style="Impressionism"]/artefact/title



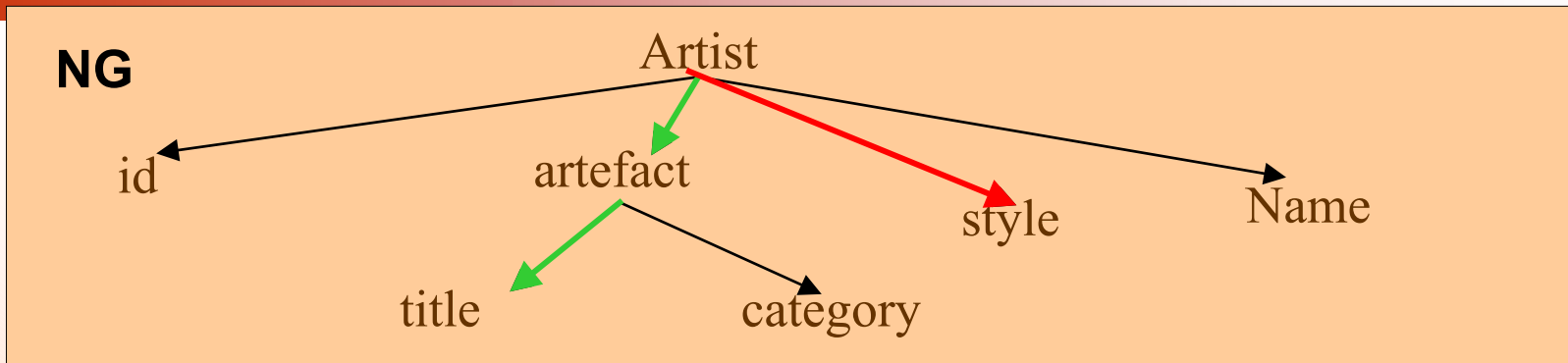
XMAP Reformulation Algorithm - Steps

1. *Identifying the paths in Q .*
2. *Looking for candidate paths in all source schemas related to NG.*

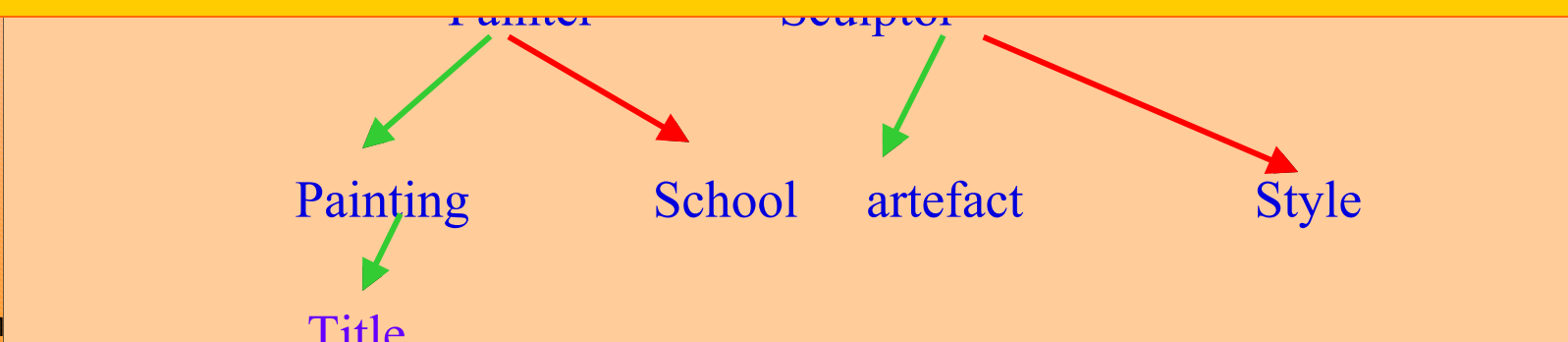
XMAP Document of schema NG

```
<?xml version="1.0" ?>
<XMAP>
<sourceSchema>NG</sourceSchema>
<Rule cardinality="MappingN-1">
<destinationSchema>BM</destinationSchema>
<sourcePath> /artist/first-name </sourcePath>
<sourcePath>/artist/last-name</sourcePath>
<destinationPath> /Info/Name </destinationPath>
</Rule>...
<Rule cardinality="Mapping1-N">
<destinationSchema> BM </destinationSchema>
<sourcePath> /artist/style </sourcePath>
<destinationPath> /Info/Kind/Painter/School </destinationPath>
<destinationPath> /Info/Kind/Sculptor/Style </destinationPath>
</Rule>
<Rule cardinality="Mapping1-N">
<destinationSchema> BM </destinationSchema>
<sourcePath> /artist/artefact/title </sourcePath>
<destinationPath> /Info/Kind/Painter/Painting/Title</destinationPath>
<destinationPath> /Info/Kind/Sculptor/Artefact</destinationPath>
</Rule>...
</XMAP>
```

Example



```
<Rule cardinality="Mapping1-N">  
<destinationSchema> BM </destinationSchema>  
<sourcePath> /artist/artefact/title </sourcePath>  
<destinationPath> /Info/Kind/Painter/Painting/Title</destinationPath>  
<destinationPath> /Info/Kind/Sculptor/artefact</destinationPath>  
</Rule>
```



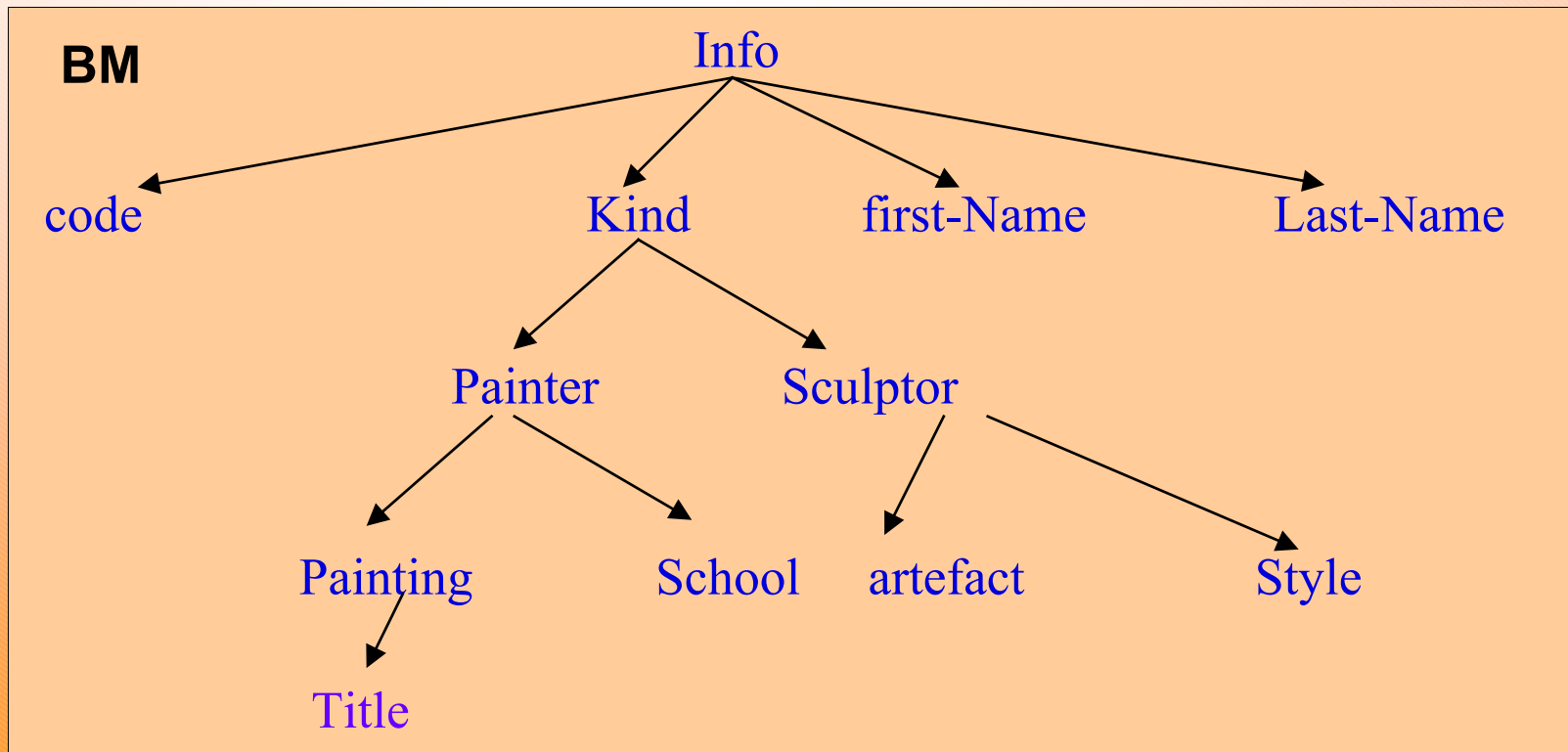
XMAP Reformulation Algorithm - Steps

1. *Identifying the paths in Q*
2. *Looking for candidate paths in all source schemas related to NG.*
3. *Pruning of Candidate schemas*
4. *Constructing Reformulated Queries*

XMAP Reformulation Algorithm - Example

$Q_{R1} = /ArtistInfo/category/painter[school="Impressionism"]/painting/title$

$Q_{R2} = /ArtistInfo/category/sculptor[style="Impressionism"]/artefact$



XMAP Reformulation Algorithm - Steps

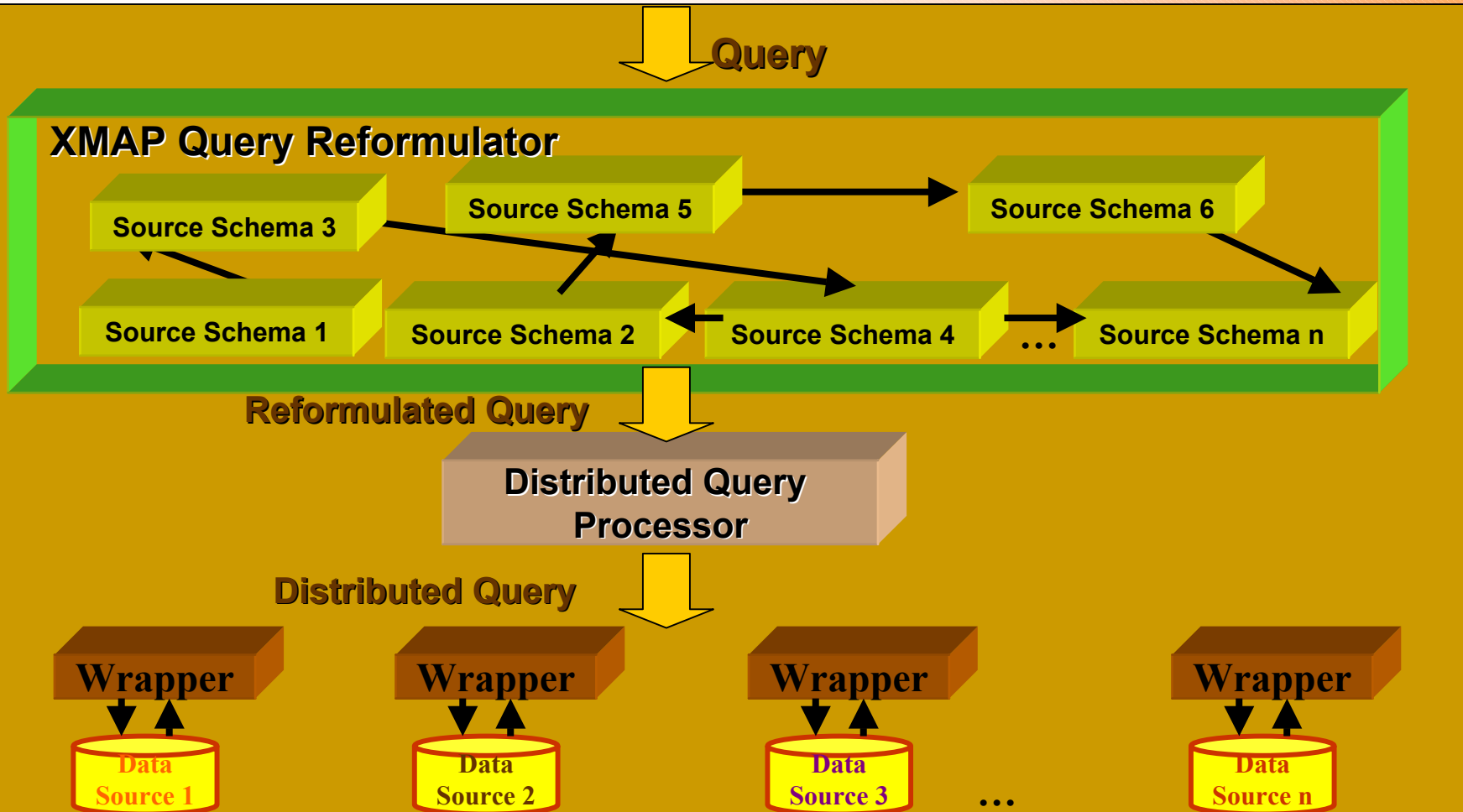
1. *Identifying the paths in Q*
2. *Looking for candidate paths in all source schemas related to NG.*
3. *Pruning of Candidate schemas.*
4. *Constructing Reformulated Queries*
5. *Recursive invocation of the algorithm*

The *Grid Data Integration System*

The Grid Data Integration System (GDIS) is a service-based architecture for data integration on Grid-enabled databases

- Offers a wrapper/mediator-based approach to integrate data sources
 - Adopts the XMAP decentralized mediator approach to handle semantic heterogeneity over data sources
 - Syntactic heterogeneity is hidden behind OGSA-DAI wrappers
- Exposes data integration utilities as Grid Data Services
 - Mapping Specifications
 - XMAP Reformulation Algorithm

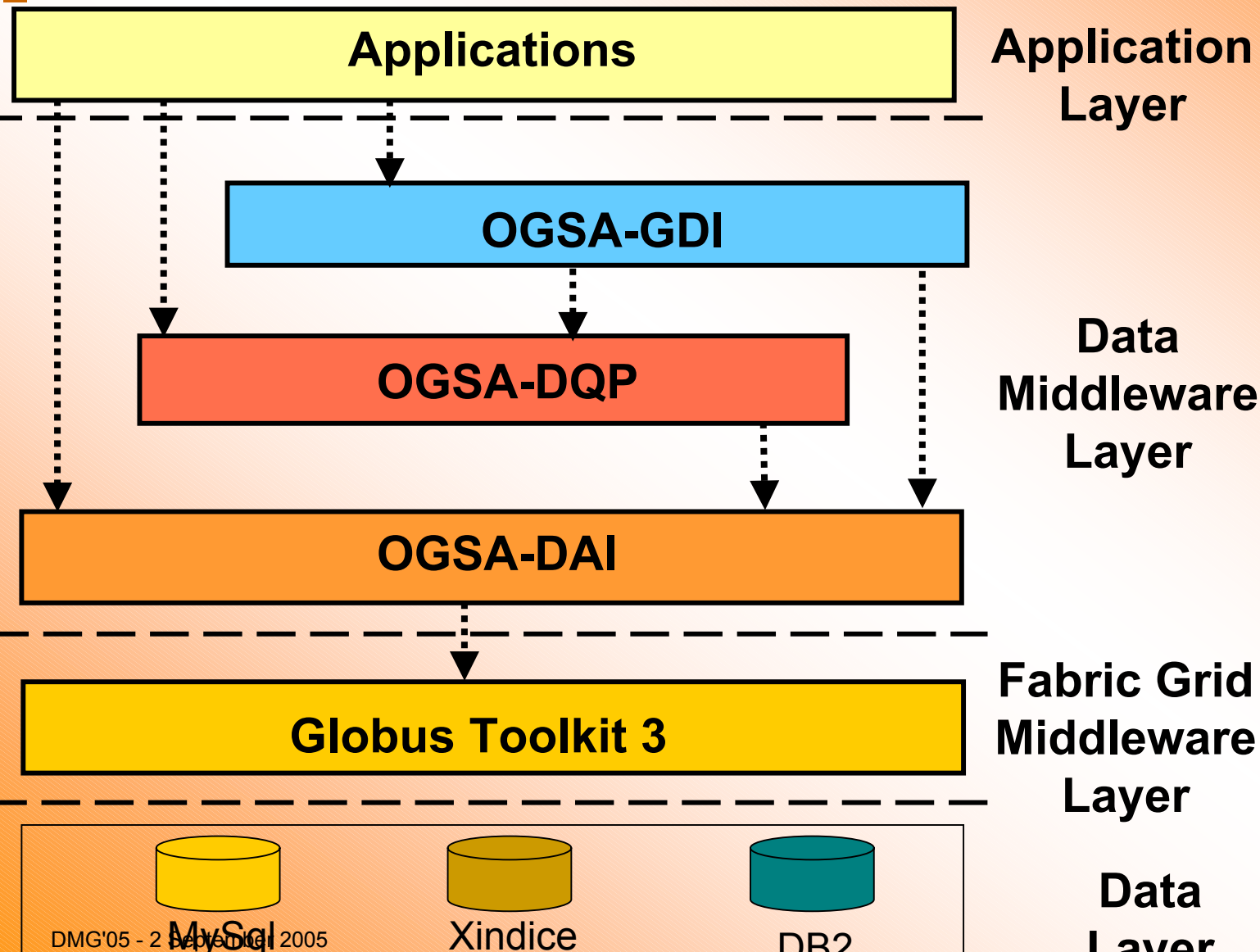
Wrapper/Mediator Approach in GDIS



GDIS Logical Model

- ❖ A set of Grid nodes
- ❖ Each node can
 - ❑ Provide Data sources (*Data Provider*)
 - ❑ Provide Schemas (*Mediator*)
 - ❑ Expose Semantic Mappings
 - ❑ Formulate queries (*Client*)
- ❖ Challenges of a wrapper/mediator-based integration system
 - ❑ Processing nodes
 - ❑ Execution nodes
 - ❑ Data integration nodes
 - ❑ Wrapper nodes

GDIS Layered Architecture



Conclusions

- ❖ XMAP proposes a decentralized solution to address data heterogeneity among XML databases
 - ❑ Integration approach based on flexible and scalable semantic connections among small set of database schemas
 - ❑ XMAP is deployed in GDIS, a service-based Grid architecture
- ❖ The XMAP framework has been recently implemented using Java 1.4 and integrated in the GDIS system using OGSA-DAI 5.0 and Globus Toolkit 3.2.1
- ❖ Future directions
 - ❑ PARIS: using XMAP for reformulating queries in a P2P architecture
 - ❑ Embed the XMAP framework within the OGSA-DQP engine