Content and Storage Management in gCube

Diligent

From Digital Objects to Content across eInfrastructures

Pasquale Pagano
CNR –ISTI

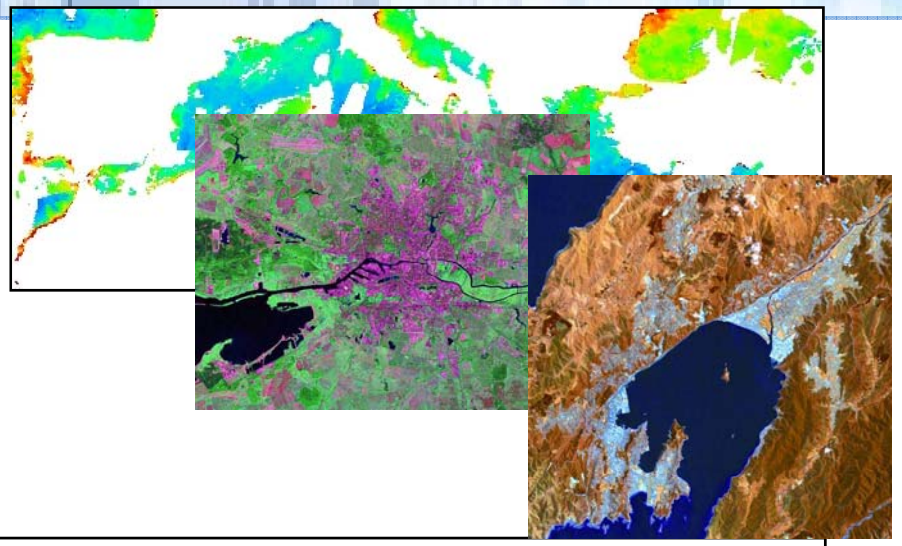
on behalf of

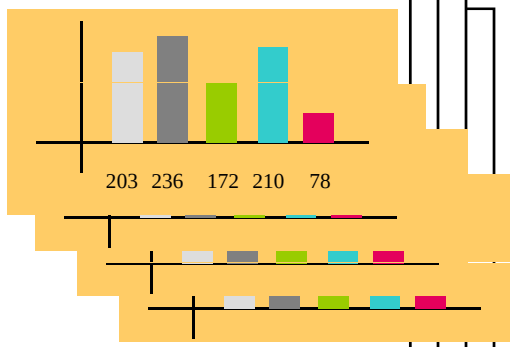
Heiko Schuldt
Dept. of Computer Science
University of Basel
Switzerland

UNI BASEL

Content & Storage Management: Challenges

- Store large volumes of digital content in the Grid
  - But there is much more:
    - Metadata for each object
    - Automatically extracted features per object (e.g., color histo-grams for images)
    - Storage properties (e.g., size, etc.)
    - … and all that highly inter-connected






```
<DIMAP_DOCUMENT>
 <DATASET>MER_RR__2P</DATASET>
 <INSTRUMENT>MER</INSTRUMENT>
 <RESOLUTION_TYPE>RR</RESOLUTION_TYPE>
 <PRODUCT_LEVEL>2P</PRODUCT_LEVEL>
 <PRODUCT_NAME>MERIS</PRODUCT_NAME>
 <BAND_USED>___ALGAL_1</BAND_USED>
 <BAND_USED_NORM>ALGAL 1</BAND_USED_NORM>
 <START_DATE>2005-08-01</START_DATE>
 <END_DATE>2005-08-07</END_DATE>
 <LONMIN_INT>17000</LONMIN_INT>
 <LATMIN_INT>12000</LATMIN_INT>
 <LONMAX_INT>22000</LONMAX_INT>
 <LATMAX_INT>13500</LATMAX_INT>
 <COVER_REGIONS>World</COVER_REGIONS>
 <OVERLAP_REGIONS> World Europe Bigger_Europe Mediterranean
                   Iberia North_Atlantic Africa North_Africa Middle_East
 </OVERLAP_REGIONS>
 <DATA_FILE_FORMAT>ENVI</DATA_FILE_FORMAT>
 ...
</DIMAP_DOCUMENT>
```

Introduction

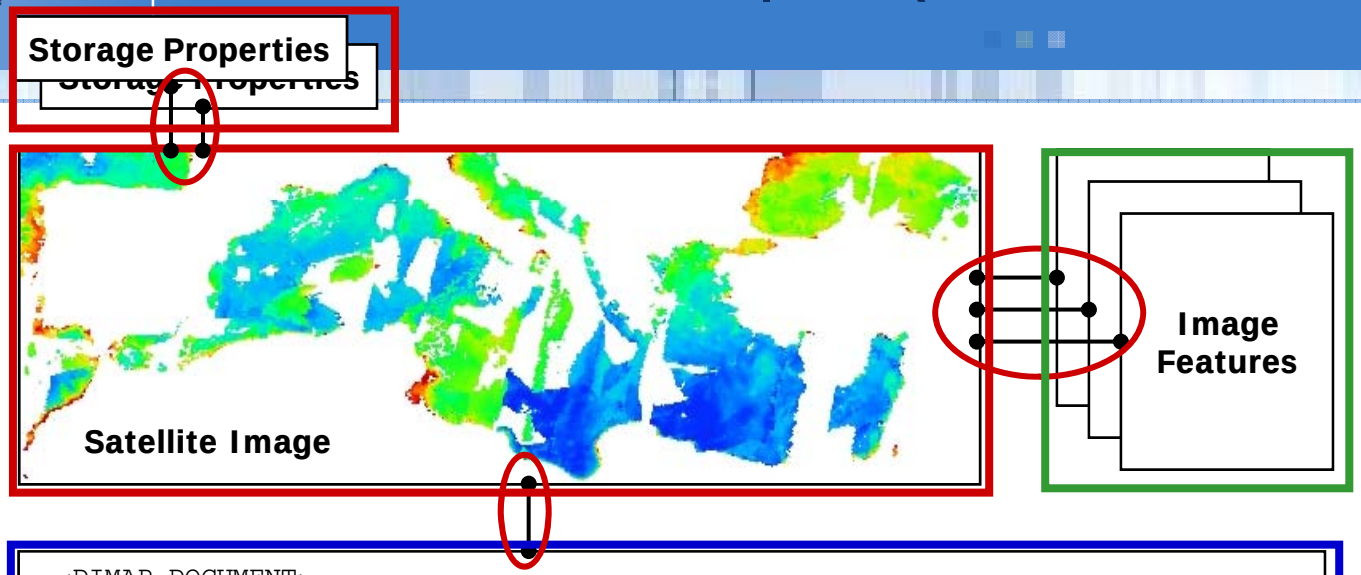
- gCube Data Management provides means for
  - Persistently storing and physically structuring of content.
  - Logical grouping of content in collections – independent of physical locations.
  - Logical sharing of content among several collections.
  - Complex content consisting of several parts and having multiple representations.
  - Replication and partition of content.
  - Subscription and notification
  - Populating VREs by importing/linking pre-existing content.

- gCube exploits Grid technology file-system-like functionalities to manage content storage

Content example (Earth Observation)

Diligent

Content & Storage Management



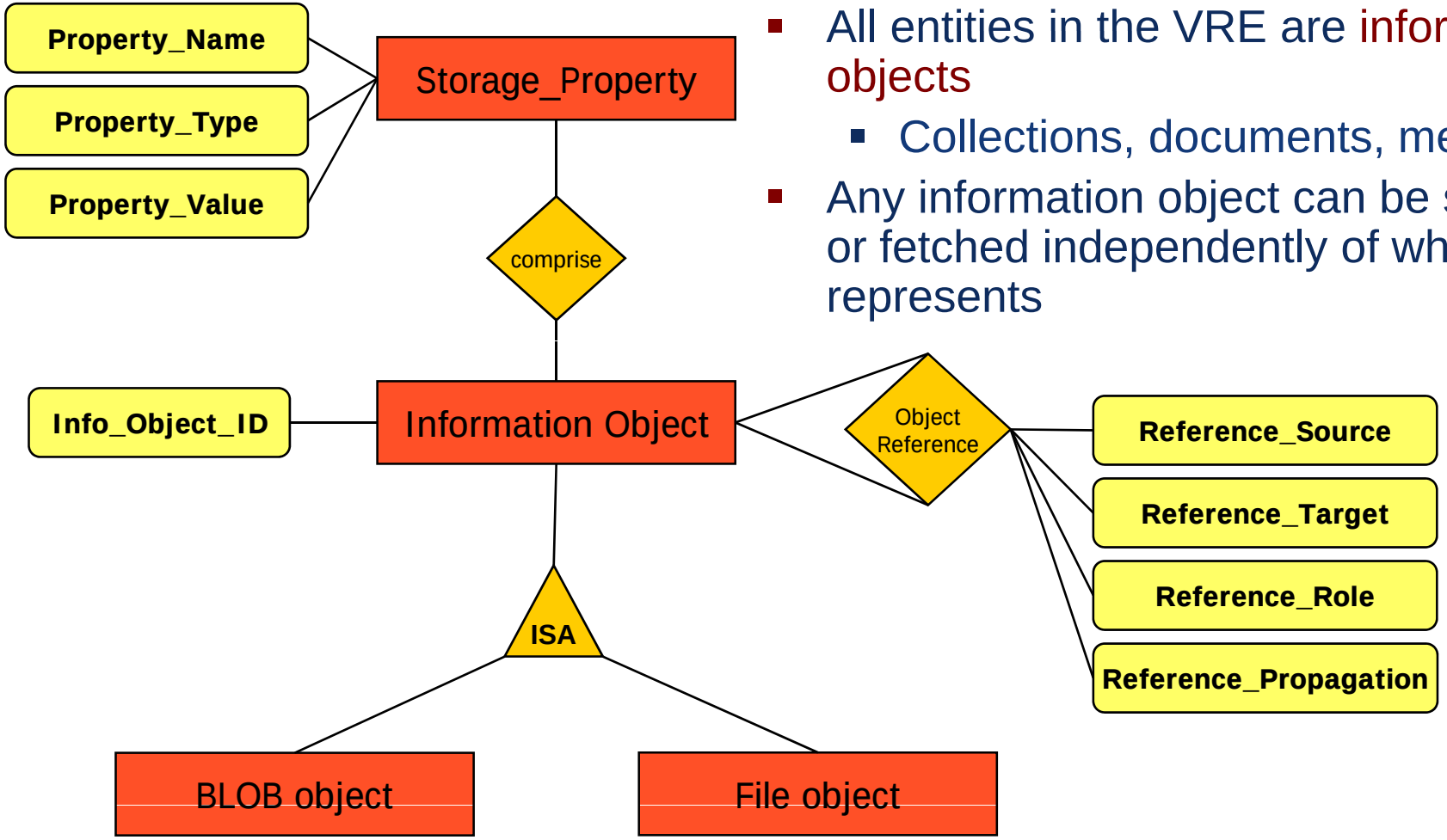
Feature Extraction

Metadata Management

Metadata as XML Document

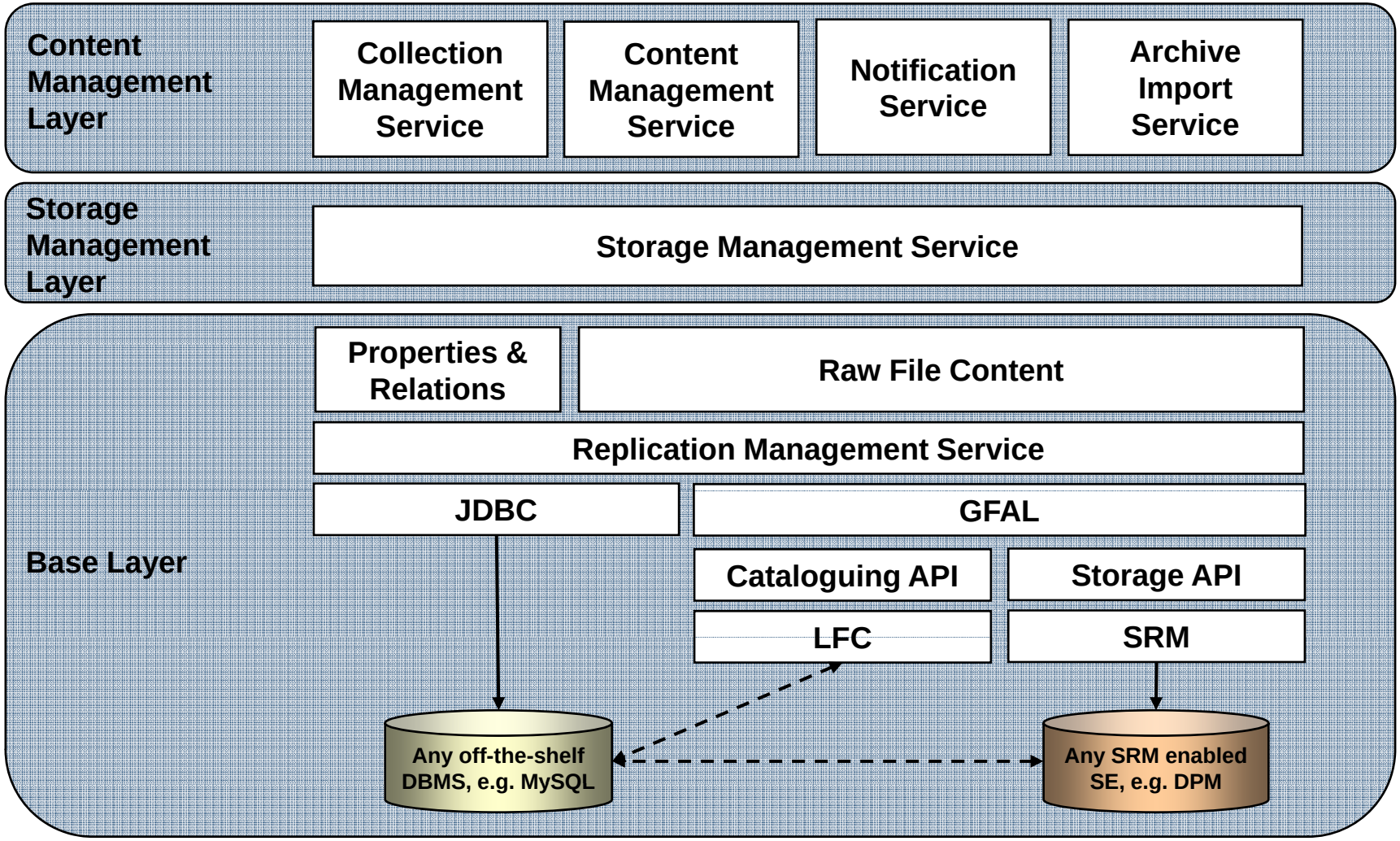
```
<DIMAP_DOCUMENT>
 <DATASET>MER_RR__2P</DATASET>
 <INSTRUMENT>MER</INSTRUMENT>
 <RESOLUTION_TYPE>RR</RESOLUTION_TYPE>
 <PRODUCT_LEVEL>2P</PRODUCT_LEVEL>
 <PRODUCT_NAME>MERIS</PRODUCT_NAME>
 <BAND_USED>___ALGAL_1</BAND_USED>
 <BAND_USED_NORM>ALGAL 1</BAND_USED_NORM>
 <START_DATE>2005-08-01</START_DATE>
 <END_DATE>2005-08-07</END_DATE>
 <LONMIN_INT>17000</LONMIN_INT>
 <LATMIN_INT>12000</LATMIN_INT>
 <LONMAX_INT>22000</LONMAX_INT>
 <LATMAX_INT>13500</LATMAX_INT>
 <COVER_REGIONS>World</COVER_REGIONS>
 <OVERLAP_REGIONS> World Europe Bigger_Europe Smaller_Europe Mediterranean
           Iberia North_Atlantic Africa North_Africa Middle_East Portugal
 </OVERLAP_REGIONS>
 <DATA_FILE_FORMAT>ENVI</DATA_FILE_FORMAT>
 ...
</DIMAP_DOCUMENT>
```

Document Model

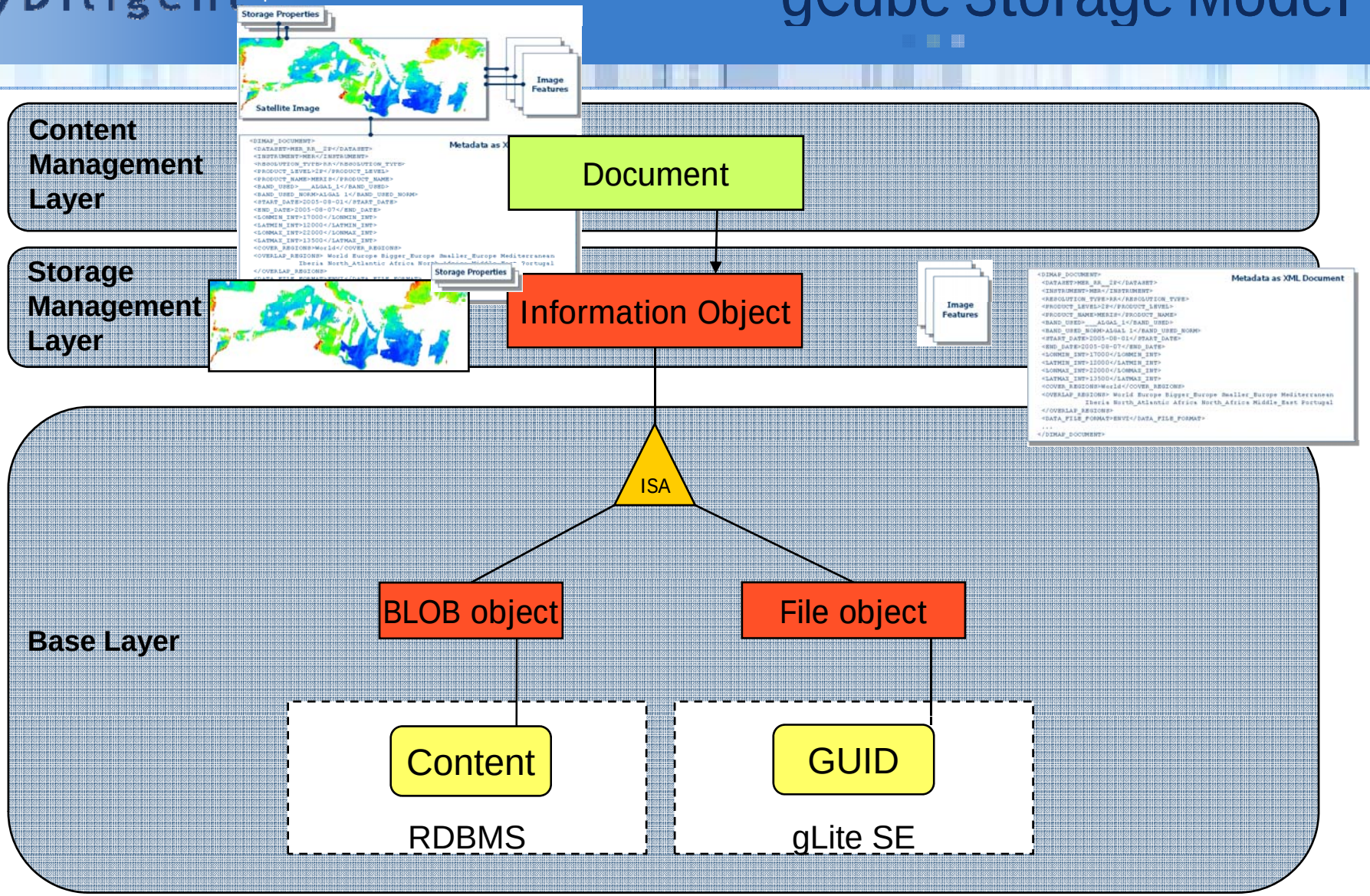


- All entities in the VRE are information objects
  - Collections, documents, metadata
- Any information object can be stored or fetched independently of what it represents

Data Management Architecture



gCube Storage Model



Where to Store What?

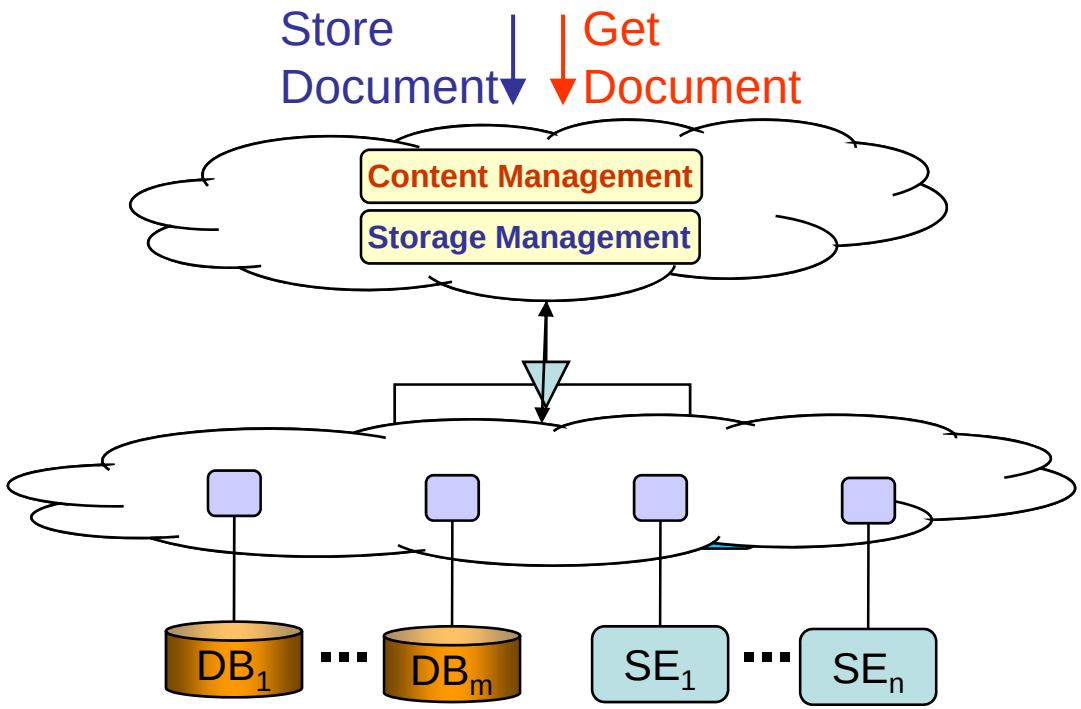
- An information object is stored in the relational database management system (RDBMS) and in the Storage Element (SE)
  - The RDBMS is used to store
    - properties of documents
    - relationships between documents
  - Grid storage is used to store raw data

Replication

- Replication
  - Fully or partially duplicates raw data among the nodes of a distributed system
- Replication management: responsible for the maintenance of replicas
  - Ensures consistency of multiple copies of the same data object
  - Identification of master site (original –and updateable– copy) and slave sites (replica)
- Basic approaches for maintaining replicas : eager vs. lazy
  - Eager replication: synchronizes replicas within the boundaries of the update transaction
  - Lazy replication: decouples synchronization from the updating transaction (replication done in separate transaction)

Replication Management Service

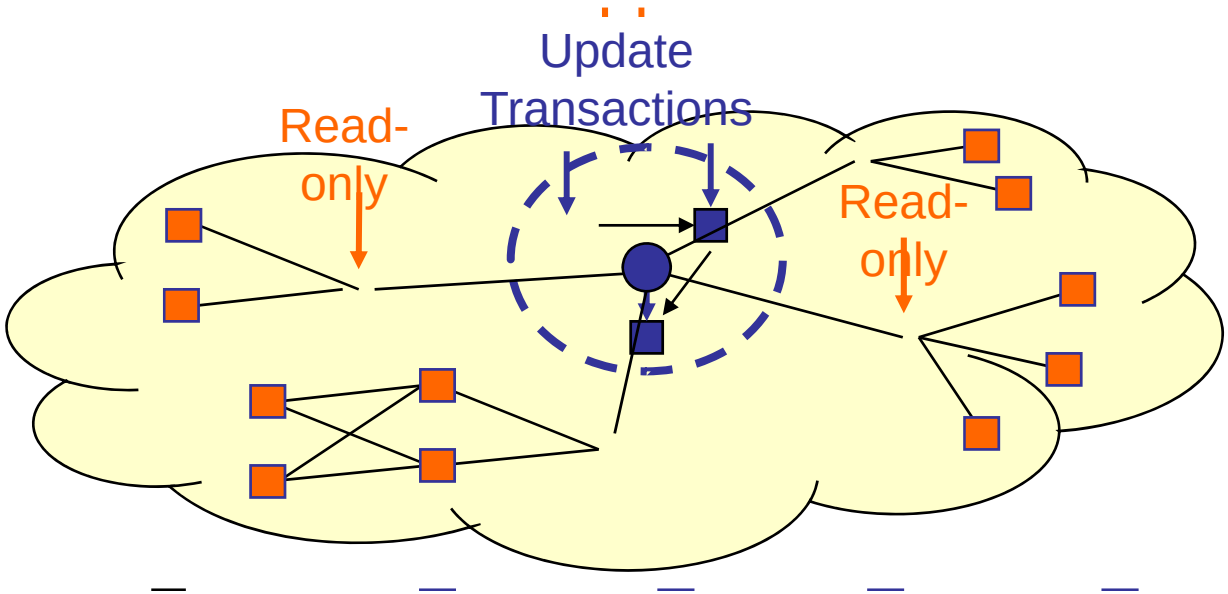
- Replication Management service is an internal service of the Base Layer
- Goal:
  - increase the degree of availability of content
  - Provides transparent access to data storage



Replication Management Service

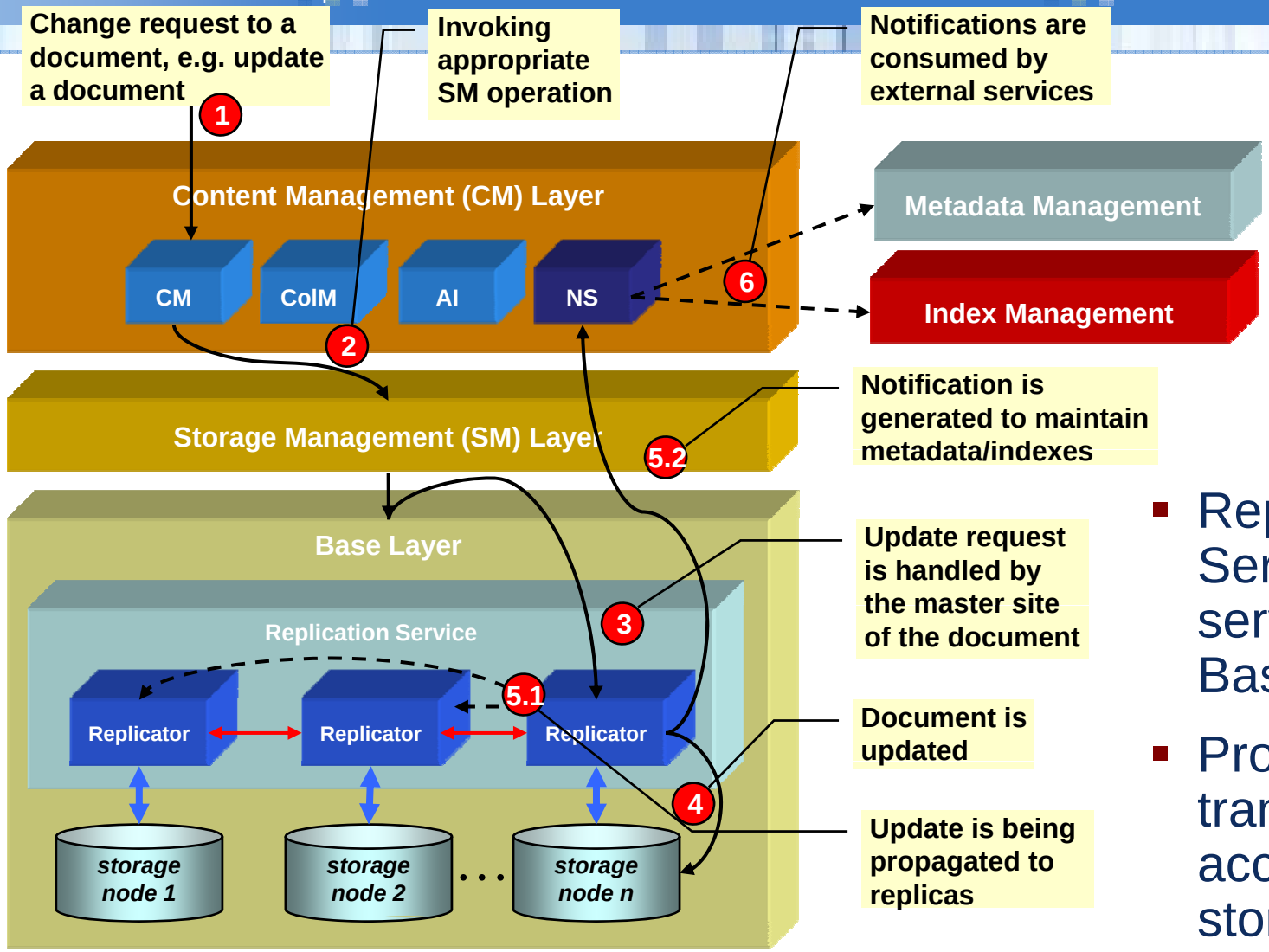
- Features
  - Ensure consistency of multiple copies of the same data object
  - Support data with different degrees of freshness
  - Guarantee consistent reads for all objects of a collection
    - Handle replication internally lazily, although being eager from the user‘s perspective
  - Adjust to changing load by dynamically creating new replicas

Advanced Replication in the Grid



- Update and read-only storage nodes in the Grid
- Query can be served by any node; changes only to be sent to update node
- Many update nodes per object: Correct serialization and propagation
- Specific features for replication in the Grid
  - Replicas subscribe for changes instead
  - Large number of heterogeneous nodes

Replication in gCube



- Replication Service: internal service of the Base Layer
- Provides transparent access to data stores

Content & Storage Mgmnt in gCube: Summary

- Basic tools and protocols are in place but gCube Data Management orchestrates all of them in order to
  - Associate the different parts of an information object
  - Associate information objects and all its meta data
  - Transparently replicate information objects and their meta data

Grid (gLite) Storage

- LFC: **L**CG **F**ile **C**atalog
  (LCG: **L**HC **C**omputing **G**rid / LHC: **L**arge **H**adron **C**ollider)
  - Centralized catalog for storing locations of files stored in the grid
  - Complete catalog can be replicated
- SRM: **S**torage **R**esource **M**anager. Interface to
  - Copy a file on a storage element
  - Gather information about a file stored into a storage element (SE)
  - Remove a file from a SRM storage
  - Retrieve information about a SRM managed storage.
- GFAL: **G**rid **F**ile **A**ccess **L**ibrary
  - provides calls for catalog interaction, storage management and file access and can be very handy when an application requires access to some part
- DPM: **D**isk **P**ool **M**anager
  - APIs for accessing local storage