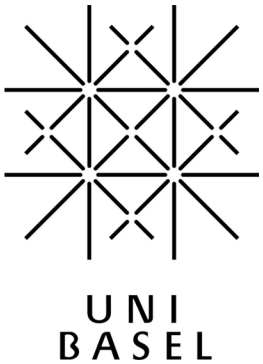
Requirements imposed by the new DLs

Tutorial at JCDL 2007
June, 19th 2007



Donatella Castelli (CNR-ISTI)

Agenda

09:00 – 09:20 **Introduction & Motivations**

09:20 – 09:45 **New DLs requirements**

09:45 – 10:30 **Underlying Technologies and their promises** (SOA, P2P, Grid)

10:30 – 10:45 *Coffee break*

10:45 – 12:00 **Solutions for decentralized DL infrastructures** (with BRICKS Demos)

12:00 – 12:30 **DelosDLMS - the DELOS Digital Library Management System**

12:30 – 13:30 Lunch

13:30 – 14:00 **DelosDLMS Demos**

14:00 – 15:00 **Building DL services on the Grid** (DILIGENT)

15:00 – 15:30 *Coffee break*

15:30 – 16:45 **DILIGENT Demos**

16.45 – 17:00 **Conclusions and future directions**

DELOS: Grand 10-Year Vision

"The potential exists for digital libraries to become the **universal knowledge repositories and communication conduits for the future**, a common vehicle by which everyone will **access, discuss, evaluate, and enhance information of all forms**"

Requirements - Cost

- DLs creation and maintenance must be "**cheap**" as many of the organizations that demand a DL are small, distributed, and dynamic; they use the DL to support temporary activities such as courses, exhibitions, projects, etc.

Requirements - Information in different form

- Multimedia documents (images, audio-videos, 3D-objects)
- Data (observation data, experimental data, specific elaborations outcomes)
- Information objects with no physical analogous
- On-demand information objects
- etc.

On-demand information objects

- a fixed text
- a pollution map
- a table summarizing data

  from millions of observed satellite measures
- a graph reporting an analytical trend of certain information extracted from a great amount of observed data

International Report on
Mediterranean Sea Chlorophyll Distribution during year 2003

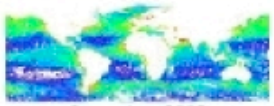
1. Scientific and Societal Concerns

Any scheme to monitor the ocean biota and their environment must strive to address the major scientific and societal concerns of the day pertaining to marine life. This section summarises some major concerns that emerged during discussions at the meeting. Many other concerns could have been included, but space precludes a complete listing of concerns.

1.1. Biodiversity and Conservation

Marine biodiversity is not easy to assess and is generally poorly known. There are many complicating factors, including a three-dimensional, fluid, mobile environment, its vastness, and its challenging depths. Away from shore, primary producers and primary grazers are usually small, drifting forms that undergo spatial variability and seasonal changes.

The larger invertebrate grazers have a range of life history stages, often with planktonic and benthic phases. Many large animals are migratory. Ocean habitats can be linked by the dispersal of planktonic larvae, and in this way, the systems can be interconnected even at a distance.

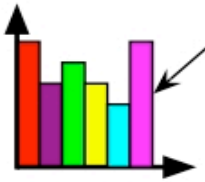


Jan – Apr 2003

Finally, the higher-order diversity of life is much greater in the oceans than in terrestrial systems—there are 13 unique phyla in the oceans and only one on land. Marine biodiversity is essentially the evolutionary history of life. In general, long-term environmental stability seems to increase biodiversity and, conversely, global climate change can be expected to decrease it.

	X1	X2	X3	X4	X5	X6	X7	X8	X9
Y1	12	13	15	26	11	34	45	45	54
Y2	32	12	46	67	21	22	44	12	44
Y3	23	33	56	77	32	44	12	55	33
Y4	44	34	12	55	34	45	12	22	44

Measures of yyy



Values of xxx

Automatically updated with the most recent data

On-demand information objects (cont.)

- Access to many different, large, heterogeneous information sources
- Use of specialized services
- Large processing capabilities

Requirements – universal knowledge repository

- Distributed content and metadata management
- Sharing: transparent access to distributed content, flexible metadata brokering and metadata integration, access control, etc.

Requirements–
access, discuss, evaluate, and enhance

- New services for handling new information objects,
  e.g. creation, visualisation
- "Collaboratoria" specific services
- Capabilities for adding new services
- Capabilities for creating new services as composition of existing ones

Requirements– Community specific DL

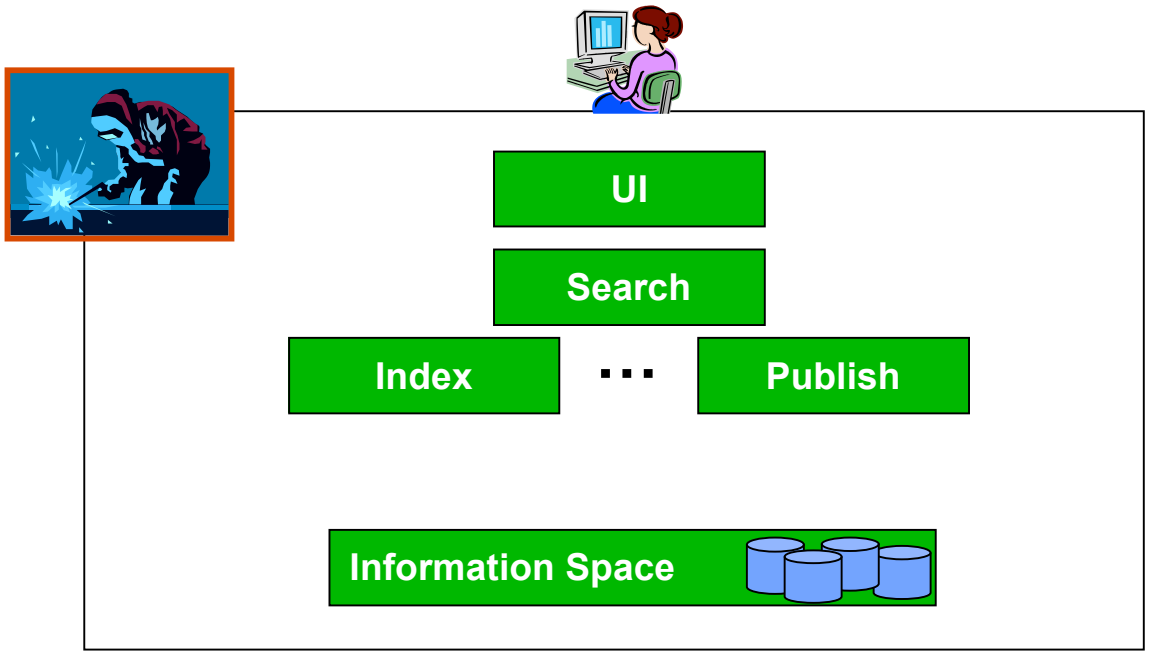
- Customizability of the DL functionality
- Personal Information space
- Richer user interfaces and interactions

Requirements– Quality of service

- Availability
- Scalability
- Response time
- Usability
- Integrity
- Security
- etc.

Meeting the challenges:
DL systems vs distributed infrastructures

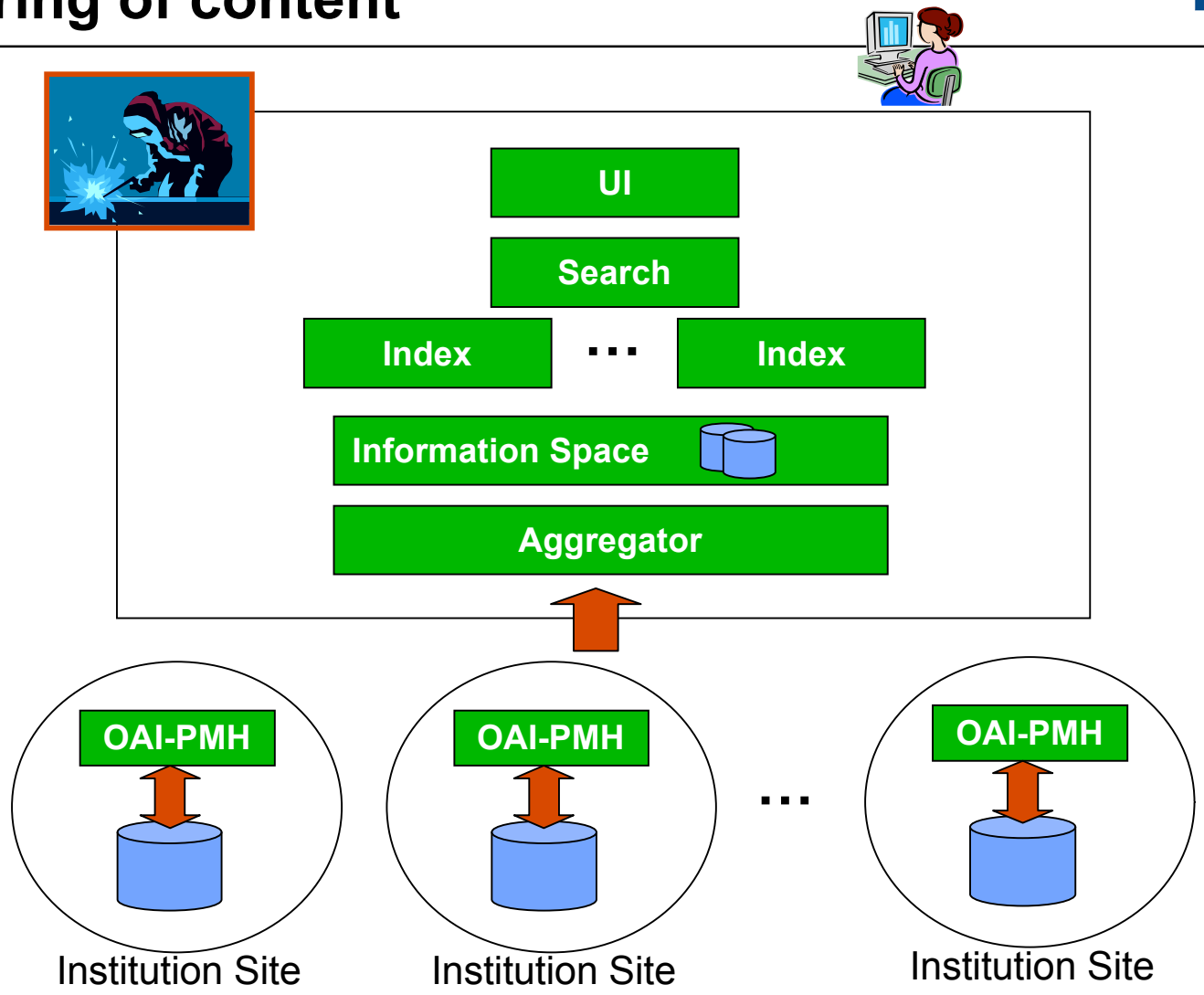
DL Systems in the past



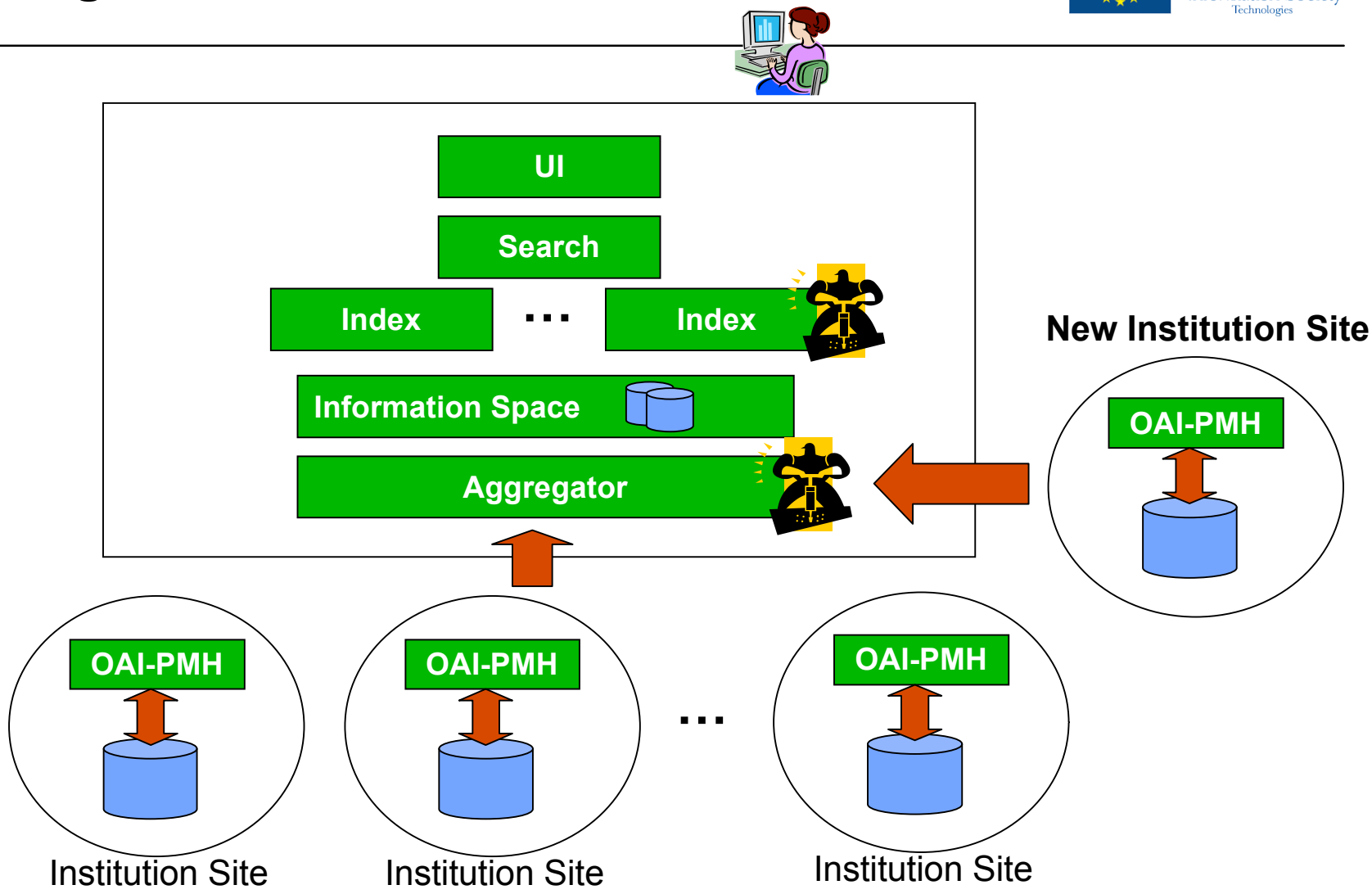
Lack of sustainability

- The construction and management of a DLS requires high investments and specialized personnel, content production is very expensive, multimedia and data handling requires high computational resources
- Years are spent in designing and setting up a DLS
- The systems lack interoperability and the services provided are difficult to reuse

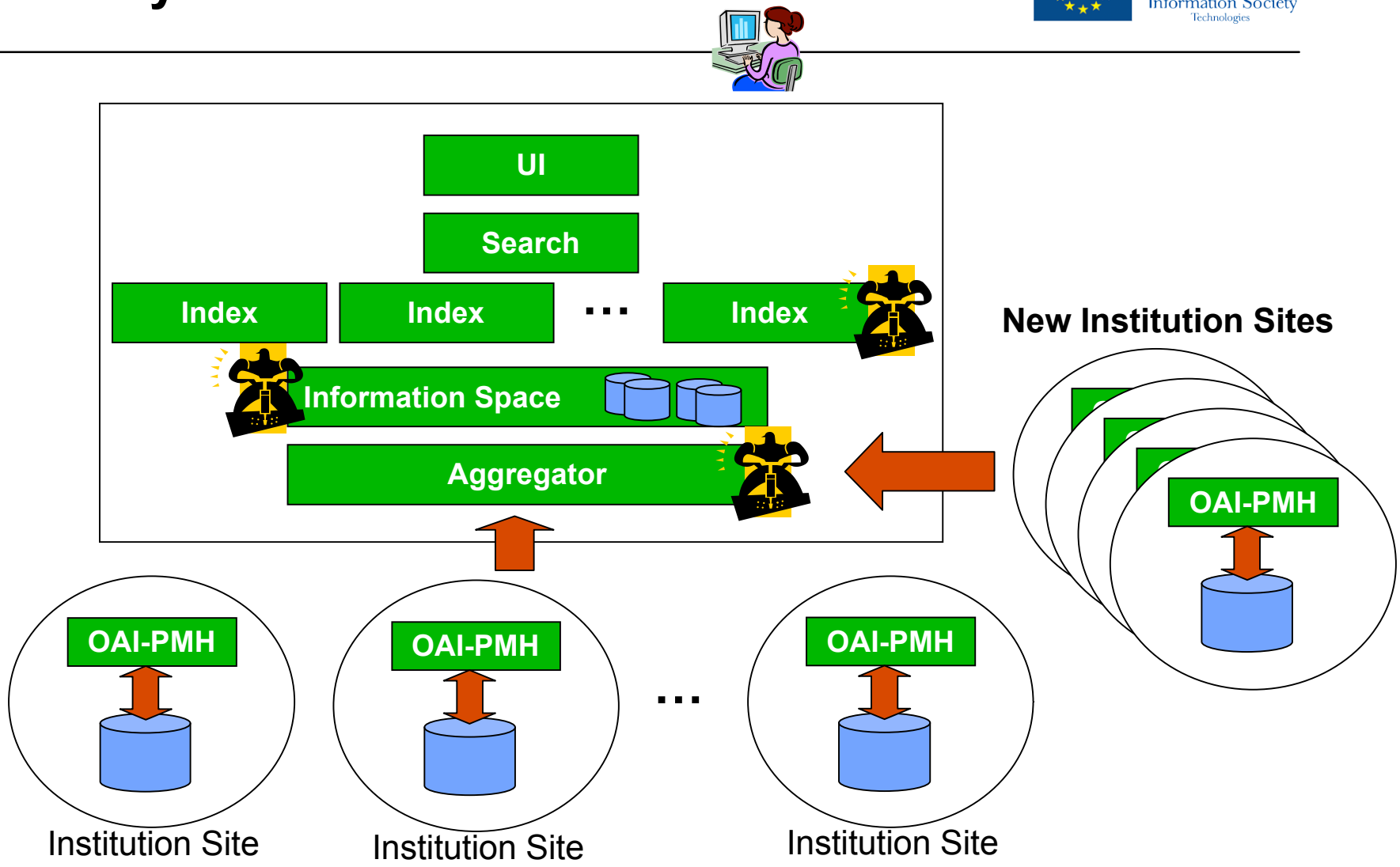
A solution: DL Systems – sharing of content



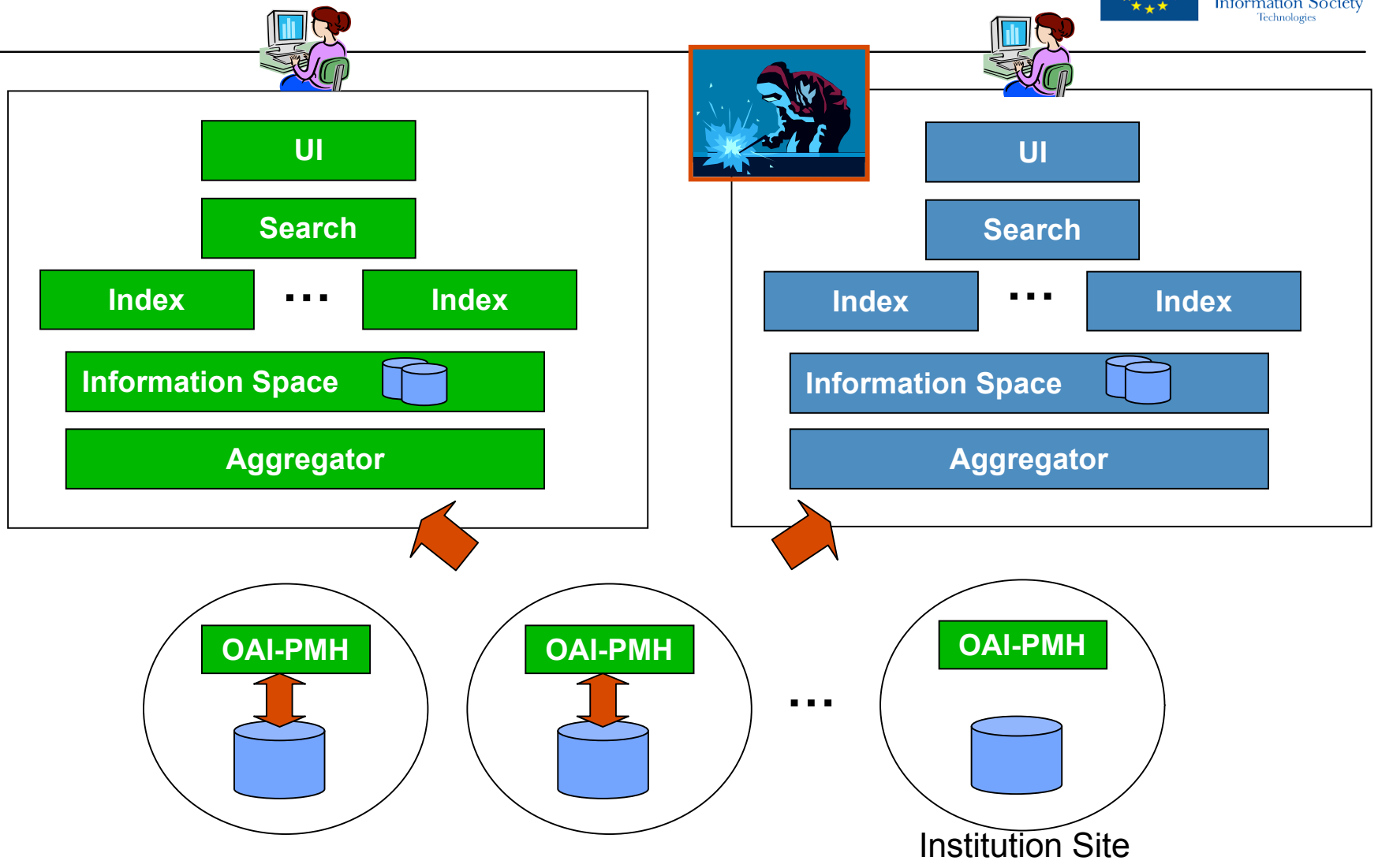
Still high maintenance costs



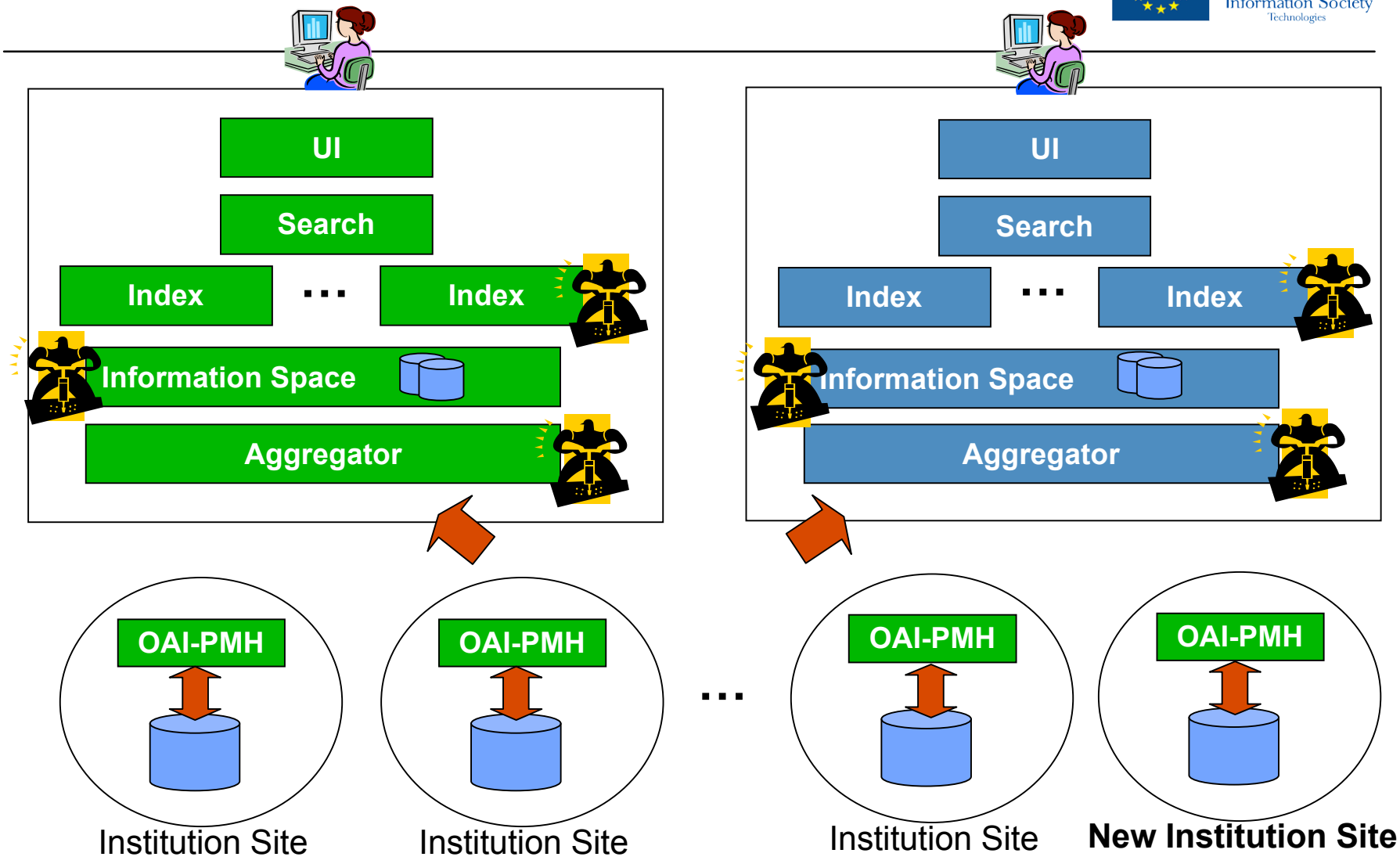
Still hardly scalable



Still not reusable



Duplicate efforts



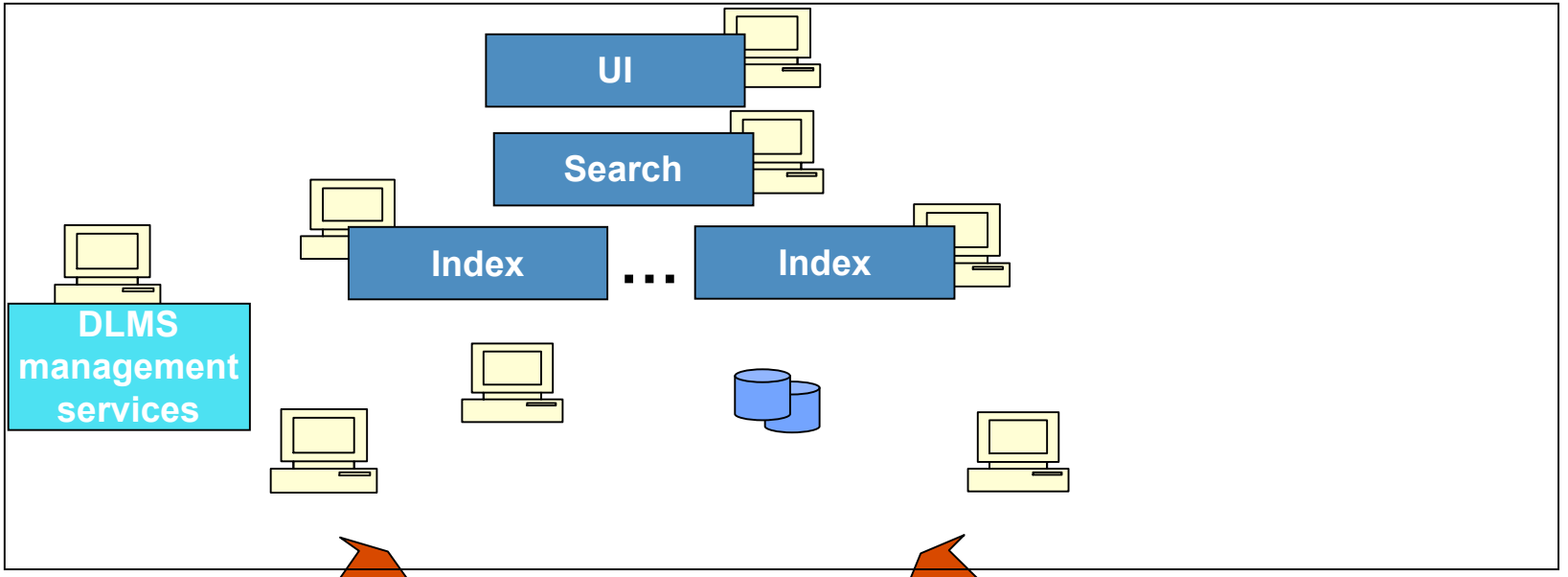
DLMS

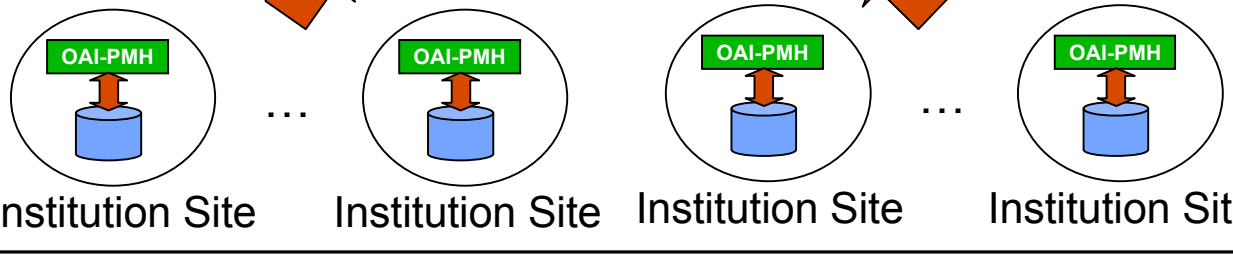
DL Designer

DL System Administrator

Service Repository

- UI
- Search
- Index
- Store
- Aggregator
- User Profiling
- ⋮
- Others





Content Resources

Maintenance through the DLMS

Information Society Technologies

DL Designer

Service Repository

- UI
- Search
- Index
- Store
- Aggregator
- User Profiling
- ⋮
- Others

DL System Administrator

UI

Search

Index ⋯ Index

DLMS management services

Store

Aggregator

Aggregator

OAI-PMH

OAI-PMH

OAI-PMH

Content Resources

Institution Site

Institution Site

New Institution Sites

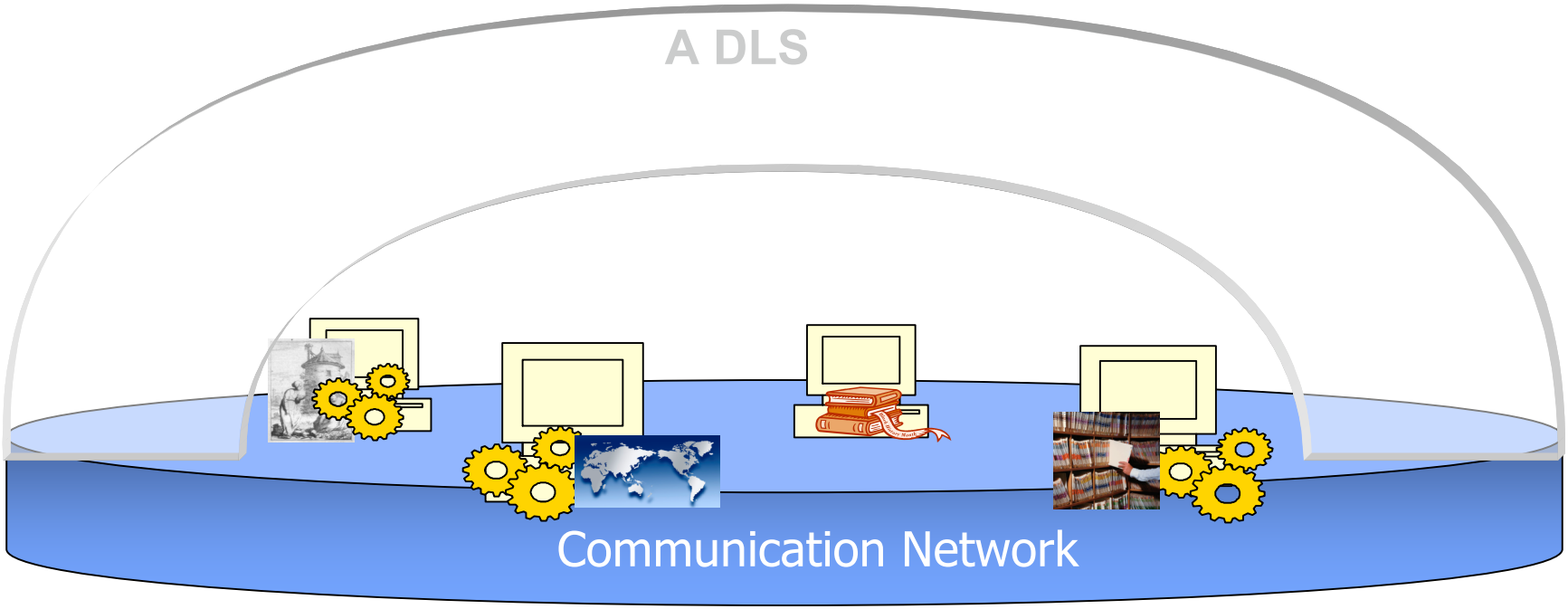
BRICKS
Building Resources for Integrated Cultural Knowledge Services

DELOS
Network of Excellence on Digital Libraries

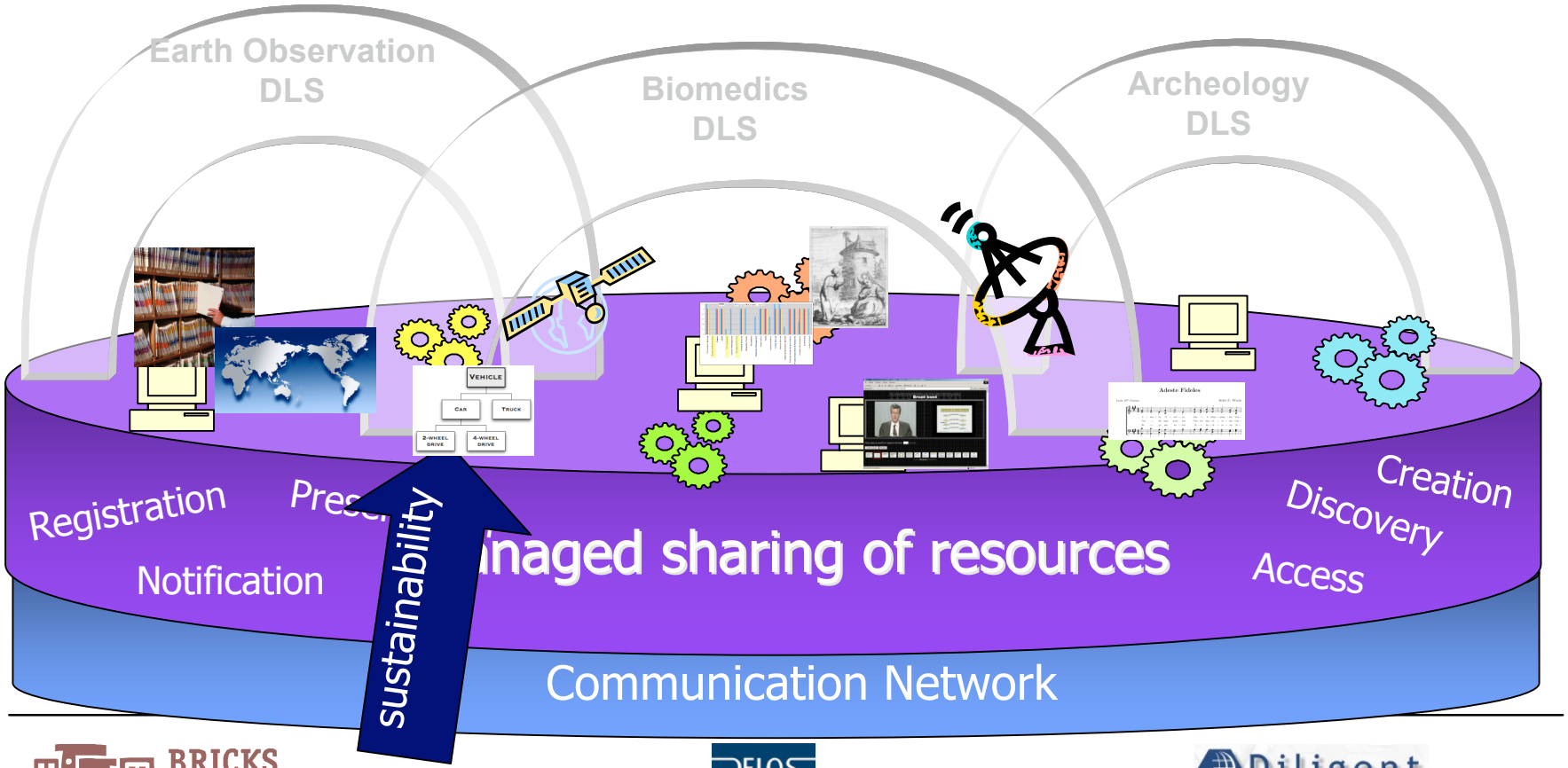
Diligent
A DIgital Library Infrastructure on Grid ENabled Technology

Infrastructures: A new paradigm

"The underlying foundation or basic framework (as of a system or organization)" *[Merriam-Webster]*



Resource sharing infrastructure



e-infrastructures – three examples

- BRICKS (http://brickscommunity.org)
- DELOS-DLMS (http://www.delos.info)
- DILIGENT (www.diligentproject.org)

Questions & Discussions

Agenda

09:00 – 09:20 **Introduction & Motivations**

09:20 – 09:45 **New DLs requirements**

09:45 – 10:30 **Underlying Technologies and their promises** (SOA, P2P, Grid)

10:30 – 10:45 *Coffee break*

10:45 – 12:00 **Solutions for decentralized DL infrastructures** (with BRICKS Demos)

12:00 – 12:30 **DelosDLMS - the DELOS Digital Library Management System**

12:30 – 13:30 Lunch

13:30 – 14:00 **DelosDLMS Demos**

14:00 – 15:00 **Building DL services on the Grid** (DILIGENT)

15:00 – 15:30 *Coffee break*

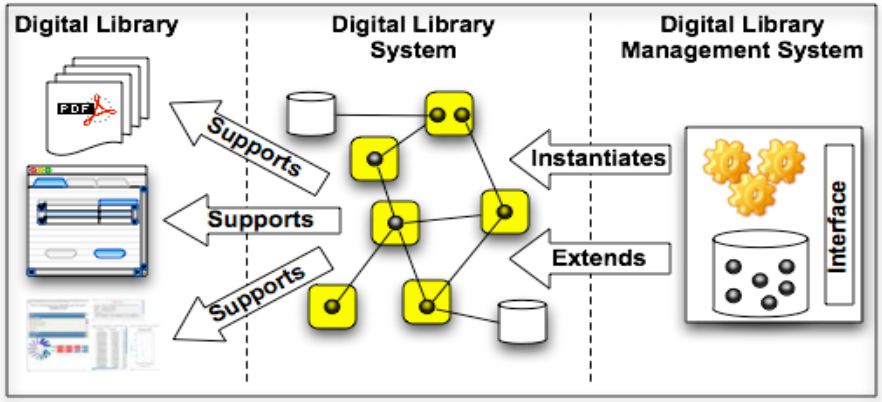
15:30 – 16:45 **DILIGENT Demos**

16.45 – 17:00 **Conclusions and future directions**

Introductory Concepts

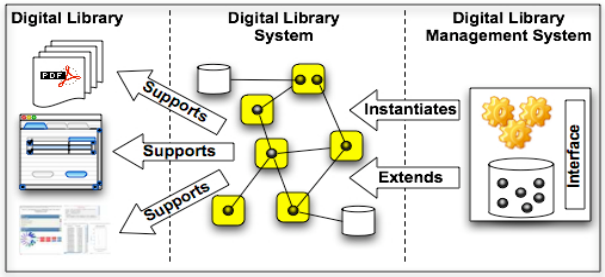
Introductory concepts: DL “systems”

- Digital Library
- Digital Library System
- Digital Library Management System



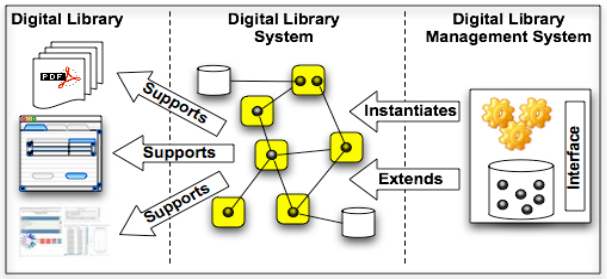
Digital Library

*A (potentially virtual) organization that comprehensively collects, manages, and preserves for the long term rich **digital content** and offers to its **user** communities specialized **functionality** on that content, of measurable **quality**, and according to prescribed **policies**.*



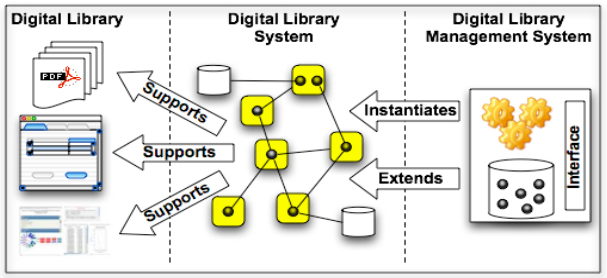
Digital Library System

A software system that is based on a (potentially distributed) ***architecture*** *and provides all functionality that is required by a particular Digital Library. Users interact with a Digital Library through the corresponding Digital Library System.*



Digital Library Management System

A generic software system that provides the appropriate software infrastructure to both (i) produce and administer a Digital Library System that incorporates all functionality that is considered foundational for Digital Libraries and (ii) integrate additional software offering more refined, specialized, or advanced functionality.



Introductory concepts: DL actors

- End-user
- DL designer
- DL system administrator
- DL application developer

Information Society
Technologies

- Exploit the DL functionality for providing, consuming, and managing the DL Content as well as some of its other constituents. They perceive the DL as a stateful(*) entity that serves their functional needs. DL end-users may be partitioned into:
    1. *Content Creator*
    2. *Content Consumer*
    3. *Librarian*

(*)The state of the DL corresponds to the state of its resources, i.e., it consists of the collections of information objects managed by the DL, its set of authorized users, its functionality, and its set of policies. This state changes during the Digital Library lifetime according to the functionality activated by the users and their inputs.

- Exploit their knowledge of the application semantic domain to define, customize, and maintain the Digital Library so that it is aligned with the information and functional needs of its end-users. They provide:
  - Functional configuration parameters:

    e.g. result set format, query language, user profile formats, document model
  - Content configuration parameters:

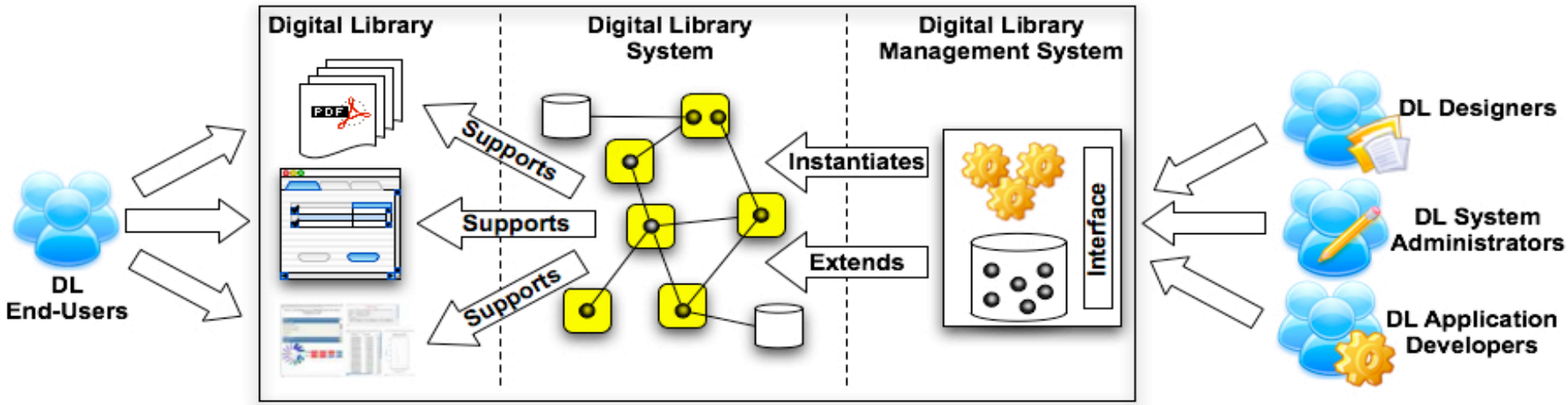
    e.g., repositories of content, ontologies, classification schemas, authority files, and gazetteers

- Select the software components necessary to create the Digital Library System needed to serve the required DL and decide where and how to deploy them. They identify the architectural configuration that better fits the DLS in target ensuring the appropriate level of quality. They also provide architectural configuration parameters:

  e.g. selected software components, hosting nodes, components allocation

- Develop the software components of DLMSs and DLSs, realizing the necessary functionality

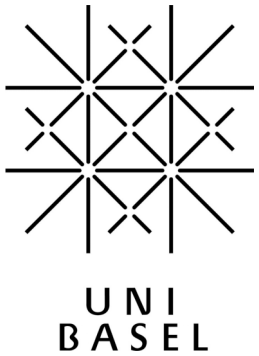
(Actors – Systems) interaction



Building Digital Libraries on Service Oriented Architectures

Underlying Technologies

Tutorial at JCDL 2007
June, 19th 2006



Thomas Risse (L3S)
Pasquale Pagano (CNR-ISTI)

Agenda

09:00 – 09:20 **Introduction: Motivation & Challenges**

09:20 – 09:45 **Challenges of bringing DL to distributed Infrastructures**

09:45 – 10:30 **Underlying Technologies and their promises** (SOA, P2P, Grid)

10:30 – 10:45 *Coffee break*

10:45 – 12:00 **Solutions for decentralized DL infrastructures** (with BRICKS Demos)

12:00 – 12:30 **DelosDLMS - the DELOS Digital Library Management System**

12:30 – 13:30 Lunch

13:30 – 14:00 **DelosDLMS Demos**

14:00 – 15:00 **Building DL services on the Grid** (DILIGENT)

15:00 – 15:30 *Coffee break*

15:30 – 16:45 **DILIGENT Demos**

16.45 – 17:00 **Conclusions and future directions**

Overview

- Service-oriented Architectures
  - Definition
  - Service Model
  - Web Service Stack
  - Example: Supply Chain
- Grid Computing
  - What is GRID computing?
  - Why is it different?
  - Why do it?
  - Grid Peculiarities
- Peer to Peer Infrastructures
  - The nature of peer-to-peer systems
  - Some history
  - Application domains
  - The Data access challenge
- Summary

Service Oriented Architectures (SOA)

Definitions

Gartner Group (technical definition)

- Web Services are loosely coupled software components that interact with one another dynamically via standard Internet technologies.

Forrester Research (business definition)

- Web Services are automated connections between people, systems and applications that expose elements of business functionality as a software service and create new business value.

Service Oriented Computing

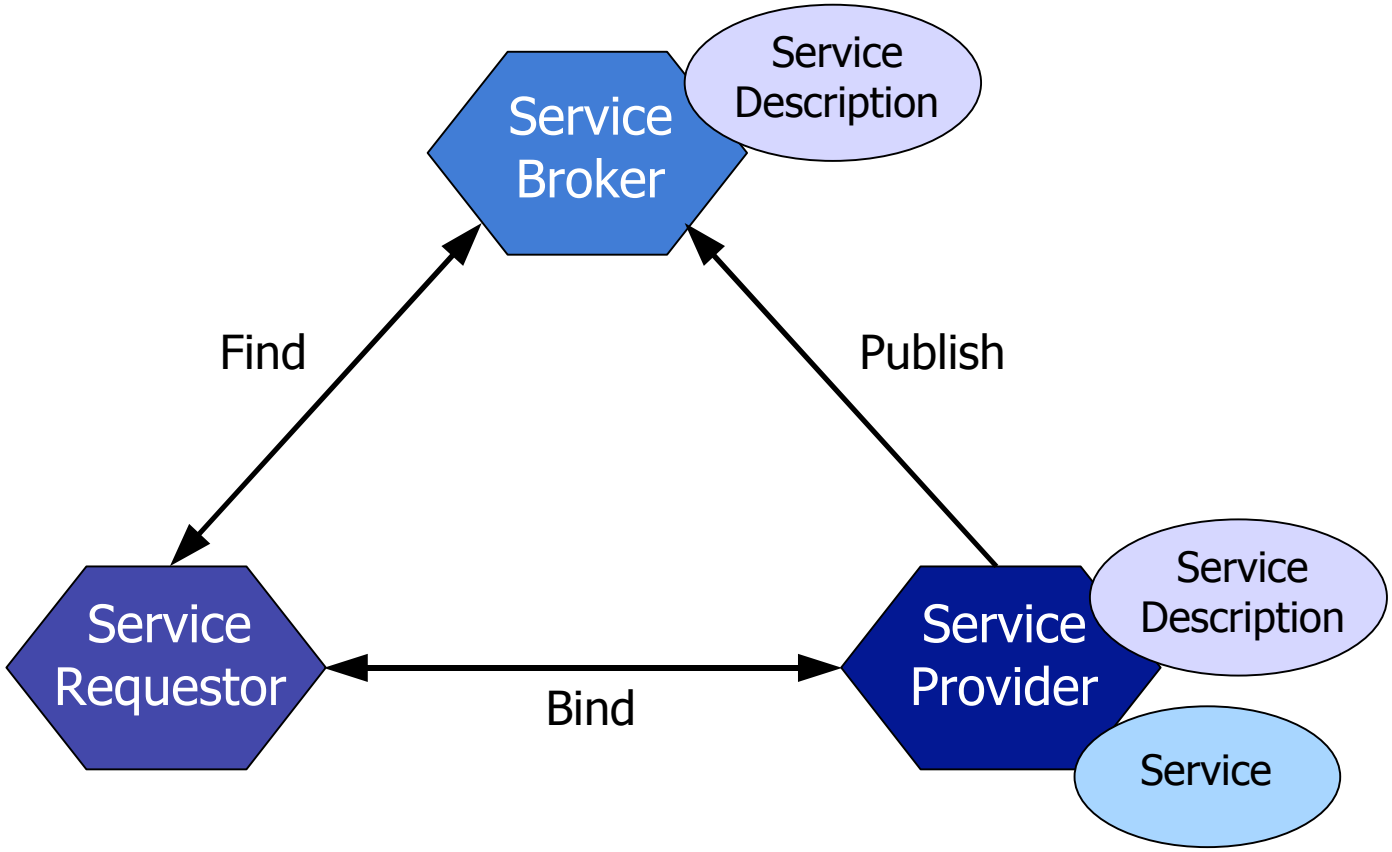
Goals of Service Oriented Computing

- Distributed interoperable Systems
- Cost reduction due to reuse of services
- Flexibility
- Easy adaptation to future developments

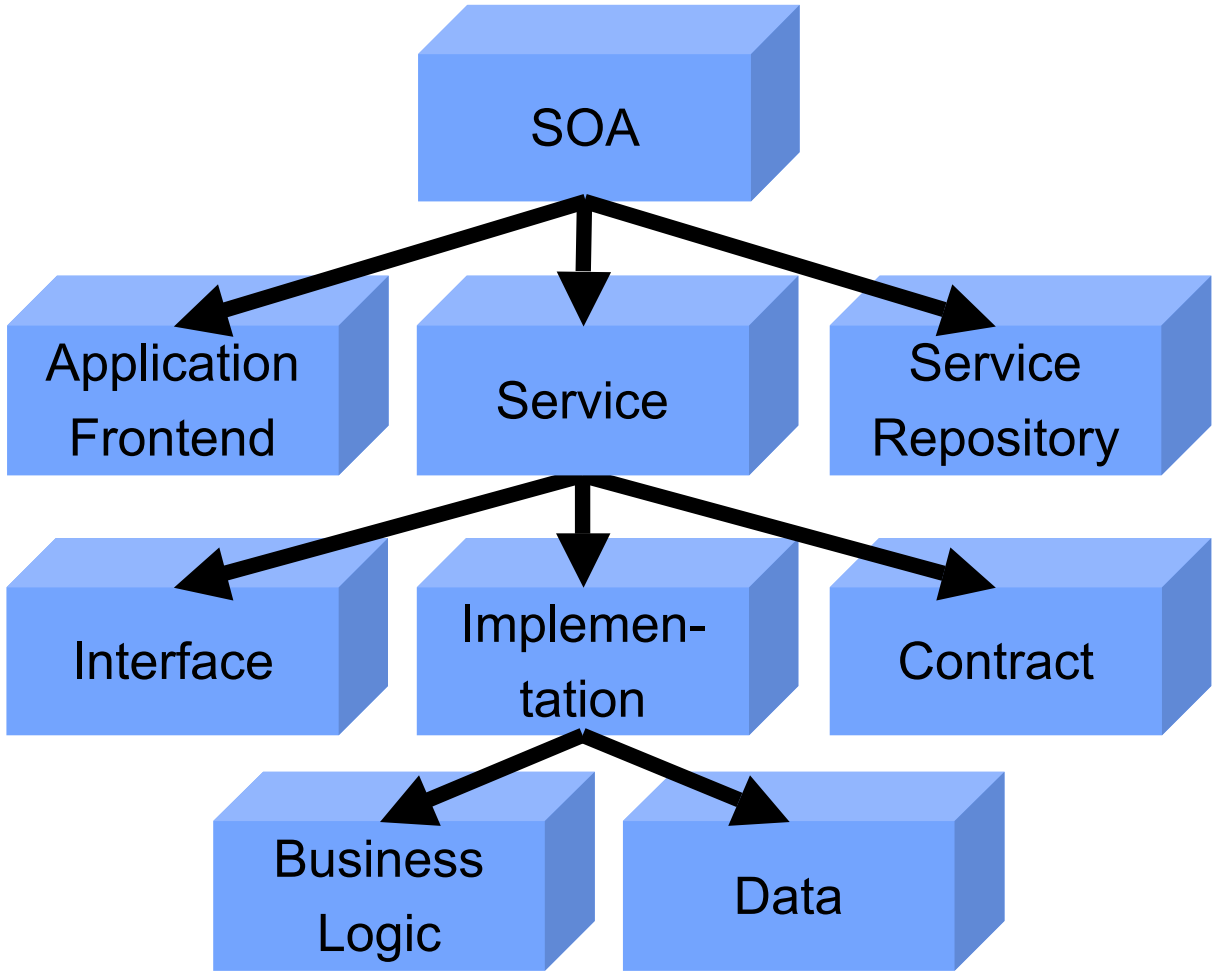
Properties of services

- Autonomous
- Loosely coupled
- Independent from the Platform / Operating System
- Independent from the Programming language
- Accessed using a well-defined interface
- Self describing

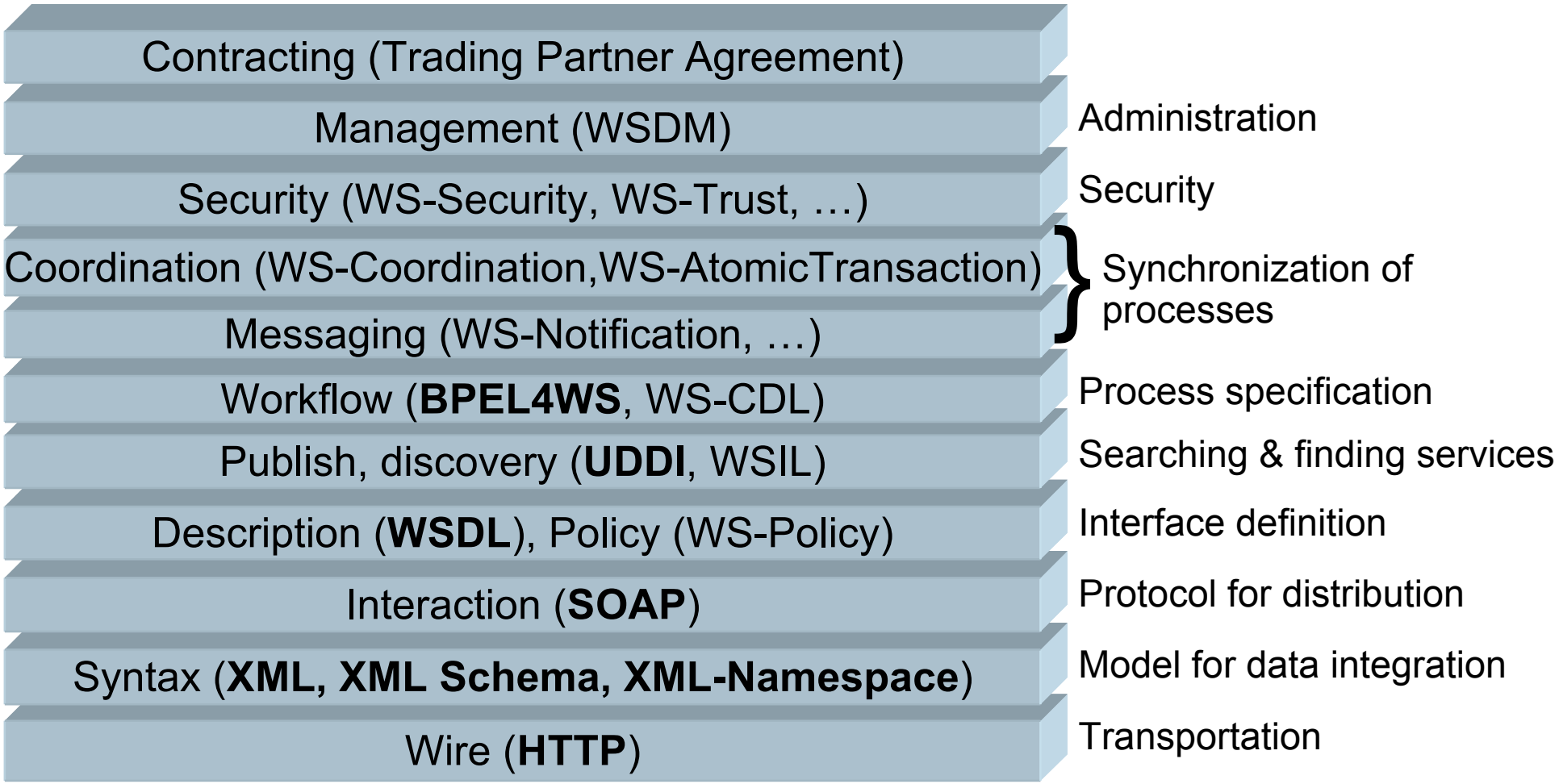
Service Model



Elements of SOA



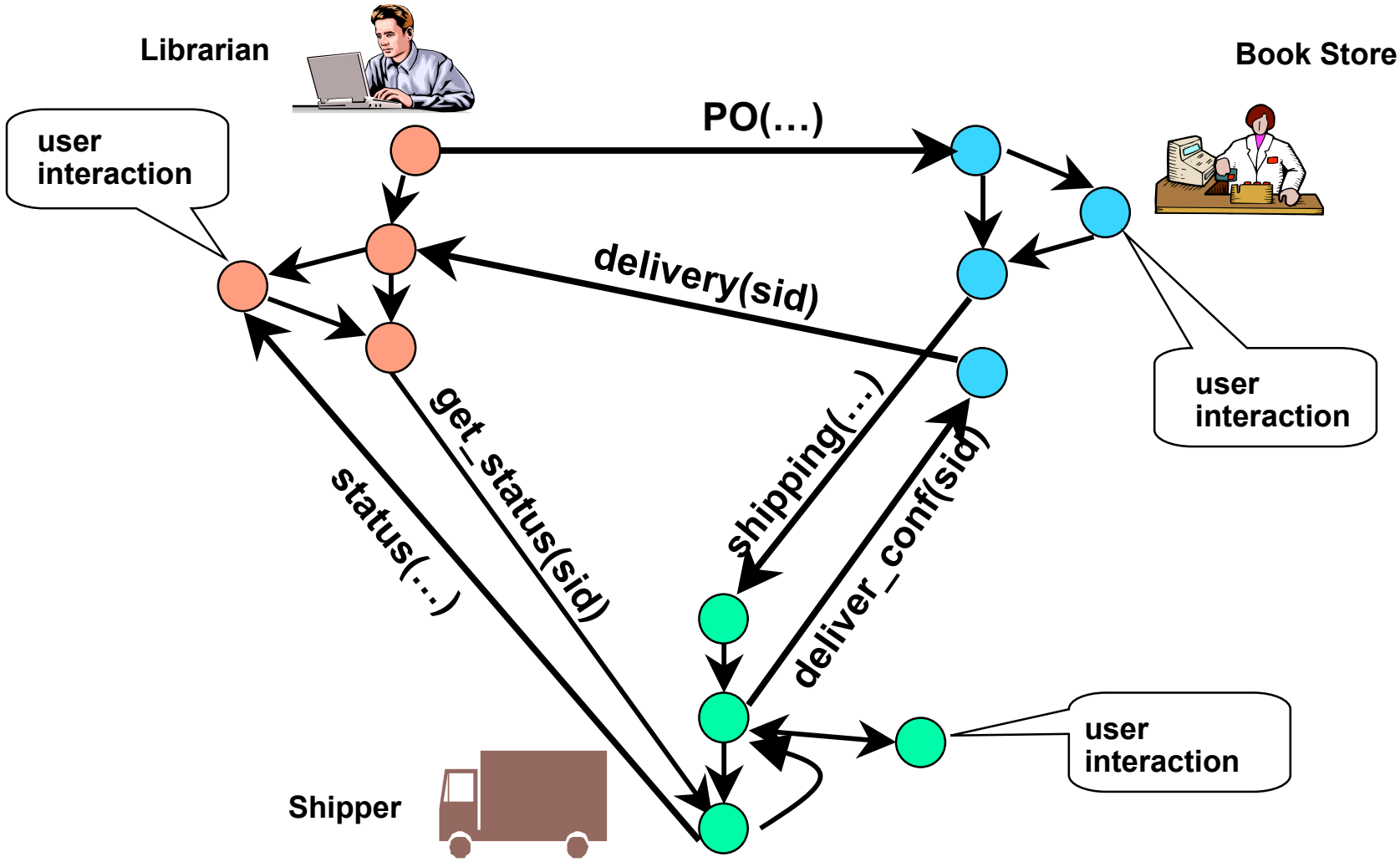
Web Services Stack



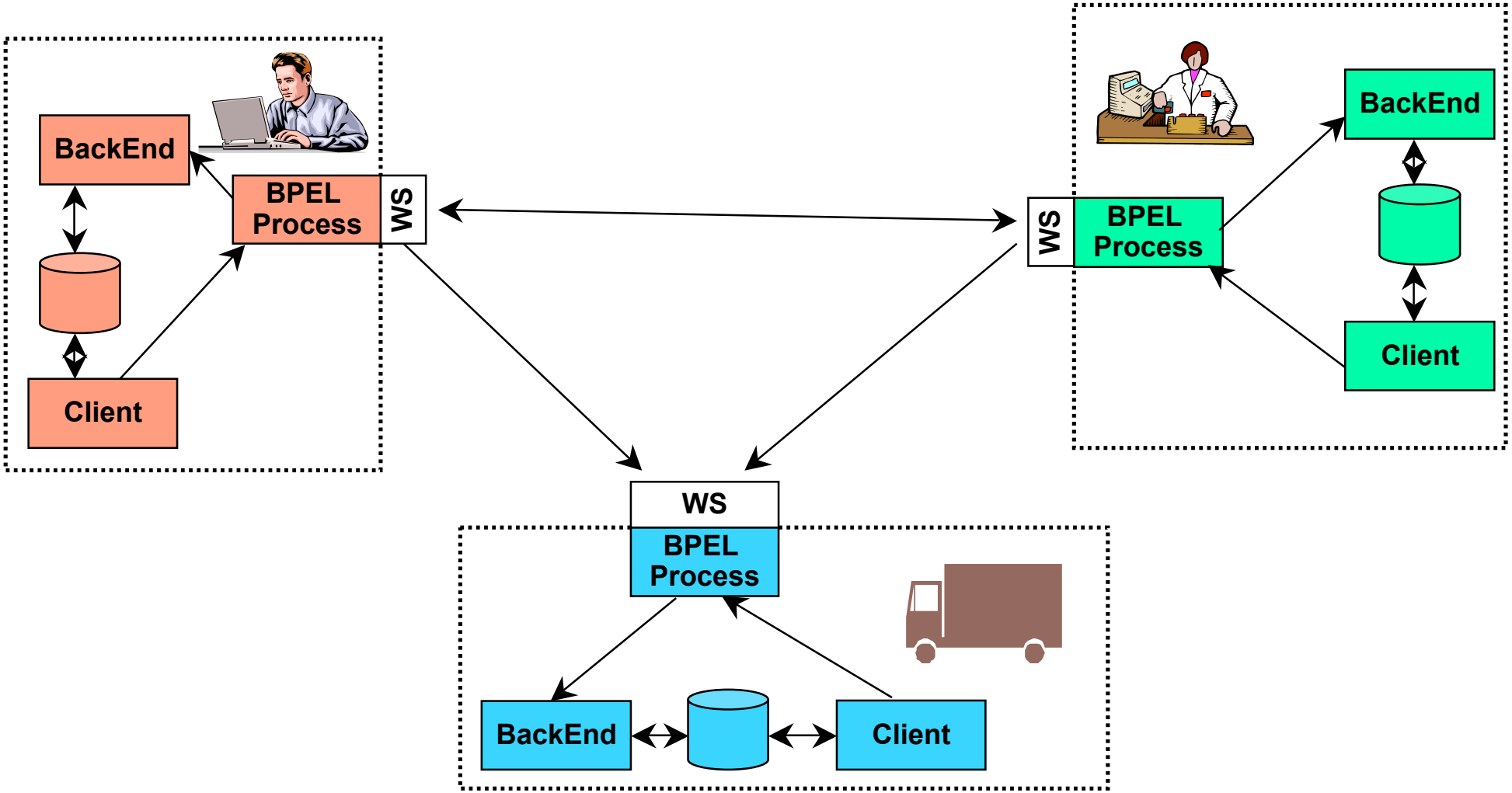
Relevant Standards Overview

XML	A universal model for data exchange and data integration
XML Schema	Defines the schema of a XML document, makes syntactical restrictions, defines structural patterns, defines data types
XML Namespace	Provide means to avoid naming conflicts in XML documents
SOAP	Simple Object Access Protocol - A universal communication protocol
WSDL	WS Description Language - Description of the WS interfaces, parameters, etc.
WS-Policy	General-purpose model to describe the policies of a Web service
UDDI	Universal Description, Discovery and Integration A standard for publication and discovery of information
BPEL4WS	Business Process Execution Language Specification of workflow for the composition of services
WS-Notification	Defines the publish/subscribe pattern for message oriented systems
WS-Coordination	Coordination of distributed actions. Includes transaction management.
WS-Security	Secure SOAP messages
WS-Trust	Management of trust relationships
WSDM	Distributed Management of Web Services

Supply Chain Scenario without Services



Supply Chain Scenario with Services



Advantages and Disadvantages

Advantages

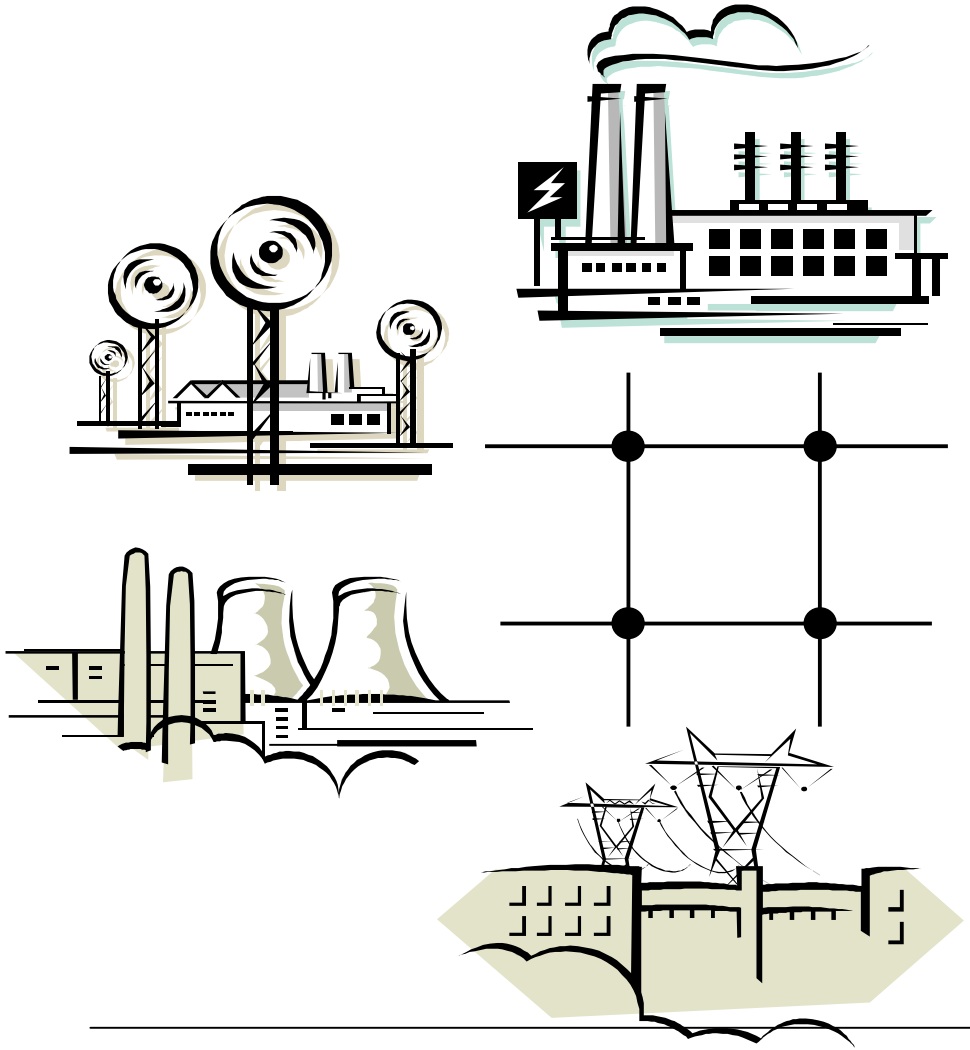
- Clear Description of Services and Interfaces
- Transparent access to Legacy Systems
- Higher Flexibility and Dynamics
- Widely accepted Web Service standards
- Various software is available

Disadvantages

- Semantic standards are still under development
- Several complementing standards are in development
- No Resource Sharing
- Depends partly on central management services

Grid Computing

Grid Infrastructures: Basic Idea …

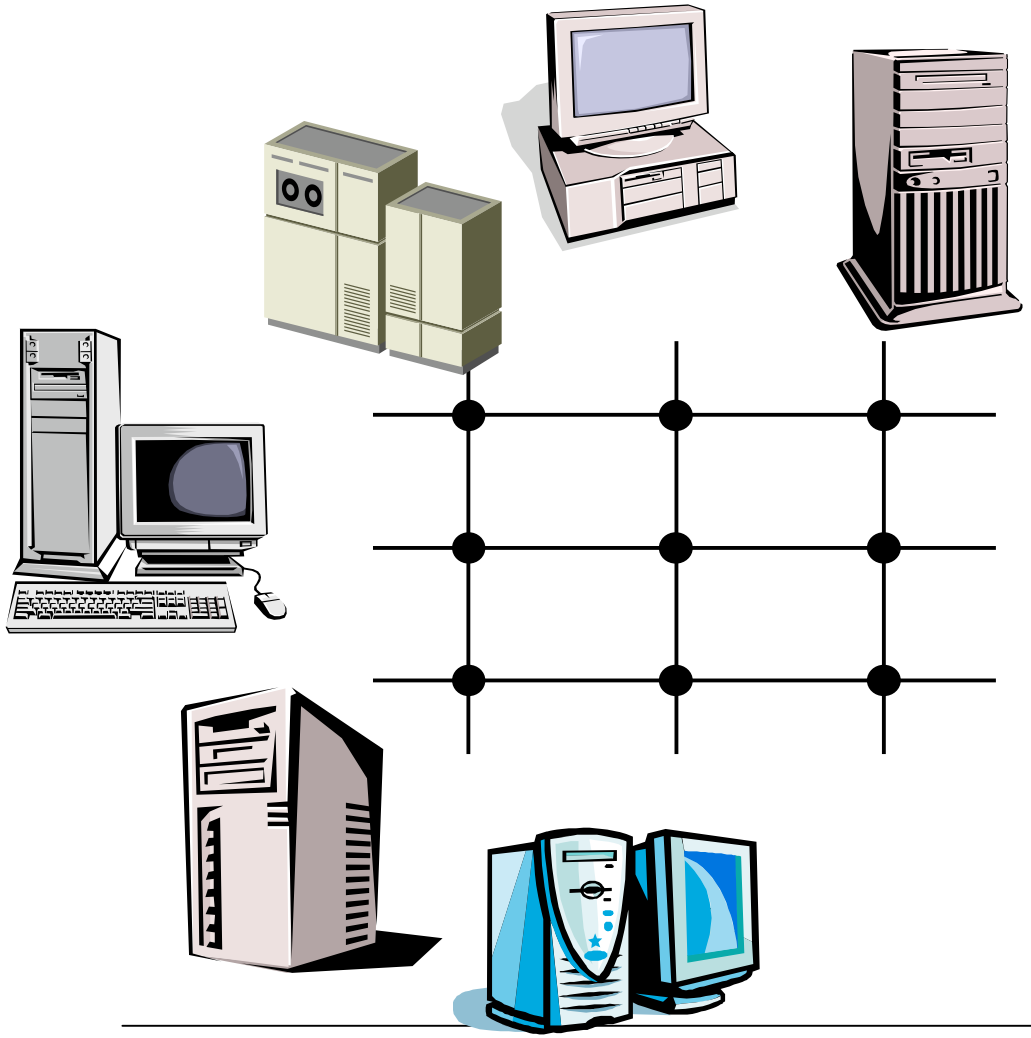


Basic Idea:

- Resources are available in a network without limitation

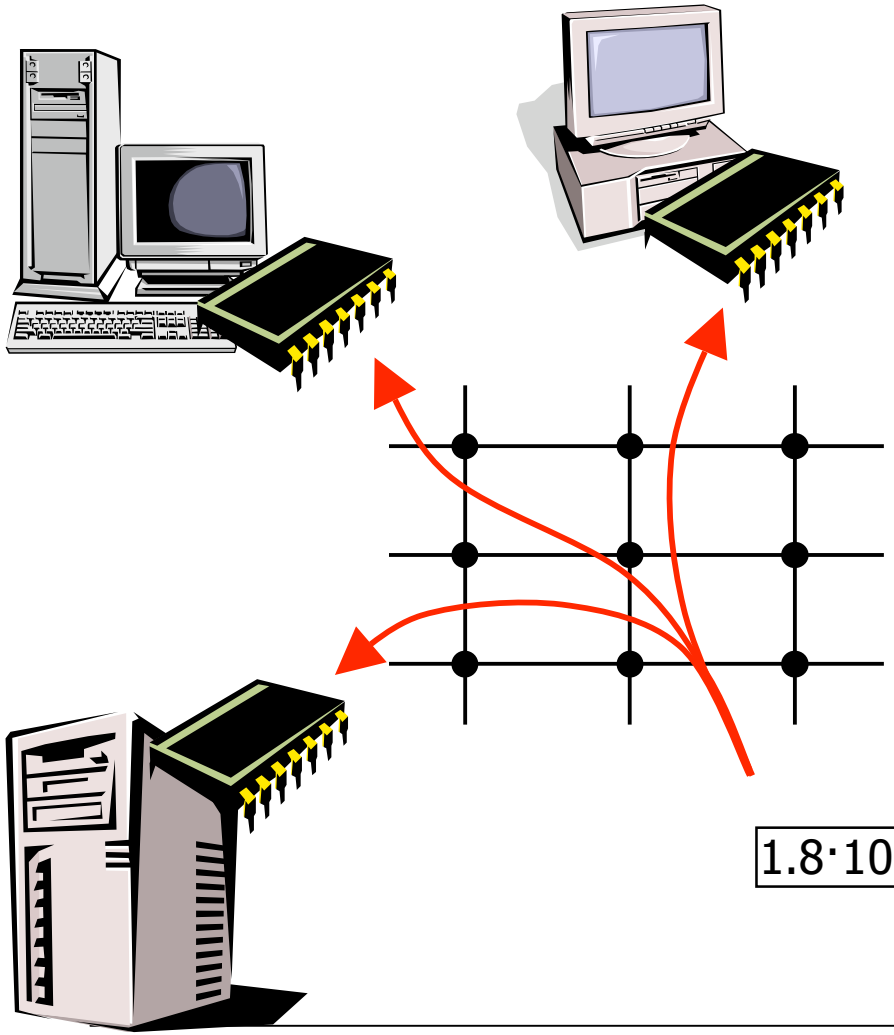
- Simple access to resources; users do not need to be aware of their origin/location
- Example: power line

… Grid Infrastructures: Basic Idea



Analogously, also computer and computing resources of a distributed system shall be exploited

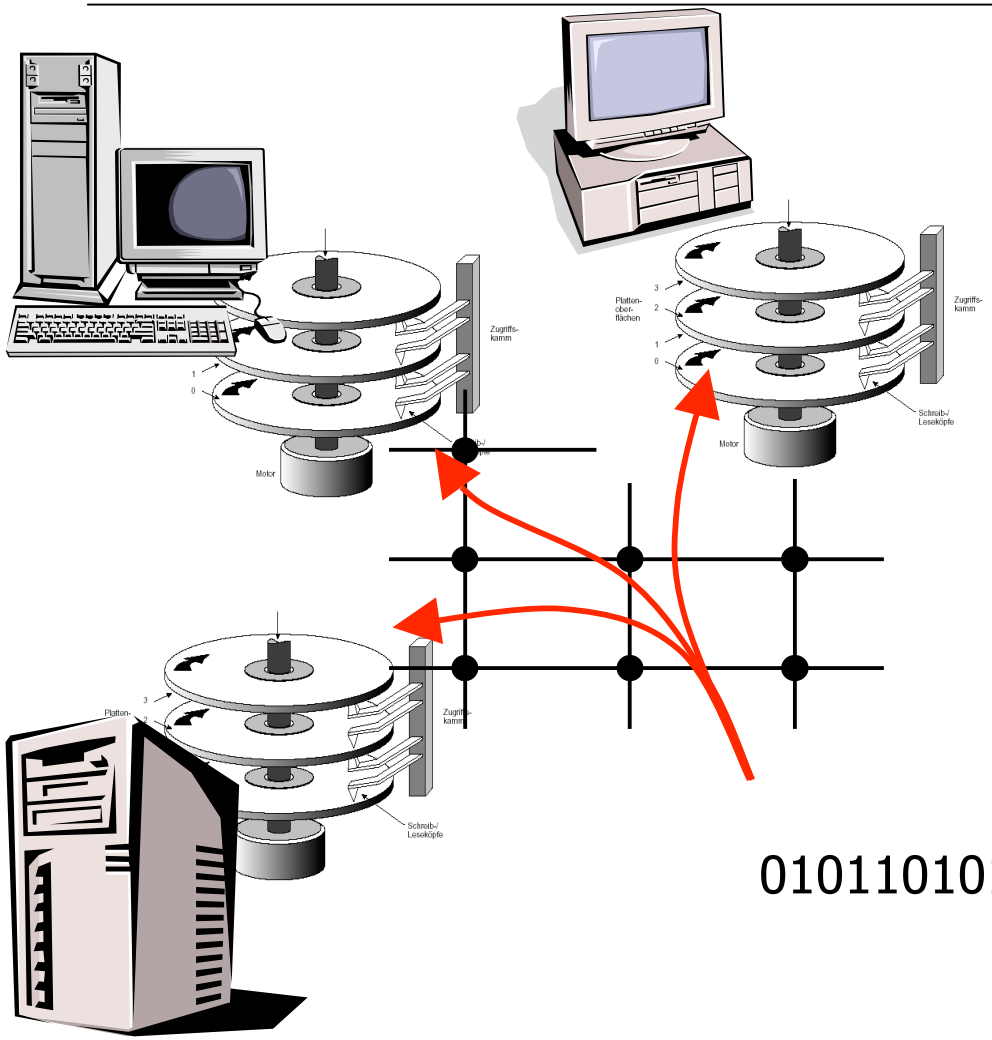
Computational Grid



- Resource which is shared in the Grid: CPU
- Large, complex computationally intensive problems are split in smaller ones and are processed in parallel in the Grid

$1.8 \cdot 10^{342} * 1.35 \cdot 10^{520} *$ $9.23 \cdot 10^{911} / 8.51 \cdot 10^{100} * ...$

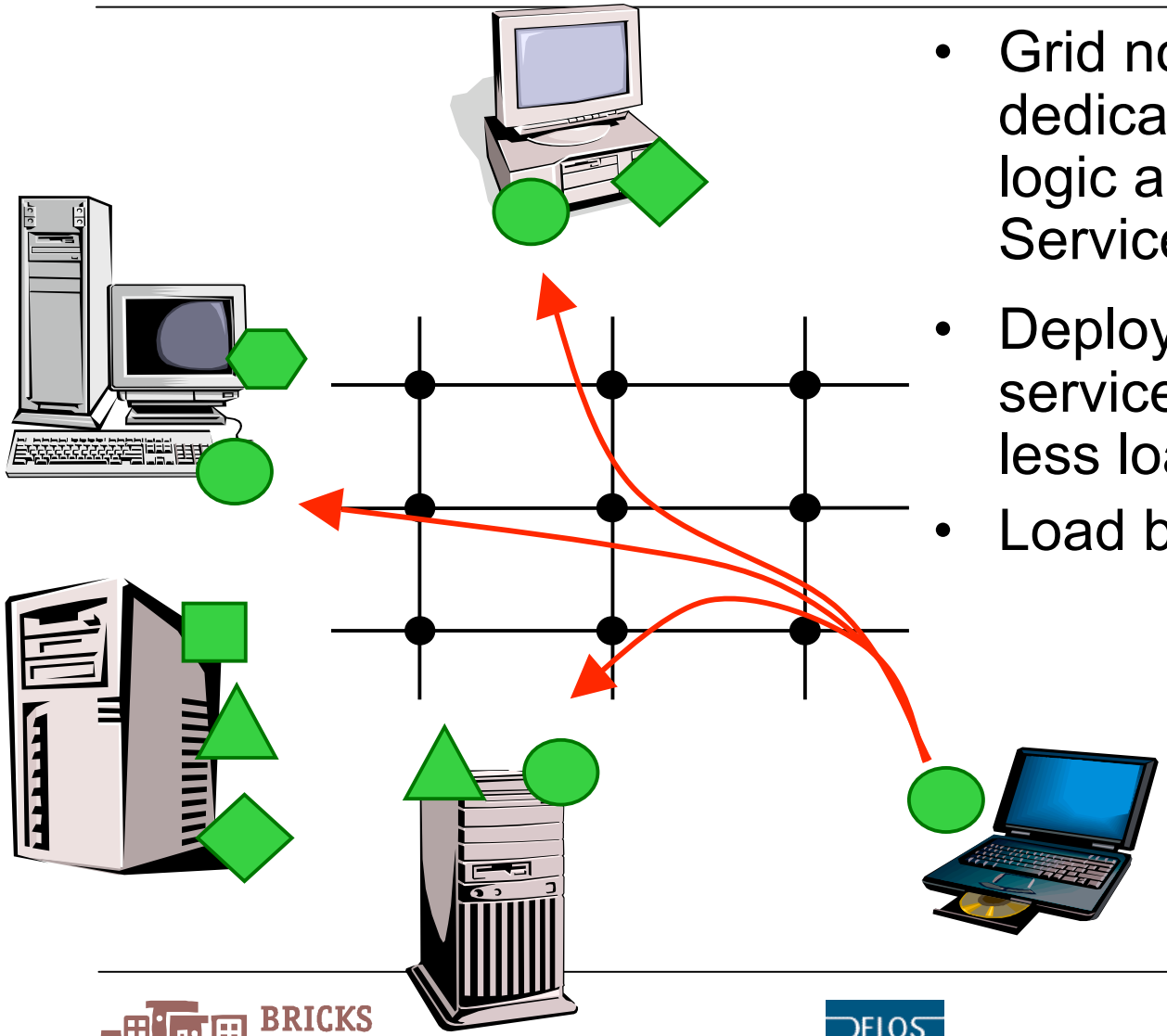
Data Grid



- Grid nodes (computers) provide storage
- The data Grid shall act as a large-scale, distributed database
- New data can be distributed among available storage nodes in the Grid

0101101010010101001011101011011011

Service Grid



- Grid nodes provide dedicated application logic as services (Grid Services)
- Deployment of new services if necessary, on less loaded hosts
- Load balancing in the Grid

What is GRID computing?

[dreamer's vision]

Whereas the Web is a service for sharing information over the Internet, the Grid is a **service for sharing computer power and data storage** capacity over the Internet. The Grid goes **well beyond simple communication** between computers, and aims ultimately to turn the global network of computers into **one vast computational resource**.

[scientist's vision]

Grid computing provides **flexible, secure, coordinated resource sharing** mechanisms among dynamic collections of individuals, institutions, and resources (***virtual organizations***).

GRID computing is about

- Resource sharing
- Secure access
- Resource use
- Wide area network
- Virtual organization

- Open Standards

Grid computing is about Resource Sharing

Resources are:

- computers, storage, data, remote software, sensors, networks, …
- owned by many different organizations
  - ž exists within different administrative domains,
  - ž run different software,
  - ž different security and access control policies.

Resource sharing is not about getting something for nothing, but is a situation where everyone concerned sees the advantage of sharing.

Grid computing is about Secure Access

Resource Sharing is always conditional: issues of trust, policy, negotiation, payment, …

Grid deals with:

- **Access policy** - resource providers and users must define clearly and carefully what is shared, who is allowed to share, and the conditions under which sharing occurs;
- **Authentication** - you need a mechanism for establishing the identity of a user or resource;
- **Authorization** - you need a mechanism for determining whether an operation is consistent with the defined sharing relationships.

Grid computing is about Resource Use

Efficient use of resources. No matter how many resources you have, there will always be times when there is a queue of people waiting to use them.

On the Grid, the **information about the different activities being submitted** are accessible, and the Grid MW is able to calculate the **optimal allocation of resources**.

Grid activities are beyond client-server communication and includes **coordinated problem solving**: distributed data storage and analysis, computation, collaboration, …

Grid computing is about Wide Area Network

The **performance** of wide area networks has been **doubling every nine months** or so over the last few years.

- WANs operate at 155 Mbps when in 1985 the US supercomputer centers were connected at 56 Kbps

Is it enough?

Ultra-low latency so there is no delay when for applications when working distributed on the Grid

Compensation for any failure that occurs on the Grid during a calculation, be it a transmission error or a PC crash

Grid computing is about Virtual Organization

Dynamic, multi-institutional aggregation of resource providers and consumers.

Community of different organizations:

- overlays on classic organization structures
- implies different system administrators, users, institutional goals, and often socio-political constraints

Large or small, static or dynamic

Is the Grid vision new?

"We will perhaps see the spread of 'computer utilities', which, like present electric and telephone utilities, will service individual homes and offices across the country."

Len Kleinrock, 1967

Is the Grid the first attempt?

Condor (University of Wisconsin)

- Started already in 1988
- Software system that creates a High-Throughput Computing (HTC) environment
- Effective utilization of computing power (workstations, servers, clusters)
- Specialized workload management system for compute-intensive jobs

Why is this Different?

Lack of central control

- Where activities run
- When they run

Shared resources

- conflict, evolve

Communication and coordination

- Cross-domain, cross-operating system, cross-resource type, …

So why do it?

- Computations that need to be done with a human acceptable time limit
- Data that cannot fit on one site
- Data produced by multiple sites
- Peaks of computations that are limited to specific events
- Applications that need to be run bigger, faster, …

Grid computing peculiarities

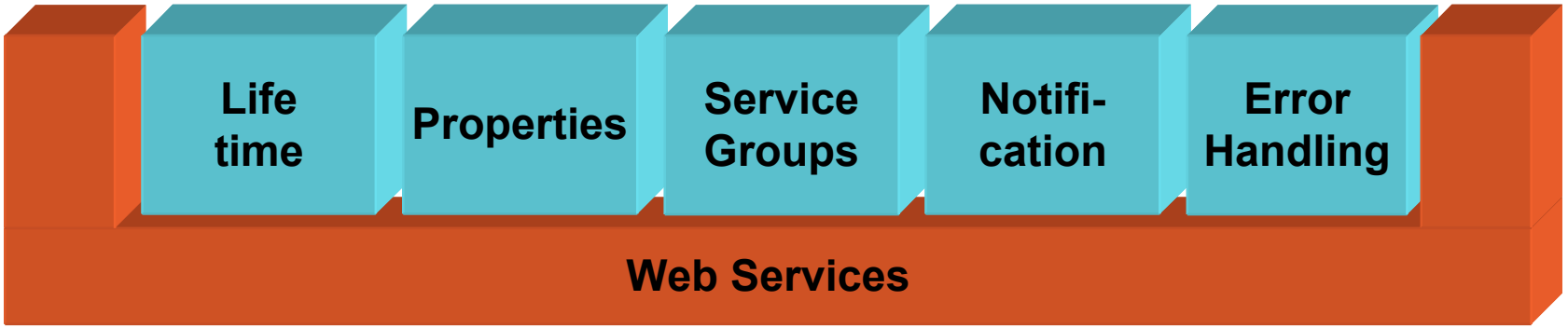
Distributed management

- Of physical resources
- Of software services
- Of communities and their policies

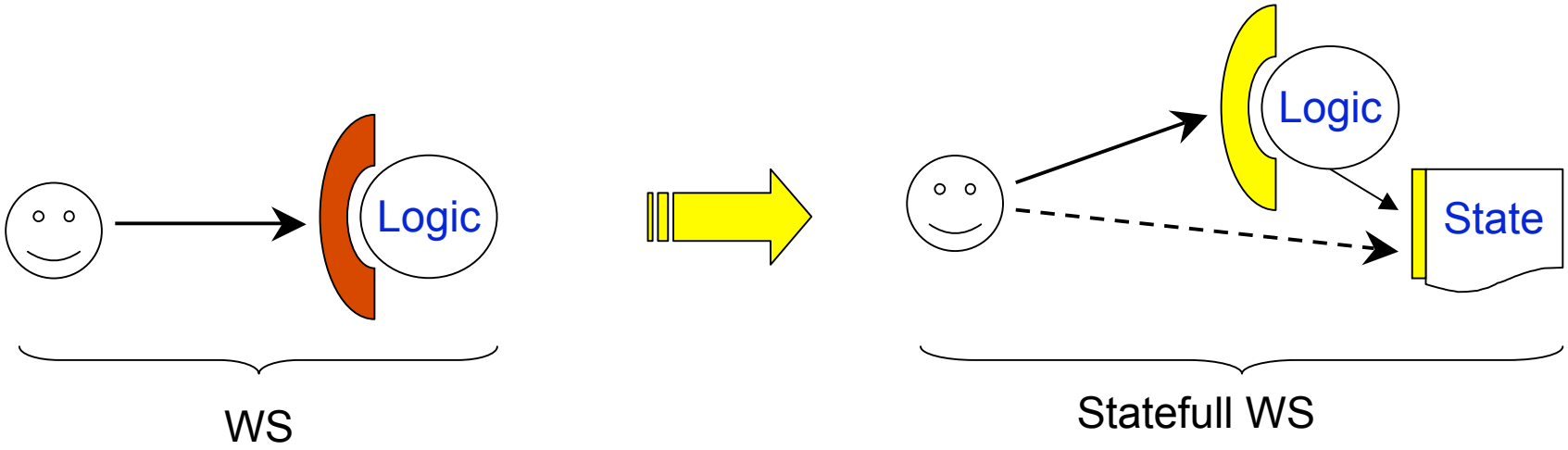
Unified approach

- Build on Web Service framework
  - ž Common management abstractions & interfaces
- Use WSRF, WS-Notification to represent/access state
  - ž Common state management

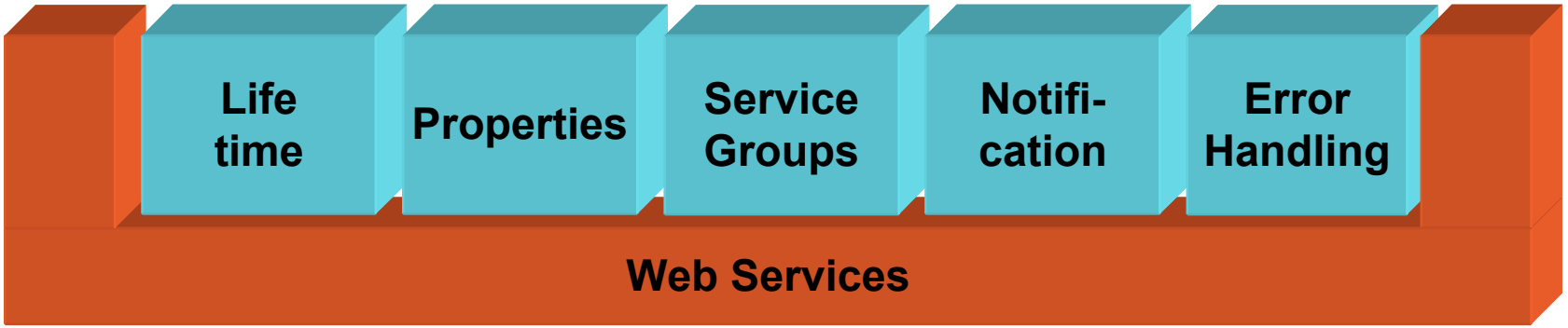
Web Service Resource Framework (WSRF)



Unified way to model and interact with stateful web services



Web Service Resource Framework (WSRF)



Unified way to model and interact with stateful web services

Lifetime (WS-ResourceLifetime)

- Factory based dynamic creation of services
- Instances are created with a limited lifetime
- Prevent services from consuming resource indefinitely (“Garbage Collection”)

Properties (WS-ResourceProperties)

- Defines type and values of a resource state

Service groups (WS-ServiceGroups)

- Collection of grid services (e.g. all resources of a cluster)
- E.g. to distribute an action to a set of services

Notification (WS-Notification)

- Notification about state changes
- Applies traditional publish/subscribe paradigm

Error Handling (WS-BaseFaults)

- Defines base handling of communication errors

Advantages and Disadvantages

Advantages

- Clear Description of Resources and Interfaces
- Dynamic sharing of resources
- On-demand services exploitation
- Cross-organizations trusted environment
- Widely accepted Web Service standards

Disadvantages

- Reference implementations are still in development
- Several complementing specifications are in development
- Complex middleware requires maintenance and administration overhead

Where do these slides come from?

These slides elaborates ideas taken from:

- Globus alliance: www.globus.org
- Cern lab: gridcafe.web.cern.ch/
- Diligent project: www.diligentproject.org
- Dave Snelling talk at EGEE 2006 conference: http://egee-technical.web.cern.ch/egee-technical/conferences/EGEE06

Further readings

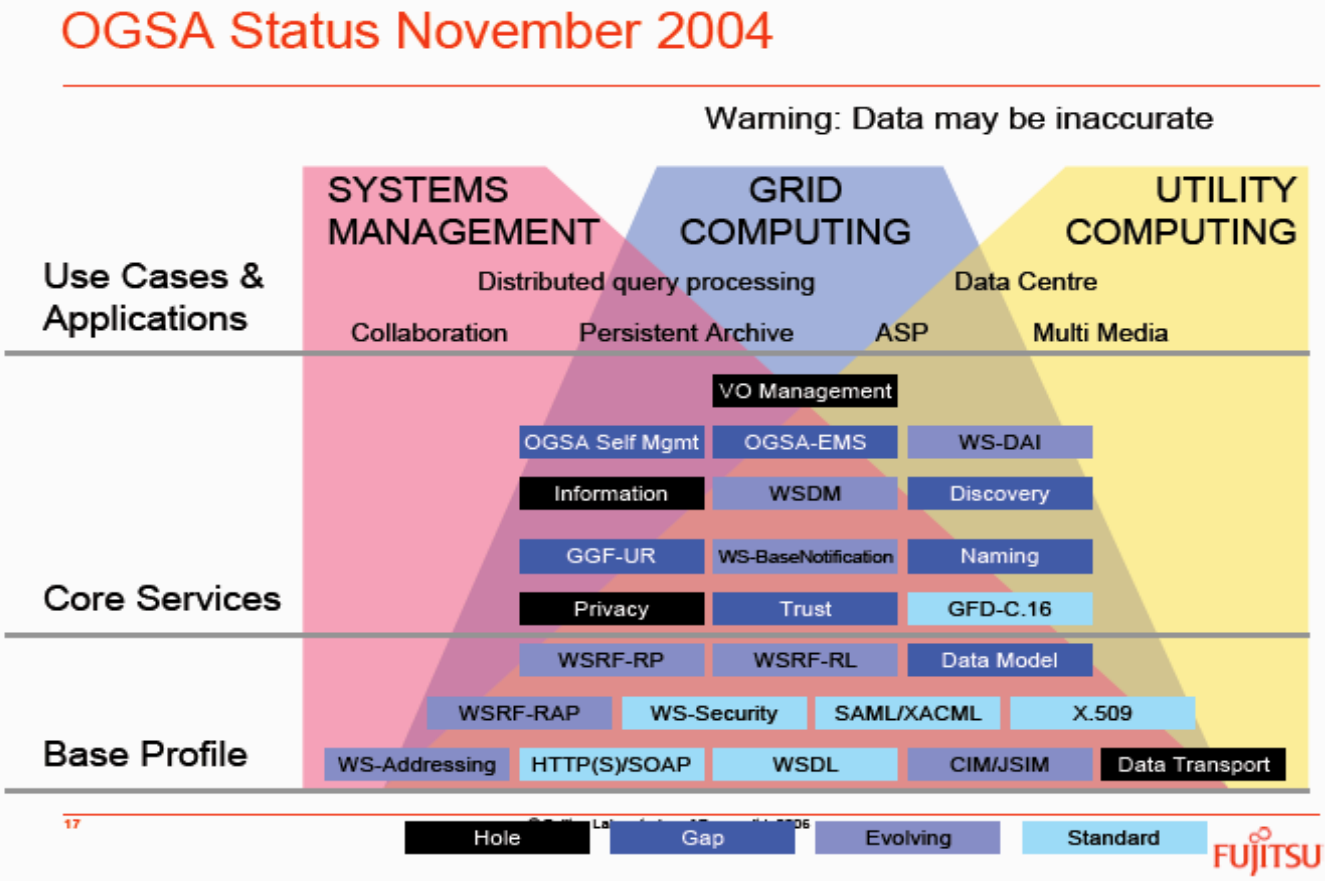
- F. Berman, G. Fox, A. Hey (Eds.). **Grid Computing – Making the Global Infrastructure a Reality**. John Wiley & Sons, 2003, ISBN: 0-470-85319-0.
- I. Foster, C. Kesselman (Eds.). **The Grid: Blueprint for a New Computing Infrastructure**. Morgan Kaufmann Publishers, 2003. ISBN: 1-55860-933-4
- I. Foster, C. Kesselman, J. Nick, S. Tuecke. **The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration**. Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.
- I. Foster, C. Kesselman, S. Tuecke. **The Anatomy of the Grid: Enabling Scalable Virtual Organizations**. International J. Supercomputer Applications, 15(3), 2001.
- I. Foster. **What is the Grid? A Three Point Checklist**. GRIDToday, July 20, 2002.
- I. Foster and A. Iamnitchi**. On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing**. IPTPS '03

Complementing specifications

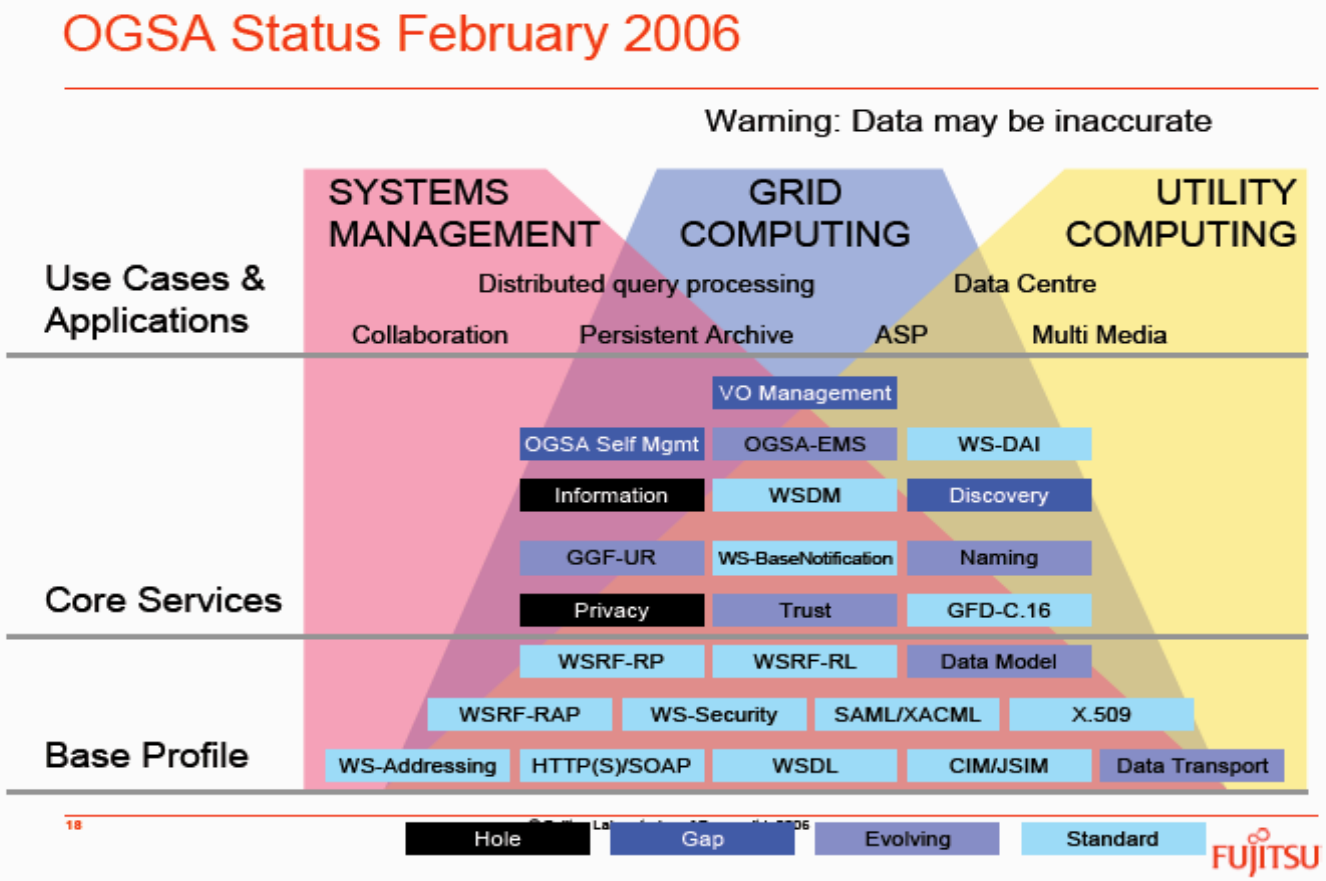
Concept	OGSI	WSRF	WS-M	RW-RT	OGSA-*
Resource		√	√	√	√
Properties	√	√	√	√	√
Notification	√	√	√		√
Lifecycle	√	√	√	√	√
Composition	√	√	√	√	√
Faults	√	√			√
Collections	√	√	√		√
Naming	√				√

Orange = Historical, Blue = Evolving, Green = Standard

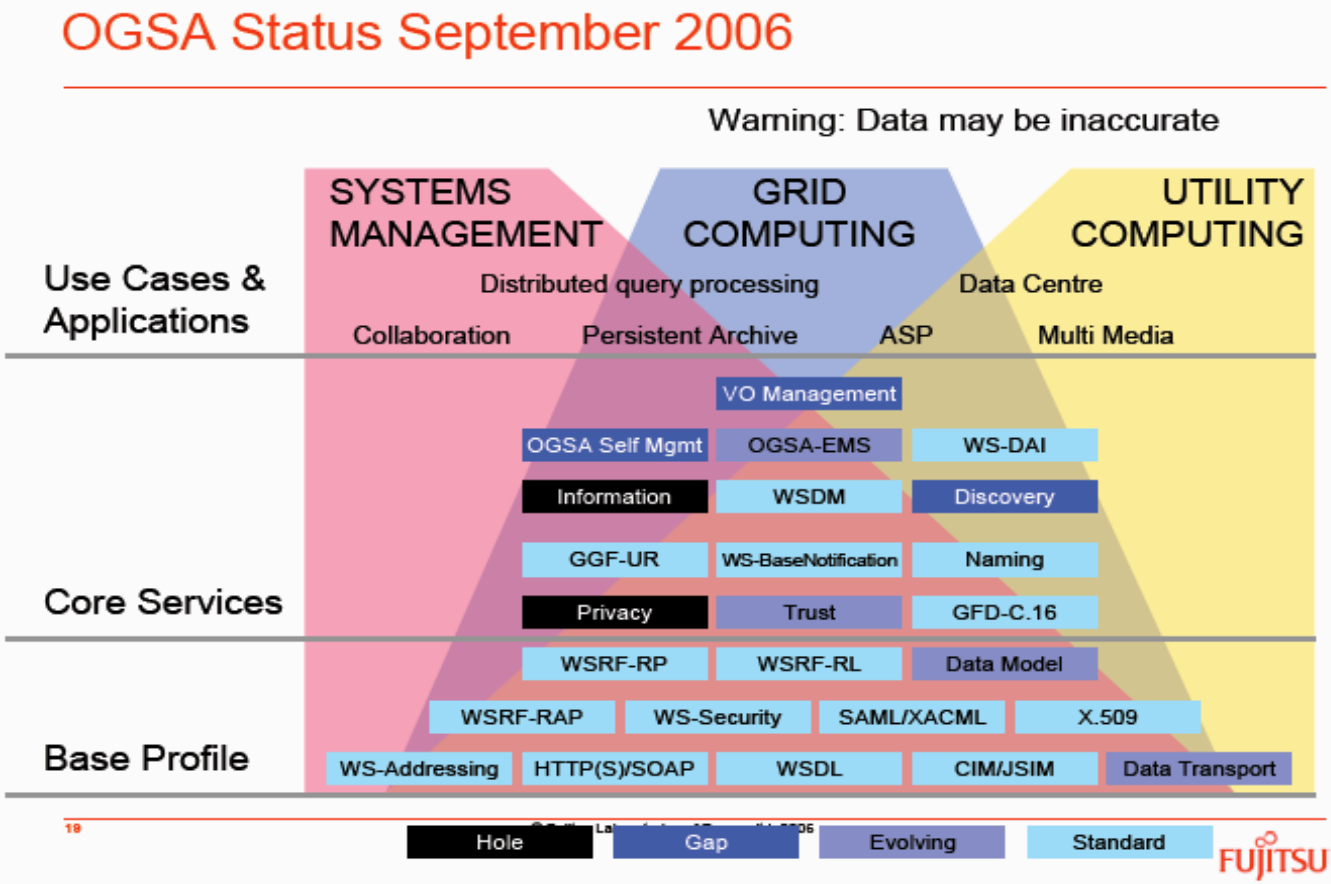
Specifications Evolution



Specifications Evolution



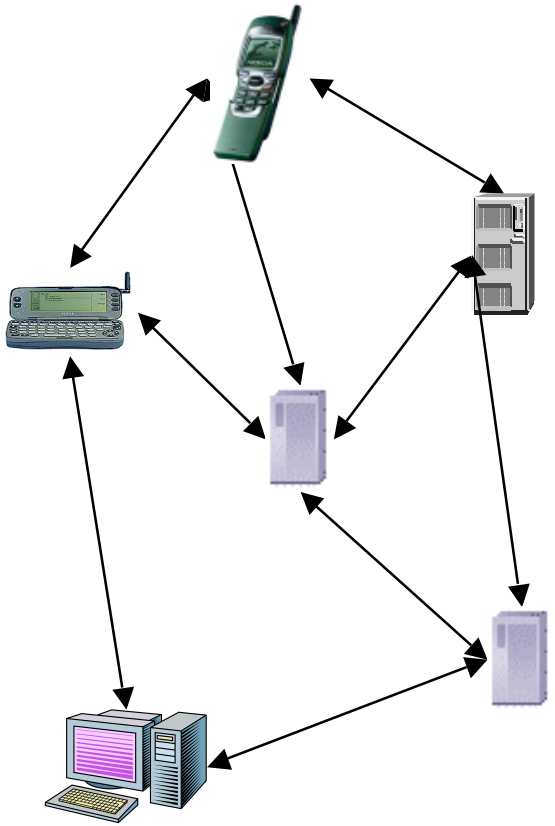
Specifications Evolution



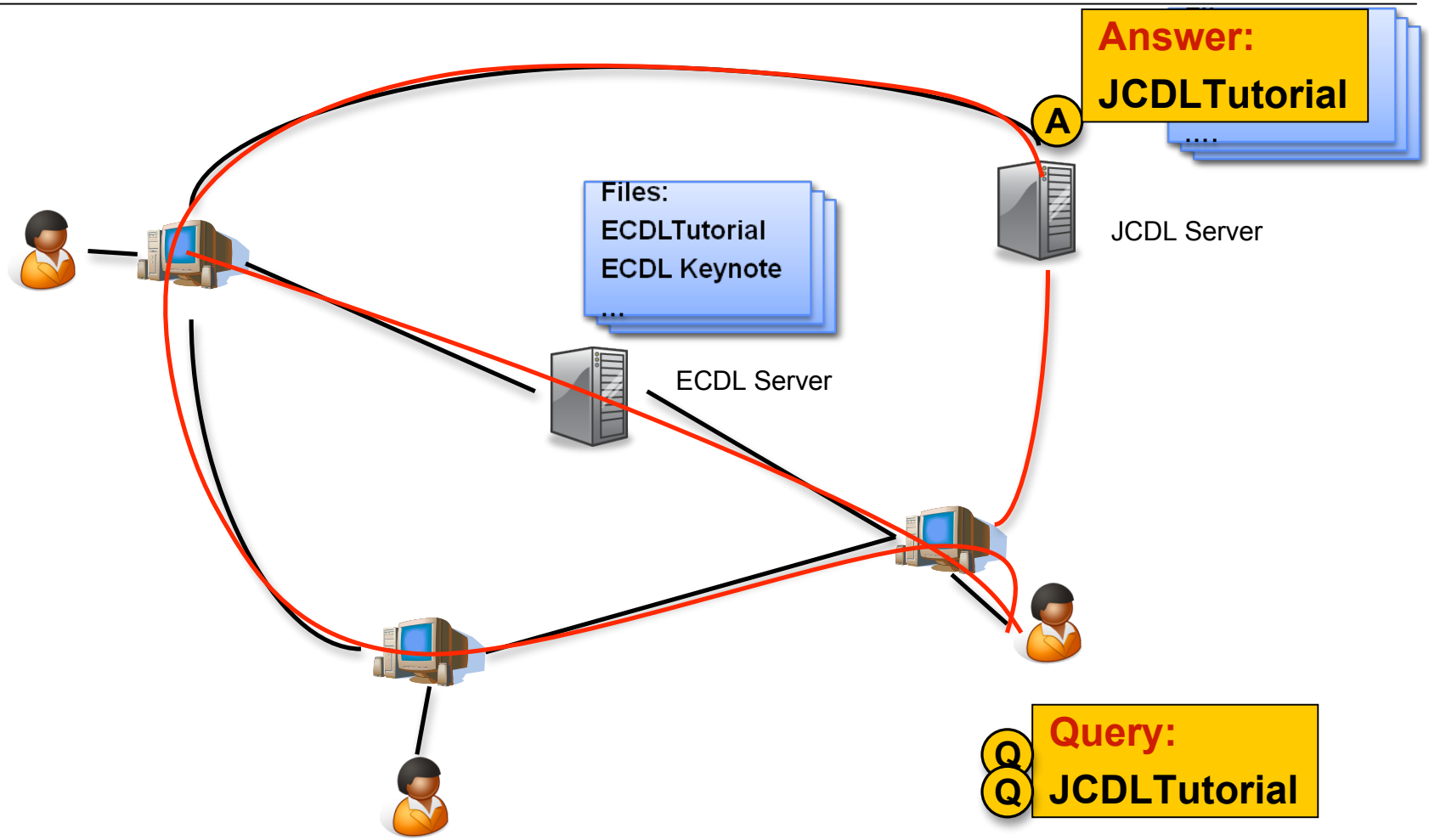
Peer-to-Peer Computing

The Nature of P2P

- Peers act as client and server
  - Peers communicate directly
  - Both provide services, e.g. storage, calculations
- Peers are autonomous
  - Owner decide e.g. when a service is active
- No central administration
- No global system view
  - Peers have only some knowledge about their neighborhood
- Highly dynamic
  - Peers appear and disappear whenever they like
- Unreliable connections
- Mostly self organizing
  - Services and clients are actively looking for appropriate partners



Simple P2P File Sharing Example



Some History

P2P receives a lot of attention ...

- File sharing systems
  - ž Gnutella, Freenet, Napster, ...

... but P2P is nothing new

- Telephone systems
- Federated databases
- ARPAnet
  - ž First networks were P2P systems
- Usenet, Domain Name Service
  - ž Servers communicate in a P2P manner
- Service Oriented Architectures

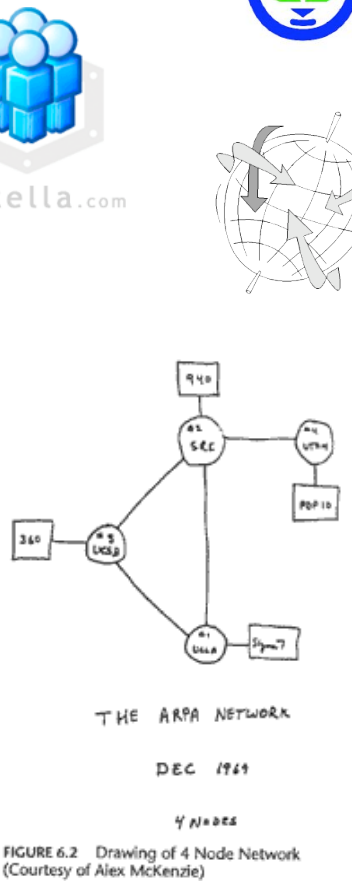


FIGURE 6.2 Drawing of 4 Node Network (Courtesy of Alex McKenzie)

Advantages of Peer-to-Peer

- Scalability
  - ž P2P solutions should be scalable by definition
- Availability of information, services, …
  - ž Redundancy increases the availability
- Flexibility
  - ž Flexible reaction on changes,
    e.g. leaving of peers, changing from
    wire to wireless communication
- Low administration costs
  - ž Self organization requires no central administration

Disadvantages

- Advantages grow with the number of peers
  - To accomplish the advantages many peers are necessary

- Software has to deal with P2P properties
  - Some P2P properties (e.g. dynamicity) are challenging for the development
  - Software components can provide functions to other peers as a service (e.g. Microsoft .Net)

Application Domains

- Resource sharing
  - Joined usage of unutilized resources (e.g. CPU, storage)

- Content sharing
  - Sharing of files (music, video), calendar, spam sender data, …
  - News distribution
  - Searching for information
  - Examples: Gnutella, Freenet, Napster, Groove, Hive, Cloudmark SpamNet

- Collaboration
  - Peers work together
    - Collaboration within the project team, e.g. document creation
    - Instant Messaging
  - Benefits
    - Autonomous
    - Offline working
    - No administration
  - Examples: Groove (Collaboration), Skype (Messaging), …

Challenge: Data Access

Data Access Structures

- Methods to query and access „nomadic data”
- Partial knowledge

Simple approaches

- Breadth-first-search (Gnutella)
- Depth-first-search (Freenet)

Disadvantage

- High accumulated bandwidth

More sophisticated approaches

- Use global unique identifier (GUID) of object for routing,
  (DHT, Plaxton Routing, Chord, Tapestry, Pastry, CAN, P-Grid)

Summary

Web Services

- Well established standard for interoperabel services
- Lightweight communication protocol
- Easy to use
- Depends partly on central management services
  → Fully decentralized Web Services are developed in BRICKS

GRID Computing

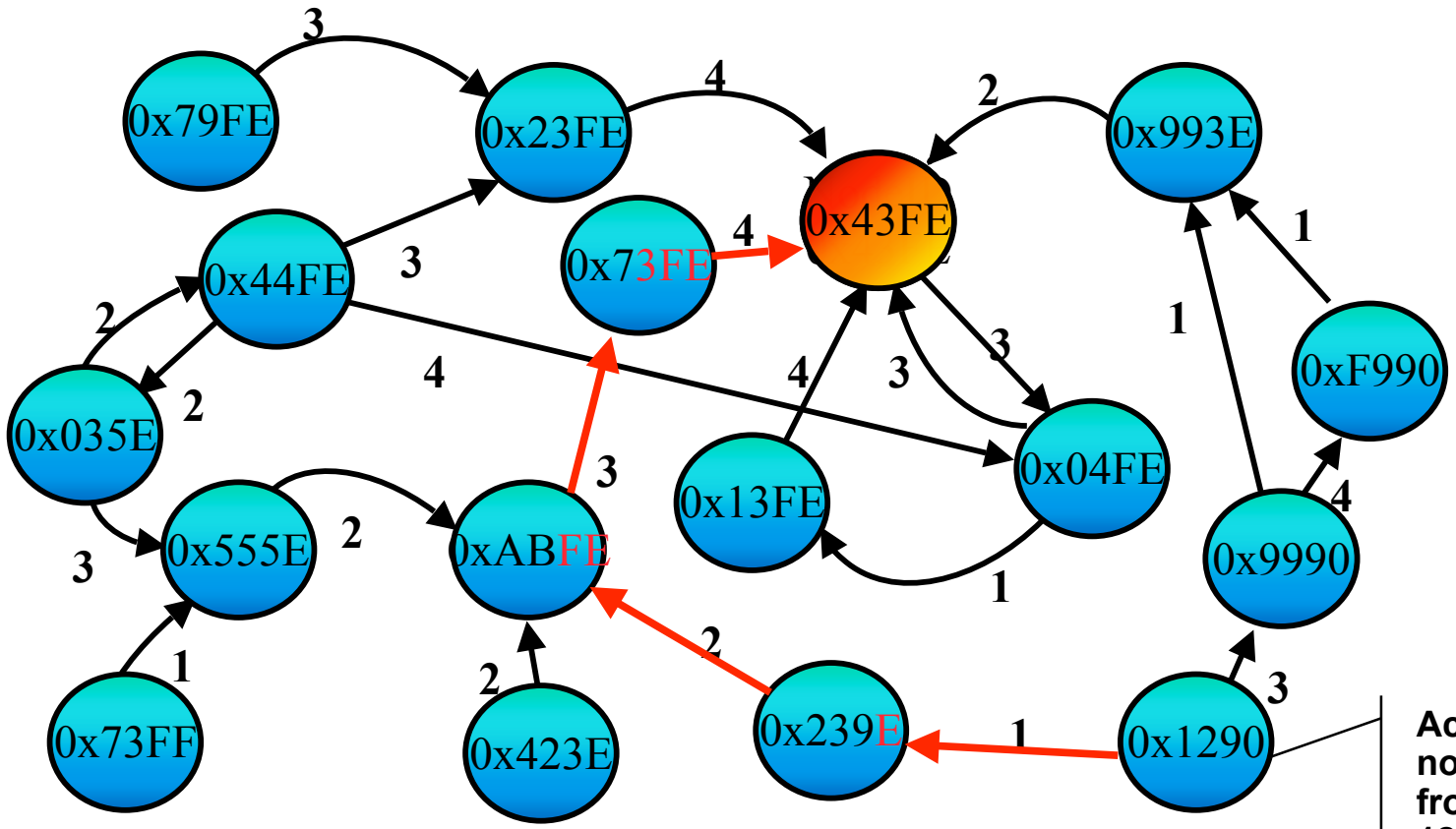
- More generalized
- Extends the service approach to virtualize resources
- Allows flexible resource usage

Peer-to-Peer

- Describes the way of communication between entities
- Self-organizing system of autonomous entities
- Highly flexible and scalable

Questions & Discussions

Example DHT Routing

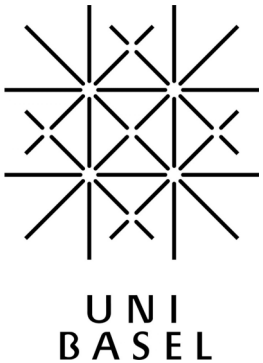


- Suffix Routing → *Neighbor-map* with *i* levels
- Nodes on level *i* have *i-1* equal digits

Building Digital Libraries on Service Oriented Architectures

Solutions for decentralized DL infrastructures: issues, solutions and advantages in the framework of BRICKS

Tutorial at JCDL 2007
June, 19th 2006



Thomas Risse (L3S)

Agenda

09:00 – 09:30 **Introduction: Motivation & Challenges**

09:30 – 10:00 **Challenges of bringing DL to distributed Infrastructures**

10:00 – 10:30 **Underlying Technologies and their promises** (SOA, P2P, Grid)

10:30 – 10:45 Coffee break

10:45 – 12:00 **Solutions for decentralized DL infrastructures** (with BRICKS Demos)

12:00 – 12:30 **DelosDLMS - the DELOS Digital Library Management System**

12:30 – 13:30 Lunch

13:30 – 14:00 **DelosDLMS Demos**

14:00 – 15:00 **Building DL services on the Grid** (DILIGENT)

15:00 – 15:30 Coffee

15:30 – 16:45 **DILIGENT Demos**

16.45 – 17:00 **Conclusions and future directions**

Overview

- The BRICKS Project
- BRICKS Workspace Demo
- BRICKS Content Management - Brief Overview
- BRICKS Metadata Management
- BRICKS Collection Management
  - or - How to navigate through the BRICKS information space?
- BRICKS Application Demo
  Archaeological Sites Application Prototype

The BRICKS Project

http://www.brickscommunity.org/

BRICKS - Project Identity Card

- Project Acronym: BRICKS - Building Resources for Integrated Cultural Knowledge Services
- Consortium: 24 organizations from 9 countries
- Thematic area: Digital Libraries Services
- Duration: 42 months
- Started in January 2004
- Budget: 12,2 Mill. Euro
- Homepage: http://www.brickscommunity.org/
- Technical Partners: Engineering, Italy; Fraunhofer IPSI; CNR-ISTI, Italy; Metaware, Italy; ARC Seibersdorf Research GmbH, Austria; EPFL, Switzerland; Canoo, Switzerland; Uni. of Athens, Greece; Uni. of Sheffield, UK; Scuola Normale Superiore di Pisa, Italy; Uni. of Florence, Italy; Oxford ArchDigital, UK; PolyDisplay, Norway
- User Partners: Italian Ministry of Culture, Italy; Vatican Secret Archives; Uffizi Gallery, Italy; Schönbrunn Palace, Austria; Austrian National Library, Austria; European Museum Forum, UK; Museum of Cycladic Art, Greece; Russian Cultural Heritage Network, Russia; Museums, Libraries and Archives Council, UK; Studio Azzurro, Italy

Access to professional cultural resources today

Situation

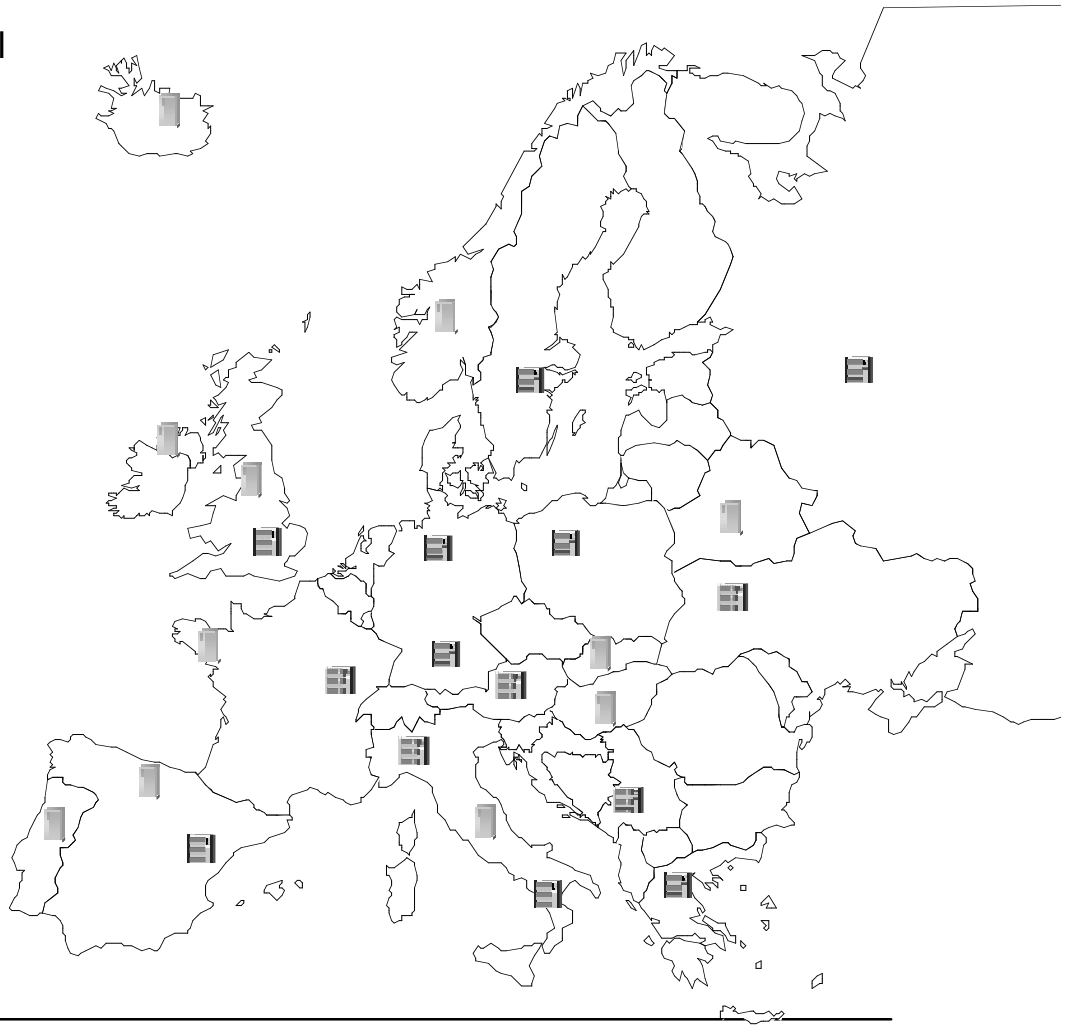
- Large number of independent distributed digital information sources
- Often professional sources are not public
- Still a large number of interesting sources are missing, e.g. small museums

Find and access information sources (examples)

- Search engines, e.g. CiteSeer, DBLP, Google
- Z39.50 for some libraries
- Personal Knowledge
- …

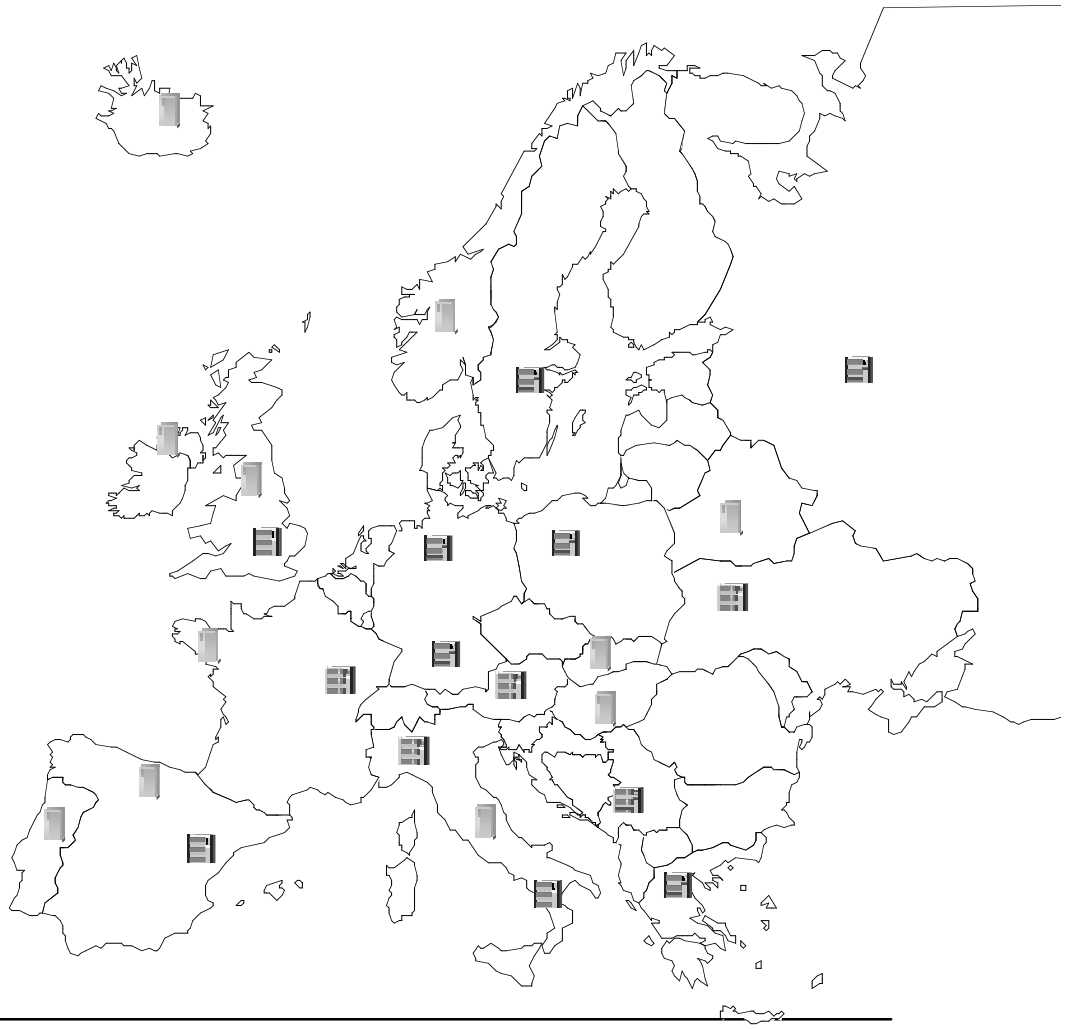
Disadvantages

- Time consuming
- Limited openness
- Seldom collaboration support
- Seldom language support



Overall Goals of BRICKS

- Transparent access to distributed available information sources
- Retrieval of information with knowledge support
- Multi-lingual
- Protection of intellectual properties
- Low Cost
- Easy installation and maintenance
- Platform-independent



BRICKS Approach

Service oriented

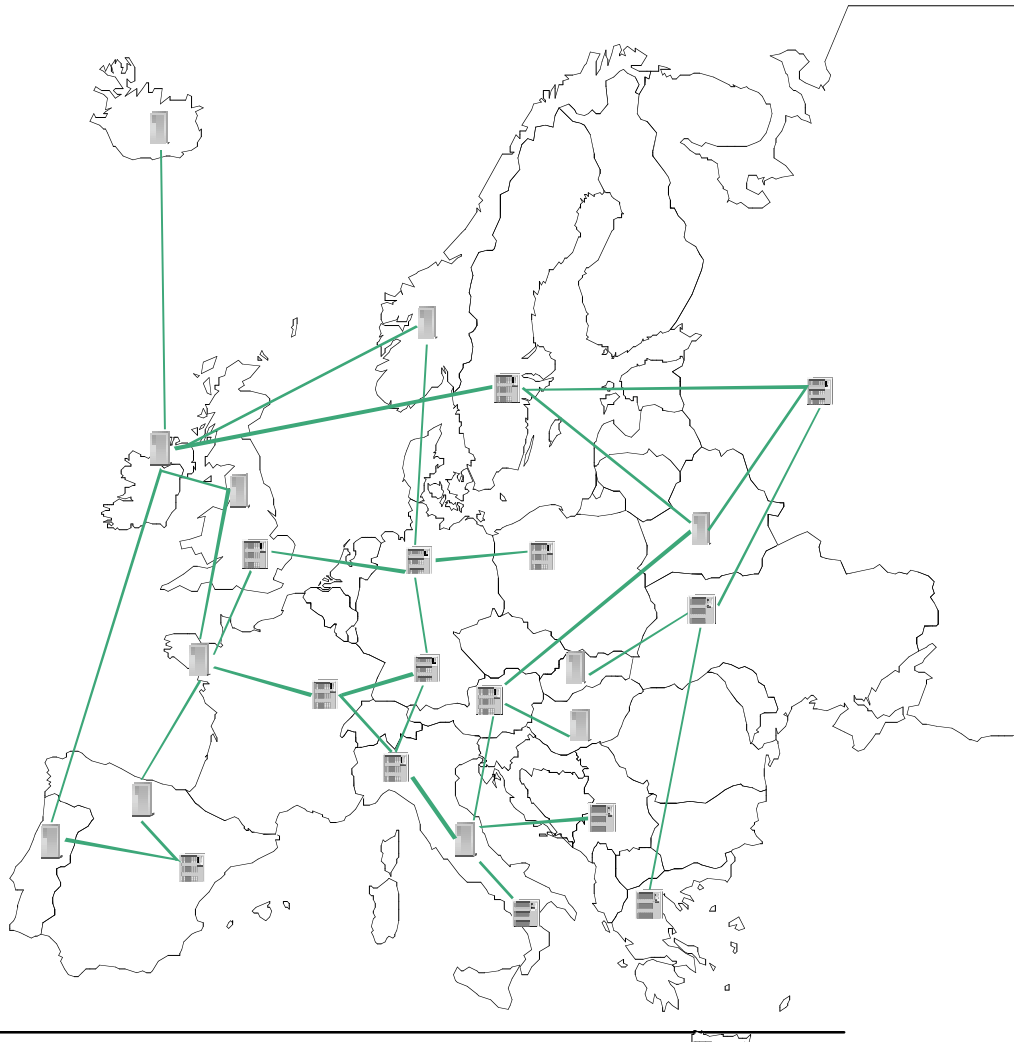
- Standardized interface descriptions based on Web Services
- Platform independent
- Flexible composition of services

Decentralized Organization

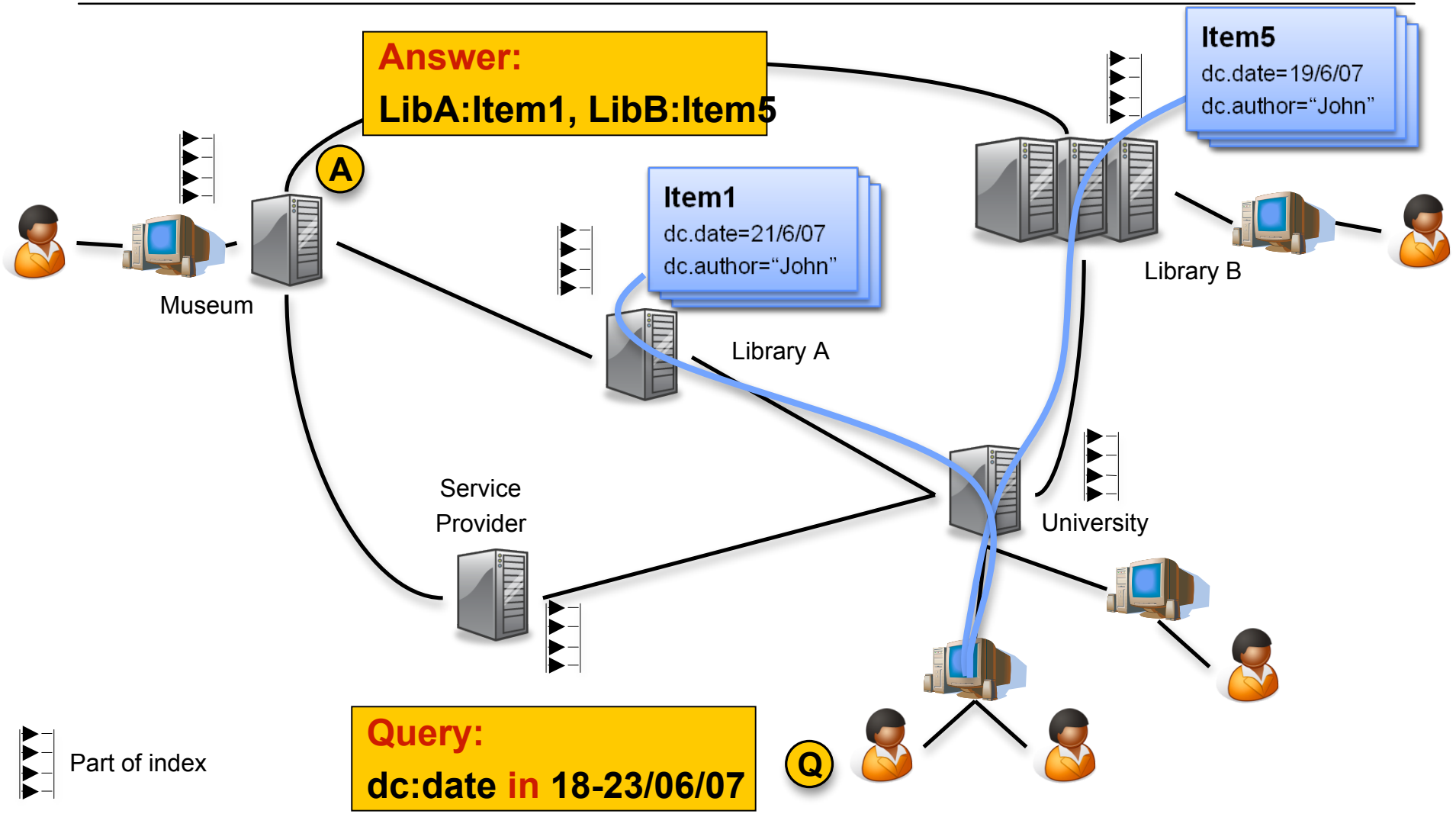
- Avoid central coordination
  → No Single Point of Failure
  → No central maintenance organization necessary
- Highly scalable
- Increased reliability
- Minimized maintenance cost

Usage & Costs

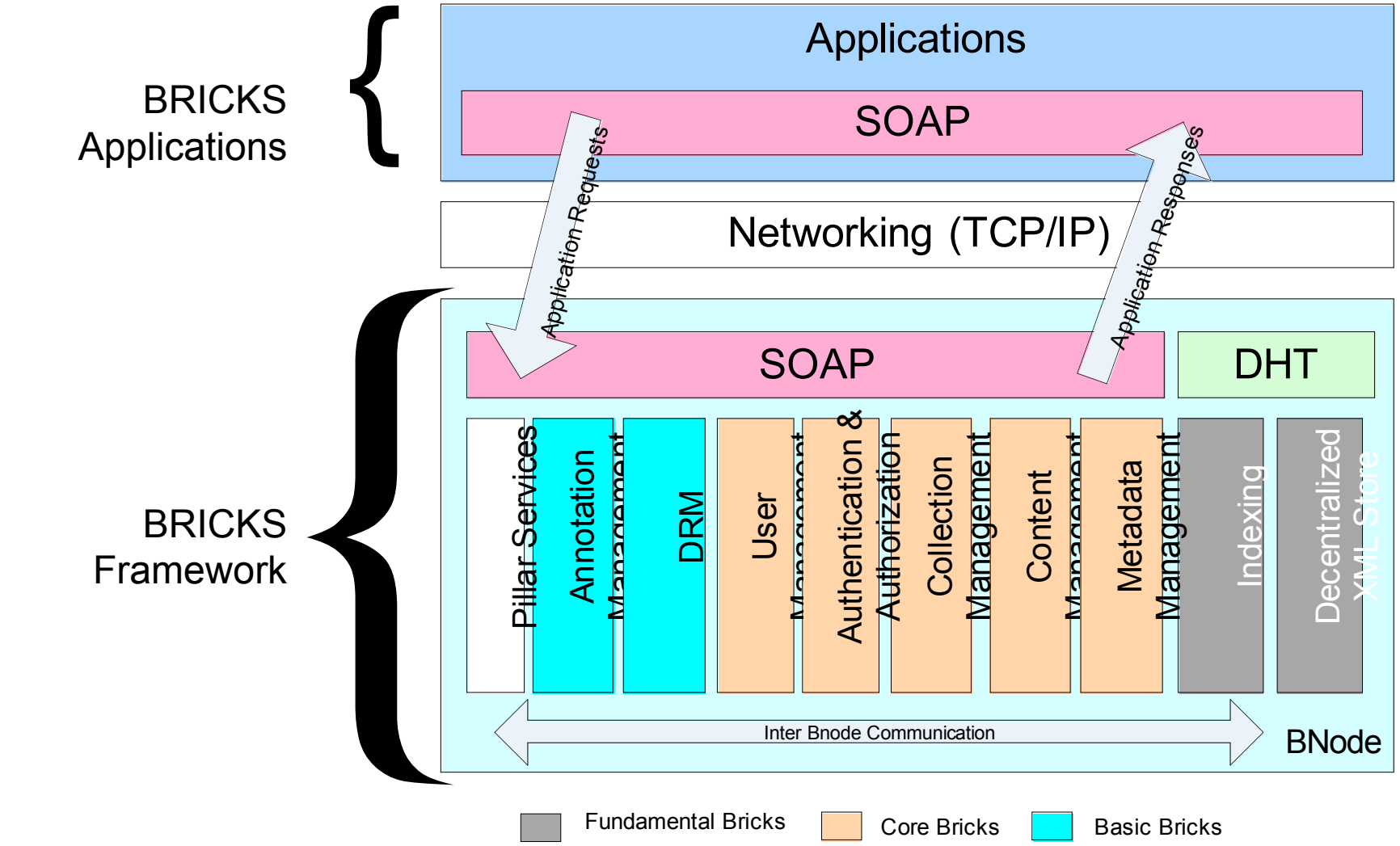
- Easy installation
- Open Source lowers the barrier to use BRICKS
- Allows tailoring to user needs



BRICKS Node (BNode) Access



BNode Architecture



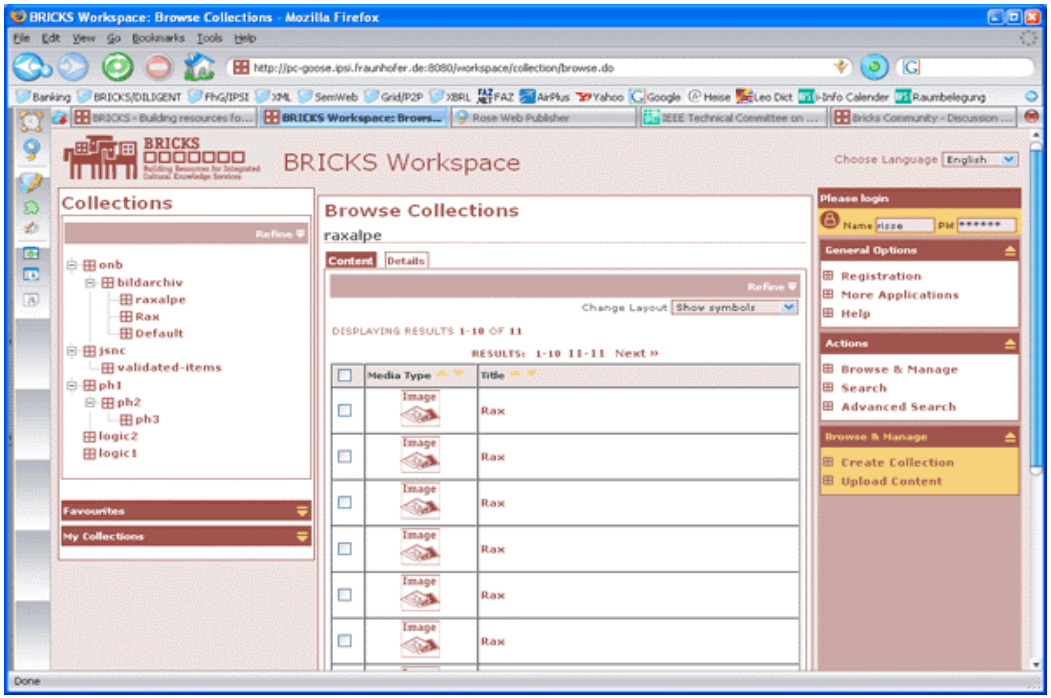
BRICKS
Workspace Demo

The BRICKS Workspace

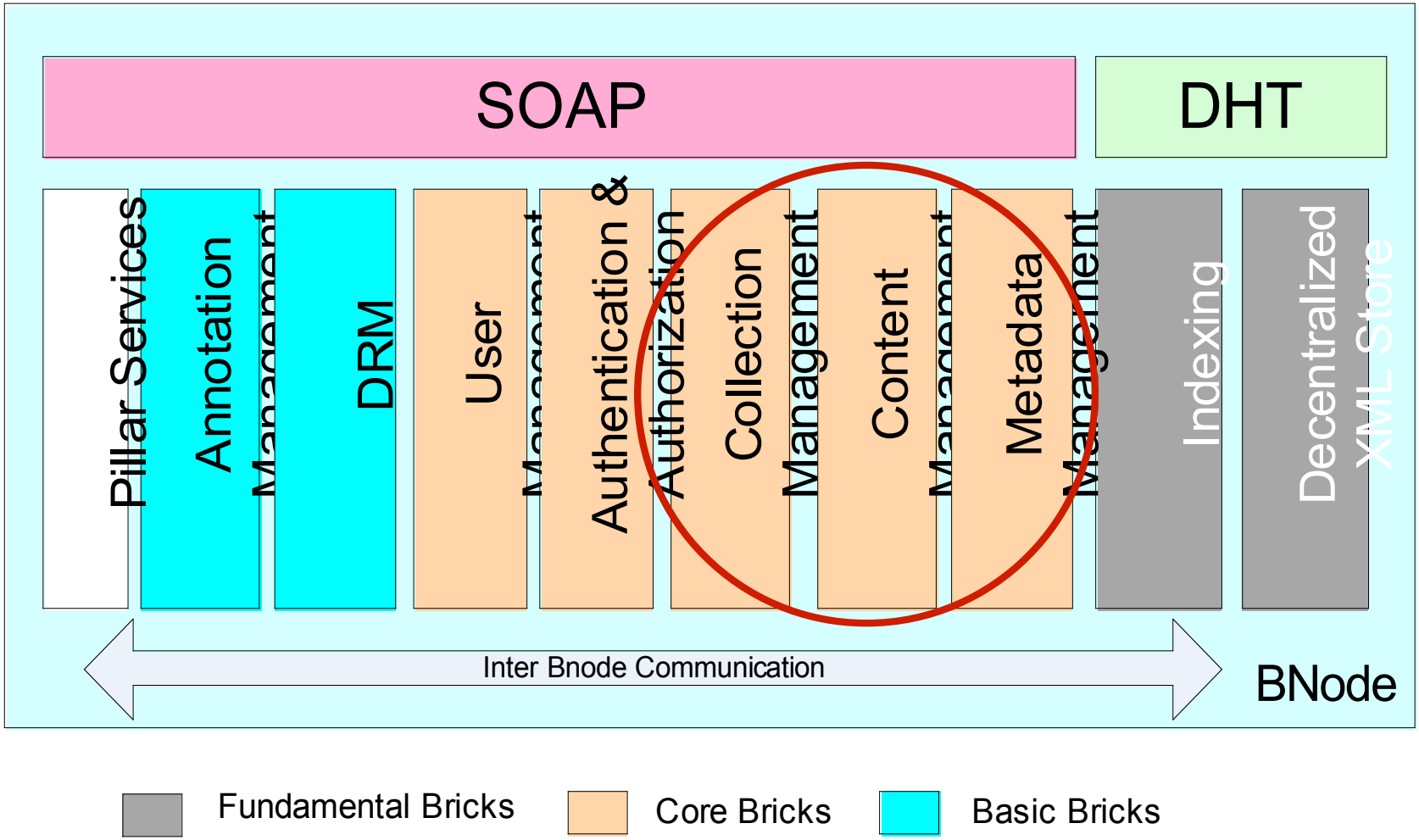
- What does it demonstrate?
  - A web application (thin client) accessing BRICKS Foundation services
  - Navigation through the BRICKS information space

- Target audience
  - General end-users (citizens)
  - Application developers (as an example)

- Alternatives
  - BRICKS Desktop
    An ECLIPSE based native user interface
  - Domain & Application specific interfaces



Some more BRICKS Framework Details



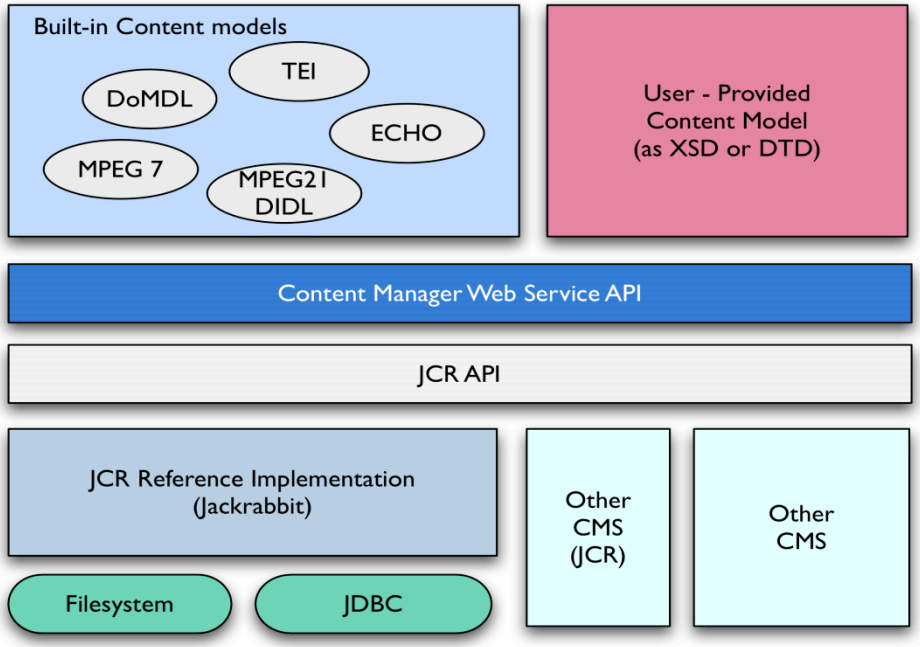
BRICKS

Content Management

Brief Overview

BRICKS Content Management

- BRICKS is based on the Java Content Repository
- Content Repository for Java™ Technology API (JSR-170)
  - Born from the need to standardize the proprietary content repositories
  - Supports a wide range of applications
  - Provides a unified API
    - Creation and access
    - Versioning
    - Access control
    - Full text searching
  - API is independent from the Back-End storage systems, e.g. file system, WebDAV repository, XML database, SQL database
  - JCR based systems: Apache-Jackrabbit, Magnolia, Alfresco, …
  - BRICKS extends JCR with a Web Service interface
  - Provide some meta-content models, e.g. DoMDL Model, MPEG-21, MPEG-7, TEI-Lite, …



BRICKS

Metadata Management

Metadata

A lot of definitions → Always a good reason for long discussions

Professional Users (e.g. Librarians) have a concrete view

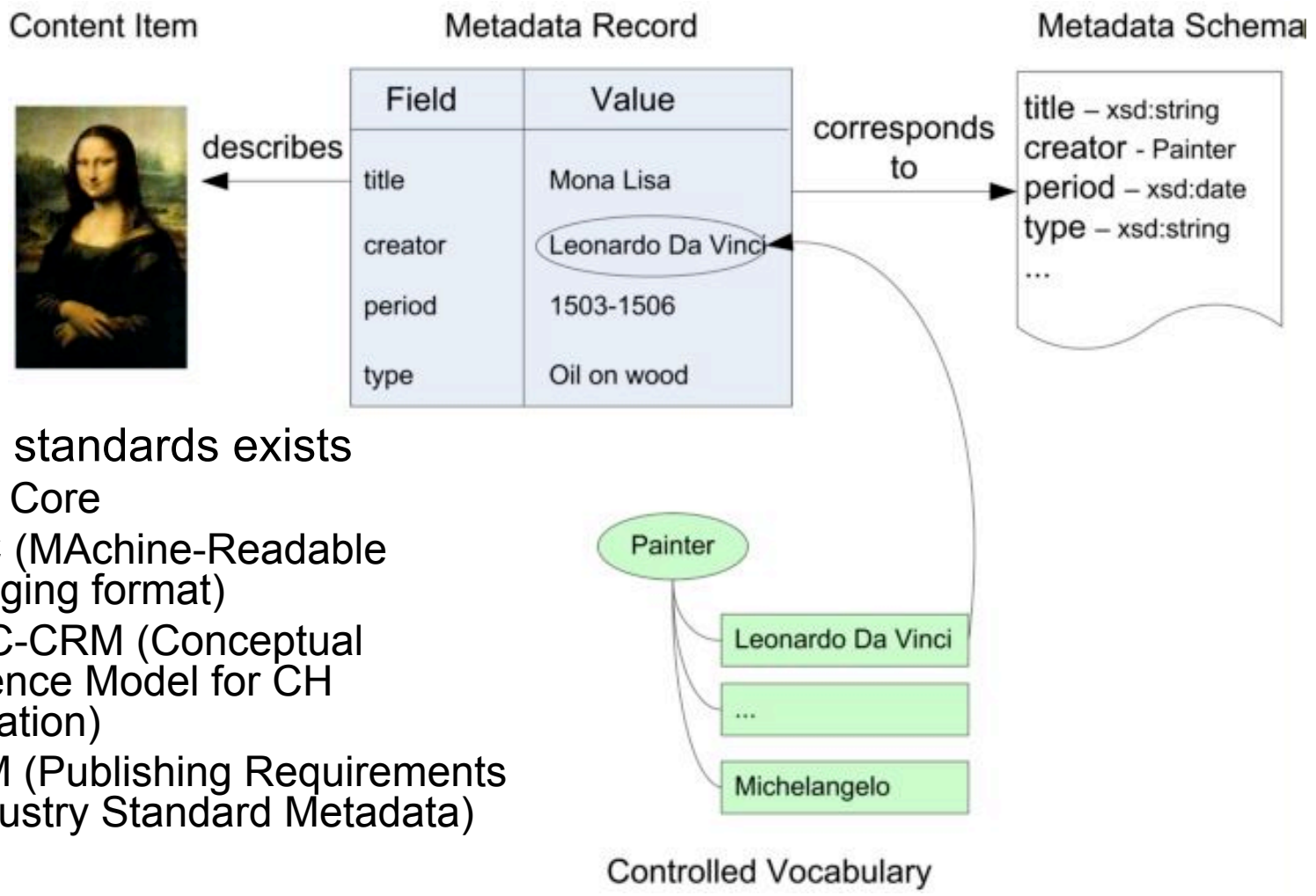
Metadata are value-added information that professional users create to arrange, describe, track and access information objects

Different Types of Metadata →

Minerva classification

Typ	Definition
Administrative	Metadata used in managing and administering information resources
Descriptive	*Metadata used to describe or identify information resources*
Preservation	Metadata related to the preservation management of information resources
Technical	Metadata related to how a system functions or metadata behave
Use	Metadata related to the level and type of use of information resources

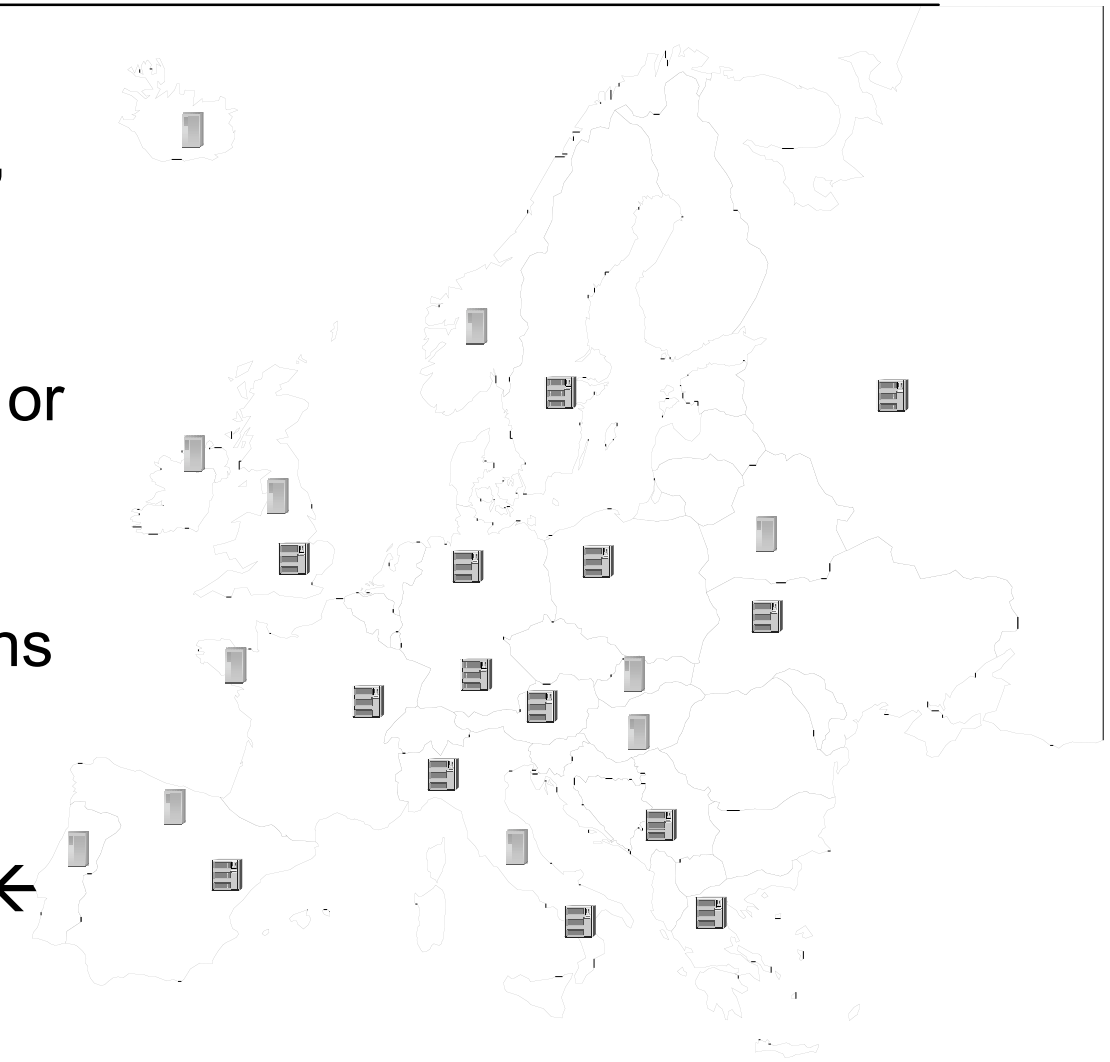
Descriptive Metadata



- Several standards exists
  - Dublin Core
  - MARC (MAchine-Readable Cataloging format)
  - CIDOC-CRM (Conceptual Reference Model for CH information)
  - PRISM (Publishing Requirements for Industry Standard Metadata)

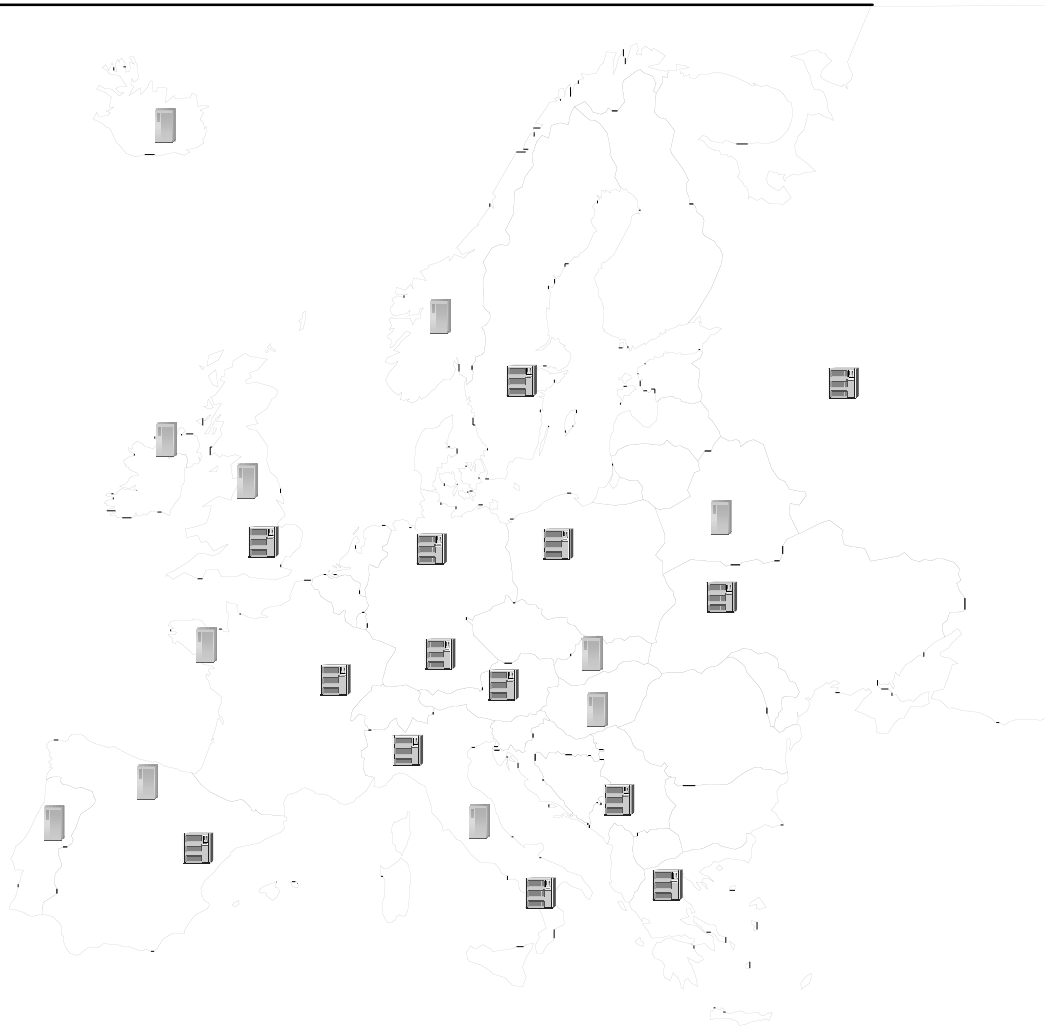
Metadata Storage Requirements

- Durability
  e.g. descriptive metadata, annotations
- Availability/Accessability
  e.g. descriptive metadata or annotations
- Scalability
  e.g. for service descriptions or descriptive metadata
- Consistency
  e.g. descriptive metadata ← → content



Metadata Storage Modes (1/2)

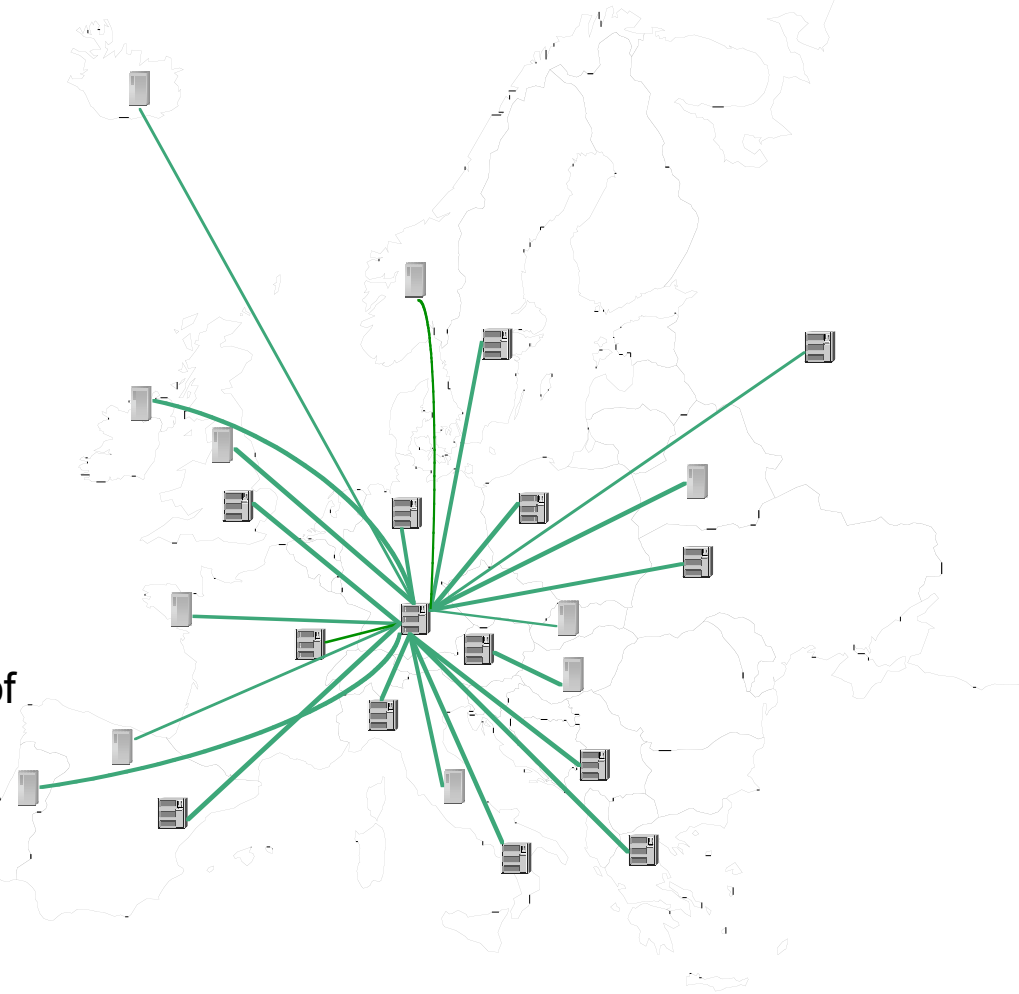
- Local Storage
  - Advantage: simple
  - Disadvantage: no guarantee of 100% availability



Metadata Storage Modes (1/2)

- Local Storage
  - Advantage: simple, well-known, …
  - Disadvantage: no guarantee of 100% availability

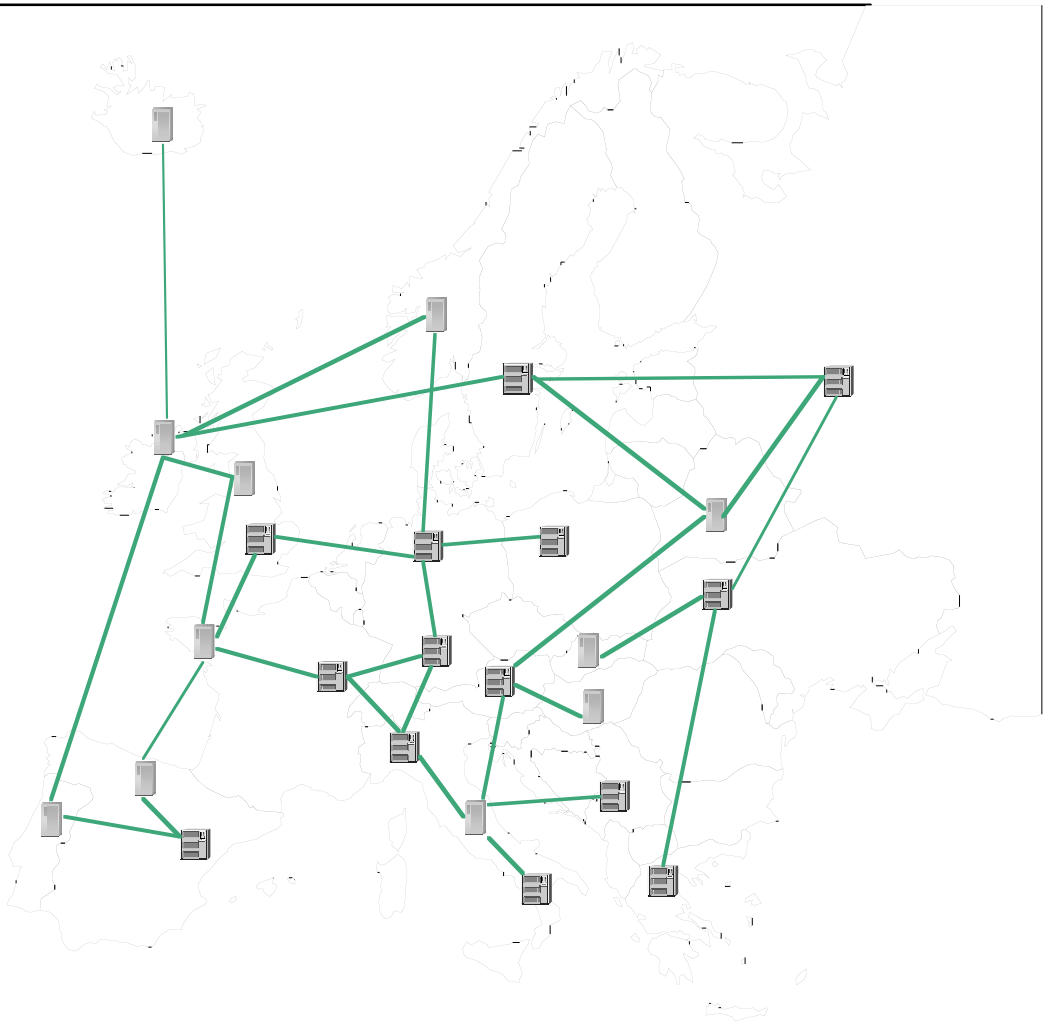
- Central Storage
  - Advantage
    - ž Well-known
    - ž Easy to implement transaction guarantees
  - Disadvantage
    - ž Scalability problems for large volume of data and requests
    - ž High concentration of resources at one place (bandwidth, space, CPU)
    - ž High costs
    - ž Central point of failure



Metadata Storage Modes (2/2)

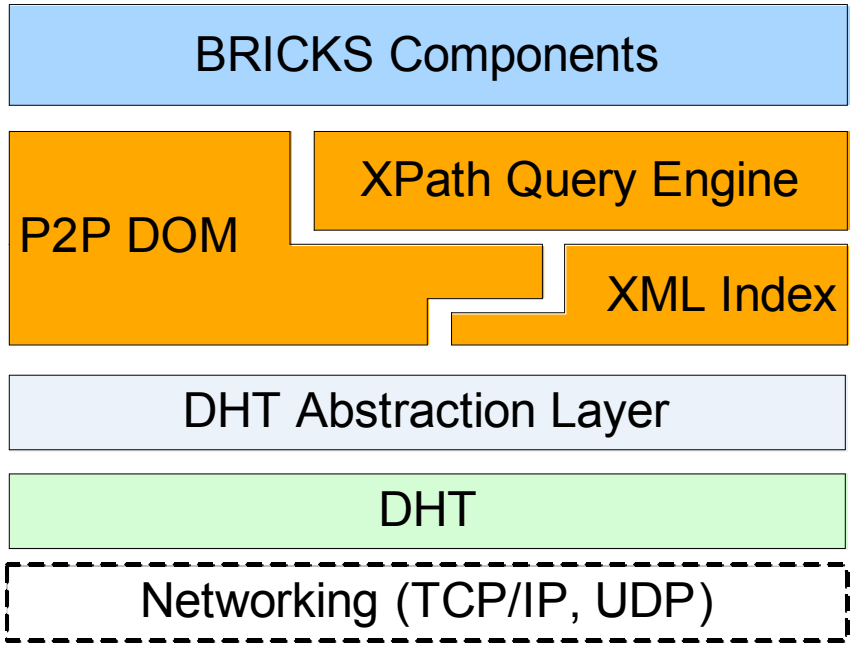
Decentralized Storage

- Advantage
  - ž High scalability
  - ž Fault tolerance, no central point of failure
  - ž Better usage of available resources
  - ž Decentralized architecture → decentral storage
  - ž Transparent access
- Disadvantage
  - ž Slower access → an index speed it up in many cases
  - ž Complicated logic behind
  - ž No transaction guarantees



Decentralized Storage in BRICKS

- Used for administrative metadata
- Data spread within BRICKS community, but transparent access to users
- Uses P2P layer for BNode communication
  - Discovering neighboring peers
  - Routing and processing messages within BRICKS community, but without global topology knowledge
- Implements
  - Well-known W3C DOM API for creating and accessing XML documents
  - XPath language for querying XML documents
- Built-in protocols for
  - Maintaining high data availability through replication
  - Concurrency and consistency control



Storage Locations of Metadata Types in BRICKS

- Local storage
  - Descriptive metadata (with decentralized index)
  - Technical (EXIF, etc.)
  - Security metadata (ACL, etc.)
  - Annotations
  - Ontologies (with decentralized discovery)

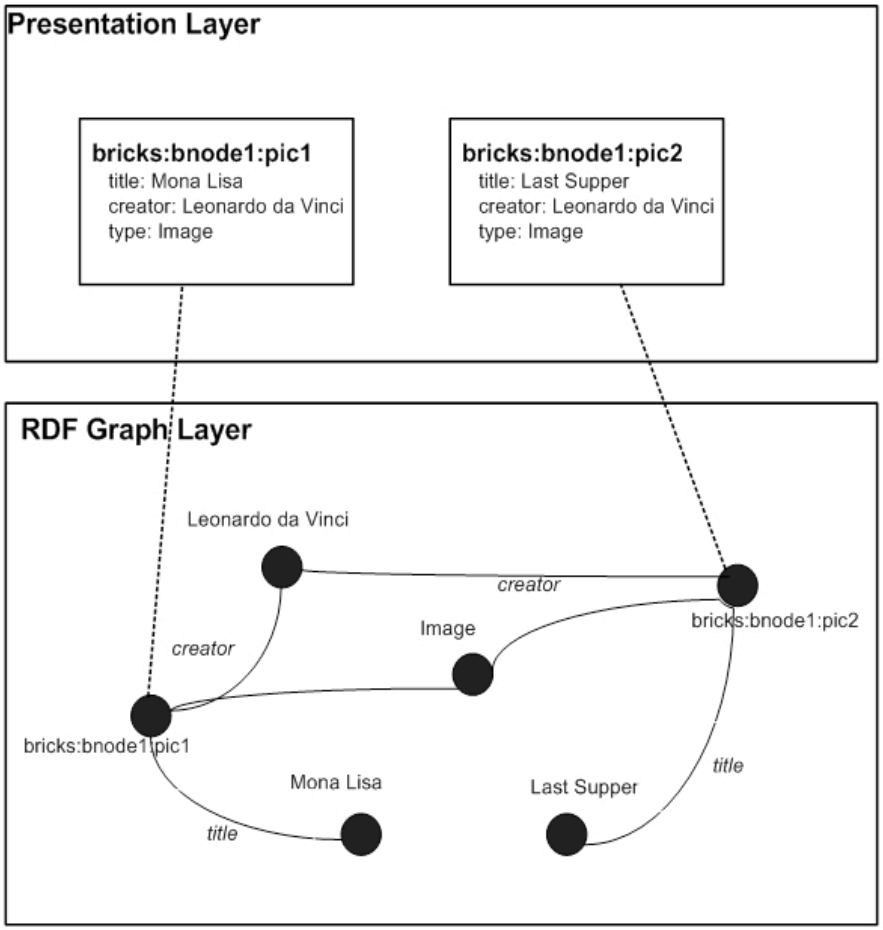
- Decentralized
  - Service descriptions
  - Collection descriptions

Descriptive Metadata in BRICKS

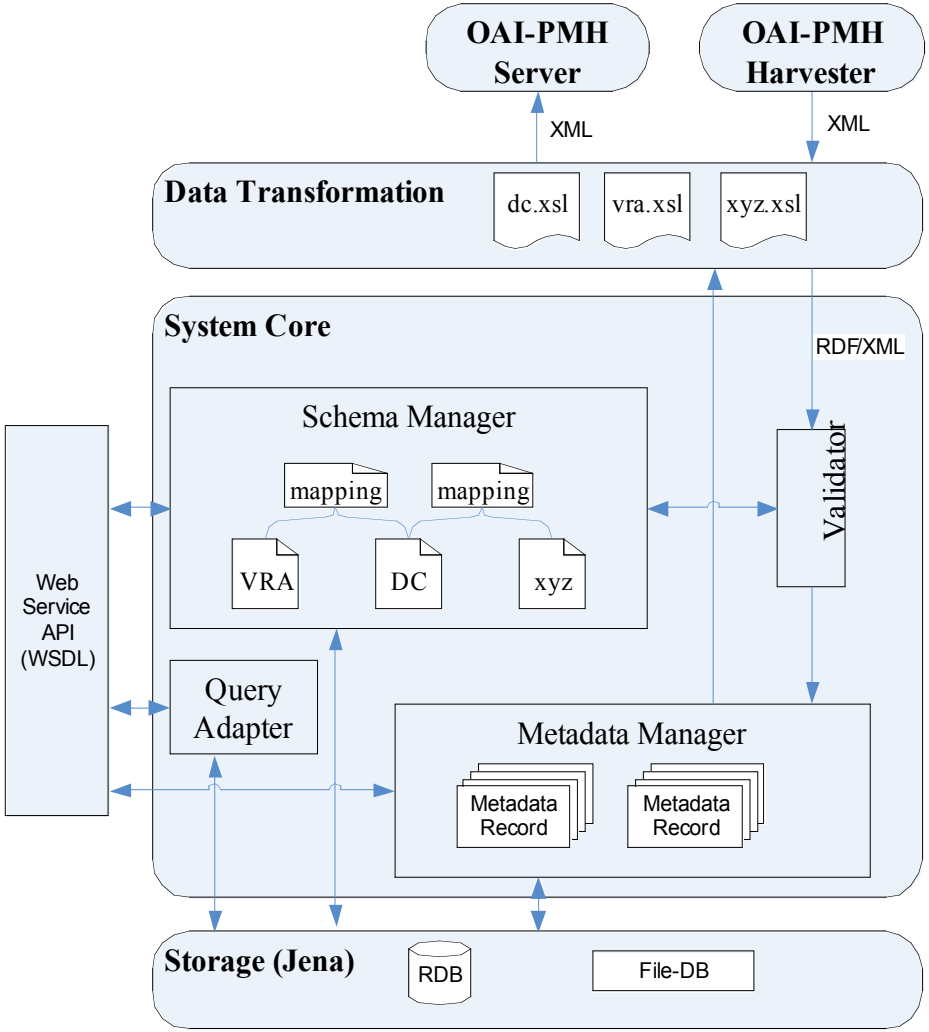
- The BRICKS **Metadata Manager**
  - is responsible for managing cultural assets from various institutions
  - must support arbitrary metadata formats from various institutions
- Schema/Format definition
  - the semantics of records is modeled and exposed in OWL-DL
- Bibliographic records
  - are internally stored and represented in RDF
- Controlled vocabularies, Thesauri, etc.
  - supported if they are represented in RDF, RDFS or OWL

Metadata Manager - Design Decisions

- RDF and OWL
  - are well-suited for handling heterogeneous metadata internally (!)
- But:
  - do NOT show triples or anything else related to RDF or OWL to the user or front end developers!
  - nobody wants to obtain triples in a query result
- In BRICKS we've decided to
  - completely hide RDF and OWL from users and high-level applications
  - expose it only for machines



Architecture of the BRICKS Metadata Manager



BRICKS and Existing Systems

- The BRICKS idea is not to replace but to integrate with existing systems
- Rely on already accepted protocols in the Digital Libraries domain to tap existing metadata and content databases
- Currently available:
  - import of “existing” metadata via the OAI protocol for metadata harvesting (OAI-PMH)
    - easy to understand
    - easy to use and minimal implementation effort for institutions
    - gives us a minimum level of interoperability (Simple Dublin Core)
  - Wrapping of SRU sources in under development

BRICKS Collection Management

or

How to navigate through the BRICKS information space?

Outline

- What is a collection manager
- Requirements on collections
- Implementing the collection management
- Searching a distributed DL
- Implementing the search mechanism

Outline

- What is a collection manager
- Requirements on collections
- Implementing the collection management
- Searching a distributed DL
- Implementing the search mechanism

What is a Collection Manager

- A Collection Manager is a mediator between:
  - the applications
  - and the DL contents.

- Applications may range from GUIs to arbitrarily complex programs for performing domain specific tasks.

- DL contents are the digital objects stored in the DL and their associated information.

- The job of the CM is:
  - to represent the DL contents via a conceptual model that is understandable and intuitive from an application viewpoint
  - to offer primitives for the manipulation of this model in support of the application tasks.

The DL Conceptual Model

- We can view the contents of a DL as a set of documents, that is multimedia complex objects, with associated information, which supports the basic services offered on these objects.

- Distinctive features:
  - the size is typically $O(10^3)$ - $O(10^6)$
  - the members are very heterogeneous, *e.g.* in
  - Structure
  - Types
  - Language
  - Format
  - Contents

- *How do we go about taming such a set?*

The DL Conceptual Model

- The problem is not new: organization!

> *"Hence we view the organization of a digital library network as basically an abstraction mechanism in terms of which details from a lower level of representation are suppressed. This is a crucial issue when dealing with large digital libraries networks — just as structuring techniques are important in the development of programs"*
>
> (Mylopoulos & Levesque, 1979)

- Digital Library Collections are an abstraction mechanism!

Outline

- What is a collection manager
- Requirements on collections
- Implementing the collection management
- Searching a distributed DL
- Implementing the search mechanism

Requirements: Content Providers

Content providers structure its object space in sets of items termed "collections“
→ Natural to mirror real-world collections and to hold the primary content of the DL.

BRICKS

•These containers are called Physical Collections

•A physical collection is a set of content items which belong to the same content provider and are homogeneous *from the provider point of view:*

- items are of the same kind
- items are described by the same metadata format(s)
- Items have same digital rights
- …

•In BRICKS the contribution of a content provider to the DL is always defined in terms of one or more physical collections, thus when content items are added to (removed from) physical collections, they are added to (removed from) the digital library.

Requirements: Content Providers

- Conversely: **a content item exists in the digital library only if there is a physical collection holding it**. Physical collections partition the DL information space: they are pair wise disjoint and their union makes up the content of the DL.

- **Physical collections are structured in (physical) sub-collections**, a notion that has been found a useful organizational mechanism by content providers.

- **A physical collection can have an arbitrary number of sub-collections** but only one parent collection. The graph of the sub-collection relation is a forest, each tree of which has a physical collection at the root. Within this tree, a content item of the physical collection belongs to exactly one sub-collection.

- **Physical collections are a central notion in the BRICKS content model**: not only content is organized by physical collection, but so is the discovery of resources and the definition of logical collections.

Requirements: Content Consumers

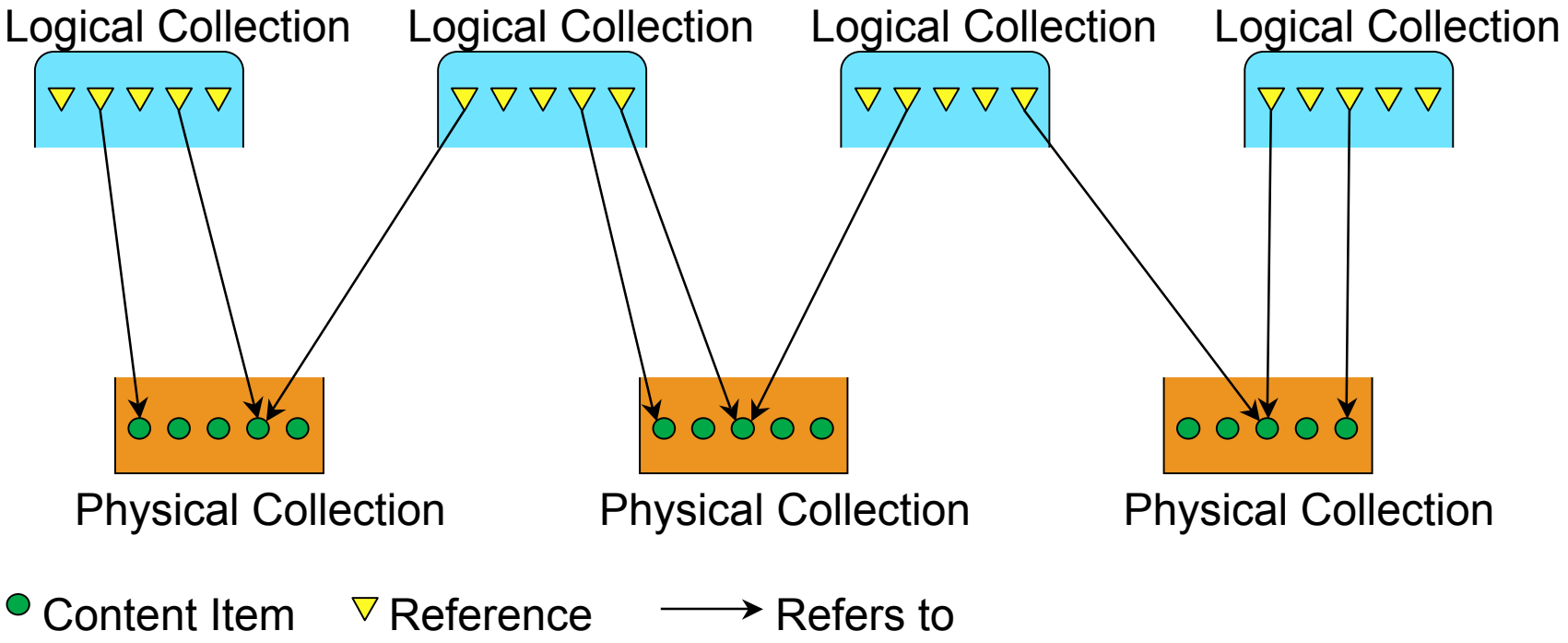
- People go to libraries to acquire knowledge for carrying out their own tasks.
- Typically, they search the whole library and end up with a subset of the library contents, consisting of the items that are relevant to them.
- This subset may be regarded as the *consumer view of the library*.
- The view is never static:
  - Consumers may evolve it manually, by adding newly discovered items or removing no longer useful ones
  - The view may evolve automatically: consumers describe their needs in some language and the items that satisfy these needs are added to the view:
    - *pull mode*: the consumer initiates the process (BRICKS)
    - *push mode*: someone else initiates the process (*e.g.* publish-subscribe)

Requirements: Content Consumers

- This notion of *view* is captured in BRICKS by Logical Collections.
- A Logical Collection in BRICKS is a set of *references* to DL items, with identity.
- Any registered BRICKS user can create and operate upon Logical Collections.
- By creating logical collections, BRICKS users organize the information space and at the same time enrich the Digital Library content: once a logical collection is created, it becomes part of the digital library information space and can be *e.g.* searched by other users as any other digital library resource.
- Operations on Logical Collections affect *references* not content items!

Requirements: Content Consumers

- A BRICKS content item may be referenced in many Logical Collections.
- Conversely, a Logical Collection may contain references to many items, coming from many different Physical Collections.



Static Collections

- BRICKS logical collections come in two flavors:
  - Static Collections
  - Dynamic Collections (a.k.a. Stored Queries)

- A static collection can be evolved manually by its owner, by inserting or removing references.
  - If the collection has been created with the intent of holding, for instance, all works of Maurolico on conics (intention), there is no guarantee that at any moment the collection indeed contains (extension) references to all Maurolico works on conics available in the digital library.
  - A static collection is a static container which communicates with the rest of the digital library only via the intervention of the users authorized to modify its extension.

Dynamic Collections

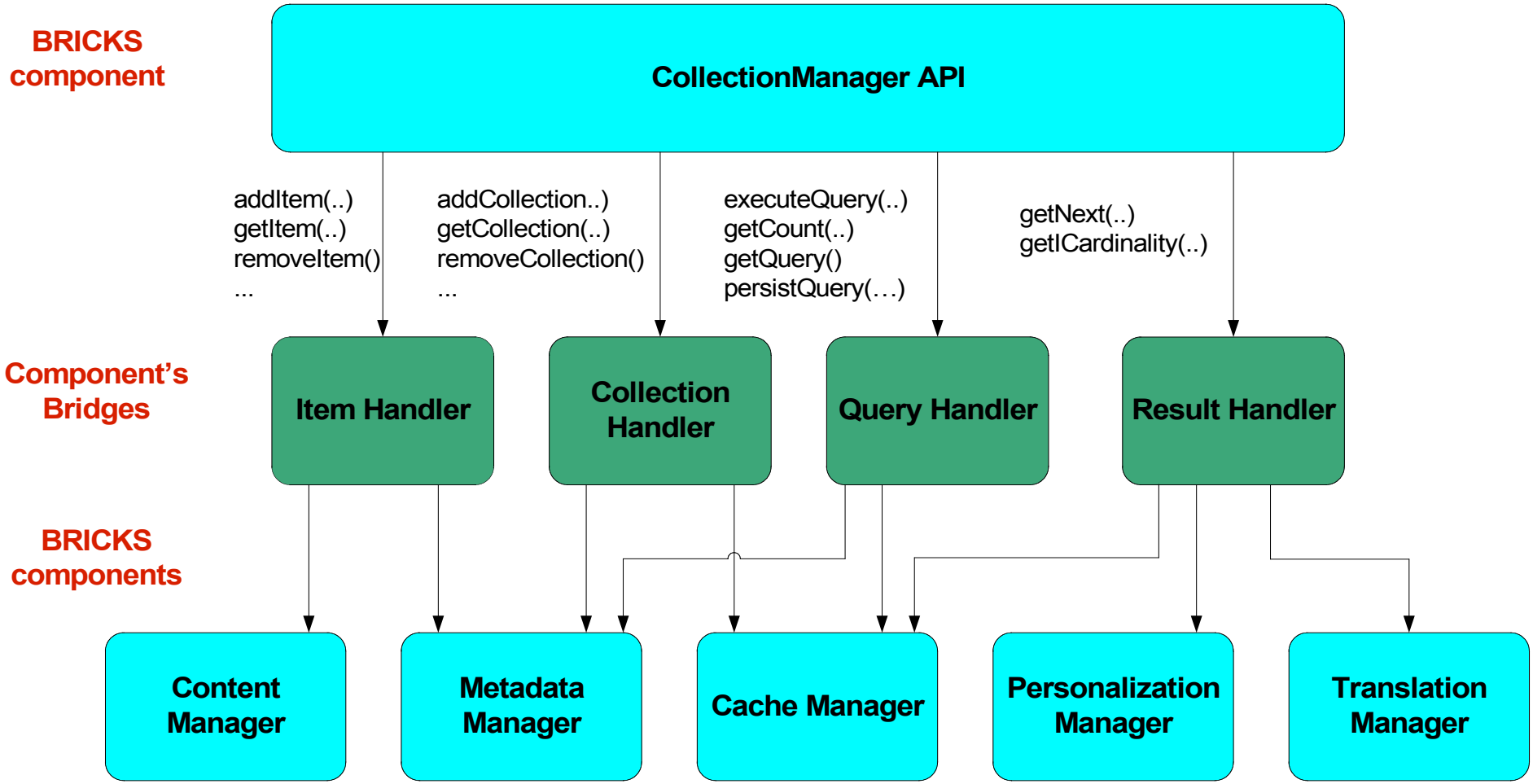
- A dynamic collection is defined by a query over the DL content, including other collections.
- Any time the dynamic collection is accessed, either in browsing or via a query, the defining query is evaluated.
- Dynamic collections evolve automatically as the DL content evolves.
- Dynamic collections evolve **only** automatically: no operation is offered to add or remove references from a dynamic collection.

- Static collections are extension-driven:
  - for consumers who cannot describe their needs other than extensionally, i.e. by pointing at the relevant items
- Dynamic collections are intension-driven:
  - for consumers who can describe their needs as a BRICKS query
    - BRICKS supports only the pull-mode (for now)

Outline

- What is a collection manager
- Requirements on collections
- Implementing the collection management
- Searching a distributed DL
- Implementing the search mechanism

Architecture

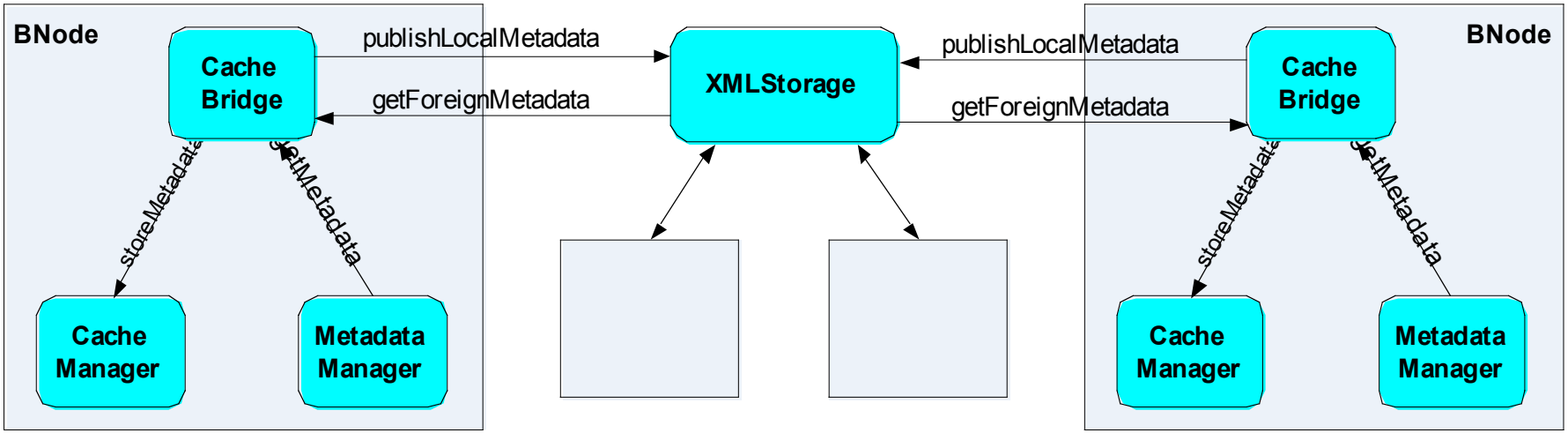


Cache Bridge: Start-up

- In every BNode, the metadata of the locally defined collections are stored in the local Metadata Manager
- At BNode start-up, the Cache Bridge:

1. Publishes the local collection metadata from the local Metadata Manager
   - into the local Cache and
   - into the decentralized XML Storage (to be retrieved by foreign BNodes)

2. gets foreign collection metadata from the XML Storage into the local Cache

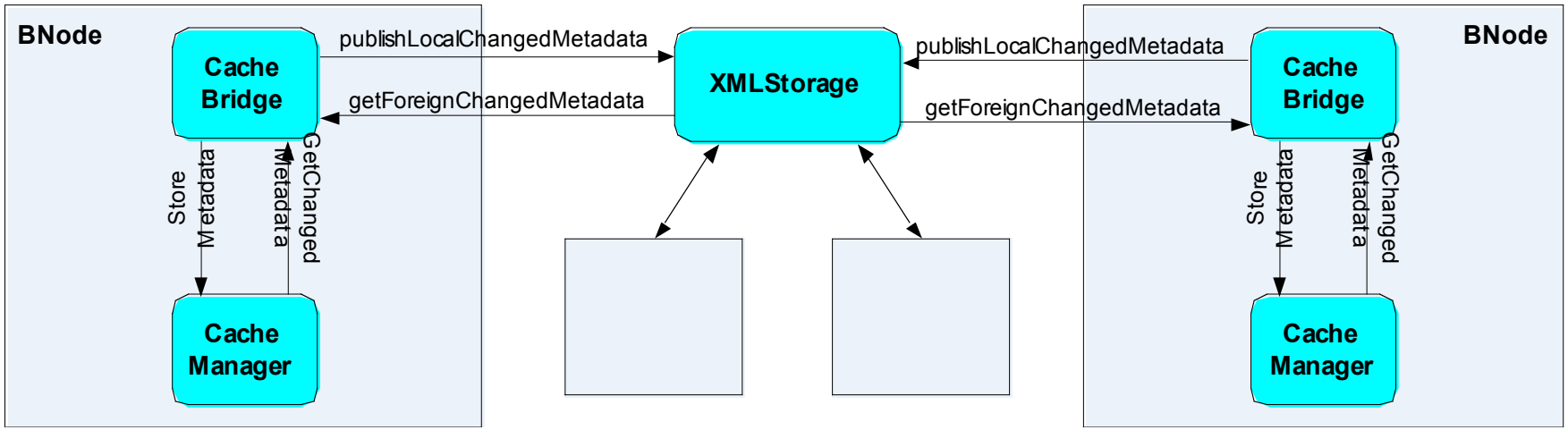


Cache Bridge : Runtime

At runtime, whenever the local collections metadata are modified (by some operation)

1. the Cache Bridge gets the the modified local metadata from the Cache Manager and publishes them to the XML Storage for the foreign BNodes to retrieve them upon synchronization

2. Periodically synchronizes the foreign collections metadata with the XML Storage



Outline

- What is a collection manager
- Requirements on collections
- Implementing the collection management
- Searching a distributed DL
- Implementing the search mechanism

Searching a distributed DL

- The users of a DL form a very heterogeneous class, ranging from casual, computer-illiterate people to highly specialized scholars.
- Al these users expect the DL system help them discover the DL objects of interest, independently from:
  - Location
  - Language
  - Type
  - Format
  - Structure
- in a way that reflects their preferences, if any, amongst which language plays a fundamental role.

Requirements

- An important distinction: application-dependent *vs* application-independent discovery
- Application-dependent: based on criteria highly specialized with respect to a class of applications
  - x-ray images similar to a given one
  - brilliant executions of a piece of music
  - video sequences giving a feeling of anguish
  - spectacular takeovers in Formula 1 races
- Application independent: objective criteria
  - movies directed by Woody Allen
  - recent books on bio-informatics
  - Chopin's preludes played by Pollini

The query facility of a DL must be:

- extensible: it must be possible to plug-in it specialized search engines, devoted to capture the semantics of application-dependent search criteria
- flexible: it must be possible to express different kinds of queries, each addressing a different level of skill or knowledge of the DL contents
- user-sensitive: it must adapt to the preferences of the user
- efficient: it must respond as fast as possible

Outline

- What is a collection manager
- Requirements on collections
- Implementing the collection management
- Searching a distributed DL
- Implementing the search mechanism

Types of queries:

- *Collection queries*: allow to discover collections by stating a Boolean condition on the collection metadata

- *DL objects queries*:
  - ž allow to discover content objects
  - ž may be contextualized to certain BNodes or collections
  - ž may be personalized
  - ž come in 3 flavors:
    - Simple
    - Advanced
    - Ontology

Simple queries

- A simple query is the simplest form of query and most popular
- It is meant to serve casual users or users having a rather vague idea of the desired resources.
- These users typically do not want to address their search to specific metadata attributes, or operators.
- The language for simple queries allows to express sequence of unconstrained terms, very much in the style of e.g. Web search engines. In their search, users will be able to use:
  - wild cards: ca?lo databas*
  - phrases: “jakarta apache”
  - proximity operators: "jakarta apache"~10
  - Boolean operators: "jakarta apache" AND "jakarta lucene“
- In a simple search, metadata records are seen as texts whose words are the metadata attribute values.

Advanced queries

- For users who can characterize their information needs very precisely in terms of a metadata schema, such as librarians or expert library users.

- An advanced query is a
  - Boolean combination of simple conditions
  - on metadata fields, possibly coming from different schemas:

DC.creator = “Bob” AND XY.date > 01.01.2000

- In an advanced search, metadata records are seen as sets of (attribute value) pairs, exactly like database records, which may or may not satisfy a query stating simple conditions on such attributes.

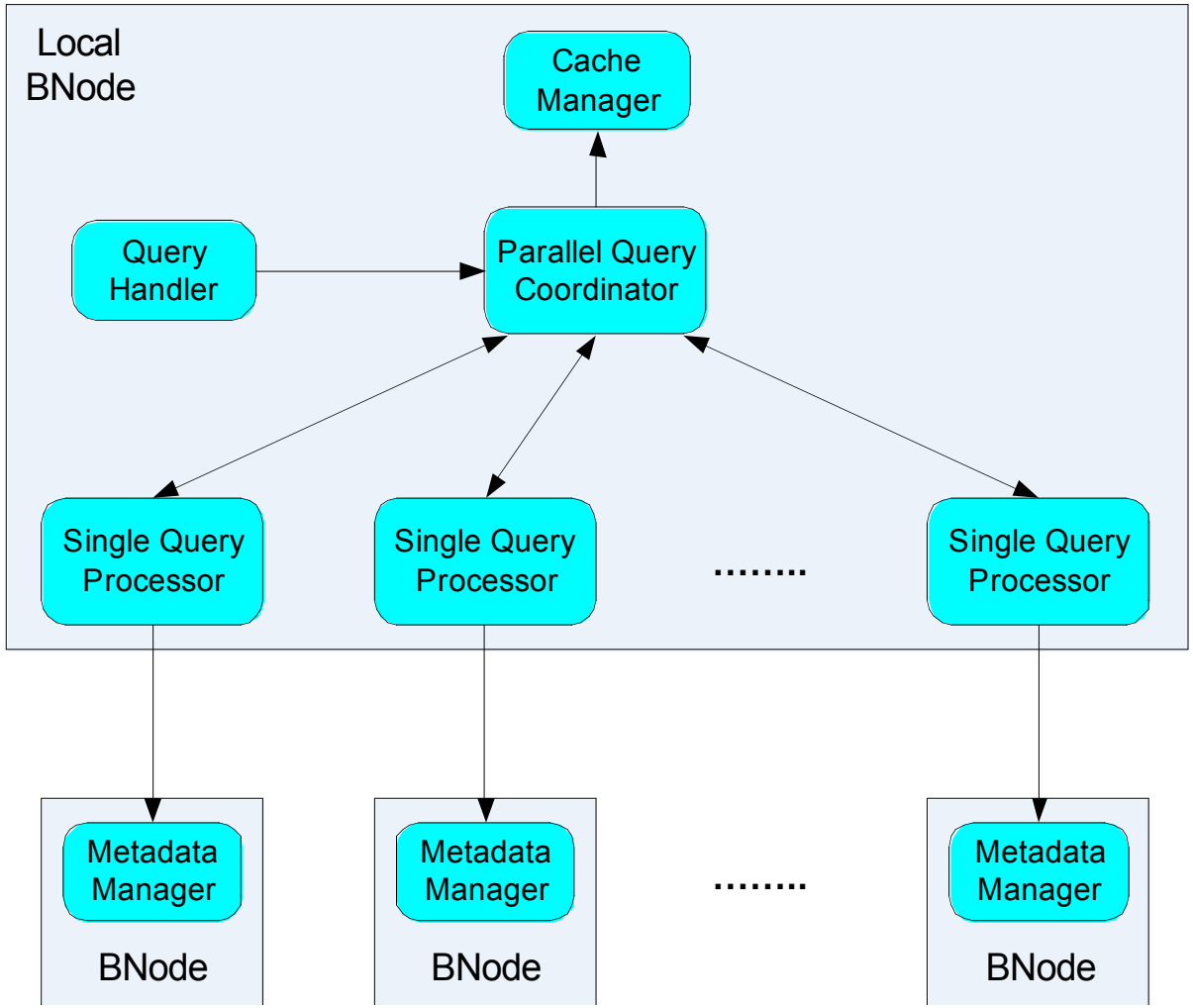
Ontology queries

- For highly specialized users, who are familiar with an ontology formally conceptualizing their domain of interest, and want to retrieve documents annotated by descriptions derived from that ontology.

- An ontology query is a SPARQL expression:

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?title
WHERE { <http://example.org/book/book1> dc:title ?title }
```

Query Handler: Overall Architecture



Archaeological Sites Application Prototype

Finds[1] identification today

- The public is an important source for understanding the past and archeological sites
- Thousands of objects are discovered by people during walking, traveling, gardening, etc.
- Museum curators, archaeologists, students, amateurs all need to identify these finds
- Traditional way of working
  - Examining objects
  - Preliminary identification
  - Comparison to specialist reference collections, e.g. roman coins
    - Sometime object descriptions are enough
    - Sometime it is the starting point for a scientific analysis process
  - Comparison to objects in the museum collection
  - Record object in detail

[1] With *Finds* we mean *archeological finds*

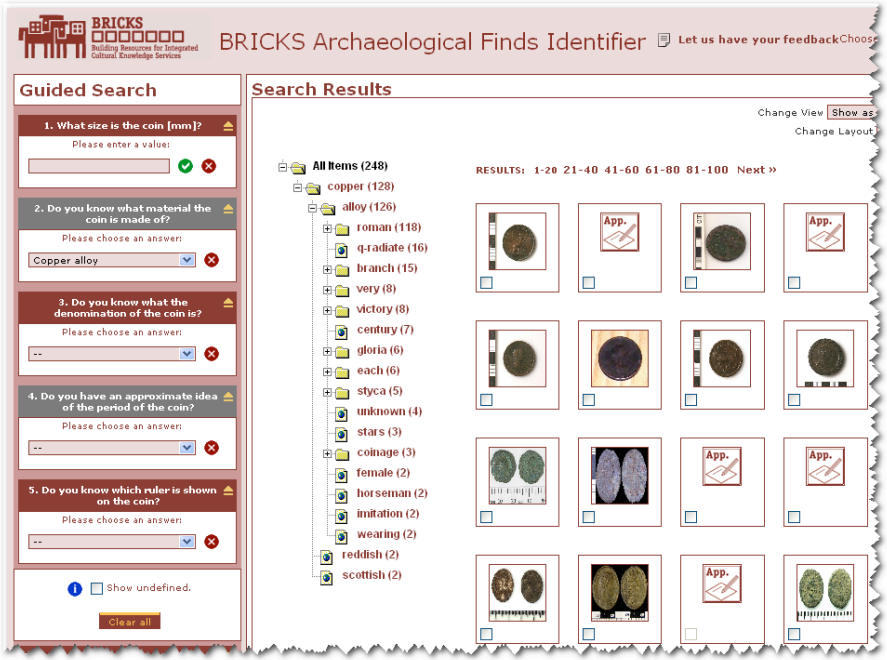
Finds identification today

- Disadvantage of the traditional way
  - The reference collections have to be known by the curator
  - Curator needs information about new collections, e.g. when a museum opens its archive
  - Curators need to access them one by one
  - Collections are heterogeneous from point of view of data, search facilities, user interface, languages, etc.

- As a consequence curators invest a lot of time in the identification process

Finds identification with BRICKS

- Examining objects
- Preliminary identification
- Comparison to specialist reference collections
  - Easy access to all relevant archeological collections (Even to unknown collection)
  - Advanced search facilities
    Support of ontologies, translation services, different types of search
  - One application to access all collections
  - Collaborative identification processes
- Comparison to objects in the museum collection
- Record object in detail



Archaeological Sites Application Prototype

DEMO

Lessons Learned

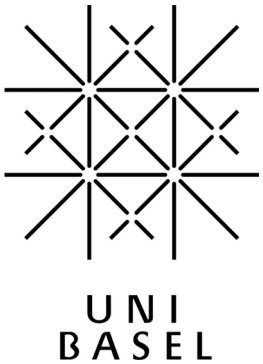
- Developers and system designers have a longer learning curve about service oriented technologies
- A service oriented design is different from traditional design, e.g. the number of function should be limited and well considered
- Communication costs between services are often underestimated
- The communication of the concept of distributed architectures to the end user is a hard process
- An early prototype is helpful for the communication between users and developers

Questions & Discussions

Bringing Digital Libraries to Distributed Infrastructures

DelosDLMS - the DELOS Digital Library Management System

Tutorial at JCDL 2007
June 19, 2007



Heiko Schuldt (University of Basel)

Agenda

09:00 – 09:20	**Introduction: Motivation & Challenges**
09:20 – 09:45	**Challenges of bringing DL to distributed Infrastructures**
09:45 – 10:30	**Underlying Technologies and their promises** (SOA, P2P, Grid)
10:30 – 10:45	*Coffee break*
10:45 – 12:00	**Solutions for decentralized DL infrastructures** (with BRICKS Demos)
12:00 – 12:30	**DelosDLMS - the DELOS Digital Library Management System**
12:30 – 13:30	Lunch
13:30 – 14:00	**DelosDLMS Demos**
14:00 – 15:00	**Building DL services on the Grid** (DILIGENT)
15:00 – 15:30	*Coffee break*
15:30 – 16:45	**DILIGENT Demos**
16.45 – 17:00	**Conclusions and future directions**

The DELOS Project

www.delos.info

DELOS Network of Excellence

An initiative funded until December 2007 by the European Commission, under the 6th Framework Program.

Presently it has 61 members:

European Research Consortium for Informatics and Mathematics ►Institute 0f Information Science and Technologies – CNR ► Swiss Federal Institute of Technology Zurich ►University of Bath ►University of Athens ►Technical University of Crete ► University of Florence ►Fraunhofer Institute for Integrated Publication and Information Systems – IPSI ►University of Glasgow ► University of Duisburg-Essen ►National Research Institute for Mathematics and Computer science in the Netherlands – CWI ► Foundation for Research and Technology – Hellas ►University of Rome "La Sapienza" ►National Technical University of Athens ► University of Padua ►University of Milan ►Max-Planck Institute for Informatics – MPII ►Oldenburg Research and Development Institute for Information Technology Tools and Systems – OFFIS ►Queen Mary & Westfield College ►University of Strathclyde ► Ionian University ►University of Paris-Sud XI ►University of Southampton ►The University of Edinburgh ►Institute for Information Processing and Computer Supported new Media – IICM ►Vienna University of Technology ►University of Urbino ►Norwegian University of Science and Technology – NTNU ►Lund University ►French National Institute for Research in Computer Science and Control – INRIA ►National Archives of the Netherlands ►University of Bremen ►Austrian Academy of Sciences ►University of Cologne ►Forma Consortium ►Swedish Institute of Computer Science ►University of Modena and Reggio-Emilia ►Masaryk University of Brno ►University of Amsterdam ►University of Lugano – USI ►Computer and Automation Research Institute of the Hungarian Academy of Sciences – SZTAKI ►University of Bari ►Health Information Technologies Tyrol ►University of Lancaster ►Athens University of Economics and Business ►University of Glamorgan ►University of Queensland ►Austrian National Library ►Goettingen State and University Library ►Imperial College ►Institute of Knowledge Sharing ►Center of Cognitive Systems Engineering ►Virtual Resource Centre for Knowledge about Europe – CVCE ►University of Science and Technology of Lille ► University of Konstanz ►University of Basel

DELOS main objective

To define and conduct a joint program of activities in order to integrate and coordinate the on-going research activities of the major European research teams in the field of digital libraries for the purpose of developing the next generation digital library technologies

DELOS - Grand 10-Year Vision

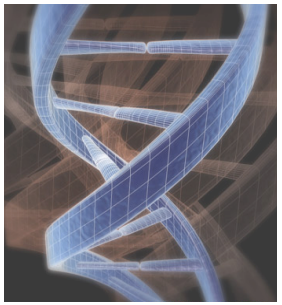
Digital libraries should enable any citizen to access all human knowledge anytime and anywhere,in a friendly, multi-modal, efficient, and effective way, by overcoming barriers of distance, language, and culture and by using multiple Internet-connected devices

The potential exists for digital libraries to become the universal knowledge repositories and communication conduits for the future, a common vehicle by which everyone will access, discuss, evaluate, and enhance information of all forms

General activities

- Journal, conference proceedings, workshop publications, books
- DELOS Thematic Workshops
- Definition of a Reference Model for Digital Libraries
- Integrated Prototype Development
- Task workshops/meetings, Joint workshops
- Summer Schools
- Researcher Exchanges
- Short Visits

DELOS Research Directions



Foundations

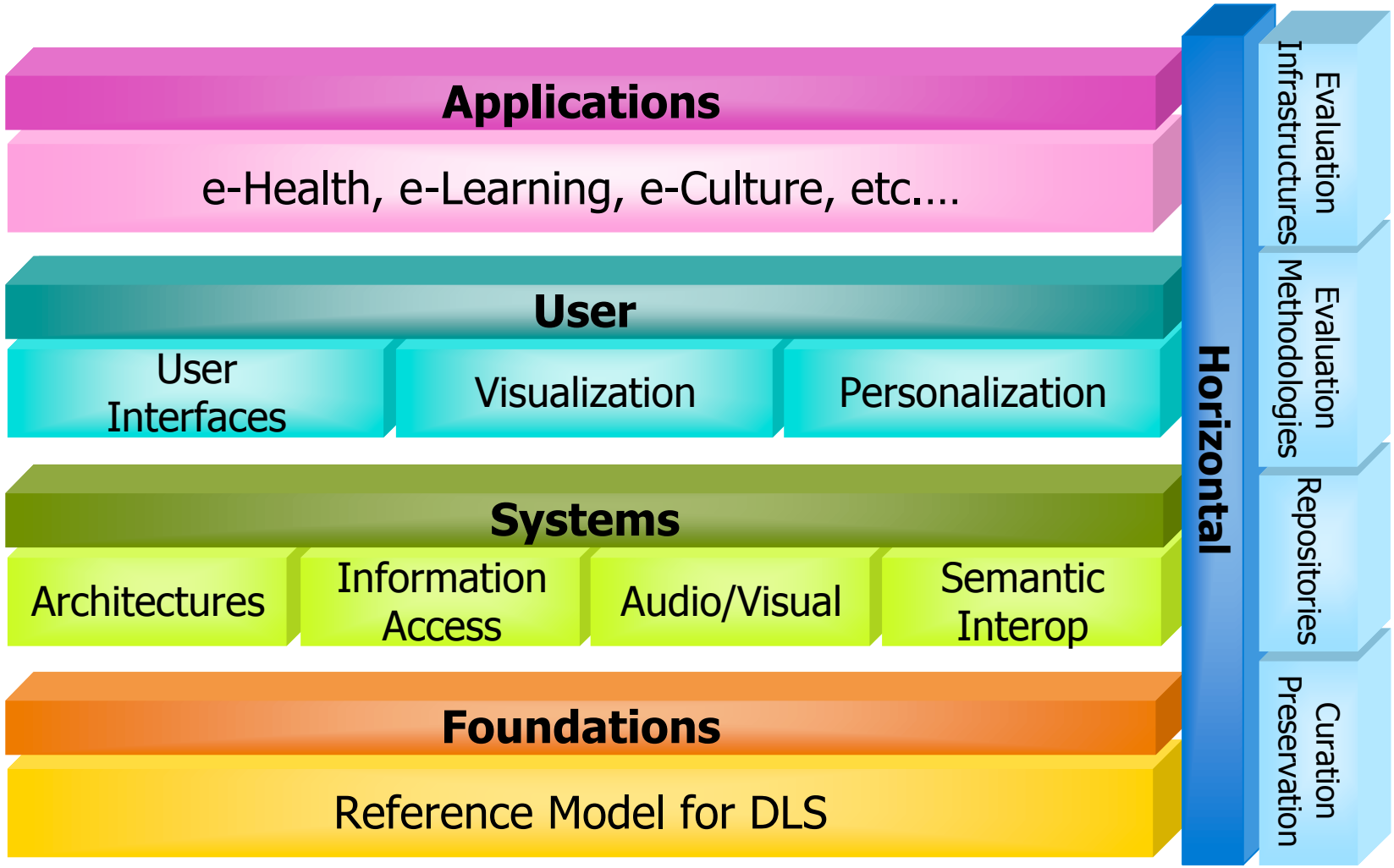
System

User

Horizontal

Applications

DELOS Research Directions

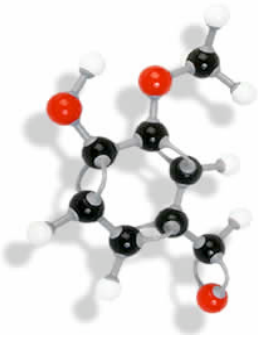


Foundations Research Issues

Reference Model for DLS

Formalize a conceptual framework for Digital Library systems

- to serve as a yardstick of quality and richness
- to specify features and properties of generic DLMS
- to clarify relationships among
  - digital libraries, digital repositories, digital archives,
  - search engines, information infrastructures,
  - knowledge commons



System-related Research Issues

Architectures

- Peer-to-peer architectures
- Grid middleware
- Service-oriented architectures

Information Access

- Indexing for complex and novel data
- Query routing in complex distributed Digital Libraries

System-related Research Issues

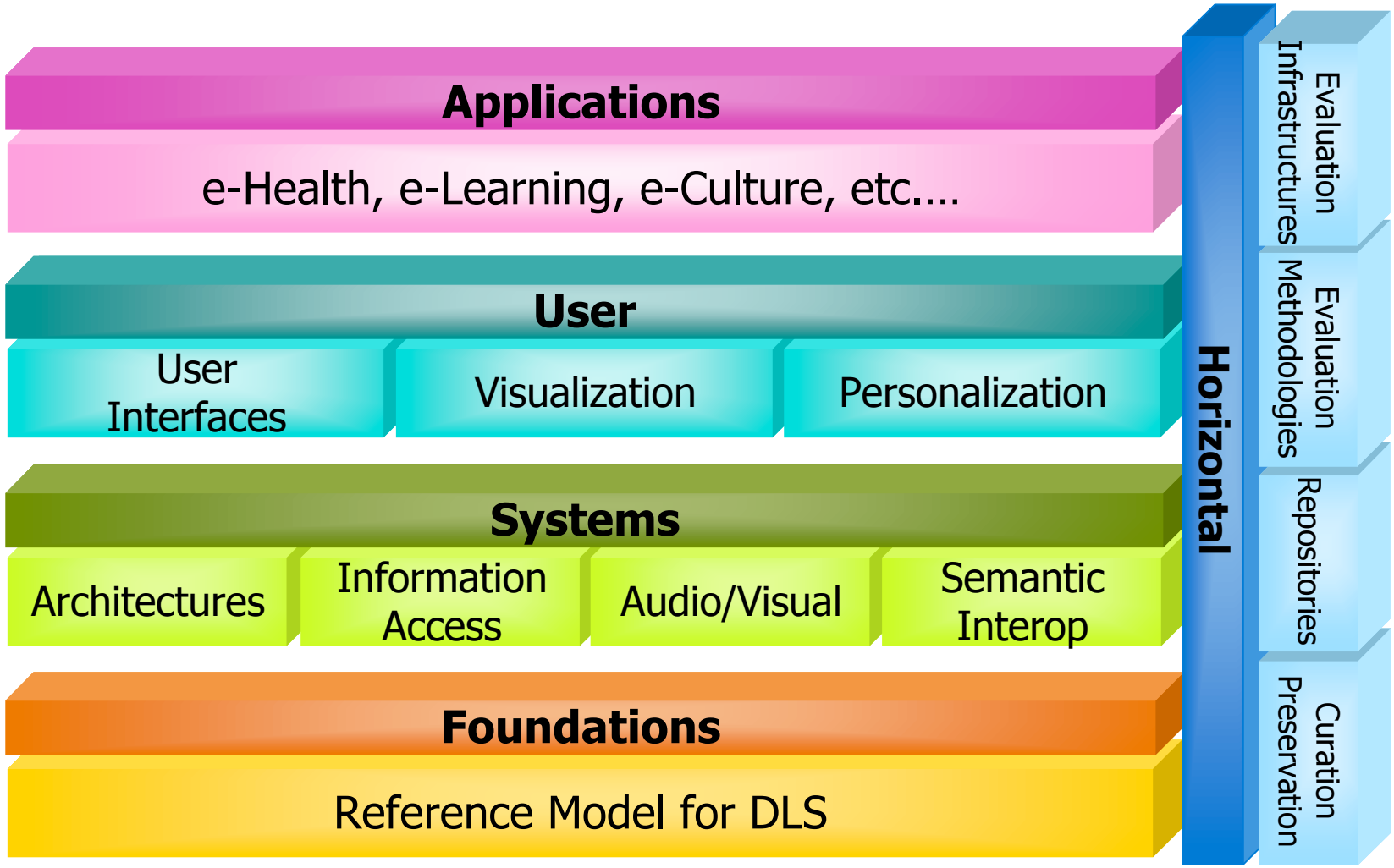
Audio/Visual

- Automatic metadata extraction
- Context-aware content-based retrieval
- Audio/visual interfaces

Semantic Interop.

- Methods for the integration of heterogeneous ontologies and domain-specific knowledge organization systems
- Interoperability with e-Learning applications

DELOS Research Directions



User-related Research Issues

User Interfaces

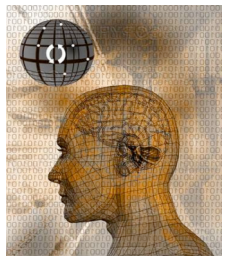
- Framework for new digital library interfaces
- Task-oriented user interfaces
- Cooperation/collaboration tools, e.g., annotations

Visualization

- Self-adaptability to small screens
- Visual analysis and exploration of query results

Personalization

- Modelling foundations for user preferences and context
- Personalization of user interactions
- Peer-similarity-based query routing decisions
- User log analysis for profiling



Horizontal Research Issues

Curation/Preservation

- Integration of preservation functionality
- Establishment of a testbed and evaluation framework for preservation techniques
- Automating selection and ingest processes

Repositories

- Common grounds: access policy, operational environment
- Infrastructure repositories for research and learning

Horizontal Research Issues

Evaluation Methodologies

Standard frameworks for comparative evaluation of DL Systems

- Definition of standard events in a DL environment
- Identification of appropriate metrics
- Establishment of information repositories

Evaluation Infrastructures

- Cross Language Evaluation Forum (CLEF)
- Initiative for Evaluation of XML retrieval (INEX)

Applications Research Issues

e-Health

- Virtual electronic health records
- Integration of multiple medical information streams

e-Learning

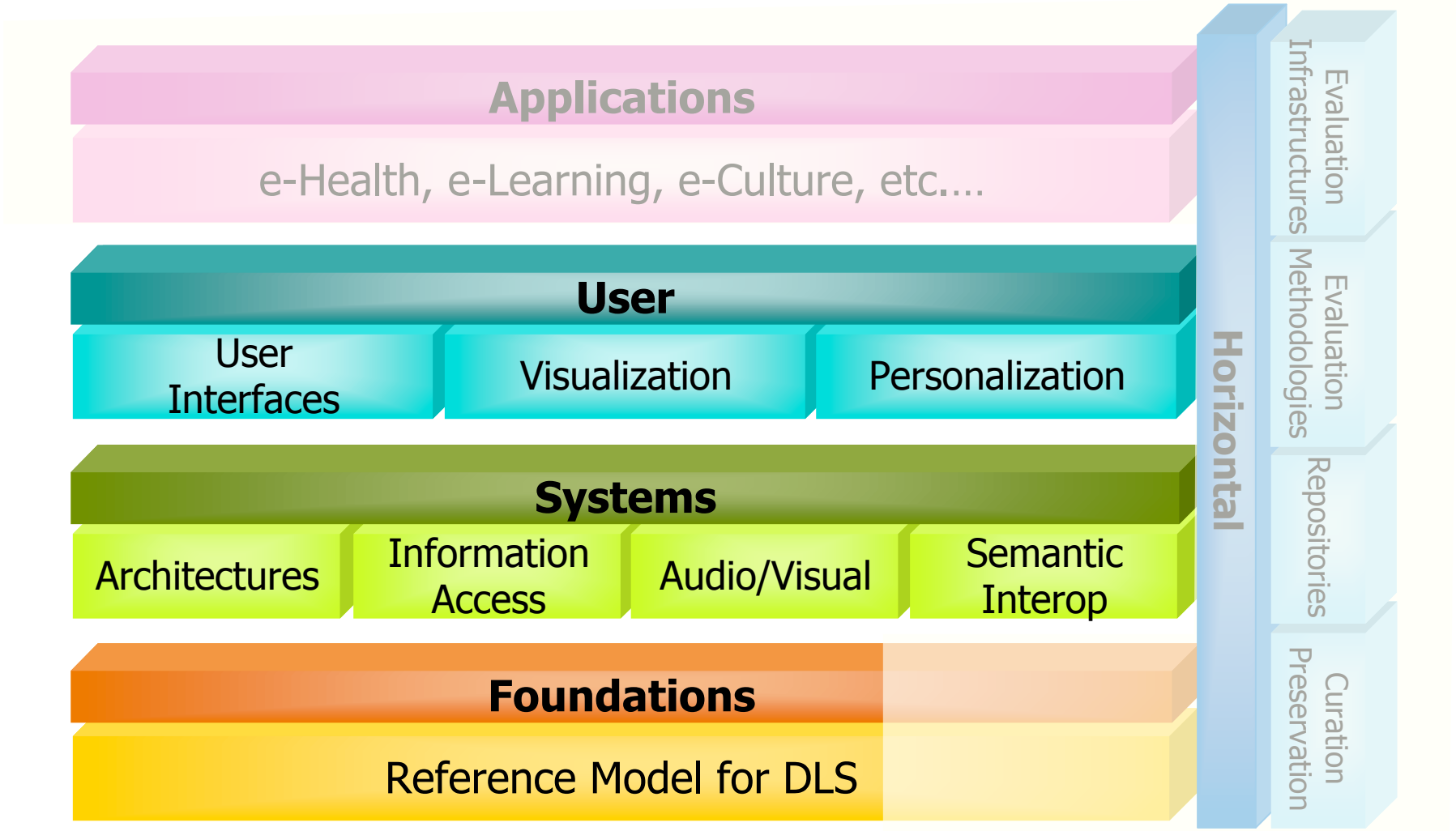
- Interoperability of e-Learning applications

e-Culture

- Integration of upper-level ontologies
- Mapping of core ontologies to schemas and KOS

...

Delos DLMS


Applications
e-Health, e-Learning, e-Culture, etc....
User
User Interfaces
Visualization
Personalization
Systems
Architectures
Information Access
Audio/Visual
Semantic Interop
Foundations
Reference Model for DLS
Horizontal
Evaluation Infrastructures
Evaluation Methodologies
Repositories
Curation Preservation

The DelosDLMS Challenges

- Have a demonstrator for future Digital Library Management Systems that not only shows new combined text and audio-visual search functionality and personalized browsing by new adaptable information visualization and relevance feedback tools at the interface
- but also proves that generic systems can be built that not only support finding of relevant information but also enable to annotate and process found information,
- to integrate sensor data stream processing,
- and – from a systems engineering point of view - allows simple configuration and adaptation while being reliable and scalable

DelosDLMS

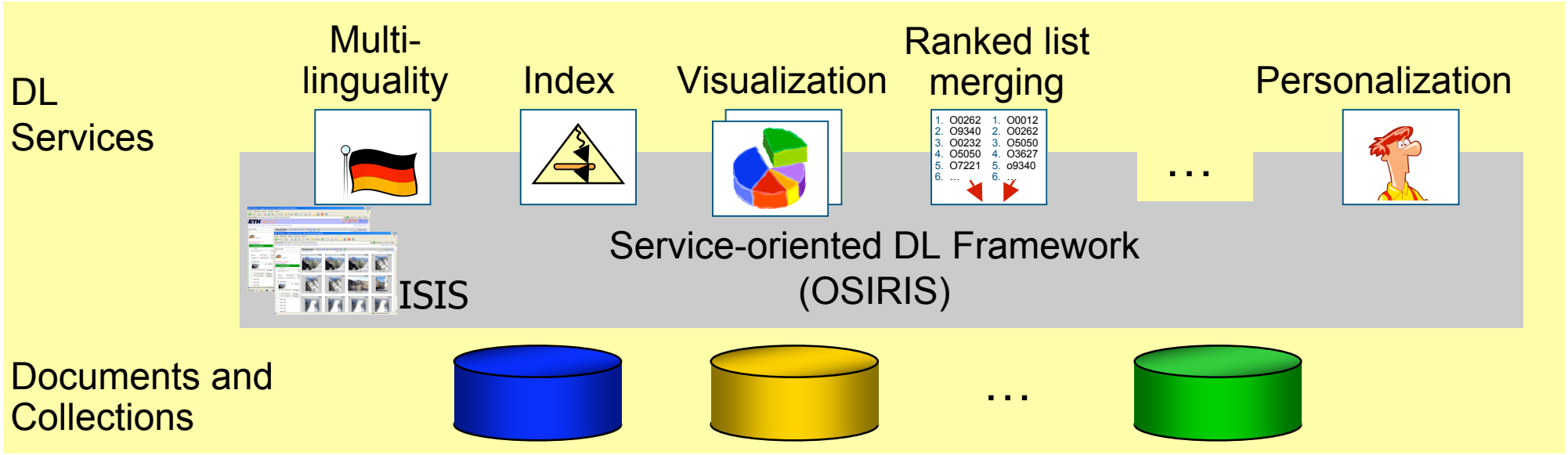
- DELOS has developed so far highly sophisticated DL functionality
- The goal of DelosDLMS is to integrate this functionality into an prototype for the future Digital Library
- Basis: ISIS/OSIRIS infrastructure originally developed at ETH Zürich, now being extended at the University of Basel
  - ISIS: Content-based multimedia similarity search
  - OSIRIS (Open Service Infrastructure for Reliable and Integrated process Support)
    - ž underlying distributed P2P process management infrastructure
- DelosDLMS is a partial implementation of the DL Reference Model
- Specialized DL functionality from (DELOS and non-DELOS) partners is made available and integrated by means of (web) services

Key DelosDLMS Partners

- U Basel: DelosDLMS Infrastructure, Content-based multimedia retrieval
- U Konstanz: Visualization, Intelligent Browsing
- ETH Zurich: Interactive Paper
- U Duisburg: Interface Functionality
- U Crete: NL, Speech, Video Services
- U Padova: Annotations
- U Florence: 3D Objects & Video Retrieval
- U Vienna: Music Retrieval
- U Roma: Metadata Visualization
- MPI Saarbrücken: P2P Search
- U Athens: Personalization

DelosDLMS – Overview

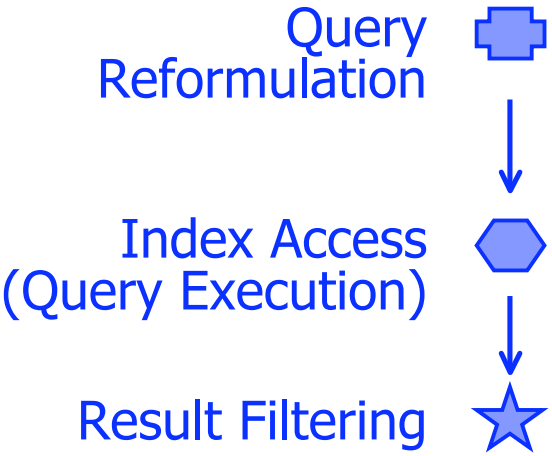
- DL Functionality is available by dedicated services
- DL Applications can be built by combining existing DL services (possibly from different providers), independent of content
- Leads to extensible systems that can easily be adapted

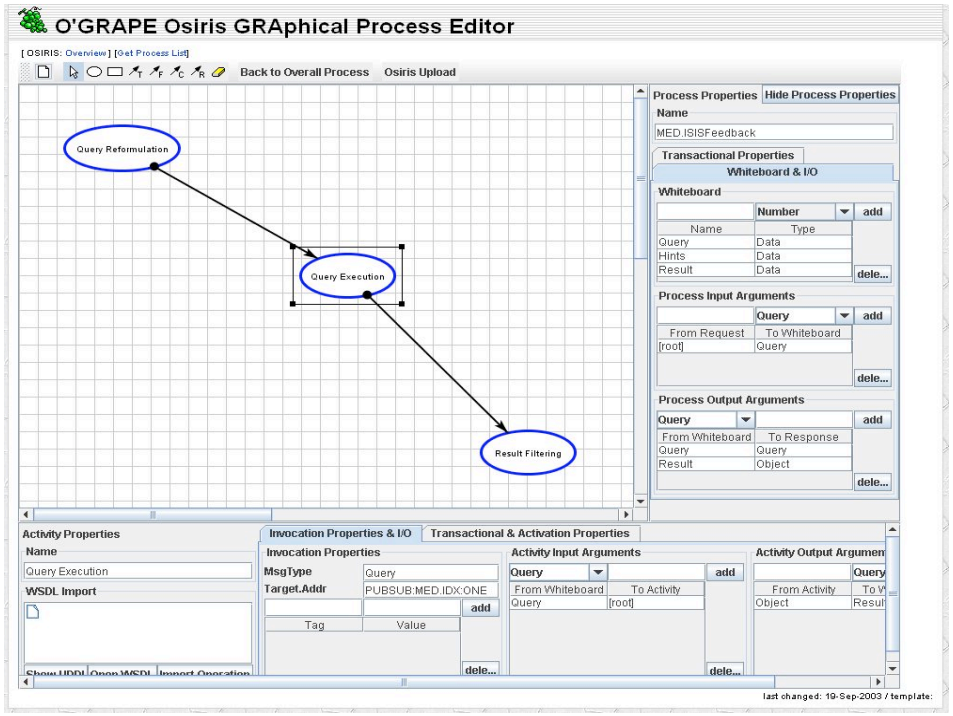


Defining DL Applications in DelosDLMS

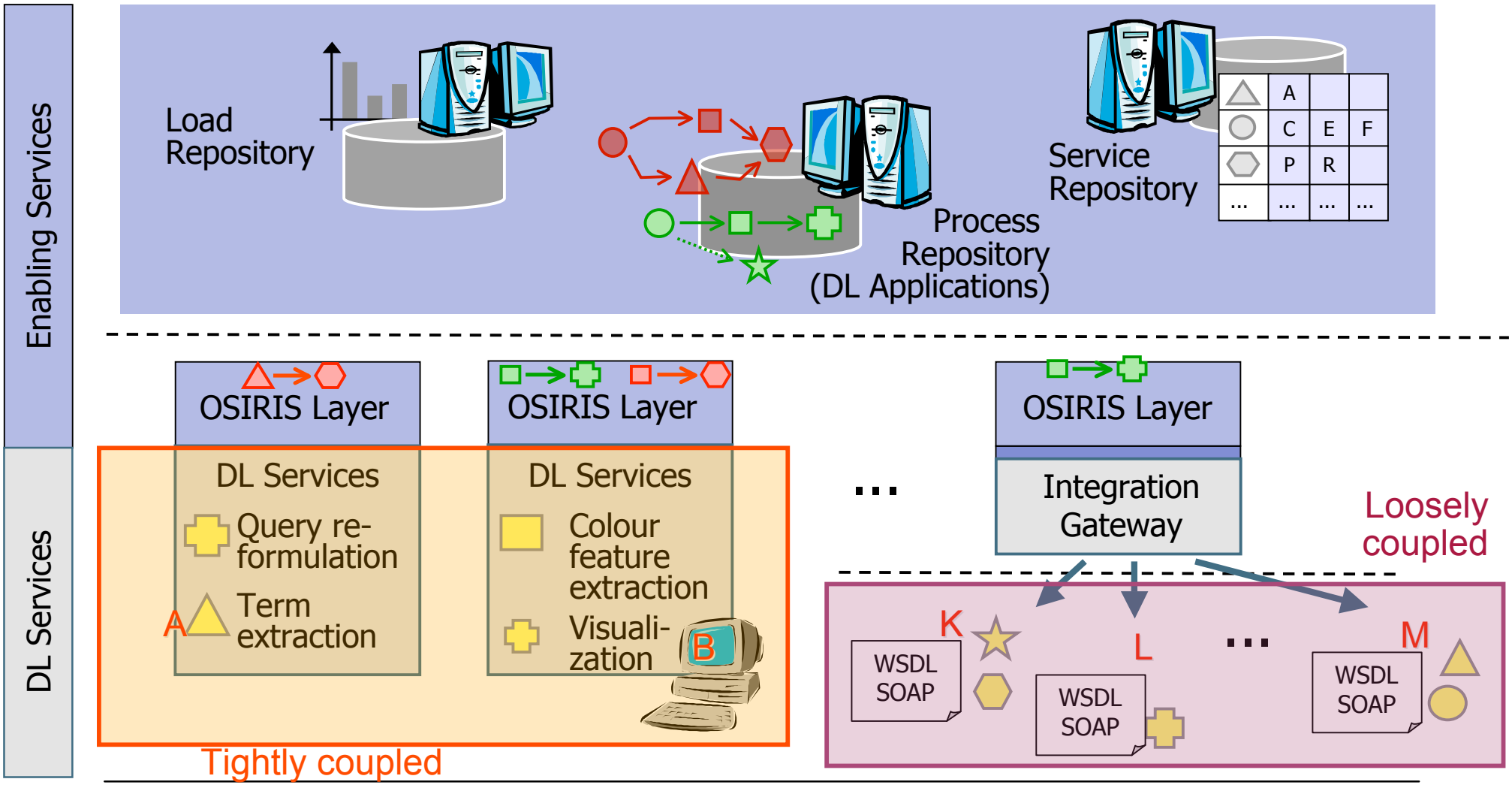
Development of OSIRIS DL Applications (= Processes)

- combination of existing services
  („Programming in the Large“)





OSIRIS Infrastructure: Integration of Services



OSIRIS: Classification of Services

Tightly coupled services:

- First-class citizens in the OSIRIS network
- Integrated into the local OSIRIS layer
- Quality-of-Service:
  - Provide added value for services (e.g., compensation, restart, etc)
  - Load balancing
  - Execution guarantees for applications consisting of such services

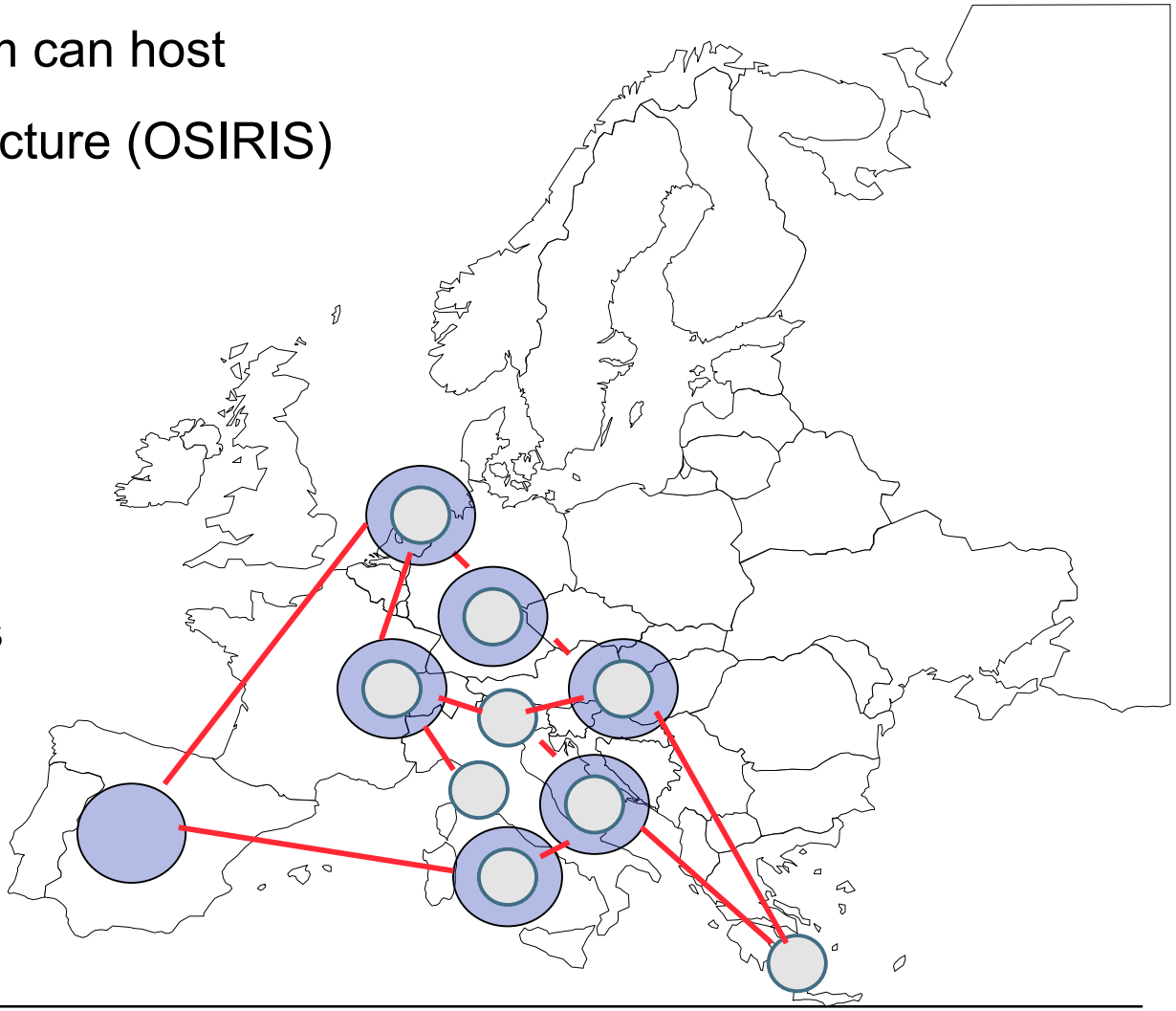
Loosely coupled services

- Easy to integrate (only web service interface is needed, i.e. description via WSDL, invocation via SOAP)
- Only “best effort”, no
  - sophisticated failure handling
  - Load balancing, etc.

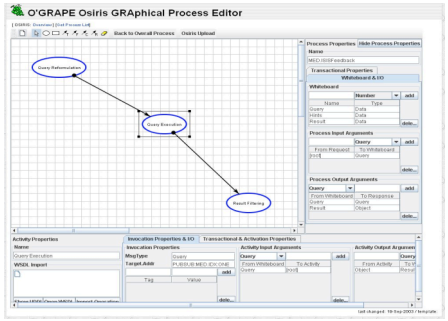
DelosDLMS Infrastructure: Deployment

Nodes in the system can host

- Both the infrastructure (OSIRIS) and DL Services
- Only the Infrastructure
- Only DL Services



Executing DL Applications in the OSIRIS Infrastructure



1. Define process-based DL application

Query Reformulation

Index Access (Query Execution)

Result Filtering

2. Upload application

Processes

3. Execute

OSIRIS

DL Services

OSIRIS

DL Services

OSIRIS

DL Services

OSIRIS

DL Services

Result Filtering

Index Access

Delos DLMS Features …

- Content-based Image and Video Retrieval (U Basel)
- Audio Search based on features (TU Vienna)
- 3D object retrieval (U Florence)
- Novel and advanced user interfaces
  - Daffodil (U Duisburg)
  - MedioVis (U Konstanz)
  - DARE (U Roma) integrated into MedioVis
  - Exploratory browsing by SOM (U Konstanz)
- iPaper as input device (ETH Zürich)
- Annotation management (U Padua)

… Delos DLMS Features

- Semantic Video Services (TU Crete)
  - Speech interface
  - Ontology
- Soccer videos including automatically extracted semantics from (U Florence)

Collections in DelosDLMS

ETH World: 625‘026 Images from the ETH website. Extracted features:

- Color features (Global & five regions)
- Texture features (Global & five regions)
- Keywords (from metadata)

Art Gallery: 16’266 art objects with image and metadata

ISIS: 53‘837 images

MED: 50‘143 medical images

Audio: 1‘185 MP3s (for Audio-Retrieval)

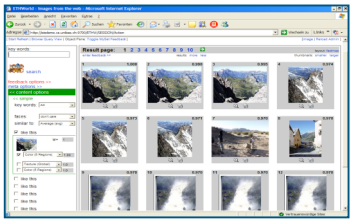
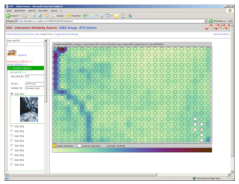
Video: 5 movies

- Keywords (from subtitles)
- Image features (from single shots)

IMDB: 45‘361 records with a small cover or poster and structured metadata

Where DelosDLMS started from

Information Society Technologies

ISIS / OSIRIS (ETH / U. Basel)	3D Retrieval (U. Firenze)	Audio Retrieval (TU Vienna)	SOM (U. Konstanz)	Paper++ (ETHZ)	Daffodil (U Duisburg)
					
Vis, Index, RF — ISIS; OSIRIS	3D Retrieval	Audio Retrieval	Visualization	Active Paper	Text Search, Author Search, ...
Images, Video, Audio	3D	Audio			

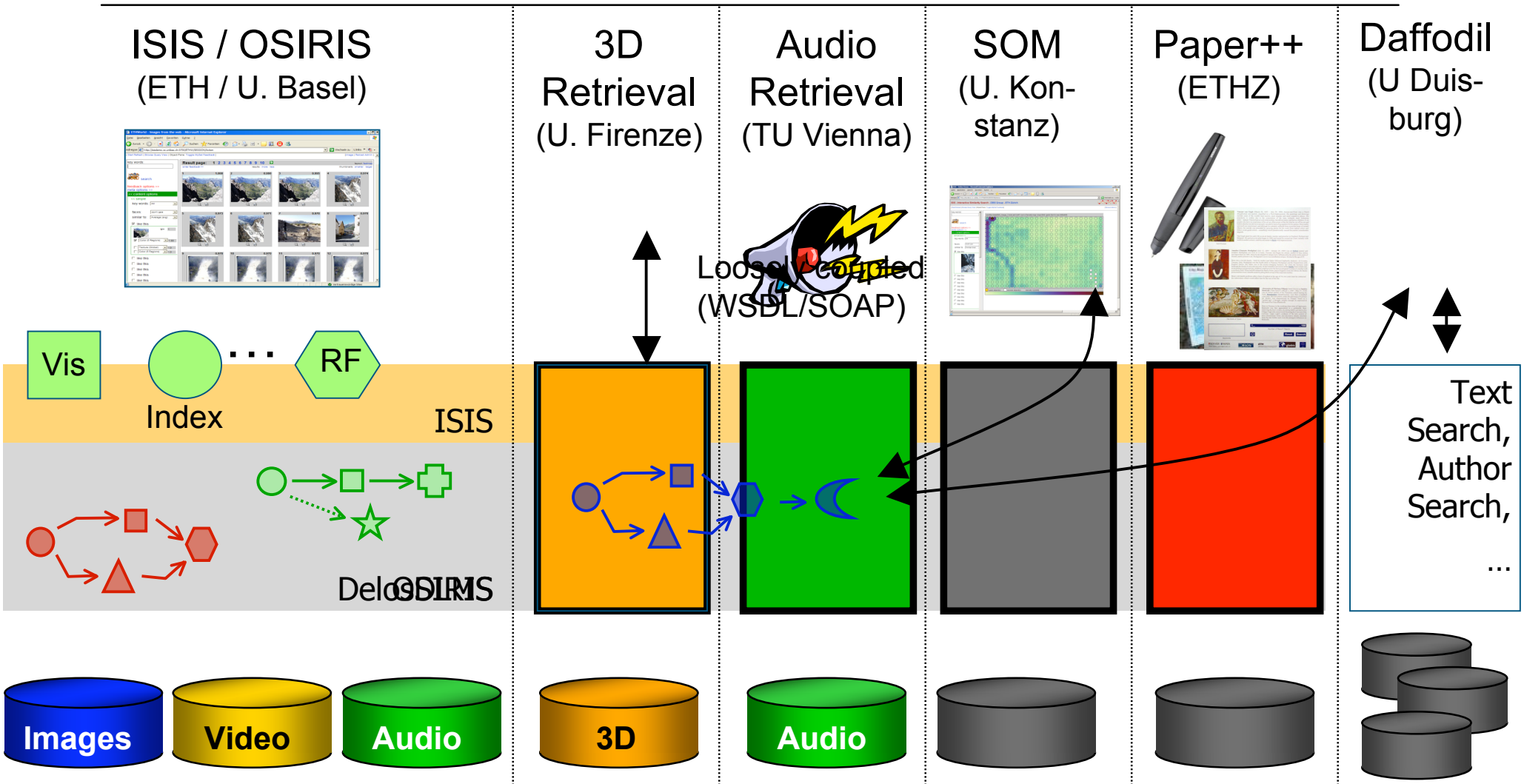
BRICKS Building Resources for Integrated Cultural Knowledge Services

DELOS Network of Excellence on Digital Libraries

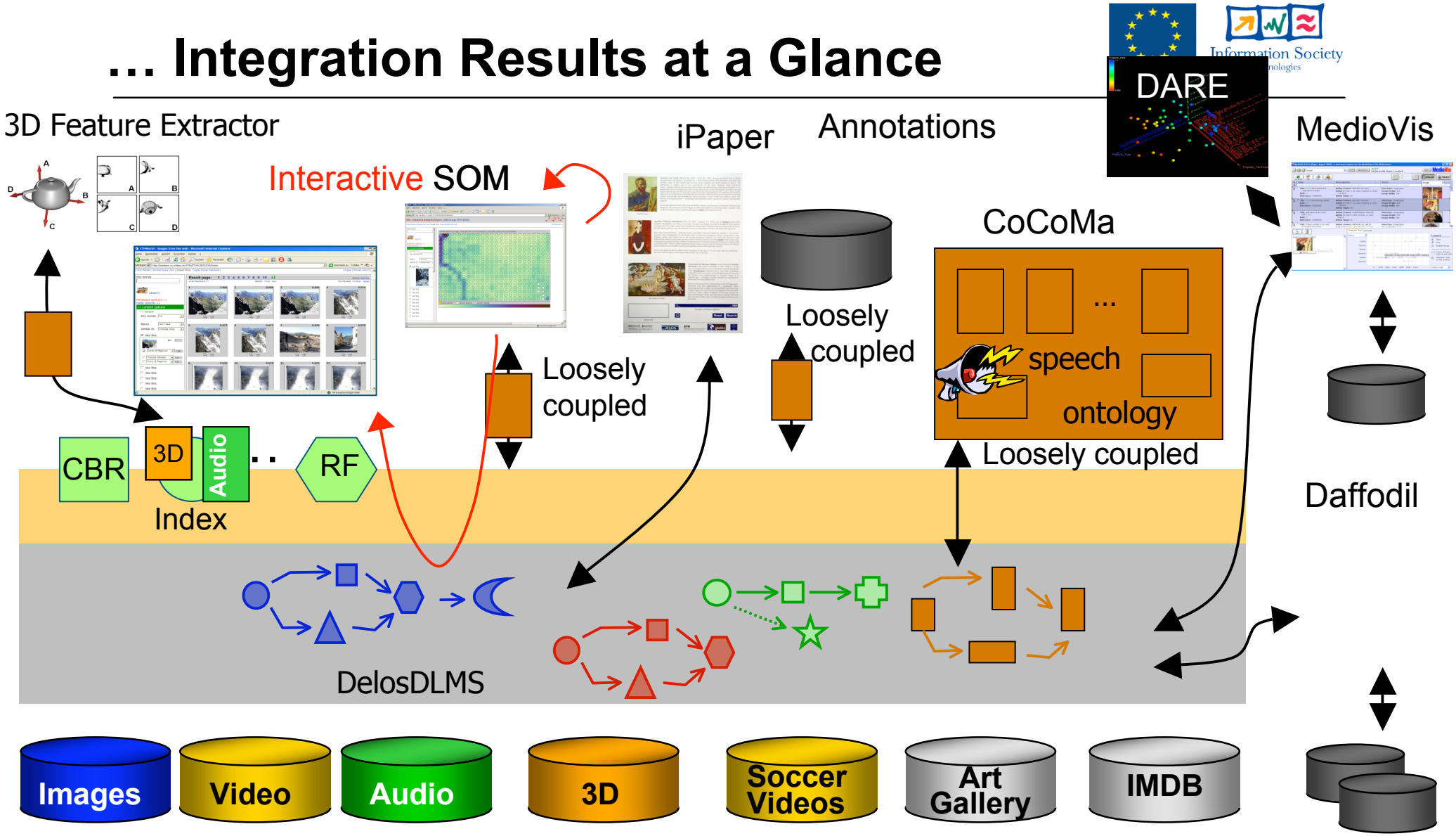
Diligent A DIgital Library Infrastructure on Grid ENabled Technology

... and many more

Integration Results at a Glance …

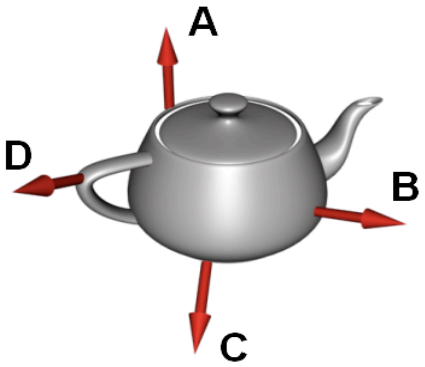


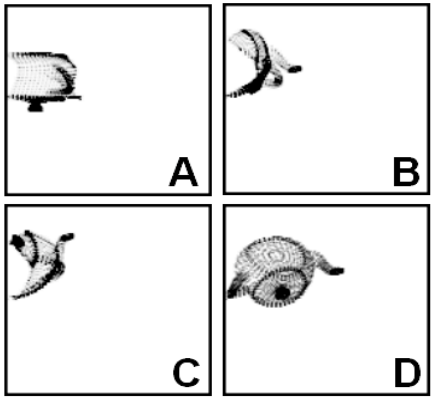
… Integration Results at a Glance



3D Retrieval (U Florence)

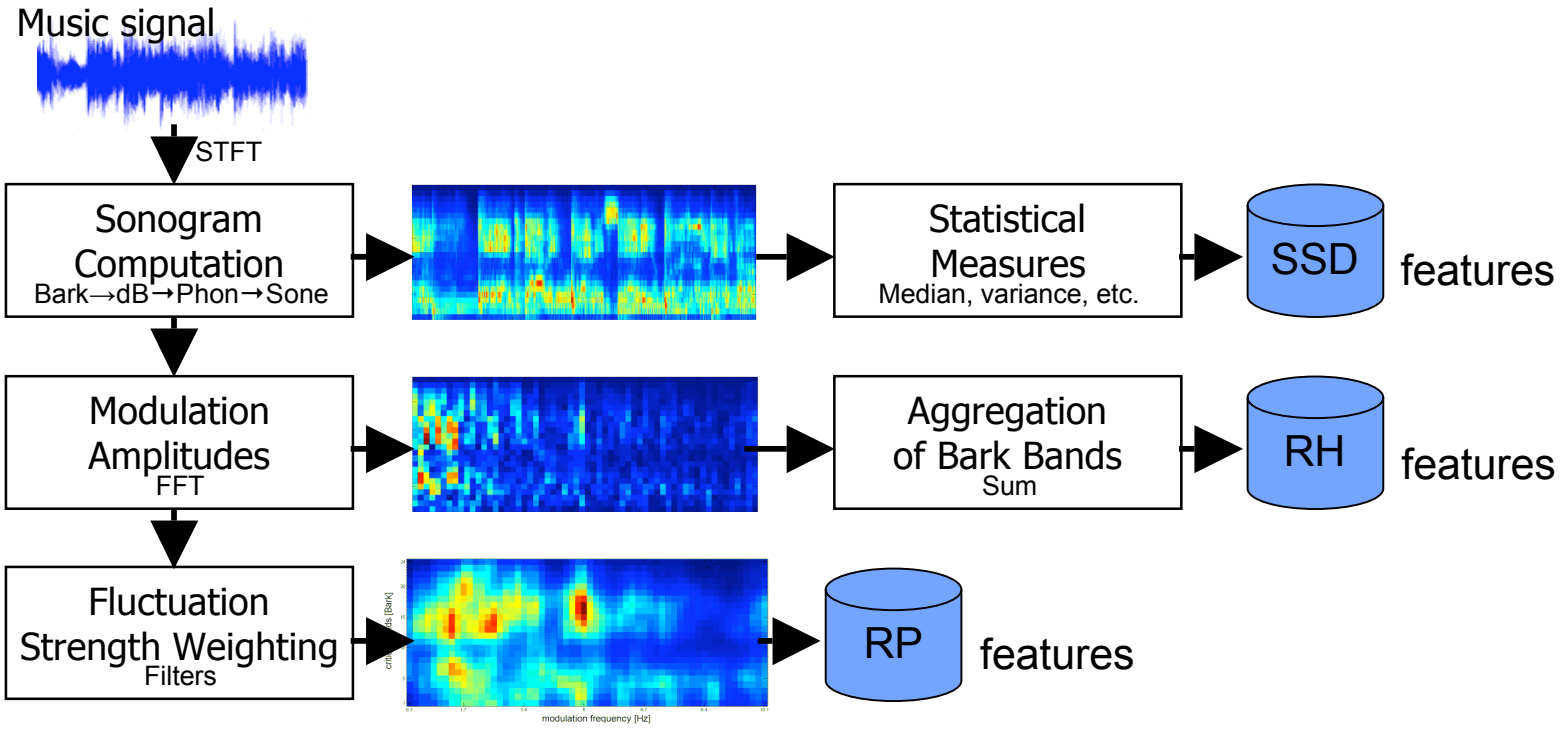
- Content-based 3D object retrieval
- web service for the extraction of 3D descriptors based on curvature correlograms integrated into the ISIS VA-File index
- Further information: Prof. A. del Bimbo, University of Florence





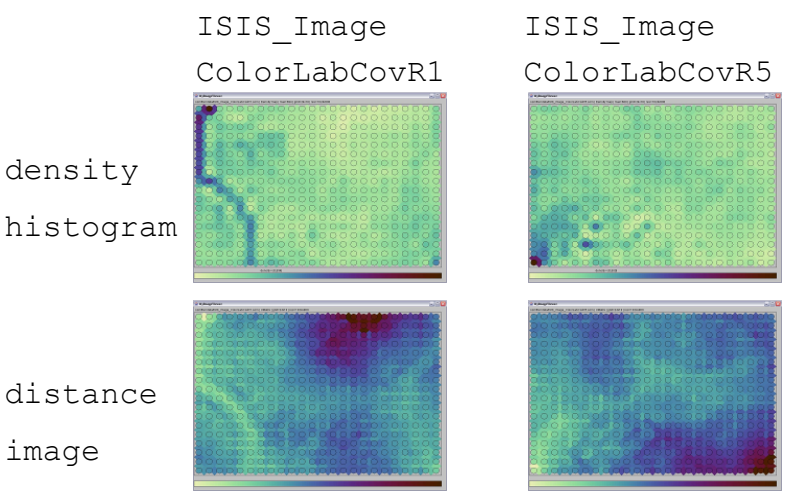
Audio Retrieval (TU Vienna)

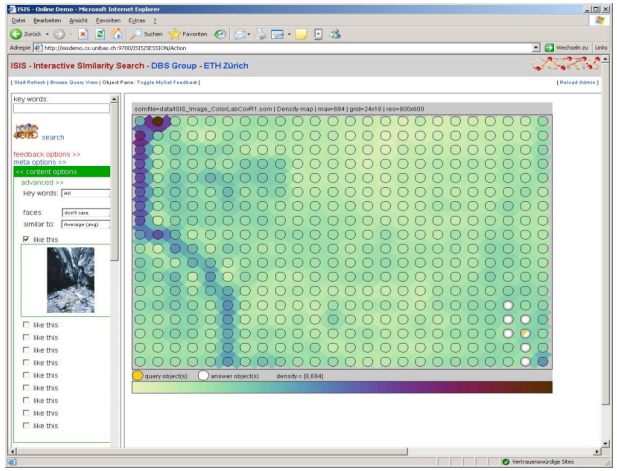
- Specialized (award-winning) audio feature extractors (TU Vienna)
- Offline integration into ISIS index (VA file)
- Further information: Prof. A. Rauber, Technical University of Vienna



SOM Visualization (U Konstanz)

- SOM Visualization of the complete feature space
- Illustration of query object(s) and result set
- Density images, distance images supported
- Further information:
  Prof. D. Keim, University of Konstanz





Daffodil (U Duisburg-Essen)

- Integration of ISIS/OSIRIS content-based image similarity search into Daffodil
- Seamless combination with other Daffodil client functionality
- Further Information:
  Prof. N. Fuhr, U Duisburg-Essen

Active Paper (ETH Zürich)

Active paper front-end to ISIS search

- Keyword search
- Image similarity

Application:

- Exhibition catalogue

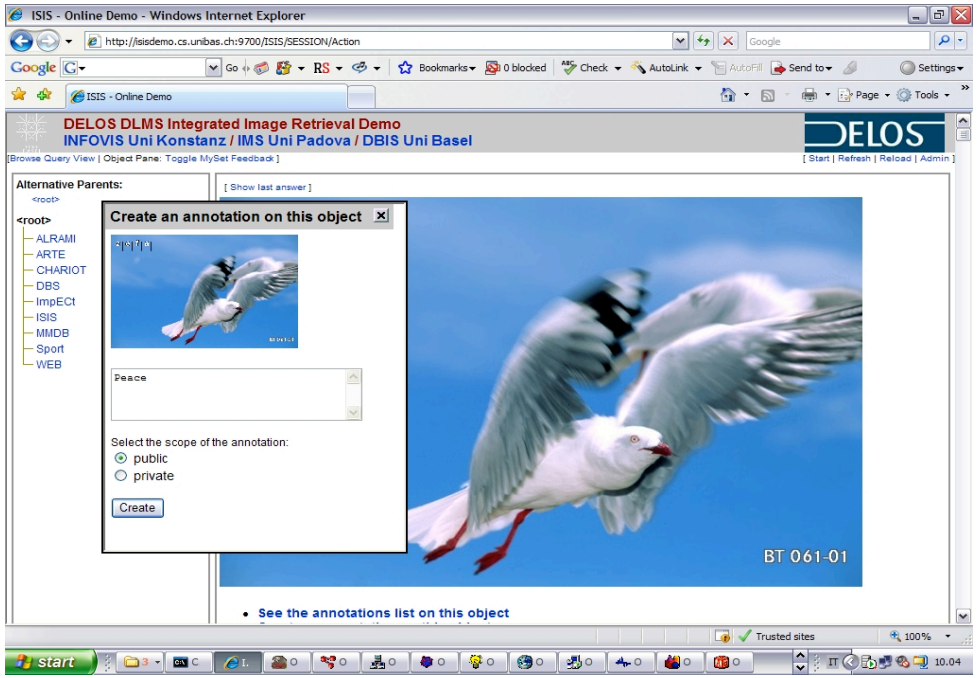
Further information: M. Norrie, ETH Zürich

Further activities (planned)

- Query by Sketching
- Integration of active desk / screen

Annotation Management (U Padua)

- Support for adding annotations to objects added
- Annotations are displayed together with all other metadata of an object
- Keyword search on metadata only, on annotations only or on both
- Further information: Prof. M. Agosti, U Padua

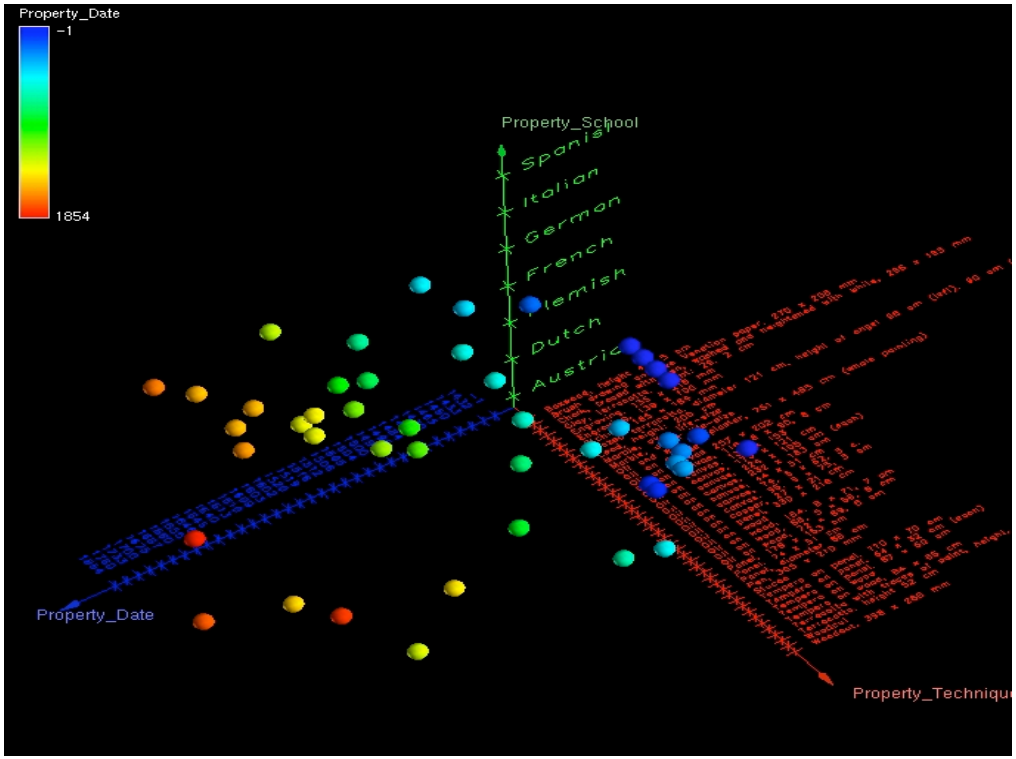


MedioVis (U Konstanz)

- User-centered support of analytical and browsing-oriented search strategies
- Easy-to-use visual search and exploration in DLs
- Multiple coordinated views to simultaneously use different visualizations
- Zoomable user interfaces
- Simple integration of external information sources from the Web (e.g. Google Maps)
- Integration:
  - MedioVis to access DelosDLMS search functionality
  - Visualizations (e.g. DARE, SOM) available through MedioVis
- Further information: Prof. H. Reiterer, U Konstanz

DARE (U Rome)

- Advanced analysis and assistance instrument for exploring large data sets
- Allows users to identify interesting properties, not explicitly expressed in the data set
- DARE is integrated into the MedioVis interface and operates on content coming from DelosDLMS
- Further information:
  Prof. G. Santucci, U Rome



CoCoMa (TU Crete)

Content- and context-aware multimedia content retrieval, delivery and presentation functionality

- Semantic-based Multimedia Retrieval
- Content-based Multimedia Retrieval
- Multimedia Adaptation
- Multimedia Content Delivery
- User Profile Access
- Ontology Access

Further information:
Prof. S. Christodoulakis,
TU Crete

DelosDLMS: Lessons Learned

- Partners develop with different platforms and programming languages
  - Open platform independent service standard necessary
    - ž SOAP Web-services and emerging extensions like WSRF
    - ž SOA is an excellent paradigm for building distributed systems out of existing components from different providers
- Even if technical integration for some service is easy to achieve, semantic integration is in most cases still an issue

DelosDLMS: Future Activities

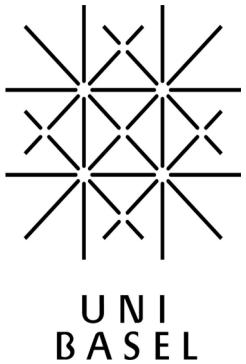
- Combine annotation management and personalization (U Athens)
- Add support for P2P data management and search (Minerva P2P Search Engine, MPI Saarbrücken)
- Term Extraction Services (CNR)
- XML Query Services (CNR, U Konstanz)

Questions & Discussion

Building Digital Libraries on Service Oriented Architectures

Diligent
DIgital Library Infrastructure on Grid ENabled Technology

Tutorial at JCDL 2007
June, 19th 2006



Pasquale Pagano (CNR-ISTI)
Heiko Schuldt (Uni. Basel)
George Kakaletris (Univ. Of Athens)

Agenda

09:00 – 09:20 **Introduction: Motivation & Challenges**

09:20 – 09:45 **Challenges of bringing DL to distributed Infrastructures**

09:45 – 10:30 **Underlying Technologies and their promises** (SOA, P2P, Grid)

10:30 – 10:45 *Coffee break*

10:45 – 12:00 **Solutions for decentralized DL infrastructures** (with BRICKS Demos)

12:00 – 12:30 **DelosDLMS - the DELOS Digital Library Management System**

12:30 – 13:30 Lunch

13:30 – 14:00 **DelosDLMS Demos**

14:00 – 15:00 **Building DL services on the Grid** (DILIGENT)

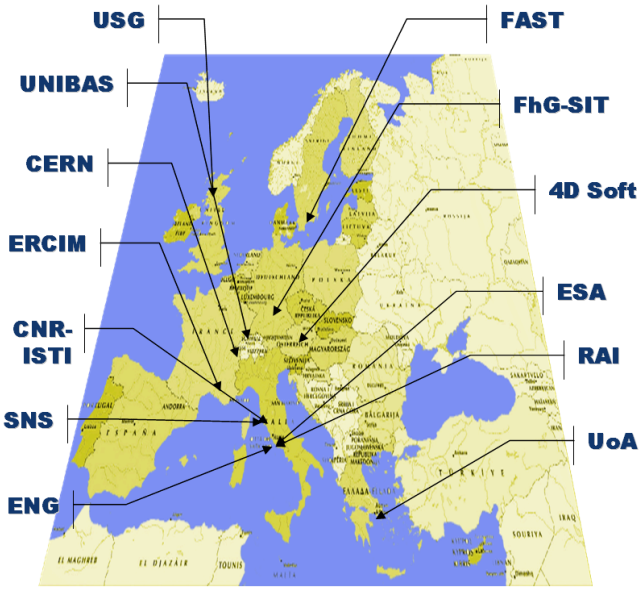
15:00 – 15:30 *Coffee break*

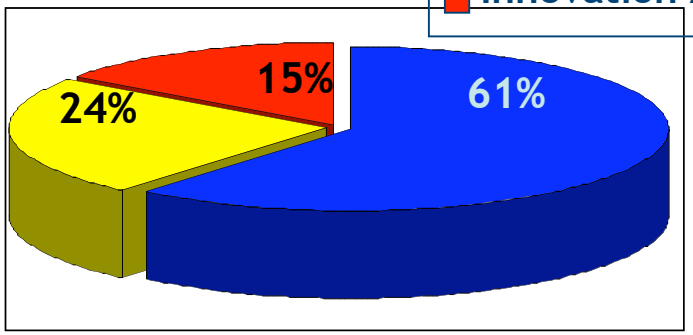
15:30 – 16:45 **DILIGENT Demos**

16.45 – 17:00 **Conclusions and future directions**

Project Numbers

- *Duration*: 39 Months
- *Start/end date*: 09/2004 – 11/2007
- *Effort*: **1150** p/m
- *Cost*: 9.546.561 €
- *EU funding*: 6.300.000 €





European Research Consortium for Informatics and Mathematics (ERCIM) France

National & Kapodistrian University of Athens (UoA) - Greece

Fraunhofer-Secure Information Technology (FhG-SIT) - Germany

European Organization for Nuclear Research (CERN) - Switzerland

University of Strathclyde (USG) United Kingdom

European Space Agency (ESA) Italy

4D SOFT Software Development Ltd. (4D Soft)- Hungary

Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo" (CNR-ISTI) - Italy

University of Basel (UNIBAS) Switzerland

Engineering Ingegneria Informatica SpA (ENG) - Italy

Fast Search & Transfer ASA (FAST) Norway

Scuola Normale Superiore (SNS) Italy

RAI Radio Televisione Italiana (RAI) Italy

In a nutshell

Provides a

test-bed Infrastructure on Grid-ENabled Technology

that allows members of dynamic virtual organizations to create on-demand transient virtual digital libraries
based on shared computational and storage resources
to exploit applications and multimedia and multi-type content.

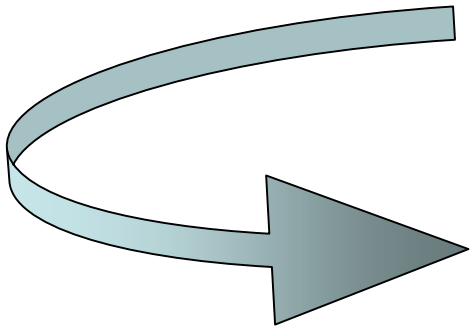
In a nutshell

Information Society Technologies Cont.

Provides a

test-bed Infrastructure on Grid-ENabled Technology

that allows members of dynamic virtual organizations to create
on-demand transient virtual digital libraries
based on shared computational and storage resources
to exploit applications and multimedia and multi-type content.

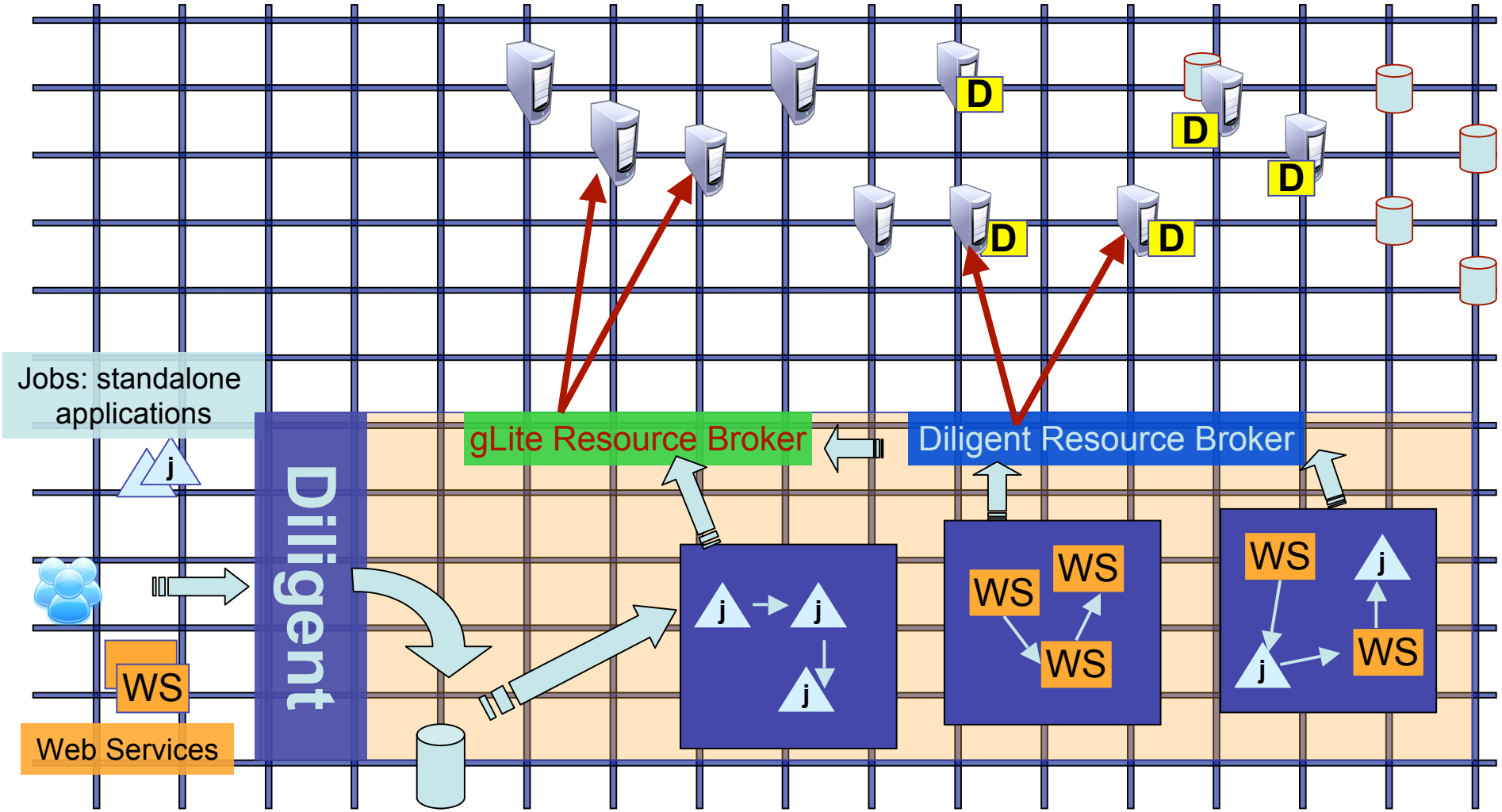


Dynamically identified content sources equipped with the functionality to access, discuss, evaluate, and enhance digital objects of all forms

Diligent objectives

- *management of large number of distributed VOs;*
- *access to and handling of distributed multi-focused data and services;*
- *on-demand efficient processing of huge amounts of data;*
- *orchestration of user defined services, with defined QoS (e.g. scalability, reliability);*
- *knowledge preservation*
  - *storage of derived data as well as dependencies*
  - *traceability of the operations performed*

Diligent Infrastructure



Diligent Infrastructure

Information Society Technologies Cont.

Diligent is an infrastructure for distributed DL applications that are designed as a set of cooperating services.

The infrastructure is composed by:

- the set of grid resources (computing, storage)
- the set of DL services (content and storage management, search, index, …)
- the set of processes defined to manage such resources
- the set of collections created to bring together the content to manage
- the set of enabling services (core services: information system, Keeper Manager, …)

Diligent Infrastructure

Information Society Technologies Cont.

Consumers

Diligent Infrastructure

Service A

Service B

Service C

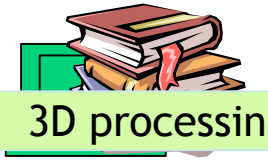
DLCreati
ser

Service D

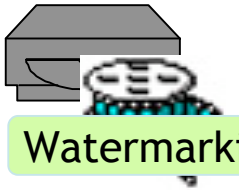
Service E

Open and dynamic infrastructure is a KEY

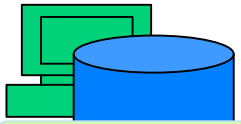
Providers



3D processing



Watermarking

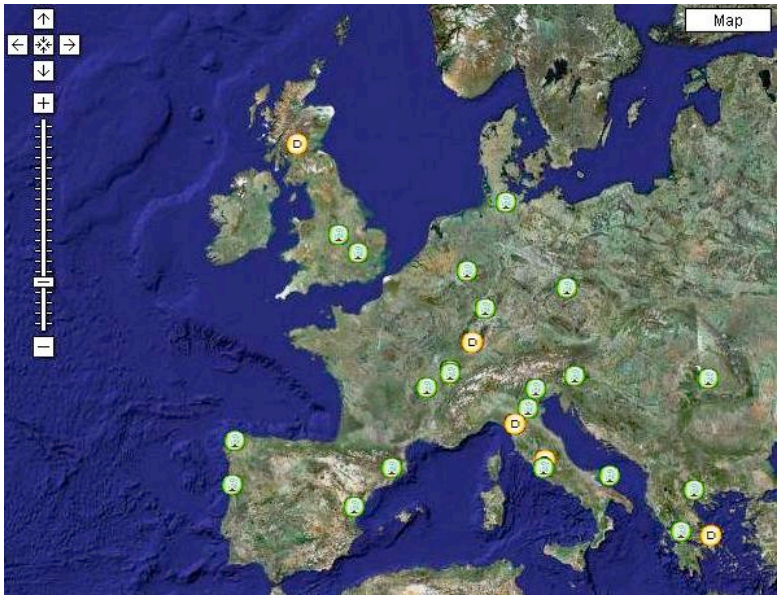


Feature extraction

Speech recognition

Diligent infrastructure today

- Across Europe



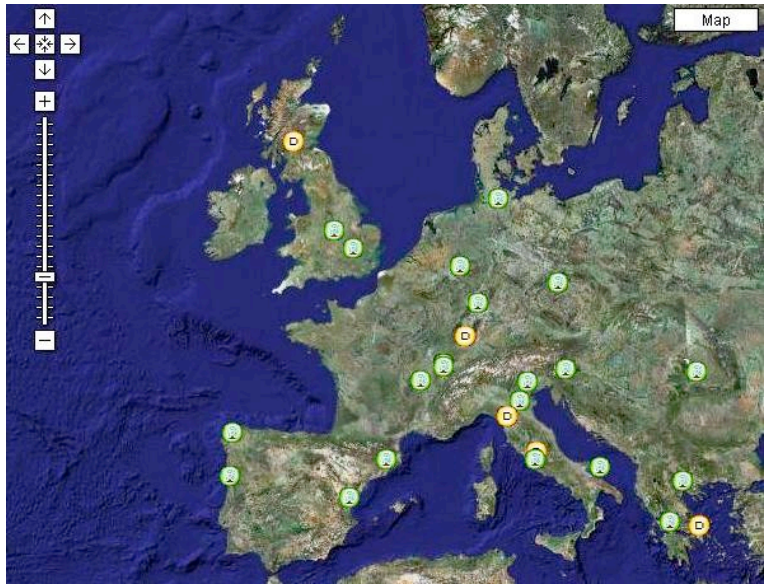
- Across Infrastructure
  - ✓ ESA EO Grid infrastructure
  - ✓ EGEE PPS infrastructure
  - ✓ Diligent infrastructure

- To serve
  - ✓ Earth observation VO – ImpECt
  - ✓ Cultural heritage VO - ARTE

Diligent infrastructure today

Cont.

- ✓ Content generation
  - ✓ High level geophysical products generation (e.g. mosaics, chlorophyll distribution profiles, vegetation index, atmospheric profiles)
  - ✓ Environmental reports generation



- ✓ Content protection
  - ✓ watermarking

Diligent for the DL communities

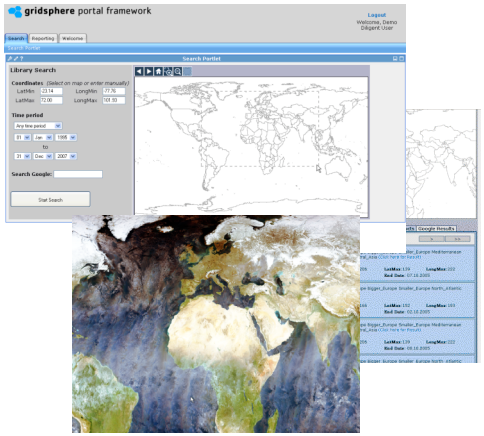
- The DILIGENT service infrastructure dramatically changes the **DL development model** used by distributed and dynamic organisations and communities

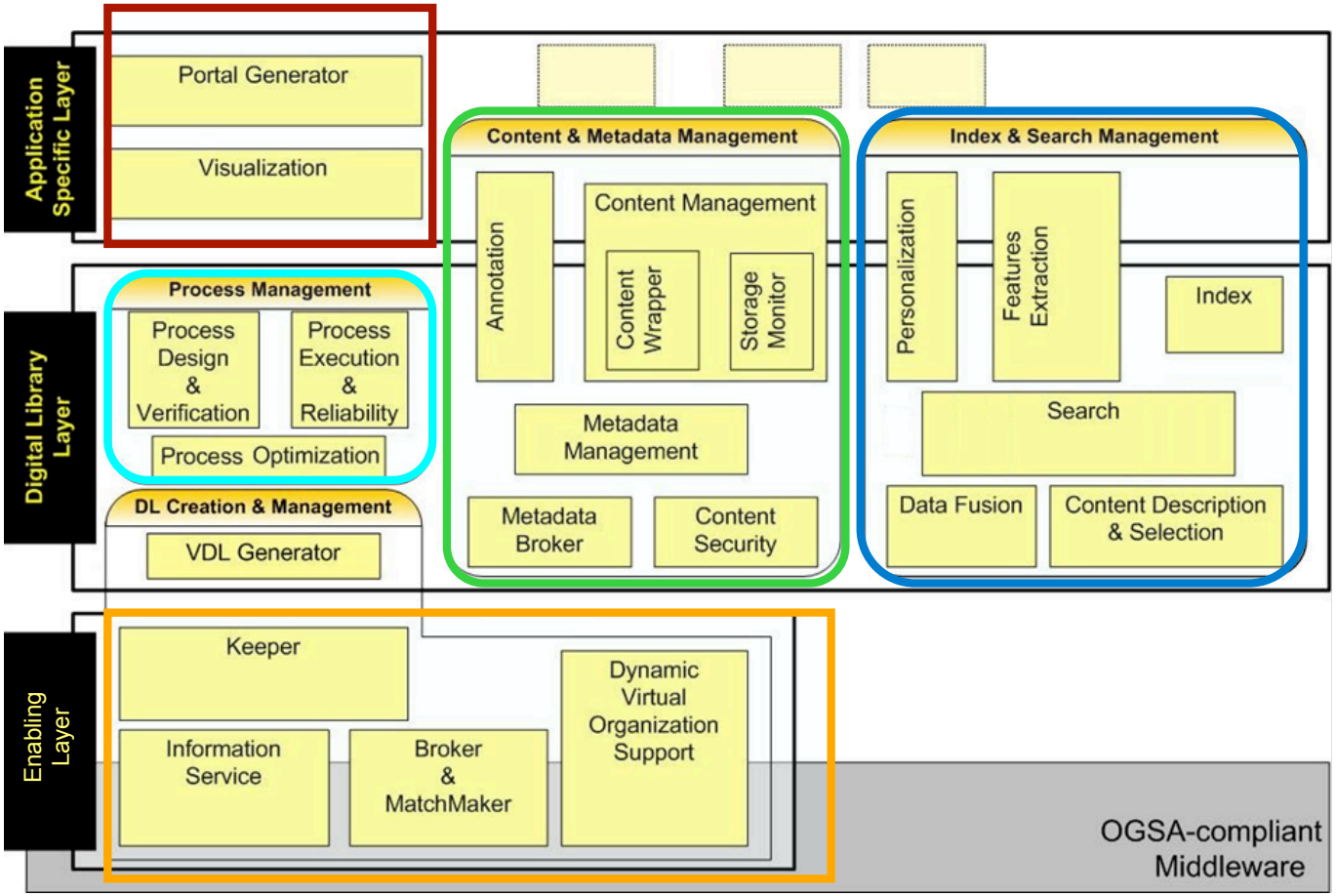
- Using DILIGENT, the organisations and communities are able to setup their own environment:

  - When and for the time they need it
  - Exploiting existing Grid-based services
  - Accessing to and handling of distributed multi-focused data and services
  - Orchestrating user defined services, with defined QoS (wrt. scalability, reliability)
  - Profiting from a shared storage and computational set of resources
  - Sharing data and services in a collaborative and efficient way

Diligent Architecture

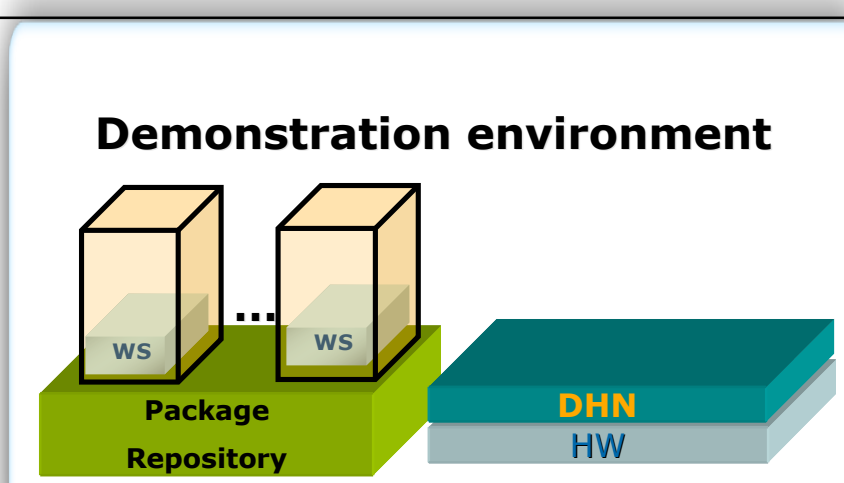
Diligent Architecture

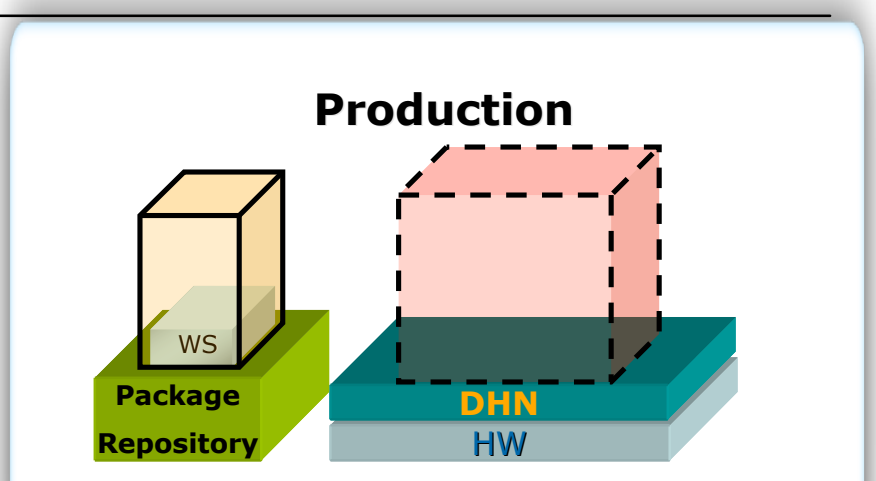


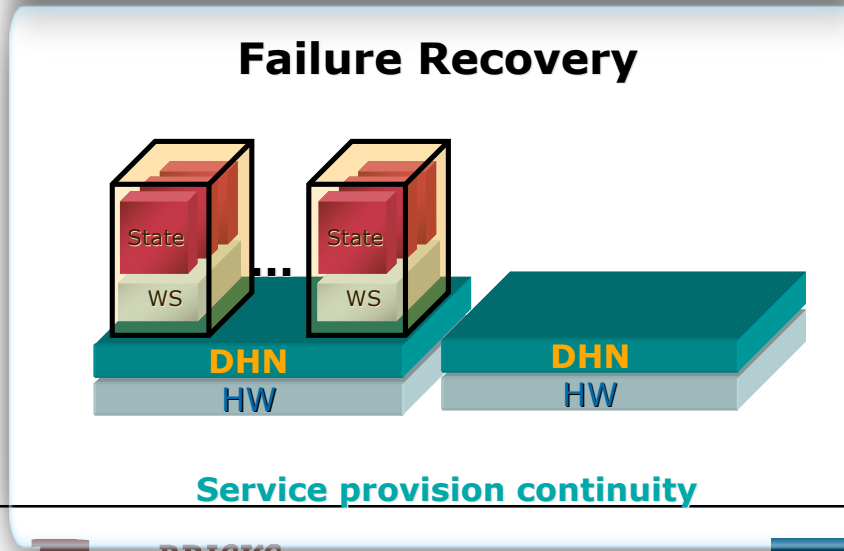


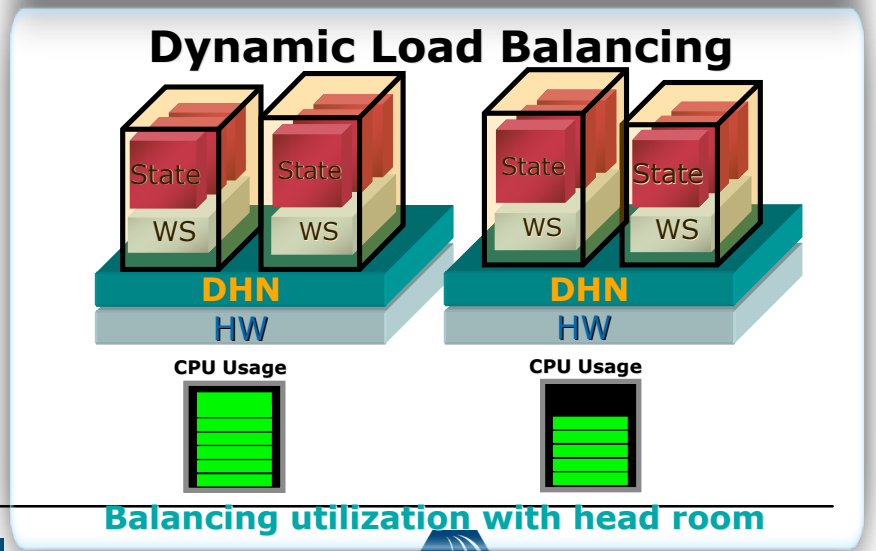
Diligent about Services

Diligent about Services









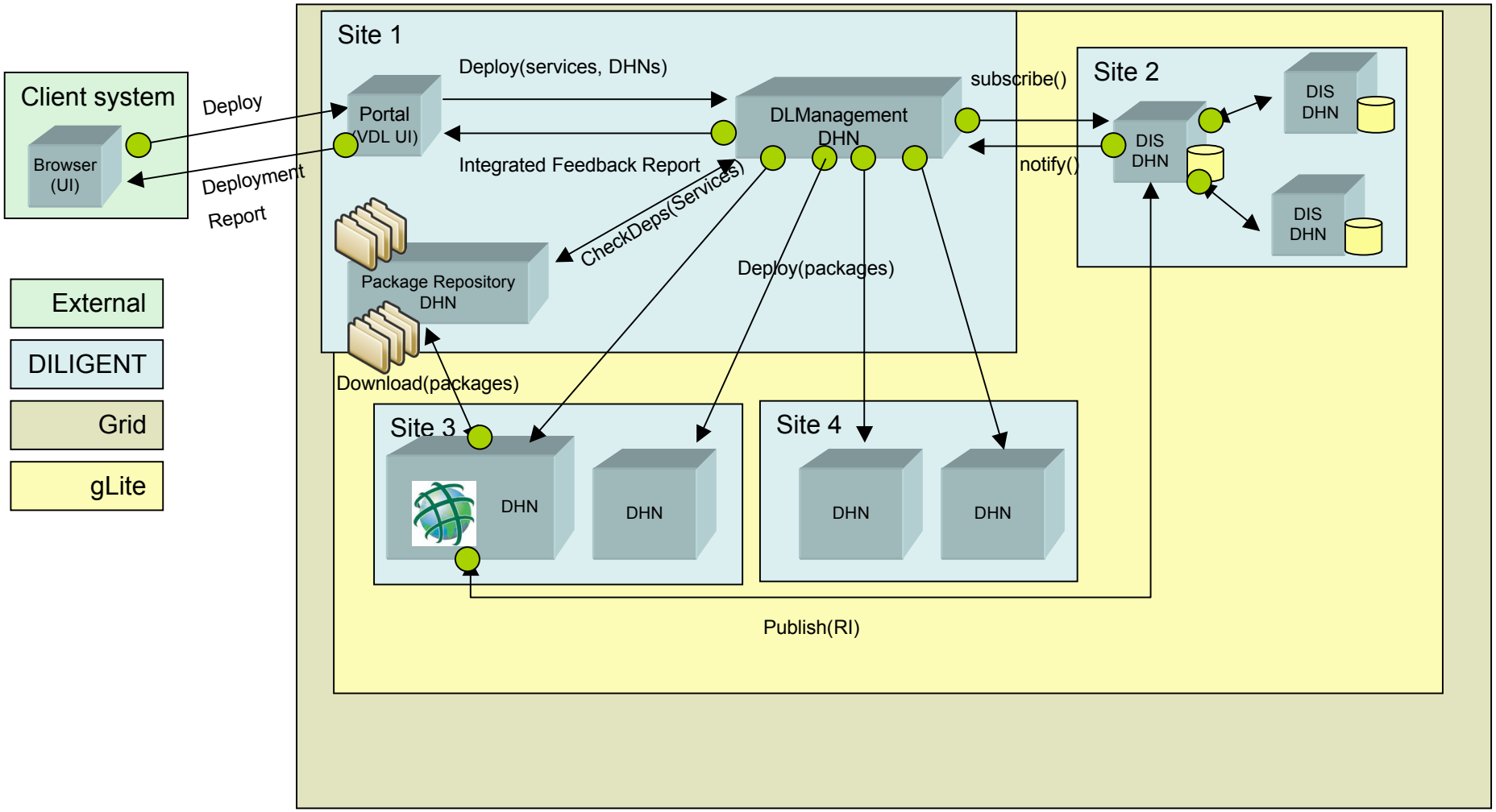
Diligent about Services

Infrastructure supports the service management

- by enabling:
  - Remote deployment
  - Environment configuration
  - Lifetime management
  - Service provision continuity
  - Usage normalisation

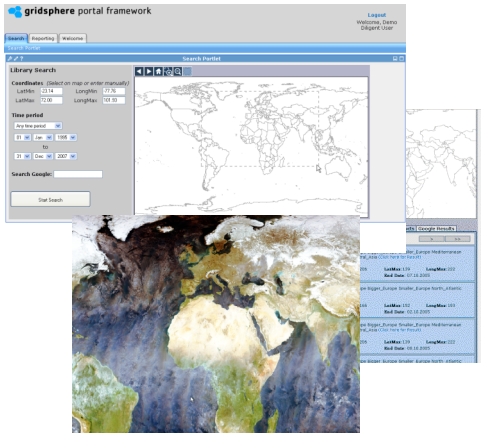
- and transparently managing:
  - Failure recovery
  - Dynamic load balancing

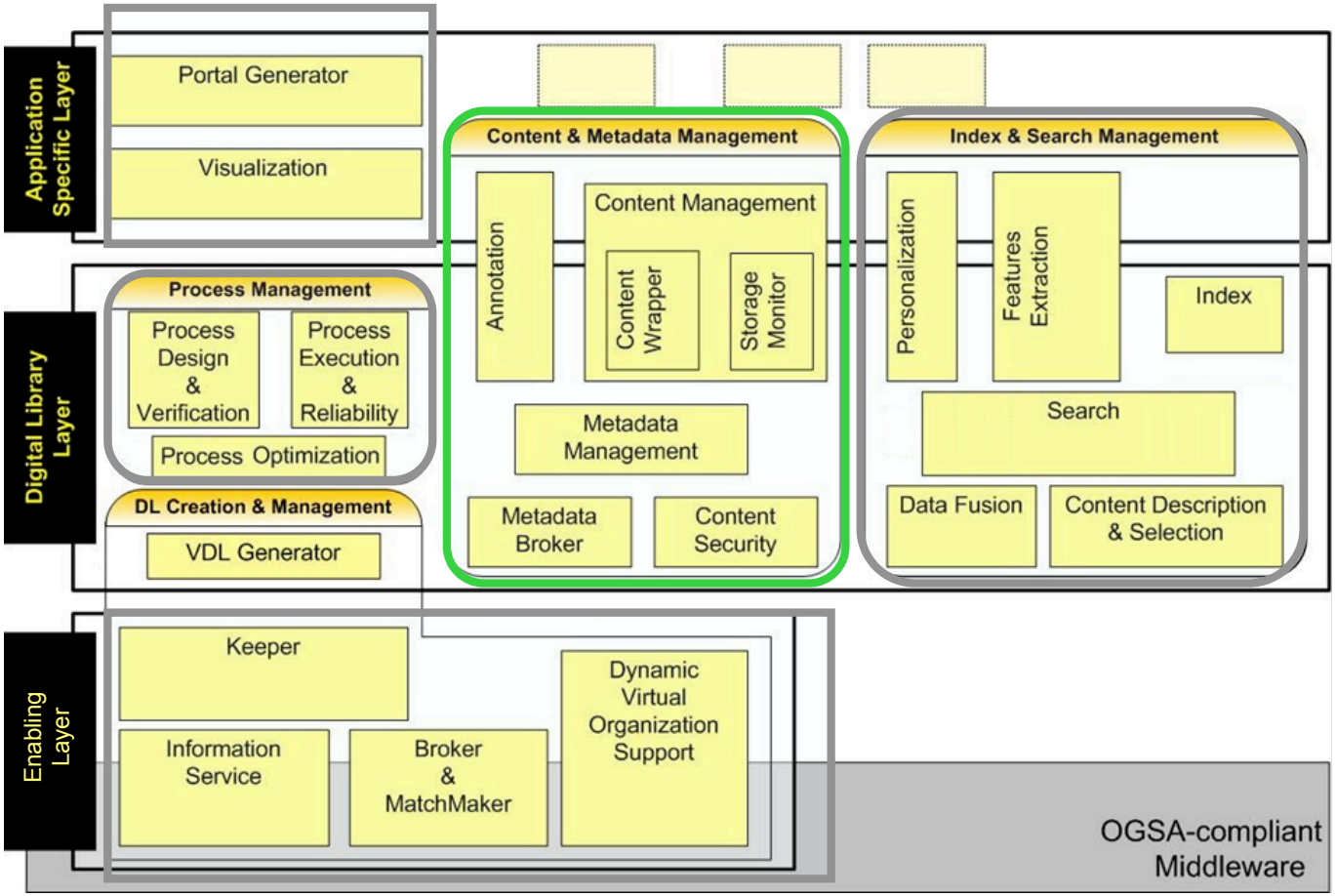
Dynamic Deployment Operations



Diligent about Content

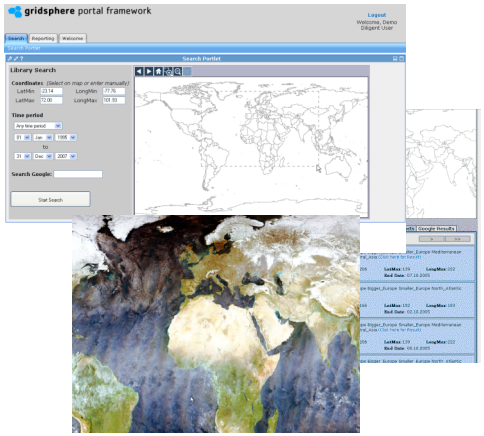
Diligent Architecture

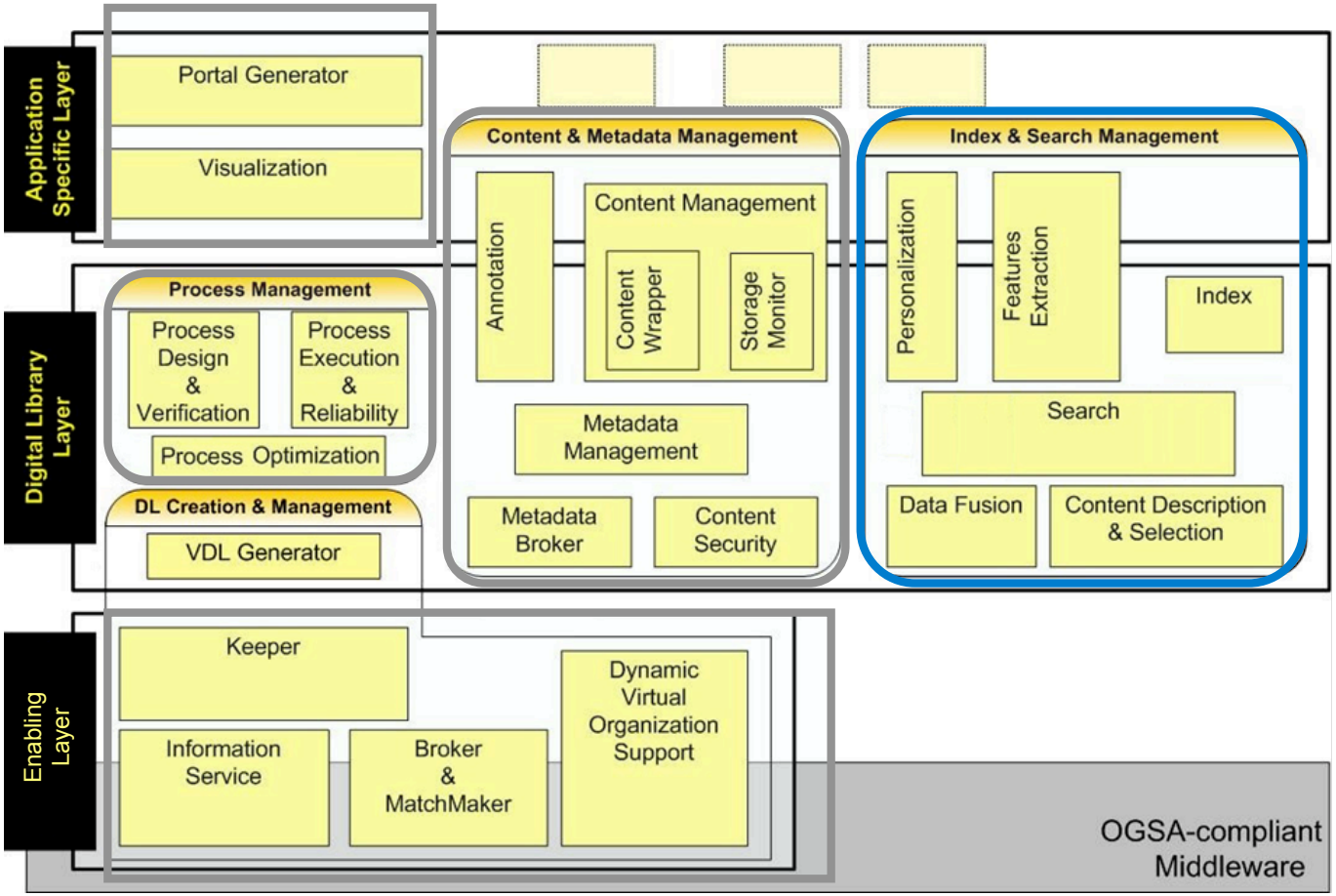




Diligent about Search

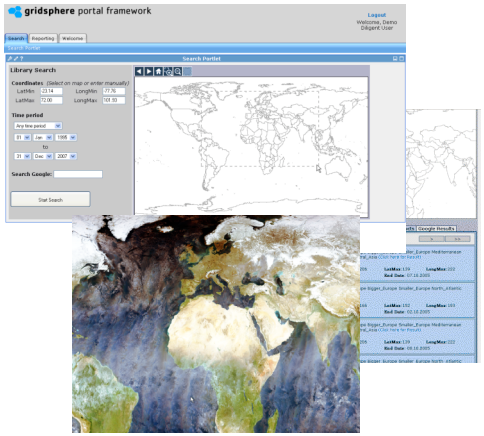
Diligent Architecture

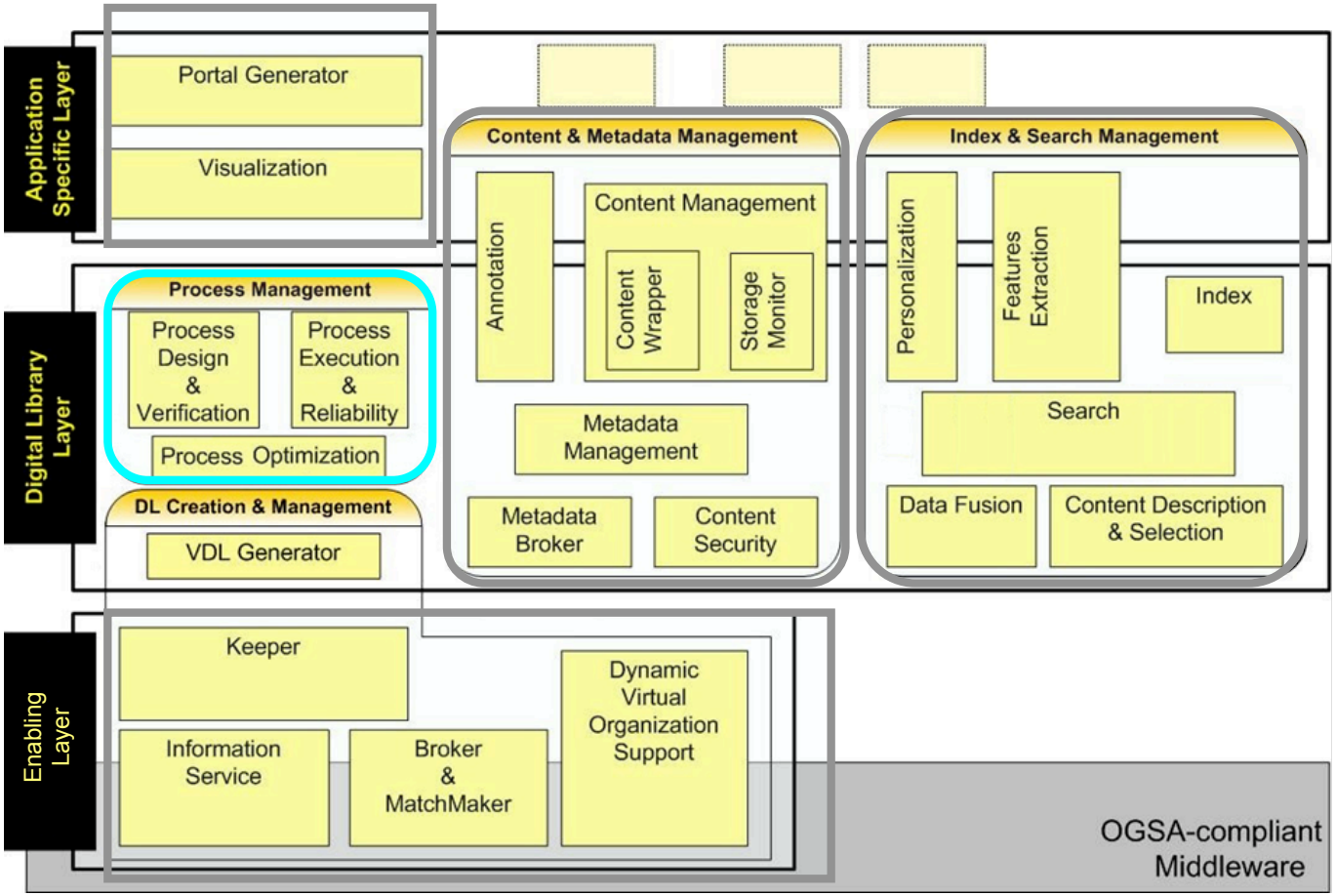




Diligent about Process Management

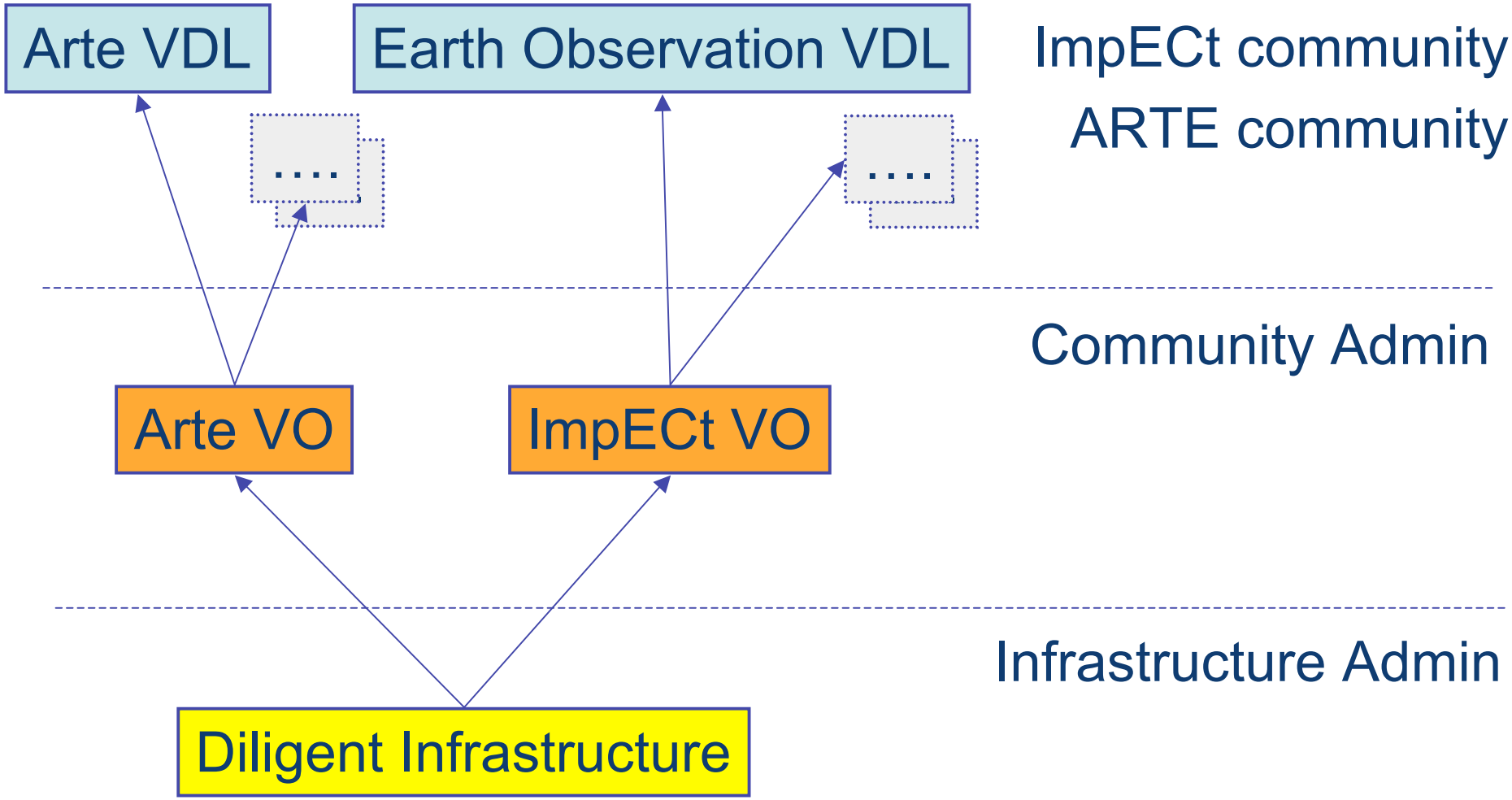
Diligent Architecture



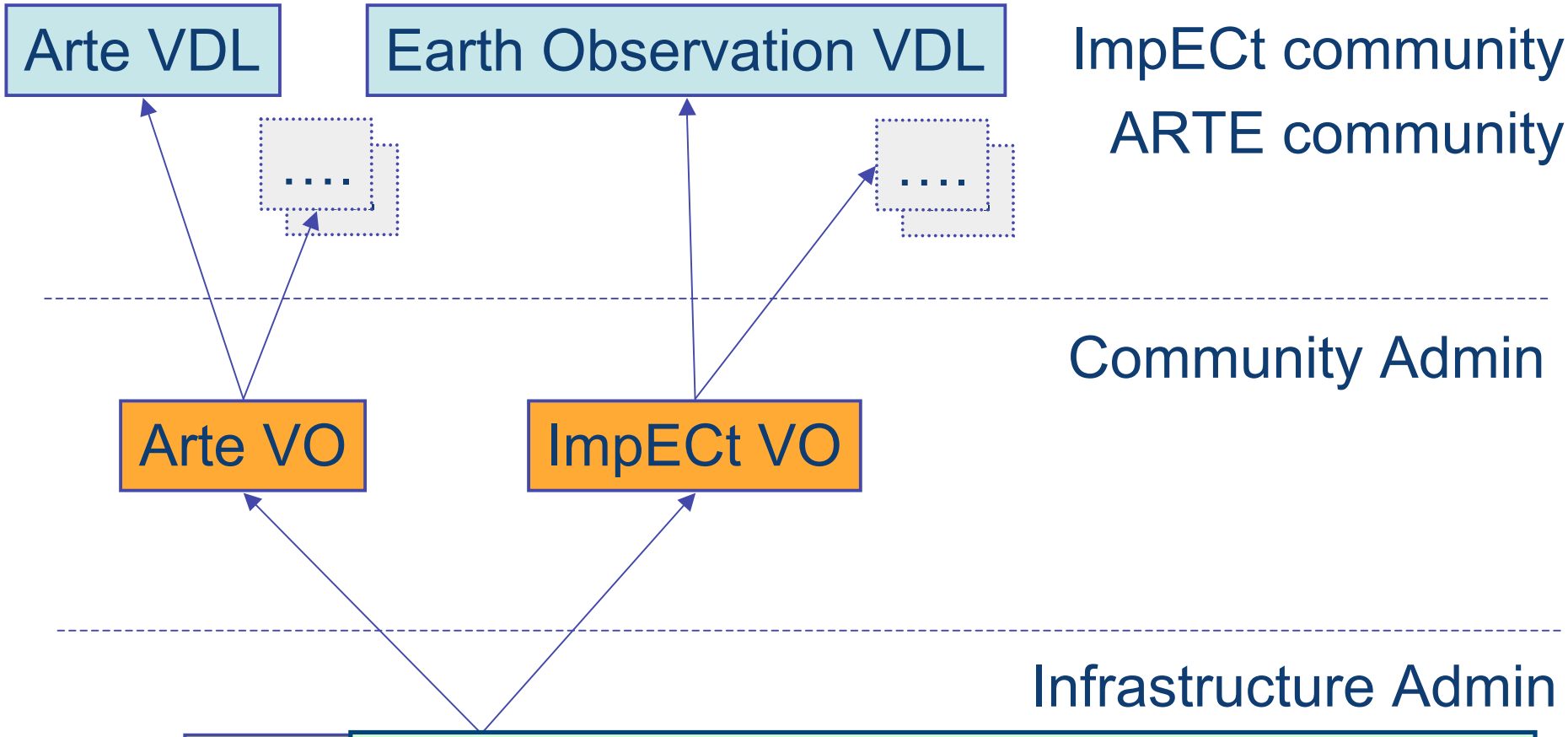


Demos

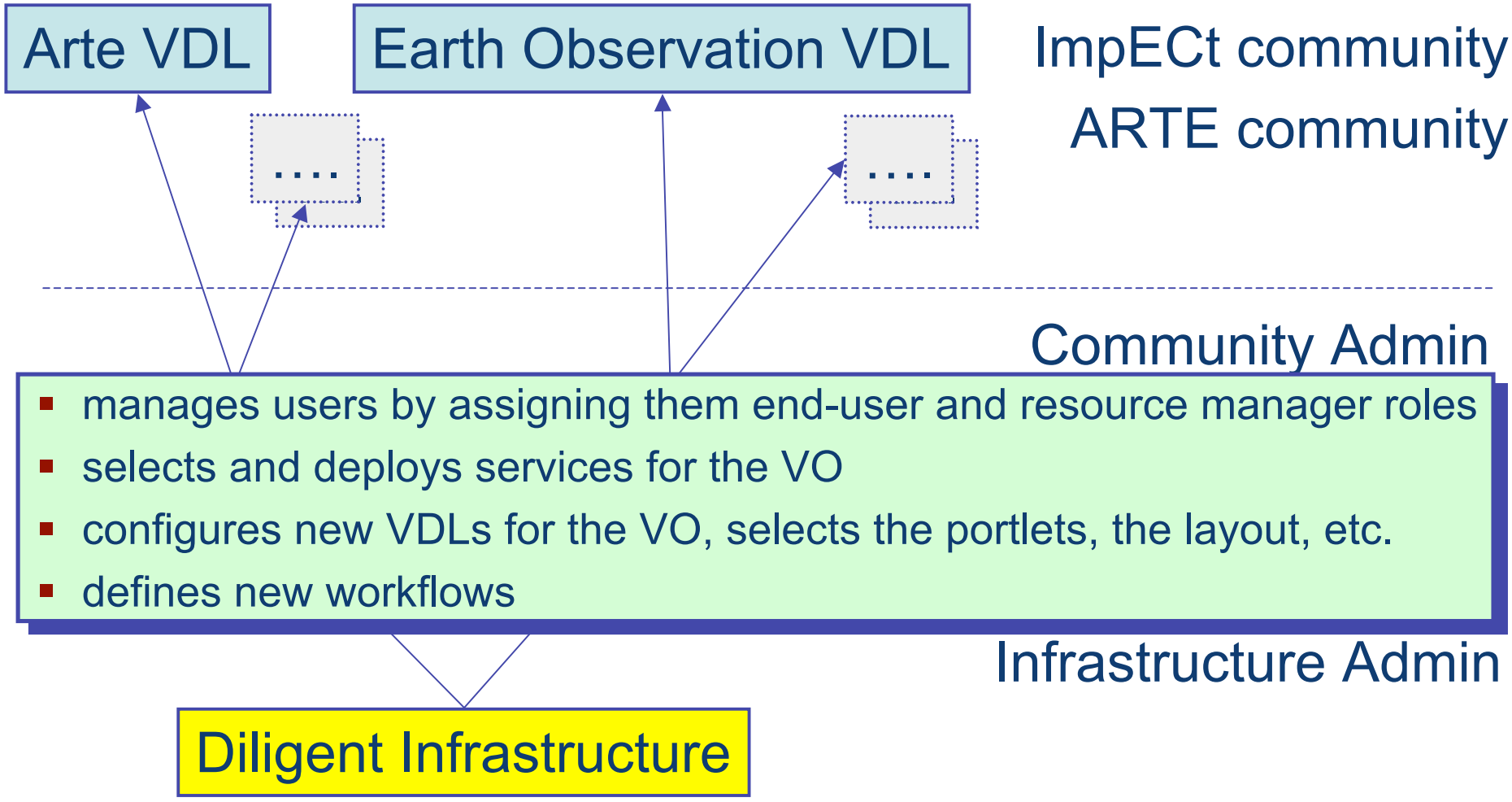
DILIGENT Infrastructure



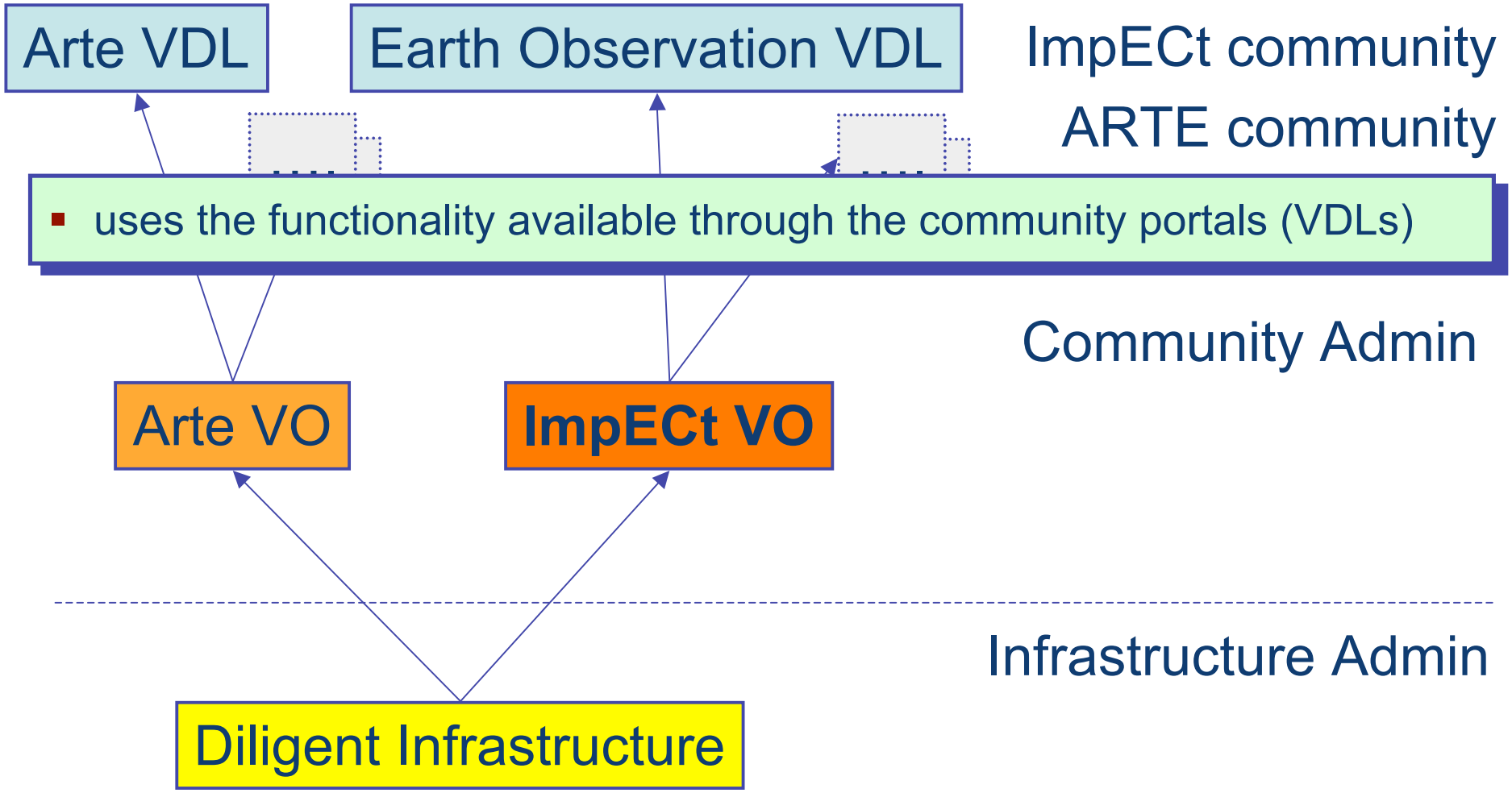
DILIGENT Infrastructure



DILIGENT Infrastructure



DILIGENT Infrastructure



ImpECt use case

Build an ad-hoc user-defined virtual research environment where content, applications and services collaborate together to generate on-demand new valuable information, e.g. complex environmental reports

ImpECt Resources from the Web...

ImpECt Resources – Services

- ✓ High level geophysical products generation (e.g. mosaics, chlorophyll distribution profiles, vegetation index, atmospheric profiles)
- ✓ Environmental reports generation
- x Data analysis and visualisation services

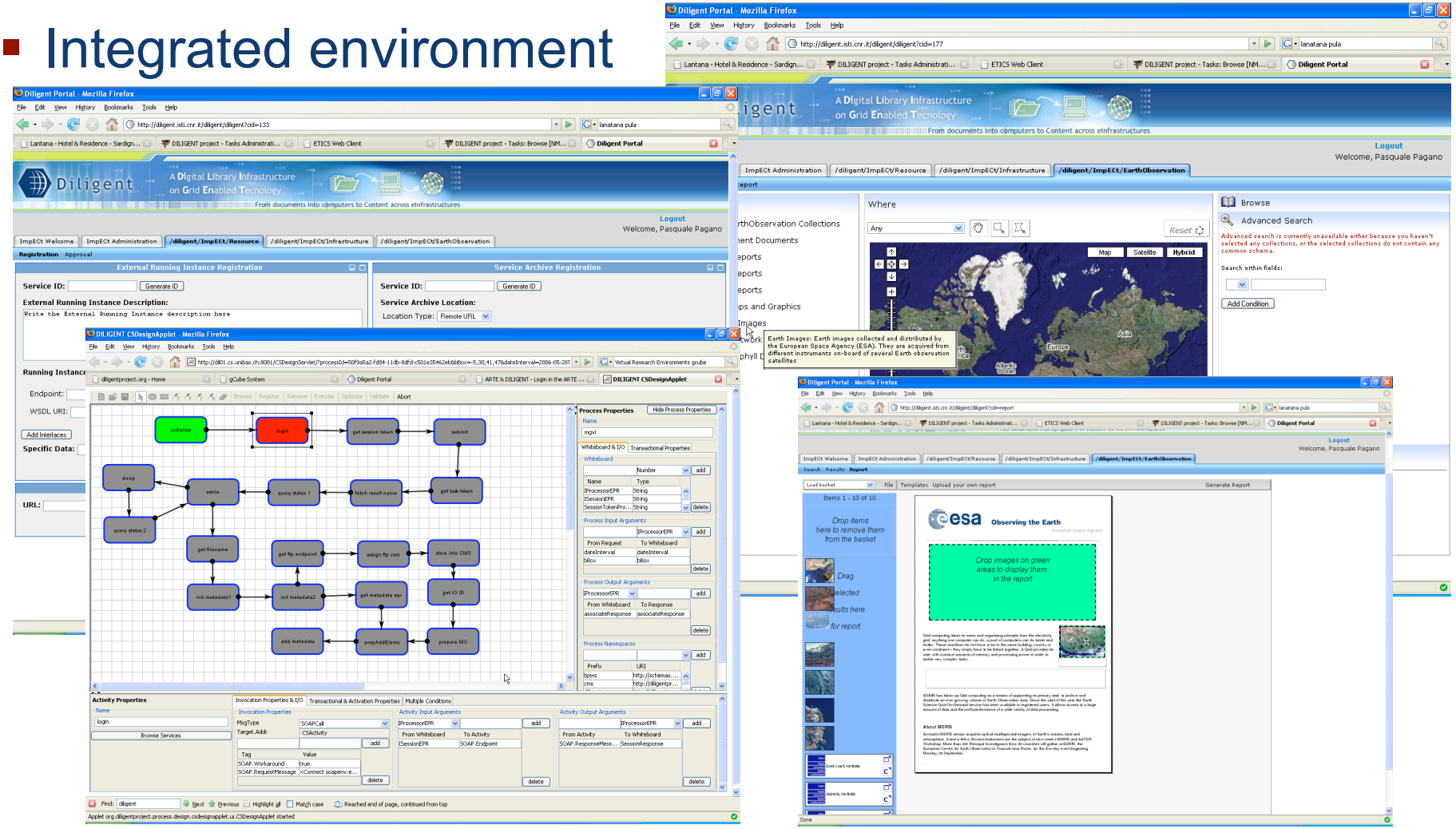
from

- ✓ ESA Grid-on-demand (eogrid.esrin.esa.int)
- ✓ Diligent ad hoc developed
- x EO web portal (www.eoportal.org)
- x GMES services (www.gmes.info)

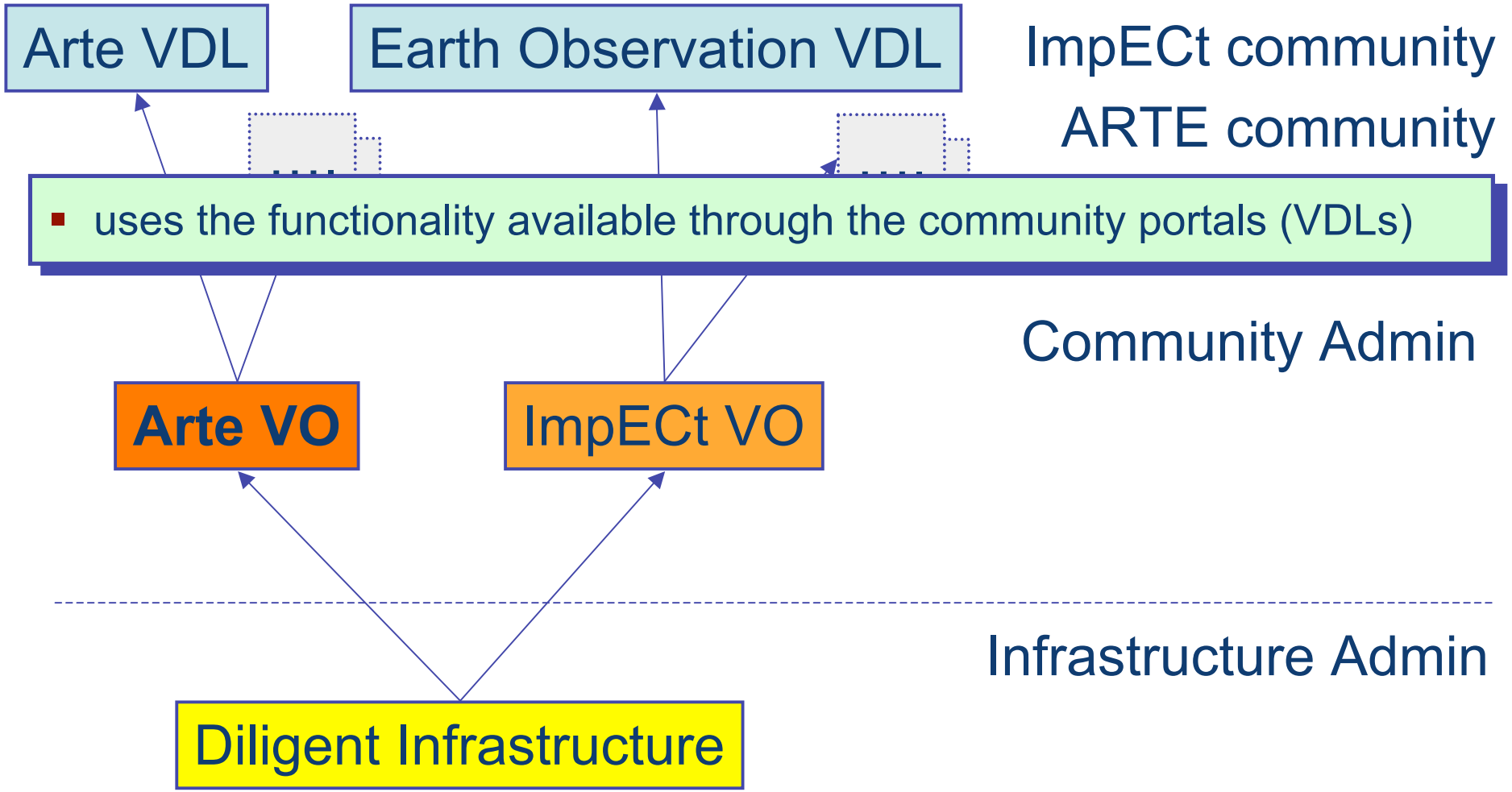
x will be integrated soon

Diligent for ImpECt

- Integrated environment



DILIGENT Infrastructure

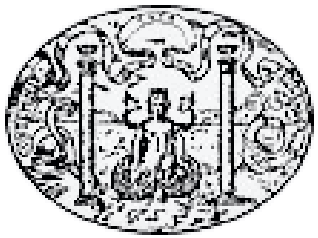


The ARTE Community

- **ARTE** (Applicazione di Ricerche e Tecnologie di Editoria digitale) is a community of researchers coming from scholars located all over the world, that work together to set up the basis for a new research discipline that merges experiences from the humanity, social science, and communication research areas.
  - SNS (Scuola Normale Superiore): Studies the complex relationship between words and images in the literary tradition with computerized tools
  - RAI: Italian's public broadcaster who aims to disseminate history, science, art and culture through audiovisuals

What is it about ?

- Sharing
  - “machinery”
  - information and knowledge
- Reduce cost of ownership and use
- Open new opportunities for processing content
- Bring together multidisciplinary domains:
  - To user service
  - To research scope
- Service domain aware users
- *Inter-operability of information*
- A test-bed for proving and discarding concepts



ARTE Archives Past Obstacles Volumetrics & Specialties

- Heterogeneous collections of high diversity
  - Several data formats
  - Embedded data issues
  - Different availability
- Unconventional collection constituency
  - "Small" number of items per collection
  - Complex documents with well defined metadata
- Images resistant to similarity search
  - High complexity
  - Black-and-white
- Huge volume videos without rich metadata

Conclusion

- To promote knowledge exchange
- To increase applications interoperability

➲ to allow communities to access, aggregate and manipulate data and services from a wide variety of independently information sources and resource providers

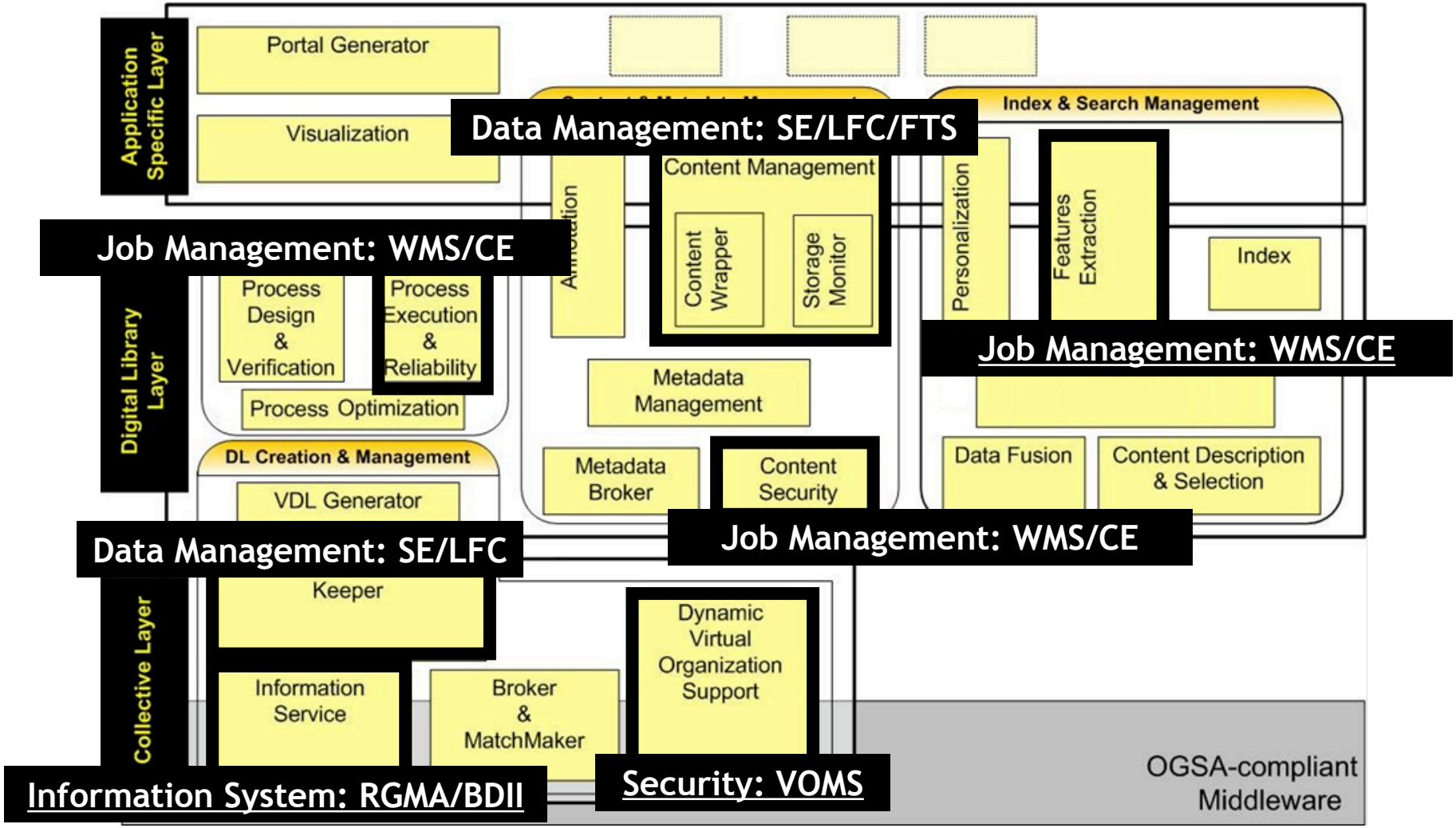
Links

Web site: www.diligentproject.org

Info: info@diligentproject.org

Additional slides

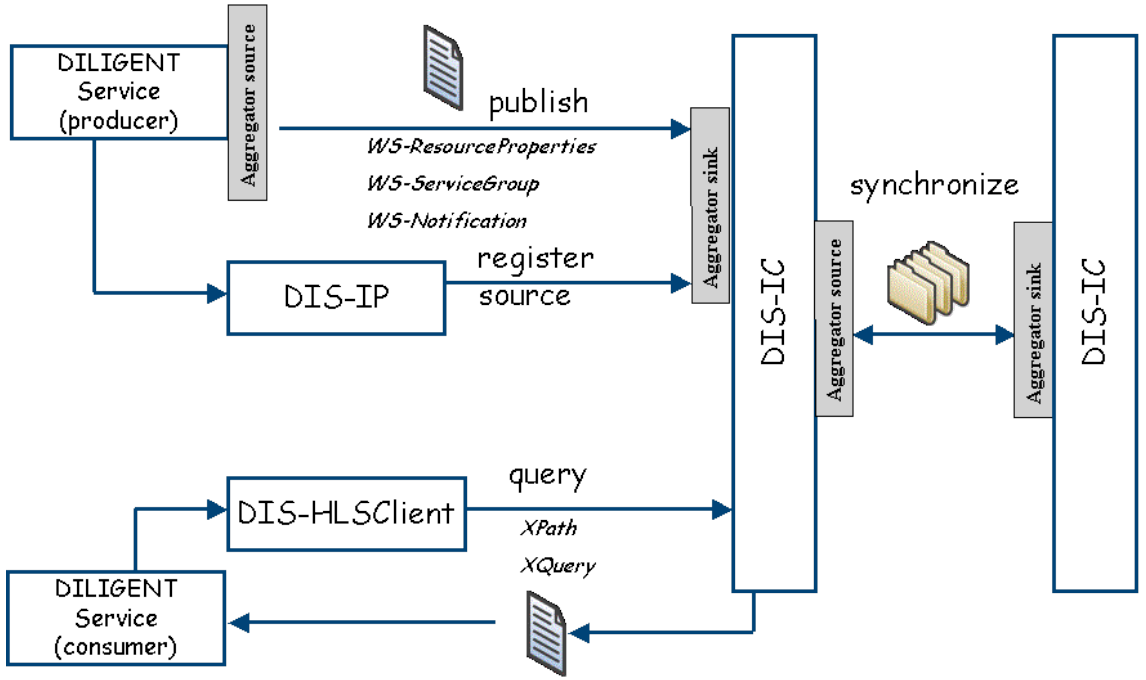
Diligent WSs – gLite grid services interactions



DILIGENT Information System

The DILIGENT Information System (DIS) service is responsible for the aggregation, storage and monitoring of information about DILIGENT resources. This information is provided through query and subscription interfaces.

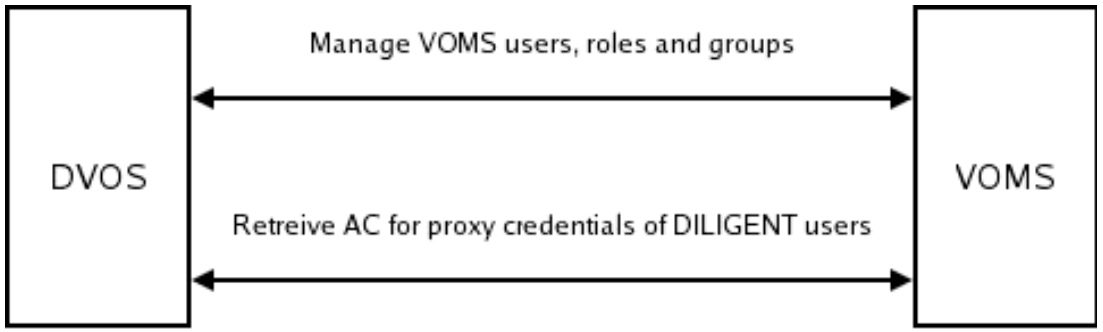
The DIS exploits gLite by aggregating information about gLite resources (SEs, CEs) from BDII.



Dynamic Virtual Organization Support

The Dynamic Virtual Organization Support (DVOS) service provides a robust and flexible security framework based on an advanced authentication and authorization model. It also offers notification support and dynamic aggregation of resources and users/groups

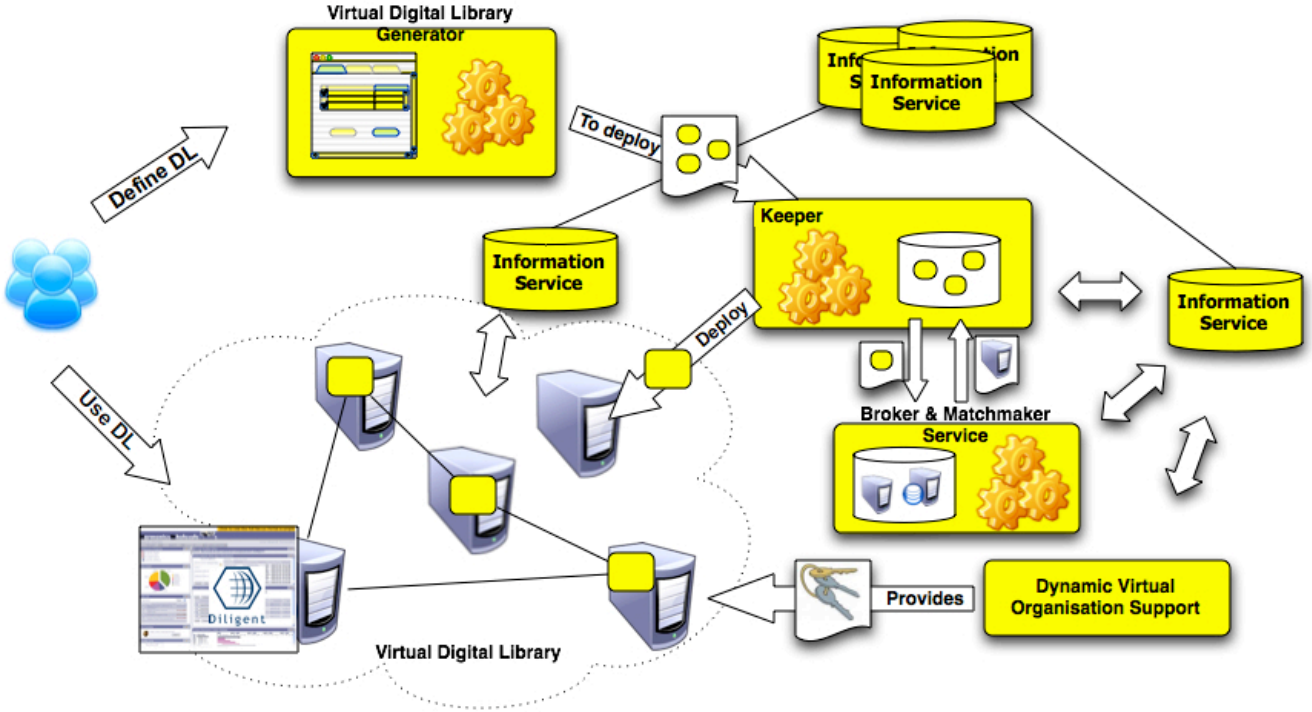
The DVOS exploits gLite by relying in VOMS for its AuthZ service



Keeper

The Keeper service is responsible for the creation of DLs. This includes the management of the DILIGENT packages, the selection of the appropriate DILIGENT resources and the dynamic deployment, monitoring and re-allocation of DILIGENT services.

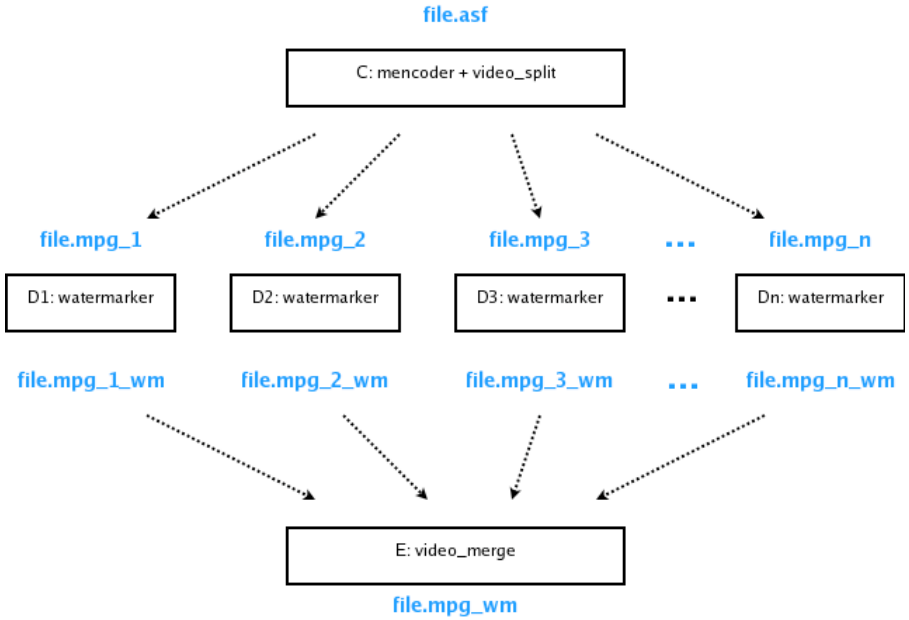
The Keeper exploits gLite by storing the DILIGENT packages in gLite SE.



Content Security

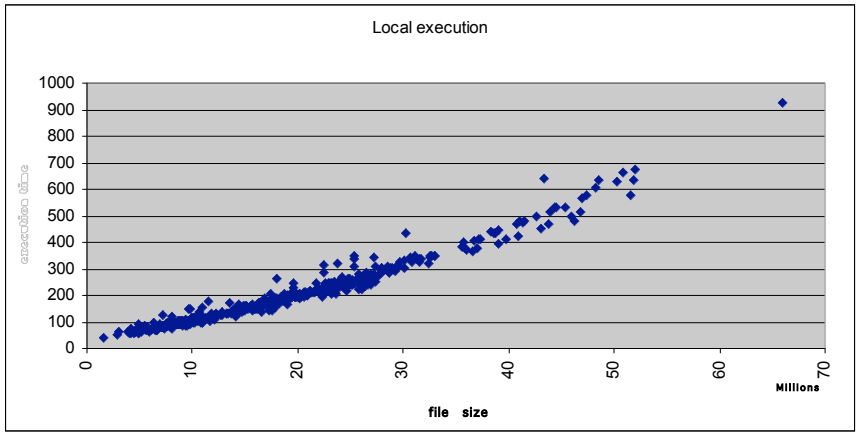
The Content Security service deals with the specialties of protecting multimedia content providing authenticity, integrity and confidentiality to the DILIGENT data through the execution of grid-enabled watermarking algorithms for different media types.

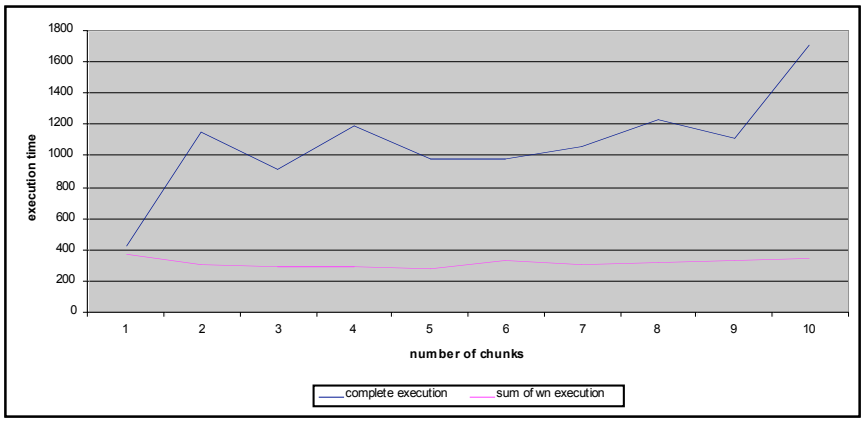
- The Content Security exploits gLite by executing a gridified version of the watermarking application in gLite CEs.



Content Security

- Analysis of remote execution - Strategy
  - Conversion from asf to mpeg format
  - Splitting to n independent chunks
  - Gridified parallel watermarking of every chunk
  - Merging of marked chunk to complete video
- Local Execution
  - Dual + HT Intel(R) Xeon(TM) CPU 2.80GHz
  - Linear execution of watermark embedding in video collection
  - Completion time: 28.9 hours
- Grid Execution
  - Distribution of watermark embedding submission to 39 worker nodes
  - 535 successful, 86 resubmissions
  - Completion time: 2.6 hours





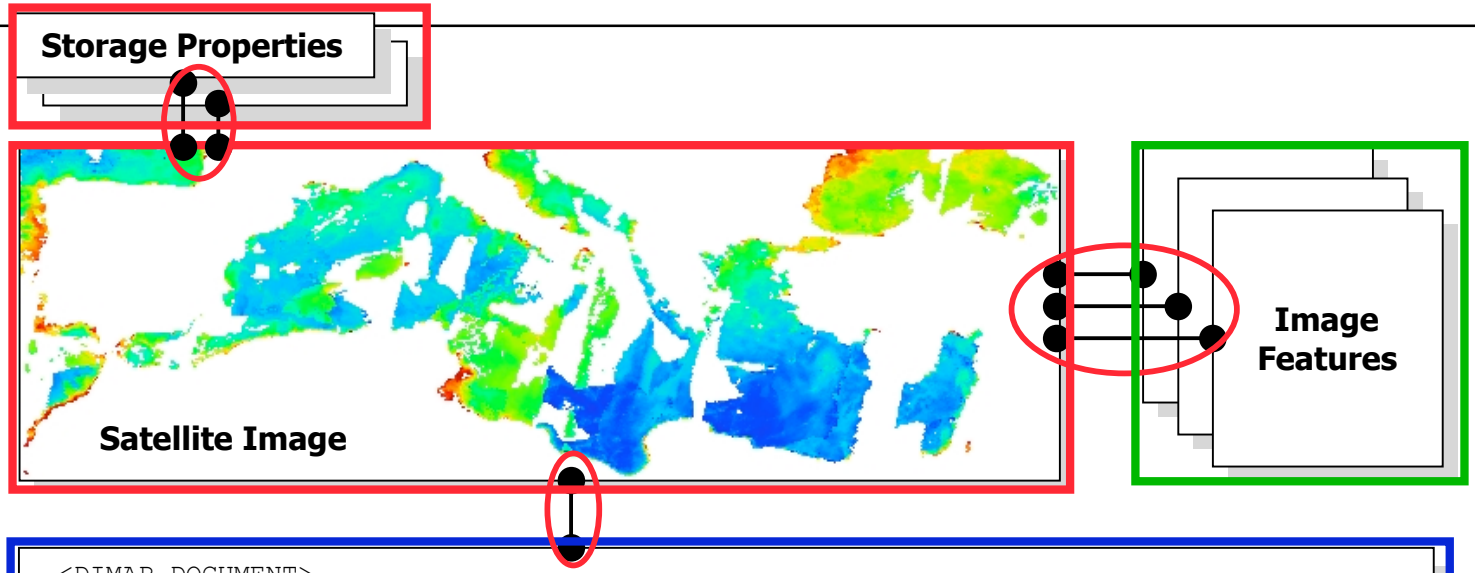
Content management: Introduction

- The Content and Storage Management provides means for
  - persistently storing,
  - physically structuring, and
  - efficiently fetching

  any content in the Grid enabled Digital Library
- Current Grid technology focuses mostly on performance for batch processing. But: a DL also requires
  - more than file-system-like functionalities is required to manage content
    - ž e.g., Content must be interrelated in multiple ways and attributed by various application specific attributes
  - real-time performance while searching and browsing the content
- All content management-related functionality is provided as service

Sample Document (Earth Observation)

Content & Storage Management



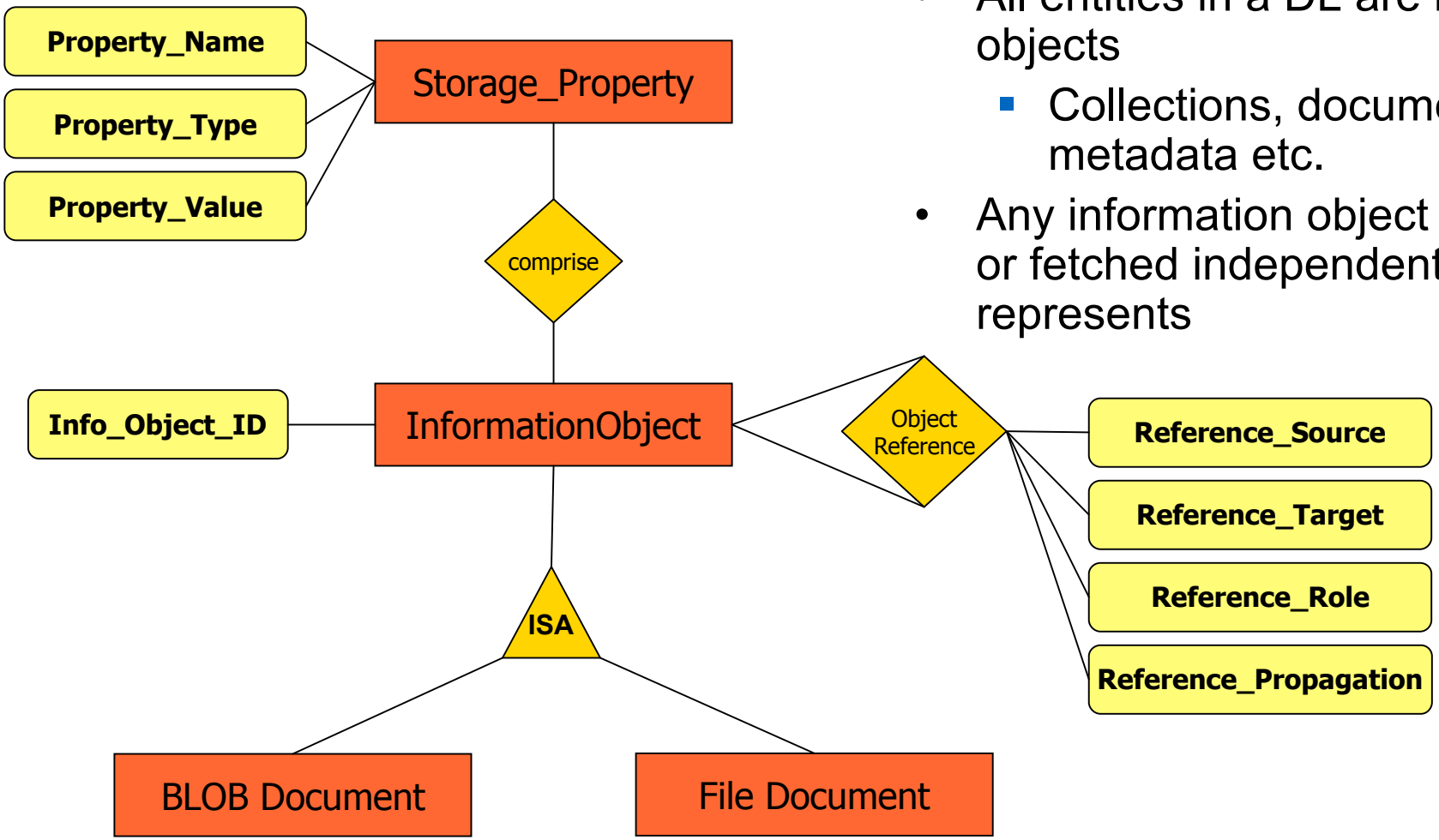
Feature Extraction

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>
```

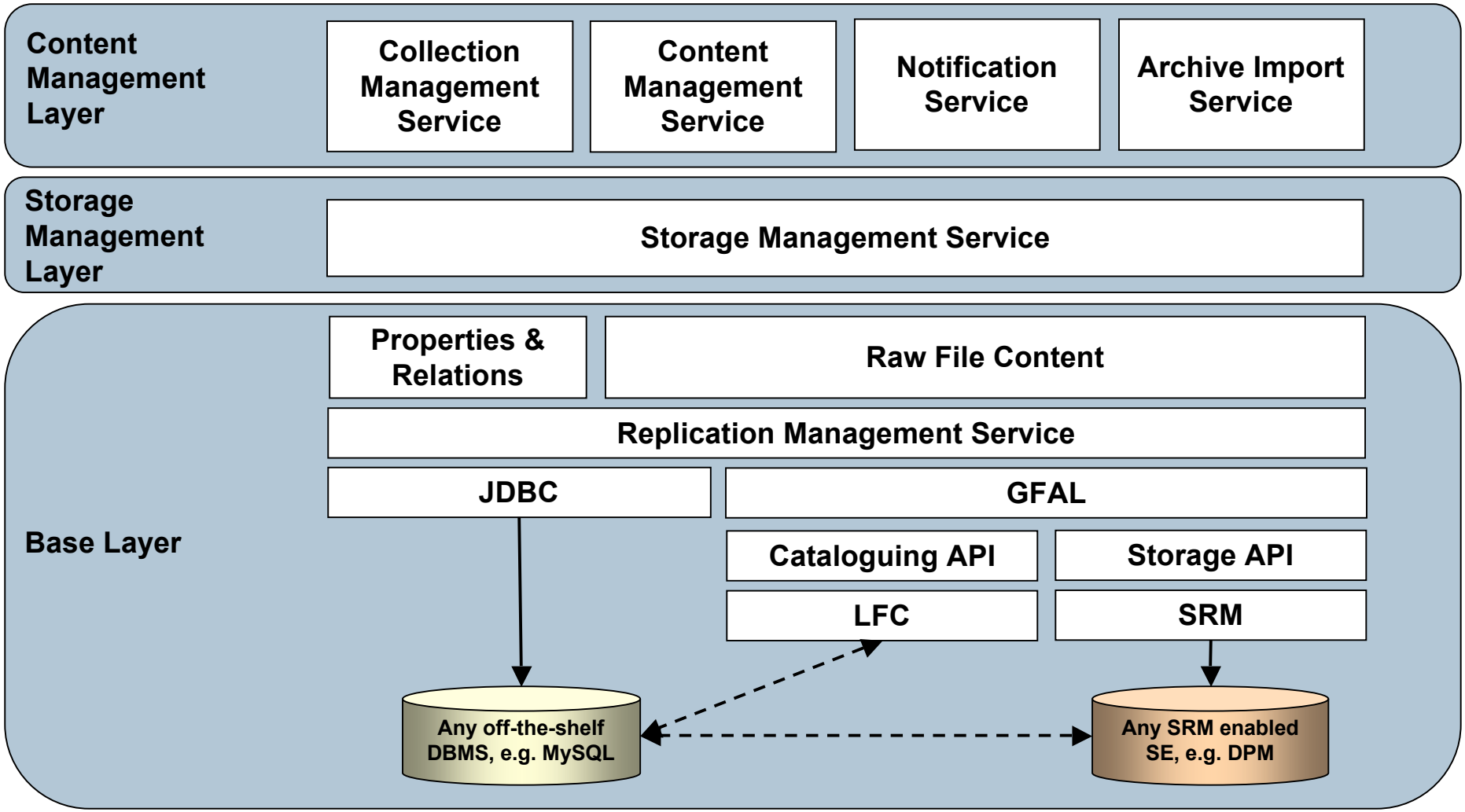
Metadata Management

Document Model



- All entities in a DL are information objects
  - Collections, documents, metadata etc.
- Any information object can be stored or fetched independently of what it represents

Layered Data Management Architecture



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.

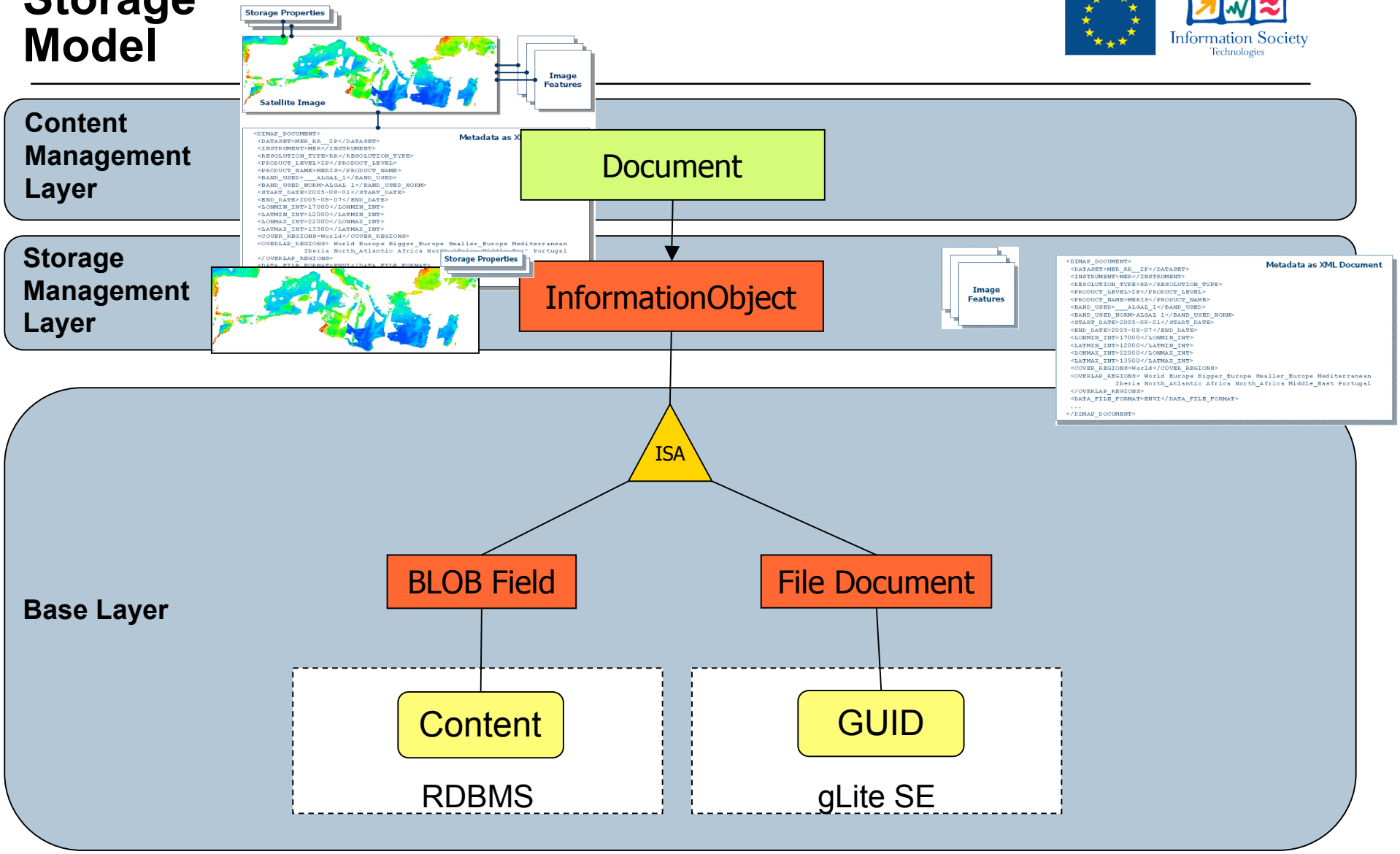
GFAL: **G**rid **F**ile **A**ccess **L**ibrary

- 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

Storage Model



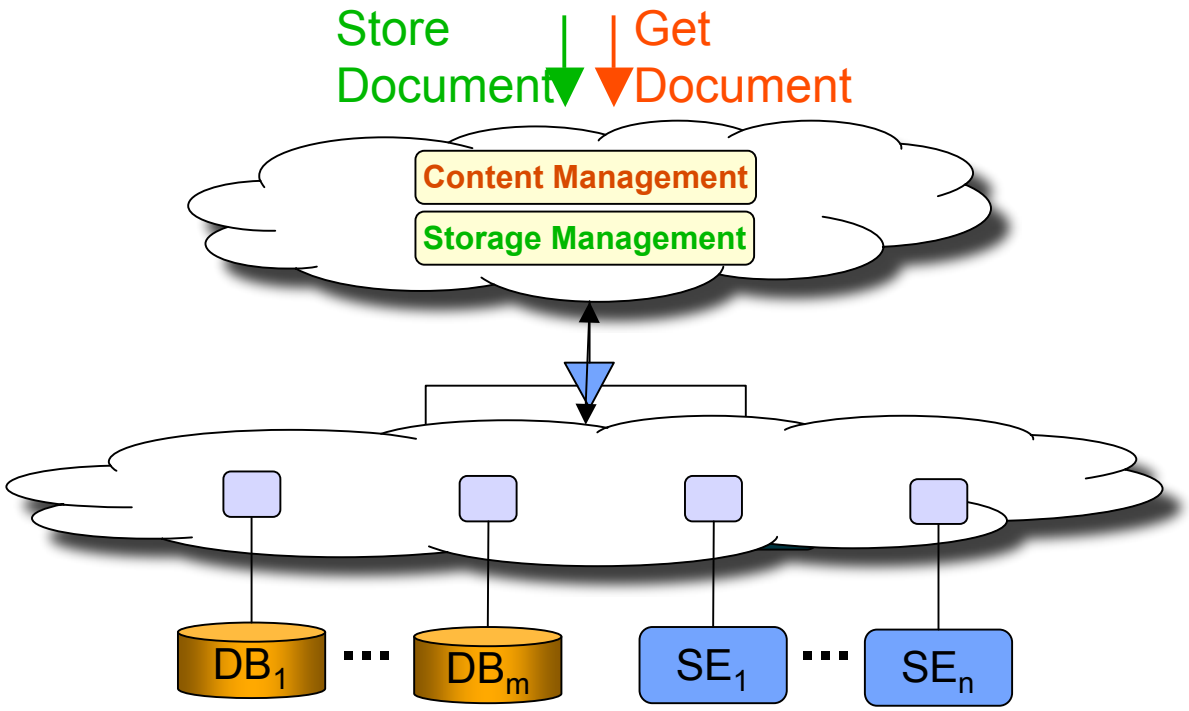
Where to Store What?

An information object can be stored either in the RDBMS or in the Storage Element

- RDBMS is a natural choice to store properties of documents and relations between documents
- Decision about storage (database or Grid storage) for raw content can be based on heuristics, e.g.,
  - Store files always in a storage element (SE)
  - Store files larger than 2 MB in SE

Replication Management Service

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



Content Management: Replication …

Replication

- Essential technique to improve availability at data level
- Fully or partially duplicates data objects (e.g., documents) 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
- Usually distinction between master site (original –and updateable– copy) and slave sites (replica)
- Basic approaches for maintaining replicas : eager vs. lazy
  - ž Eager replication: synchronizes replicas within the boundaries of the update transaction
  - ž Lazy replication: decouples synchronization from the updating transaction (replication done in separate transaction)

… Content Management: Replication

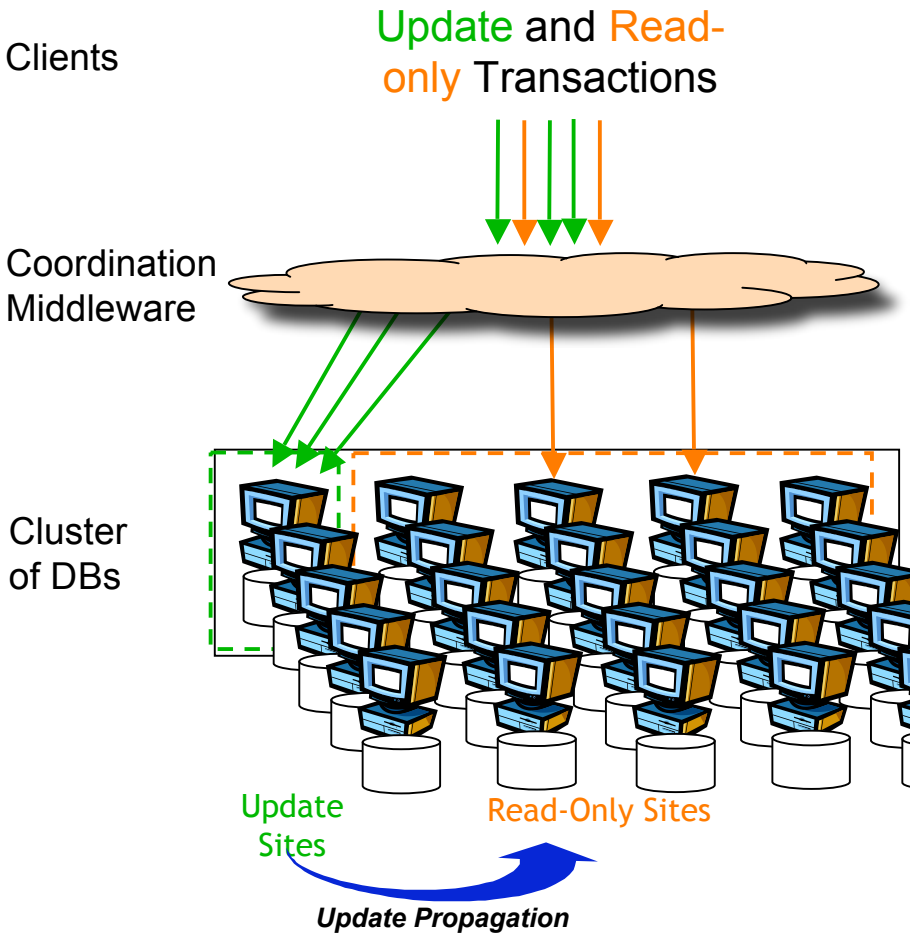
Replication support in the Grid is currently still underdeveloped

- Just creating copies of documents without sophisticated maintenance

Goal in DILIGENT: Apply a sophisticated replication protocol designed for database clusters to the Grid

- 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
- Adjust to changing load by dynamically creating new replicas

Example: Replication in a Cluster of Databases



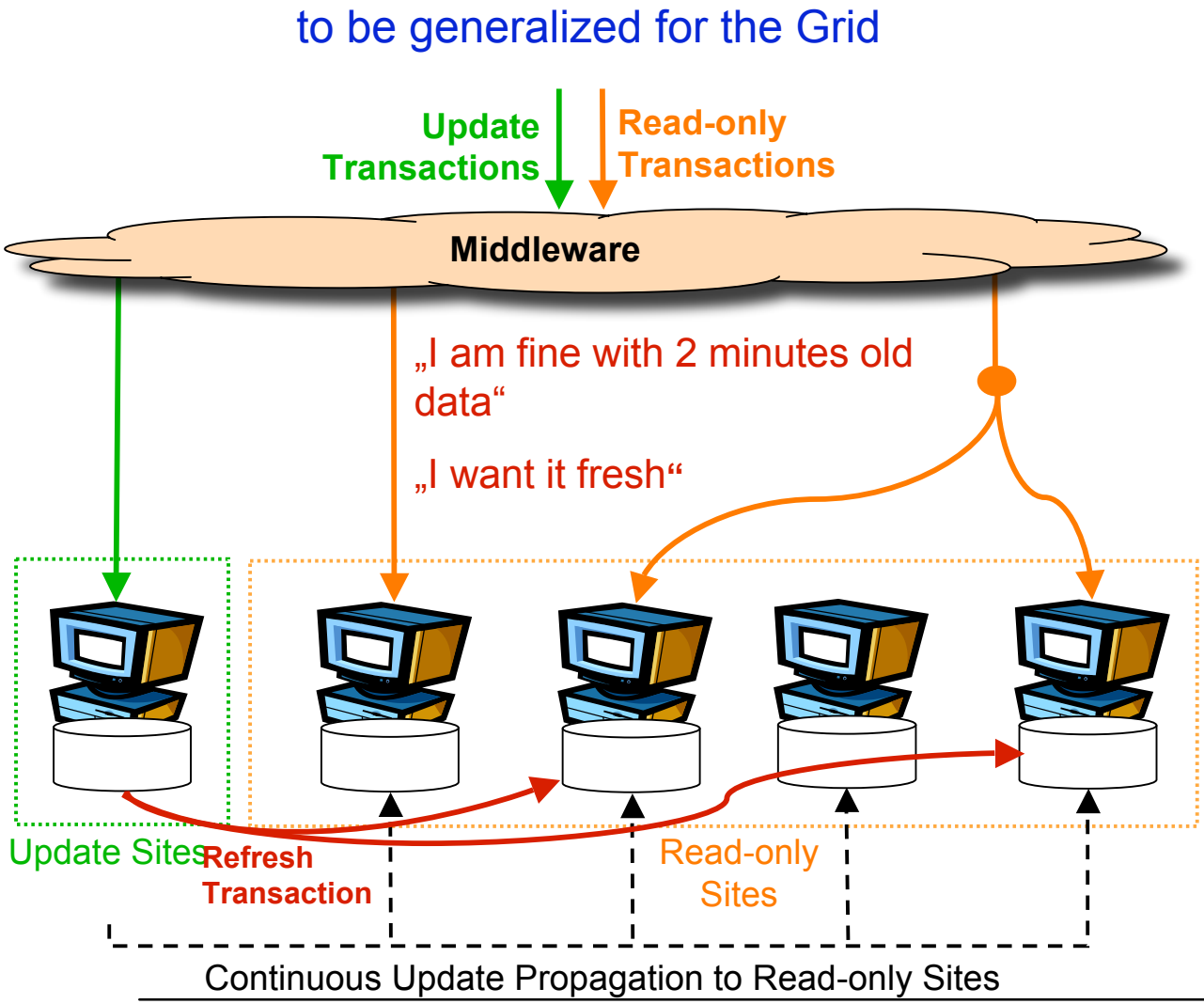
Cluster of databases

- Network of off-the-shelf PCs
- Each running a commercially available RDBMS

Middleware Architecture

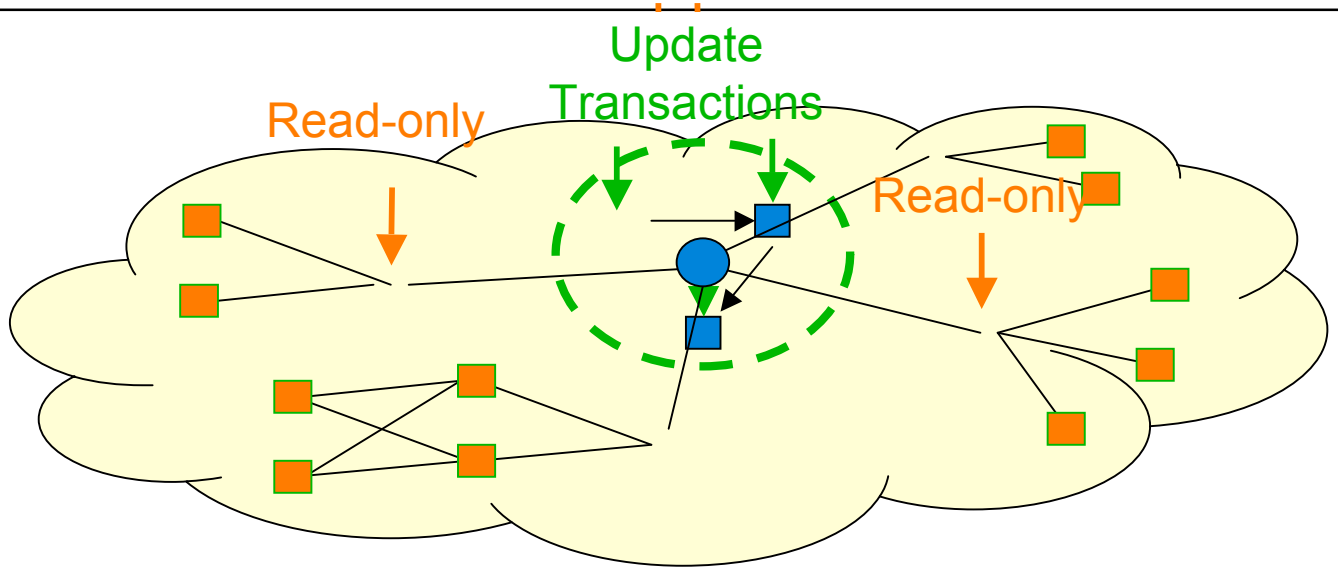
- Clients access the cluster over the middleware only
  - ž Two disjoint parts (update / read-only)
  - ž Updates occur at the update sites first, then propagated to the read-only sites
  - ž Replication is internally lazy, although it is eager from user's perspective
- The "Scale-out" vision
  - ž Adding new nodes for to increase parallelism and performance

A Sophisticated Approach to Replication in a Database Cluster



- Read-only transactions may span over many nodes providing intra-query parallelism
- Users may specify their freshness requirements
- Read-only sites are kept as up-to-date as possible via continuous update propagation transactions
- On-demand refresh transactions can be used to bring data up to the required freshness level
  - Otherwise, this will eventually be achieved by propagation transactions
- Consistent reads ensured by disallowing refresh/propagation transactions to change the version of data to be read

Advanced Replication in the Grid



Update and read-only storage nodes distributed in the Grid

A query can be served by any node while changes can only be sent to an update node

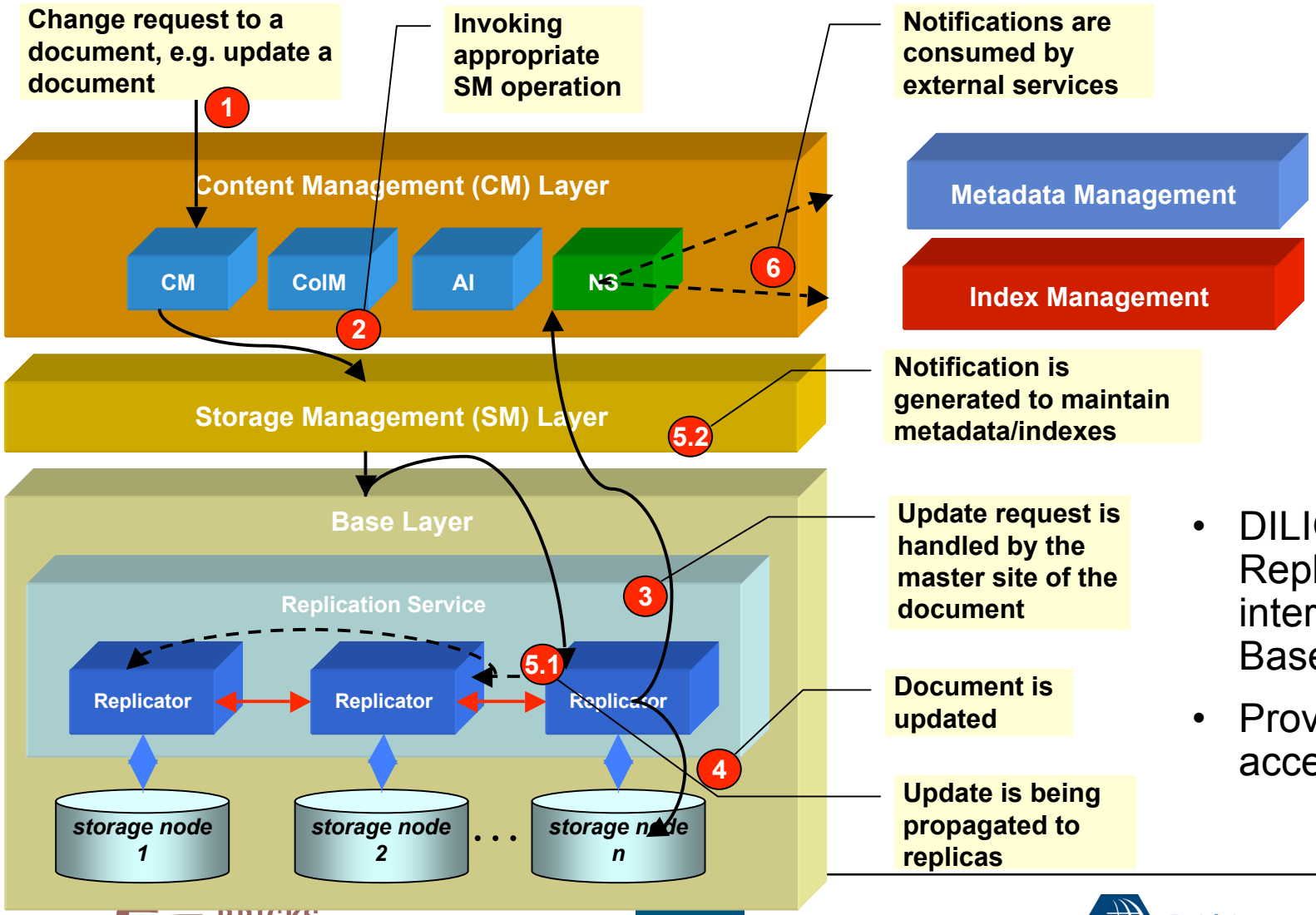
Multi-master lazy replication (many update nodes)

- Serialization and propagation of updates is an issue
- Update and read-only nodes at the level of specific data sets (e.g. a collection or partitions of it)

Specific features for replication in the Grid

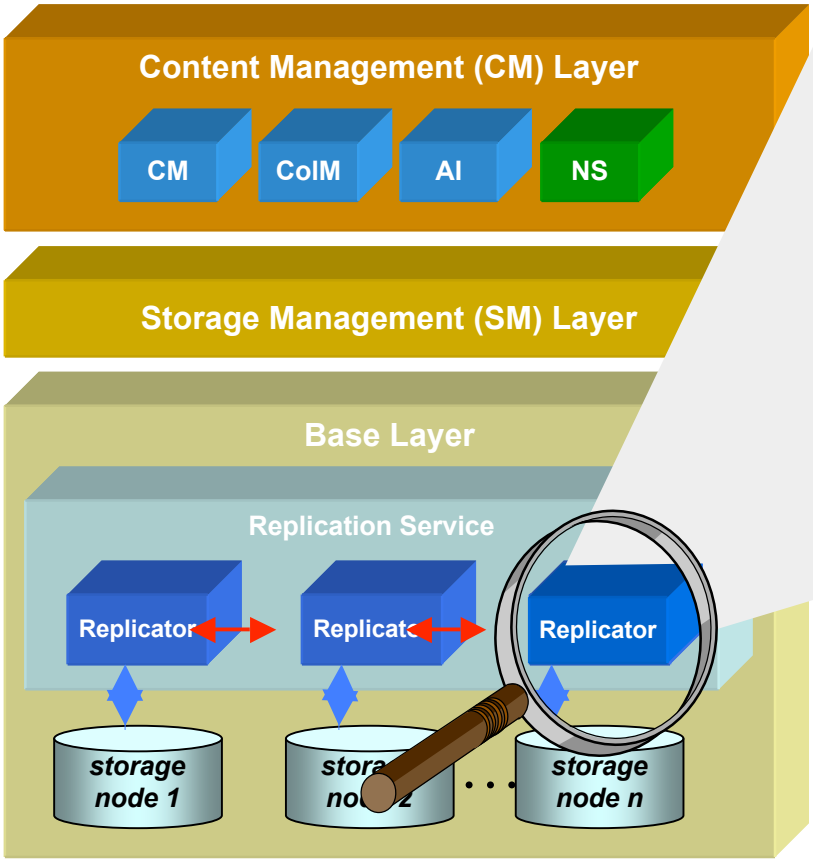
- Broadcast of updates not feasible, replicas subscribe for changes instead
- More nodes which are heterogeneous
- **Failures** are more likely to happen

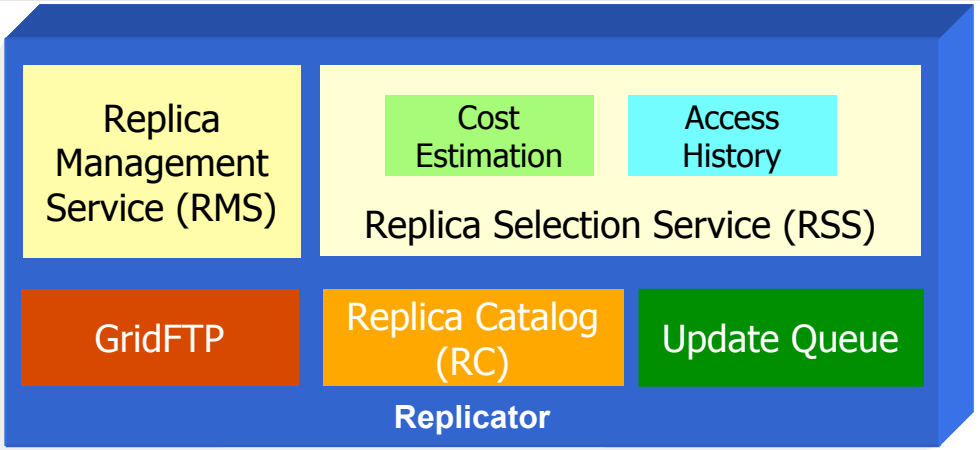
Content Management: Replication …



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

… Content Management: Replication





RMS allows to maintain RC and to manage replicas, e.g., create, delete, list replicas etc.

RSS

- find the appropriate replica, e.g., *bestReplica* via its cost estimation module
- maintains access history for dynamic replica allocation decisions (e.g., replicating data which is frequently accessed)

GridFTP protocol for moving data around the Grid, e.g. FTS.

RC to identify the locations of files

Update queues at each storage node to collect updates from master nodes.

Lessons Learned

- In terms of content management, DLs are much more demanding than ‘traditional’ Grid applications like high energy physics or astronomy
- Basic tools and protocols are in place but they need to be extended 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

The Goals

Goals the DILIGENT Search Engine

- Operate on the OGSA environment
- Consume physical and virtual, standards-compliant resources
- Provide standards-compliant resources for building higher level services
- Provide means for extending and customizing base implementation
- Overcome performance issues Service-Composition

Foreseeing

- The emergence of diverse, autonomous, pluggable elements
- The ability to compose complex information and data processing environments
- The maximization of the resources placed at the disposal of DL managers and users
- The reduction of cost of ownership and use
- The creation of a “field” for further experimentation with IR technologies

The Outcome

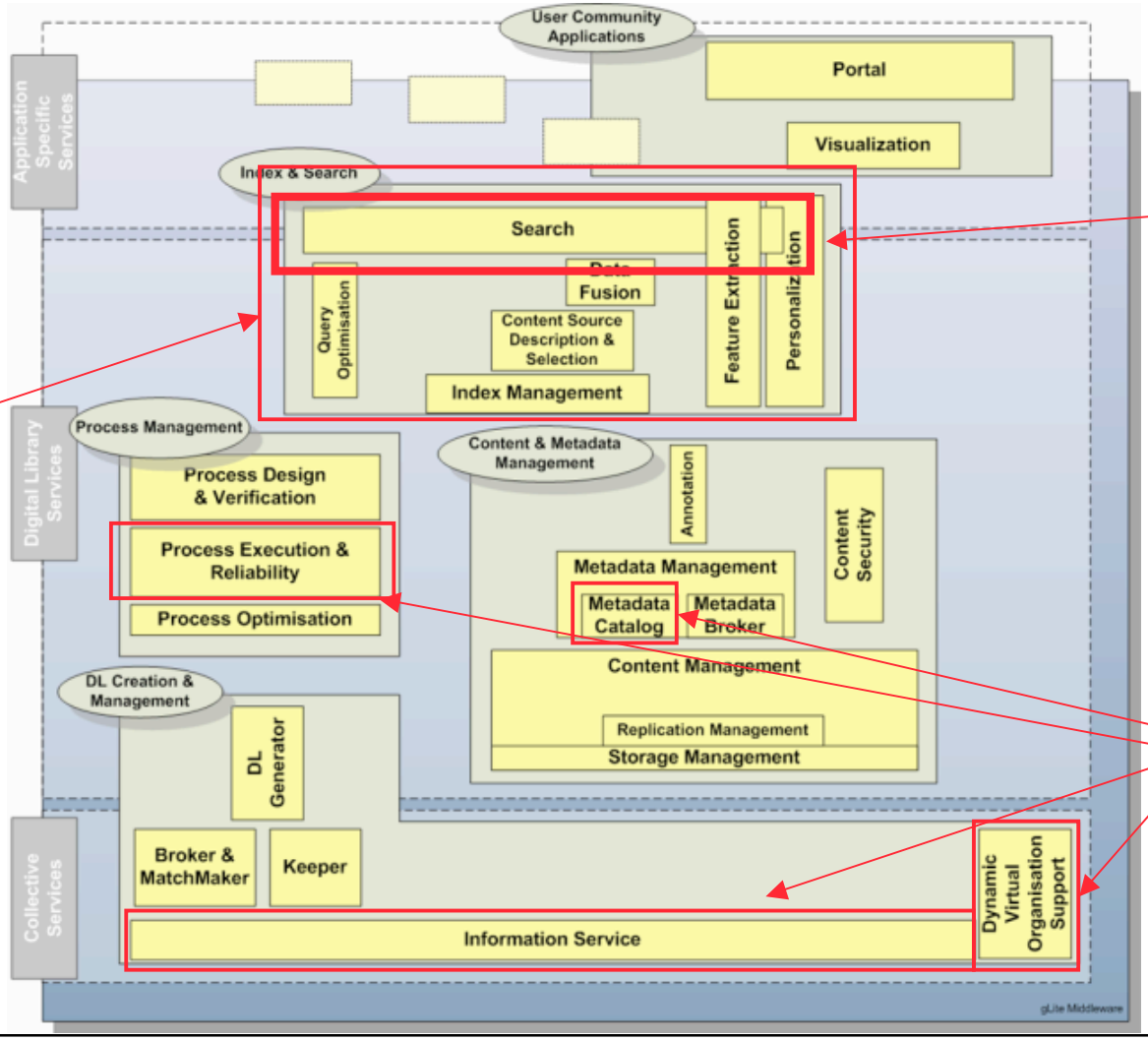
A Query Language
- Proprietary - Exposes the openness of the system

A Search Service
- The orchestrator of Information Retrieval
- Consolidates query and environment information
- Prepares and plans retrieval execution

Numerous “worker” services
- XML processing (joiners, sorters, transformers, filterers)
- Lookups (FT indices, XML indices, Geo indices, external sources etc)
- Fusion / Merging of results

The ResultSet, a data transport mechanism to
- Overcome the paging/streaming limitations of WS
- Provide flow control
- Overcome implementation limitations of containers
- Exploit typical grid mechanisms

The Search Pipeline

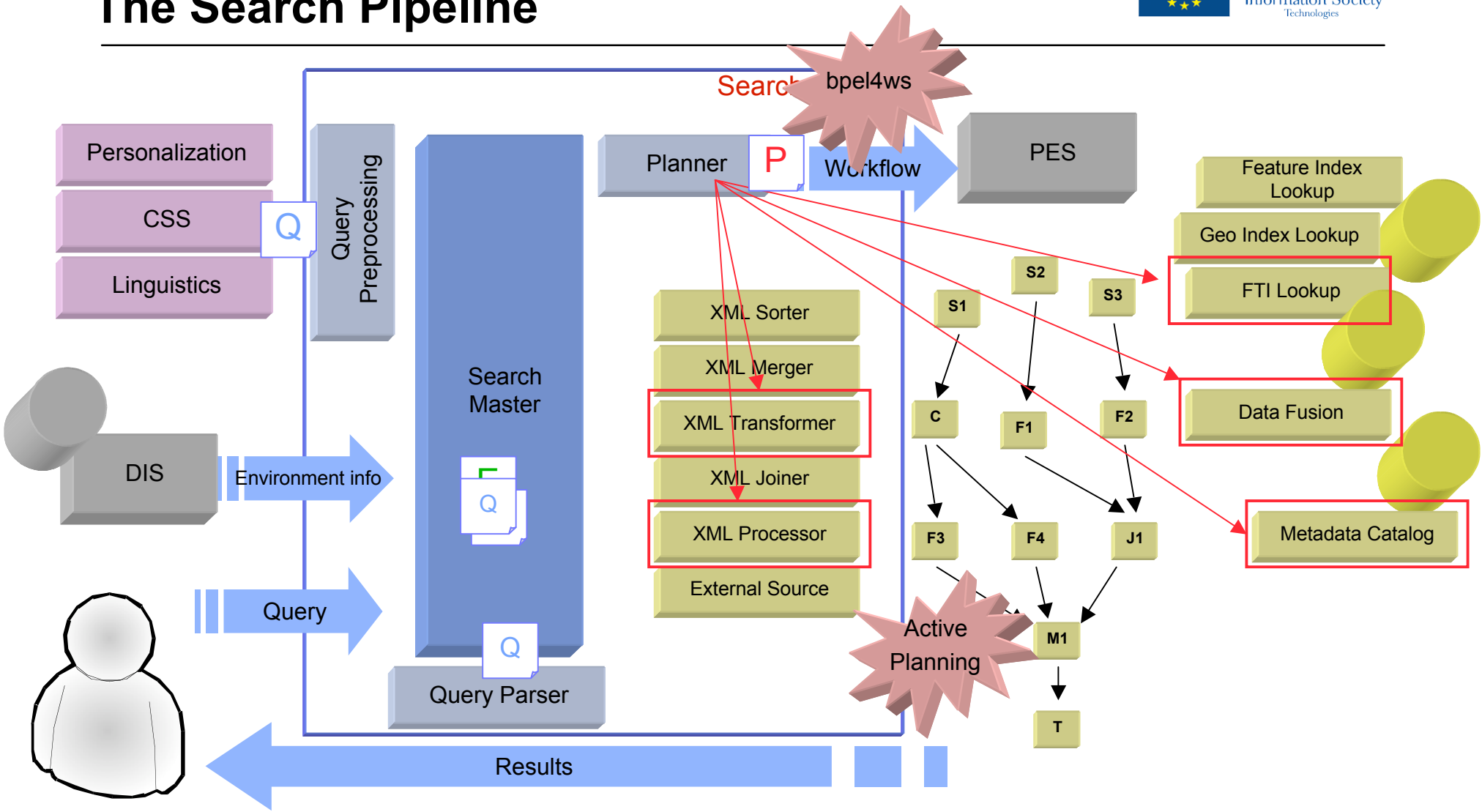


Search Service

Logically co-grouped with, and directly depends on…

Direct dependencies

The Search Pipeline



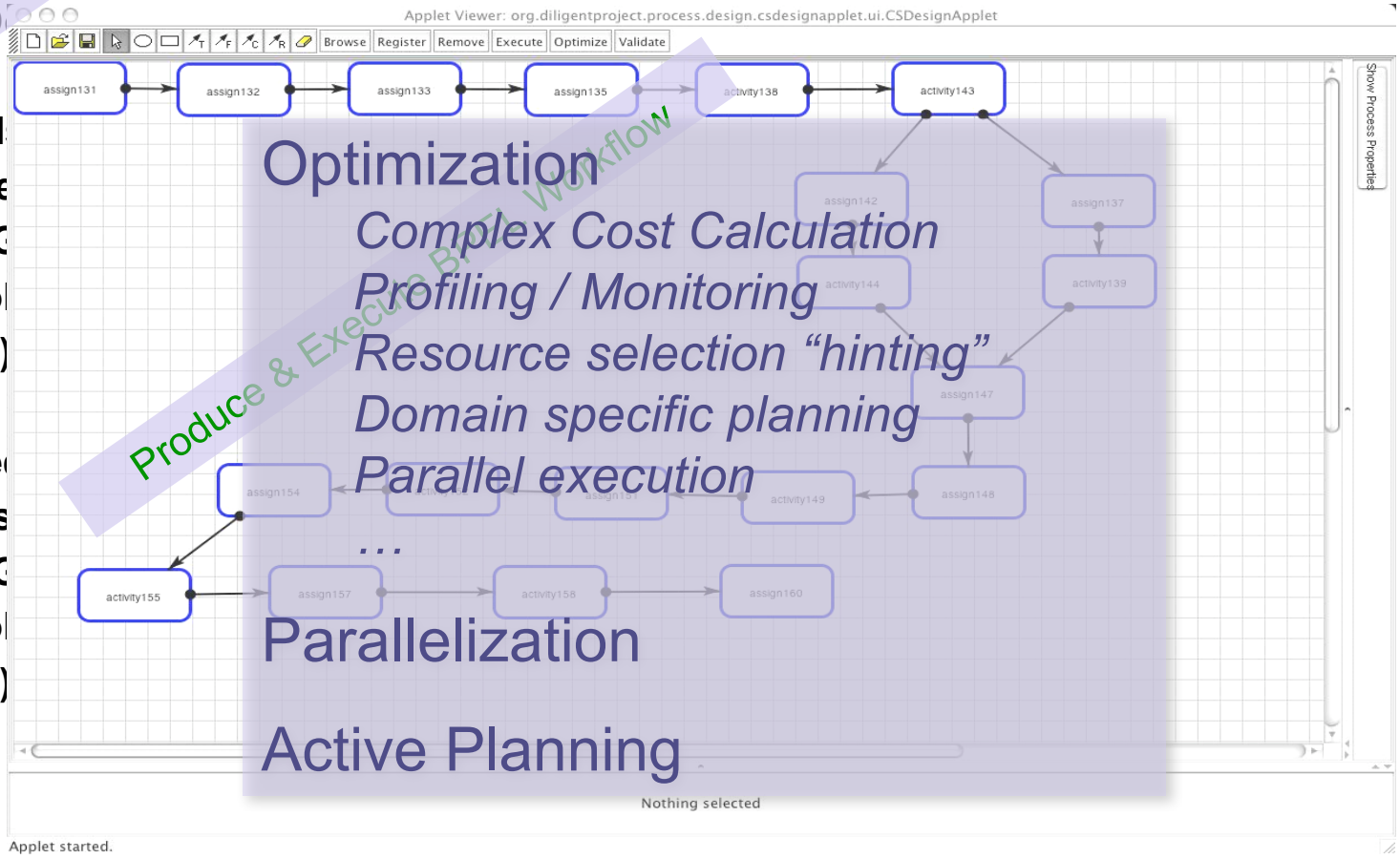
Search behind the scenes: Always executing workflows

```
project by 'title', 'description', 'subject'
    on (keeptop 20
        on (sort ASC by 'D
            on (merge
                on (fielded
                    by 'title
                    in 'ENG
                    on 'Col
                    as 'dc')

                and (fielde
                    by 'des
                    in 'ENG
                    on 'Col
                    as 'dc')
            )
        )
    )
```

Query

Produce & Execute BPEL Workflow



Optimization

Complex Cost Calculation
Profiling / Monitoring
Resource selection "hinting"
Domain specific planning
Parallel execution
...

Parallelization

Active Planning

Search behind the scenes: It can get complex…

```
project by 'title', 'date' on
   (sort ASC by 'DocID' on
      (merge on
         //MAP REPORTS
         keeptop 8 on
            (sort ASC by 'RankID' on
               (join inner by 'DocID' on
                  (fulltextsearch by 'Mediterranean' in 'ENGLISH' on 'd369b3e0-fa4c-11db-a297-9c01d805f283')
               and
                  (fulltextsearch by 'Environmental' in 'ENGLISH' on 'd369b3e0-fa4c-11db-a297-9c01d805f283')))

         keeptop 8 on (sort ASC by 'RankID' on (join inner by 'DocID' on (fulltextsearch by 'Mediterranean' in 'ENGLISH'
on 'd369b3e0-fa4c-11db-a297-9c01d805f283') and (fulltextsearch by 'Environmental' in 'ENGLISH' on 'd369b3e0-fa4c-
11db-a297-9c01d805f283')))
         // EEA reports
         keeptop 8 on
            (sort ASC by 'RankID' on
               (fieldedsearch by 'date' contains '*1999*' on
                  (join inner by 'DocID' on
                     (fulltextsearch by 'air polution' in 'ENGLISH' on '25ad3c50-fa41-11db-a270-9c01d805f283')
                  and
                     (fulltextsearch by 'european' in 'ENGLISH' on '25ad3c50-fa41-11db-a270-9c01d805f283')
                  )
               )
            )
      )
   )
```

Search Operators: A Closer Look

- Sorting / Filtering results
  - Index lookups
  - Metadata processing (xpath/xquery)
  - Row-by-row evaluator
- Joining results
  - Joining metadata from various sources.
  - Inner / Outer joins
- Merging results
  - Plain unions.
  - Data Fusion to merge ranked lists.
- Querying External Sources
  - Google, JDBC data sources, ISIS/OSIRIS system etc.
- Aggregation functions
  - Calculate values over entire resultsets.
  - Necessary for conditional execution.
- Transformations
  - Powered by XSL/XSLT.
- Invocation of custom processors.

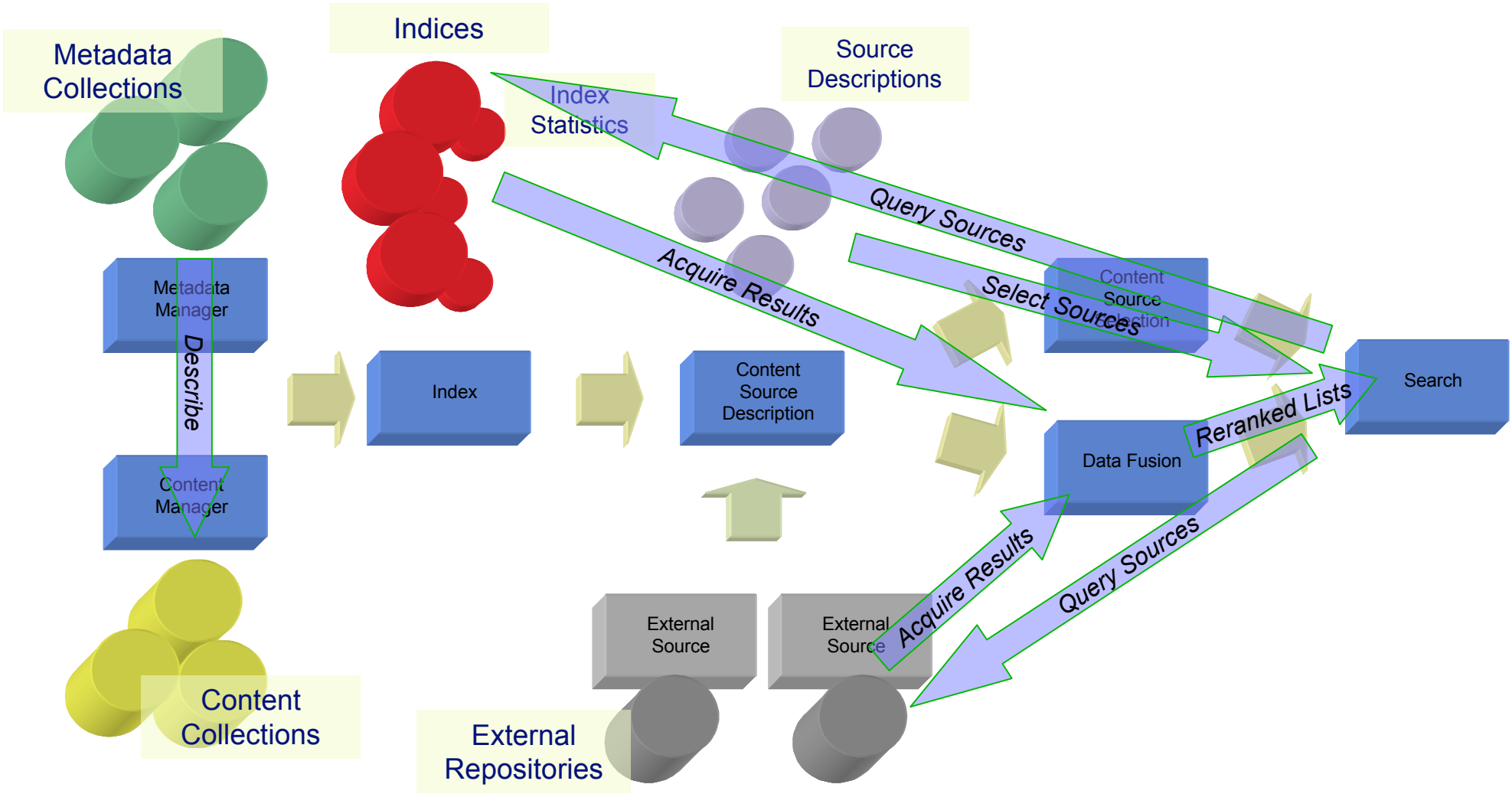
Search operators are:

- typical Web Services or Web Service Resources
- described n DIS via appropriate Profile
- forced to comply with the ResultSet data exchange mechanism Search-specific usage

Levels and techniques of optimization in Search

- Pre-query optimisation:
  - Keeper service monitors and adapts the DL layout for optimal resource usage.
- Content Source Selection:
  - Filters out collections unlikely to be contain what the user is looking
  - Uses query supplied terms and automatically pre-constructed Content Source Descriptors
- Query Planning:
  - Cost based optimisation performed.
  - Heuristics and space-search
- Process Execution:
  - Process optimisation service selects and allocates the appropriate resource to carry out a task.
- On-The-Spot processing:
  - ResultSet mechanism to allow local filtering of large XML chunks of data.
- Further mechanisms to facilitate efficient searches:
  - Indices.
  - ResultSet transport mechanism to bypass WS-* shortcomings and facilitate paged data exhanges.

Distributed Information Retrieval Components



Major Search Service Dependencies

Indices

- Serve queries via proprietary languages
- Can manage linguistic issues
- Types
  - ž Forward :
    - B-Trees (field lookups).
    - R-Trees (geo-spatial search).
    - High-dimensional VA files (content based search).
  - ž Inverted:
    - Full Text Indices

Metadata Catalog

- (Serve the Metadata)
- Serves direct XML queries on Metadata

Distributed Information Retrieval Components

- Describe and select Content Sources
- Fuse results stemming from different sources

Interoperability of Information in DILIGENT

Powered by the infrastructure

- Multiple schema hosting Metadata Management Service
- Powerful schema transforming engine: The Metadata Broker

Supported by Search

- Selection of common schemas for cross-collection semistructured searches
- Support for on-the-fly projection

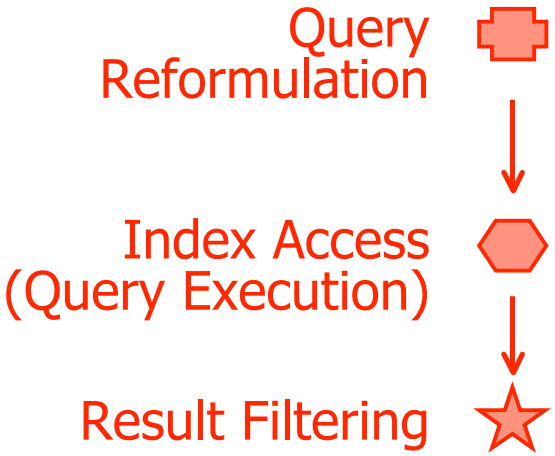
Assisted by the user interface

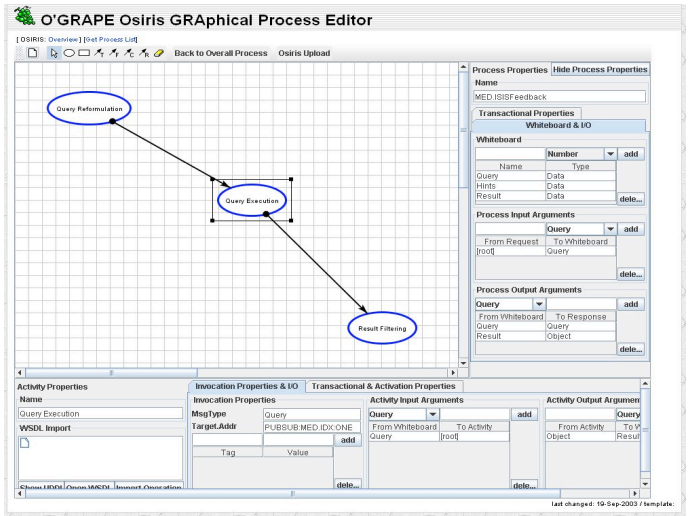
- Composing queries to exploit search capabilities
- Exploits admin's or end-user's knowledge on hosted information structure

Questions

Short Recap:
Distributed Digital Library Applications

- Digital libraries are large-scale collections of digital contents
- Digital content is stored and maintained in specialized applications and systems and access to this content is provided by dedicated services
- Applications in Digital Libraries require to integrate these distributed services (and contents of different providers) into a coherent whole
  = „Programming in the Large“ (composition of services into processes)





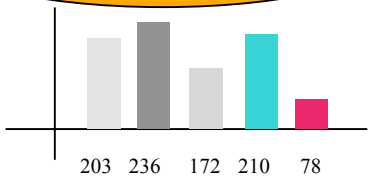
Process Management: Example Search

Similarity search over multimedia documents

Service

1. Query

2. Extract Features

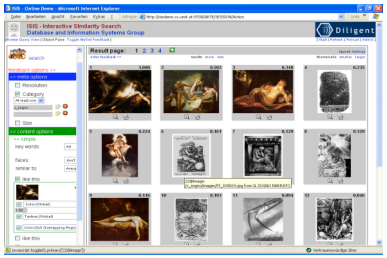


3. Query Index

Index

4. Access Content & Create Result Set

5. Present Results



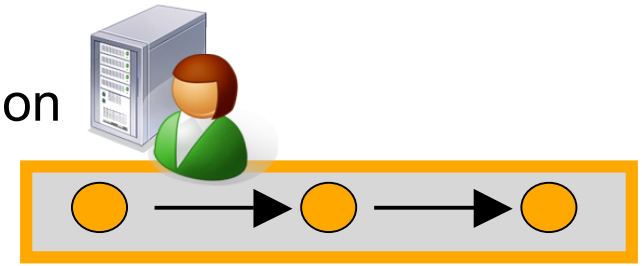
Process

Processes in DILIGENT

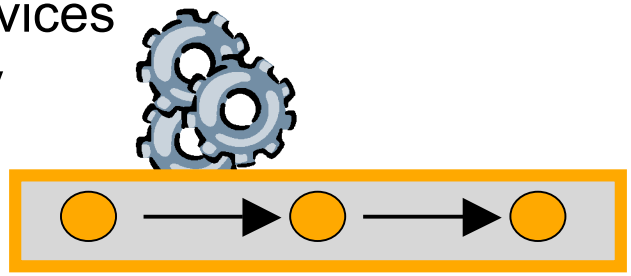
- Following the SOA paradigm, processes are the first choice in DILIGENT to define and execute applications on the basis of available services
- DILIGENT's approach to Process Management on the Grid consists of three main services
  - Process Design and Verification
    - Provide a graphical user interface for specification and analysis of processes
  - Process Execution and Reliability
    - Distribute process support in the Grid (avoid centralized workflow engine!)
    - Dynamic allocation of resources
    - Sophisticated failure handling
  - Process Optimization
    - Structural process modifications to maximise parallelism

Application vs. System Processes

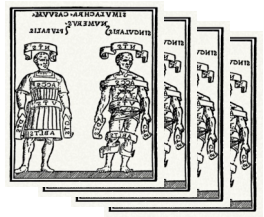
- In DILIGENT, there are two classes of processes
- Application processes
  - Defined by the user (or at least according to the requirements of a user community)
  - Provides functionality for a particular application
  - Invoked by users

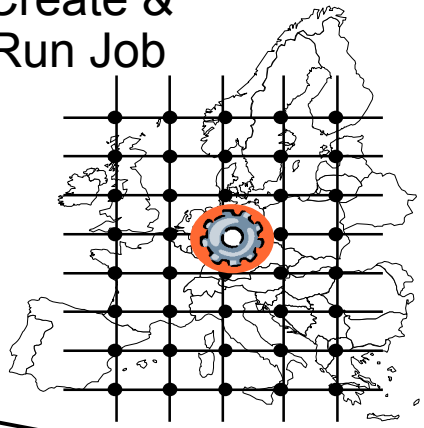


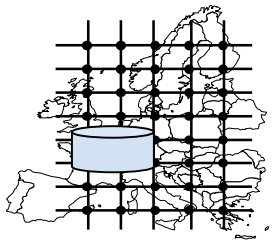
- System processes
  - Defined by the DL Administrator
  - Provides functionality needed by the DILIGENT system
  - Automatically invoked by other DILIGENT services
    - E.g., for administering a DL in flexible way
    - Usually invisible to the end users

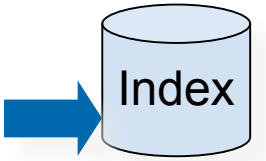


Sample System Process: Make new content available









Content Management | Feature Extraction | Index

1. Insertion of New objects. Create object
2. Create & Run Job
3. Store & associate metadata
4. Update Index

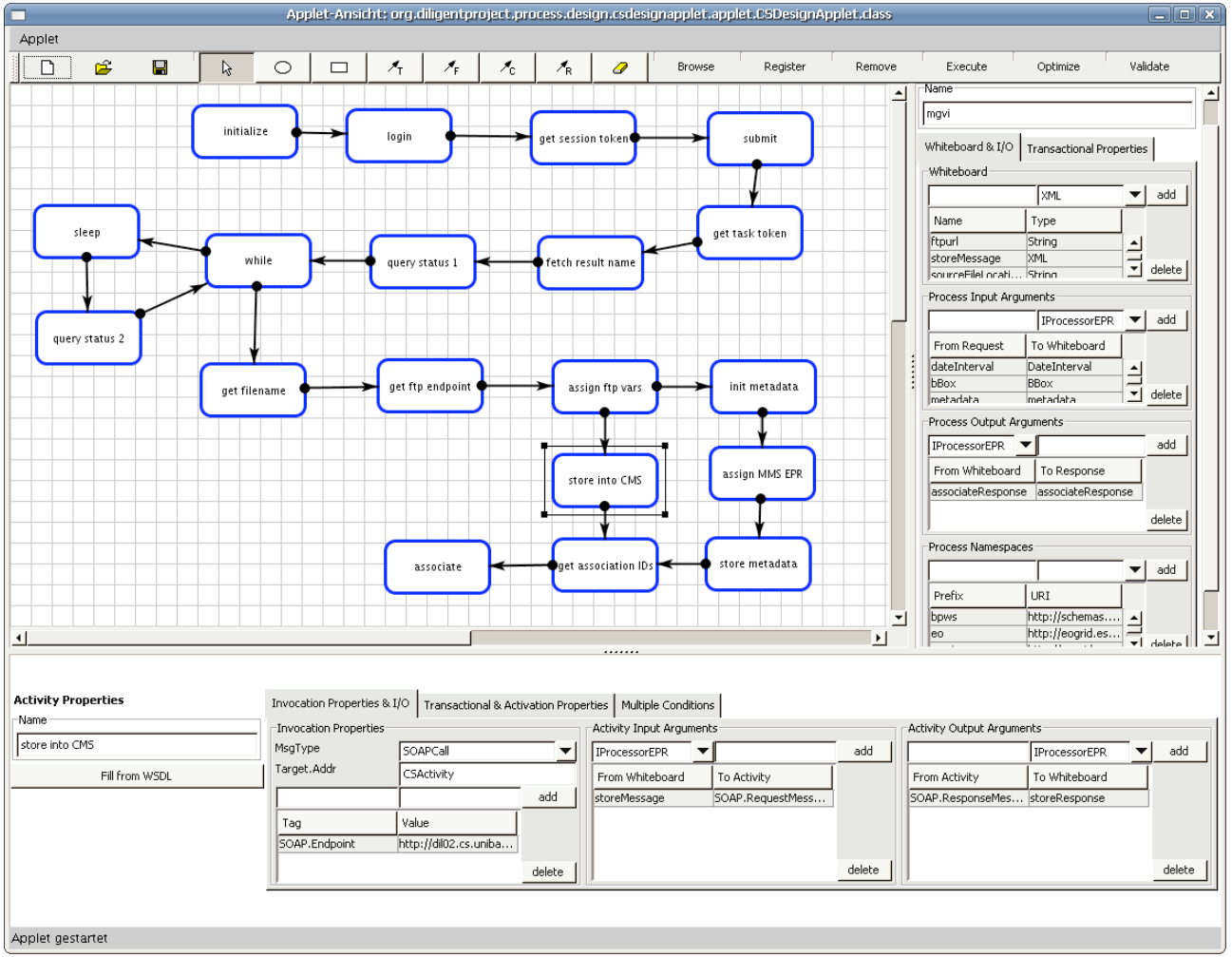
Advantages of using System Processes and the Process Execution Service

- Has built-in failure handling
- Process can easily be extended/adapted without need to alter underlying services

Sample Application Process: Meris Global Vegetation Index (MGVI) …

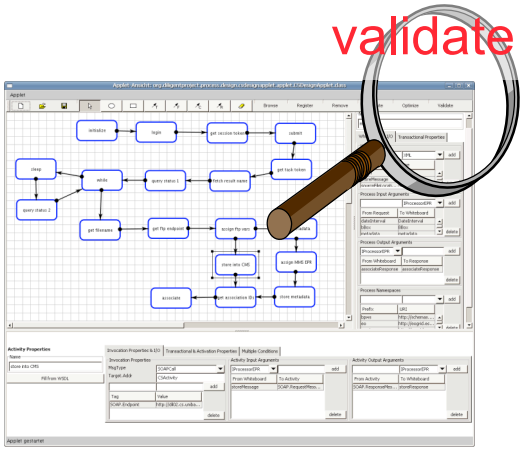
- MGVI provides information about vegetation at specified geographical region during specified timeframe
- Basis: Grid job provided by the European Space Agency (ESA); accessible via Web Service interface
- Seamless integration of (external) ESA web service and DILIGENT services, all invoked in a single process
- Process Steps
  - Parameters: Bounding box, datetime interval, (auto-generated) metadata
  - Log in to EOGrid
  - Submit parameterized job
  - Periodically poll until job finished (can take up to 15 mins)
  - Job stores output on intermediate FTP server
  - Tell Content Management to fetch data
  - Associate metadata with content and store in Metadata Repository

… Sample Application Process: Meris Global Vegetation Index (MGVI)



Process Design and Verification

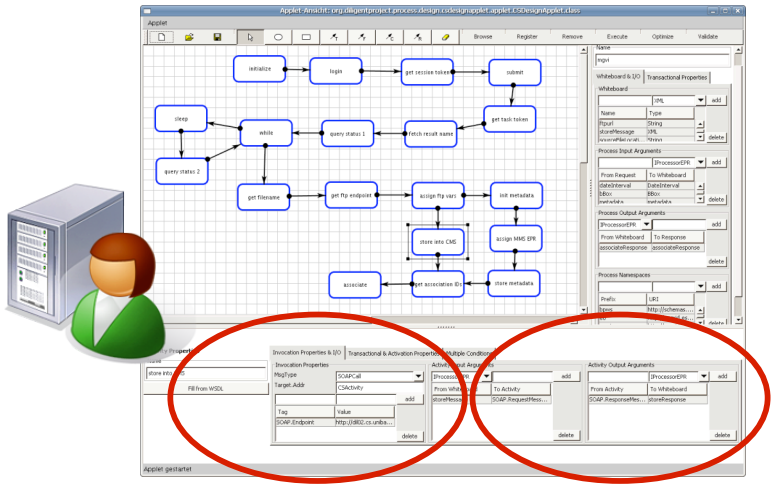
- Unlike conventional programming from scratch, the building blocks of processes are already in place and need to be combined appropriately by defining control and data flow dependencies
- Appropriate graphical tools for process design and specification need to be provided
- Failure situations have to be anticipated and appropriate failure handling strategies have to be added to the process specification
- Verification algorithms have to be integrated into graphical process design tools in order to be able to formally prove the correctness of processes already at build-time
- Integrate quality-of-service (QoS) aspects in the verification of processes and to derive provable qualitative prognoses on the execution of processes



Process Definition

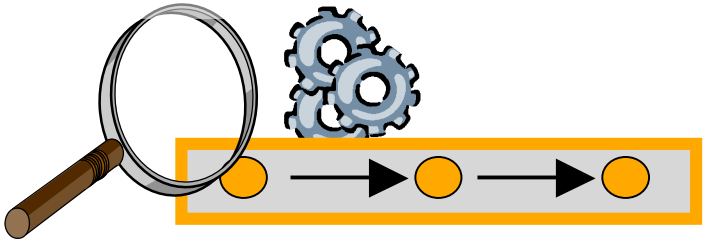
DILIGENT supports two different approaches to process design

- New process manually created via **`CSDesign`** applet
  - List of available services retrieved automatically from the DIS
  - Service parameters available for data flow specification



- Process is automatically created by another DILIGENT service (e.g., Search)
  - Validated by **`CSFinalizer`**

CSFinalizer

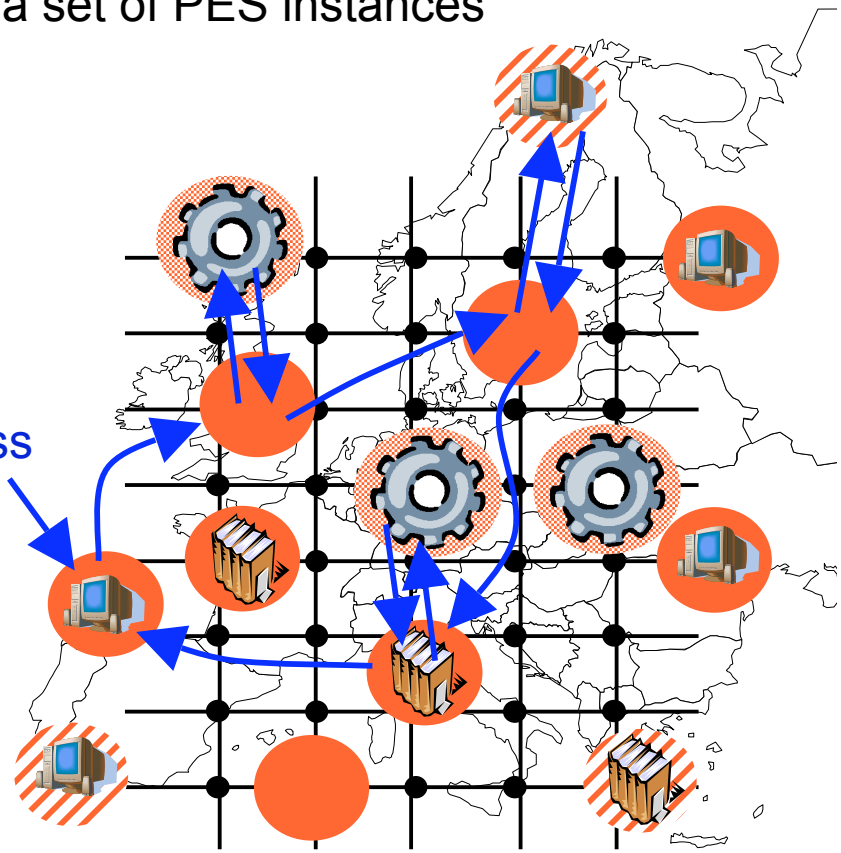


Process Execution and Reliability

- A major requirement for DL applications and the infrastructure running (process-based) applications is that the correct execution –according to the process specification– can be enforced at run-time
- Reliability means that process execution should not be affected even in case of failures or changes in the environment
- Rather, sophisticated routing of service requests within a service grid is required in order to equally balance the load within the services of the grid and to ensure high response times for processes

Process Execution

- Process execution is realized in a distributed way by dedicated DILIGENT services: PES (process execution service)
- Process execution is shared dynamically across a set of PES instances
- Services that can be invoked
  - DILIGENT services
  - gLite jobs (via generic gLite Job wrapper)
  - External Web services



DILIGENT service (with PES locally deployed)

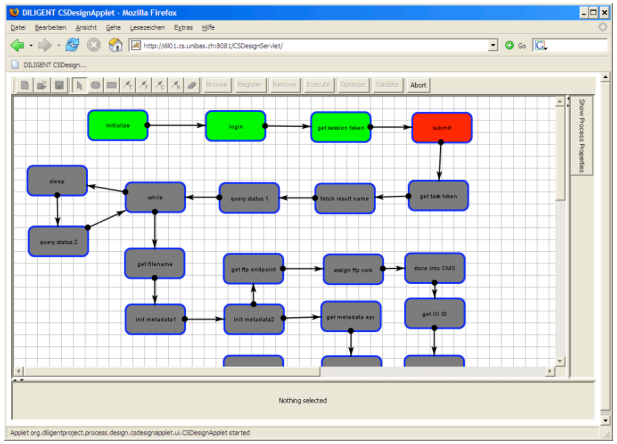
DILIGENT service (without local PES)

gLite job

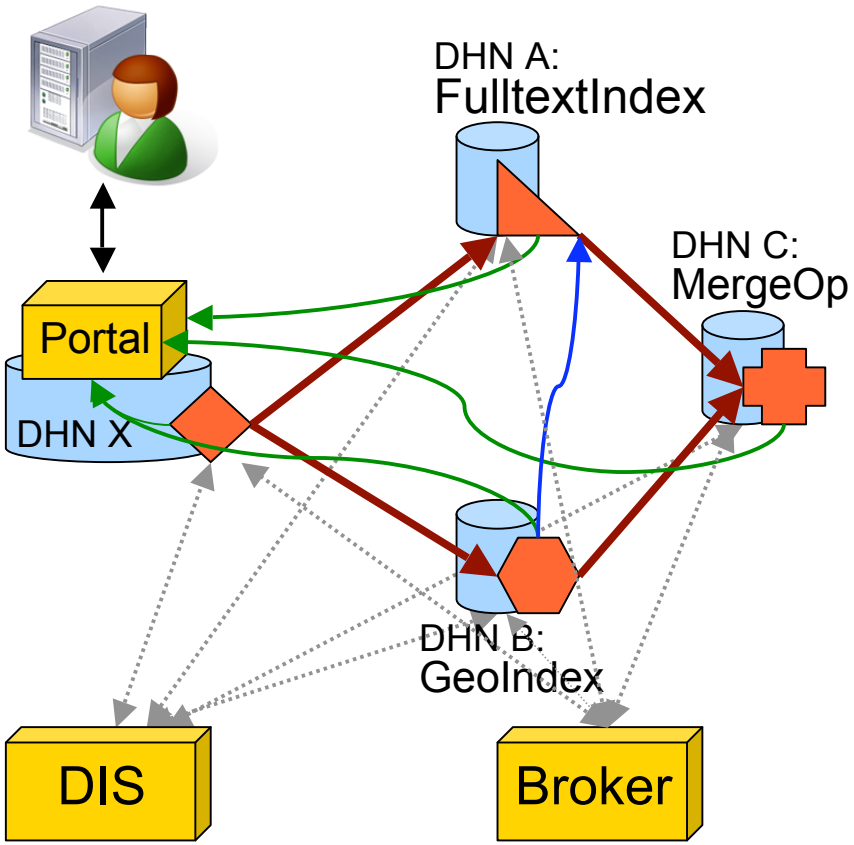
gLite job wrapper

Process Monitoring

- Distributed and reliable Process Management is one of the main features of the DILIGENT system
  - But it is basically system functionality that cannot be seen directly
  - A monitoring interface has been added that allows the DL Administrator to keep track of running process instances
  - Based on the Process Design applet, state (= active services) is graphically highlighted
  - Uses notifications to get information on state changes (monitoring completely distributed processes is far from being a trivial task)



Process Execution: Overview



- **`CSEngine`** runs on every DHN
- Process orchestration distributed in the infrastructure
- Service deployment information from DIS
- Process definitions possibly from DIS
- Notifications (all brokered):
  - Process status information (monitoring)
  - Join coordination

Process flow

Join notifications (crucial for execution)

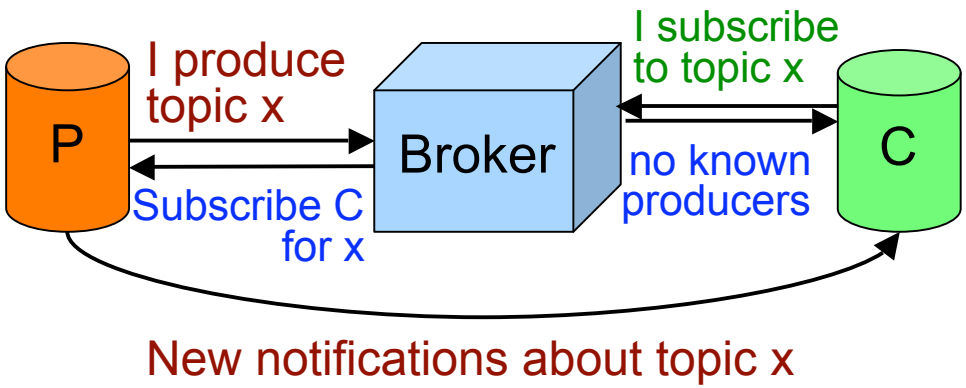
status notifications („optional“)

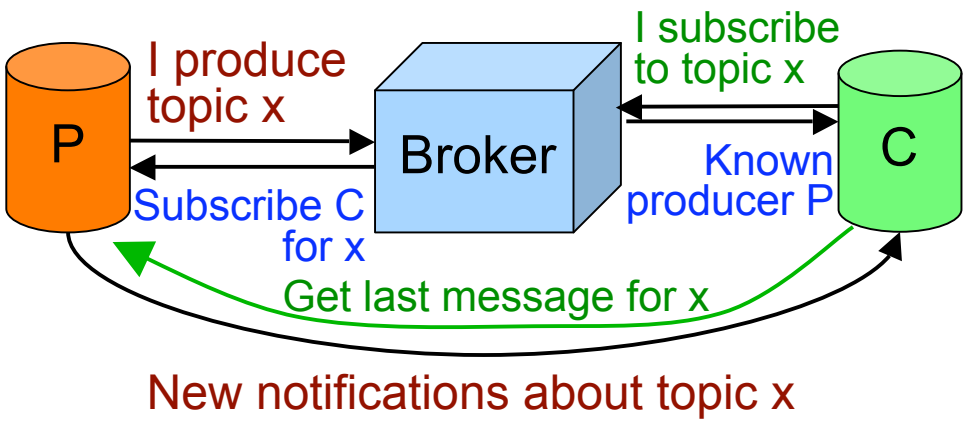
Collective Layer interactions

Brokered notifications

Brokered notifications – extending WS-BaseNotifications; rationale:

- Subscribers do not know where topic is produced
- Producers do not know where subscribers are
- E.g. join notifications: location of all execution nodes is dynamic in the system, have to agree on common join node
- In addition, topic might be „made available“ before any subscribers present – need to keep the info available
- NB: the actual notifications are „p2p“, only the subscription is brokered

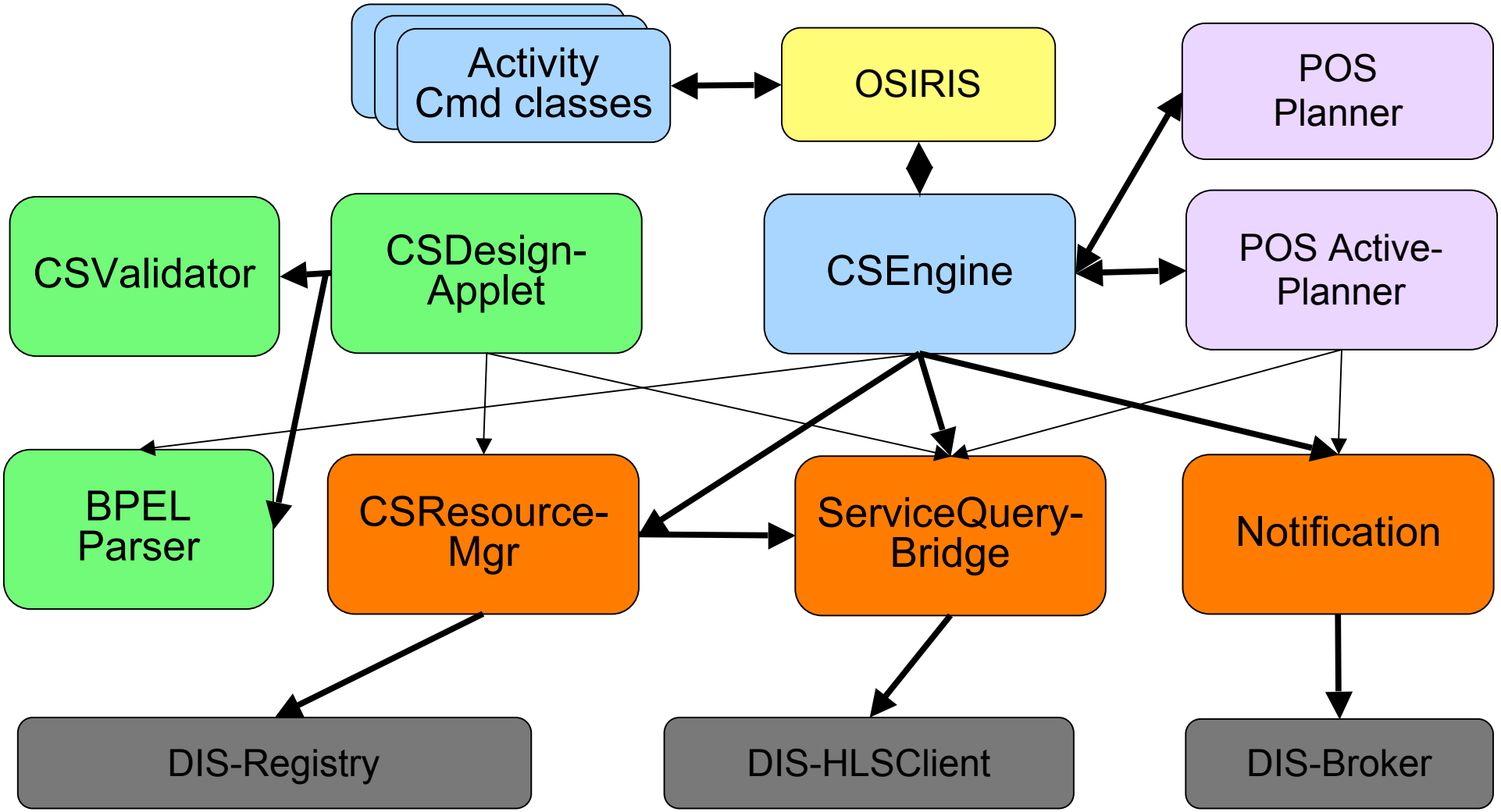




Process Optimization

- User queries are being submitted to the Query Process Optimization Service
- Processes allowing to search within the content of the DL have to return results to their users as soon as possible
- This service consolidates information provided by services such as resource description, indexing and personalization and produces a “query” execution plan which is actually a verified, optimized process (workflow) that will ultimately deliver the desired results.
- Process optimization is an important task that may change the description of a process while, at the same time, the provable correctness of these processes must be preserved

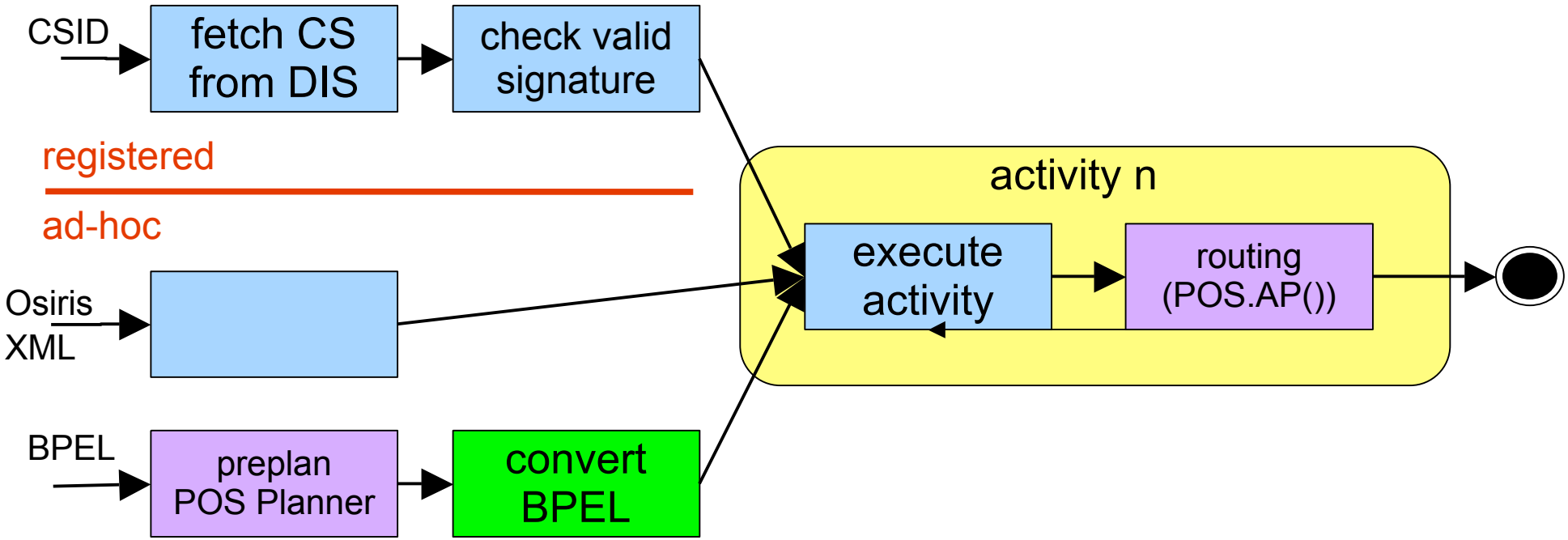
Overview of Process Components



Process Execution: Interaction of Services

DILIGENT distinguishes two „types“ of processes: registered and ad-hoc

- Registered: defined e.g. via portlet, stored in DIS
- Ad-hoc: pass process definition on invocation (e.g. search)



Lessons Learned

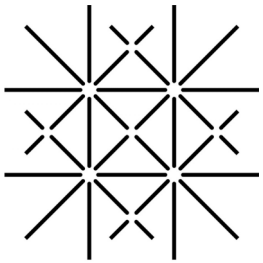
- Processes facilitate application development in a SOA environment
- Traditional process management considers a centralized process engine (or a centralized instance database)
  - this does not scale to Grid-size
  - DILIGENT provides a fully distributed approach to process management that
    - avoids any single point of failure
    - allows to dynamically adjust to changing environments (e.g., when new services are available)

Building Digital Libraries on Service Oriented Architectures

ISTITUTO DI SCIENZA E TECNOLOGIE
DELL'INFORMAZIONE "A. FAEDO"

Conclusions & Future Directions

Tutorial at JCDL 2007
June, 19th 2007



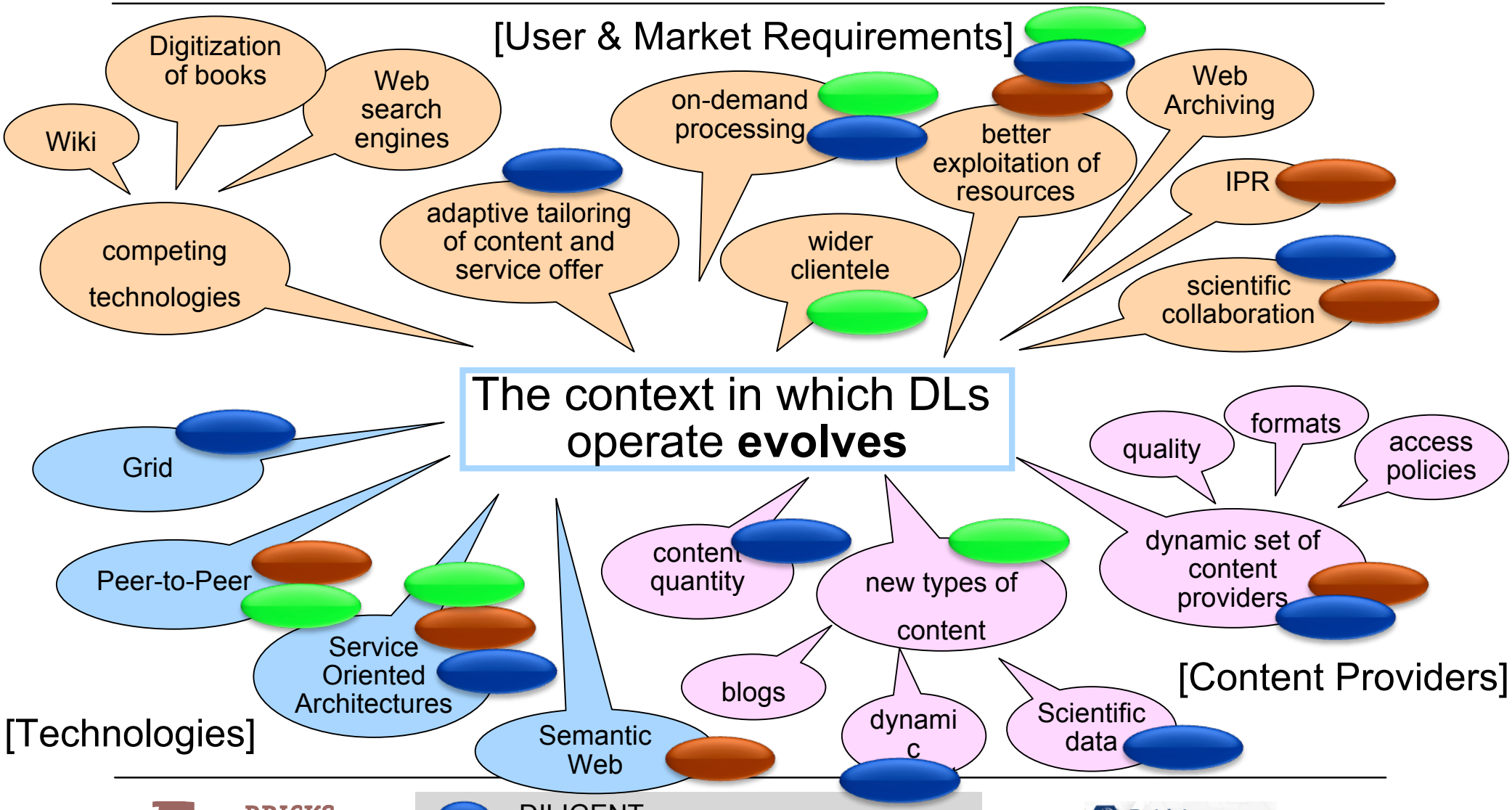
UNI
BASEL

Claudia Niederée (L3S) + All

Agenda

09:00 – 09:20	**Introduction: Motivation & Challenges**
09:20 – 09:45	**Challenges of bringing DL to distributed Infrastructures**
09:45 – 10:30	**Underlying Technologies and their promises** (SOA, P2P, Grid)
10:30 – 10:45	*Coffee break*
10:45 – 12:00	**Solutions for decentralized DL infrastructures** (with BRICKS Demos)
12:00 – 12:30	**DelosDLMS - the DELOS Digital Library Management System**
12:30 – 13:30	Lunch
13:30 – 14:00	**DelosDLMS Demos**
14:00 – 15:00	**Building DL services on the Grid** (DILIGENT)
15:00 – 15:30	*Coffee break*
15:30 – 16:45	**DILIGENT Demos**
16.45 – 17:00	**Conclusions and future directions**

Summary

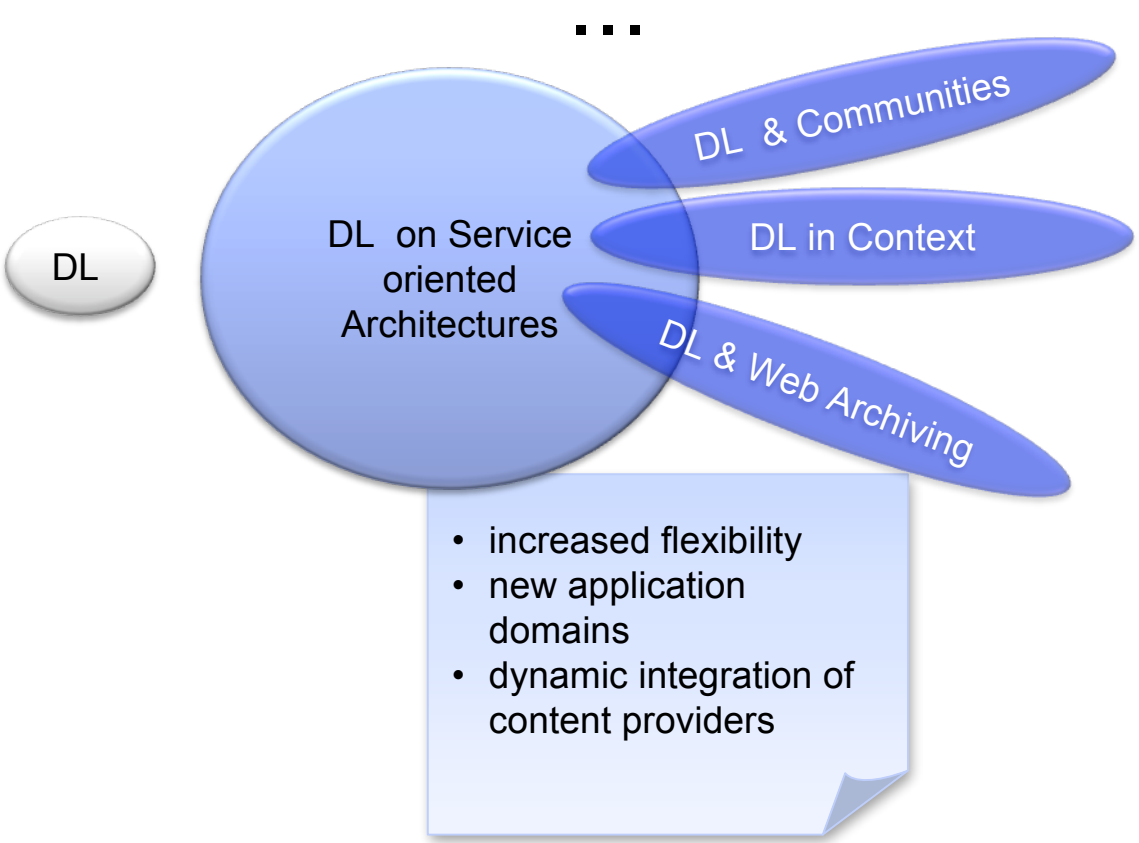


DILIGENT
BRICKS
DELOS

Next Steps for Establishment & Take Up

- Consolidation & Standarization
  - compatibility
  - exchangable modules
- Attraction of a wider community
  - cultural heritage related
  - new application areas (e.g. e-Science)
- Solving of open challenges
  - IPR
  - quality assurance
  - long-term preservation
  - information integration
  - interoperability

Future Directions: DL in Context



DL and Web Archiving

- Increased role of Web in information access, content creation, reflecting opinions, performing transactions etc. in daily life activities, society, culture, politics, education, etc.
- → increasing importance of sound Web archiving
- first generation of Web archiving technology in place
- but: poor access methods for Web archives e.g. archive.org
- little integration with other information sources
- special challenges due to inherent structure w.r.t access

- DL opportunities:
  - apply experience of DL community in metadata, search, etc.
  - integrate Web archive collection with DL collection for more comprehensive information provision
  - develop new types of access methods (time, evolution) and application for targeted communities

DL in Context

Support for the individual so far:

- standard query based access
- Personalization support (e.g. considering interests and preferences)
- Supprt for personal digital libraries (e.g. Daffodil) and virtual libraries (DILIGENT)

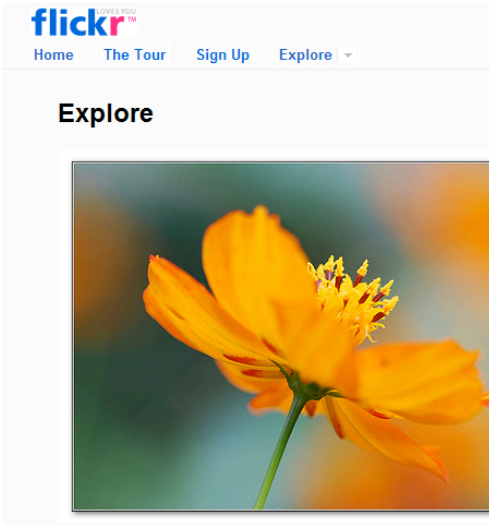
DL Opportunities

- Improved personalization support
- integration of DL functionality in working processes
- targeted pro-active information provision
- requires process- and context modelling
- application example: e-Science

DL & Communities

- Strong trend: Involvement of communities
  - community based content creation (e.g. Wikipedia)
  - community-based content collection (e.g. YouTube, Flickr)
  - community-based recommendation + rating (e.g. online Bookstores)
  - community based tagging (folksonomies, ...

- DL Opportunities
  - use the intelligence of communities to improve DL content and service offer (e.g. content enrichment and interlinking)
  - exploit DL technologies in community-based systems
  - create even richer content by combining community created content with DL content in seamless ways.



Questions & Discussions