FP7 Project nr 226487 Project start date: 01 May 2009
Acronym: EuroGEOSS
Project title: EuroGEOSS, a European approach to GEOSS
Theme: FP7-ENV-2008-1: Environment (including climate change)
Theme title: ENV.2008.4.1.1.1: European Environment Earth Observation system supporting INSPIRE and compatible with GEOSS (Global Earth Observation System of Systems)
Web site: www.eurogeoss.eu

D.2.2.2. Specification of EUROGEOSS Initial Operating Capacity

Title	D.2.2.2. Specification of EUROGEOSS Initial Operating Capacity
Creator	WP2 WG: Multidisciplinary Interoperability
Creation date	01/02/2010
Date of last revision	25/02/2010
Subject	Initial Operating Capacity of the EuroGEOSS infrastructure
Status	☐ Draft ☒ Final
Publisher	EuroGEOSS, WP2
Type	Text
Description	
Contributor	
Format	Pdf
Source	
Rights	☐ Restricted ☒ Public
Identifier	EuroGEOSS_D_2_2_2_final
Language	En
Relation	WP2
Coverage	Not applicable

These are Dublin Core metadata elements. See for more details and examples http://www.dublincore.org/

TABLE OF CONTENTS

FIGURES

TABLES

ACRONYMS AND ABBREVIATIONS

Abbreviation	Name
ADC	Archtecture and Data Committee
AIP-2	Architecture Implementation Pilot, Phase 2
AOC	Advanced Operating Capacity
BPEL	Business Process Execution Language
CAP	Common Agricultural
CEN	European Committee for Standardization
CRS	Coordinate Reference System
CSR	Component and Services Registry
CS-W	Catalog Service
CT	Consolidation Team
DEM	Digital Elevation Models
DOPA	Digital Observatory of Protected Areas
DoW	Description of Work
DS DS	Data Specifications Drafting Team
EbRIM	electronic business Registry Information Model
ENM	Ecological Niche Model
EO	Earth Observation
EO-DAIL	Earth Observation Data Access & Integration Layer Implementation
ERCS	Emergency Response Core Services
ESA	European Space Agency
ESDI	European Spatial Data Infrastructure
EU	European Union
FP7	Seventh Framework Programme
FTS	Fast Track Service
GAS	GMES Atmosphere Service
GCI	GEOSS Common Infrastructure

GEMS	Global and regional Earth-system (Atmosphere) Monitoring using Satellite and in-situ data (GEMS)
GEO	Group on Earth Observations
GeoRM	Geo Rights Management
GeoRSS	Geospatially-enabled RSS and Atom feeds
GEOSS	Global Earth Observation System of Systems
GIGAS	GEOSS, INSPIRE and GMES an Action in Support
GMES	Global Monitoring for Environment and Security
GSCB	Ground Segment Coordination Body
HMA	Heterogeneous Mission Accessibility
IG	Implementation Group
INSPIRE	Infrastructure for Spatial Information in Europe
IOC	Initial Operating Capacity
IPCC	Intergovernmental Panel on Climate Change
IR	Implementing Rules
ISO	International Organization for Standardization
JRC	Joint Research Centre
LMCS	Land Monitoring Core Service
MACC	Monitoring Atmospheric Composition and Climate
MCS	Marine Core Service
MS	Member State
NDVI	Normalize Difference Vegetative Index
NGO	Non-Governmental Organization
NSDI	National Spatial Data Infrastructure
O&M	Observation and Measurement
OGC	Open Geospatial Consortium
OWS	OGC Web Services
PROMOTE	PROtocol MOniToring for the GMES Service Element
QC	Quality Control
SA	Support Action

SAFER	Services and Applications For Emergency Response
SBA	Societal Benefit Area
SEIS	Shared Environmental Information System
SensorML	Sensor Markup Language
SIF	Standards and Interoperability Forum
SIR	Standards and Interoperability Registry
SOS	Sensor Observation Service
UML	Unified Modeling Language
UN	United Nations
UNEP	United Nations Environment Programme
W3C	World Wide Web Consortium
WCMC	World Conservation Monitoring Centre
WCS	Web Coverage Service
WCS-T	Web Coverage Service, Transactional
WFS	Web Feature Service
WFS-T	Web Feature Service, Transactional
WMS	Web Map Service
WPS	Web Processing Service
WP	Work Package
XML	eXtensible Markup Language

1 INTRODUCTION

EuroGEOSS demonstrates the added value to the scientific community and society of making existing geographic systems and applications interoperable and used within the GEOSS and INSPIRE frameworks. The project is building an initial operating capacity for a European Environment Earth Observation System in the three strategic areas of Drought, Forestry and Biodiversity.

The concept of inter-disciplinary interoperability requires research in advanced modeling from multi-scale heterogeneous data sources, expressing models as workflows of geo- processing components reusable by other communities, and ability to use natural language to interface with the models.

1.1 Purpose and scope

This document describes the Initial Operating Capacity (IOC) of EuroGEOSS to support multidisciplinary interoperability. The first implementation of the EuroGEOSS IOC described in this report takes into account most of the requirements outlined in the review of the main European and international initiatives addressing multidisciplinary interoperability –i.e. the EuroGEOSS Deliverable 2.2.1 reporting on requirements from GEOSS, INSPIRE, GMES, SEIS and related initiatives and projects[1].

The requirements collected have contributed to provide specifications, guidelines and prototypical implications in order to support the implementation of the EuroGEOSS IOC and the infrastructures in each of the EuroGEOSS thematic areas.

To explain the EuroGEOSS IOC functionalities and their usability, the present document introduces first the *brokering framework* concept. Then, Chapter 2 discusses the broker component implemented, which is the EuroGEOSS IOC enabling technology. Full details of the broker implementation are given in Appendix A. Chapter 3 explains how to populate the IOC and make specific components available through this capacity. Finally, the report presents the interfaces and clients for accessing the IOC.

1.2 A Brokering Framework

According to the ultimate System of Systems implementation approach, the EuroGEOSS IOC adopts the Information Modeling Approach (MDA) by sharing metadata and implementing systems interoperability through international standards. That is achieved by making use of the Service-Oriented technology. However the multi-disciplinary environment, which characterizes the EuroGEOSS capacity, requires the support of different interoperability data models and protocols (i.e. standards). This is a challenge for the IOC clients which must implement a plethora of different interoperability protocols and data models. To address this issue, an extended SOA approach can conveniently be used to provide a harmonized access service: the SOA-brokering approach (see Figure 1).

[1] http://www.eurogeoss.eu/Documents/EuroGEOSS_D_2_2_1.pdf

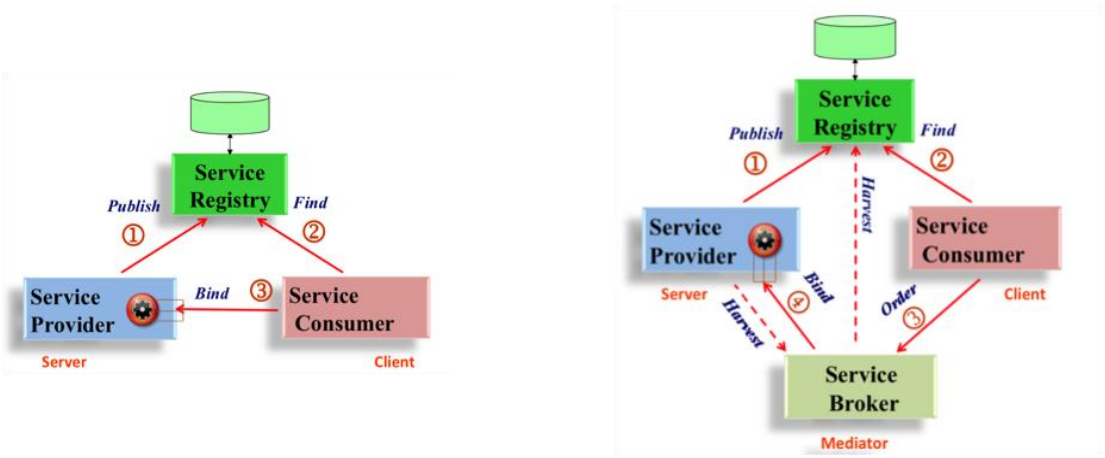

Figure 1 – SOA and SOA-brokering Approaches

2 EUROGEOSS IOC – BROKERING COMPONENT

The EuroGEOSS brokering component implements a framework to federate well-accepted catalog, inventory and access standard services. As for International standards, OGC CS-W, WCS, WMS and WFS are supported. Community standards are also supported: for example the UNIDATA THREDDS/OPeNDAP inventory service, SeaDataNet CDI (Common Data Index) and GBIF (Global Biodiversity Information Facility) catalog and access services can be federated. A precise list appears in the section about registration of new components.

In turn, international standard interfaces are published by the broker: CS-W Core, CS-W ISO, CS-W ebRIM CIM and EO extension packages.

Besides, an extended catalog interface is available in order to provide additional functionalities to the users - status feedback, catalog configuration, etc. Furthermore, distributed queries over a hierarchical data model are supported, with support for incremental results retrieval.

The support of query distribution allows clients to discover and evaluate resources, managed by heterogeneous servers, using a uniform and consistent interface. It is possible to finalize selections by any combination of the following general criteria for discovery: spatial extent, temporal extent, keywords (free text search) and data source (i.e. Where, When, What, Who); ISO 19139 properties are also valid queryables, as well as EO and CIM ones.

The brokering framework is comprised of several modular components which provide the capabilities of one extended distributed catalog model.

In the Appendix of this document the brokering component architecture description is given in terms of separate but interrelated viewpoints, as in the Reference Model for Open Distributed Processing “methodology”. The description might be especially useful to the aim of developing a client to the brokering component. The following viewpoints are detailed in the Appendix:

The **information viewpoint**, which focuses on the semantics of the information and the information processing performed. It describes the information managed by the system and the structure and content type of the supporting data.

The **computational viewpoint**, which describes functional decomposition on the system into objects which interact at interface level. It describes the functionalities provided by the system and its functional decomposition.

The **engineering viewpoint**, which focuses on the mechanisms and functions required to support distributed interactions between objects in the system. It describes the distribution of processing performed by the system to manage the information and provide the broker functionalities.

The **technology viewpoint**, which focuses on the choice of technology of the system. It describes the technologies chosen to provide the processing and presentation of information.

3 HOW TO REGISTER A NEW COMPONENT/SERVICE

The following guidelines are to be used in order to register a new component or a service into the EuroGEOSS framework. Three general cases can be distinguished:

1. Publishing a standard service on the brokering component
2. Publishing data on a thematic area catalog
3. Establishing a custom interoperability arrangement

3.1 Publishing a standard service on the brokering component

EuroGEOSS partners that fall in this case have their geospatial resources already served by an operative standard service whose type is one listed in the following table:

ServiceType	Supported Versions	Description
CDI	*1.0.3, 1.3, 1.4*	*Common data index service; in use by the marine community*
Deegree	*2.2*	*Deegree catalog service based on CSW 2.0.2 ISO AP*
Extended brokering interface	*6.0*	*Extended catalog interface based on Broker 6.0*
GBIF		*Global Biodiversity Facility services*
GENESI DR		*GENESI DR project service interface*
GeoNetwork	*2.2.0, 2.4.1*	*GeoNetwork catalog service based on CSW 2.0.2 ISO AP*
OGC CSW	*2.0.2 (CORE, AP ISO, ebRIM/CIM, ebRIM/EO)*	*Catalogue Services for the Web*
OGC WCS	*1.0, 1.1*	*Web Coverage Service*
OGC WFS	*1.0*	*Web Feature Service*
OGC WMS	*1.3.0, 1.1.1*	*Web Map Service*
OpenSearch		*OpenSearch engines*
THREDDS	*1.0.1, 1.0.2*	*Thematic Realtime Environmental Distributed Data Services*

Table 1 – Services Supported by EuroGEOSS Broker

The service will be readily available in the EuroGEOSS framework after its registration on the brokering component.

Contact one of these partners to complete the registration process:

- [BRGM] Mauclerc Anthony A.Mauclerc@brgm.fr

- [JRC] Lorenzino Vaccari lorenzino.vaccari@jrc.ec.europa.eu
- [CNR] Mattia Santoro santoro@imaa.cnr.it

The following information are needed:

- Service Type (e.g. “WMS”)
- Service Version (e.g. “1.3.0”)
- Title (e.g. “[Marine] Salinity data service”)
- Service Description (e.g. “This service is deployed at the University of __ and publishes accurate salinity data for the marine community”)

3.2 Publishing data on a thematic area catalog

If a catalog is already available in EuroGEOSS for the thematic area of interest, you can register your data and/or (when appropriate) metadata directly on it. In this case contact the responsible party of the given catalog and send her/him a description of the available data and/or metadata that you want to publish. Further steps to be arranged between the catalog responsible party and the data publisher will lead to the complete publication.

3.3 Establishing a custom interoperability arrangement

The third possibility applies in case the data service to be registered is not of a type natively supported by the brokering component. In this case direct registration cannot be accomplished. However, a special arrangement could be established between the new component and the broker. This consists in extending the brokering component by the creation of a new accessing protocol capable of integrating the new service in the framework.

To obtain more information, documentation and support on the development of a new accessing protocol contact:

- [CNR] Mattia Santoro santoro@imaa.cnr.it
- [CNR] Enrico Boldrini boldrini@imaa.cnr.it

4 HOW TO ACCESS THE IOC

EuroGEOSS infrastructure can be accessed in several ways. It is possible to gain immediate access with readily-available clients such as GI-go and GeoNetwork (described in Section 5), or access through the interfaces described in this Section.

The adopted brokering component exposes more than one interface towards external systems. Each exposed interface implements an international standard specification for catalog service. This *profiling feature* makes the EuroGEOSS infrastructure accessible by a wide range of clients; furthermore, publishing international standard interfaces ensures interoperability with the main international infrastructures (GEO/GEOSS (GEO, 2005), GMES (http://ec.europa.eu/gmes/overview.htm), INSPIRE (European Parliament, 2009), etc.).

The following interface specifications are supported and exposed by the EuroGEOSS broker:

- OGC CSW Core (OGC, 2007core): this interface is recommended by the GEO/GEOSS;
- OGC CSW ISO Application Profile (OGC, 2007iso): this interface is identified by INSPIRE Implementing Rules (IRs) as the reference for ESDI catalog services;

- OGC CSW ebRIM/EO Extension Package (OGC, 2008eo): this extension package of the CSW is recommended by the GMES/ESA-HMA initiative (http://earth.esa.int/hma/index.html);
- OGC CSW ebRIM/CIM (OGC, 2007cim);
- OGC CSW OpenSearch Extension (OGC, 2008os) and the GENESI-DR (Ground European Network for Earth Science Interoperations - Digital Repositories) (http://www.genesi-dr.eu/) catalog interface which is based on it.

A non-standard interface is also exposed by the EuroGEOSS broker. This interface was introduced with the aim to extend the common functionalities of CSW and is based on the GI-cat framework (CNR-ESSI Lab, 2010).

In the next sections we provide more detailed descriptions of the implemented interfaces, in particular the broker extended interface – since it is not standardized.

4.1 OGC CSW/Core interface

The CSW/Core interface (v.2.0.2) is the “common” interface for all the CSW application profiles. All the CSW based catalogs must publish this interface in order to provide interoperability through the support of queries on a basic data model (based on Dublin Core). All the mandatory methods were implemented by the Broker:

Operation	Description	Binding
OGC_Service.GetCapabilities	OGC_Service.getCapabilities	HTTP/GET + HTTP/POST + SOAP
CSW Discovery.DescribeRecord	Discovery.describeRecordType	HTTP/GET + HTTP/POST + SOAP
CSW Discovery.GetRecords	Discovery.query	HTTP/POST + SOAP
CSW Discovery.GetRecordById	Discovery.present	HTTP/GET + HTTP/POST + SOAP

Table 2 - Mandatory Methods of CSW Core Interface

OGC document (OGC, 2007core) can be used as a reference to implement a client capable of accessing this Broker interface.

4.2 OGC CSW/ISO interface

The Broker can publish a CSW/ISO interface. This OGC standard interface is defined in specification (OGC, 2007iso), that can be used as a reference to implement a client to access broker through this interface.

The Broker, in particular, implements the following (mandatory) methods:

Operation	Description	Binding
OGC_Service.GetCapabilities	OGC_Service.getCapabilities	HTTP/GET + HTTP/POST + SOAP
CSW Discovery.DescribeRecord	Discovery.describeRecordType	HTTP/GET + HTTP/POST + SOAP
CSW Discovery.GetRecords	Discovery.query	HTTP/POST + SOAP
CSW Discovery.GetRecordById	Discovery.present	HTTTP/GET + HTTP/POST + SOAP

Table 3 - Mandatory Methods of CSW ISO AP Interface

An extended function has been added to the *Filter* element (see the specification), it is called *isChildOrSelf* and it can be used to support hierarchical queries.

4.3 OGC CSW/ebRIM (CIM extension package) interface

The broker can publish a CSW ebRIM/CIM interface. This OGC standard interface is defined in specification (OGC, 2007cim), that can be used as a reference to implement a client to access broker through this interface.

The Broker, in particular, implements the following (mandatory) methods:

Operation	Description	Binding
OGC_Service.GetCapabilities	OGC_Service.getCapabilities	HTTP/GET + HTTP/POST + SOAP
CSW Discovery.DescribeRecord	Discovery.describeRecordType	HTTP/GET + HTTP/POST + SOAP
CSW Discovery.GetRecords	Discovery.query	HTTP/POST + SOAP
CSW Discovery.GetRecordById	Discovery.present	HTTTP/GET + HTTP/POST + SOAP
CSW Discovery.GetRepositoryItem	Discovery.GetRepositoryItem	HTTTP/GET

Table 4 - Mandatory Methods of CSW/ebRIM CIM Interface

An extended function has been added to the *Filter* element (see the specification), it is called *isChildOrSelf* and it can be used to support hierarchical queries.

4.4 OGC CSW/ebRIM (EO extension package) interface

The Broker can publish a CSW ebRIM/EO interface. This OGC standard interface is defined in specification (OGC, 2008eo), that can be used as a reference to implement a client to access Broker through this interface.

Broker in particular implements the following (mandatory) methods:

Operation	Description	Binding
OGC_Service.GetCapabilities	OGC_Service.getCapabilities	HTTP/GET + HTTP/POST + SOAP
CSW Discovery.DescribeRecord	Discovery.describeRecordType	HTTP/GET + HTTP/POST + SOAP
CSW Discovery.GetRecords	Discovery.query	HTTP/POST + SOAP
CSW Discovery.GetRecordById	Discovery.present	HTTTP/GET + HTTP/POST + SOAP
CSW Discovery.GetRepositoryItem	Discovery.GetRepositoryItem	HTTTP/GET

Table 5 - Mandatory Methods of CSW/ebRIM EO Interface

An extended function has been added to the *Filter* element (see the specification), it is called *isChildOrSelf* and it can be used to support hierarchical queries.

4.5 OGC CSW OpenSearch Extension

OpenSearch (http://www.opensearch.org/Home) is a collection of technologies that allow publishing search results in a format suitable for syndication and aggregation. It is a way for websites and search engines to publish search results in a standard and accessible format.

OpenSearch mainly consists of:

1. **OpenSearch Description document**: XML file that identify and describe a search engine;
2. **OpenSearch Query Syntax**: describe where and how to retrieve the search results;
3. **OpenSearch Response**: format for providing open search results.

The OpenSearch Description document allows clients to retrieve service metadata from a server. The following is an example of such a document:

```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/">
   <ShortName>Web Search</ShortName>
   <Description>Use Example.com to search the Web.</Description>
   <Tags>example web</Tags>
   <Contact>admin@example.com</Contact>
   <Url type="application/rss+xml"
        template="http://example.com/?q={searchTerms}&amp;pw={startPage?}&amp;format=atom"/>
</OpenSearchDescription>
```

The most relevant part of the document is represented by the *Url* element, which describes the interface through which a client can make requests for a resource, such as search results, search suggestions, or additional description documents. The *template* attribute of the *Url* element can be used to represent a parameterized form of the URL by which a search engine is queried.

Search clients will process the template and attempt to replace each instance of a template parameter, generally represented in the form {name}, with a value determined at query time.

(OGC, 2008os) is an official OGC change request which proposes to extend both the OpenSearch specification (adding some new template parameters to the default ones) and the CSW interface.

The aim is to allow the Catalog Service to support OpenSearch requests. It follows a brief description of the proposed extensions.

4.5.1 OpenSearch templates extensions

OpenSearch defines a set of standard template parameters (http://www.opensearch.org/Specifications/OpenSearch/1.1#OpenSearch_1.1_parameters) that enable clients to search using several constraints, such as keyword or language.

As well as this standard parameters, several others can be defined by different communities. The (OGC, 2008os) defines some new parameters (for the support of the ebRIM application profile) and suggests the use of geographic (OpenSearch-Geo extensions) and time (OpenSearch-Time extensions) parameters defined in the Geospatial context.

4.5.2 CSW operations extension

The CSW operation *Discovery.GetRecords* (see section 4.1), allows clients to retrieve the set of available metadata records with the client-desired parameter values.

Since this operation is not really amenable to support the OpenSearch specification, the OpenSearch change request proposes to add a couple of new operations, *GetRecordsSimple* and *GetAssociation*.

GetRecordsSimple

As the name suggests, this operation may be seen as a simplified version of the *Discovery.GetRecords*; in particular, it allows the catalogue to receive queries according to the OpenSearch syntax via the HTTP-GET protocol. The following is an example of the *Url* element of such a description document.

```
 <Url type="application/atom+xml"
template="http://www.pvretano.com/cwwrs/cubeserv.cgi?service=CSW&
      version=3.0.0&
      request=GetRecordsSimple&
      resultType=hits&
      outputFormat=text/xml&
      outputSchema=http://www.w3.org/2005/Atom&
      startPosition={startIndex?}&
      maxRecords={count?}&
      searchTerms={searchTerms?}&
      objectType={ogcos:objectType?}&
      bbox={geo:box?}&
      class={ogcos:class?}&
      scope={ogcos:scope?}&
      time={ogcos:time?}&
      temporalOp={ogcos:temporalOp?}"/>
```

GetAssociation

This operation uses the previous mentioned template parameters for the support of the ebRIM application profile, and allows the catalogue to receive requests that involve a particular kind of association for a specified record.

Accessing the Broker using OpenSearch

The mentioned change request does not specify mandatory obligation for none of two described operations. Since the *GetAssociation* operation is intended only for the support of particular aspects of ebRIM application profile, only the *GetRecordsSimple* operation was implemented by the Broker. This allows clients to query the Broker using the OpenSearch Description document expressly published. Query responses are presented in *Atom Syndication Format*.

The following three steps must be executed by a client which wants to query the Broker using OpenSearch interface :

1. Retrieving the Broker OpenSearch Description document
2. Substituting the desired template parameters with an opportune values, thus obtaining the corresponding HTTP-GET request
3. Sending the request

The first step can be done in two ways. If the OpenSearch Description Document URL is already known to the client, work is done.

```
<link rel="search"
      type="application/opensearchdescription+xml"
      href="http://example.com/content-search.xml"
      title="Content search" />
```

Otherwise, a client can take advantage of the OpenSearch *Autodiscovery* feature. This feature enables clients to retrieve the Description Document through the use of *link* elements which must be included, for example, in an HTML page.

Since the Broker includes a *link* element in its home page, a client can get the Description Document knowing only the Broker home page.

The *link* element has the following properties:

- The *type* attribute contains the value "application/opensearchdescription+xml".
- The *rel* attribute contains the value "search".
- The *href* attribute contains the URI which resolves to the Broker OpenSearch description document.

When available, the search bar of the browsers represents another interesting way to access the broker through the OpenSearch interface. At the moment, this feature is available only using Mozilla Firefox browser (version 2 or higher). The Firefox search bar situated in the upper-right corner, comes with some preloaded search engines such as Google and Yahoo, but other engines can be easily added. Once added, a search engine can be queried entering a keyword in the search bar, and receiving immediate answers from the chosen engine.

Following this simple steps, the broker can be added as a Firefox available search engine:

1. Open the Broker homepage with Mozilla Firefox browser (version 2 or higher)

Figure 2 – Search Tab in Mozilla Firefox Browser

2. As depicted in Figure 2, the down arrow on the search bar should now be highlighted

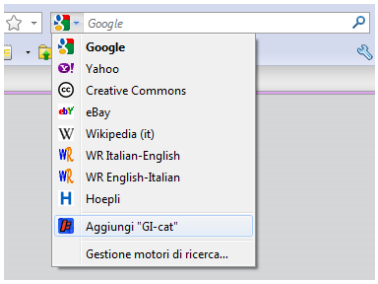

Figure 3 – Add a EuroGEOSS/GI-cat OpenSearch catalog (using an Italian Operative System it is displayed as: "Aggiungi GI-cat")

3. Click on the down arrow and select "Add GI-cat" (see Figure 3)

Figure 4 - Search Tab in Mozilla Firefox Browser with GI-cat Search Engine

4. Enter a keyword in the search bar and press the enter button (or click on the lens icon)

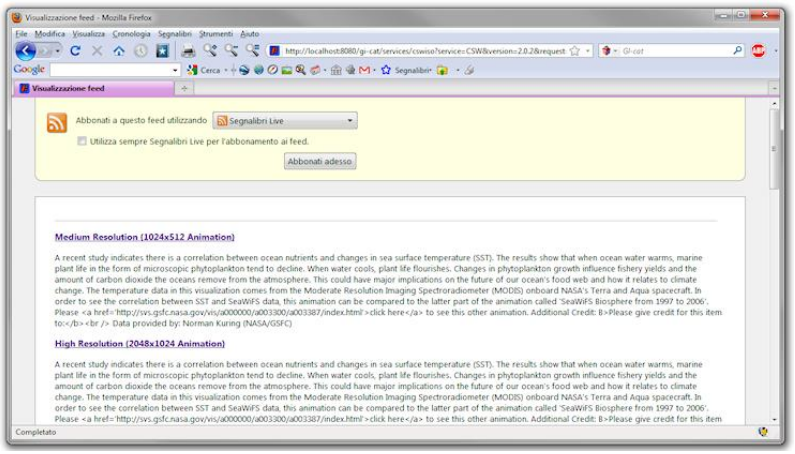

Figure 5 – Search Results Visualized as Atom Feeds

5. Matching results are visualized as Atom feeds

4.5.3 OpenSearch interface extension for addressing the GENESI-DR Interface

The Broker also implements an extension of the OpenSearch interface to interoperate with GENESI-DR[2] (Ground European Network for Earth Science Interoperations - Digital Repositories) research infrastructure. GENESI-DR project aims to establish an open Earth Science Digital Repository access for European and world-wide science users. The project is led by the European Space Agency (ESA).

GENESI-DR defines several components; one of these is the Catalogue Access Service (CAS) which provides access to the distributed Discovery Agents for querying and publishing metadata. The Broker GENESI-DR interface allows for integration in the infrastructure as a CAS component. Since this component (CAS) uses an OpenSearch interface, the Broker publishes an appropriate OpenSearch Description document. Accessing the Broker through this interface can be done following the same steps described in the previous section.

4.6 EuroGEOSS Broker Extended Interface

This interface is designed to overcome the limitations of the CSW interface. In particular, it provides incremental query result presentation and feedback support, as well as a manager interface to dynamically configure the Broker. Moreover, it defines methods for enabling a practical performance of catalog browsing task.

The interface operations can be logically divided into four main functional modules:

- *Discovery module*: used to discover resources metadata, either by means of catalog browsing or through a query operation.
- *Session module*: used to abort a running session or to get feedback on a session status.
- *Access module*: used to get binary dataset associated to a resource or a resource preview.
- *Manager module*: used to get catalog capabilities and settings and dynamically configure Broker.

Moreover, the next EuroGEOSS Broker release will introduce additional functional modules, namely: harvesting, and transactional modules.

In the rest of this section, we provide descriptions of the each operation, grouped by functional module. At the EuroGEOSS broker documentation page (EuroGEOSS, 2010) we provide several examples of request/response documents for the methods described below as well as their UML diagrams.

4.6.1 Discovery Module

Table 2 lists the operations belonging to this functional module.

Operation	Description	Binding
GetResource	Retrieve the resource with the given id	HTTP/SOAP
GetContent	Browse the children resources of a resource	HTTP/SOAP
Query	To perform a query on the catalog	HTTP/SOAP

Table 6 - EuroGEOSS Broker: Discovery Interface Operations

[2] http://www.genesi-dr.eu/

GetResource

This operation retrieves the description of the resource with the given identifier(s), encoded in the Broker XML data model.

Name	Type	Description	Obligation
Resource Identifier	List<String>	Identifier of the Resource	O (default = ROOT)
OperationID	String	Identifier of the Operation	M

Table 7 – Parameters of GetResource Method in EuroGEOSS Broker Interface

The resource identifiers are assured to be unique for a given resource in a given catalog. The default value of the *Resource Identifier* parameter is the list containing only one element: "ROOT" – that is the catalog root in the catalog hierarchical model.

The *OperationID* parameter is used by the methods belonging to the Session functional module (e.g. *Cancel*, *Status*, etc.).

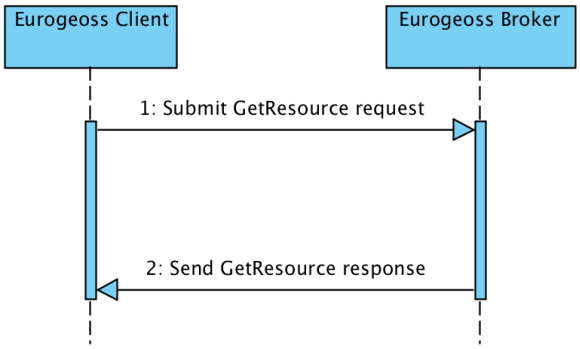

Figure 6 - GetResource sequence diagram

GetContent

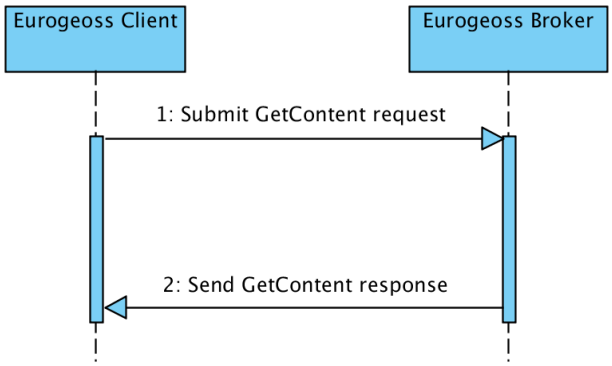

Figure 7 - GetContent operation sequence diagram

The *GetContent* operation retrieves the content of the resource identified by a given *Resource Identifier*; the content of a resource is a list of resources that are to be intended as "children" (or

also “subsets”) of the resource itself. When the *Deep* parameter is set to true, the children of the requested resource are all visited until their leafs. The default value of the *Deep* parameter is false.

The default value of the *Resource Identifier* parameter is “ROOT”.

For the *OperationID* see the GetContent paragraph.

Name	Type	Description	Obligation
Resource Identifier	String	Identifier of the Resource	O (default = ROOT)
OperationID	String	Identifier of the Operation	M
Deep	Boolean	This option enables the recursive visit of the children nodes of the Resource or not	O (default=true)

Table 8 - Parameters of GetContent Method in EuroGEOSS Broker Interface

This operation gives the clients the ability to perform a browsing task on the catalog. Through the client, users can browse the content of the Broker, and are able to interactively find resources of interest. By browsing the catalog users can discover also additional resources identifiers. These can be used to execute subsequent *GetContent* operations to further explore the catalog tree of resources or to constraint a subsequent query.

Query

The *Query* operation is used to discover which resources federated by the Broker satisfy the constraints contained in the query. These constraints are expressed in the *csw:GetRecords* parameter. This element can be encoded as an XML document according to several supported schemas: CSW Core, CSW ISO, CSW ebRIM/CIM, CSW ebRIM/EO. To identify the schema for the *csw:GetRecords* element we use the Uniform Resource Name (URN) of the corresponding OGC standard. The link to the OGC web page listing these URNs can be found at EuroGEOSS broker documentation page (EuroGEOSS, 2010).

The results of the query will be encoded using the same schema of the *csw:GetRecords* XML document submitted through the query request.

Name	Type	Description	Obligation
OperationID	String	Identifier of the Operation	M
OriginatorSchema	String	URN identifying the schema of the csw:GetRecords element	O (default = urn:ogc:serviceType:CatalogueService:2.0.2:HTTP:ISO19115/19119")
Synchronous	Boolean	Defines the type of the query response	O (default=true)
Scope	Enum: [LOCAL|DISTRIBUTED|FULL]	Scope of the query	O (default = DISTRIBUTED)
Timestamp	String	Timestamp of the Operation	O
HopCount	Integer	Maximum number of brokers that can be visited in series	O (default=5)
MaxResults	Integer	Maximum number of results to be returned	O (default = unbounded)
csw:GetRecords	XML Document	The “real” query is expressed using CSW GetRecords format.	C (see Synchronous and Asynchronous paragraphs)

Table 9 - Parameters of Query Method in EuroGEOSS Broker Interface

The *HopCount* parameter specifies the maximum number of “hops” from one broker to another allowed in the query propagation. This parameter was introduced to address the problem of eventual cyclic queries.

The optional *Scope* parameter can be used for querying only local resources (value set to LOCAL), only remote resources (value set to DISTRIBUTED) or both local and remote resources (value set to FULL).

Synchronous Query

By setting the *Synchronous* parameter to true, the full response is returned after catalog computation. This operation may require long time. In many cases it is preferable to asynchronously query the Broker (see the next section). In case of synchronous requests, the *csw:GetRecords* parameter is mandatory.

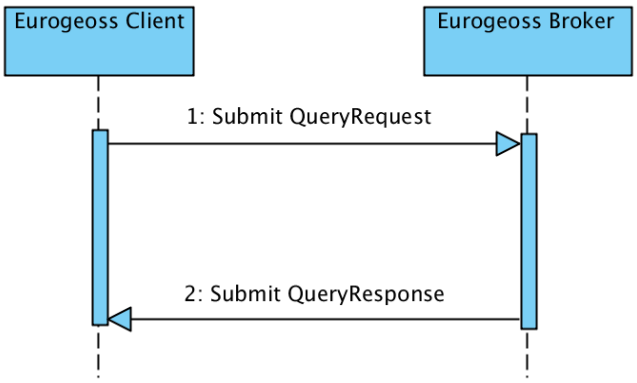

Figure 8 - Query operation (synchronous) sequence diagram

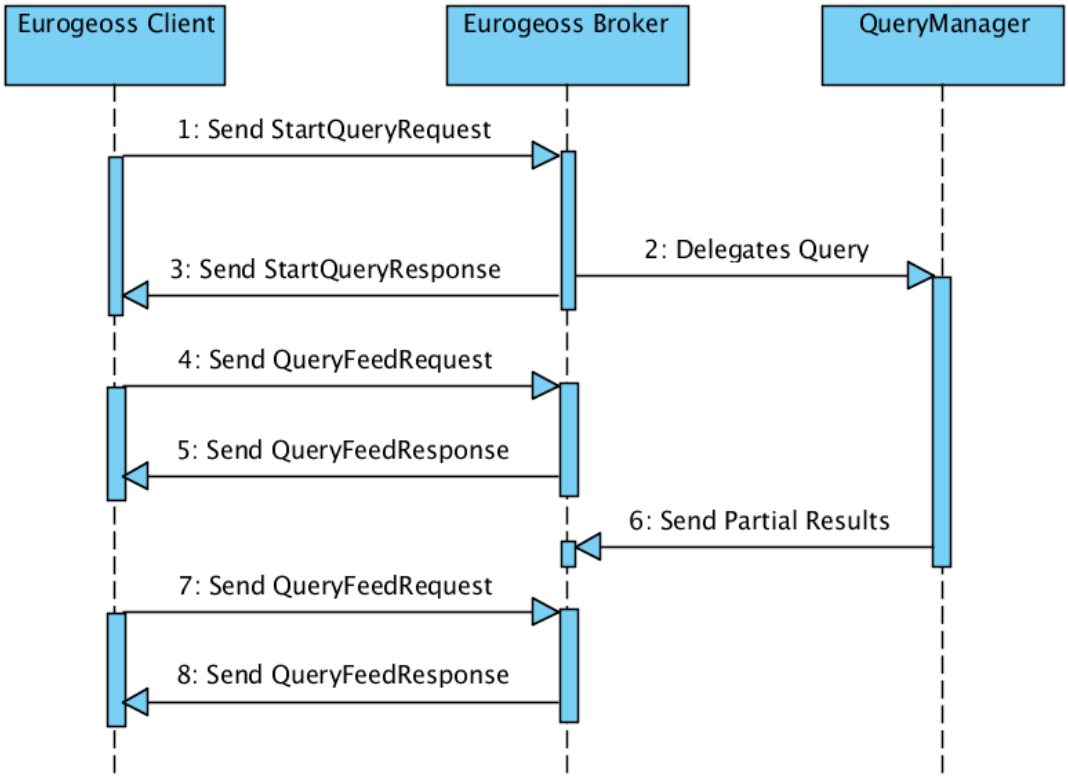

Figure 9 - An asynchronous query operation sequence diagram

Asynchronous Query

By setting the *Synchronous* parameter to false, an incremental query result retrieval mechanism (based on polling) is started.

Two phases can be distinguished:

a. **Query start**: in this phase the client sends the request, and the query elaboration starts on the catalog side. A client must assign a *OperationID* attribute to the request by using an UUID algorithm or similar mechanisms. The catalog should send back an acknowledge (ACK) message. In this phase the *csw:GetRecords* parameter is mandatory.

b. **Results retrieval**: in this phase the client requests results of a previously sent query (identified by the *OperationID* parameter). In this phase *Query* requests must NOT contain the *csw:GetRecords* parameter. Response from the catalog contains a (partial) set of results. The response contains a *QueryResponse* object; this is identified by a *OperationID* and can contain one or more *Parts*. Each *Part* can contain a set of resources and is identified by the *id* of the provenience provider (a distributed scenario is assumed). A specific attribute, called *completed*, in the *QueryResponse* object indicates if more parts are expected to come from the catalog or not.

4.6.2 Session Module

Operation	Description	Binding
Status	To retrieve feedback on a previous request	HTTP/SOAP
Cancel	To terminate a previous query operation	HTTP/SOAP

Table 10 - EuroGEOSS Broker: Session Interface Operations

Status

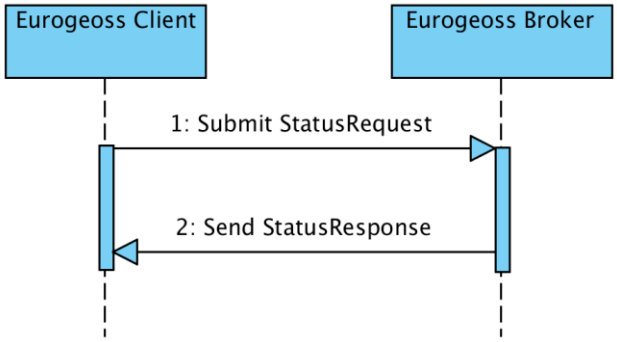

Figure 10 - Status operation sequence diagram

This method allows to retrieve feedback about the status of a previously started operation. The parameter is a list of *OperationID*s. When no *OperationID* is specified the broker assumes that all the ongoing operations are requested.

Name	Type	Description	Obligation
OperationID	List<String>	Identifier(s) of the operation(s) whose status is requested	O

Table 11 - Parameters of Status Method in EuroGEOSS Broker Interface

Cancel

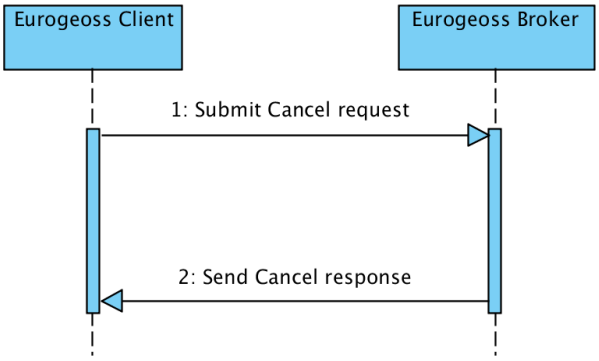

Figure 11 - Cancel operation sequence diagram

The *Cancel* request is used to terminate a previously started operation. The only required parameter of a *Cancel* request is the identifier of the operation (*OperationID*) to be canceled. It is also possible to specify the cause of the cancelation (*Cause* parameter). Moreover, it is possible to specify which component must cancel the request through the *TergetID* parameter. In this way it possible to cancel a query request only for one federated resource (e.g. because a timeout occurred) rather than for all federated resources.

Name	Type	Description	Obligation
OperationID	String	Identifier of the operation to be closed	M
Cause	Enum: [USER|TIMEOUT|ERROR]	The reason which caused the cancelation	O
TargetID	String	The identifier of the component to which the cancel request is submitted	O

Table 12 - Parameters of Cancel Method in EuroGEOSS Broker Interface

4.6.3 Access Module

Operation	Description	Binding
GetAccessParameters	Retrieves the XForms fields needed by Broker to access a given resource	HTTP/SOAP
GetData	Gets the binary data block associated to a resource OR the URL to use for subsequent data access OR the preview of the given resource	HTTP/SOAP

Table 13 - EuroGEOSS Broker: Access Interface Operations

An access operation (*GetData*) is usually preceded by a *GetAccessParameters* operation to get the available access parameters (these depend on the access service which provides the data). The access parameters are returned to the client in a XHTML+XForms document. Filling such a document, through the client graphical interface, the user will be able to download the data in desired format, coordinate reference system, spatial and temporal extents and resolutions, etc.

GetAccessParameters

Many of the access services which can be federated by the EuroGEOSS Broker (OGC WMS, WFS, WCS, etc) support complex access requests; that is, requests contain a set of parameters defining some properties which the downloaded dataset must have (e.g. bounding box, coordinate reference system, etc.). The *GetAccessParameters* operation retrieves available parameters for the dataset of interest.

Name		Type	Description	Obligation
OperationID		String	Identifier of the operation	M
CIOnlineResource		XML Document	ISO 19139 element describing the	M
	Name	String	Identifier of the resource on the catalog	M
	Protocol	String	URN (or *defacto* standard name) identifying the connection protocol to be used for the access operation	M
	Endpoint	String	URL of the access service for the resource	M

Table 14 - Parameters of GetAccessParameters Method in EuroGEOSS Broker Interface

The parameters of this method are listed in Table 14. To identify the resource of interest the client uses the *Name*, *Protocol* and *Endpoint* parameters (to be submitted encoded in a *CIOnlineResource* XML element, taken from ISO 19139). A client can obtain a valid *CIOnlineResource* for the catalog as result of a previous discovery operation.

The possibility to parameterize this request with a particular *CIOnlineResource* allows clients to choose a particular access service to be used for downloading the resource, rather than using a service chosen by the broker.

The *Name* field is the resource identifier of the resource which the user wants to access. The *Protocol* field specifies which access service shall be used for the access operation (OGC WCS, OGC WMS, OPeNDAP, etc.) among the supported ones for the resource of interest. We use URNs that can be downloaded from the broker documentation page (EuroGEOSS, 2010). The *Endpoint* field is the URL of the access service which publishes the resource of interest.

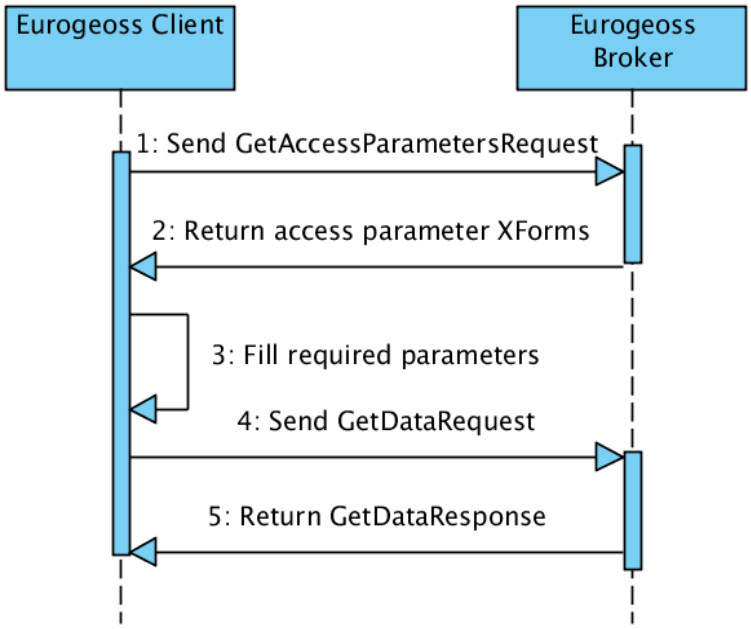

Figure 12 - Access operation sequence diagram

The operation returns an XHTML+XForms document representation of the parameters. The W3C site http://www.w3.org/MarkUp/Forms/ is the reference information for this emerging standard.

The following description is quoted from the site:

Traditional HTML Web forms don't separate the purpose from the presentation of a form. XForms, in contrast, are comprised of separate sections that describe what the form does, and how the form looks. This allows for flexible presentation options, including classic XHTML forms, to be attached to an XML form definition.

The following illustrates how a single device-independent XML form definition, called the XForms Model, has the capability to work with a variety of standard or proprietary user interfaces:

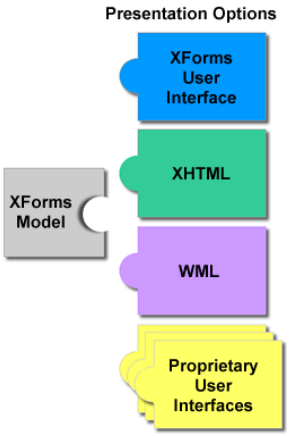

Figure 13 – Presentation Options of an XForm Model - Credits W3C

The XForms User Interface provides a standard set of visual controls that are targeted toward replacing today's XHTML form controls. These form controls are directly usable inside XHTML and

other XML documents, like SVG. Other groups, such as the Voice Browser Working Group, may also independently develop user interface components for XForms.

An important concept in XForms is that forms collect data, which is expressed as XML instance data. Among other duties, the XForms Model describes the structure of the instance data. This is important, since like XML, forms represent a structured interchange of data. Workflow, auto-fill, and pre-fill form applications are supported through the use of instance data.

Broker conveniently uses this standard to express the needed parameters for the access operation. A client could show to the user a visual representation of the XForms input page (see Figure 13) or automatically fill it. The result is an XML instance data that can be consequently used as parameter to the access operations.

GetData

The *GetData* operation allows to retrieve:

- the binary data block pointed by the resource as a SOAP with attachments message;
- the preview of the data again as a SOAP with attachments message;
- the URL of the pointed resource to use for subsequent data access.

The parameter *Type* of the request is used to select one of the previous options.

The *Resource Identifier* and the *CIOnnlineResource* parameters are mutually exclusive. They both are used for identifying the resource to download. Moreover, the *AccessParameters* parameter, described below, will be used by the broker only when the *CIOnlineResource* element is submitted.

The *AccessParameters* parameter is an XML document (optional for some resources), obtained after the *GetAccessParameters* operation (the XForms XML instance data) and filled in by the user (through the client graphical interface). This parameter is optional; by the way, a client that wants to take advantage of all the potentialities of the broker is strongly suggested to use this feature (which allows to retrieve the data in the desired format, coordinate system, resolution, etc.).

Name		Type	Description	Obligation
OperationID		String	Identifier of the operation	M
Resource Identifier		String	Identifier of the Resource	C
CIOnlineResource		XML Document	ISO 19139 element describing the	C
	Name	String	Identifier of the resource on the catalog	M
	Protocol	String	URN (or *defacto* standard name) identifying the connection protocol to be used for the access operation	M
	Endpoint	String	URL of the access service for the resource	M
AccessParameters		XML document	Access Parameters filled in by the client (user)	O
Type		Enum= [PREVIEW|DATA|URL]	Type of requested access	M

Table 15 - Parameters of GetData Method in EuroGEOSS Broker Interface

4.6.4 Manager Module

Operation	Description	Binding
GetBrokerCapabilities	Retrieves Catalog Capabilities (e.g. supported interfaces)	HTTP/SOAP
GetActiveConfiguration	Retrieve the configuration currently in use by Broker	HTTP/SOAP
InsertConfiguration	Store a new catalog configuration	HTTP/SOAP
ActivateConfiguration	Activate a stored configuration	HTTP/SOAP
DeleteConfiguration	Remove a configuration	HTTP/SOAP
GetConfigurations	Retrieve the named configuration or a full list	HTTP/SOAP

Table 16 - EuroGEOSS Broker: Manager Interface Operations

The operations belonging to this functional module are available on an endpoint different from the broker endpoint. If the broker endpoint is *http://<BROKER-SERVER>/services/catalog?*, the methods of the manager module can be found at *http://<BROKER-SERVER>/services/manager?*.

The methods of this module are protected with a simple *Username* and *Password* system. These parameters are encoded in the SOAP header according to the ws-security (WSSE) standard (OASIS, 2004).

GetBrokerCapabilities

This operation has no parameters, it returns the catalog capabilities document, containing the supported accessors (types and version) and profilers, to facilitate a subsequent catalog configuration.

GetActiveConfiguration

This operation has no parameters. It returns the activated catalog configuration (encoded as an XML document). The configuration contains the resources federated by the catalog.

InsertConfiguration

The *InsertConfiguration* operation is used to save a configuration in the broker.

Name	Type	Description	Obligation
Broker Configuration	XML Document	This XML document contains all information needed by the broker to set up a new configuration	M

Table 17 - Parameters of InsertConfiguration Operation

This method requires only one parameter: the XML document describing the configuration to insert. Details about the schema of the XML document can be found at EuroGEOSS broker documentation page (EuroGEOSS, 2010).

ActivateConfiguration

This operation can be used to activate a stored configuration, identified by its name (*ID* parameter). Otherwise it is also possible to submit with this request an entire XML document with the configuration to be activated. These two parameters are mutually exclusive.

The activation causes the catalog to change its configuration (e.g. resources being federated or published interfaces).

Name	Type	Description	Obligation
ID	String	Identifier of the configuration to activate	C
Broker Configuration	XML Document	This XML document contains all information needed by the broker to set up a new configuration	C

Table 18 - Parameters of ActivateConfiguration Operation

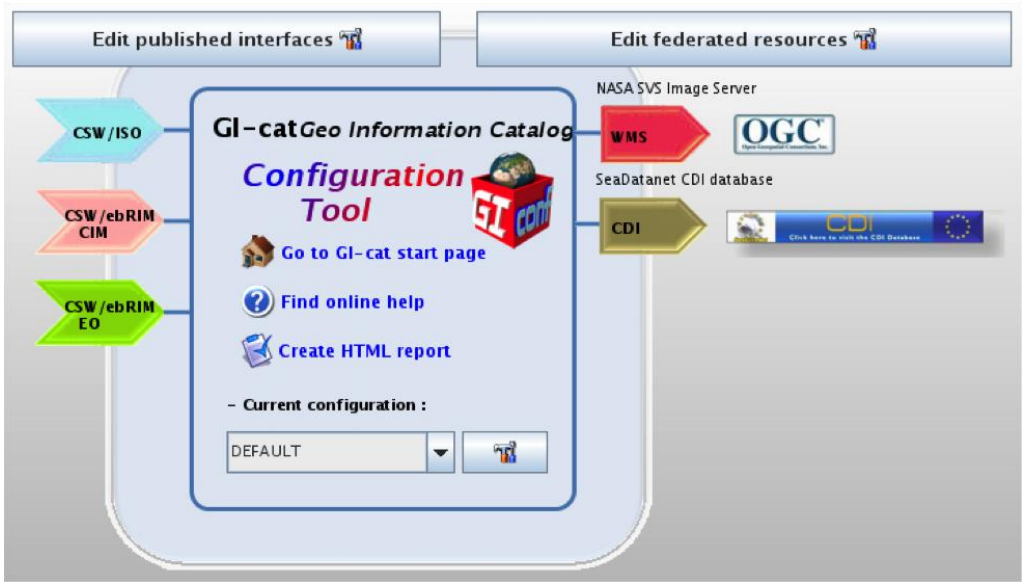

Figure 14 - A graphical representation of the active profile. The profilers (the interfaces) are on the left and the accessing protocols (the resources) on the right. In the bottom part of the window the name of the active configuration is shown.

DeleteConfiguration

This operation is used to remove a stored configuration from the catalog. The only parameter for this method is the *ID* of the configuration to delete.

Name	Type	Description	Obligation
ID	String	Identifier of the configuration to be deleted	M

Table 19 - Parameters of DeleteConfiguration Operation

GetConfigurations

Used to get the list of stored configurations. The client can request a list of configurations (with the *ID* list) of a given *Type* – when *Type* is set to ACTIVE, only one configuration will be returned. When the request doesn't specify any *ID* parameter, all configurations of the requested *Type* are returned.

A client could present the list to the user and make him or her choose a profile to be activated.

Name	Type	Description	Obligation
ID	List<String>	Identifier of the configuration(s) to return	O
Type	Enum: [ALL|ACTIVE]	This XML document contains all information needed by the broker to set up a new configuration	M

Table 20 - Parameters of GetConfigurations Operation

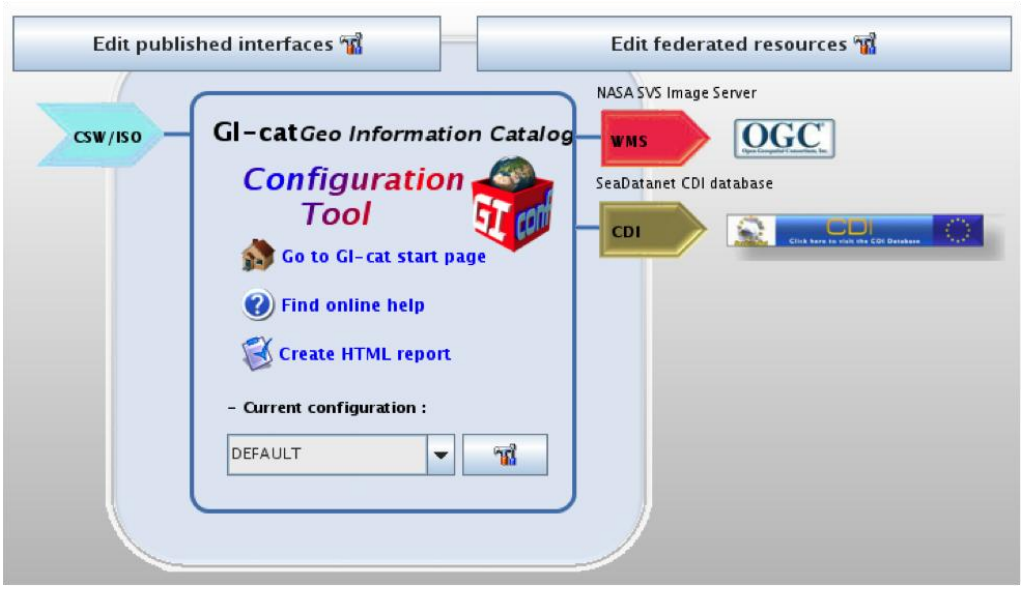

Figure 15 – Graphical representation of the catalog state after the operations of store and activation of the new profile.

5 ACCESS THROUGH AVAILABLE CLIENTS

This Section describes how to access the EuroGEOSS IOC Broker directly through available clients. We describe the GI-go and GeoNetwork clients and will introduce additional clients in further updates of this document.

5.1 GI-go Client

GI-go is a graphical client developed by the ESSI Lab/IMAA-CNR and University of Florence at Prato. Its query engine is based on the GI-cat business logic, allowing discovery and access of geospatial resources across distributed and heterogeneous data sources with a uniform interface based on ISO 19115/19119 application profile.

Taking advantage of the EuroGEOSS broker extended interface, GI-go provides many functionalities such as resource preview and download, visualization of query feedback, and visualization of incremental query results.

5.1.1 Quick user guide

In the next sections a quick user guide is presented, illustrating the main application functionalities. An exhaustive user guide is available from the EuroGEOSS broker documentation page (EuroGEOSS, 2010).

Configuring data sources

The above mentioned data sources, can be configured using the “Configuration Manager” tool as later on described. GI-go can have many source configurations, each correspondent to a user defined set of data sources, and it is distributed with a sample configuration in order to make the application immediately usable.

Configurations can be exported as files, which can later on imported. In order to work with the services that constitute the EuroGEOSS infrastructure, a proper configuration file has been created and published online; following steps explain how to retrieve and import the EuroGEOSS configuration.

1. Open GI-go; the sample configuration is in used

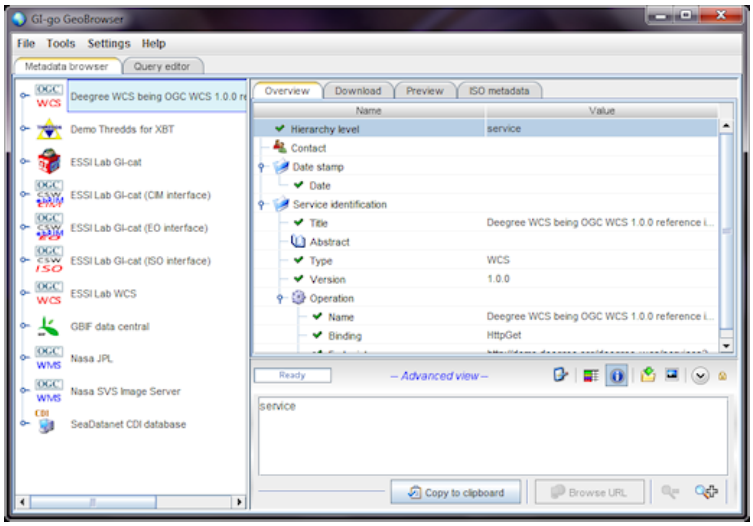

Figure 16 - GI-go Metadata browser tab showing sample configuration

2. Download the EuroGEOSS configuration file here: http://zeus.pin.unifi.it/twiki/pub/GIgo/WebHome/Eurogeoss.gcp

3. In the GI-go toolbar, select Tools -> Configuration manager

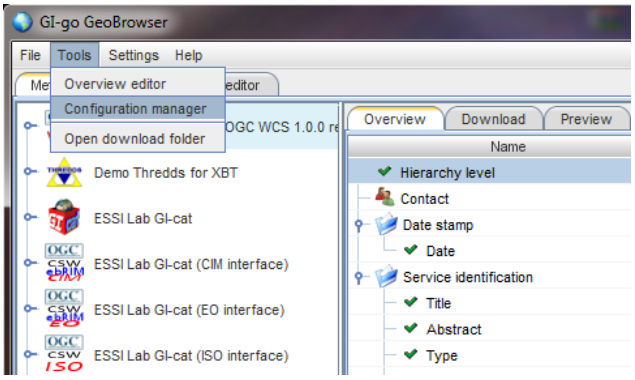

Figure 17 - GI-go Tools toolbar

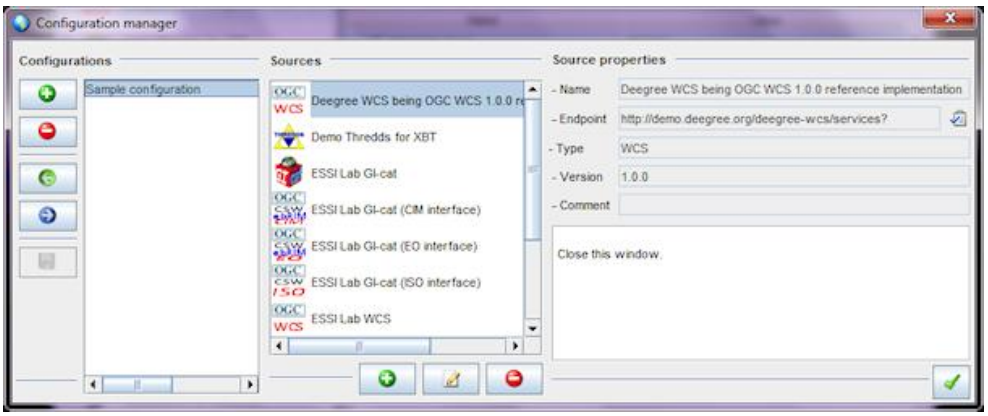

Figure 18 - GI-go Configuration Manager with sample configuration selected

4. Import the downloaded configuration file

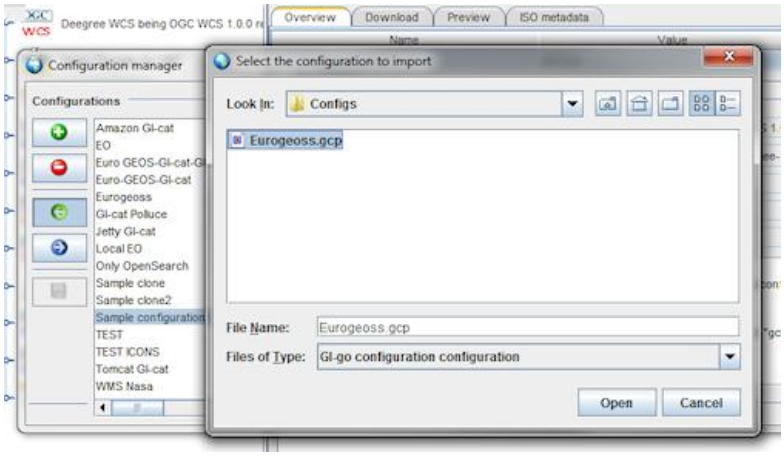

Figure 19 - Importing EuroGEOSS configuration

5. Select EuroGEOSS configuration and close the manager window

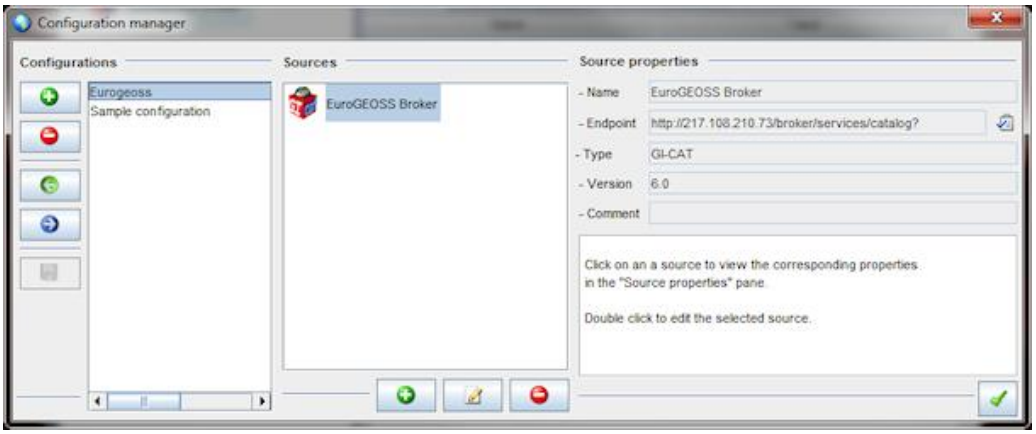

Figure 20 - GI-go Configuration Manager with EuroGEOSS configuration selected

As depicted in the following screenshots, the GI-go graphical user interface provides several tabs, each of which allows to access a particular set of functionalities.

Metadata browsing

On the left side (Figure 16, Figure 21), there is a tree structure which allows to browse metadata describing the selected resource (service, dataset or dataset collection). This feature is implemented using the broker extended interface, in particular the *GetContent* operation.

On the right side, there are some tabs which provide the tools/functionalities described below.

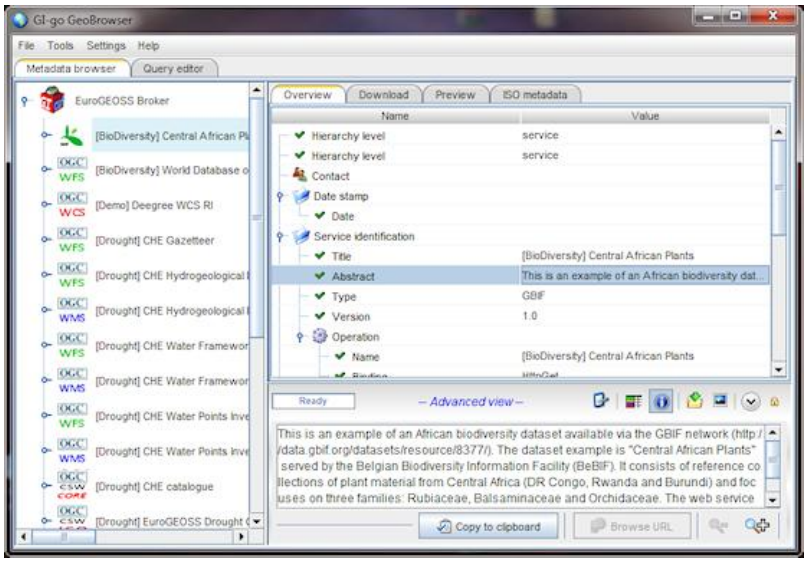

Figure 21 - GI-go Metadata browser tab showing EuroGEOSS configuration

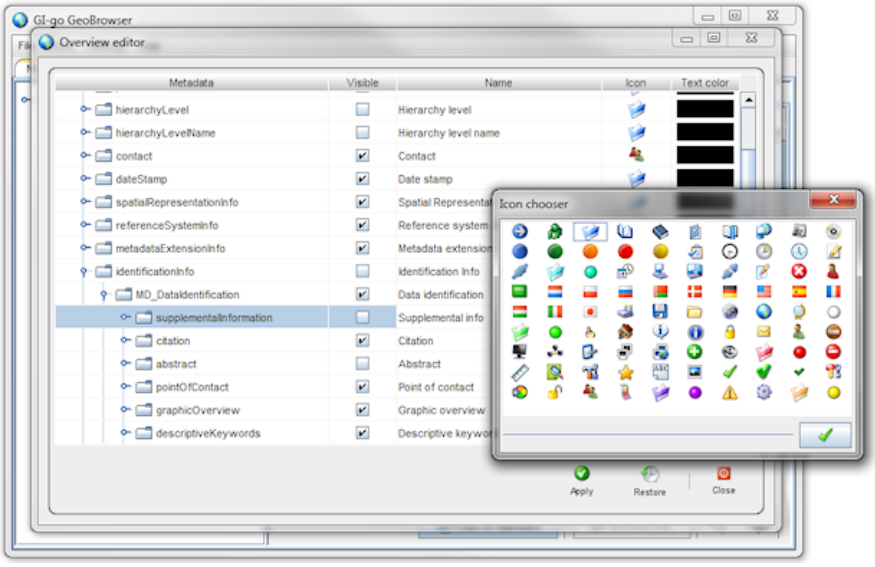

Figure 22 - GI-go Overview editor

The “Overview” tab shows a customizable resource metadata overview. With the “Overview Editor” (available in the menu toolbar Tools -> Overview editor as depicted in Figure 17), the user can select which metadata should be visualized, selecting it from an ISO 19115/19119 predefined set. In order to improve readability, the user can also edit the default metadata name and the rendered text color. Finally, the user can associate selected metadata to an icon, in order to highlight or to better represent the metadata semantic.

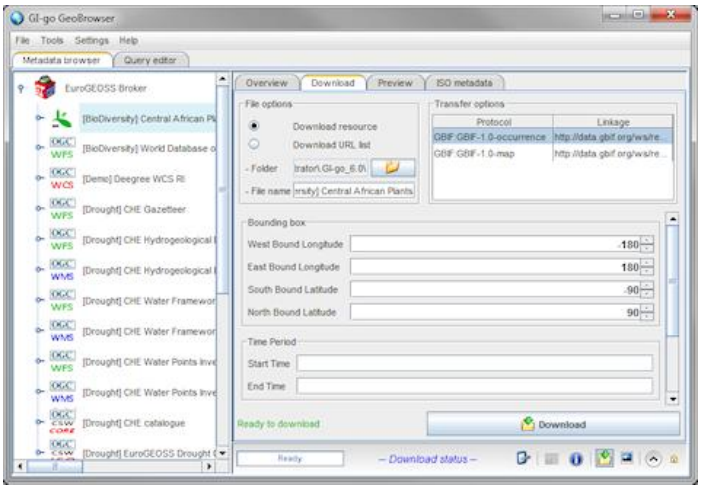

Figure 23 - GI-go Download tab

The “Download” tab allows the download (when available) of the metadata related resource, and provides a form by which to enter several download settings. This features take advantage respectively of the broker extended interface *GetData* and *GetAccessParameters* operations.

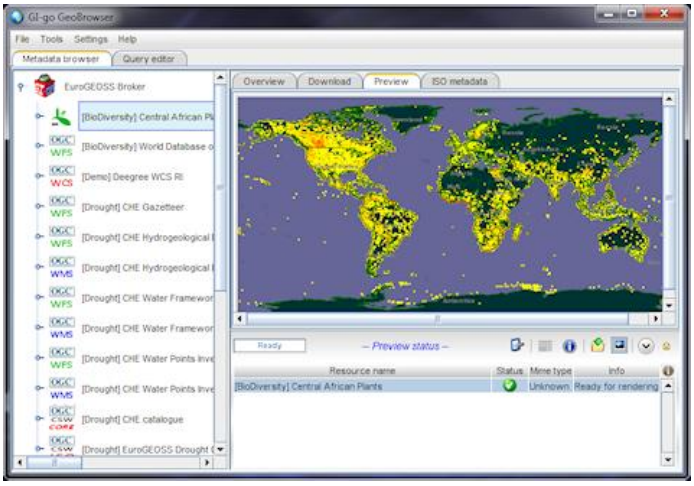

Figure 24 - GI-go Preview tab

The “Preview” tab shows a preview (if available) of the metadata related resource; images and xml files (and in some particular case) HTML pages can be visualized. This feature is based on the broker extended interface preview operation (that is, *GetData* with *Type* parameter set to preview).

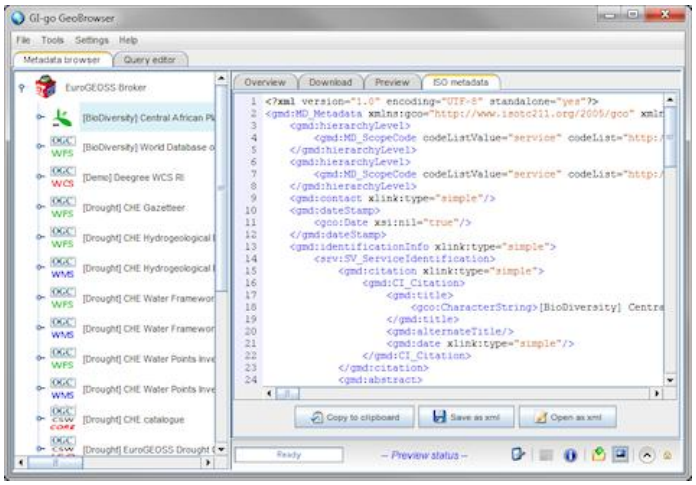

Figure 25 - GI-go ISO metadata tab

The “ISO metadata” tab, shows the “raw” xml resource metadata, and allows to save it in a file or to open it in a temporary file with the default system editor.

Query editing and execution

Figure 26 - GI-go Query editor tab with all criteria disabled

Using the “Query editor” tab, GI-go allows to make queries based on the common search criteria “who”, “where”, “when” and “what” respectively editable by clicking the “Sources”, “Area”, “Time” and “Keywords” buttons. When one of this button is deselected, the correspondent criterion is disabled.

As depicted in Figure 26, initially, no criterion is enabled (all the criteria buttons are deselected). In this case, the query would be executed without constraints, resulting in all the available data of all the available data sources.

Figure 27 - GI-go Query editor tab with all criteria correctly edited

Figure 27 shows the “Query editor” tab with all the criteria correctly edited. Besides the criteria buttons, a light gray icon indicates the criterion is disabled (see Figure 26). A red icon indicates that the correspondent criterion is not correctly selected (for example, the “Sources” criterion is enabled but no source is selected, or the “Time” criterion is enabled and only one time period is selected). A green icon indicates the correct editing of the criterion.

Thanks to the broker query distribution capability, queries can be distributed to one or more (or all) the data sources available in the current configuration. To include a source in the query, the corresponding check box must be selected (see Figure 27). To include all the available sources, user can either select all the check boxes, or (more simply) deselect the “Sources” button (see Figure 26).

The area constraint can be edited both by manually inserting the bounding box coordinates, or by selecting an area using the small world map on the right side of the “Area” pane. A list of predefined regions is also available (see Figure 28).

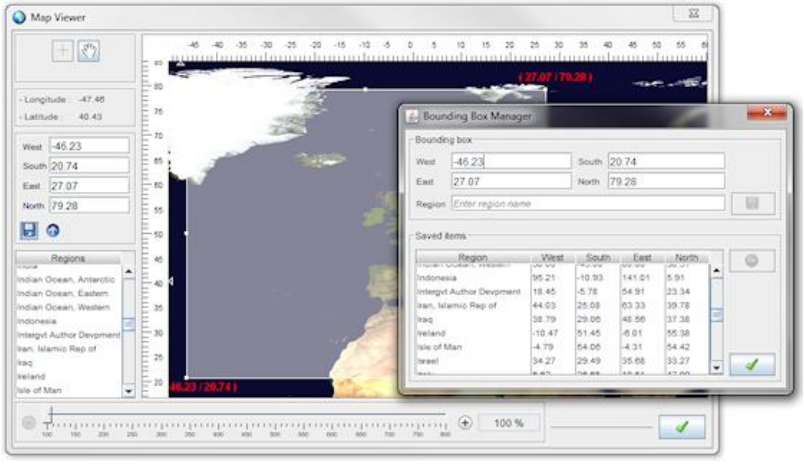

Figure 28 - GI-go Map Viewer

By clicking on the “Advanced” button, the “Map Viewer” tool is showed allowing user to select an area in a big world map with zoom functionality; using the “Bounding Box Manager” the selected area can be saved and added to the predefined ones (see Figure 28).

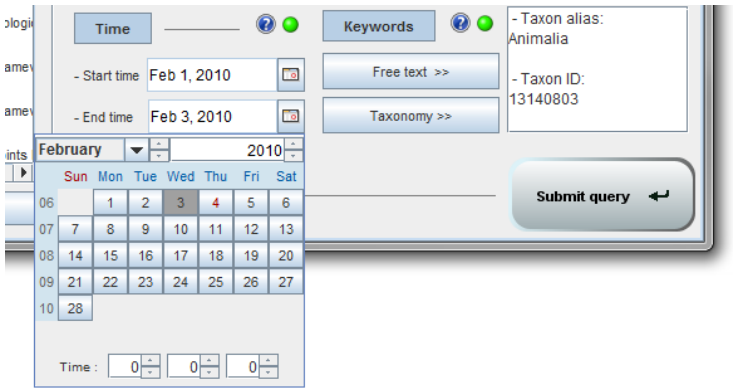

Figure 29 - GI-go time constraint editing

Time constraint can be edited by clicking on the buttons besides the “Start time” and “End time” fields, and selecting the desired date and time in the calendar (see Figure 29).

Figure 30 - GI-go keyword search editor

Keywords can be edited by clicking on the “Free text” button, as depicted Figure 30.

By clicking the “Taxonomy” button (see Figure 30), user can select a keyword choosing it from a tree representing the structure of the GBIF Data Portal.

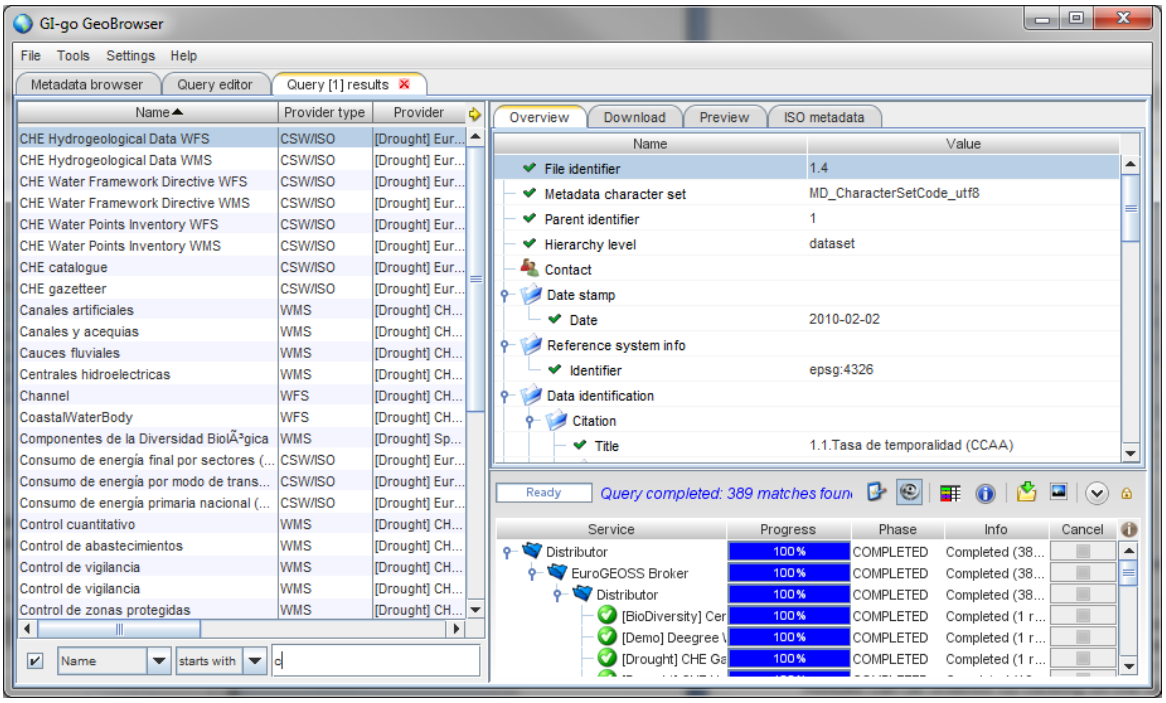

Figure 31 - GI-go query result tab

Once the “Submit query” button is pressed, a new tab is added; this shows the query result as they are returned by the currently queried source (see Figure 31). This feature is based on the broker *Query* operation with the *Synchronous* parameter set to false

This tab provides the same functionalities as the “Metadata browser” tab, but such functionalities are applied to the query results listed in the table placed on the left side.

The query progress of each source as well as query status feedback, are visible in the right-lower pane. Furthermore, the query process of a specified source can be canceled (if the progress is not already 100%) by clicking the correspondent “Cancel” button (see Figure 31). This features uses operations of the Session functional module of the broker extended interface.

Results can be ordered by clicking on the table header, and they can be filtered as depicted in Figure 31.

5.2 GeoNetwork client

GeoNetwork is an open source software designed to improve accessibility to a wide variety of data together with the associated ancillary information (metadata), at different scales and from multidisciplinary sources; data are organized and documented in a standard and consistent way (documentation is available at http://geonetwork-opensource.org/). GeoNetwork implements both the Portal and Catalog components of a Spatial Data Infrastructure (SDI) defined in the OGC Geospatial Portal Reference Architecture.

GeoNetwork provides tools for managing and publishing metadata on spatial data and related services; furthermore, we will see that it can also be used as a particular client for the EuroGEOSS broker. The interoperability between a GeoNetwork instance and the EuroGEOSS broker gives two important results:

- a GUI for executing queries and visualizing results on a browser

- a catalog service that provides functionalities for the metadata management (e.g. metadata insertion/editing/removing) and the harvesting configuration.

5.2.1 EuroGEOSS broker – GeoNetwork configurations

Three types of links between a GeoNetwork instance and the EuroGEOSS broker were experimented. This paragraph deals with two main components: a GeoNetwork instance and the EuroGEOSS broker. GeoNetwork sub components are: a web publisher which offers all the GeoNetwork functionalities through a web browser; a database component to store metadata and a CSW/ISO publisher. Details about the EuroGEOSS broker were described in chapter 2.

Configuration 1 - Federating GeoNetwork in the EuroGEOSS broker

In Figure 32 the first configuration is depicted. The interoperability between GeoNetwork and EuroGEOSS broker is based on the CSW/ISO interface provided by GeoNetwork and used by the EuroGEOSS broker. This configuration allows to retrieve metadata stored in the internal database of GeoNetwork through the CSW/ISO interface published by the Broker.

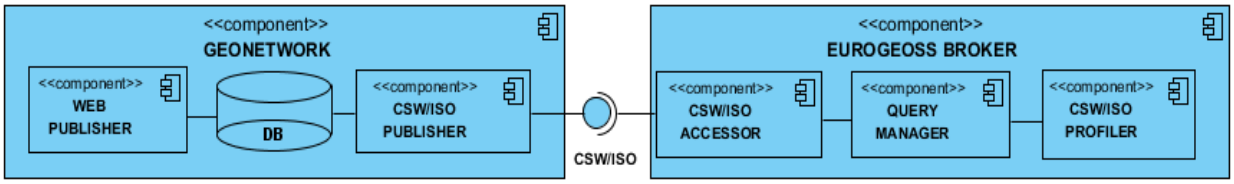

Figure 32 - Federating GeoNetwork in the EuroGEOSS broker

Configuration 2 - Harvesting the EuroGEOSS broker from GeoNetwork

Figure 33 depicts the second configuration between the two components. This configuration inverts the direction of the query flow; in fact, through the CSW/ISO interface, GeoNetwork accesses metadata from the EuroGEOSS broker. It must be underlined that GeoNetwork allows the federation of web data sources for harvesting purposes only; thus in this case queries are not really propagated towards the Broker, but the content of the Broker is periodically harvested (using CSW/ISO interface) and stored in the GeoNetwork database.

The advantage of this configuration is that GeoNetwork can be used as a browser client for the metadata provided by the EuroGEOSS broker, with the possibility to execute queries and manage the metadata trough the graphical interface of GeoNetwork.

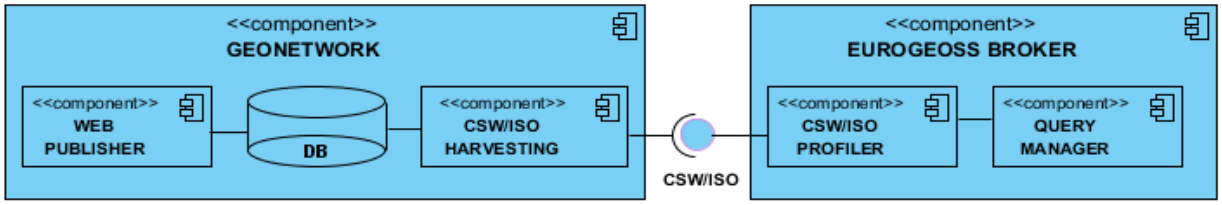

Figure 33 - Harvesting the EuroGEOSS broker from GeoNetwork

Configuration 3 - Modified GeoNetwork

A limitation of configuration 2 is the high harvesting time needed to store all the metadata provided by the EuroGEOSS broker to the internal database.

To address this issue, GeoNetwork has been modified as in Figure 34 in order to allow a real query propagation towards EuroGEOSS Broker (and thus, in turn, to the different types of federated data sources). This avoids the need to store the metadata into the GeoNetwork internal database. The query parameters are set through the graphical interface of GeoNetwork as well as the presentation of results.

A second advantage can be achieved setting up a GeoNetwork catalog among the federated resources in the EuroGEOSS broker instance. Such a configuration allows in turn the EuroGEOSS broker to run queries against the GeoNetwork internal database. This enables both the harvesting and the metadata editor functionalities provided by GeoNetwork and the distributed search functionality of EuroGEOSS broker available in a consistent way through the same graphical interface.

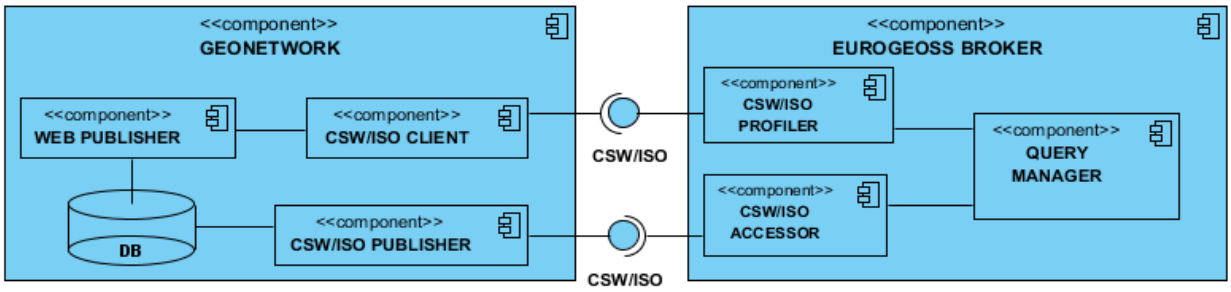

Figure 34 – Modified GeoNetwork Component Diagram

5.2.2 Setting up the EuroGEOSS broker – GeoNetwork configurations

Configuration 1 - Federating GeoNetwork in the EuroGEOSS broker

See section 3.1.

Configuration 2 - Harvesting the EuroGEOSS broker from GeoNetwork

GeoNetwork can be configured trough its web interface. After logging in as an administrator it is possible to access the page of Administrator; in this page it is possible to access services for managing metadata, users and system configuration. Clicking on the link *Harvesting Management* the harvesting main page is opened. This page lists the nodes that have been configured for the harvesting so far (Figure 35).

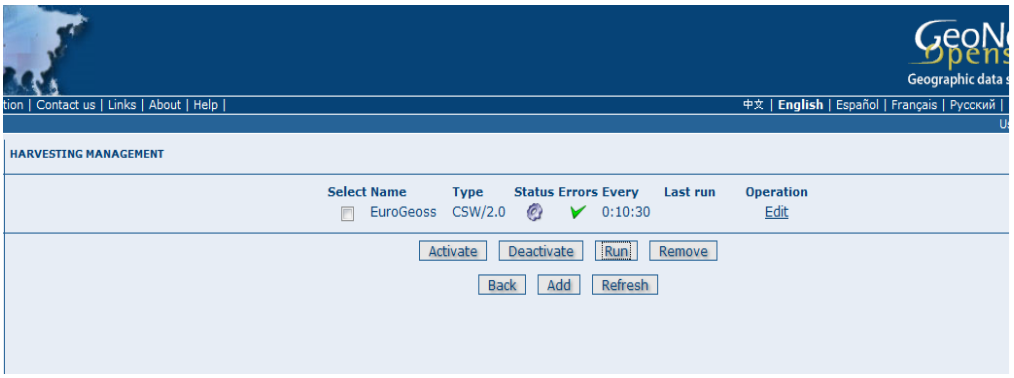

Figure 35 - GeoNetwork harvesting configuration

The *Add* button in the main page allows adding new nodes to be harvested. When creating a new node, several parameters can be set; the main one is the service protocol implemented by the remote server (e.g. OGC WMS,WCS,CSW ISO AP).

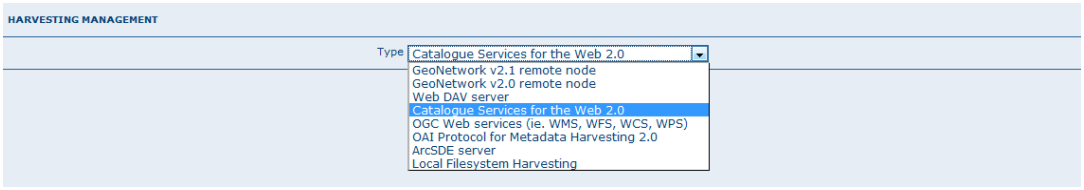

Figure 36 - GeoNetwork select type for harvesting

To configure GeoNetwork in order to harvest the data federated by the EuroGEOSS broker, it is necessary select *Catalogue Services for the Web 2.0* and click the *Add* button. After pressing it, you will reach the page where it is necessary to set several parameters (Figure 37). In particular the *Service URL* field shall be filled I with *GetCapabilities* URL of the Broker CSW/ISO interface:

http://217.108.210.73/broker/services/cswiso?service=CSW&version=2.0.2&request=GetCapabilities

After configuring the harvesting node for EuroGEOSS broker, pressing the *Run* button starts the harvesting process on EuroGEOSS broker.

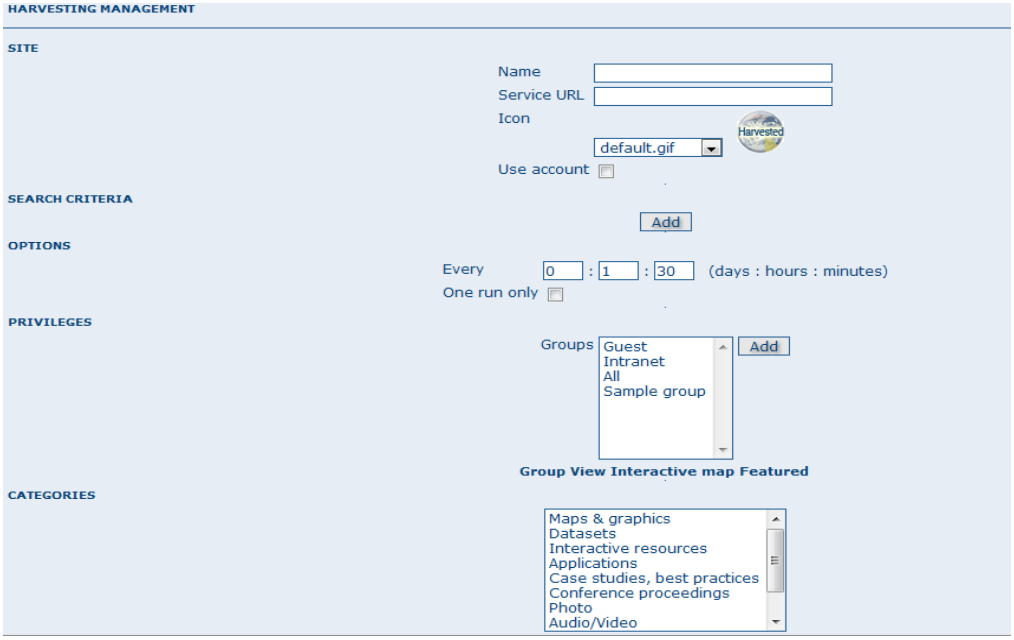

Figure 37 - Page Set Harvesting Node

Configuration 3 - Modified GeoNetwork

The procedure to configure the modified GeoNetwork is similar to the procedure described above. From Administrator page, click on *Configure Broker* to access the *Broker Endpoint Configuration* (Figure 38). This page lists CSW/ISO nodes towards which queries are propagated. It is possible to add and remove a CSW/ISO node. In order to add a new node, only the name and URL of CSW/ISO service parameters are required. To add the CSW/ISO interface of EuroGEOSS broker the following URL shall be used:

http://217.108.210.73/broker/services/cswiso

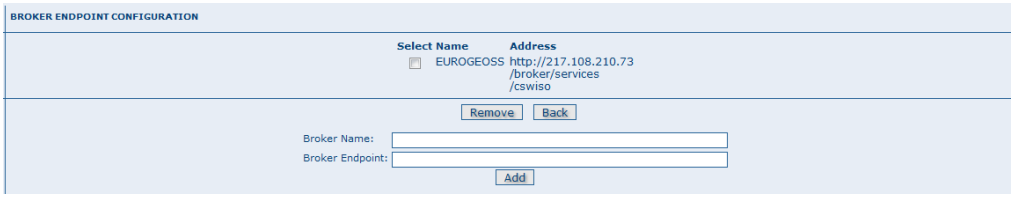

Figure 38 - Broker endpoint Configuration

As previously outlined, the EuroGEOSS broker can federate the GeoNetwork service as well, enabling both the harvesting and the metadata editor functionalities provided by GeoNetwork.

Once configured, through the GeoNetwork home page it is possible to execute a query with or without parameters (such as keyword, bounding box or time extents using advanced search).

Figure 39 depicts the metadata results of a query on EuroGEOSS broker.

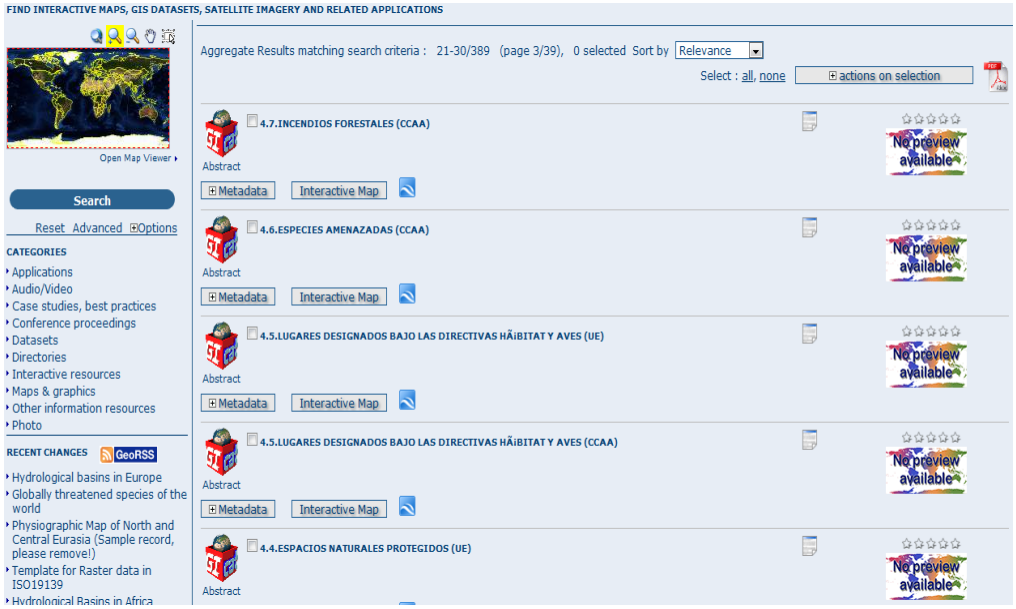

Figure 39 - Metadata results query EuroGEOSS broker

6 REFERENCES

CNR-ESSI Lab, “GI-cat Homepage”, available at http://zeus.pin.unifi.it/cgi-bin/twiki/view/GIcat/ [accessed on 19/02/2010

European Parliament, 2009. Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE)

EuroGEOSS, 2010, “The EuroGEOSS Broker, technical documents”, available at http://www.eurogeoss.eu/

Global Earth Observation, 2005, “10-Year Implementation Plan”, available at: http://earthobservations.org/docs/GEOSS%2010-Year%20Implementation%20Plan%20%28GEO%201000%29.pdf.

International Standard Organization, 2003, “Geographic Information – Metadata”, [ISO:19115].

International Standard Organization, 2005, “Geographic Information – Services”, [ISO:19119].

International Standard Organization, 2007, “Geographic Information – Metadata – XML schema implementation”, [ISO:19119].

OASIS, 2004, “Web Services Security User Name Token Profile”, available at: http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf

Open Geospatial Consortium, 2007core, “OpenGIS catalog Services Specification, version 2.0.2”, [OGC 07-006r1].

Open Geospatial Consortium, 2007iso, “OpenGIS catalog Services Specification 2.0.2 – ISO Metadata Application Profile”, [OGC 07-045].

Open Geospatial Consortium, 2008eo, “EO Products Extension Package for ebRIM (ISO/TS 15000-3) Profile of CSW 2.0 (0.1.9)”, [OGC 06-131r4].

Open Geospatial Consortium, 2007cim, “OGC Cataloguing of ISO Metadata (CIM) Using the ebRIM profile of CS-W (0.1.7)”, [OGC 07-038].

Open Geospatial Consortium, 2008os, “Change Request: OpenSearch Extensions of OGC Catalogues”, [OGC 08-169].

7 APPENDIX

In this Appendix the brokering component architecture description is detailed in terms of separate but interrelated viewpoints, as in the Reference Model for Open Distributed Processing "methodology".

7.1 Information view – The broker data model

The UML diagram depicted in Figure 40 shows the conceptual data model used by the brokering component; it supports cataloging, discovery and access of geospatial data.

The data model is based on the catalog information model described by OGC Catalog Service for Web (CSW) ISO Application Profile (AP) (OGC, 2007iso): the standard used to describe geospatial metadata is the ISO 19115 standard (ISO, 2003), while metadata definitions needed to describe a geoinformation service are specified in the ISO 19119 standard (ISO, 2005).

The base abstract information type managed by the brokering component, is represented by the concept of *GI-Resource* (i.e. GeoInformation resource) that may be optionally associated with a content and described by specific information (*Metadata*) to support discovery, evaluation and use of the information itself.

The *Metadata* abstract concept is realized by the *CoreMetadata* class. The class is taken directly from ISO 19139 specification (ISO, 2007) (the *gmd:MD_Metadata* element); the class defines metadata for geographic data in compliance with the standard. Queries can be performed on properties of *gmd:MD_Metadata,* a basic set of properties is identified to support common and general search criteria (e.g. "who", "what", "when", "where").

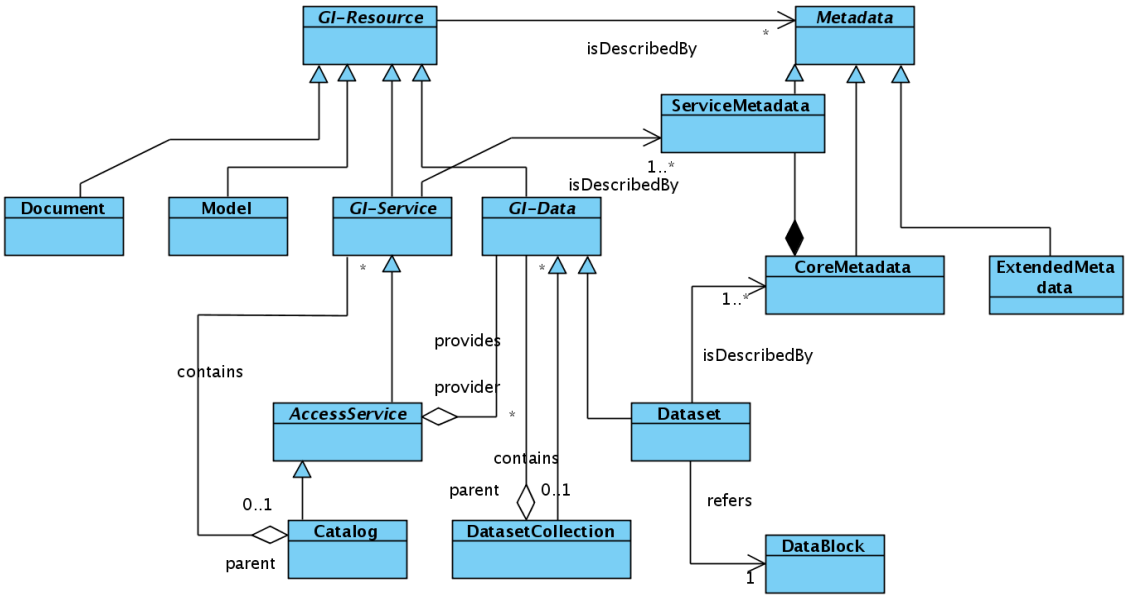

Figure 40 - Data model in use by the brokering component

In addition to *CoreMetadata*, extended *Metadata* objects can also be used to describe GI-Resources, in order to preserve the heterogeneous information coming from data models which are not ISO based or which cannot be mapped to ISO data model (e.g. rim-based data models). In fact, in order to preserve the ebRIM/EO Products semantics (OGC, 2008eo), the extended

metadata element of type *rim:RegistryPackage* can be used. Queries can be performed directly on all available queryable EO Products and results can be presented according to the "EO Products data model" schema.

The "data" type resources are provided by means of an *AccessService* and they may be hierarchically aggregated in *DatasetCollections.*

A *DatasetCollection* may be described by means of *Metadata*, in order to support base search criteria, or just as documentation purpose.

The atomic data unit is the *Dataset,* which is a terminal element in the resource hierarchy (leaf), and described by means of mandatory *ISOMetadata,* in order to support the base search criteria. A *Dataset* contains or references a *DataBlock* of data encoded in some known format.

The resources of type "service" are described by means of mandatory *ServiceMetadata.* A service of type *Catalog* may contain data and other services. This allows to implement both simple catalogs (that are metadata repositories with discovery functionalities) and distributed catalogs which provide distribution and other advanced functionalities (e.g. mediation).

The data model XML schema is published at EuroGEOSS broker documentation page (EuroGEOSS, 2010).

It contains the definitions of XML elements used by the catalog service protocol.

7.2 Computational view

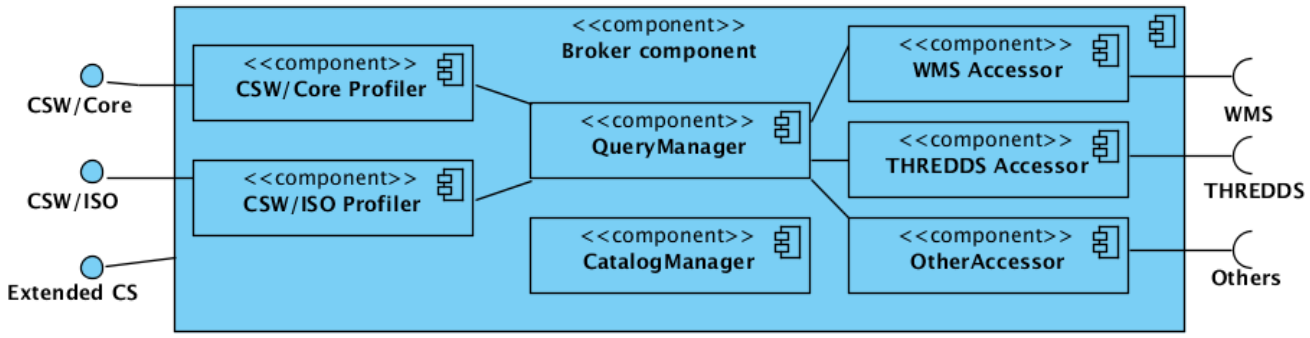

Figure 41 – Example of broker deployment with macro components shown

The broker modular architecture is quite complex but, in this context, we focus only on the components that need to be understood to the aim of developing a client to it.

Two important classes of components play an important role in the broker architecture: the *Profilers* and the *Accessors*. Another important component is the *QueryManager*: a description of the components follows, it will refer to the simple deployment of Figure 41 and the sequence diagram of Figure 42 for the examples.

Profilers expose standard catalog service interfaces and they are also in charge of mediation between external protocols and the internal one. The mediation is usually achieved both at a functional and at a data model level. *Profilers* for CSW/ISO (OGC, 2007iso) and CSW-Core (OGC, 2007core), CSW ebRIM/CIM (OGC, 2007cim) and CSW ebRIM/EO (OGC, 2008eo) have already been developed and tested. Thus, broker functionalities are accessible by standard CSW clients. Other profilers include an OpenSearch interface to enable discovery from a browser.

To exemplify the work done by a CSW/ISO profiler, consider a *GetRecords* operation expressed using a “Filter” on ISO properties (see Figure 42). The CSW/ISO *Profiler* receive the *GetRecords* request and has to translate it into a query expressed on the properties of the broker internal data model. The query execution is at this point delegated to the internal *QueryManager*. At a given point the system will produce a result, for example a set of resources that match the query constraints. The result set is sent back to the CSW/ISO *Profiler*: its task at this point is to send back to the client a correct CSW/ISO *GetRecords* response, by translating the result set expressed in broker data model into a result set for the given application profile (in this case ISO).

The internal component of the broker called *QueryManager* manages queries, in particular takes charge of the query distribution and the collection of the results.

In order to provide distributed search (as well as access capabilities) across many different kinds of services (such as WCS, WMS, and so on), mediation functionalities are central. To the aim of accomplishing the mediation task, the *QueryManager* delegates the queries on specific components which are called *Accessors.* The role of these components is to interact with specific service providers enabling the broker to access and discover data from heterogeneous services. This is done again through data model mapping and protocol mediation.

For example, a WMS *Accessor* is able to carry out a query expressed in the broker internal model by making the needed WMS requests to its WMS server. The response resources coming from the WMS server and matching the initial query are then translated into the broker data model and returned to the *QueryManager* for the collection of results.

The sequence diagram depicted in Figure 42 shows the interaction among the presented components and their main functionalities in the case of a sample query of a CSW client to the EuroGEOSS broker federating a WMS service.

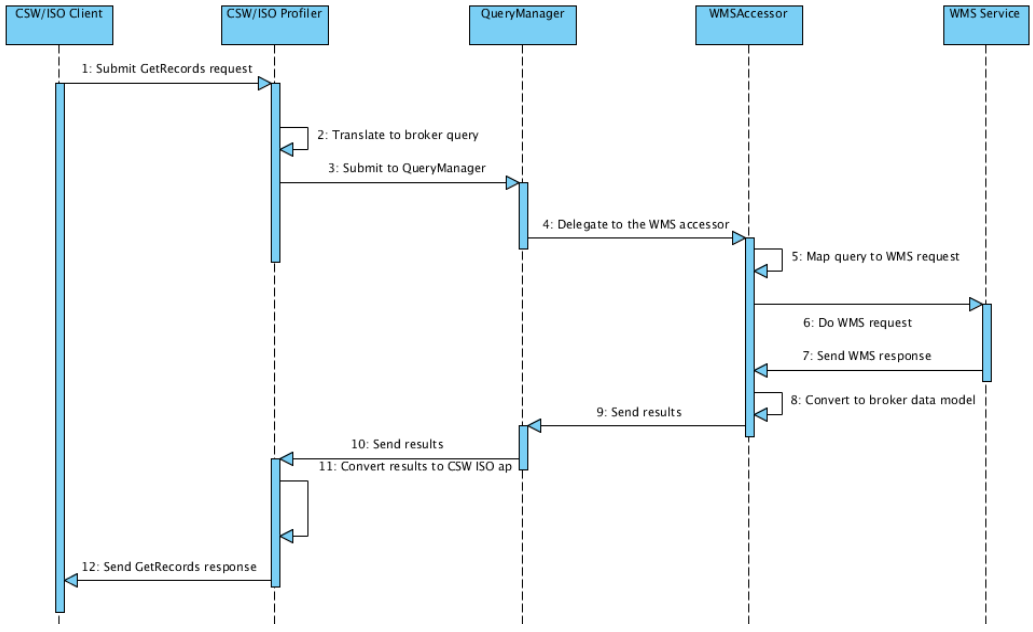

Figure 42 – Example query sequence diagram for a request using the CSW/ISO interface. The Broker of this example federates a single WMS service.

An extended SOAP catalog interface is also published by the broker component. This interface overcomes some limitations of the OGC CSW catalog interface. In particular feedback support (i.e. status monitoring of an ongoing query) and incremental query results presentation capabilities are

provided. Methods for the broker configuration are also present in this interface (e.g. to modify the federated resources).

7.3 Engineering view

The broker developed in EuroGEOSS project is a middleware service that can be placed between one or more clients and one more servers. In a normal context, we may have a CSW/ISO client that queries the broker by using the CSW/ISO interface. The broker distributes the queries to its federated services (performing functional and data model mapping, if needed); then the broker retrieves the resulting resources and presents them back to the client.

Figure 43 shows an example of a deployment scenario. In this example two different clients, a CSW/ISO client and the GI-go GeoBrowser access the EuroGEOSS broker component through either the CSW/ISO interface or the broker extended catalog interface. The configuration tool also uses the extended interface to configure the catalog (e.g. to federate different resources or expose different interfaces).

Two different servers are federated by the broker in this example: a CSW/ISO server and a WMS server.

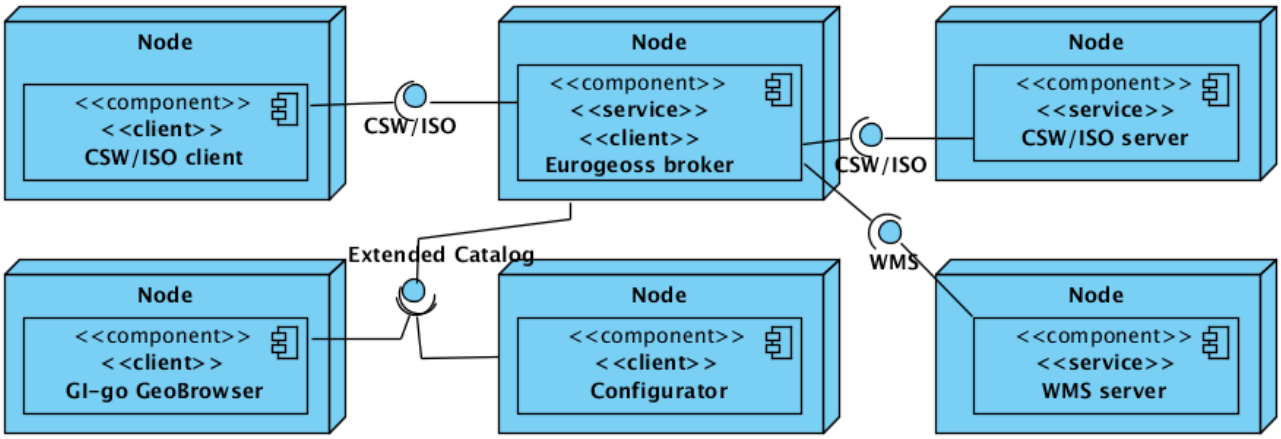

Figure 43 – Example of broker deployment scenario

7.4 Technology view

The EuroGEOSS broker has been developed using Java Standard Edition technology as the target platform (version 1.6), to make multi-platform deployment scenarios possible.

In particular, the broker is a web service that runs on Apache Tomcat and can be installed on Linux, Windows or on a Mac Operating Systems. To implement the SOAP layer JAXWS was used. JAXB was used as XML binding technology.