Software for Distributed Metadata Catalogue Services to Support the EU Portal

Final report

Kristian Senkler, Dr. Uwe Voges, Dr. Udo Einspanier
con terra GmbH

Ioannis Kanellopoulos, Michel Millot, Gianluca Luraschi, Cathal Thorne, Lars Bernard, Paul Smits
European Commission, DG JRC,
Institute for Environment and Sustainability
Software for Distributed Metadata Catalogue Services to Support the EU Portal

Final report

Kristian Senkler, Dr. Uwe Voges, Dr. Udo Einspanier, con terra GmbH

Ioannis Kanellopoulos, Michel Millot, Gianluca Luraschi, Cathal Thorne, Lars Bernard, Paul Smits
European Commission, DG JRC, Institute for Environment and Sustainability

Version : 1.1
Status : Final
Last revision : 28/8/2006
Mission Statement of the Institute for Environment and Sustainability

The mission of the Institute for Environment and Sustainability is to provide scientific and technical support to the European Union’s policies for protecting the environment and the EU strategy for sustainable development.

Contact Information

Ioannis Kanellopoulos
European Commission
Joint Research Centre (DG JRC)
Institute for Environment and Sustainability (IES)
Spatial Data Infrastructures Unit
I-21020 Ispra (VA), Italy
Tel: +39 0332 786491, Fax: +39 0332786325
Ioannis.Kanellopoulos@jrc.it

http://ies.jrc.ec.eu.int
http://www.jrc.cec.eu.int

Kristian Senkler
con terra GmbH
Martin-Luther-King-Weg 24
48155 Münster, Germany
Tel: +49 (0)251 7474-408, Fax: +49 (0)251 7474-100
senkler@conterra.de

Cover: Ramón Sánchez Pérez de Lara

Legal Notice
The contents of this document do not necessarily reflect the official opinion of the European Commission or the European Union Institutions. Neither the European Commission nor any person acting on behalf of the Commission is responsible for the use that might be made of the information contained in this production.

A great deal of additional information on the European Union is available on the Internet. It can be accessed through the Europa Server http://europa.eu.

EUR 22337 EN

ISSN 1018-5593

Luxembourg: Office for Official Publications of the European Communities

© European Communities, 2006

Reproduction is authorized provided the source is acknowledged

Printed in Italy
Executive summary

The main goal of this study was to support the development of the prototype EU geoportal to access distributed metadata catalogues and search for metadata in a harmonized and interoperable way. This includes a number of software components, all of them off-the-shelf products, listed as follows:

- Catalogue service: a centralized broker (itself a catalogue service) that executes a query coming from a client, distributes the query to a known set of federated catalogue services, cumulates the results returned and gives this response back to the client.
- Service monitor: a component to monitor the quality of services in an SDI
- Security manager: a component that handles security issues within an SDI
- Map client: a component to access standardized web map services and provide additional navigation and query tools to the user.

The main challenge is to set up the catalogue service as a broker to support distributed search against federated catalogues. Furthermore, the integration of catalogue broker and the remaining software components into the existing EU portal software is crucial. In addition to an appropriate federated search mechanism, a concept for the management of federated catalogue service is developed and implemented in the broker component.

The study is tightly aligned with the work of the INSPIRE drafting teams (DT) on metadata and networks services. The results of this study provide useful feedback in the development of the INSPIRE draft implementing rules. Thus, the study is meant to be a test case for two reasons both the rules defined by the INSPIRE DTs so far and the degree of interoperability of existing catalogue specifications.

The basic assumption is determined by the concept of distributed catalogue services defined by the OGC base specification CS 2.0. This concept provides a means to derive specific profiles of catalogue service specifications from the base specification whereas certain aspects have to be supported by the derived profiles (like query language, queryables, bindings, minimum Dublin core information model, etc.). Catalogue services based on a profile should then be able to interoperate on the level of that base specification.

Along with the base specification two existing profiles were considered: one based on ebRIM information model (a more general model coming defined by OASIS) and the ISO profile (a more specific model based on ISO 19115 and ISO 19119).

Many catalogues from different European member states contributed their catalogue implementations and knowledge to the study. In the end, two ISO based catalogue service implementations, two ebRIM and two OGC core catalogues (which implement the plain base specification) were integrated.
Specific interoperability tests were developed to test how the considered catalogue services support the underlying specification and, as a matter of concept, the base specification.

The interoperability tests demonstrate that specifications are not adequately supported by the implementations: none of the catalogue services could be queried by the catalogue service broker without the development of special adaptors.

An adaptor is a filter that is plugged between the broker and the target catalogue service to translate the request to the federated catalogue service and the response back to the broker in a way that a standardized communication could be established. A number of adaptors are developed to catch up with the implementation specific inconsistencies of the considered catalogue services.

The reasons for this are, in most of the cases, too many degrees of freedom in the underlying specifications. This leads to different interpretations of the specification and, in the end, in non interoperable catalogue service implementations. This is true for both the base specification and the derived profiles (ebRIM and ISO). In short, the following major problems were identified:

- Many aspects of the catalogue specifications are ambiguously defined.

- Concerning the underlying information models it is very erroneous to translate one model to the other (semantically and syntactically) since no standardized mapping rules exist.

- The concept of federated search should be better documented and integrated into the specifications: what it means, how it works, etc.

The problems identified in the study may be overcome if the existing specifications are amended by the suggestions put forward in this report. Therefore the results of the study are brought to the relevant standardization bodies and consortiums to refine upcoming versions of catalogue specification. On top of that the results have an important impact on the development of the draft implementing rules for INSPIRE and will be considered mainly in the work of the metadata and network services drafting teams.
# Table of Contents

1 **Introduction**  
   1.1 Objectives  
   1.2 Relation to INSPIRE  

2 **Relevant specifications**  

3 **System architecture**  
   3.1 General overview  
   3.2 Catalogue service broker  
   3.3 Catalogue service registry – server side  
   3.4 Catalogue service registry – client side  
   3.5 Catalogue service broker – server side  
      3.5.1 Overview  
      3.5.2 Information model of the broker  
      3.5.3 Decisions  
      3.5.4 Mapping core search properties  
      3.5.5 Response properties  
      3.5.6 Interfaces and bindings of the broker  
      3.5.7 Distributed search: real-time search vs. catalogue harvesting  
         3.5.7.1 Real-time distributed search  
         3.5.7.2 Catalogue harvesting  
         3.5.7.3 Resource harvesting  
      3.5.8 Integration strategies for federated catalogues  
         3.5.8.1 ISO-Profile-based OGC Catalogues  
         3.5.8.2 ebRIM- and other non ISO-Profile OGC Catalogues  
         3.5.8.3 OGC CS 2.0 Common Profile catalogues  
      3.5.9 Integration of federated catalogues via “adaptors”  
   3.6 Catalogue service broker – client side  
   3.7 Service Monitor  
   3.8 Gazetteer  
   3.9 Map Client  

4 **Catalogue service integration**  
   4.1 Integration of catalogue services  
      4.1.1 ebRIM Catalogue service A
4.1.1 Expected behaviour
4.1.2 Interoperability test
4.1.3 Problems
4.1.4 Solution

4.1.2 ebRIM Catalogue service B
4.1.2.1 Expected behaviour
4.1.2.2 Interoperability test
4.1.2.3 Problems
4.1.2.4 Solution

4.1.3 OGC Core catalogue service A
4.1.3.1 Expected behaviour
4.1.3.2 Interoperability test
4.1.3.3 Problems
4.1.3.4 Solution

4.1.4 OGC Core catalogue service B
4.1.4.1 Expected behaviour
4.1.4.2 Interoperability test
4.1.4.3 Problems
4.1.4.4 Solution

4.1.5 ISO catalogue service A
4.1.5.1 Expected behaviour
4.1.5.2 Interoperability test
4.1.5.3 Problems
4.1.5.4 Solution

4.1.6 ISO catalogue service B
4.1.6.1 Expected behaviour
4.1.6.2 Interoperability test
4.1.6.3 Problems
4.1.6.4 Solution

5 Conclusions and Recommendations

5.1 Shortcomings of the OGC specifications
5.1.1 Shortcomings of the OGC CSW 2.0 base specification
5.1.1.1 Discovery operations
5.1.1.2 Transactional operations
5.1.1.3 Federated search

5.2 Shortcomings of the information model
1 Introduction

This document is the final report of the study. Any relevant technical aspect needed to fulfil the requirements of the study (e.g. access to distributed catalogues; accessibility of catalogues; adaptors; known bugs and solutions) are described in detail.

The main focus of this report is on the evaluation of existing catalogue service instances against the underlying specifications and the integration of these catalogue services into the EU Portal. Further components considered in the study are described briefly.

The document is divided into three sections organized as follows:

- **System architecture**: the general architecture is described. Each component is considered and specified.

- **Catalogue service integration**: this is the main part of the document. Each catalogue considered during the study is described technically. The shortcomings of the implementations and the underlying specifications are described. Furthermore it is shown how to integrate each of these catalogue services in the overall system.

- **Conclusions and recommendations**: This part summarizes the conclusions and the recommendations

1.1 Objectives

The aim of this study is to provide JRC with software tools for implementing certain aspects of meta-information management, web service security and SDI quality management.

Both the software and con terra’s hands-on experience in this field shall be made available for the JRC to allow it to explore state-of-the-art technology and its implications.

This work will provide support for the technical developments within INSPIRE and in particular towards the development of the European Community geoportal.

The overall objectives of the study are to set up and test an environment, where a centralized catalogue service broker is integrated into the INSPIRE geoportal to provide a harmonized access to federated catalogues throughout Europe. These catalogues may provide different service interfaces and different information models for their metadata; it is the goal to realize the best degree of interoperability in the catalogue service network and to report shortcomings and advantages of up to date software, implementations and specifications.
1.2 Relation to INSPIRE

The work in this study is strongly related to the work within the INSPIRE Drafting teams, especially with Network services and Metadata. The software developed and deployed during this study makes use of the current results of the Drafting teams and acts as a testbed by implementing these intermediate results.

The results of the implementation go back to the drafting team and will be evaluated and considered in upcoming versions of the draft implementing rules.

2 Relevant specifications

The following specifications and international standards have been considered during the study:

- ISO 19115: 2003 and derived profiles for the description of spatial data and applications
- ISO 19119: 2005 and derived profiles for the description of geo-services
- ISO 19139 as XML Implementation schema v0.9
- Dublin Core (ISO 15836:2003) with RDF/XML encoding
- OGC CSW 2.0 (OGC Document 04-021r3) and referenced OGC specifications
- OGC CAT 2.0 SRW as a stateless variant of Z39.50 for web service implementation with SOAP binding
- OGC ISO19115 / ISO19119 Application Profile (AP) for CSW 2.0 (OGC Document 04-038r2)
- INSPIRE Implementation Rules (current state of discussions)

To fulfil the requirements of the proposal, the following components of con terra’s sdi.suite are used:

- terraCatalog – for the collection, management and exploitation of meta-information
- serviceMonitor – quality management for the service network
- mapClient – provides access to map and data services
- securityManager – for managing users and access rights in spatial data infrastructures
- terrainServer – for 3D rendering scenes
3 System architecture

3.1 General overview

The following section gives an overview about the system architecture that is applied in the study.

The overall system has a typical multi-tier architecture. This architecture is depicted in Figure 1. The presentation, business and service tiers are discussed in more detail in the technical specifications of the study.

![Figure 1: System architecture](image-url)
3.2 Catalogue service broker

The catalogue service broker is the major component of the system when it comes to distributed search. The idea of a distributed search is to start a search from a known node and to search transitively as many catalogues as possible with the same set of queryables. The Catalogue Service Broker is the starting point for a distributed search. Therefore the main focus of the Catalogue Service Broker is to integrate one or more external (remote or federated) catalogues in its distributed search (BrokerService).

It merges the results coming from the remote OGC CAT 2.0 Catalogues as well as from Catalogues with differing information models / interfaces. The remote catalogues are accessed via specific adapters/connectors. Merging of the results includes:

- transformation of the records coming from specific catalogues into the broker’s result set schema
- aggregation of the different result sets
- provision of the results in a requested block size, starting with a specific result entry (e.g. entry 21)\(^1\)

In addition to “real-time” distributed search the Catalogue Service Broker is also capable to store the results coming from remote catalogues into the local database (Catalogue-Harvesting / Caching) and to process a logical distributed search based on the data of a remote catalogue stored in the local database. Such local mirrors can be executed in nightly batch-processes. This functionality may be enabled or disabled by the administrator.

The broker does not maintain any metadata by itself, except metadata describing the remote catalogues. The catalogues which will be used in a distributed search may be selected by the results of a search on these metadata.

The broker does not only access remote catalogues with OGC conformant interfaces but also provides the following OGC conformant interfaces by itself:

- OGC Web Catalogue Service (CAT 2.0.0), supporting the Dublin Core based information model
- OGC CAT 2.0 AP ISO19115/19119 (DE-Profile 1.0.1), supporting the ISO19115/19119 based information model

---

\(^1\) In most cases a client sends a filter query to the broker and indicates that the result should start at a specific index number and should include a maximum amount of records. Google does it the same: a user formulates a search term, presses enter, the presents result 1 to 10. If the user clicks on the “next” link in the browser, the same query is sent again but with an indicator, that the client now requests the next 10 result starting from index 11.

In a distributed environment, this is even more complicated, as the next, let’s say 10 records a client request, are originated from 2 federated catalogues. So the broker has to take care of this algorithm to support the client in flipping through the pages of a result set.
The information model of the Catalogue Service Broker is based on the information model of the ISO19115/ISO19119 Application Profile (DE-Profile 1.0.1) of the OpenGIS® Catalogue Services Specification (CS) 2.0.0. As stated in CS 2.0.0, the Dublin Core (Version 1.1:ISO Standard 15836-2003, February 2003) base profile that provides a basic set of information objects has to be supported by each catalogue instance, hence by the Catalogue Service Broker.

3.3 Catalogue service registry – server side

This section describes how distributed catalogue service instances are defined in the broker software (terraCatalog 2.1). The following assumptions are made for a distributed catalogue:

- The distributed catalogue must be accessible via the HTTP protocol
- The distributed catalogue could be compliant to the OGC CS-W 2.0 ISO-Profile
- If no AP ISO Profile is implemented, it must be compliant to the OGCCORE specification of the OGC CAT 2.0, e.g. a direct implementation or another profile implementation

Every distributed catalogue (used for a real-time online search or for a catalogue harvesting process) has to be described in the catalogue registry including the following information:

- All necessary technical information to get access to a distributed catalogue. To register a catalogue there must be an applicable adapter. A set of special adapters for some of the catalogues described in chapter 3.5.9 were developed in this study.
- Metadata for the distributed catalogue service. Because a catalogue service is a web service, it must be described by service metadata (ISO 19119).

All information about the federated catalogues is stored in the database of the broker. This includes the technical description and the metadata for the description of a distributed catalogue. There is a 1:1 relationship between these two information areas. The technical information includes:

- URL – URL of the distributed catalogue (Discovery Web Service)
- Name – Name of distributed catalogue
- Caching (Harvesting) – Is harvesting required for the catalogue service? Runtime access to remote catalogue interface (Caching: 0), or Harvesting of the metadata content of a remote catalogue service and store in the local metadata base without loosing the name of the source (Caching: 1)
- Caching frequency (Harvesting Interval) – Scheduler controller
- Binding – Selection of a supported protocol binding, e.g. POST
• Profile – Name of the adaptor by which the catalogue can be accessed

• Pipeline – if the adaptor in use is based on the terraCatalog’s XSLT Adaptor Concept, there must be defined a pipeline, describing which xslt-Stylesheets must be used for transforming the requests as well as transforming the results

• Information model – Supported information model of catalogue: ISO-model or OGCCORE (only)

• Type (must be ‘distributed’)

• Timeout – Setting of the timeout interval for distributed catalogue service enquiries

• Status – Activate/deactivate distributed search for catalogue service

These entries are managed by the administrator tool. With this tool, the administrator can insert or update the required information to access a distributed catalogue for an online search or for the catalogue harvesting process in the catalogue registry.

3.4 Catalogue service registry – client side

On the client side an administrator of the system can manage the federated catalogues accessible by the catalogue service broker. This is done by two tasks:

• Administer the technical information of the federated catalogue. These are:
  
  o Name
  
  o URL
  
  o Harvesting (cached/not cached)
  
  o Harvesting interval
  
  o Binding
  
  o Profile
  
  o Information model
  
  o Pipeline
  
  o Timeout (in ms)
  
  o Status
  
  o CSW Metadata set
• In addition, the administrator has to describe the federated catalogue by ISO 19119 metadata. This metadata record is linked to the technical information by using the file identifier of the metadata record. This description is needed to enable a user of the system to query for federated catalogues in the local repository to be considered in the distributed search.

The following figure shows a screenshot of the administration interface.

![Figure 2: Administration of federated catalogues](image)

After the federated catalogue has been published to the system, any user of the client can use this catalogue as a federated catalogue in her query. To do so, a user has to select the catalogues she wants to consider during the search. This can be done by “Select catalogues” from the menu. A detailed description of this workflow can be found in the technical manual of the catalogue service broker.
3.5 Catalogue service broker – server side

3.5.1 Overview

The Catalogue Service Broker is the starting point for a distributed search. Therefore the main focus of the Catalogue Service Broker is the integration of one or more external (remote or federated) catalogues in its distributed search (BrokerService). Figure 4 illustrates the concept of the Catalogue Service Broker.

It merges the results coming from the remote OGC CAT 2.0 Catalogues as well as from Catalogues implementing the ISO- or the ebRIM Profile of the OGC CAT 2.0 specification. The remote catalogues are accessed via specific adaptors/connectors if their implementation differs from that of the standard.

Merging of the results includes:

- Selection of the usable results coming from remote catalogues
- integration of these records into the brokers result set
- provision of the results in a requested block sizes, starting with a specific result entry (e.g. entry 21)

![Figure 3: Distributed search (domain) based on OGC CAT 2.0 Specification](image-url)
Additionally to “real-time” distributed search the Catalogue Service Broker is also capable to store the results coming from remote catalogues into the local database (Catalogue-Harvesting / Caching) and to process a logical distributed search based on the data of a remote catalogue stored in the local database. Such local mirrors can be executed in nightly batch-processes.

The broker does not maintain any metadata by itself, except metadata coming from the harvesting of single metadata sets or metadata describing the remote catalogues of the catalogue registry. Therefore, metadata provided by a remote server in a “real-time” distributed search are not permanently stored on disk, but are kept in-memory for the sole purpose of the display of the results to the user. The catalogues which are used in a distributed search may be selected by the results of a search on these metadata.

3.5.2 Information model of the broker

The information model of the Catalogue Service Broker is based on the information model of the ISO19115/ISO19119 Application Profile (DE-Profile 1.0.1) of the OpenGIS® Catalogue Services Specification (CS) 2.0.0. As stated in OGC CS 2.0.0, the Dublin Core (Version 1.1:ISO Standard 15836-2003, February 2003) base profile that provides a basic set of information objects has to be supported by each catalogue instance, hence by the Catalogue Service Broker.

As assumed in the CS 2.0.0 ISO19115/ISO19119 Application Profile the Catalogue Service Broker supports the XML based encoding (OGC Filter Encoding 1.0.0) of the OGC_Common Catalogue Query Language (CQL) (see OGC CS 2.0 specification). This includes:

- support for all comparison operators
- support for all logical operators
- support for the following expressions:
  - property name
  - literal
- support for the following spatial operators\(^2\):
  - Intersects
  - Disjoint
  - BBOX

\(^2\) all spatial operators support geometries in WGS84
In addition, the plain textual encoding of CQL version 2.0 is supported. But most of the remote catalogues will not have implemented the textual encoding, so that this encoding does not play an important role.

The Catalogue Service Broker supports the core queryables as defined in the OGC catalogue specification CS 2.0 (by name) as well as nearly all additional queryables as defined in the CS 2.0.0 ISO19115/ISO19119 Application Profile.

If a query against the Catalogue Service Broker includes queryables as defined in the CS 2.0.0 ISO19115/ISO19119 Application Profile, then remote catalogues supporting only Dublin Core OGC catalogue specification base specification will not be incorporated in the distributed search.

The Catalogue Service Broker supports all result sets as defined in the OGC catalogue specification CS 2.0 (based on the Dublin Core Metadata Set) as well as all result sets as defined in the CS 2.0.0 ISO19115/ISO19119 Application Profile.

If a query against the Catalogue Service Broker should return a result set as defined in the CS 2.0.0 ISO19115/ISO19119 Application Profile, then remote catalogues supporting only Dublin Core OGC catalogue specification base specification will not be incorporated in the distributed search.

The Catalogue Service Broker supports the result set encoding as defined in the OGC catalogue specification CS 2.0 as well as the ISO19139 based result set encoding as defined in the CS 2.0.0 ISO19115/ISO19119 Application Profile (DE-Profile 1.0.1).

### 3.5.3 Decisions

Concerning the realization of the catalogue service broker within the study, the following decisions have been made:

1. Federated catalogue services are distinguished by the information model they provide. Three different types are recognized:
   - CSW Based catalogues: Catalogue services based on the OGCCORE specification (CAT 2.0)
   - ebRIM catalogues: Catalogue services based on the ebRIM profile of OGC CAT 2.0
   - ISO AP catalogues: Catalogue services based on the ISO profile of OGC CAT 2.0

   Catalogue services that fall into the first two categories are accessed on OGCCORE level, which means that only Dublin Core encoded metadata elements are found.
Catalogue services that fall into the last category are accessed on ISO Profile level, which means that the additional queryables and response properties defined by that profile are considered.

2. Four adapters were realized with the catalogue service broker to access the federated catalogue services considered in the study:
   - Adapter for CSW ISO Application Profile
   - Adapter for ebRIM catalogue service
   - Adapter for Dublin Core Catalogue services
   - Adapter for ArcIMS metadata service with ESRI CS-W Connector

   All instances of Dublin Core catalogues are queried with OGCSUMMARY, as OGCBRIEF only return the identifier of the metadata set (too few information) and OGCFULL might result in a proprietary information model as being the payload of the response message. This is true for an ArcIMS service with the ESRI CS-W Connector. OGCSUMMARY is seen as the greatest denominator.

3.5.4 Mapping core search properties

Table 1 lists the relevant queryables from the CS-W 2.0 specification with regard to OGCCORE queryables and additional ISO queryables. It shows how to map all conceptual elements defined by article 18(2) of the INSPIRE proposal for a directive to both OGCCORE queryables and additional ISO queryables defined by the CS-W ISO 19115/19119 application profile.

Elements that have no direct counterpart concerning the OGCCORE queryables are mapped to AnyText. By doing so, a user might at least find something that matches to a query against the catalogue.
<table>
<thead>
<tr>
<th>No.</th>
<th>Article 18 (2)</th>
<th>OGCCCORE queryable</th>
<th>Additional ISO queryable</th>
<th>ISO Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Keywords</td>
<td>Subject</td>
<td>Subject</td>
<td>MD_Metadata.identificationInfo. _MD_Identification. descriptiveKeywords. MD_Keywords.keyword [0..*]</td>
</tr>
<tr>
<td>2</td>
<td>classification of spatial data and services</td>
<td>AnyText</td>
<td>TopicCategory</td>
<td>MD_Metadata.identificationInfo. MD_DataIdentification. topicCategory [0..*]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ServiceType</td>
<td>MD_Metadata.identificationInfo. SV_ServiceIdentification. serviceType</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HierarchyLevelName</td>
<td>MD_Metadata. hierarchyLevelName[0..*]</td>
</tr>
<tr>
<td>3</td>
<td>spatial data quality and accuracy</td>
<td>AnyText</td>
<td>Denominator</td>
<td>MD_Metadata.identificationInfo. MD_DataIdentification. spatialResolution. MD_Resolution. equivalentScale. MD_RepresentativeFraction. denominator</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DistanceValue</td>
<td>MD_Metadata.identificationInfo. MD_DataIdentification. spatialResolution. MD_Resolution. distance. Distance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DistanceUOM</td>
<td>MD_Metadata.identificationInfo. MD_DataIdentification. spatialResolution. MD_Resolution. distance. Distance@uom</td>
</tr>
<tr>
<td>4</td>
<td>degree of conformity with the harmonised specific-</td>
<td>AnyText</td>
<td>AnyText</td>
<td>Whole resource text. (This queryable will be skipped)</td>
</tr>
<tr>
<td></td>
<td>tions provided for in Article 11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>geographical location</td>
<td>BoundingBox</td>
<td>BoundingBox</td>
<td>MD_Metadata.identificationInfo. _MD_DataIdentification.extent. EX_Extent.geographicElement. EX_GeographicBoundingBox</td>
</tr>
</tbody>
</table>
In case of the extended query, some search properties depend on the resource type the user has selected. E.g., in case of a query that searches only for ‘dataset’, the property ‘service type’ will be disabled. Same is true for ‘Topic category’ in case of a query that searches only for ‘service’.

This mapping might change in the future, since the Draft Implementing Rules (IR) for metadata are still under development. If mappings change with respect to the outcome of this draft IR, table 1 might be adjusted to be compliant with the IR.

### 3.5.5  Response properties

The response properties presented to the user in the client application comply with the metadata elements considered by DT Metadata for response so far.

### 3.5.6  Interfaces and bindings of the broker

The broker does not only access remote catalogues with OGC (conformant) interfaces but provides also the OGC conformant interfaces by itself.

- **OGC Web Catalogue Service (CAT 2.0.0)**, supporting the Dublin Core based information model
- **OGC CAT 2.0 AP ISO19115/19119 (DE-Profile 1.0.1)**, supporting the ISO19115/19119 based information model
So it supports the interfaces of a CS-W catalogue capability class of the CS 2.0.0 ISO19115/ISO19119 Application Profile\(^3\). Since both the OGC_Service- and the Discovery-interface are a mandatory part of the CS 2.0 base specification as well as of the CS 2.0.0 ISO19115/ISO19119 Application Profile there is no mismatch between the Catalogue Service Broker and other OGC CS 2.0 based Catalogues.

In the OGC sense, the Catalogue Service Broker is a CS-WT Catalogue that implements both the CS-W Discovery interface and the CS-WT Manager interface (see Figure 4).

![Figure 4: CSW capability classes (from OGC 04-038r1: OpenGIS® Catalogue Services Specification 2.0 -ISO19115/ISO19119 Application Profile for CSW 2.0)](image)

Because the Catalogue Service Broker implements the CS 2.0.0 ISO19115/ISO19119 Application Profile (DE-Profile 1.0.1), it is based on the HTTP protocol binding (which already defined in the CSW OGC CS 2.0 base specification). While some operations support the encoding of the request messages as keyword-value pairs within a request URI, all operations support the usage of a XML entity-body. Responses are XML-encoded.

The HTTP encoding of catalogue operation requests uses HTTP GET with keyword-value pairs (KVP) encoding and HTTP POST with XML encoding. Requests and responses may also be embedded in the SOAP messaging framework. Table 2 summarises the supported CSW ISO operations and their encoding methods that are supported by the Catalogue Service Broker.

---

\(^3\) Except the optional „GetDomain“ operation
Table 2 - Operation request encoding

<table>
<thead>
<tr>
<th>CSW(T) ISO Operation</th>
<th>Request encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetCapabilities</td>
<td>XML(SOAP) and KVP</td>
</tr>
<tr>
<td>DescribeRecord</td>
<td>XML (POST+SOAP)</td>
</tr>
<tr>
<td>GetRecords</td>
<td>XML (POST+SOAP)</td>
</tr>
<tr>
<td>GetRecordById</td>
<td>XML (POST+SOAP) and KVP</td>
</tr>
<tr>
<td>HarvestRecords</td>
<td>XML (SOAP)</td>
</tr>
<tr>
<td>Transaction</td>
<td>XML (SOAP)</td>
</tr>
<tr>
<td></td>
<td>KVP = keyword-value pair</td>
</tr>
</tbody>
</table>

3.5.7 Distributed search: real-time search vs. catalogue harvesting

The following section describes general access strategies in detail that have been applied:

3.5.7.1 Real-time distributed search

Real-time distributed access means that a user driven query is routed to one or more federated catalogue services. The results of these federated queries are added to the result list (together with an indicator where the specific metadata set is originated)⁴ and send to the user (figure 6).

The advantage of this approach is that the received metadata sets are up-to-date. The disadvantage is that the response time of the overall query depends on the slowest federated catalogue (or the timeout interval as defined for this catalogue in the administration part of the catalogue serviced broker), thus a query may take some time. Furthermore, if the remote catalogue is “down” it provides no results.

---

⁴ In fact this is only true, if all catalogues being selected by the user are ISO AP compliant catalogues, since the indicator for the originator is only part of the brief result set of the ISO AP and not part on Dublin Core.
3.5.7.2 Catalogue harvesting

This functionality provides a means to harvest the entire content of a federated catalogue. The catalogue service broker sends a request to the catalogue to receive any metadata set that could be requested and stores it in its local storage (together with an indicator of where the metadata set is originated).

The initial harvesting is an asynchronous process that is triggered by the administrator of the broker. If a federated catalogue service is cached entirely, the ‘isCached’-flag will be set to ‘true’. In this case, the catalogue broker searches only the local ‘mirrored’ metadata sets in case of a user query and produces a cumulated result list that is send to the user. From a user point of view it is a real distributed query, because cached entries in the result list are indicated as being originated from a federated catalogue service same as it is the case for a real time distributed access.

The advantage of this approach is that it is very fast, as a query does not depend on the run time behaviour of different federated catalogues, but only depends on the performance of the catalogue broker.

The disadvantage is that cached metadata sets may not be up to date, as the caching intervals may be longer than the update interval of the metadata set stored at the remote provider’s side.
Both ISO AP and OGCCORE compliant catalogue services might be cached into the local storage. In the latter case the resulting Dublin Core metadata sets are translated internally to ISO 19115 compliant metadata.

3.5.7.3 Resource harvesting

With resource harvesting no federated catalogue is queried. Instead, it is possible to add content to the local storage of the catalogue service broker by providing access to .xml-files accessible through a web server.

In the best case, the XML files comply with ISOFULL result set as defined by CS-W 2.0 ISO AP and might thus be stored directly in the repository. If the XML files do not comply, a user may also provide an XSLT file with transformation instructions.

On the client side interface of the catalogue service broker, the administrator initializes one or more scheduled harvesting “jobs”. These “jobs” include the URL based path to the file, an optional XSLT file, a harvesting interval and an e-mail address.

By starting the harvesting request, the xml file is initially requested, validated and optionally transformed and stored in the local repository that will be searched in the next query request. The harvesting process is asynchronous and will be repeated periodically depending on the given harvesting interval.

The metadata provider might update the xml files on his local web server. The updated metadata will be updated with the next harvesting interval.

Any time, the catalogue broker has harvested the remote resource the provider of this resource is informed by an email. This is also the case, if some errors occur during harvesting.
Deleting a resource harvesting job is very straightforward: if harvesting fails consecutively for a number of times (based on the configuration), the catalogue service broker assumes that the remote provider has removed the resource from the web server and thus the harvesting job is deleted.

### 3.5.8 Integration strategies for federated catalogues

The strategy of integrating a federated catalogue depends on the specification and, unfortunately, on the implementation of this very catalogue service instance. Three different characteristics can be considered.

#### 3.5.8.1 ISO-Profile-based OGC Catalogues

Because OGC CSW 2.0 AP ISO Profile Catalogues use the same information model, queryables, result sets and the same Catalogue Interface/Bindings integration of these Catalogues is straightforward. The broker can redirect the queries to these catalogues and can directly integrate its results.

#### 3.5.8.2 ebRIM- and other non ISO-Profile OGC Catalogues

There exist two different approaches to integrate an ebRIM catalogue service into a federated search: query the ebRIM catalogue service as an OGCCORE catalogue or query it in an ebRIM manner and translated request/responses to the internal information model of the broker (i.e. ISO).

The latter approach is not feasible, since:

- ... a mapping of ISO search attributes (as defined by the CS 2.0.0 ISO19115/ISO19119 Application Profile) to ebRIM is not standardized
- ... a mapping from the additional queryables as defined in the CS 2.0.0 ISO19115/ISO19119 Application Profile to the ebRIM Information model is not standardized. Hence it cannot be expected to get the correct results by specifying a proprietary mapping.
- ... ISO result sets are not supported.
- ... a mapping of the result sets as defined in the CS 2.0.0 ISO19115/ISO19119 Application Profile from the ebRIM Information model is not standardized. Hence it cannot be expected to produce the correct results by specifying a proprietary mapping from ebRIM to ISO.

For that reason, ebRIM catalogue services are expected to be queried on level of the OGCCORE specification, i.e. OGC CS 2.0 Common profile.
3.5.8.3 OGC CS 2.0 Common Profile catalogues

Although the OGC CS 2.0 AP ebRIM information model is different to the information model of the OGC CS 2.0 AP ISO as well as different to the OGC CS 2.0 base specification, it is possible for an ISO based catalogue (like the Catalogue Service Broker) to query ebRIM catalogues and obtain useful results using the common record schema. This is achieved by using the OGC CSW 2.0 common profile (OGCCORE or Core) which includes the list of core queryable properties, the common record schema and the common interfaces and protocol bindings (see figure below).

![Figure 7: Concept of profiles](image)

The salient feature of the common profile is the information model which declares the `csw:Record` element. The `csw:Record` element is the root element of the common profile's information model and is a container for a subset of Dublin Core metadata elements. The details of the figure are as follows:

**OGCCORE queryables**

According to the OGC CSW 2.0 base specification, all compliant catalogues must support a view of their information model that maps to the `record.xsd` schema. Thus, a client should be able to query any OGC CSW 2.0 catalogue, regardless of the underlying information model, using the search attributes (elements) defined in the `record.xsd` schema (see base specification).
OGCCORE result sets

Thus, a client could also expect that an OGC CSW 2.0 catalogue is capable of returning result sets using the elements of the `record.xsd` schema. These brief-, summary- and full-representations are also defined in the base specification.

Interfaces and Operations

Beside the specific interfaces of a specific profile Catalogue every OGC CS 2.0 Catalogue minimally supports the OGC_Service- and the Discovery-interface with the GetRecords- and GetRecordById-operations of the CSW catalogue capability class of the OGC CSW 2.0. These are the mandatory interfaces of every OGC Catalogue Server.

Binding

Regarding to the last paragraph the HTTP/XML/POST binding for the GetRecords and the HTTP/XML/GET binding for the GetRecordById operation must be supported.

Query language

Regarding to the OGC CS 2.0 base specificaion OGC Catalogue Server must minimally support the OGC Filter Encoding (version 1.0.0 or version 1.1.0) query language because this is the only mandatory query language of the OGC CSW 2.0 GetRecords operation.

3.5.9 Integration of federated catalogues via “adaptors”

For the integration of remote/federated catalogues in a distributed search of the Catalogue Service Broker the following preconditions must minimally be fulfilled:

- Semantic mappings from the information model of the remote catalogue to the information model of the broker must be possible
- Mapping from the interface model of the remote catalogue to the interface model of the broker must be possible

Considering these preconditions there exist 5/6 classes of adaptors (connectors) for the Catalogue Service Broker (with increasing complexity from 1 to 5):

1. Routing Adaptors (more a simple connector), for remote catalogues supporting the same information model, the same interface model, the same encodings and the same protocol binding (here: for OGC CS 2.0.0 AP ISO19115/19 DE-Profile 1.0.1 SOAP) (CSW20DE SOAP in figure 9)

2. Encoding Transition Adaptors, for remote catalogues supporting the same information model, interface model and binding, but a variant encoding (e.g. OGC CS 2.0.0 AP ISO19115/19 DE 1.0.1 SOAP /G) (CSW20ISO SOAP in figure 9)
3. **Binding Transition Adaptors**, for remote catalogues supporting the same information model and interface model (with the same or different encoding), but different protocol binding. Two subtypes could be differentiated:

3.1. **Simple Binding Transition Adaptor**, which does only a switch from HTTP/SOAP to HTTP/POST (not used here)

3.2. **Complex Binding Transition Adaptor**, switching between really different bindings, like from HTTP/SOAP to CSW 2.0 POST ArcIMS or from HTTP/SOAP to HTTP/POST but doing some additional mappings (CSW2.0 POST Arc in figure 9)

4. **Profile Transition Adaptors**, for remote catalogues implementing another profile of the OGC CS 2.0 specification with another enhanced information model (thus supporting minimally the Dublin Core information model and the common OGC CS 2.0 interface model), where it is possible to use these additional information by processing a semantic mapping to the information model of the Catalogue Service Broker (e.g. CSW 2.0 POST ebRIM) (CSW2.0 POST RIM/I in figure 9)

5. **Catalogue Protocol Transition Adaptors**, for remote catalogues supporting another Catalogue Services Protocol, supporting minimally an information model which can be mapped to the Dublin Core information model as well as supporting an interface model with operations to which the main query operations of the CS 2.0 AP ISO19115/19 can be mapped. It may support additional information (with variant or deviant encoding) which can be mapped to the Catalogue Service Brokers information model (e.g. ESRI ArcIMS Catalogue Service) (not used here)
Every Adaptor has to implement specific interfaces and must return the results in Dublin Core format (encoding as defined in OGC CS 2.0) or in ISO19139 format (as defined in the OGC CS 2.0 AP ISO19115/19 specification) depending on the request (Dublin Core (brief/summary/full), isoProfile (brief/summary/full)).

Most of the adaptors are instances of terraCatalog’s XSLT/Pipeline Adaptor implementation or are derived from the corresponding classes. The XSLT/Pipeline Adaptor uses XSL translations. A so called pipeline configuration which belongs to every XSLT/Pipeline Adaptor defines which sequence of XSL transformations has to be applied to an xml based request coming from the broker, before the request is send by a HTTP Connector (HTTP/POST or HTTP/POST/SOAP) to the remote catalogue. Vice versa the pipeline configuration also defines a sequence of XSL transformations which is applied to the xml based response coming from the broker, before the response is returned to the broker.
3.6 Catalogue service broker – client side

On the client side a user is able to select the catalogue service to be considered in a distributed search by querying the local repository. Catalogues from the result list might then be added to the list of active catalogues that will be considered in a distributed search. In case of a query (both extended and simple), these catalogues are queried by the catalogue service broker and the cumulated results are presented to the user.

Any query consists of two sections: a query for the brief result that processes a brief result list and a query for the full metadata set, if the user selects one entry from the result list and queries for detailed metadata.

On the client side, the different types of catalogues enforce different user interfaces.

- Simple/Quick search: Any catalogue is queried with the OGCCORE queryable ‘AnyText’, that indicates to search the entire text resource of the federated catalogue service. If ‘AnyText’ is not supported, a combination of ‘Title’, ‘Subject’ and ‘Abstract’ are used. In addition, a filter to the geographic location can be set interactively on a map, resulting in a query including the ‘BoundingBox’ queryable.
• Extended/Expert search: depending on the catalogues selected by the user, different user interfaces will be rendered on the client side. The user interface looks in any case similar to the user, but the mapping of the queryables changes, if an OGCCORE query is considered: queryables that have no counterparts in OGCCORE queryables are mapped to ‘AnyText’ (see Table 1)

If there exists a non-ISO catalogue in the list of selected catalogues (i.e. a Dublin Core Catalogue), all catalogues being considered in the federated search are queried on OGCCORE level. Thus, the result sets from this query will be Dublin Core.

If only ISO AP compliant catalogues are selected, the catalogues will be queried on ISO-PROFILE level and the results will consist of ISO elements.

The following figure shows the extended search web interface.

![Figure 10: User interface of the extended search](image)

The shortcomings in case of a Dublin core result set are that the information in the client shown to the user is very limited (as Dublin Core is very limited in terms of content). Additional services (like invoking a map server) might not be invoked by the user, as this information is missing (e.g. service type) in a Dublin Core result set.

3.7 Service Monitor

The service monitor is a component to support efficient quality management in an SDI. Once registered, Web services are monitored continuously and evaluated. In the event of
any disturbance or if any availability criteria are not met, the responsible administrators or service operators will be notified automatically, either by email or by SMS (in this case an SMS provider account is required).

The monitoring is performed by invoking the web interface of the various OGC service types (incl. WMS, WFS, CSW) at configurable intervals. If these access instances do not fulfil the prescribed quality requirements with regard to availability, or if the service is out of reach, notification will be sent out by email or SMS message to the responsible persons for that specific service.

![Client of the service monitor](image)

**Figure 11: Client of the service monitor**

In addition to the monitoring and notification functions, index data regarding availability and response times can be retrieved in the form of a report via the evaluation function. The service monitor component possesses a web interface which can be integrated as a web service in a variety of environments and applications.

Two major extensions on the service monitor have been developed for this study:

- a ranking of monitored services
- a graphical output of specific statistics on selected services.
In addition to the existing features of the service monitor, a ranking of monitored services is realized. The monitored services can be shown in a list sorted by average availability, both in an ascending or descending order. This feature is only available in the service monitor client.

Figure 12: Ranking capabilities of monitored services
If service metadata are maintained and managed by the catalogue service broker locally and the associated service instances are monitored by the service monitor, the live status of the service instance (dead/alive) will be shown in the result list of the catalogue service broker client.

### 3.8 Gazetteer

The Gazetteer maps place names to geometries to help a user formulate a spatial filter. The client utilises a standard OpenGIS Web Feature Service (WFS) V. 1.0.0 interface. A user can enter a place name of interest. The name is used in the filter section of a `GetFeature` request. The client then displays the names of all matching features contained in the WFS response. When the user selects one of the names, the client requests the geometry of that feature from the WFS and displays it on the map. This geometry can be used as a spatial filter in a `GetRecords` request to the catalogue.
The WFS that hosts the Gazetteer runs on top of an Oracle database. The feature collections are represented by tables containing columns of the Oracle object relational geometry data type SDO_Geometry.

The data hosted by the WFS is a Gazetteer database from EUROSTAT. This is an enhanced version of the gazetteer available from the National Imagery and Mapping Agency in the USA. From this database schema, only the spatial table GAZETTEERFEATURES is included in the WFS. It contains 5405128 point features. The NAME column is used to select the features according to a client’s filter. To allow a more focused search, the table is classified according to the 9 different feature types included in the GAZETTEERFEATURES table:

- Populated Areas
- Administrative Regions
- Vegetation
- Locality or Area
- Undersea
- Streets, highways, roads, or railroad (Traffic Routes)
- Hypsographic
- Hydrographic
- Spot Feature

This is realised by one database view for each of the feature classes. These views are registered as spatial tables and then served by the WFS.

The import of the database dump supplied by JRC was not error-free. The table GAZETTEERCHARSETS contained in the schema description was missing. For the GAZETTEERFEATURES table, the entry in the Oracle Spatial metadata view USER_SDO_GEOM_METADATA was also missing. This was corrected manually. Also, a spatial index was not created by the import. Such errors can be produced if the schema was not exported by the owning Oracle user, but some other user (e.g. SYSTEM). Another problem is that some of the special characters contained in several place names are not represented correctly in Oracle clients. All problems were encountered while importing on two different Oracle installations.

3.9 Map Client

The mapClient is a standard component of the sdi.suite to invoke map services (both ArcIMS and OGC Web Map services). It is integrated into the catalogue service broker: If a user query leads to a service that has a valid URL in its metadata description, a button will be rendered in the brief or detailed metadata view that invokes the mapClient with that specific service. The user can then directly browse the data provided by that service.
Figure 15: mapClient example

The mapClient delivered is of version 2.0.1.
4 Catalogue service integration

In the following sections all federated catalogues which were primarily taken into consideration are listed and briefly described. Thereafter only the really integrated Catalogues are described in more detail, including the integration strategies used therefore.

4.1 Integration of catalogue services

In the following section the results of the integration work is described. Each catalogue that has been integrated during the study (successfully or not) is described as follows:

- Expected behaviour: based on the catalogue’s interface and information model description, a special behaviour is expected, e.g., an OGC CS 2.0 catalogue should be able to handle requests correctly that comply with this interface. The expected behaviour determines the kind of interoperability test that is applied.

- Interoperability test: how interoperability was tested against this catalogue service instance.

- Problems: the problems that occurred during testing.

- Solution: the solution(s) that has been developed to integrate this special catalogue service instance. In most cases, this means how the appropriate adaptor was implemented and applied.

4.1.1 ebRIM Catalogue service A

4.1.1.1 Expected behaviour

<table>
<thead>
<tr>
<th>Integration as ..</th>
<th>Not possible</th>
</tr>
</thead>
<tbody>
<tr>
<td>OGC CSW 2.0 AP ISO-Profile-based OGC Catalogue</td>
<td>Not possible</td>
</tr>
<tr>
<td>OGC CSW 2.0 Common Profile View of the catalogue</td>
<td>Should be possible (see paragraph 3.5.8.2)</td>
</tr>
</tbody>
</table>

4.1.1.2 Interoperability test

Standardized query via the OGC CSW 2.0 Common profile.
4.1.1.3 Problems

This ebRIM implementation does not support the OGC CSW 2.0 in the standardized way (although returns OGCCORE result sets). The following problems were identified:

- No support of the standardized OGCCORE queryables, instead it’s necessary to use ebRIM specific queryables.

- The attribute *typeName* of the query element must have the value: “Extrinsic-Object”

- Additional query child element must have the value “/ExtrinsicObject”

Example for a valid OGCCORE query:

```xml
<ns1:GetRecords maxRecords="10" outputFormat="text/xml" outputSchema="OGCCORE" requestId="csw:1" resultType="results" service="CSW" startPosition="1" version="2.0.0" xmlns="http://www.opengis.net/cat/csw"
xmlns:ns1="http://www.opengis.net/cat/csw">
  <ns1:Query typeNames="dataset datasetcollection application service">
    <ns1:ElementSetName typeNames=">full</ns1:ElementSetName>
    <ns1:Constraint version="1.0.0">
      <ns1:Filter xmlns:ns1="http://www.opengis.net/ogc">
        <ns1:PropertyIsEqualTo>
          <ns1:PropertyName>title</ns1:PropertyName>
          <ns1:Literal>NGA</ns1:Literal>
        </ns1:PropertyIsEqualTo>
      </ns1:Filter>
    </ns1:Constraint>
  </ns1:Query>
</ns1:GetRecords>
```

Example for a translated query to the ebRIM catalogue:

```xml
<csw:GetRecords xmlns:csw="http://www.opengis.net/cat/csw" maxRecords="10" outputFormat="text/xml" outputSchema="OGCCORE" requestId="csw:1" resultType="results" service="CSW" startPosition="1" version="2.0.0">
  <ns1:Query xmlns:ns1="http://www.opengis.net/cat/csw" xmlns="http://www.opengis.net/cat/csw" typeNames="ExtrinsicObject">
    <csw:ElementName>/ExtrinsicObject</csw:ElementName>
  </ns1:Query>
</csw:GetRecords>
```

A typical result of a response is as follows (excerpt):

```xml
<?xml version="1.0" encoding="utf-8"?>
  <csw:SearchStatus status="complete"/>
</csw:GetRecordsResponse>
```
Since this ebRIM catalogue is not very fast and it provides a lot of data, harvesting/caching is in general a good option.
4.1.1.4 Solution

As a result of the first problem, a full mapping of an OGC CS 2.0 Common Query is not possible. Thus, the catalogue’s content is fully harvested via an empty query, iterating over the whole result set. The adaptor has to generate such a query for a defined block size. During a refinement of this attempt, a mapping of the OGC CSW 2.0 requests to OGC CSW 2.0 AP ebRIM requests was developed.

<table>
<thead>
<tr>
<th>Adaptor</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>Profile Transition Adaptor (PTA), mapping to/from OGC CSW 2.0.1 AP ebRIM (POST).</td>
</tr>
<tr>
<td><strong>Description / Implementation</strong></td>
<td>Basis is the XSLT/Pipeline Adaptor Concept (see paragraph 3.5.9). There is one XSLT stylesheet transforming the OGC CSW 2.0 Common Request to a Request understood by this ebRIM Catalogue. The response of the catalogue is already a valid OGC-CORE full result. This is converted to a minimal ISO full result (and from here by the XSLT pipeline adaptor implementation to one of the result sets asked for by the client).</td>
</tr>
</tbody>
</table>

4.1.2 ebRIM Catalogue service B

4.1.2.1 Expected behaviour

The catalogue is based on the OGC CSW 2.0 AP ebRIM information model.

<table>
<thead>
<tr>
<th>Integration as ..</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>OGC CSW 2.0 AP ISO-Profile-based OGC Catalogue</td>
<td>Not possible.</td>
</tr>
<tr>
<td>OGC CSW 2.0 Common Profile View of the catalogue</td>
<td>Should be possible (see paragraph 3.5.8.2)</td>
</tr>
</tbody>
</table>

4.1.2.2 Interoperability test

Standardized query via the OGC CSW 2.0 Common profile.
4.1.2.3 Problems

This implementation does not support the OGC CSW 2.0 in the standardized way. The following problems were identified:

- Non conformant implementation of OGC CSW 2.0 core. It is not capable to fulfil queries based on OGCCORE queryables.
- It is not possible to query for OGCCORE result sets, only for ebRIM results, so the results have to be translated to OGCCORE/ISO formats.
- It is unclear what API the service conforms to (not conformant with current ebRIM Profile RFP). It is not interoperable with other ebRIM catalogue.
- The metadata do not seem to have a spatial reference (geometries) at all, thus spatial queries will always return empty result sets.
- If a "full" result is requested there are often timeouts.
- Creates invalid values for `dateTime` data type (replacing 'T' with ') with respect to ISO 8601.
- Queries based on ebRIM slot values (e.g. for `dc:subject`) return an incorrect `numberOfRecordsMatched`: the catalogue returns the accumulated number of slot values of all the returned metadata sets, thus resulting in far too many hits.

An example of a translated query understood by the catalogue would be:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<csw:GetRecords xmlns:csw="http://www.opengis.net/cat/csw" maxRecords="1"
    outputFormat="application/xml; charset=UTF-8" outputSchema="csw:profile"
    requestId="csw:1" resultType="results" service="CSW" startPosition="1"
    version="2.0.0">
    <Query xmlns="http://www.opengis.net/cat/csw" typeNames="Dataset">
        <ElementSetName typeNames="Dataset">summary</ElementSetName>
        <csw:Constraint xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:ogc="http://www.opengis.net/ogc"
            xmlns:gml="http://www.opengis.net/gml" version="1.0.0">
            <Filter xmlns="http://www.opengis.net/ogc">
                <PropertyIsLike wildCard="*" singleChar="?" escape="!">
                    <ogc:PropertyName>/Dataset/Name/LocalizedString/@value</ogc:PropertyName>
                    <Literal>*e*</Literal>
                </PropertyIsLike>
            </Filter>
        </csw:Constraint>
    </Query>
</csw:GetRecords>
```

A typical result of a response is shown here:
Although the catalogue is not very fast, harvesting/caching is not an option, because the amount of data is too large.
4.1.2.4 Solution
Since this catalogue is a non-conformant implementation OGC CSW 2.0 core, a full mapping of an OGC CS 2.0 Common Query was not possible. An adaptor was realized that maps OGC CSW 2.0 requests to OGC CSW 2.0 AP ebRIM requests.

<table>
<thead>
<tr>
<th>Adaptor</th>
<th>Description / Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>Profile Transition Adaptor (PTA), mapping to/from OGC CSW 2.0.1 AP ebRIM (POST).</td>
</tr>
<tr>
<td><strong>Description / Implementation</strong></td>
<td>Basis of the adaptor is XSLT/Pipeline Adaptor Concept (see paragraph 3.5.9). There is one XSLT stylesheet transforming the OGC CSW 2.0 Common Request to a Request understood by the ebRIM catalogue. The response of the catalogue is first translated to a useful OGCCORE full result and from here to a minimal ISO full result (and from here by the XSLT pipeline adaptor implementation to one of the result sets asked for by the client).</td>
</tr>
</tbody>
</table>

4.1.3 OGC Core catalogue service A

4.1.3.1 Expected behaviour
The catalogue is based on OGC CSW 2.0 specification, not on a profile.

<table>
<thead>
<tr>
<th>Integration as ..</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OGC CSW 2.0 AP ISO-Profile-based OGC Catalogue</strong></td>
<td>Not possible.</td>
</tr>
<tr>
<td><strong>OGC CSW 2.0 Common Profile View of the catalogue</strong></td>
<td>Should be possible (see paragraph 3.5.8.2)</td>
</tr>
</tbody>
</table>

4.1.3.2 Interoperability test
Standardized query via the OGC CSW 2.0 Common profile.

4.1.3.3 Problems
The implementation does not support the OGC CSW 2.0 in the standardized way. It is implemented rudimentary with errors and non-interoperability, e.g. the following problems were identified:
• The catalogue cannot handle regular filters like:

```xml
<ns1:Filter xmlns:ns1="http://www.opengis.net/ogc">
  <ns1:PropertyIsEqualTo>
    <ns1:PropertyName>title</ns1:PropertyName>
    <ns1:Literal>SOMEPLACE</ns1:Literal>
  </ns1:PropertyIsEqualTo>
</ns1:Filter>
```

• Unfortunately the interface changed continuously during the course of the study

• Returns OGCCORE result sets, but the encoding was not OGC conformant

• A maximum of 8 hits in one result block was returned

One example of a valid OGC CSW 2.0 query which is interpreted by the catalogue correctly is as follows:

```xml
<GetRecords maxRecords="10" outputFormat="text/xml" outputSchema="OGCCORE" requestID="csw:1" resultType="results" service="CSW" startPosition="1" version="2.0.0" xmlns="http://www.opengis.net/cat/csw">
  <Query typeNames="dataset">
    <ElementSetName typeNames=""/>summary</ElementSetName>
  </Query>
</GetRecords>
```

A typical result of a response is as follows (excerpt):

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<GetRecordsResponse version="2.0.0" xmlns:csw="http://www.opengis.net/cat/csw">
  <csw:RequestId>csw:1</csw:RequestId>
  <csw:SearchStatus status="complete" timestamp="2006-05-29T13:51:01.061+02:00"/>
  <csw:SearchResults elementSet="summary" resultSetId="TP-Processor381148903457671" recordSchema="OGCCORE" numberOfRecordsMatched="29" numberOfRecordsReturned="5" nextRecord="6" expires="2006-05-29T14:51:01.061+02:00">
    <csw:AbstractRecord><![CDATA[<?xml version="1.0" encoding="ISO-8859-1"?>
  <csw:SummaryRecord xmlns:csw="http://www.opengis.net/cat/csw"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ows="http://www.opengeospatial.net/ows"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xml:lang="es" xml:fileID="ign-adssll"
  dc:identifier=ign-adssll
  dc:Collection=Series
  dc:type=Collection
  dc:type=Malla
  dc:type=DIGITAL
  dc:type=MAP
  dc:title=SOMETITLE
  dc:title=ADSSLL
  dc:subject=ADMINISTRACION.LIMITES TERRITORIALES.LIMITES ADMINISTRATIVOS
  dc:subject=ADMINISTRACION.ORGANIZACION TERRITORIAL DEL ESTADO.MUNICIPIOS]]></csw:SummaryRecord>
</GetRecordsResponse>
```
4.1.3.4 Solution

Since the results from this catalogue were erroneous a full mapping of an OGC CS 2.0 Common Query was not possible. Thus the catalogue’s content was harvested entirely via an empty query, iterating over the whole result set. The associated adaptor generated such a query for a defined block size. The erroneous result sets are then translated to valid OGCCORE result sets.

In a second phase further integration of this catalogue service was stopped, because of the erroneous result sets. 5

<table>
<thead>
<tr>
<th>Adaptor</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Complex Binding Transition Adaptor (CBTA), mapping to/from OGC CSW 2.0.1 (POST) with complex encoding translations.</td>
</tr>
<tr>
<td>Description / Implementation</td>
<td>A specific Java based Adaptor was developed, which is derived from the new terraCatalog XSLT/Pipeline Adaptor Concept (see paragraph 3.5.9). There is one XSLT stylesheet transforming the OGC CSW 2.0 Common Request to a Request understood by the catalogue. The response of the catalogue is translated to an ISO full result which can be cached.</td>
</tr>
</tbody>
</table>

4.1.4 OGC Core catalogue service B

4.1.4.1 Expected behaviour

The catalogue is based on OGC CSW 2.0 specification, not on a profile.

---

5 If result sets become stable and comply with the OGC CSW 2.0 specification, it should be possible without a large amount of work, to develop an adaptor based on the XSLT/Pipeline Adaptor Concept.
Final Report: Software for distributed Metadata Catalogue services to support the EU Portal

### 4.1.4.2 Interoperability test

Standardized query via the OGC CSW 2.0 Common profile.

### 4.1.4.3 Problems

The implementation does not support the OGC CSW 2.0 in the standardized way. In detail, the following problems were identified:

- Only OGCCORE brief and summary are supported, a “full” request returns ArcCatalogue XML document, not DC.
- The `numberOfRecordsMatched` parameter is always 0, this is a big problem as the algorithm for distributed search relies on this information.
- The `propertyIsLike` comparison operator requires “%” to be used as a wildcard.
- The bounding box of the resulting metadata document in the summary element set is not encoded as defined in the CS-W standard.
- The following queryables are not supported: `Envelope`, `CRS`, `Association`.
- A new namespace for DC defined (CSW 1.0.1 schema).
- The following logical operators do not deliver coherent results: Not, Or
- All spatial operators select the same results. It seems that only `BBox` is supported.
- Requests for all metadata fail: there must be a constraint present in the request; this does not conform to the XML Schema of the OGC CSW 2.0 specification.
- This service returns errors on requests that work with the test installation:
  - Some filters on `AnyText` produce errors, e.g. ' cé%' (if too many expanded query terms are found for the literal value??)
  - some „simple“ filters work (e.g.: `AnyText` like ’airport’)

<table>
<thead>
<tr>
<th>Integration as ..</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>OGC CSW 2.0 AP ISO-Profile-based OGC Catalogue</td>
<td>Not possible.</td>
</tr>
<tr>
<td>OGC CSW 2.0 Common Profile View of the catalogue</td>
<td>Should be possible (see paragraph 3.5.8.2)</td>
</tr>
</tbody>
</table>
• Problems still open:
  
  o approach to construct a "faked" filter to select all metadata did not succeed, proposals by ESRI were not successful
  
  o "creation" resulted in error: [ERR1032] Unable to retrieve documents
  
  o BBox resulted in error: can't declare any more prefixes in this context

One example of a valid OGC CSW 2.0 query which is understood by the catalogue is:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<csw:GetRecords maxRecords="13" outputFormat="text/xml"
    outputSchema="OGCCORE" requestId="csw:1" resultType="RESULTS"
    service="CSW" startPosition="1" version="2.0.0"
    xmlns:csW="http://www.opengis.net/cat/csw"
    xmlns:ns1="http://www.opengis.net/ogc"
    xmlns:gml="http://www.opengis.net/gml">
    <Query typeNames="csw:record" xmlns="http://www.opengis.net/cat/csw">
        <ElementSetName typeNames="csw:record">summary</ElementSetName>
        <Constraint version="1.0.0">
            <ns1:Filter xmlns:ns1="http://www.opengis.net/ogc">
                <ns1:PropertyIsLike escape="!" singleChar="#" wildCard="%">
                    <ns1:PropertyName>Title</ns1:PropertyName>
                    <ns1:Literal>Airport%</ns1:Literal>
                </ns1:PropertyIsLike>
            </ns1:Filter>
        </Constraint>
    </Query>
</csw:GetRecords>
```

A typical result of a response is as follows:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<csw:GetRecordsResponse xmlns:csW="http://www.opengis.net/cat/csw"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:ogc="http://www.opengis.net/ogc"
    xmlns:req="http://www.esri.com/metadata/http-request/
    xmlns:ows="http://www.opengis.net/ows">
    <csW:SearchStatus timestamp="2006-06-09T12:25:11.134+02:00"
        status="subset"/>
    <csW:SearchResults elementSet="summary" recordSchema="csw:Record"
        numberOfRecordsMatched="0" numberOfRecordsReturned="2" nextRecord="3">
        <csW:SummaryRecord>
            <dc:Identifier>{0A72C579-43D7-418D-6649-AF15C1005F6A}</dc:Identifier>
            <dc:type>document</dc:type>
            <dc:title>AirPorts Pan Europe</dc:title>
            <dct:spatial xmlns:dct="http://purl.org/dc/terms/"
                northlimit=71.6133025724;
4.1.4.4 Solution

Because of the problem that the `numberOfRecordsMatched` attribute in the response is always 0, special behaviour had to be implemented to calculate this information, as the algorithm for distributed search relies on this value to deliver correct results.

Because the catalogue does not support OGCCORE full results, a summary result is requested which is transformed to OGCCORE full, including the proprietary encoded bounding box.

<table>
<thead>
<tr>
<th>Adaptor</th>
<th>Complex Binding Transition Adaptor (CBTA), mapping to/from OGC CSW 2.0.1 (POST) with complex encoding translations.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>A specific Java based Adaptor was developed, which is derived from the XSLT/Pipeline Adaptor Concept (see paragraph 3.5.9). There is one XSLT stylesheet transforming the OGC CSW 2.0 Common Request to a request understood by the catalogue. Because of the problem that the <code>numberOfRecordsMatched</code> attribute in the response is always 0, a new adaptor class had to be implemented, since the algorithm for distributed search relies on this value to deliver correct results. This adaptor class first requests a required number of records in OGCCORE summary format, which are transformed to OGCCORE full and finally to ISO full. In a second step, the adaptor requests all records matching the filter in OGCCORE brief format. Only the first few bytes of the response are read out, until the <code>numberOfRecordsReturned</code> attribute is known, then the stream to the server is closed.</td>
</tr>
</tbody>
</table>
4.1.5 ISO catalogue service A

4.1.5.1 Expected behaviour

The catalogue is based on the OGC CSW 2.0 AP ISO 19115/19 (DE 1.0.1) Profile.

<table>
<thead>
<tr>
<th>Integration as ..</th>
<th>Should be possible.</th>
</tr>
</thead>
<tbody>
<tr>
<td>OGC CSW 2.0 AP ISO-Profile-based</td>
<td></td>
</tr>
<tr>
<td>OGC Catalogue</td>
<td></td>
</tr>
<tr>
<td>OGC CSW 2.0 Common Profile View of</td>
<td></td>
</tr>
<tr>
<td>the catalogue</td>
<td></td>
</tr>
</tbody>
</table>

4.1.5.2 Interoperability test

Standardized query via the OGC CSW 2.0 Common Profile as well as via the OGC CSW 2.0 AP ISO Profile.

4.1.5.3 Problems

Currently the OGC CSW 2.0 Common Profile View is not supported by the implementation.

In the case of querying for brief- and summary-result sets via the OGC CSW 2.0 AP ISO View the XML document returned uses a qualified attributeForm instead of an unqualified. This does not conform to the CSW 2.0 AP ISO (DE-Profile 1.0.1) specification.

A typical result of a response is shown here:

```xml
<csw:GetRecordsResponse xmlns:csw="http://www.opengis.net/cat/csw">
  <csw:SearchStatus status="complete"/>
  <csw:SearchResults nextRecord="0" numberOfRecordsMatched="5" numberOfRecordsReturned="5">
    <brief:MD_Metadata
      xmlns:brief="http://schemas.opengis.net/iso19115brief"
      xmlns:gml="http://www.opengis.net/gml"
      xmlns:smXML="http://metadata.dgiwg.org/smXML"
      xmlns:xlink="http://www.w3.org/1999/xlink"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://schemas.opengis.net/iso19115brief http://someserver/schemas/iso19115_brief.xsd">
      <brief:fileIdentifier>
        <smXML:CharacterString>054b5b00-11c8-4125-9ce0-792e17dd3fd3</smXML:CharacterString>
      </brief:fileIdentifier>
      <brief:hierarchyLevel>
```

---

Page: 43
<smXML:MD_ScopeCode
codelist="http://metadata.dgiwg.org/codelistRegistry?MD_ScopeCode"
codelistValue="dataset"> dataset </smXML:MD_ScopeCode>
</brief:hierarchyLevel>
<brief:contact>
<smXML:CI_ResponsibleParty>
<smXML:individualName>
<smXML:CharacterString> SOMENAME </smXML:CharacterString>
</smXML:individualName>
<smXML:organisationName>
<smXML:CharacterString> SOMEORGANISATION </smXML:CharacterString>
</smXML:organisationName>
<smXML:contactInfo>
<smXML:CI_Contact>
<smXML:phone>
<smXML:CI_Telephone>
<smXML:voice>
<smXML:CharacterString> 0511 567 567 77 </smXML:CharacterString>
</smXML:voice>
</smXML:CI_Telephone>
</smXML:phone>
<smXML:address>
<smXML:CI_Address>
<smXML:deliveryPoint>
<smXML:CharacterString> SOMEADDRESS </smXML:CharacterString>
</smXML:deliveryPoint>
<smXML:city>
<smXML:CharacterString> SOMEPLACE </smXML:CharacterString>
</smXML:city>
<smXML:postalCode>
<smXML:CharacterString> 30655 </smXML:CharacterString>
</smXML:postalCode>
<smXML:country>
<smXML:CharacterString> SOMECOUNTRY </smXML:CharacterString>
</smXML:country>
<smXML:electronicMailAddress>
<smXML:CharacterString> SOMEEMAIL </smXML:CharacterString>
</smXML:electronicMailAddress>
</smXML:CI_Address>
</smXML:CI_Contact>
<smXML:role>
<smXML:CI_RoleCode
codelist="http://metadata.dgiwg.org/codelistRegistry?CI_RoleCode"
codelistValue="originator"> originator </smXML:CI_RoleCode>
</smXML:role>
</smXML:CI_ResponsibleParty>
</brief:contact>
<brief:identificationInfo>
  <brief:MD_DataIdentification>
    <brief:title>
      <smXML:CharacterString> Geological Map 1 : 25 000
    </brief:title>
    <brief:topicCategory>
      <smXML:MD_TopicCategoryCode> geoscientificInformation
    </brief:topicCategory>
    <brief:extent>
      <smXML:EX_Extent>
        <smXML:description>
          <smXML:CharacterString> DESCRIPTION
        </smXML:description>
        <smXML:geographicElement>
          <smXML:EX_BoundingPolygon>
            <smXML:extentTypeCode>
              <smXML:Boolean> true          </smXML:Boolean>
            </smXML:extentTypeCode>
            <smXML:innerBoundaryIs>
              <gml:PolygonType>
                <gml:outerBoundaryIs>
                  <gml:LinearRing>
                    <gml:coordinates cs="," decimal="." ts="">
                      8.985912,52.194655 8.993796,52.191592 8.993036,52.190513
                      8.992858,52.189174 8.994102,52.187225 8.995734,52.185883
                      8.998956,52.187302 9.00106,52.187652 9.00095,52.185902 9.000774,52.184265
                      9.001078,52.183197
                      ………
                      8.996191,52.195166 8.993983,52.196166 8.991192,52.196273 8.987301,52.19535
                      8.985912,52.194655
                    </gml:coordinates>
                  </gml:LinearRing>
                  <gml:innerBoundaryIs>
                    <gml:LinearRing>
                      <gml:coordinates cs="," decimal="." ts="">
                        8.659334,53.605934 8.658371,53.607338 8.656446,53.607914
                        8.657284,53.609007 8.655307,53.60854 8.653164,53.608273
                        …...
                        8.658,53.604343 8.658233,53.605458 8.659334,53.605934
                      </gml:coordinates>
                    </gml:LinearRing>
                    <gml:innerBoundaryIs>
                      <gml:LinearRing>
                        <gml:coordinates cs="," decimal="." ts="">
                          8.484762,53.222601 8.486838,53.22049 8.489566,53.218911
                          8.491533,53.217167 8.492757,53.214926 8.491582,53.214283
                          8.494838,53.209849 8.496732,53.21045 8.498031,53.209413 8.496824,53.20895
                          8.5003,53.206089
                          ….....
                          8.486022,53.227378 8.481264,53.235759 8.480809,53.242424
                          8.480776,53.235272 8.485634,53.227305 8.48232,53.226401 8.484762,53.222601
                        </gml:coordinates>
                      </gml:LinearRing>
                    </gml:innerBoundaryIs>
                  </gml:innerBoundaryIs>
                </gml:PolygonType>
              </smXML:innerBoundaryIs>
            </smXML:EX_BoundingPolygon>
          </smXML:EX_BoundingPolygon>
        </smXML:innerBoundaryIs>
      </smXML:EX_BoundingPolygon>
    </smXML:Ex_Extent>
  </brief:extent>
</brief:MD_DataIdentification>
In addition, the catalogue service returns service metadata without any parameter definition for an operation. This is not conformant to the underlying specification (OGC CSW 2.0 AP ISO, DE-Profile 1.0.1).

4.1.5.4 Solution

Because the catalogue does not support the OGC CS 2.0 Common Profile View, every OGC CS 2.0 Common Query must first be translated into a CSW 2.0 AP ISO Query and the ISO results must after that be translated into OGCCORE results (if these are queried for).

It is necessary to strip off the `boundingPolygon` of an ISO document returned by the catalogue.

Additional service metadata sets must be enriched by some “dummy” parameter definitions for every operation.

<table>
<thead>
<tr>
<th>Adaptor</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td>Although in general not necessary (every AP ISO implementation should support OGCCORE), for the integration we needed a Profile Transition Adaptor (PTA), mapping from OGC CS 2.0 Core to OGC CSW 2.0 AP ISO (SOAP).</td>
</tr>
<tr>
<td><strong>Description / Implementation</strong></td>
<td>Basis is the XSLT/Pipeline Adaptor Concept (see paragraph 3.5.9). There is one XSLT stylesheet transforming the OGC CS 2.0 Common Request to an AP ISO Request understood by the catalogue. The response of the catalogue is translated to a useable ISO full result (and from here by the XSLT pipeline adaptor implementation to one of the result sets asked for by the client).</td>
</tr>
</tbody>
</table>

4.1.6 ISO catalogue service B

4.1.6.1 Expected behaviour

The implementation is based on the OGC CS 2.0 AP ISO 19115/19 (DE 1.0.1) Profile.
Integration as..

<table>
<thead>
<tr>
<th>OGC CSW 2.0 AP ISO-Profile-based OGC Catalogue</th>
<th>Should be possible.</th>
</tr>
</thead>
<tbody>
<tr>
<td>OGC CSW 2.0 Common Profile View of the catalogue</td>
<td>Should be possible (see paragraph 3.5.8.2)</td>
</tr>
</tbody>
</table>

4.1.6.2 Interoperability test

Standardized query via the OGC CSW 2.0 Common profile as well as via the OGC CSW 2.0 AP ISO Profile.

4.1.6.3 Problems

None encountered.

4.1.6.4 Solution

The catalogue service broker can connect to a CSW 2.0 AP ISO (DE-Profile 1.0.1) conformant catalogue by using the standard ISO routing adaptor. Within this adaptor no specific transformation of request or response has to be made.

<table>
<thead>
<tr>
<th>Adaptor</th>
<th>Description / Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Standard ISO Routing Adaptor (RA)</td>
</tr>
<tr>
<td>Shortname / Pipeline</td>
<td>ISO</td>
</tr>
<tr>
<td>Description / Implementation</td>
<td>This inherent adaptor is a standard routing adaptor, based on a Java implementation. The query is forwarded by this adaptor without transformations and the results can be integrated without transformations into the overall result set of the Catalogue service broker.</td>
</tr>
</tbody>
</table>
5 Conclusions and Recommendations

This chapter summarizes the major obstacles and shortcomings that occurred during the entire study. The main focus is on the inconsistencies between catalogue interfaces, both with respect to profiles and the core specification.

As a result, a number of recommendations are formulated that should be considered during the next iteration of the evolvement of catalogue specifications.

5.1 Shortcomings of the OGC specifications

5.1.1 Shortcomings of the OGC CSW 2.0 base specification

5.1.1.1 Discovery operations

GetRecords

- The syntax how to use the common queryables in filter statements is not unambiguously defined in the OGC CS 2.0 Specification. Therefore some implementations use the common queryables in full XPath Syntax (e.g. /csw:Record/dc:title) while others use the name (as defined on the queryables table in the CS 2.0 specification).

- It’s currently not clear if all common queryables are mandatory (in the OGC CS 2.0 specification). If not, catalogues should deliver a list of queryables in its capabilities document; currently, this is not standardized in the OGC CS 2.0 specification.

- The typename attribute of a GetRecords request is ambiguously defined:
  - as attribute of QueryType
  - as attribute of ElementSetNameType
  - as core queryable

- The Outputschema attribute of a GetRecords request is ambiguously defined:
  - Some implementations use 'http://schemas.opengis.net/csw/2.0.0/record.xsd' others use “csw:Record”.
  - The specification declares 'csw:Record' as outputSchema (Table 66 of the specification) as well as 'OGCCORE' (used only once) (see section 10.8.4.4 in the base spec).
• There is no attribute standardized in the OGCCORE result sets, so that the value of this attribute can be passed on to a later `GetRecordById` request

**GetRecordById**

• The OUTPUTSCHEMA is not defined as input parameter, so e.g. “csw:Record” cannot be requested for an ISO Catalogue.

### 5.1.1.2 Transactional operations

The specification of the transactional operations is insufficient:

• The `Update` operation has to be specified more precisely

• There is a need for an external control of transactions
  
  o If the SOAP protocol is used it is recommended to support the WS-Transaction Interface. This leads to new requirements: the description of non-functional requirements in service descriptions:

    ▪ Does the CSW support WS-Transactions?
    ▪ Does the CSW support WS-ReliableMessaging?
    ▪ It is still open where this could possibly be described (WSDL, ISO 19119, …)

• Harvest-Operation: The RESOURCETYPE has to be specified more exactly.

### 5.1.1.3 Federated search

The federated search specification is insufficient:

• Federated search should be better documented: what it means, how it works, etc. A conceptual model in the catalogue base specification would be a great help for the developers.

• A CSW Capabilities document should include a list of possible catalogues (including descriptions) where it can distribute a query to.

• It should be possible to include names/identifiers in a `GetRecords` request, defining which catalogues to include in the next search.6

---

6 In our implementation we did this using a project specific protocol in the SOAP header of the requests on the broker
• The search response / result set should include the following information:
  o Which distributed catalogue delivers how many hits?
  o From which distributed catalogue does a metadata record originate?
  o Ideally: runtime information of the search on a distributed catalogue

5.2 Shortcomings of the information model

Concerning the underlying information models there exist a number of problems when it comes to map elements from one model to the other semantically:

• Because the semantic of the content of the OGC Dublin Core is not well defined, a mapping from DC to ISO or ebRIM to ISO is very difficult

• Because the ebRIM profile does not define a content based information model, the mapping from ebRIM information to DC or ISO is difficult

In the first place this is a problem when trying to translate requests / responses between profiles and / or the base specification.
Abstract

This document is the final report of the JRC-con terra GmbH study “Software for distributed Metadata Catalogue services to support the EU Portal”. It describes the underlying system architecture and the relationship of the deployed components. The main focus of this document is on the development of a catalogue service broker to enable a distributed search against various catalogues available in Europe. The catalogues being considered for distributed search were evaluated and tested. The results of these tests, the resulting shortcomings of the underlying specifications and the workarounds to integrate these catalogues in the system are described in detail.
The mission of the JRC is to provide customer-driven scientific and technical support for the conception, development, implementation and monitoring of EU policies. As a service of the European Commission, the JRC functions as a reference centre of science and technology for the Union. Close to the policy-making process, it serves the common interest of the Member States, while being independent of special interests, whether private or national.