

Open Geospatial Consortium Inc.

Date: 2005-04-27

Reference number of this OGC™ project document: **OGC 04-038r2**

Version: 0.9.3

Category: OpenGIS® Catalogue Services Application Profile

Editors: Dr. Uwe Voges, Kristian Senkler

OpenGIS® Catalogue Services Specification 2.0 - ISO19115/ISO19119 Application Profile for CSW 2.0

Copyright notice

Copyright © Open Geospatial Consortium, Inc (2005)

Warning

This document is not an OGC Standard. It is distributed for review and comment. **Double Warning: The Catalogue 2.0 Specification is currently in revision to correct a number of known errors. While the changes to the specification will be minimal, the result is that this Application Profile of CSW will also need to be modified. The Catalogue revisions should be completed this summer. The results of the changes will be released as a Corrigendum. At that time, this Application Profile will be edited and submitted for approval by the OGC membership.**

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: OGC™ Recommendation Paper
Document subtype: Application Profile
Document stage: Publicly Available
Document language: English

Table Of Contents

1	SCOPE	11
2	CONFORMANCE	12
3	NORMATIVE REFERENCES	12
4	TERMS AND DEFINITIONS	13
5	SYMBOLS AND ABBREVIATIONS	16
6	SYSTEM CONTEXT	18
6.1	APPLICATION DOMAIN	18
6.2	ESSENTIAL USE CASES	18
6.2.1	<i>Publish metadata</i>	20
6.2.2	<i>Discover metadata</i>	20
6.2.3	<i>Harvest metadata</i>	21
7	INFORMATION MODELS	22
7.1	CAPABILITY CLASSES.....	22
7.2	CATALOGUE INFORMATION MODEL	23
7.2.1	<i>Information Resources</i>	25
7.2.1.1	Datasets and dataset collections	25
7.2.1.2	Service (loosely-/tightly-/mixed-coupled).....	26
7.2.1.3	Applications	28
7.2.2	<i>CSW extensions</i>	28
7.2.2.1	Extensions to ISO19115:2003.....	28
7.2.2.2	Extensions to ISO19119:2003.....	28
7.2.3	<i>Mappings to the common XML Record format</i>	30
7.2.3.1	OGC core queryable properties.....	30
7.2.3.2	OGC common returnable properties	33
7.2.4	<i>Additional search properties</i>	34
7.3	SUPPORTED DATA BINDINGS	38
7.3.1	Dataset	38
7.3.2	Dataset collection.....	38
7.3.3	Service	38
7.3.4	Application	43
7.4	RESULT SETS.....	43
7.4.1	BRIEF Resultset	44
7.4.2	SUMMARY Resultset	46
7.4.3	FULL Resultset.....	52
7.5	SERVICE INFORMATION MODEL.....	52
7.6	NATIVE LANGUAGE SUPPORT	52
8	EXTERNAL INTERFACES	54
8.1	IMPORTED PROTOCOL BINDING (RELATIONSHIP TO THE COMMON MODEL).....	54
8.2	INTERFACE SPECIFICATIONS.....	55
8.2.1	<i>OGC Service Interface</i>	55
8.2.1.1	GetCapabilities Operation.....	55
8.2.2	<i>CSW Discovery Interface</i>	58
8.2.2.1	GetRecords Operation.....	58
8.2.2.2	GetRecordById Operation.....	62
8.2.2.3	DescribeRecord Operation.....	63
8.2.2.4	GetDomain Operation	65
8.2.3	<i>CSWT Manager Interface</i>	67
8.2.3.1	Transaction Operation.....	67
8.2.3.2	Harvest Operation	72
8.2.3.3	Record locking	75
8.2.4	<i>Error handling</i>	75
8.3	QUERY FACILITIES	76
8.4	IMPLEMENTATION GUIDANCE	77

8.4.1	<i>Technical issues</i>	77
8.4.2	<i>Semantic issues</i>	77
8.4.3	<i>Metadata set example</i>	78
8.5	SECURITY CONSIDERATIONS	82
ANNEX A (NORMATIVE) ABSTRACT TEST SUITE		85
ANNEX B (NORMATIVE) DESIGN RATIONALE		88
ANNEX C (NORMATIVE) WSDL SPECIFICATION CSW PROFILE		89

Figures

Figure 1: Overall system use cases.....	19
Figure 2: Publish metadata	20
Figure 3: Discover metadata.....	20
Figure 4: Harvest metadata.....	21
Figure 5: CSW capability classes	23
Figure 6: Basic classes - Excerpt form ISO19115:2003.....	24
Figure 7: Service metadata class diagram (ISO19119:2003)	27
Figure 8: CSW Extensions to ISO19119:2003.....	29

Tables

Table 1 - Operations provided by CSW capability classes.....	22
Table 2 - Class descriptions.....	24
Table 3 - Class descriptions Service Metadata.....	27
Table 4 - Data dictionary for CSW_ServiceIdentification.....	29
Table 5 - Data dictionary for CSW_CoupledResource.....	30
Table 6 - OGC core queryable properties.....	30
Table 7 - Composition of compound element “Envelope”.....	32
Table 8 - Mapping to common returnable properties.....	33
Table 9 - Mapping dct:spatial.....	34
Table 10 - Additional queryable properties common to all information resources.....	35
Table 11 - Additional queryable properties for datasets, dataset collection and applications.....	35
Table 12 - Composition of union SpatialResolution.....	36
Table 13 - Additional queryable properties for services.....	36
Table 14 - Overview schema / resultset relationships.....	44
Table 15 - Mapping CSW(T) ISO operations to CSW operations.....	54
Table 16 - Operation request encoding.....	55
Table 17 - Parameters in GetCapabilities operation request.....	56
Table 18 - Parameters in GetRecords operation request.....	58
Table 19 - Parameters in GetRecordById operation request.....	62
Table 20 - Parameters in DescribeRecord operation request.....	64
Table 21 - Parameters in GetDomain operation request.....	66
Table 22 - Parameters in Transaction operation request.....	67
Table 23 - Parameters of Insert-,Update-, Delete-Operation.....	68
Table 24 - Parameters of Insert-Operation.....	68
Table 25 - Parameters of Update-Operation.....	69
Table 26 - Parameters of Delete-Operation.....	70
Table 27 - Parameters in Harvest operation request.....	73
Table 28 - Exception codes and meanings (from OGC Common).....	75

Listings

Listing 1 - XML Schema of services.....	38
Listing 2 - Schema of a BRIEF resultset response	44
Listing 3 - Schema of a SUMMARY resultset response.....	46
Listing 4 - Schema of a FULL resultset response	52
Listing 5 - Example code for language and encoding definition (UTF-8)	53
Listing 6 - Example code for language and encoding definition (ISO-8859-1)	53
Listing 7 - OGC_Service: WSDL interface definition	57
Listing 8 - SearchResultType definition.....	61
Listing 9 - GetRecordByIdResponseType definition	63
Listing 10 - TransactionResponse definition.....	71
Listing 11 - HarvestResponse definition	74
Listing 12 - Tightly coupled service description example	78

i. Preface

This document explains how Catalogue Services based on the ISO19115/ISO19119 Application Profile for the OpenGIS® Catalogue Services Specification v2.0 are organised and implemented for the discovery, retrieval and management of data metadata, services metadata and application metadata.

ii. Submitting organisations

The following organisations submitted the original document or its revisions to the Open GIS Consortium, Inc. in response to the OpenGIS Web Service Testbed 2 (OWS2).

con terra Gesellschaft für Angewandte Informationstechnologie mbH

lat/lon Gesellschaft für raumbezogene Informationssysteme mbH

Contributing Entities

The submitting entities were grateful for the contributions from the following companies in the development and revision of this Interface Specification:

Bundesamt für Kartographie und Geodäsie (BKG)

CeGi - Center for Geoinformation GmbH

Geodateninfrastruktur SIG Brandenburg

Geodateninfrastruktur SIG Nordrhein-Westfalen (GDI NRW)

Geoforschungszentrum Potsdam

InGeoForum

Innenministerium Nordrhein-Westfalen

Institut für Geoinformatik, Universität Münster

Koordinierungsstelle UDK/GEIN im Niedersächsischen Umweltministerium

Landesbetrieb Geoinformation und Vermessung Hamburg

Landesvermessung und Geobasisinformation Niedersachsen

Landesvermessung und Geobasisinformation Brandenburg

Landesvermessungsamt Nordrhein-Westfalen (LVerMA NRW)

Wirtschaftsministerium Hessen/AdV

iii. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Contact	Company	Address	Phone	Email
Uwe Voges	con terra GmbH	Martin-Luther-King-Weg 24, 48155 Münster, Germany	Voice: +49-251-7474-402 Fax: +49-251-7474-100	voges@conterra.de
Kristian Senkler	con terra GmbH	Martin-Luther-King-Weg 24, 48155 Münster, Germany	Voice: +49-251-7474-408 Fax: +49-251-7474-100	senkler@conterra.de
Markus Müller	lat/lon GmbH	Meckenheimer Allee 176, 53115 Bonn, Germany	Voice: +49-+177 2470742 Fax: +49-228 73-2153	mueller@lat-lon.de

iv. Revision history

Date	Internal version	Editor	Primary clauses modified	Description
28Jan2004	0.0	Voges	N/A	Initialised Document
10Feb2004	0.1	Voges	N/A	Included first decisions from the "Arbeitskreis Metainformationssysteme"
22Mar2004	0.2	Voges/Senkler		Definition of information model, queryable attributes, result sets,...
22Mar2004	0.2	Senkler	3 Normative references 7.2.1.2 Service Instance	Some general modifications
31Mar2004	0.3	Voges/Senkler	All	Inserted decision from the WG Metainformation. Added section 8.1, Added Operations to External Interfaces
07Apr2004	0.3	Senkler	6.2	Added first version of use cases. To be refined....
16Apr2004	0.3	Senkler	7.x Information model	Added Information model
20Apr2004	0.3	Senkler/Voges	N/A	Revision in several places

26Apr2004	0.3	Senkler	7.2.1/7.2.2	Integration of information model and CSW extensions
28Apr2004	0.3	Voges	8.1/8.2	Adoption to actual OWS Common and CSW 2.0 specification
10May2004	0.3	Voges	7.2.3, 8.3	Core querieables: Association attribute defined, additional search properties defined
12May2004	0.3	Senkler	Preface, 6.2, 7.2.x, 7.3, 7.4, 7.5, 7.7 Appendix A & B	Updated some figures, added the entire information model; added Mapping to common XML record format; added supported data bindings, added result sets; added native language support, added Appendix B and A
18May2004	0.4	Voges	7,8	Some smaller changes in chapter 7. Revised the chapter 8 concerning final changes of the OGC CS 2.0 spec.
18May2004	0.5	Senkler	7.2.4 (modified)	Added HasSecurityConstaints
24May2004	0.6	Voges	8.3, 8.2.2.3	Added query facilities, DescribeRecordType defined, editorial changes
27May2004	0.7	Voges	7.2.3.1	CRS in urn syntax,
04June2004	0.8	Voges	1, 4, 5, 7.2, 7.3	Revised the chapters based on the review comments
05June2004	0.9	Senkler	2, 3, 6, 7, 8, Annexes	Revised the chapters based on the review comments
14June2004	0.9.1	Voges	8	Some smaller "bug-fixes"
12July2004	0.9.2	Senkler	8.2.1.1, Table 8	"Upper-cased" Request parameters, corrected mapping to dc:language
22Dec2004	0.9.3	Voges	7.3.3, 7.4, 8.4	xml schemas updated, implementation guidance, editorial changes.

v. Changes to the OpenGIS® Catalogue Services Specification

According to the application profile the OpenGIS® Abstract Specification requires no further changes that go beyond the required changes stated in OpenGIS® Catalogue Services Specification v2.0.

vi. Future work

Future work may include the extension of the profile to interoperate with UDDI service registry instances. In addition, this profile specification will be adapted towards any changes resulting from the ISO 19139 specification process.

vii. Foreword

This document, through its implementation profile, references several external standards and specifications as dependencies:

1. ISO/IEC TR 10000-1:1998. *Information Technology – Framework and taxonomy of International Standardised Profiles – Part 1: General principles and documentation framework*. Technical Report, JTC 1. Fourth edition. Available [online]: [http://www.iso.ch/iso/en/ittf/PubliclyAvailableStandards/c030726_ISO_IEC_TR_10000-1_1998\(E\).zip](http://www.iso.ch/iso/en/ittf/PubliclyAvailableStandards/c030726_ISO_IEC_TR_10000-1_1998(E).zip).
2. ISO/IEC 10746-2:1996. *Information Technology – Open Distributed Processing – Reference Model: Foundations*. Common text with ITU-T Recommendation X.902. Available [online]: [http://www.iso.ch/iso/en/ittf/PubliclyAvailableStandards/s018836_ISO_IEC_107462_1996\(E\).zip](http://www.iso.ch/iso/en/ittf/PubliclyAvailableStandards/s018836_ISO_IEC_107462_1996(E).zip).
3. Unified Modeling Language (UML) Version 1.3, The Object Management Group (OMG): <http://www.omg.org/cgi-bin/doc?formal/00-03-01>
4. The Extensible Markup Language (XML), World Wide Web Consortium, <http://www.w3.org/TR/1998/REC-xml-19980210>

Annex A, the Abstract Conformance Test Suite, is normative to this specification and shall be implemented when a computing environment requires catalogue services. All other annexes are informative and provide background information, such as terminology and alternative implementation approaches.

Introduction

Catalogue services are the key technology for locating, managing and maintaining distributed geo-resources (i.e. geospatial data, applications and services). With catalogue services, client applications are capable of searching for geo-resources in a standardised way (i.e. through standardised interfaces and operations) and, ideally, they are based on a well-known information model, which includes spatial references and further descriptive (thematic) information that enables client applications to search for geo-resources in very efficient ways.

Whereas interfaces and operations of catalogue services are well defined, it is left up to the developer of the system to define a specific information model which a catalogue service instance provides. This includes, but is not limited to, supported query languages, available search terms, response/result sets, etc. This point is of major importance with respect to interoperability between different catalogue service instances.

In Germany, running catalogue instances result from work being done within the SDI NRW Initiative¹. Members of this initiative have developed an ISO-based application profile for ISO19115 metadata for geodata and ISO19119-based metadata for tightly and loosely-coupled geospatial applications and geospatial services. The foundation of this SDI NRW catalogue profile was the OGC catalogue specification (v1.1.1), the OGC Web Registry Server (WRS) 0.0.2, OGC Web Services Stateless Catalogue Profile (StCS) 0.0.6 and ISO 19115/19119 for content description.

OGC's catalogue revision working group (CS-RWG) has revised and integrated the catalogue implementation specification v1.1.1 that have resulted in CS 2.0. One part of this OGC specification comprises the definition of application profiles according to ISO 19106 (*Geographic information – Profiles*). The overall goal of these profiles is to improve interoperability between systems conforming to a specific profile. Experience has shown that the need for application profiles results from the fact that in practice, there is no single solution for catalogue services that fits every user's needs. As stated in CS 2.0, a base profile that provides a basic set of information objects has to be supported by each catalogue instance; in addition, application profiles for different information communities should be specified.

Hence, this document specifies an application profile for ISO 19115/ISO 19119 metadata with support for XML encoding per ISO 19139 and HTTP protocol binding. It relies on requirements coming from the CS/CSW 2.0 specification. The application profile will form the basis of conformance tests and reference implementations.

1 Scope

This application profile document specifies the interfaces, bindings, and encodings required to publish and access digital catalogues of metadata for geospatial data, services, and applications that comply with the given profile. Metadata act as generalised properties that can be queried and returned through catalogue services for resource evaluation and, in many cases, invocation or retrieval of the referenced resource.

¹ Spatial Data Infrastructure North Rhine Westphalia (federal state of Germany)

2 Conformance

Abstract conformance to the mandatory catalogue service interfaces is described in Annex A. Test data and queries are not included in this profile.

3 Normative references

The following normative documents contain provisions that, through reference in this text, constitute provisions of this part of OGC 03-108. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

IETF RFC 2141 (May 1997), *URN Syntax*, R. Moats

<<http://www.ietf.org/rfc/rfc2141.txt>>

IETF RFC 2396 (August 1998), *Uniform Resource Identifiers (URI): Generic Syntax*, Berners-Lee, T., Fielding, N., and Masinter, L., eds.,

<<http://www.ietf.org/rfc/rfc2396.txt>>

IETF RFC 2616 (June 1999), *Hypertext Transfer Protocol – HTTP/1.1*, Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T., eds.,

<<http://www.ietf.org/rfc/rfc2616.txt>>

IANA, Internet Assigned Numbers Authority, MIME Media Types, available at

<<http://www.iana.org/assignments/media-types/>>

ISO/IEC TR 10000-1:1998. *Information Technology – Framework and taxonomy of International Standardised Profiles – Part 1: General principles and documentation framework*. Technical Report, JTC 1. Fourth edition. Available [online]:

<<http://www.iso.ch/iso/en/ittf/>

PubliclyAvailableStandards/c030726_ISO_IEC_TR_10000-1_1998(E).zip>.

ISO/IEC 10746-2:1996. *Information Technology – Open Distributed Processing – Reference Model: Foundations*. Common text with ITU-T Recommendation X.902. Available [online]:

<[http://www.iso.ch/iso/en/ittf/PubliclyAvailableStandards/s018836_ISO_IEC_107462_1996\(E\).zip](http://www.iso.ch/iso/en/ittf/PubliclyAvailableStandards/s018836_ISO_IEC_107462_1996(E).zip)>.

ISO 19106:2003, *Geographic Information – Profiles*

ISO 19115:2003, *Geographic Information – Metadata*

ISO/DIS 19119, *Geographic Information – Services*

ISO 19139, *Geographic information - Metadata - Implementation specification*

ISO 8601:2000, *Data elements and interchange formats -- Information interchange -- Representation of dates and times*

OGC 99-113, OGC Abstract Specification Topic 13: Catalogue Services

OGC 02-006, OGC Abstract Specification Topic 12: OpenGIS Service Architecture

OGC 02-059, Filter Encoding Implementation Specification

OGC 04-016r2, OWS Common Implementation Specification, January 2004

OMG UML, Unified Modeling Language, Version 1.3, The Object Management Group (OMG):
<http://www.omg.org/cgi-bin/doc?formal/00-03-01>

W3C Recommendation January 1999, Namespaces In XML, <http://www.w3.org/TR/2000/REC-xml-names>.

W3C Recommendation 6 October 2000, Extensible Markup Language (XML) 1.0 (Second Edition), <http://www.w3.org/TR/REC-xml>

W3C Recommendation 2 May 2001: XML Schema Part 0: Primer,
<http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/>

W3C Recommendation 2 May 2001: XML Schema Part 1: Structures,
<http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

W3C Recommendation 2 May 2001: XML Schema Part 2: Datatypes,
<http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

W3C Recommendation (24 June 2003): SOAP Version 1.2 Part 1: Messaging Framework,
<http://www.w3.org/TR/SOAP/>

WSDL, Web Services Description Language (WSDL) 1.1. Available [online]:
<http://www.w3.org/TR/wsdl>

In addition to this document, this specification includes several normative XML Schema files. These are posted online at the URL <http://schemas.opengis.net/>. These XML Schema files are also bundled with this document. In the event of a discrepancy between the bundled and online versions of the XML Schema files, the online files shall be considered authoritative.

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply:

4.1

Application profile

set of one or more base standards and - where applicable - the identification of chosen clauses, classes, subsets, options and parameters of those base standards that are necessary for accomplishing a particular function [ISO 19101, ISO 19106]

4.2

client

software component that can invoke an **operation** from a **server**

4.3

data level

stratum within a set of layered levels in which data is recorded that conforms to definitions of types found at the application model level [ISO 19101]

4.4

dataset collection

collection of datasets sharing the same product specification [ISO 19113, ISO 19114, ISO 19115]

4.5

geographic dataset

dataset with a spatial aspect [ISO 19115]

4.6

geographic information

information concerning phenomena implicitly or explicitly associated with a location relative to the Earth [ISO 19128 draft]

4.7

georesource

geographic information of a specific type (e.g. geographic dataset, geographic application, geographic service)

4.8

identifier

a character string that may be composed of numbers and characters that is exchanged between the client and the server with respect to a specific identity of a resource

4.9

interface

named set of operations that characterise the behaviour of an entity [ISO 19119]

4.10

metadata dataset (metadataset)

metadata describing a specific dataset [ISO 19101]

4.11

metadata entity

group of metadata elements and other metadata entities describing the same aspect of data

NOTE 1 A metadata entity may contain one or more metadata entities.

NOTE 2 A metadata entity is equivalent to a class in UML terminology [ISO 19115].

4.12

metadata schema

conceptual schema describing metadata

NOTE ISO 19115 describes a standard for a metadata schema. [ISO 19101]

4.13

metadata section

subset of metadata that defines a collection of related metadata entities and elements [ISO 19115]

4.14

operation

specification of a transformation or query that an object may be called to execute [ISO 19119]

4.15

parameter

variable whose name and value are included in an operation **request** or **response**

4.16

qualified name

name that is prefixed with its naming context

EXAMPLE The qualified name for the road no attribute in class Road defined in the Roadmap schema is RoadMap.Road.road_no. [ISO 19118].

4.17

request

invocation of an **operation** by a **client**

4.18

response

result of an **operation**, returned from a **server** to a **client**

4.19

schema

formal description of a model [ISO 19101, ISO 19103, ISO 19109, ISO 19118]

4.20

server

service instance

a particular instance of a **service** [ISO 19119 edited]

4.21

service

distinct part of the functionality that is provided by an entity through interfaces [ISO 19119]

capability which a service provider entity makes available to a service user entity at the interface between those entities [ISO 19104 terms repository]

4.22

service interface

shared boundary between an automated system or human being and another automated system or human being [ISO 19101]

4.23

service metadata

metadata describing the **operations** and **geographic information** available at a **server** [ISO 19128 draft]

4.24

state

condition that persists for a period

NOTE The value of a particular feature attribute describes a condition of the feature [ISO 19108].

4.25

transfer protocol

common set of rules for defining interactions between distributed systems [ISO 19118]

4.26

version

version of an Implementation Specification (document) and XML Schemas to which the requested operation conforms

NOTE An OWS Implementation Specification version may specify XML Schemas against which an XML encoded operation request or response must conform and should be validated.

5 Symbols and abbreviations

Some frequently used abbreviated terms:

API	Application Program Interface
COTS	Commercial Off The Shelf
CQL	Common Query Language
CRS	Coordinate Reference System
CSW	Catalogue Service-Web
DCE	Distributed Computing Environment
DC	Dublin Core
DCMI	Dublin Core Metadata Initiative
DCP	Distributed Computing Platform
HTTP	HyperText Transport Protocol
ISO	International Organisation for Standardisation
OGC	Open GIS Consortium
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
UML	Unified Modeling Language

Date: March 22, 2004

ISO19115/ISO19119 Appl.-Profile CSW 2.0

URI Uniform Resource Identifier

URL Uniform Resource Locator

URN Uniform Resource Name

UTF-8 Unicode Transformation Format-8

WSDL Web Service Definition Language

W3C World Wide Web Consortium

XML eXtensible Markup Language

6 System context

This section focuses on the purpose, scope and policies of catalogue services that comply with the given profile. It documents special requirements and describes the context of use.

6.1 *Application domain*

A metadata repository managed by a catalogue implementing this application profile deals with metadata about geospatial data, geospatial services and applications.

It is intentional that the profile specified in this document does not attempt to specify a general-purpose catalogue. Rather, it allows the retrieval and management of the metadata objects referred to above.

This application profile has no specific disciplinary focus. All communities working with these sorts of geospatial information are addressed. Typical communities are surveying, environment, geology, landscaping, water management, power industry, telecommunications etc.

The intention is to implement a generally understood information model based on standard metadata with only a few relationships among the catalogue items. Usage should be as simple as possible, implementing a set of use cases typical in the geospatial community.

The requirements of the information model, search properties, details of the results sets and interfaces were defined in close cooperation with users in various communities as well as software vendors.

This profile allows for a catalogue to accept a request from a client and distribute the request to one or more other catalogues within a federation. In this case, the metadata entries managed by the other catalogues become available to their own clients. It is possible to start a search from only one known location and to search as many catalogues as possible with the same set of attributes.

6.2 *Essential use cases*

This section describes essential use cases for the purpose of demonstrating typical interactions between users, as well as a catalogue service that supports the specified application profile. Figure 1 shows the overall system that contains major interactions between the actors.

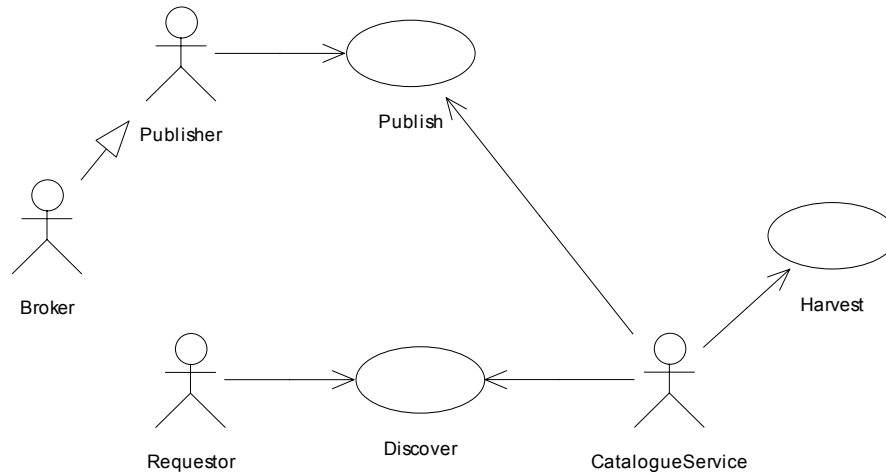


Figure 1: Overall system use cases

An actor is a person, organisation, or external system that plays a role in one or more interactions with the system. Four actors are identified:

Publisher: A publisher publishes metadata descriptions to a catalogue. By doing so, he enables the discovery of that description record by a requestor entity. This actor is also the owner of the geo-resource that he describes.

Broker: This actor is a specialised publisher that publishes and maintains metadata records on behalf of the owner of georesources.

Requestor: This actor searches for metadata records in a catalogue service, either by browsing or through more complicated queries.

Catalogue Service: This is a system that handles the discovery and publishing of metadata entries. Furthermore, this actor has the ability to harvest metadata records from other catalogue services.

The following sections describe the use cases in more detail.

6.2.1 Publish metadata

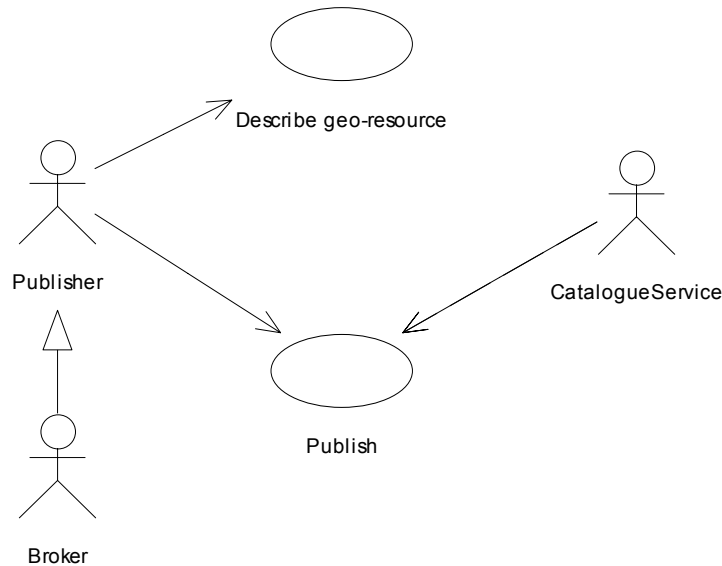


Figure 2: Publish metadata

Description: A publisher describes geo-resources by applying ISO19139 as specified by this document. A geo-resource might be a service, a geodataset (single or collection) or an application. The publisher owns the georesource. As an alternative, the publisher might be a broker that does not own the geo-resource, but describes and publishes metadata descriptions to a catalogue service on behalf of a publisher.

Pre-conditions: The publisher knows the URL of the catalogue service has knowledge about the transaction interface and has the right to access the catalogue service.

Post-conditions: The metadata record is either successfully published to the catalogue service or publishing fails due to a non-valid metadata description.

6.2.2 Discover metadata

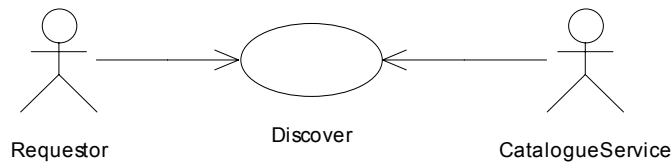


Figure 3: Discover metadata

Description: A requestor discovers metadata entries in a catalogue service either by browsing the content of the catalogue or by placing certain query terms. If a service is discovered that fits his search terms, he can bind to this service in accordance with the information in the result sets of the catalogue service.

Pre-conditions: The requestor knows the location of the catalogue service.

Post-Condition: The requestor receives a valid catalogue response (due to a valid request) with a result set that contains all the information that fits the requestor's query.

6.2.3 Harvest metadata

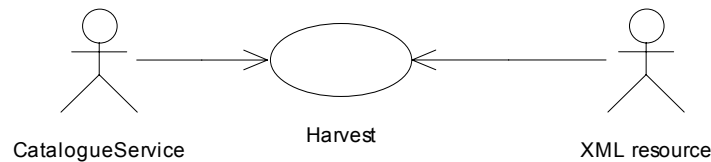


Figure 4: Harvest metadata

Description: A catalogue service may harvest metadata records from a given XML resource, i.e. a metadata description that complies with XML schemas provided by this specification. This could be a metadata description of services or geodata or, additionally in case of services, a capabilities document of an OGC service that complies with OGC Common Implementation Specification.

Pre-conditions: The resource is a valid XML document that complies with the schemas given by this profile. The XML resource must be accessible over a network.

Post-Condition: The XML resource is inserted into the catalogue storage and is available immediately in case of an adequate query.

7 Information models

This view focuses primarily on the information structures and the semantics of information processing (i.e. what the system is about); it describes the public information model that is employed by the catalogue service and the interfaces through which it is accessed. The syntax for all supported representations of the metadata objects is defined.

Note: Mappings to information objects are described by using qualified names according to ISO 19118. For example, the qualified name of resource constraints of a resource being described in a metadata record is MD_Identification.resourceConstraints.

7.1 Capability classes

This section defines the capability classes of the catalogue service. The application profile distinguishes several capability classes based on the general catalogue information model.

Since both the OGC_Service and the Discovery functions must be provided by all conforming implementations, they are a mandatory part of the capability class (CSW-Catalogue). A 'read-only' catalogue service has to provide operations labelled 'CSW'. In addition, a transactional catalogue service has to provide operations labelled 'CSWT'.

The additional Manager functions, providing a standardised interface for the active management (push-model) or the passive harvesting (pull-model) of metadata is mandated to be an optional part of the profile (CSWT Catalogue).

Table 1 - Operations provided by CSW capability classes

Capability class label	Operations provided
CSW	OGC_Service.GetCapabilities CSW Discovery.DescribeRecord CSW Discovery.GetRecords CSW Discovery.GetDomain (optional) CSW Discovery.GetRecordById
CSWT	As above, plus: CSW Manager.Harvest CSW Manager.Transaction

All mandatory and optional classes and their associated operations are listed in Table 1. The additional Manager functions, providing a standardised interface for the active management (push-model) or the passive harvesting (pull-model) of metadata is mandated to be an optional part of the profile (CSWT Catalogue).

Table 1 The logical model of the capability classes is described in the following figure:

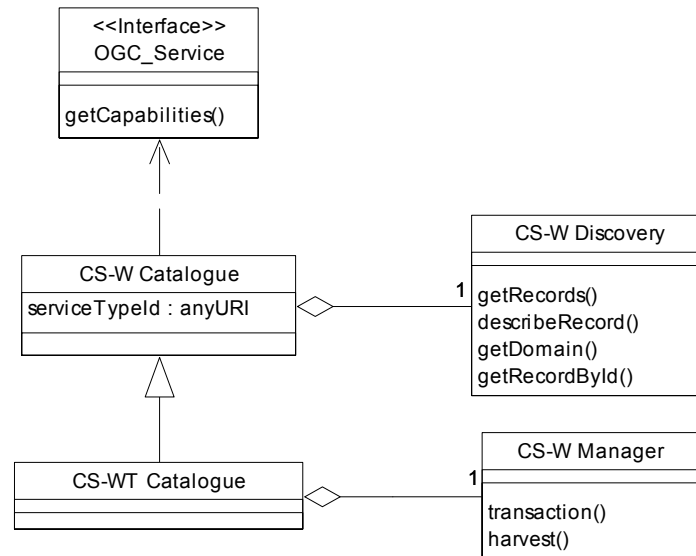


Figure 5: CSW capability classes

OGC_Service – This interface is the logical representation of any OGC-compliant service. It defines the `getCapabilities()` operation;

CSW Catalogue – Implements the OGC_Service interface. This class provides simple synchronous discovery, but no asynchronous discovery is supported. A CSW compliant catalogue must implement at least this functionality².

CSWT Catalogue – This class is a specialisation of the CSW Catalogue. It provides transactional capabilities and is optional.

7.2 Catalogue information model

The CSW information model is based on the international standard for metadata description ISO 19115:2003. In addition, the catalogue uses a metadata description for service metadata based on the draft international ISO 19119:2003 standard to facilitate the management of service metadata. The main purpose of the information model is to provide a formal structure for the description of information resources that can be managed by a catalogue service that complies with the application profile.

ISO19115:2003 specifies a general purpose model for metadata descriptions. In the following section, this document only refers to the changes that have been applied to ISO19115:2003 and ISO 19119:2003 to set up the information model for this application profile. For a more comprehensive description of the model please refer to the original specification documents.

² Except the `getDomain()` operation, which is optional.

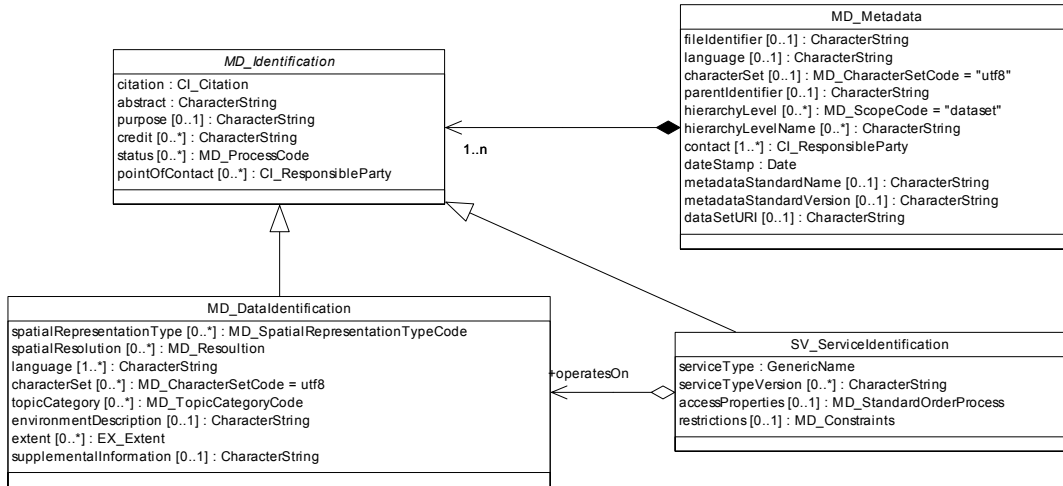


Figure 6: Basic classes - Excerpt form ISO19115:2003

Figure 6 gives a high level overview of the basic classes of the information model. The classes belong to basic packages that are specified by ISO 19115:2003.

Table 2 - Class descriptions

Class name	Description
MD_Metadata	Contains Metadata entity set information. The MD_Metadata entity is an aggregate of MD_Identification and further classes that are suppressed due to clarity, but explained in detail in ISO19115:2003. [ISO19115:2003 A.2.1]
MD_Identification	This abstract class contains information to uniquely identify the information resource that has to be described. MD_Identification is mandatory. It may be implemented as MD_DataIdentification or SV_ServiceIdentification. [ISO19115:2003 A.2.2]
MD_DataIdentification	Subclass and concretion of the abstract class MD_Identification. According to the application profile, MD_DataIdentification describes either data or applications. [ISO19115:2003 A.2.2]
SV_ServiceIdentification	Subclass and concretion of the abstract class MD_Identification. SV_ServiceIdentification gives a high level description of services according to ISO19119:2003. A service might be 'loosely coupled' (with no associated data), 'tightly coupled' (with associated data) or 'mixed coupled'. This distinction is done by setting the couplingType attribute of the CSW_ServiceIdentification class [see ISO19119:2003 7.4.2]

7.2.1 Information Resources

Information resources are entities that can be managed by a CSW-compliant catalogue service. These resources are described and encoded based on the supported information model. The CSW information model supports the description of the following information resources:

Information resource	Description	Logical model
Dataset	An identifiable collection of data	ISO 19115
Dataset Collection	A collection of datasets sharing the same product specification	ISO 19115
Service	A service instance hosted on a specific set of hardware and accessible over a network. A service is tightly coupled, loosely coupled or mixed coupled.	
Loosely coupled	A service instance that is not associated to a specific dataset or dataset collection. Loosely-coupled services may have an association with data types through the service type definition. Dataset metadata need not be provided in the service metadata.	ISO 19119
Tightly coupled	A service that is associated with a specific dataset or dataset collection. Service metadata shall describe both the service and the geographic dataset, the latter being defined in accordance with ISO 19115.	ISO 19119/ISO 19115
Mixed coupled	A service that is associated with a specific dataset or dataset collection. Service metadata shall describe both the service and the geographic dataset, the latter being defined in accordance with ISO 19115. But this service instance can also be used with external data (i.e. data that is not described by the operatesOn association).	ISO 19119/ISO 19115
Application	An information resource that is hosted on a specific set of hardware and accessible over a network.	ISO 19115

7.2.1.1 Datasets and dataset collections

Metadata descriptions of geographic datasets and dataset collections are completely covered by ISO19115:2003. No further extensions have been made to this profile.

Furthermore, according to ISO19115:2003, profiles may be defined that may have additional elements. These profile extensions may be used in connection with the information model at hand, but may not contradict ISO19115:2003, ISO19119:2003 or the extensions being made in this document.

7.2.1.2 Service (loosely-/tightly-/mixed-coupled)

Next to datasets and dataset collections, the catalogue service manages service instance metadata. ISO19119:2003 defines metadata for the description of geographic service instances. These records can be managed and searched using a catalogue service in a similar manner to the metadata entities described in the previous section. The metadata elements for a service provide sufficient information to allow a client to invoke the service based on the metadata record.

As stated in ISO19119:2003, a service instance:

“[...] may be tightly-coupled with a dataset instance, or it may be un-associated with specific data instances, i.e. loosely-coupled. Loosely-coupled services may have an association with data types through the service type definition. In the tightly-coupled case, the service metadata shall describe both the service and the geographic dataset, the latter being defined in accordance with ISO 19115:2003. For the loosely-coupled case, dataset metadata need not be provided in the service metadata.”

Service metadata descriptions consist of the following parts:

- Identification information as `serviceType` and `serviceTypeVersion`, identification information inherited from ISO19115:MD_Identification and optional access properties and restrictions
- Metadata describing the service instance (operations and parameters) including the DCP-dependent `connectionPoints` (service endpoints) of the operations, ideally a WSDL document including constraints on the permitted values of the service operations (primary in the tightly-coupled-case).
- Optional Metadata (ISO19115:2003), describing the geodata with which the service may be associated.

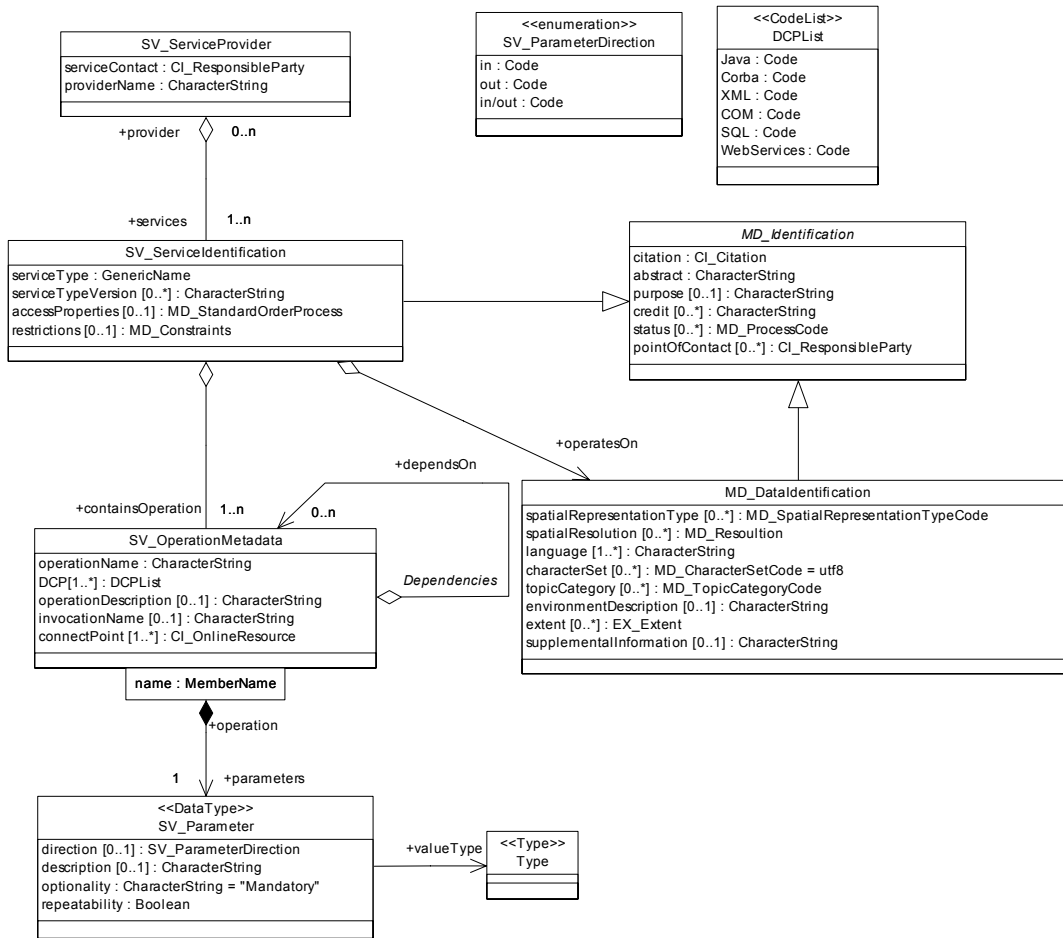


Figure 7: Service metadata class diagram (ISO19119:2003)

Unless described in Table 2, the following table gives a description of classes provided by the above diagram.

Table 3 - Class descriptions Service Metadata

Class name	Description
SV_ServiceProvider	This class describes an organisation that provides services. [ISO19119:2003 C.2.4]
SV_OperationMetadata	This class describes the signature of one and only one method provided by the service.[ISO19119:2003 C.2.3]
SV_Parameter	This class defines the parameter of an operation. [ISO19119:2003 C.2.7]

7.2.1.3 Applications

In the context of this profile, an application is defined as an information resource that is accessible over the Internet and does not fit into the category of services described in the above sections. They are primarily clients (here: HTTP-based) with a specialised, proprietary user interface.

Therefore, they can be described by a valid ISO 19115:2003 metadata entry, including an online link to their location.

7.2.2 CSW extensions

The application profile at hand demands some extensions to the referenced ISO specifications. These extensions and recommendations are described in the following paragraphs.

7.2.2.1 Extensions to ISO19115:2003

No extensions to the ISO19115:2003 specification have to be made. Admittedly, some recommendations are made to clarify the usage of specific elements of a metadata entity. These recommendations are as follows:

- `MD_Metadata.fileIdentifier`: unique identifier for the given metadata entity. The usage of a UUID (Universal Unique Identifier, as specified by <http://www.ietf.org/internet-drafts/draft-mealling-uuid-urn-03.txt>) is strongly recommended to ensure identifier's uniqueness.
- `MD_Metadata.parentIdentifier`: unique identifier for a dataset collection that is described elsewhere and which the metadata entity belongs to. The usage of a UUID (Universal Unique Identifier, as specified by <http://www.ietf.org/internet-drafts/draft-mealling-uuid-urn-03.txt>) is strongly recommended to ensure identifier's uniqueness.

7.2.2.2 Extensions to ISO19119:2003

Particularly with regard to service metadata descriptions by ISO19119:2003, this application profile demands some further extensions to the ISO model. Figure 8 gives an overview of the extensions.

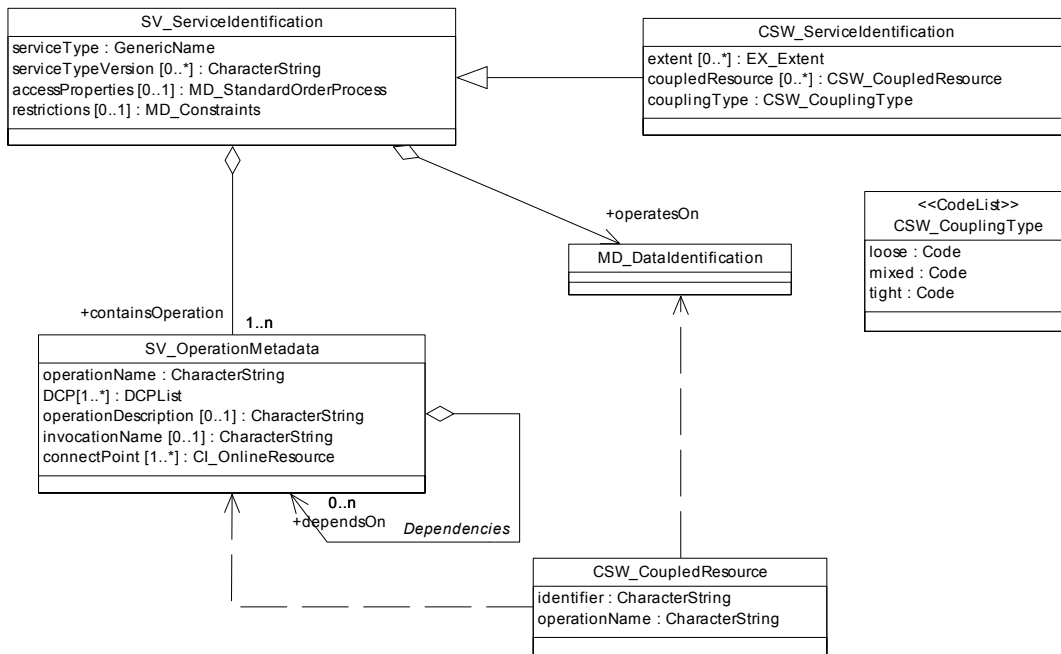


Figure 8: CSW Extensions to ISO19119:2003

Two new classes (CSW_ServiceIdentification and CSW_CoupledResource) and a specialisation of a CodeList is introduced. The following data dictionaries describe in details each class' functionality.

The class CSW_ServiceIdentification extends the functionality of SV_ServiceIdentification in that a service can optionally define its geographic extent.

Table 4 - Data dictionary for CSW_ServiceIdentification

	Attribute name/ Role name	Definition	Obligation /Condition	Maximum occurrence	Data type
1.	extent	extent information including the bounding box, bounding polygon, vertical, and temporal extent of the service	O	N	EX_Extent
2.	coupledResource	further description of the data coupling in case of tightly	O	N	CSW_CoupledResource

		coupled services			
3.	couplingType	Type of coupling between service and associated data (if exists)	M	1	CSW_CouplingType

The class `CSW_CoupledResource` defines which service operations can operate on a specific dataset or dataset collection. Therefore, this class is only useful in case of a tightly coupled service instance.

Table 5 - Data dictionary for CSW_CoupledResource

	Attribute name/ Role name	Definition	Obligation /Condition	Maximum occurrence	Data type
1.	operationName	Name of the service operation	M	1	CharacterString
2.	identifier	Name of the identifier of a given tightly coupled dataset	M	1	MD_Identifier

Note: `CSW_CoupledResource` demands that a given `operationName` or `identifier` MUST refer to an existing `operationName` given by `SV_OperationMetadata.operationName` or an `identifier` given by `MD_DataIdentification.citation.identifier`, respectively.

7.2.3 Mappings to the common XML Record format

7.2.3.1 OGC core queryable properties

The queryable properties are those properties on which a catalogue client can formulate a filter expression. The goal of defining a minimum set of queryable properties is to enable **cross-catalogue** (implementing this profile) discovery, where the same queries can be executed against any catalogue service without modification. This allows queries against different catalogues without prior negotiation on information content.

This requires a set of metadata properties that can be used to characterise a **particular** resource type. To enable **cross-profile** discovery the CSW 2.0 core queryable properties must be included in this set. These core queryable properties can be used to characterise **any** resource.

The core queryables as defined in the OGC catalogue specification and its mappings are defined in the following table.

Table 6 - OGC core queryable properties

Name	Definition	Data type	Property Mapping to Information Model
Subject ^a	The topic of the content of the resource ^b	CharacterString	MD_Metadata.identificationInfo.MD_Identifier.identificationInfo.MD_Keywords.keyword [0..*]
Title ^a	A name given to the resource	CharacterString	MD_Metadata.identificationInfo.MD_Identifier.citation.CI_Citation.title

	resource		ation.citation.CI_Citation.title
Abstract ^a	A summary of the content of the resource	CharacterString	MD_Metadata.identificationInfo.MD_Identifier.abstract
Format ^a	The physical or digital manifestation of the resource ^f	CharacterString	MD_Metadata > MD_Distribution > MD_Format.name
Identifier ^a	An unambiguous reference to the resource within a given context	Identifier	MD_Metadata.identificationInfo.MD_Identifier.citation.CI_Citation.identifier[0..*]
Modified ^c	Date on which the resource was last changed	Date-8601, example: 1963-06-19	MD_Metadata.identificationInfo.MD_Identifier.citation.CI_Citation.date[dateType==revision]
AnyText	This queryable represents the catalogue entry as a whole. Query-Sample: ...AnyText like '%satellite image%'...	CharacterString	Whole resource text.
Type ^a	The nature or genre of the content of the resource. Type can include general categories, genres or aggregation levels of content. ^g	Codelist, one of: vector, grid, textTable, tin, stereoModel, video If Type == "Service", the parameter will be ignored.	MD_Metadata.identificationInfo.MD_DataIdentification.spatialRepresentationType [0..n]
Envelope ^d	A bounding box for identifying a geographic area of interest	Envelope, see Table 7	Envelope, see Table 7
CRS	Coordinate Reference System (Authority and ID) for the Envelope	Identifier ^e	The Identifier must be an URI. The only mandatory CRS is WGS84 having the URN: "urn:opengis:crs:EPSG::4326".
Association	Complete statement of a one-to-one relationship	Association	Because of the imprecise specification of this queryable property, it will not be supported in this version. ³
<p>a Dublin Core Metadata Element Set, version 1.1:ISO Standard 15836-2003 (February 2003)</p> <p>b Typically, a Subject will be expressed as keywords, key phrases or classification codes that describe a topic of the resource. Recommended best practice is to select a value from a controlled</p>			

³ The modelling of association will be done by additional queryables.

	vocabulary or formal classification scheme.
c	DCMI metadata term < http://dublincore.org/documents/dcmi-terms/ >.
d	Same semantics as EX_GeographicBoundingBoxclass in ISO 19115.
e	If not supplied, the Envelope CRS is a Geographic CRS with the Greenwich prime meridian.
f	Dublin Core Metadata Element Set, version 1.1:ISO Standard 15836-2003: Typically, Format may include the media-type or dimensions of the resource. Format may be used to determine the software, hardware or other equipment needed to display or operate the resource.
g	Dublin Core Metadata Element Set, version 1.1:ISO Standard 15836-2003: Type includes terms describing general categories, functions, genres, or aggregation levels for content. To describe the physical or digital manifestation of the resource, use the FORMAT element.

Table 7 - Composition of compound element “Envelope”

Name	Definition	Data type	Property Mapping to Information Model
WestBoundLongitude	Western-most coordinate of the limit of the dataset extent, expressed in longitude in decimal degrees (positive east)	numeric	MD_Metadata.identificationInfo.MD_DataIdentification.extent.EX_GeographicBoundingBox.WestBoundLongitude
SouthBoundLatitude	Southern-most coordinate of the limit of the dataset extent, expressed in latitude in decimal degrees (positive north)	numeric	MD_Metadata.identificationInfo.MD_DataIdentification.extent.EX_GeographicBoundingBox.SouthBoundLongitude
EastBoundLongitude	Eastern-most coordinate of the limit of the dataset extent, expressed in longitude in decimal degrees (positive east)	numeric	MD_Metadata.identificationInfo.MD_DataIdentification.extent.EX_GeographicBoundingBox.EastBoundLongitude
NorthBoundLatitude	Northern-most, coordinate of the limit of the dataset extent, expressed in latitude in decimal degrees (positive north)	numeric	MD_Metadata.identificationInfo.MD_DataIdentification.extent.EX_GeographicBoundingBox.NorthBoundLongitude

7.2.3.2 OGC common returnable properties

The following table lists the mapping between core returnable properties and properties defined by this profile.

Table 8 - Mapping to common returnable properties

Dublin Core metadata element name	Term used in application profile	Property Mapping to Information Model
dc:creator	Creator	MD_Metadata.identificationInfo.MD_Identification.pointOfContact.CI_ResponsibleParty.role.CI_RoleCode [codeValue='originator']
dc:publisher	Publisher	MD_Metadata.identificationInfo.MD_Identification.pointOfContact.CI_ResponsibleParty.role.CI_RoleCode[codeValue='publisher']
dc:contributor	Contributor	MD_Metadata.identificationInfo.MD_Identification.pointOfContact.CI_ResponsibleParty.role.CI_RoleCode[role='author']
dc:language	Language	MD_Metadata.identificationInfo.MD_DataIdentification.language ⁴
dc:rights	Rights	MD_Metadata.identificationInfo._MD_Identification.resourceConstraints.MD_Constraints
dc:title	Title	MD_Metadata.identificationInfo.MD_Identification.citation.CI_Citation.title
dc:subject	Subject	MD_Metadata.identificationInfo.MD_Identification.descriptiveKeywords.MD_Keywords.keyword[0..*]
dct:abstract	Abstract	MD_Metadata.identificationInfo.MD_Identification.abstract
dc:date	Modified	MD_Metadata.identificationInfo.MD_Identification.citation.CI_Citation.date[dateType==revision]
dc:type	Type	MD_Metadata.identificationInfo.MD_DataIdentification.spatialRepresentationType [0..n]
dc:format	Format	MD_Metadata > MD_Distribution > MD_Format.name
dc:identifier	Identifier	MD_Metadata.identificationInfo.MD_Identification.citation.CI_Citation.identifier
dc:source	Source	A reference to a resource from which the present resource is derived. Will not be supported in this version.
dc:relation	Relation, Source, Target	A reference to a related resource. Will not be supported in this version.
dct:spatial	Envelope, CRS	See Table 9

⁴ This is true if hierarchyLevel != "service", otherwise it's blank

Table 9 - Mapping dct:spatial

Dublin Core metadata element name	Term used application profile	Mapping properties
Envelope		
	WestBoundLongitude	MD_Metadata.identificationInfo.MD_DataIdentification.extent.EX_GeographicBoundingBox.WestBoundLongitude
	SouthBoundLatitude	MD_Metadata.identificationInfo.MD_DataIdentification.extent.EX_GeographicBoundingBox.SouthBoundLatitude
	EastBoundLongitude	MD_Metadata.identificationInfo.MD_DataIdentification.extent.EX_GeographicBoundingBox.EastBoundLongitude
	NorthBoundLatitude	MD_Metadata.identificationInfo.MD_DataIdentification.extent.EX_GeographicBoundingBox.NorthBoundLatitude
CRS	CRS	<p>Must have the following structure:</p> <p>“<Authority>::<ID>” (mandatory: WGS84 having the URN: “urn:opengis:crs:EPSG::4326”, other optional) where:</p> <p>Authority results from:</p> <p>MD_Metadata.referenceSystemInfo. MD_ReferenceSystem.ReferenceSystemIdentifier.RS_Identifier.authority</p> <p>ID results from:</p> <p>MD_Metadata.referenceSystemInfo. MD_ReferenceSystem.ReferenceSystemIdentifier.RS_Identifier.code</p>

7.2.4 Additional search properties

In additional to the core queryables, this profile defines the following queryable properties, which an implementation of this profile must minimally support (Table 10).⁵ The catalogue should deliver these queryable properties in its Capabilities document. The following should be noted: If a catalogue entry holds a null-value for queryable *X*, this entry does *not* fulfill any query constraint on that queryable except “is Null”.

The additional search properties are separated by the requested information resource (e.g. defined by the parameter **TYPENAMES** in the GetRecords operation). If search properties are applied on an information resource which does not support this search property, the catalogue throws an exception. However, there are also additional search properties which are common to all information resources.

⁵ For a catalogue instance it may also be possible to define further additional queryable properties.

Table 10 - Additional queryable properties common to all information resources

Name	Definition	Data type	Property Mapping to Information Model
Language	The language of the metadataset.	CharacterString: ISO 639-1:2002 language codes (two letters)	MD_Metadata.language
AlternateTitle	Alternate title of the georesource	CharacterString	MD_Metadata.identificationInfo.MD_Identification.citation.CI_Citation.alternateTitle
CreationDate ⁶	Creation Date of the georesource	Date-8601, example: 1963-06-19	MD_Metadata.identificationInfo.MD_Identification.citation.CI_Citation.date.date[MD_Identification.citation.date.dateType == creation]
Organisationname	Name of the organisation providing the georesource	CharacterString	MD_Identification.citation.CI_Citation.citedResponsibleParty.OrganisationName
HasSecurityConstraints	Are there any security constraints?	Boolean (CharacterString), one of "true" or "false"	MD_Identification_resourceConstraints.MD_SecurityConstraints
HierarchyLevelName	Name of the hierarchy levels for which the metadata set is provided.	CharacterString	MD_Metadata.hierarchyLevelName[0..*]
ParentIdentifier	Fileidentifier of the metadata to which this metadata is a subset (child)	Identifier	MD_Metadata.parentIdentifier
KeywordType	Methods used to group similar keywords	Codelist (MD_KeywordTypeCode), one of: discipline, place, stratum, temporal, theme	MD_Metadata.identificationInfo.MD_Identification.descriptiveKeywords.MD_Keywords.type

For information resources of the types 'dataset', 'dataset collection' and 'application', the following additional search properties must be supported.

Table 11 - Additional queryable properties for datasets, dataset collection and applications

Name	Definition	Data type	Property Mapping to Information Model
TopicCategory	Main theme(s) of the dataset.	CodeList, s. ISO19115:MD_TopicCategoryCode	MD_Metadata.identificationInfo.MD_DataIdentification.topicCategory[0..*]

⁶ RevisionDate can be queried by the "Modified" property of OGC core queryables

GeographicDescription Code	Description of the geographic area using identifiers. Will be ignored, if Envelope is used.	CharacterString	MD_Metadata.identificationInfo.MD_DataIdentification.extent.EX_Extent.EX_GeographicExtent.EX_GeographicDescription.GeographicIdentifier.code
SpatialResolution	Factor which provides a general understanding of the density of spatial data in the dataset.	Spatial Resolution, see Table 12	
TempExtent_begin	Temporal extent information: begin	DateTime-8601	MD_Metadata.identificationInfo.MD_DataIdentification.extent.EX_Extent.temporalElement.EX_TemporalExtent.extent.TM_Primitive
TempExtent_end	Temporal extent information: end	DateTime-8601	MD_Metadata.identificationInfo.MD_DataIdentification.extent.EX_Extent.temporalElement.EX_TemporalExtent.extent.TM_Primitive

Table 12 - Composition of union SpatialResolution

Name	Definition	Datatype	Property Mapping to Information Model
Denominator	Level of detail expressed as a scale factor or a ground distance. Here: the number below the line in a vulgar fraction. Only used, if DistanceValue and DistanceUOM are not used.	Integer	MD_Metadata.identificationInfo.MD_DataIdentification.spatialResolution.MD_Resolution.equivalentScale.MD_RepresentativeFraction.denominator
DistanceValue	Sample ground distance. Here: the distance as decimal value. Only used, if Denominator is not used.	Float, sample: 12.75	MD_Metadata.identificationInfo.MD_DataIdentification.spatialResolution.MD_Resolution.distance.Distance.value
DistanceUOM	Sample ground distance. Here: the name of the unit of measure. Only used, if Denominator is not used.	CodeList, one of: meter, ...	MD_Metadata.identificationInfo.MD_DataIdentification.spatialResolution.MD_Resolution.distance.Distance.uom

For information resources of type 'service' the following additional search properties must be supported.

Table 13 - Additional queryable properties for services

Name	Definition	Data type	Property Mapping to Information Model
ServiceType	Name of a service type.	ServiceTypeID, Codelist: "WFS", "WMS", ...	MD_Metadata.identificationInfo.SV_ServiceIdentification.serviceType
ServiceTypeVersion	The version of a service type.	Codelist: "1.0", "2.0", "1.1.1", ...	MD_Metadata.identificationInfo.SV_ServiceIdentification.serviceTypeVersion

OperatesOn	Name of the identifier of a given tightly coupled dataset.	Identifier. Example: OperateOn = "58f202ac-22cf-11d1-b12d-002035b29092"	MD_Metadata.identificationInfo.SV_ServiceIdentification.operatesOn.MD_DataIdentification.citation.CI_Citation.identifier
Operation	Name of a service operation.	CharacterString One of the available operations. Example: Operation = "Insert"	MD_Metadata.identificationInfo.SV_ServiceIdentification.containsOperation.SV_OperationMetadata.operationName
DCP	Communication service handling the communication between the objects.	Codelist, one of: Java, Corba, SQL, XML, COM, HTTPGet, HTTPPost, HTTPSoap	MD_Metadata.identificationInfo.SV_ServiceIdentification.containsOperation.SV_OperationMetadata.operationName.DCP[1..*]
CouplingType	The coupling type of this service.	Codelist, one of: loose, mixed, tight	CSW_ServiceIdentification.couplingType.CSW_CouplingType.code

With the queryable properties 'OperatesOn', 'Operation' and 'DCP' it is possible to search for services which are tightly-coupled with data in different ways:

- If only 'OperatesOn = <identifier1>' is specified, a search for a service operating on <identifier1> is executed (with an arbitrary Operation)
- If OperatesOn = <identifier1> and Operation = 'Update' is specified, a search for a service operating on <identifier1> with the Update Operation is executed.
- If 'OperatesOn = <identifier1> and Operation = 'Update' and DCP = 'HTTPSoap' is specified, a search for a service operating on <identifier1> with the Update Operation, using the HTTP/Soap Binding is executed.

In an 'and' connected filter-part, all attributes (values) refer to the same metadata entity. For example:

ServiceType = 'WFS' and OperatesOn = <identifier1> and Operation = 'Update'

In this example we are **not** looking for a WFS operating on <identifier1> which also has an Update-Operation, rather we are looking for a WFS, which operates on <identifier1> by using the Update-Operation.

Example 2: An 'and' connected filter-part of the form:

ServiceType = 'WFS' and OperatesOn = <identifier1> and OperatesOn = <identifier2> and Operation = 'Update' and Operation = 'Insert'

In this example we are **not** looking for a WFS, operating on <identifier1> and <identifier2> which also has an Update- and an Insert-Operation, rather we are looking for a WFS, which operates on <identifier1> and <identifier2> and can therefore use the Update and the InsertOperation.

With the CouplingType attribute, it is possible to search for services according to their data coupling type.

7.3 Supported data bindings

This section describes supported representations of each of the information objects specified in section 7.2.1. Currently, the only data binding supported is XML. Any information object that is to be managed by a catalogue service complying with this profile must apply this presentation form. The encoding of any information object in this profile is based on ISO19139:2004 v0.9.

7.3.1 Dataset

According to the given information model, datasets shall be described entirely by ISO19139:2004. No extensions to the XML encoding defined by this specification have to be made. In addition, the following rules apply to dataset XML encoding according to this profile:

- To determine whether the provided metadata record is a dataset, the property `MD_Metadata.hierarchyLevel` must have the value "dataset".
- To uniquely identify a metadata set, `MD_Metadata.fileIdentifier` must have an appropriate id-value

For a detailed description, please refer to <http://metadata.dgiwg.org/metadata/> or see the zip-archive accompanying this document.

7.3.2 Dataset collection

According to the given information model, dataset collection shall be described entirely by ISO19139:2004. No extensions to the XML encoding defined by this specification have to be made. In addition, the following rules apply to dataset XML encoding according to this profile:

- To determine whether the provided metadata record is a dataset, the property `MD_Metadata.hierarchyLevel` must have the value "datasetcollection".
- To uniquely identify a metadata set, `MD_Metadata.fileIdentifier` must have an appropriate id-value

For a detailed description, please refer to <http://metadata.dgiwg.org/metadata/> or see the zip-archive accompanying this document.

7.3.3 Service

According to the given information model, services (tightly or loosely coupled) shall be described by the XML Schema given by Listing 1. This schema makes use of complex types defined by ISO19139:2004 and shall be used in conjunction with that encoding.

Listing 1 - XML Schema of services

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://schemas.opengis.net/iso19119"
xmlns:smXML="http://metadata.dgiwg.org/smXML"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.opengis.net/iso19119"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <!-- =====
    * Topic:          Metadataschema for CS-W 2.0/ISO Profile
  *
    * Filename:       iso19119.xsd
  *
  * =====
```

```

* Description: XML Schema encoding of ISO 19119:2003.
*
* Last update: 01.10.2004
*
* Editor(s): con terra GmbH
*
* Kristian Senkler (senkler@conterra.de)
*
* Uwe Voges (voges@conterra.de)
*
*
* (referenced types are taken from smXML v0.9, available under
* http://metadata.dgiwg.org/ISO19115/ISO19139_v0_9.htm)
*
=====
-->
<!-- =====
Import schemas
===== -->
<xs:import namespace="http://metadata.dgiwg.org/smXML"
schemaLocation="./smXMLv09/smXML/metadataEntity.xsd"/>
<!-- =====
SV_ServiceIdentification
===== -->
<xs:complexType name="SV_ServiceIdentification_Type">
  <xs:complexContent>
    <xs:extension base="smXML:_MD_Identification_Type">
      <xs:sequence>
        <xs:element name="serviceType"
type="smXML:CharacterString_PropertyType"/>
        <xs:element name="serviceTypeVersion"
type="smXML:CharacterString_PropertyType" maxOccurs="unbounded"/>
        <xs:element name="accessProperties"
type="smXML:MD_StandardOrderProcess_Type" minOccurs="0"/>
        <xs:element name="restrictions"
type="smXML:MD_Constraints_PropertyType" minOccurs="0"/>
        <xs:element name="operationMetadata"
type="SV_OperationMetadata_PropertyType" maxOccurs="unbounded"/>
        <xs:element name="operatesOn"
type="smXML:MD_DataIdentification_PropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="SV_ServiceIdentification"
type="SV_ServiceIdentification_Type"
substitutionGroup="smXML:_MD_Identification"/>
<xs:complexType name="SV_ServiceIdentification_PropertyType">
  <xs:choice>
    <xs:element ref="SV_ServiceIdentification"/>
    <xs:element ref="smXML:Reference"/>
  </xs:choice>
</xs:complexType>
<!-- =====

```

```

SV_OperationMetadata
===== -->
<xs:complexType name="SV_OperationMetadata_Type">
  <xs:sequence>
    <xs:element name="operationName"
type="smXML:CharacterString_PropertyType"/>
    <xs:element name="DCP" type="SV_DCPList_PropertyType"
maxOccurs="unbounded"/>
    <xs:element name="operationDescription"
type="smXML:CharacterString_PropertyType" minOccurs="0"/>
    <xs:element name="invocationName"
type="smXML:CharacterString_PropertyType" minOccurs="0"/>
    <xs:element name="connectPoint"
type="smXML:CI_OnlineResource_PropertyType" maxOccurs="unbounded"/>
    <xs:element name="parameters" type="SV_Parameter_PropertyType"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="SV_OperationMetadata"
type="SV_OperationMetadata_Type"/>
<xs:complexType name="SV_OperationMetadata_PropertyType">
  <xs:choice>
    <xs:element ref="SV_OperationMetadata"/>
    <xs:element ref="smXML:Reference"/>
  </xs:choice>
</xs:complexType>
<!-- =====
SV_ServiceProvider
===== -->
<xs:complexType name="SV_ServiceProvider_Type">
  <xs:sequence>
    <xs:element name="serviceContact"
type="smXML:CI_ResponsibleParty_PropertyType"/>
    <xs:element name="providerName"
type="smXML:CharacterString_PropertyType"/>
    <xs:element name="services"
type="SV_ServiceIdentification_PropertyType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="SV_ServiceProvider"
type="SV_ServiceProvider_Type"/>
<xs:complexType name="SV_ServiceProvider_PropertyType">
  <xs:choice>
    <xs:element ref="SV_ServiceProvider"/>
    <xs:element ref="smXML:Reference"/>
  </xs:choice>
</xs:complexType>
<!-- =====
SV_Parameter
===== -->
<xs:complexType name="SV_Parameter_Type">
  <xs:sequence>
    <xs:element name="name" type="smXML:MemberName_PropertyType"/>
    <xs:element name="direction"
type="SV_ParameterDirection_PropertyType"/>
    <xs:element name="description"
type="smXML:CharacterString_PropertyType"/>

```



```

    <xs:element name="optionality"
type="smXML:CharacterString_PropertyType"/>
    <xs:element name="repeatability"
type="smXML:Boolean_PropertyType"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="SV_Parameter" type="SV_Parameter_Type"/>
<xs:complexType name="SV_Parameter_PropertyType">
  <xs:choice>
    <xs:element ref="SV_Parameter"/>
    <xs:element ref="smXML:Reference"/>
  </xs:choice>
</xs:complexType>
<!-- =====
SV_ParameterDirection
===== -->
<xs:simpleType name="SV_ParameterDirection_Type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="in"/>
    <xs:enumeration value="out"/>
    <xs:enumeration value="in/out"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="SV_ParameterDirection"
type="SV_ParameterDirection_Type"/>
<xs:complexType name="SV_ParameterDirection_PropertyType">
  <xs:choice>
    <xs:element ref="SV_ParameterDirection"/>
    <xs:element ref="smXML:Reference"/>
  </xs:choice>
</xs:complexType>
<!-- =====
SV_DCPLList
===== -->
<xs:complexType name="SV_DCPLList_Type">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="codeList" type="xs:anyURI"
use="required"/>
      <xs:attribute name="codeListValue" type="xs:string"
use="required"/>
      <xs:attribute name="codeSpace" type="xs:string"
use="optional"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:element name="SV_DCPLList" type="SV_DCPLList_Type"
substitutionGroup="smXML:CharacterString"/>
<xs:complexType name="SV_DCPLList_PropertyType">
  <xs:choice>
    <xs:element ref="SV_DCPLList"/>
  </xs:choice>
</xs:complexType>
<!-- =====
Profile extensions towards ISO 19119:2003
===== -->
<!-- =====

```

```

CSW_ServiceIdentification
===== -->
<xs:complexType name="CSW_ServiceIdentification_Type">
  <xs:complexContent>
    <xs:extension base="SV_ServiceIdentification_Type">
      <xs:sequence>
        <xs:element name="extent"
type="smXML:EX_Extent_PropertyType" minOccurs="0"/>
        <xs:element name="coupledResource"
type="CSW_CoupledResource_PropertyType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="couplingType"
type="CSW_CouplingType_PropertyType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="CSW_ServiceIdentification"
type="CSW_ServiceIdentification_Type"
substitutionGroup="SV_ServiceIdentification"/>
<xs:complexType name="CSW_ServiceIdentification_PropertyType">
  <xs:choice>
    <xs:element ref="CSW_ServiceIdentification"/>
    <xs:element ref="smXML:Reference"/>
  </xs:choice>
</xs:complexType>
<!-- =====
CSW_CoupledResource
===== -->
<xs:complexType name="CSW_CoupledResource_Type">
  <xs:sequence>
    <xs:element name="operationName"
type="smXML:CharacterString_PropertyType"/>
    <xs:element name="identifier"
type="smXML:CharacterString_PropertyType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="CSW_CoupledResource"
type="CSW_CoupledResource_Type">
  <xs:annotation>
    <xs:documentation>
*** operationName should refer to an existing
SV OperationMetadata.operationName provided by this tightly coupled
service instance
*** identifier should refer to an existing
MD_DataIdentification.citation.identifier provided by this tightly
coupled service instance
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="CSW_CoupledResource_PropertyType">
  <xs:choice>
    <xs:element ref="CSW_CoupledResource"/>
    <xs:element ref="smXML:Reference"/>
  </xs:choice>
</xs:complexType>
<!-- =====

```

```

CSW_CouplingType
===== -->
<xs:complexType name="CSW_CouplingType_Type">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="codeList" type="xs:anyURI"
use="required"/>
      <xs:attribute name="codeListValue" type="xs:string"
use="required"/>
      <xs:attribute name="codeSpace" type="xs:string"
use="optional"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:element name="CSW_CouplingType" type="CSW_CouplingType_Type"
substitutionGroup="smXML:CharacterString"/>
<xs:complexType name="CSW_CouplingType_PropertyType">
  <xs:choice>
    <xs:element ref="CSW_CouplingType"/>
  </xs:choice>
</xs:complexType>
</xs:schema>

```

7.3.4 Application

According to the given information model, applications shall be described entirely by ISO19139:2004. No extensions to the XML encoding defined by this specification have to be made. In addition, the following rules apply to dataset XML encoding according to this profile:

- To determine whether the provided metadata record is an application, the property `MD_Metadata.hierarchyLevel` must have the value "application".
- To uniquely identify a metadata set, `MD_Metadata.fileIdentifier` must have an appropriate id-value.
- The online resource of an application is identified by:
`MD_Metadata.distributionInfo.MD_Distributor.transferOption.MD_DigitalTransferOptions.onLine.CI_OnlineResource.linkage`

For a detailed description, please refer to <http://metadata.dgiwg.org/metadata/> or see the zip-archive accompanying this document.

7.4 Result sets

This section defines XML Schema presentations for valid result sets of the application profile. A representation of catalogue entries in a result set may substitute for the element **csw:AbstractRecord** defined in the `record.xsd` schema of the CSW 2.0 base specification.

Table 14 - Overview schema / resultset relationships

Schema\resultset	BRIEF	SUMMARY	FULL
OGCCore	+	+	+
Profile	+	+	+

7.4.1 BRIEF Resultset

A response to a valid catalogue service request with `ElementSetName=BRIEF`. The following schema specifies the encoding.

Listing 2 - Schema of a BRIEF resultset response

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://schemas.opengis.net/iso19115brief"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:iso19119="http://schemas.opengis.net/iso19119"
xmlns:smXML="http://metadata.dgiwg.org/smXML"
xmlns:csw="http://www.opengis.net/cat/csw"
xmlns="http://schemas.opengis.net/iso19115brief"
attributeFormDefault="unqualified">
  <!-- =====
    Imported namespaces
    ===== -->
  <xs:import namespace="http://metadata.dgiwg.org/smXML"
schemaLocation="./smXMLv09/smXML/metadataEntity.xsd"/>
  <xs:import namespace="http://www.opengis.net/cat/csw"
schemaLocation="../../record.xsd"/>
  <xs:import namespace="http://schemas.opengis.net/iso19119"
schemaLocation="./iso19119.xsd"/>
  <!-- =====
    MD_Metadata
    ===== -->
  <xs:complexType name="MD_Metadata_Type">
    <xs:sequence>
      <xs:element name="fileIdentifier"
type="smXML:CharacterString_PropertyType" minOccurs="0"/>
      <xs:element name="hierarchyLevel"
type="smXML:MD_ScopeCode_PropertyType"/>
      <xs:element name="contact"
type="smXML:CI_ResponsibleParty_PropertyType" maxOccurs="unbounded"/>
      <xs:element name="identificationInfo"
type="_MD_Identification_PropertyType" maxOccurs="unbounded"/>
      <xs:element name="federatedCatalog"
type="smXML:CI_OnlineResource_PropertyType" minOccurs="0"/>
    
```

```

    </xs:sequence>
  </xs:complexType>
  <xs:element name="MD_Metadata" type="MD_Metadata_Type"
substitutionGroup="csw:AbstractRecord"/>
  <xs:complexType name="MD_Metadata_PropertyType">
    <xs:choice>
      <xs:element ref="MD_Metadata"/>
      <xs:element ref="smXML:Reference"/>
    </xs:choice>
  </xs:complexType>
  <!-- =====
    _MD_Identification
    ===== -->
  <xs:complexType name="_MD_Identification_Type" abstract="true">
    <xs:sequence>
      <xs:element name="title"
type="smXML:CharacterString_PropertyType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="_MD_Identification" type="_MD_Identification_Type"
abstract="true"/>
  <xs:complexType name="_MD_Identification_PropertyType">
    <xs:choice>
      <xs:element ref="_MD_Identification"/>
      <xs:element ref="smXML:Reference"/>
    </xs:choice>
  </xs:complexType>
  <!-- =====
    MD_DataIdentification
    ===== -->
  <xs:complexType name="MD_DataIdentification_Type">
    <xs:complexContent>
      <xs:extension base="_MD_Identification_Type">
        <xs:sequence>
          <xs:element name="topicCategory"
type="smXML:MD_TopicCategoryCode_PropertyType" maxOccurs="unbounded"/>
          <xs:element name="extent"
type="smXML:EX_Extent_PropertyType" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="MD_DataIdentification"
type="MD_DataIdentification_Type"
substitutionGroup="_MD_Identification"/>
  <xs:complexType name="MD_DataIdentification_PropertyType">
    <xs:choice>
      <xs:element ref="MD_DataIdentification"/>
      <xs:element ref="smXML:Reference"/>
    </xs:choice>
  </xs:complexType>
  <!-- =====
    SV_ServiceIdentification
    ===== -->
  <xs:complexType name="SV_ServiceIdentification_Type">
    <xs:complexContent>

```

```

    <xs:extension base="_MD_Identification_Type">
      <xs:sequence>
        <xs:element name="serviceType"
type="smXML:CharacterString_PropertyType"/>
        <xs:element name="serviceTypeVersion"
type="smXML:CharacterString_PropertyType" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="SV_ServiceIdentification"
type="SV_ServiceIdentification_Type"
substitutionGroup="_MD_Identification_Type"/>
<xs:complexType name="SV_ServiceIdentification_PropertyType">
  <xs:choice>
    <xs:element ref="SV_ServiceIdentification"/>
    <xs:element ref="smXML:Reference"/>
  </xs:choice>
</xs:complexType>
<!-- =====
CSW_ServiceIdentification
===== -->
<xs:complexType name="CSW_ServiceIdentification_Type">
  <xs:complexContent>
    <xs:extension base="SV_ServiceIdentification_Type">
      <xs:sequence>
        <xs:element name="extent"
type="smXML:EX_Extent_PropertyType" minOccurs="0"/>
        <xs:element name="couplingType"
type="iso19119:CSW_CouplingType_PropertyType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="CSW_ServiceIdentification"
type="CSW_ServiceIdentification_Type"
substitutionGroup="SV_ServiceIdentification"/>
<xs:complexType name="CSW_ServiceIdentification_PropertyType">
  <xs:choice>
    <xs:element ref="CSW_ServiceIdentification"/>
    <xs:element ref="smXML:Reference"/>
  </xs:choice>
</xs:complexType>
</xs:schema>

```

7.4.2 SUMMARY Resultset

A response to a valid catalogue service request with `ElementSetName=SUMMARY`. This element set corresponds to the 'summary' element set in the general catalogue model. The following schema specifies the encoding.

Listing 3 - Schema of a SUMMARY resultset response

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<xs:schema targetNamespace="http://schemas.opengis.net/iso19115summary"
xmlns="http://schemas.opengis.net/iso19115summary"
xmlns:iso19119summary="http://schemas.opengis.net/iso19119"
xmlns:csw="http://www.opengis.net/cat/csw"
xmlns:smXML="http://metadata.dgiwg.org/smXML"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
attributeFormDefault="unqualified">
  <!-- =====
  Imported namespaces
  ===== -->
  <xs:import namespace="http://metadata.dgiwg.org/smXML"
schemaLocation="./smXMLv09/smXML/metadataEntity.xsd"/>
  <xs:import namespace="http://www.opengis.net/cat/csw"
schemaLocation="../../record.xsd"/>
  <xs:import namespace="http://schemas.opengis.net/iso19119"
schemaLocation="./iso19119.xsd"/>
  <!-- =====
  MD_Metadata
  ===== -->
  <xs:complexType name="MD_Metadata_Type">
    <xs:sequence>
      <xs:element name="fileIdentifier"
type="smXML:CharacterString_PropertyType" minOccurs="0"/>
      <xs:element name="language"
type="smXML:CharacterString_PropertyType" minOccurs="0"/>
      <xs:element name="characterSet"
type="smXML:MD_CharacterSetCode_PropertyType" minOccurs="0"/>
      <xs:element name="hierarchyLevel"
type="smXML:MD_ScopeCode_PropertyType" minOccurs="0"/>
      <xs:element name="contact"
type="smXML:CI_ResponsibleParty_PropertyType" maxOccurs="unbounded"/>
      <xs:element name="dateStamp" type="smXML:Date_PropertyType"/>
      <xs:element name="metadataStandardName"
type="smXML:CharacterString_PropertyType" minOccurs="0"/>
      <xs:element name="metadataStandardVersion"
type="smXML:CharacterString_PropertyType" minOccurs="0"/>
      <xs:element name="identificationInfo"
type="_MD_Identification_PropertyType" maxOccurs="unbounded"/>
      <xs:element name="dataQualityInfo"
type="DQ_DataQuality_PropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="referenceSystemInfo"
type="smXML:MD_ReferenceSystem_PropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="distributionInfo"
type="MD_Distribution_PropertyType" minOccurs="0"/>
      <xs:element name="federatedCatalog"
type="smXML:CI_OnlineResource_PropertyType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="MD_Metadata" type="MD_Metadata_Type"
substitutionGroup="csw:AbstractRecord"/>
  <xs:complexType name="MD_Metadata_PropertyType">
    <xs:choice>
      <xs:element ref="MD_Metadata"/>
      <xs:element ref="smXML:Reference"/>
    </xs:choice>
  </xs:complexType>

```

```

</xs:complexType>
<!-- =====
  _MD_Identification
===== -->
<xs:complexType name="_MD_Identification_Type" abstract="true">
  <xs:sequence>
    <xs:element name="citation" type="CI_Citation_PropertyType"/>
    <xs:element name="abstract"
type="smXML:CharacterString_PropertyType"/>
    <xs:element name="pointOfContact"
type="smXML:CI_ResponsibleParty_PropertyType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="resourceConstraints"
type="smXML:MD_Constraints_PropertyType"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="_MD_Identification" type="_MD_Identification_Type"
abstract="true"/>
<xs:complexType name="_MD_Identification_PropertyType">
  <xs:choice>
    <xs:element ref="_MD_Identification"/>
    <xs:element ref="smXML:Reference"/>
  </xs:choice>
</xs:complexType>
<!-- =====
  MD_DataIdentification
===== -->
<xs:complexType name="MD_DataIdentification_Type">
  <xs:complexContent>
    <xs:extension base="_MD_Identification_Type">
      <xs:sequence>
        <xs:element name="spatialRepresentationType"
type="smXML:MD_SpatialRepresentationTypeCode_PropertyType"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="spatialResolution"
type="smXML:MD_Resolution_PropertyType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="language"
type="smXML:CharacterString_PropertyType" maxOccurs="unbounded"/>
        <xs:element name="characterSet"
type="smXML:MD_CharacterSetCode_PropertyType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="topicCategory"
type="smXML:MD_TopicCategoryCode_PropertyType" maxOccurs="unbounded"/>
        <xs:element name="extent" type="EX_Extent_PropertyType"
minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="MD_DataIdentification"
type="smXML:MD_DataIdentification_Type"
substitutionGroup="smXML:_MD_Identification"/>
<xs:complexType name="MD_DataIdentification_PropertyType">
  <xs:choice>
    <xs:element ref="smXML:MD_DataIdentification"/>
    <xs:element ref="smXML:Reference"/>
  </xs:choice>

```



```

    </xs:choice>
  </xs:complexType>
  <!-- =====
  SV_ServiceIdentification
  ===== -->
  <xs:complexType name="SV_ServiceIdentification_Type">
    <xs:complexContent>
      <xs:extension base="_MD_Identification_Type">
        <xs:sequence>
          <xs:element name="serviceType"
type="smXML:CharacterString_PropertyType"/>
          <xs:element name="serviceTypeVersion"
type="smXML:CharacterString_PropertyType" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="SV_ServiceIdentification"
type="SV_ServiceIdentification_Type"
substitutionGroup="_MD_Identification"/>
  <xs:complexType name="SV_ServiceIdentification_PropertyType">
    <xs:choice>
      <xs:element ref="SV_ServiceIdentification"/>
      <xs:element ref="smXML:Reference"/>
    </xs:choice>
  </xs:complexType>
  <!-- =====
  CI_Citation
  ===== -->
  <xs:complexType name="CI_Citation_Type">
    <xs:complexContent>
      <xs:extension base="smXML:_Object_Type">
        <xs:sequence>
          <xs:element name="title"
type="smXML:CharacterString_PropertyType"/>
          <xs:element name="date" type="smXML:CI_Date_PropertyType"
maxOccurs="unbounded"/>
          <xs:element name="identifier"
type="smXML:MD_Identifier_PropertyType" minOccurs="0"
maxOccurs="unbounded"/>
          <xs:element name="citedResponsibleParty"
type="smXML:CI_ResponsibleParty_PropertyType" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="CI_Citation" type="CI_Citation_Type"/>
  <xs:complexType name="CI_Citation_PropertyType">
    <xs:choice>
      <xs:element ref="CI_Citation"/>
      <xs:element ref="smXML:Reference"/>
    </xs:choice>
  </xs:complexType>
  <!-- =====
  MD_Distribution
  ===== -->

```

```

<xs:complexType name="MD_Distribution_Type">
  <xs:complexContent>
    <xs:extension base="smXML:_Object_Type">
      <xs:sequence>
        <xs:element name="distributionFormat"
type="MD_Format_PropertyType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="transferOptions"
type="MD_DigitalTransferOptions_PropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="MD_Distribution" type="MD_Distribution_Type"/>
<xs:complexType name="MD_Distribution_PropertyType">
  <xs:choice>
    <xs:element ref="MD_Distribution"/>
    <xs:element ref="smXML:Reference"/>
  </xs:choice>
</xs:complexType>
<!-- =====
MD_DigitalTransferOptions
===== -->
<xs:complexType name="MD_DigitalTransferOptions_Type">
  <xs:complexContent>
    <xs:extension base="smXML:_Object_Type">
      <xs:sequence>
        <xs:element name="onLine"
type="smXML:CI_OnlineResource_PropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="MD_DigitalTransferOptions"
type="MD_DigitalTransferOptions_Type"/>
<xs:complexType name="MD_DigitalTransferOptions_PropertyType">
  <xs:choice>
    <xs:element ref="MD_DigitalTransferOptions"/>
    <xs:element ref="smXML:Reference"/>
  </xs:choice>
</xs:complexType>
<!-- =====
MD_Format
===== -->
<xs:complexType name="MD_Format_Type">
  <xs:complexContent>
    <xs:extension base="smXML:_Object_Type">
      <xs:sequence>
        <xs:element name="name"
type="smXML:CharacterString_PropertyType"/>
        <xs:element name="version"
type="smXML:CharacterString_PropertyType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

<xs:element name="MD_Format" type="MD_Format_Type"/>
<xs:complexType name="MD_Format_PropertyType">
  <xs:choice>
    <xs:element ref="MD_Format"/>
    <xs:element ref="smXML:Reference"/>
  </xs:choice>
</xs:complexType>
<!-- =====
DQ_DataQuality
===== -->
<xs:complexType name="DQ_DataQuality_Type">
  <xs:complexContent>
    <xs:extension base="smXML:_Object_Type">
      <xs:sequence>
        <xs:element name="lineage"
type="smXML:LI_Lineage_PropertyType" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="DQ_DataQuality" type="DQ_DataQuality_Type"/>
<xs:complexType name="DQ_DataQuality_PropertyType">
  <xs:choice>
    <xs:element ref="DQ_DataQuality"/>
    <xs:element ref="smXML:Reference"/>
  </xs:choice>
</xs:complexType>
<!-- =====
EX_Extent
===== -->
<xs:complexType name="EX_Extent_Type">
  <xs:complexContent>
    <xs:extension base="smXML:_Object_Type">
      <xs:sequence>
        <xs:element name="verticalElement"
type="smXML:EX_VerticalExtent_PropertyType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="temporalElement"
type="smXML:EX_TemporalExtent_PropertyType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="geographicElement"
type="smXML:_EX_GeographicExtent_PropertyType" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="EX_Extent" type="EX_Extent_Type"/>
<xs:complexType name="EX_Extent_PropertyType">
  <xs:choice>
    <xs:element ref="EX_Extent"/>
    <xs:element ref="smXML:Reference"/>
  </xs:choice>
</xs:complexType>
</xs:schema>

```

7.4.3 FULL Resultset

A response to a valid catalogue service request with `ElementSetName=FULL`. The following schema specifies the encoding.

Listing 4 - Schema of a FULL resultset response

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://schemas.opengis.net/iso19115full"
xmlns:smXML="http://metadata.dgiwg.org/smXML"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:csw="http://www.opengis.net/cat/csw"
xmlns:tns="http://schemas.opengis.net/iso19115full"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <!-- ===== imports ===== -->
  <xs:import namespace="http://metadata.dgiwg.org/smXML"
schemaLocation="./smXMLv09/smXML/metadataEntity.xsd"/>
  <xs:import namespace="http://www.opengis.net/cat/csw"
schemaLocation="./../record.xsd"/>
  <xs:import namespace="http://schemas.opengis.net/iso19119"
schemaLocation="./iso19119.xsd"/>
  <!-- ===== imports ===== -->
  <!-- =====
  MD_Metadata
  ===== -->
  <xs:complexType name="MD_Metadata_Type">
    <xs:complexContent>
      <xs:extension base="smXML:MD_Metadata_Type"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="MD_Metadata" type="tns:MD_Metadata_Type"
substitutionGroup="csw:AbstractRecord"/>
  <xs:complexType name="MD_Metadata_PropertyType">
    <xs:choice>
      <xs:element ref="tns:MD_Metadata"/>
      <xs:element ref="smXML:Reference"/>
    </xs:choice>
  </xs:complexType>
</xs:schema>
```

7.5 Service information model

This section describes the content model and syntax for service metadata of a catalogue service compliant with the profile. This model is described according to OGC 04-016r2. A compliant catalogue service shall describe its capabilities document according to the terms defined by OGC 04-016r2.

If the CSW server implements the filter predicate language as defined in [OGC 02-059], then the server must include a `Filter_Capabilities` section in the service metadata to describe which elements of the predicate language are supported (see 8.3).

7.6 Native language support

Support for multiple languages of metadata records is very straightforward. Metadata records that are supplied in different languages have to be published to the catalogue service as self-contained

instances. Each instance of a metadata record specifies its supported language by way of an appropriate attribute in the information model. This attribute is given by the ISO19115:2003 specification.

Listing 5 - Example code for language and encoding definition (UTF-8)

```
<MD_Metadata>
  <smXML:language>
    <smXML:CharacterString>de</smXML:CharacterString>
  </smXML:language>
  <smXML:characterSet>
    <smXML:MD_CharacterSetCode
      codeList="http://www.someserver.com"
      codeListValue="utf8"
      codeSpace="domainCode">004</smXML:MD_CharacterSetCode>
  </smXML:characterSet>
  [...]
  <smXML:identificationInfo>
    <iso19119:CSW_ServiceIdentification>
      <smXML:citation>
        <smXML:CI_Citation>
          <smXML:title>
            <smXML:CharacterString>
              Deutscher Titel in UTF-8
            </smXML:CharacterString>
          </smXML:title>
          [...]
        </smXML:CI_Citation>
      </smXML:citation>
      [...]
    </iso19119:CSW_ServiceIdentification>
  </smXML:identificationInfo>
</iso19115full:MD_Metadata>
```

A corresponding metadata record in English language and with support for ISO-8853-1 is defined as follows.

Listing 6 - Example code for language and encoding definition (ISO-8859-1)

```
<MD_Metadata>
  <smXML:language>
    <smXML:CharacterString>en</smXML:CharacterString>
  </smXML:language>
  <smXML:characterSet>
    <smXML:MD_CharacterSetCode
      codeList="http://www.someserver.com"
      codeListValue=" ISO-8859-1"
      codeSpace="domainCode">006</smXML:MD_CharacterSetCode>
  </smXML:characterSet>
  [...]
  <smXML:identificationInfo>
    <iso19119:CSW_ServiceIdentification>
      <smXML:citation>
        <smXML:CI_Citation>
          <smXML:title>
```

```

        <smXML:CharacterString>
            English title in ISO-8859-1
        </smXML:CharacterString>
    </smXML:title>
    [...]
</smXML:CI_Citation>
</smXML:citation>
    [...]
</iso19119:CSW_ServiceIdentification>
</smXML:identificationInfo>
</iso19115full:MD_Metadata>

```

8 External interfaces

This view describes the externally visible behaviour of the system, including the interfaces provided by its components and the supported protocol bindings. It defines the request and response message structures as part of the operation signatures, primarily the differences to that of the OGC CS 2.0 base specification. It also documents supported query facilities and some implementations guidance as well as a few security considerations.

8.1 Imported protocol binding (*Relationship to the common model*)

This profile imports the HTTP protocol binding from the CSW OGC CS 2.0 specification (CSW 2.0 specification).

Table 15 shows how the operations of this profile (CSW(T) ISO) are mapped to the operations specified by the CSW 2.0 specification. This is a full mapping in that all of the CSW(T) ISO operations have a corresponding CSW operation.

Table 15 - Mapping CSW(T) ISO operations to CSW operations

CSW operation	CSW(T) ISO operation
OGC_Service.GetCapabilities	OGC_Service.GetCapabilities
CSW-Discovery.GetRecords	CSW Discovery.GetRecords
CSW-Discovery.DescribeRecord	CSW Discovery.DescribeRecord
CSW-Discovery.GetDomain	CSW Discovery.GetDomain
CSW-Discovery.GetRecordById ^a	CSW Discovery.GetRecordById ^a
CSW-Publication.Transaction	CSWT Manager.Transaction
CSW-Publication.Harvest	CSWT Manager.Harvest
^a <i>Not defined in the general model</i>	

Most 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 shall use HTTP GET with keyword-value pairs (KVP) encoding and HTTP POST with XML encoding as specified in Section 11 of [OGC 04-016r2]. Requests and responses may also be embedded in the SOAP messaging framework.

Table 16 summarises the CSW(T) ISO operations and their encoding methods that are applied in this profile.

Table 16 - Operation request encoding

CSW(T) ISO Operation	Request encoding
GetCapabilities	XML(POST+SOAP) and KVP
DescribeRecord	XML (POST+SOAP) and KVP
GetDomain	XML (POST+SOAP) and KVP
GetRecords	XML (POST+SOAP) and KVP
GetRecordById	XML (POST+SOAP) and KVP
HarvestRecords	XML (POST+SOAP) and KVP
Transaction	XML (POST+SOAP)
KVP = keyword-value pair	

8.2 Interface specifications

This chapter describes syntax and semantics restrictions and variations of the interface operations in comparison to those of the imported CSW 2.0 HTTP protocol binding. It gives formal, language-independent interface specifications (W3C WSDL) that admit multiple programming language bindings and shows error conditions that can occur.

8.2.1 OGC_Service Interface

8.2.1.1 GetCapabilities Operation

The GetCapabilities operation allows clients to retrieve service metadata from a server. The response to a GetCapabilities request should be an XML document containing service metadata about the server.

HTTP/GET with KVP encoding and SOAP with XML encoding are the mandatory protocol bindings, while HTTP/POST with XML encoding is optional (see OGC document 04-016).

See section 7.5 for the content of the response document.

Table 17 - Parameters in GetCapabilities operation request

Name	Definition	Data type and value	Multiplicity and use	ISO App.-Profile⁷
SERVICE	Service type identifier	Character String type, not empty Value is OWS type abbreviation (e.g., “WMS”, “WFS”)	One (mandatory)	One (mandatory)
REQUEST	Operation name	Character String type, not empty Value is operation name (e.g., “GetCapabilities”)	One (mandatory)	One (mandatory)
ACCEPTVERSIONS	Prioritised sequence of one or more specification versions accepted by client, with preferred versions listed first	Sequence of Character String type, not empty Value is list of x.y.z “version” values	Zero or one (optional) When omitted, return latest supported version (see Version negotiation subclause)	Zero or one (optional) When omitted, return latest supported version (see Version negotiation subclause)

⁷ 8.2.4.

SECTIONS	Unordered list of zero or more names of requested sections in complete service metadata document ^a	Sequence of Character String type, not empty Value is list of section names Allowed section names are specified by each Implementation Specification	Zero or one (optional)	Zero or one (optional) When omitted or not supported by server, return complete service metadata document
UPDATESEQUENCE	Service metadata document version, value is "increased" whenever any change is made in complete service metadata document	Character String type, not empty Values are selected by each server, and are always opaque to clients	Zero or one (optional)	Zero or one (optional) When omitted or not supported by server, return latest service metadata document
ACCEPTFORMATS	Prioritised sequence of zero or more response formats desired by client, with preferred formats listed first	Sequence of Character String type, not empty Value is list of format identifiers Identifiers are MIME types of formats useful for service metadata documents	Zero or one (optional)	Zero or one (optional) When omitted or not supported by server, return service metadata document using MIME type "text/xml"

The WSDL interface component of the OGC_Service interface is shown in Listing 7; this is a fragment of the complete WSDL 2.0 definition for the CSW Catalogue capability class (Annex C). The namespace bindings have been omitted for clarity.

Listing 7 - OGC_Service: WSDL interface definition

```
<wsdl:interface name="OGC_Service">
  <wsdl:operation name="OWS-Common.getCapabilities">
    <wsdl:input message="ows:GetCapabilitiesRequest"/>
    <wsdl:output message="ows:GetCapabilitiesResponse"/>
    <wsdl:fault name="ExceptionReport"
      message="ows:ExceptionReport"/>
  </wsdl:operation>
</wsdl:interface>
```

8.2.2 CSW Discovery Interface

8.2.2.1 GetRecords Operation

In the CSW 2.0 specification the two general model operations **search** and **present** are combined in the form of the **GetRecords** operation, which does a search and a piggybacked present.

HTTP/GET with KVP encoding and SOAP with XML encoding are the mandatory protocol bindings, while HTTP/POST with XML encoding is optional (see OGC document 04-016).

8.2.2.1.1 Request

Table 18 specifies the parameters of the **GetRecords** operation request (KVP and XML). The column **ISO App.-Profile** shows syntax and/or semantics restrictions or variations in comparison to those of the base specification. The encoding in the table is directly suitable for the HTTP GET binding.

The XML encoding of a valid response is specified in the HTTP binding of CSW 2.0 specification. Please refer to this section to determine the appropriate XML schema.

Table 18 - Parameters in GetRecords operation request

Keyword	Datatype & Value	Optionality	ISO App.-Profile ⁸
REQUEST	Character String. Fixed values of "GetRecord". The value is case insensitive.	Mandatory	Mandatory
SERVICE	Character String. Fixed values of "CSW"	Mandatory	Mandatory
VERSION	Character String. Fixed value of "2.0.0"	Mandatory	Mandatory
NAMESPACE	List of Character String, comma separated. Used to specify a namespace and its prefix. Format must be [<prefix>:<url>. If the prefix is not specified then this is the default namespace.	Zero or one (Optional) Include value for each namespace If not included, all qualified names are in default namespace	Optional

⁸ 8.2.4.

Keyword	Datatype & Value	Optionality	ISO App.-Profile ⁸
RESULTTYPE	CodeList. One of “HITS”, “RESULTS” or “VALIDATE”.	Optional. Default value is “HITS”.	Optional. Default value is “HITS”. Indicate whether the catalogue returns the full result set (if ELEMENTSETN AME or ELEMENTNAM E are missing) or just the number of hits the query found. If the value is “HITS”, ELEMENTSETN AME or ELEMENTNAM E are ignored.
OUTPUTFORMAT	Character String Value is Mime type The only value that must be supported is “text/xml”. Other supported values may include “text/html” and “text/plain”.	Optional. Default value is “text/xml”.	Only “text/ xml” supported.
OUTPUTSCHEMA	Defined in a profile. Must support “OGCCORE”.	Optional Default value is “OGCCORE”.	Optional Must support “OGCCORE” and “Profile”. Default value is “OGCCORE”.
STARTPOSITION	PositiveInteger	Optional The default value is 1.	Optional The default value is 1.
MAXRECORDS	PositiveInteger	Optional The default values is 10.	Optional The default values is 10.
TYPENAMES	List of Character String, comma separated Unordered List of object types implicated in the query	Mandatory	Mandatory: One or more of: Service, Dataset; Datasetcollection , Application

Keyword	Datatype & Value	Optionality	ISO App.-Profile ⁸
ELEMENTSETNAME or ELEMENTNAME	List of Character String	Optional Default action is to present all metadata elements.	Optional Default action is to present all metadata elements (full). Valid values are: brief, summary, full
CONSTRAINTLANGUAGE	CodeList One of "CQL_TEXT" or "FILTER"	Must be specified with QUERYCONSTRAINT parameter.	Must be specified with QUERYCONSTRAINT parameter.
CONSTRAINT_LANGUAGE_VERSION	String May be used to specify a version number indicating which version of a specification the constraint conforms to. For example, if "FILTER" is being used, this parameter could be set to "1.0.0" indicating that the filter conforms to version 1.0.0 of the Filter Encoding Implementation Specification [OGC 02-059].	Mandatory	Mandatory
CONSTRAINT	String The predicate expression specified in the language indicated by the CONSTRAINTLANGUAGE parameter.	Optional Default action is to execute an unconstrained query.	Optional Default action is to execute an unconstrained query.
SORTBY	List of Character String, comma separated Ordered list of names of metadata elements to use for sorting the response Format of each list item is <i>metadata_element_name:A</i> indicating an ascending sort or <i>metadata_element_name:D</i> indicating descending sort	Optional Default action is to present the records in the order in which they are retrieved.	Optional Default action is to present the records in the order in which they are retrieved.

Keyword	Datatype & Value	Optionality	ISO App.-Profile ⁸
DISTRIBUTEDSEARCH	Boolean	Optional Default value is FALSE.	Optional Default value is FALSE.
HOPCOUNT	Integer	May be specified only if DISTRIBUTEDSEARCH is specified. If not specified, the default value is 2.	May be specified only if DISTRIBUTEDSEARCH is specified. If not specified, the default is 2.
RESPONSEHANDLER	URL	Zero or one (Optional) If not included, process request synchronously	Not supported: only synchronous requests supported

8.2.2.1.2 Response

This operation must respond with an XML document based on an XML schema as defined in the CSW 2.0 specification. In this schema the <SearchResults> element is a generic XML container. The content of the <SearchResults> element is the set of records returned by the **GetRecords** operation. The actual records returned by the catalogue are a substitute for the element <csw:AbstractRecord>. The following XML schema fragment defines the **SearchResultsType**:

Listing 8 - SearchResultType definition

```
<xsd:complexType name="SearchResultsType"
  id="SearchResultsType">
  <xsd:sequence>
    <xsd:element ref="csw:AbstractRecord" minOccurs="0"
      maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="resultSetId" type="xsd:anyURI"
    use="optional"/>
  <xsd:attribute name="elementSet" type="csw:ElementSetType"
    use="optional"/>
  <xsd:attribute name="recordSchema" type="xsd:anyURI"
    use="optional"/>
  <xsd:attribute name="numberOfRecordsMatched"
    type="xsd:nonNegativeInteger" use="required"/>
  <xsd:attribute name="numberOfRecordsReturned"
    type="xsd:nonNegativeInteger" use="required"/>
  <xsd:attribute name="nextRecord"
    type="xsd:nonNegativeInteger" use="optional"/>
  <xsd:attribute name="expires" type="xsd:dateTime"
    use="optional"/>
</xsd:complexType>
```

Catalogues implementing this application profile shall respond with an XML document that corresponds to one of the schemas defined in section 7.4. The following limitations are applied:

- The parameter `resultSetId` is optional,
- The Parameter `expires` is not supported in the response because of the stateless character of the profile. The result sets expires immediately.
- The value `StatusType.processing` should never be used by the status attribute.

If the client asks for the common CSW record syntax, the catalogue must return the records in one of the schemas defined in `record.xsd` (see CSW 2.0 specification). The common CSW record syntax is an XML-based encoding of Dublin Core metadata terms; it encompasses the core metadata properties.

8.2.2.2 GetRecordById Operation

The mandatory **GetRecordById** request retrieves the default representation of catalogue records using their identifier. The **GetRecordById** operation is an implementation of the **Present** operation from the general model. This operation presumes that a previous query has been performed in order to obtain the identifiers that may be used with this operation.

8.2.2.2.1 Request

Table 19 specifies the parameters of the **GetRecordById** operation request (KVP and XML). The column **ISO App.-Profile** shows syntax and/or semantics restrictions or variations in comparison to those of the base spec. The encoding in the table is directly suitable for the HTTP GET binding.

The XML encoding of a valid response is specified in the HTTP binding of CSW 2.0 specification. Please refer to this section to determine the appropriate XML schema.

Table 19 - Parameters in GetRecordById operation request

Keyword	Datatype & Value	Optionality	ISO App.-Profile ⁹
REQUEST	Character String. Fixed values of “GetRecordById”. The value is case insensitive.	Mandatory	Mandatory
SERVICE	Character String. Fixed values of “CSW”	Mandatory	Mandatory
VERSION	Character String. Fixed value of “2.0.0”	Mandatory	Mandatory

⁹ 8.2.4.

Keyword	Datatype & Value	Optionality	ISO App.-Profile ⁹
ELEMENTSETNAME	CodeList with allowed values: “brief”, “summary” or “full”	Zero or one (Optional) Default value is “summary”.	Zero or one (Optional) Valid values are: brief, summary, full Default value is: summary
ID	Comma separated list of anyURI	One (Mandatory)	One (Mandatory)

8.2.2.2.2 Response

The actual records returned by the catalogue should substitute for the element `<csw:AbstractRecord>` defined in the record.xsd schema. The following XML-Schema fragment defines the `GetRecordByIdResponseType` as part of this XML schema. This is simply the list of requested records:

Listing 9 - GetRecordByIdResponseType definition

```
<xsd:complexType name="GetRecordByIdResponseType"
  id="GetRecordByIdResponseType">
  <xsd:sequence>
    <xsd:element ref="csw:AbstractRecord" minOccurs="0"/>10
  </xsd:sequence>
</xsd:complexType>
```

Catalogues implementing this application profile should return the records in one of the schemas defined in section 7.4.

8.2.2.3 DescribeRecord Operation

The DescribeRecord operation allows a client to discover elements of the information model supported by the target catalogue service.

8.2.2.3.1 Request

Table 20 specifies the parameters for the `DescribeRecord` operation request (KVP and XML). The column **ISO App.-Profile** shows syntax and/or semantics restrictions or variations in comparison to that of the base spec. The encoding in the table is directly suitable for the HTTP GET binding.

The XML encoding of a valid response is specified in the HTTP binding of CSW 2.0 specification. Please refer to this section to determine the according XML schema.

¹⁰ The cardinality is an error resulting from the CSW base specification. It should be “maxOccurs=‘unbounded’”, but it’s not. Unfortunately, the profile cannot change this unless it is changed in the base specification.

Table 20 - Parameters in DescribeRecord operation request

Keyword	Datatype & Value	Optionality	ISO App.-Profile¹¹
REQUEST	Character String. Fixed values of “DescribeRecord”. The value is case insensitive.	Mandatory	Mandatory
SERVICE	Character String. Fixed values of “CSW”	Mandatory	Mandatory
VERSION	Character String. Fixed value of “2.0.0”	Mandatory	Mandatory
NAMESPACE	List of Character String, comma separated. Used to specify a namespace and its prefix. Format must be [<prefix>:]<url>. If the prefix is not specified then this is the default namespace.	Zero or one (Optional) Include value for each namespace If not included, all qualified names are in default namespace	Optional
TYPENAME	List of Character String, comma separated One or more qualified type names to be described	Zero or one (Optional) Default action is to describe all types known to server	Optional. One of: Service, Dataset, Dataset Collection, Application
OUTPUTFORMAT	Character String A MIME type indicating the format that the output document should have	Optional. Default value is “text/xml”.	Only “text/xml” supported.
SCHEMALANGUAGE	Character String	Zero or one (Optional) Default value is XMLSCHEMA	Only XMLSCHEMA is supported

8.2.2.3.2 Response

The <DescribeRecordResponse> element (see CSW 2.0) is the container for the <SchemaComponent> element, which contains the description in the requested schema language.

```
<xsd:complexType name="DescribeRecordResponseType">
  <xsd:sequence>
```

¹¹ 8.2.4.


```

        <xsd:element name="SchemaComponent"
                    type="csw:SchemaComponentType"
                    minOccurs="0"
                    maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SchemaComponentType" mixed="true">
  <xsd:sequence>
    <xsd:any namespace="##any" processContents="lax"/>
  </xsd:sequence>
  <xsd:attribute name="targetNamespace" type="xsd:anyURI"
                 use="required"/>
  <xsd:attribute name="parentSchema" type="xsd:anyURI"
                 use="optional"/>
  <xsd:attribute name="schemaLanguage" type="xsd:anyURI"
                 use="required"/>
</xsd:complexType>

```

If Typename is 'Dataset', 'Dataset Collection' or 'Application' the value of processContents will contain the schema for the ISO19115 data identification definition ('identification.xsd'), if Typename is 'Service' the value of processContents will contain the schema for the ISO19119 service identification definition ('iso19119.xsd'). In both cases parentSchema will hold a reference to the base schema of the whole information model (metadataEntity.xsd).

8.2.2.4 GetDomain Operation

The optional **GetDomain** operation is used to obtain runtime information about the range of values of a **metadata record element** or **request parameter**. The runtime range of values for a property or request parameter is typically much smaller than the value space for that property or parameter based on its static type definition.

This type of runtime information is useful for generating user interfaces with meaningful pick lists or for generating query predicates that have a higher chance of actually identifying a result set.

8.2.2.4.1 Request

Table 21 specifies the parameters for the **GetDomain** operation request (KVP and XML). The column **ISO App.-Profile** shows syntax and/or semantics restrictions or variations in comparison to those of the base spec. The encoding in the table is directly suitable for the HTTP GET binding.

The XML encoding of a valid response is specified in the HTTP binding of CSW 2.0 specification. Please refer to this section to determine the according XML schema.

Table 21 - Parameters in GetDomain operation request

Keyword	Datatype & Value	Optionality	ISO App.-Profile¹²
REQUEST	Character String. Fixed values of “GetDomain”. The value is case insensitive.	Mandatory	Mandatory
SERVICE	Character String. Fixed values of “CSW”	Mandatory	Mandatory
VERSION	Character String. Fixed value of “2.0.0”	Mandatory	Mandatory
PARAMETERNAME	List of Character String, comma separated Unordered list of names of an interface parameter, of the form <i>OperationName.Parameter Name</i>	Zero or one (Conditional) Include when PropertyName not included	Is not supported by this profile.
PROPERTYNAME	List of Character String, comma separated Unordered list of names of requested properties, from the information model that the catalogue is using.	Zero or one (Conditional) Include when ParameterName not included	Mandatory (Must be included by this profile, as PARAMETERNAME is not supported)

8.2.2.4.2 Response

The response is composed of one or more <DomainValues> elements (see CSW 2.0 specification).

The domain values may be a list of enumerated values (i.e. <ListOfValues>), one or more ranges of values (i.e. <RangeOfValues>), or a reference to some authoritative vocabulary (i.e. <ConceptualSchema>). An example of an authoritative vocabulary might be a standard list of animal and plant species names.

If the only child element of the <DomainValue> element is the <PropertyName> or <ParameterName> element, this shall be taken to mean that the catalogue was unable to determine anything about the specified property or parameter.

The XML encoding of a valid response is specified in the HTTP binding of CSW 2.0 specification. Please refer to this section to determine the appropriate XML schema.

¹² 8.2.4.

8.2.3 CSWT Manager Interface

The Manager Interface defines operations for creating, modifying and deleting catalogue records. This can either be done by a 'push' mechanism, the **Transaction** operation, or a 'pull' mechanism, implemented by the **Harvest** operation.

8.2.3.1 Transaction Operation

This is the 'push' mechanism, for creating, modifying and deleting catalogue records.

8.2.3.1.1 Request

In the following tables, the parameters for the **Transaction** operation request are specified.

There is no KVP encoding for transaction operation request, because there is no convenient way of encoding the transaction payloads using keyword-value pairs. Although only XML is supported, the parameters are presented in tabular form to give a better overview.

The XML encoding of a valid response is specified in the HTTP binding of CSW 2.0 specification. Please refer to this section to determine the appropriate XML schema.

The column **ISO App.-Profile** shows syntax and/or semantics restrictions or variations in comparison to those of the base specification.

Table 22 specifies only the base parameters. The <**Transaction**> element here defines an atomic unit of work and is a container for one or more insert, update and/or delete actions which are defined in the following tables.

The column **ISO App.-Profile** shows syntax and/or semantics restrictions or variations in comparison to those of the base specification.

Table 22 - Parameters in Transaction operation request

Keyword	Datatype & Value	Optionality	ISO App.-Profile ¹³
REQUEST	Character String. Fixed value of "Transaction". The value is case insensitive.	Mandatory	Mandatory
SERVICE	Character String. Fixed value of "CSW"	Mandatory	Mandatory
VERSION	Character String. Fixed value of "2.0.0"	Mandatory	Mandatory
TRANSACTION	Defines an atomic unit of work and is a container for one or more insert, update and/or delete actions, defined in Table 23.	One or more	One or more

¹³ 8.2.4.

Table 23 specifies the parameters which are valid for every insert, update and delete operation.

Table 23 - Parameters of Insert-,Update-, Delete-Operation

Keyword	Datatype & Value	Optionality	ISO App.-Profile ¹⁴
TRANSACTIONTYPE	CodeList One of “Insert”, “Update”, “Delete”	Mandatory	Mandatory
REQUESTID	May be used by a client application to associate a user-defined identifier with the operation.	Zero or one	Zero or one
VERBOSERESPONSE	May be used by a client to indicate to a server the amount of detail to generate in the rresponse. A value of FALSE means that a CSW should generate a terse or brief transaction response. A value of TRUE, or the absence of the attribute, means that the normal detailed transaction response should be generated.	Boolean, default “FALSE”	Boolean, default “FALSE”

Table 24 specifies the parameters which are valid for an insert-operation. The <Insert> element is a container for one or more records that are to be inserted into the catalogue. The schema of the record(s) must conform to the schema of the information model described using the DescribeRecord operation.

Table 24 - Parameters of Insert-Operation

Keyword	Datatype & Value	Optionality	ISO App.-Profile ¹⁵
MD_METADATA	Record(s) that conform to the schema of the information model described using the DescribeRecord operation.	Mandatory, one or more	Mandatory, one or more

¹⁴ 8.2.4.

¹⁵ 8.2.4.

Keyword	Datatype & Value	Optionality	ISO App.-Profile ¹⁵
HANDLE	Used to associate a mnemonic name for the purpose of error handling.	Zero or one	Zero or one

Table 25 specifies the parameters which are valid for an update-operation. If a complete record instance value (substituting for the <csw:AbstractRecord> element) is specified then the entire record in the catalogue shall be replaced by the value of MD_METADATA (<csw:Update>). If individual record property values are specified in MD_METADATA (<Update>), using the <csw:RecordProperty> element, then those individual property values of the catalogue record shall be updated.

The <csw:RecordProperty> element contains a <csw:Name> element and a <csw:Value> element. The <csw:Name> element is used to specify the name of the record property to be updated. The value of the <csw:Name> element may be a path expression to identify complex properties. The <csw:Value> element contains the value that will be used to update the record in the catalogue.

Table 25 - Parameters of Update-Operation

Keyword	Datatype & Value	Optionality	ISO App.-Profile ¹⁶
MD_METADATA	Accepts a complete record instance OR a list of one or more property names with their replacement values ¹⁷	Mandatory	Mandatory
TYPENAME	Character String	Zero or one	Zero or one of: Service, Dataset; Dataset Collection, Application
CONSTRAINTLANGUAGE	CodeList One of "CQL_TEXT" or "FILTER"	Must be specified with QUERYCONSTRAINT parameter.	Must be specified with QUERYCONSTRAINT parameter.

¹⁶8.2.4.

¹⁷ The name element is used to specify the name of the record property to be updated. The value of it may be a path expression to identify complex properties. The value element contains the value that will be used to update the record in the catalogue.

Keyword	Datatype & Value	Optionality	ISO App.-Profile ¹⁶
CONSTRAINT_LANGUAGE_VERSION	String May be used to specify a version number indicating which version of a specification the constraint conforms to. For example, if "FILTER" is being used, this parameter could be set to "1.0.0" indicating that the filter conforms to version 1.0.0 of the Filter Encoding Implementation Specification [OGC 02-059].	Mandatory	Mandatory
CONSTRAINT	String The predicate expression specified in the language indicated by the CONSTRAINTLANGUAGE parameter.	Mandatory.	Mandatory
HANDLE	Used to associate a mnemonic name for the purpose of error handling.	Zero or one	Zero or one

are affected by the constraint.

Table 26 specifies the parameters which are valid for a delete-operation. All record instances will be deleted which are affected by the constraint.

Table 26 - Parameters of Delete-Operation

Keyword	Datatype & Value	Optionality	ISO App.-Profile ¹⁸
TYPENAME	Character String	Zero or one	Zero or one of: Service, Dataset; Dataset Collection, Application

¹⁸ 8.2.4.

Keyword	Datatype & Value	Optionality	ISO App.-Profile ¹⁸
CONSTRAINTLANGUAGE	CodeList One of "CQL_TEXT" or "FILTER"	Must be specified with QUERYCONSTRAINT parameter.	Must be specified with QUERYCONSTRAINT parameter.
CONSTRAINT_LANGUAGE_VERSION	String May be used to specify a version number indicating which version of a specification the constraint conforms to. For example, if "FILTER" is being used, this parameter could be set to "1.0.0" indicating that the filter conforms to version 1.0.0 of the Filter Encoding Implementation Specification [OGC 02-059].	Mandatory	Mandatory
CONSTRAINT	String The predicate expression specified in the language indicated by the CONSTRAINTLANGUAGE parameter.	Mandatory.	Mandatory
HANDLE	Used to associate a mnemonic name for the purpose of error handling.	Zero or one	Zero or one

8.2.3.1.2 Response

The transaction response message conveys two pieces of information. First of all, it reports a summary of the transaction by indicating the number of records created, updated or deleted by the transaction. Secondly, the transaction response message indicates the results of each insert operation found in the transaction in the form of the <InsertResult> element.

The <InsertResult> element may appear zero or more times in the transaction response. It is used to report to the client a brief representation of each new record, including the record identifier, created in the catalogue.

The element <csw:AbstractRecord> of <TransactionResponseType> must be substituted by the brief resultset specified in section 7.4.1. The following schema fragment from CSW 2.0 base specification shows the structure of a corresponding response.

Listing 10 - TransactionResponse definition

```
<xsd:element name="TransactionResponse"
  type="csw:TransactionResponseType"/>
<xsd:complexType name="TransactionResponseType">
```

```

    <xsd:sequence>
      <xsd:element name="TransactionSummary"
        type="csw:TransactionSummaryType"/>
      <xsd:element name="InsertResult"
        type="csw:TransactionResultType"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="version" type="xsd:string"
      use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="TransactionSummaryType">
    <xsd:sequence>
      <xsd:element name="totalInserted"
        type="xsd:nonNegativeInteger"
        minOccurs="0"/>
      <xsd:element name="totalUpdated"
        type="xsd:nonNegativeInteger"
        minOccurs="0"/>
      <xsd:element name="totalDeleted"
        type="xsd:nonNegativeInteger"
        minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="requestId" type="xsd:anyURI"
      use="optional"/>
  </xsd:complexType>
  <xsd:complexType name="TransactionResultType">
    <xsd:sequence>
      <xsd:element ref="csw:AbstractRecord"
        maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="handleRef" type="xsd:anyURI"
      use="optional"/>
  </xsd:complexType>

```

The element **<csw:AbstractRecord>** shall be substituted with a 'brief' resultset of the inserted metadata record. The corresponding schema is defined in section 7.4.1.

The records must be reported in the same order in which the **<Insert>** elements appear in a transaction request and must map 1 to 1. Optionally, the handle attribute may be used to correlate a particular **<Insert>** element in the Transaction request with an **<InsertResult>** element found in the transaction response.

8.2.3.2 Harvest Operation

This is the pull mechanism that 'pulls' data into the catalogue. That is, this operation only references the data to be inserted or updated in the catalogue, and it is the job of the catalogue service to resolve the reference, fetch that data, and process it into the catalogue.

The **Harvest** operation has two modes of operation, controlled by a flag in the request. The first mode of operation is a synchronous mode in which the CSW receives a **Harvest** request from the client, processes it immediately, and sends the results to the client while the client waits. The second mode is asynchronous in that the server receives a **Harvest** request from the client, and

sends the client an immediate acknowledgement that the request has been successfully received (see CSW 2.0 specification).

8.2.3.2.1 Request

Table 27 specifies the parameters for the **Harvest** operation request (KVP and XML). The column **ISO App.-Profile** shows syntax and/or semantics restrictions or variations in comparison to that of the base spec. The encoding in the table is directly suitable for the HTTP GET binding. The schema for the XML encoding is defined in the CSW 2.0 specification.

Table 27 - Parameters in Harvest operation request

Keyword	Datatype & Value	Optionality	ISO App.-Profile ¹⁹
REQUEST	Character String. Fixed values of "HarvestRecords". The value is case insensitive.	Mandatory	Mandatory
SERVICE	Character String. Fixed values of "CSW"	Mandatory	Mandatory
VERSION	Character String. Fixed value of "2.0.0"	Mandatory	Mandatory
NAMESPACE	List of Character String, comma separated. Used to specify a namespace and its prefix. Format must be [<prefix>:]<url>. If the prefix is not specified then this is the default namespace.	Zero or one (Optional) Include value for each namespace If not included, all qualified names are in default namespace	Optional
SOURCE	URI Reference to the source from which the resource is to be harvested.	One (Mandatory)	One (Mandatory)

¹⁹ 8.2.4.

Keyword	Datatype & Value	Optionality	ISO App.-Profile ¹⁹
RESOURCETYPE	URI Reference to the type of resource being harvested. The Resource Type parameter is a reference to a schema document that defines the structure of the resource being harvested.	Zero or One (Optional)	Not supported, will be ignored.
RESOURCEFORMAT	Character String MIME type indicating format of the resource being harvested	Zero or one (Optional) Default value is <i>text/xml</i>	Only "text/xml" supported.
RESPONSEHANDLER	URL A reference to a person or entity that the CSW should respond to when it has completed processing Harvest request asynchronously (e.g. an email address)	Zero or one (Optional) If not included, process request synchronously	Zero or one (Optional) If not included, process request synchronously
HARVESTINTERVAL	Period Must conform to ISO8601 Period syntax.	Zero or one (Optional) If not specified, then harvest only once in response to the request.	Zero or one (Optional) If not specified, then harvest only once in response to the request.

8.2.3.2.2 Response

The **Harvest** operation can respond in one of two ways depending on the presence or absence of the **ResponseHandler** parameter.

If the **ResponseHandler** parameter is present, then the CSW server should verify the request syntax and immediately respond to the client with an acknowledgment message as defined in Subclause 10.12.4.4 of CSW 2.0. Later, after the server has processed the request, it should generate a **HarvestResponse** message and send it to the URI specified by the **ResponseHandler** parameter using the protocol encoded therein.

If the **ResponseHandler** parameter is not present, then the CSW server should process the **Harvest** request immediately and respond to the waiting client with a **HarvestResponse** message.

The following XML-Schema fragment defines the **HarvestResponse** message:

Listing 11 - HarvestResponse definition

```
<xsd:element name="Harvest" type="csw:HarvestType"/>
```

```

<xsd:complexType name="HarvestType">
  <xsd:complexContent>
    <xsd:extension base="csw:RequestBaseType">
      <xsd:sequence>
        <xsd:element name="Source" type="xsd:anyURI"/>
        <xsd:element name="ResourceType" type="xsd:anyURI"
          minOccurs="0"/>
        <xsd:element name="ResourceFormat" type="xsd:string"
          minOccurs="0" default="text/xml"/>
        <xsd:element name="HarvestInterval" type="xsd:duration"
          minOccurs="0"/>
        <xsd:element name="ResponseHandler" type="xsd:anyURI"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

If the **Harvest** attempt is successful, this response may include summary representations of the newly created or modified catalogue object(s). The response is the same as the **TransactionResponse**. In the TransactionResultType belonging to TransactionResponseType there is a reference used to report a brief representation of each new record, including the record identifier, created in the catalogue.

See Listing 10 for a definition of the response.

8.2.3.3 Record locking

This specification does not define a locking interface, instead relying on the underlying repository to mediate concurrent access to catalogue records.

8.2.4 Error handling

In the event that the submitted request is invalid, an exception report message is generated and returned to the client. This report complies with the definition of exception reports that are specified by OGC Common specification (see OGC 04-016r2). Table 28 shows an excerpt from that document that lists valid exception codes and meanings that are supported by this profile.

Table 28 - Exception codes and meanings (from OGC Common)

exceptionCode value	Meaning of code	“locator” value
OperationNotSupported	Request is for an operation that is not supported by this server	Name of operation not supported
MissingParameterValue	Operation request does not include a parameter value, and this server did not declare a default value for that parameter	Name of missing parameter
InvalidParameterValue	Operation request contains an invalid parameter value a	Name of parameter with invalid value
VersionNegotiationFailed	List of versions in “AcceptVersions” parameter value in GetCapabilities operation request did not include any version supported by this server	None, omit “locator” parameter

InvalidUpdateSequence	Value of (optional) updateSequence parameter in GetCapabilities operation request is greater than current value of service metadata updateSequence number	None, omit "locator" parameter
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit "locator" parameter
a When an invalid parameter value is received, it seems desirable to place the invalid value(s) in ExceptionText string(s) associated with the InvalidParameterValue value.		

The XML encoding of an exception has to comply with owsExceptionReport.xsd that is attached to the further mention OGC Common specification. See this document for details.

8.3 Query facilities

The interoperability goal is supported by the specification of a minimal abstract query (predicate) language, which must be supported by all compliant OpenGIS Catalogue Services. This query language is called OGC_Common Catalogue Query Language (CQL) (see CS 2.0 specification). It supports nested Boolean queries, text matching operations, temporal data types, and geospatial operators. CQL assists the consumer in the discovery of datasets of interest at all sites supporting the OpenGIS Catalogue Services.

OGC Filter Encoding 1.0.0 is an XML based encoding of the OGC_Common Query Language. This query language (and version) shall be supported by all catalogue instances of this profile in order to support search interoperability. In addition, the plain textual encoding of CQL version 2.0 may be supported. The capabilities document of the catalogue instance shall describe all supported query languages (and version).

The following conditions must be met by the supported query language:

- support for all comparison operators
- support for all logical operators
- support for the following expressions:
 - property name
 - literal
- support for the following spatial operators²⁰:
 - Intersects
 - Disjoint
 - BBOX

If the value of a search parameter in a metadata entry is a missing value, then the metadata set entry does not fulfill this search condition.

²⁰ all spatial operators must support geometries in WGS84 (see 7.2.3.1).

8.4 Implementation guidance

The following section gives developers help when setting up a catalogue service instance that complies with the defined application profile. Any information provided here is non-normative.

8.4.1 Technical issues

- HTTP: The base communication protocol is HTTP 1.1 as specified by IETF RFC 2616. All bindings of operations (see Table 16) MUST be consistent with HTTP/1.1 semantics. Alternative bindings may be specified for a specific service instance. Any HTTP/1.1 response message containing an entity-body must include a Content-Type header field defining the media type of that body (RFC 2616, 7.2.1). This includes the charset parameter (“application/xml; charset=utf-8”).
- SOAP: If SOAP messaging is supported, messages have to comply with SOAP 1.2 (<http://www.w3.org/TR/SOAP/>). The message payload is contained within the body part of the SOAP envelope.

8.4.2 Semantic issues

- Keywords: In metadata sets, keyword shall describe the essentials of the contents of the described resource. Ideally, the selected keywords shall comply with a given taxonomy that any catalogue uses for both metadata maintenance and queries.
- Furthermore, it is recommended taxonomies are supported on search queryables wherever possible
- Any identifier in a metadata set (most notably `MD_Metadata.fileIdentifier` and `MD_Metadata.parentIdentifier`) shall comply with a UUID (Universal Unique Identifier, as specified by <http://www.ietf.org/internet-drafts/draft-mealling-uuid-urn-03.txt>). The usage of UUIDs is strongly recommended to ensure the uniqueness of metadata sets across several catalogue services.
- The definition of the resource type in a metadata set should be done in the attribute `codeListValue` of the child element `MD_ScopeCode` of the element `hierarchyLevel`. Example:

```
<smXML:hierarchyLevel>
  <smXML:MD_ScopeCode
    codeList="http://metadata.dgiwg.org/codelistRegistry?
      MD_ScopeCode" codeListValue="application"/>
</smXML:hierarchyLevel>
```

Valid values for *hierarchyLevel* are:

- application
 - dataset
 - datasetcollection
 - service
- The following values should be used for the parameter `outputSchema` of a discovery operation:
 - csw:ogccore
 - csw:profile

- The following values should be used for the parameters *typenames* and *typename* of a operation:
 - csw:application
 - csw:service
 - csw:datasetcollection
 - csw:dataset
- The brief resultset which substitutes the **<csw:AbstractRecord>** of a transaction response (**<TransactionResultType>**) should include the **fileIdentifier** element.

8.4.3 Metadata set example

Listing 12 shows an example of a tightly coupled service description.

Listing 12 - Tightly coupled service description example²¹

```
<?xml version="1.0" encoding="UTF-8"?>
<iso19115full:MD_Metadata
xmlns:iso19115full="http://schemas.opengis.net/iso19115full"
xmlns:smXML="http://metadata.dgiwg.org/smXML"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="../iso/iso19115_full.xsd"
xmlns:iso19119="http://schemas.opengis.net/iso19119">
  <smXML:fileIdentifier>
    <smXML:CharacterString>5a389ad2-22dd-11d1-aa77-
002w2035b29092</smXML:CharacterString>
  </smXML:fileIdentifier>
  <smXML:language>
    <smXML:CharacterString>de</smXML:CharacterString>
  </smXML:language>
  <smXML:hierarchyLevel>
    <smXML:MD_ScopeCode codeList="http://www.altova.com"
codeListValue="service"/>
  </smXML:hierarchyLevel>
  <smXML:contact>
    <smXML:CI_ResponsibleParty>
      <smXML:role>
        <smXML:CI_RoleCode codeList="http://www.example.com"
codeListValue="author"/>
      </smXML:role>
    </smXML:CI_ResponsibleParty>
  </smXML:contact>
  <smXML:dateStamp>
    <smXML:Date>2004-12-22</smXML:Date>
  </smXML:dateStamp>
  <smXML:identificationInfo>
    <iso19119:CSW_ServiceIdentification>
      <smXML:citation>
        <smXML:CI_Citation>
          <smXML:title>
```

²¹ Correct syntax, content makes no sense here.

```

        <smXML:CharacterString>This is the
title</smXML:CharacterString>
    </smXML:title>
    <smXML:alternateTitle>
        <smXML:CharacterString>This is the alternate
title</smXML:CharacterString>
    </smXML:alternateTitle>
    <smXML:date>
        <smXML:CI_Date>
            <smXML:date>
                <smXML:Date>2004-05-30</smXML:Date>
            </smXML:date>
            <smXML:dateType>
                <smXML:CI_DateTypeCode
codeList="http://www.example.com" codeListValue="revision"/>
            </smXML:dateType>
        </smXML:CI_Date>
    </smXML:date>
    <smXML:date>
        <smXML:CI_Date>
            <smXML:date>
                <smXML:Date>2004-01-01</smXML:Date>
            </smXML:date>
            <smXML:dateType>
                <smXML:CI_DateTypeCode
codeList="http://www.example.com" codeListValue="creation"/>
            </smXML:dateType>
        </smXML:CI_Date>
    </smXML:date>
    <smXML:identifier>
        <smXML:MD_Identifier>
            <smXML:code>
                <smXML:CharacterString>5a389ad2-22dd-11d1-aa77-
002035fssdf92</smXML:CharacterString>
            </smXML:code>
        </smXML:MD_Identifier>
    </smXML:identifier>
    <smXML:citedResponsibleParty>
        <smXML:CI_ResponsibleParty>
            <smXML:organisationName>
                <smXML:CharacterString>company
GmbH</smXML:CharacterString>
            </smXML:organisationName>
            <smXML:role>
                <smXML:CI_RoleCode codeList=""
codeListValue="provider"/>
            </smXML:role>
        </smXML:CI_ResponsibleParty>
    </smXML:citedResponsibleParty>
    <smXML:citedResponsibleParty>
        <smXML:CI_ResponsibleParty>
            <smXML:individualName>
                <smXML:CharacterString>Uwe
Voges</smXML:CharacterString>
            </smXML:individualName>
            <smXML:role>

```

```

        <smXML:CI_RoleCode codeList=""
codeListValue="originator"/>
        </smXML:role>
    </smXML:CI_ResponsibleParty>
</smXML:citedResponsibleParty>
</smXML:CI_Citation>
</smXML:citation>
<smXML:abstract>
    <smXML:CharacterString>abstract</smXML:CharacterString>
</smXML:abstract>
<smXML:descriptiveKeywords>
    <smXML:MD_Keywords>
        <smXML:keyword>
            <smXML:CharacterString>blau</smXML:CharacterString>
        </smXML:keyword>
        <smXML:keyword>
            <smXML:CharacterString>rot</smXML:CharacterString>
        </smXML:keyword>
    </smXML:MD_Keywords>
    <smXML:type>
        <smXML:MD_KeywordTypeCode codeList=""
codeListValue="farben"/>
    </smXML:type>
</smXML:MD_Keywords>
</smXML:descriptiveKeywords>
<smXML:descriptiveKeywords>
    <smXML:MD_Keywords>
        <smXML:keyword>
            <smXML:CharacterString>keyword1</smXML:CharacterString>
        </smXML:keyword>
        <smXML:keyword>
            <smXML:CharacterString>keyword2</smXML:CharacterString>
        </smXML:keyword>
    </smXML:MD_Keywords>
</smXML:descriptiveKeywords>
<smXML:resourceConstraints>
    <smXML:MD_SecurityConstraints>
        <smXML:classification>
            <smXML:MD_ClassificationCode codeList=""
codeListValue="xxx"/>
        </smXML:classification>
    </smXML:MD_SecurityConstraints>
</smXML:resourceConstraints>
<iso19119:serviceType>
    <smXML:CharacterString>WMS</smXML:CharacterString>
</iso19119:serviceType>
<iso19119:serviceTypeVersion>
    <smXML:CharacterString>1.3</smXML:CharacterString>
</iso19119:serviceTypeVersion>
<iso19119:operationMetadata>
    <iso19119:SV_OperationMetadata>
        <iso19119:operationName>
            <smXML:CharacterString>GetMap</smXML:CharacterString>
        </iso19119:operationName>
        <iso19119:DCP>
            <iso19119:SV_DCPList codeList="http://www.example.com"
codeListValue="HTTPGet"/>
        </iso19119:DCP>
    </iso19119:SV_OperationMetadata>
</iso19119:operationMetadata>

```



```

    <iso19119:connectPoint>
      <smXML:CI_OnlineResource>
        <smXML:linkage>
          <smXML:URL>http://www.mywms.com/wms</smXML:URL>
        </smXML:linkage>
      </smXML:CI_OnlineResource>
    </iso19119:connectPoint>
    <iso19119:parameters>
      <iso19119:SV_Parameter>
        <iso19119:name>
          <smXML:MemberName>
            <smXML:aName>

<smXML:CharacterString>Request</smXML:CharacterString>
          </smXML:aName>
          <smXML:attributeType>
            <smXML:TypeName>
              <smXML:aName>
                <smXML:CharacterString/>
              </smXML:aName>
            </smXML:TypeName>
          </smXML:attributeType>
        </smXML:MemberName>
      </iso19119:name>
    <iso19119:direction>

    <iso19119:SV_ParameterDirection>in</iso19119:SV_ParameterDirection>
    </iso19119:direction>
    <iso19119:description>
      <smXML:CharacterString>OGC WMS GetMap request
name</smXML:CharacterString>
    </iso19119:description>
    <iso19119:optionality>

    <smXML:CharacterString>Mandatory</smXML:CharacterString>
    </iso19119:optionality>
    <iso19119:repeatability>
      <smXML:Boolean>>false</smXML:Boolean>
    </iso19119:repeatability>
    </iso19119:SV_Parameter>
  </iso19119:parameters>
</iso19119:SV_OperationMetadata>
</iso19119:operationMetadata>
<iso19119:operatesOn>
  <smXML:Reference uuidref="5a389ad2-22dd-11d1-aa77-
002035b29093"/>
</iso19119:operatesOn>
<iso19119:extent>
  <smXML:EX_Extent>
    <smXML:geographicElement>
      <smXML:EX_GeographicBoundingBox>
        <smXML:extentTypeCode>
          <smXML:Boolean>>true</smXML:Boolean>
        </smXML:extentTypeCode>
        <smXML:westBoundLongitude>
          <smXML:approximateLongitude>-
180.00</smXML:approximateLongitude>

```

```

        </smXML:westBoundLongitude>
        <smXML:eastBoundLongitude>

<smXML:approximateLongitude>180.00</smXML:approximateLongitude>
        </smXML:eastBoundLongitude>
        <smXML:southBoundLatitude>
            <smXML:approximateLatitude>-
90.00</smXML:approximateLatitude>
        </smXML:southBoundLatitude>
        <smXML:northBoundLatitude>

<smXML:approximateLatitude>90.00</smXML:approximateLatitude>
        </smXML:northBoundLatitude>
        </smXML:EX_GeographicBoundingBox>
    </smXML:geographicElement>
</smXML:EX_Extent>
</iso19119:extent>
<iso19119:coupledResource>
    <iso19119:CSW_CoupledResource>
        <iso19119:operationName>
            <smXML:CharacterString>GetMap</smXML:CharacterString>
        </iso19119:operationName>
        <iso19119:identifier>
            <smXML:CharacterString>5a389ad2-22dd-11d1-aa77-
002035b29093</smXML:CharacterString>
        </iso19119:identifier>
    </iso19119:CSW_CoupledResource>
</iso19119:coupledResource>
<iso19119:couplingType>
    <iso19119:CSW_CouplingType codeList=""
codeListValue="tight"/>
</iso19119:couplingType>
</iso19119:CSW_ServiceIdentification>
</smXML:identificationInfo>
<smXML:distributionInfo>
    <smXML:MD_Distribution>
        <smXML:distributionFormat>
            <smXML:MD_Format>
                <smXML:name>
                    <smXML:CharacterString>jpeg</smXML:CharacterString>
                </smXML:name>
                <smXML:version>
                    <smXML:CharacterString>1.0</smXML:CharacterString>
                </smXML:version>
            </smXML:MD_Format>
        </smXML:distributionFormat>
    </smXML:MD_Distribution>
</smXML:distributionInfo>
</iso19115full:MD_Metadata>

```

8.5 Security considerations

This document does not demand any specific security considerations regarding a compliant catalogue service. Security issues are part of the implementation specification of a catalogue service.

But it is recommended that HTTP Basic Authentication is used to prevent access to the URLs of the transaction interface, unless the requestor can provide user/password credentials. This basic authentication should be used in conjunction with HTTPS as part of a security solution.

Annex A (normative)

Abstract test suite

A.1 Read-only CSW

A.1.1 Basic CSW Client

A.1.1.1 Basic service elements

- a) Test purpose: Verify that a CSW client satisfies the requirements for request parameter rules.
- b) Test method: Generate an adequate sample of requests from the client and verify that each is a valid request.
- c) Reference: 8.1; CSW 2.0 base specification
- d) Test type: Basic

A.1.1.2 GetCapabilities Request

- a) Test purpose: Verify that a CSW client satisfies all requirements for a GetCapabilities request.
- b) Test method: Generate an adequate sample of GetCapabilities requests from the client and verify that each is a valid request.
- c) Reference: 8.2.1.1
- d) Test type: Basic

A.1.1.3 GetRecords Request

- a) Test purpose: Verify that a CSW client satisfies all requirements for a GetRecords request.
- b) Test method: Generate an adequate sample of GetRecords requests from the client and verify that each is a valid request.
- c) Reference: 8.2.2.1.1
- d) Test type: Basic

A.1.1.4 GetRecordById Request

- a) Test purpose: Verify that a CSW client satisfies all requirements for a GetRecordById request.
- b) Test method: Generate an adequate sample of GetRecordById requests from the client and verify that each is a valid request.
- c) Reference: 8.2.2.2.1
- d) Test type: Basic

A.1.1.4 DescribeRecord Request

- a) Test purpose: Verify that a CSW client satisfies all requirements for a DescribeRecord request.
- b) Test method: Generate an adequate sample of DescribeRecord requests from the client and verify that each is a valid request.
- c) Reference: 8.2.2.3.1
- d) Test type: Basic

A.1.1.4 GetDomain Request

- a) Test purpose: Verify that a basic CSW client satisfies all requirements for a GetCapabilities request.
- b) Test method: Generate an adequate sample of GetCapabilities requests from the client and verify that each is a valid request.

- c) Reference: 8.2.2.4.1
- d) Test type: Basic

A.1.2 CSW Server

A.1.2.1 Version negotiation

- a) Test Purpose: Verify that a CSW server interface satisfies the requirements for version negotiation.
- b) Test Method: Submit requests containing version number both lower than and higher than the version supported by the server. Verify that the server responses is in accord with the rules for version negotiation.
- c) Reference: 8.2.1.1; CSW 2.0 base specification
- d) Test Type: Basic

A.1.2.2 Request parameter rules

- a) Test Purpose: Verify that a CSW server interface satisfies the requirements for request parameter rules.
- b) Test Method: Generate a sample of requests from a client. Include both invalid requests and valid request that vary within the limits allowed by the rules. Verify that the server provides an appropriate response in each case.
- c) Reference: 8.2.1; CSW 2.0 base specification
- d) Test Type: Basic

A.1.2.3 GetCapabilities response

- a) Test Purpose: Verify that a basic CSW server interface satisfies all requirements of the GetCapabilities operation.
- b) Test Method: Make several GetCapabilities requests using a variety of input parameters. Verify that an appropriate response is returned in each case.
- c) Reference: 7.5; 8.2.1.1
- d) Test Type: Basic

A.1.2.3 GetRecords response

- a) Test Purpose: Verify that a basic CSW server interface satisfies all requirements of the GetRecords operation.
- b) Test Method: Make several GetRecords requests using a variety of input parameters. Verify that an appropriate response is returned in each case.
- c) Reference: 8.2.2.1.2
- d) Test Type: Basic

A.1.2.3 GetRecordById response

- a) Test Purpose: Verify that a basic CSW server satisfies all requirements of the GetRecordsById operation.
- b) Test Method: Make several GetRecordsById requests using a variety of input parameters. Verify that an appropriate response is returned in each case.
- c) Reference: 8.2.2.2.2
- d) Test Type: Basic

A.1.2.3 DescribeRecord response

- a) Test Purpose: Verify that a basic CSW server interface satisfies all requirements of the DescribeRecord operation.
- b) Test Method: Make several DescribeRecord requests using a variety of input parameters. Verify that an appropriate response is returned in each case.
- c) Reference: 8.2.2.3.2

d) Test Type: Basic

A.1.2.3 GetDomain response

a) Test Purpose: Verify that a basic CSW server interface satisfies all requirements of the GetDomain operation.

b) Test Method: Make several GetDomain requests using a variety of input parameters. Verify that an appropriate response is returned in each case.

c) Reference: 8.2.2.4.2

d) Test Type: Basic

A.2 Transactional CSW

A.2.1 Client

A.2.1.1 Transaction request

a) Test Purpose: Verify that a CSW client satisfies all requirements for a Transaction request.

b) Test Method: Generate an adequate sample of Transaction requests from the client and verify that each is a valid request.

c) Reference: 0

d) Test Type: Basic

A.2.1.1 Harvest request

a) Test Purpose: Verify that a CSW client satisfies all requirements for a Harvest request.

b) Test Method: Generate an adequate sample of Harvest requests from the client and verify that each is a valid request.

c) Reference: 8.2.3.2.1

d) Test Type: Basic

A.2.2 CSW Server

A.2.1.1 Transaction response

a) Test Purpose: Verify that a CSW server interface satisfies all requirements for a Transaction operation.

b) Test Method: Make several Transaction requests using a variety of input parameters. Verify that an appropriate response is returned in each case.

c) Reference: 8.2.3.1.2

d) Test Type: Basic

A.2.1.1 Harvest response

a) Test Purpose: Verify that a CSW server interface satisfies all requirements for a Harvest operation.

b) Test Method: Make several Harvest requests using a variety of input parameters. Verify that an appropriate response is returned in each case.

c) Reference: 8.2.3.2.2

d) Test Type: Basic

Annex B (normative)

Design rationale

The CSW catalogue profile at hand is intended to provide a catalogue service for managing metadata resources that comply with ISO19115:2003 and ISO19119:2003. Based on this information model many georesources might be described and managed:

- Geodata:
 - dataset, that is an identifiable collection of data;
 - dataset collection, that are a collection of datasets sharing the same product specification;
- Services that is an instance of a service hosted on a specific set of hardware and accessible over a network. A service is either tightly or loosely coupled.
 - Tightly coupled, that is a service that is associated with a specific dataset or dataset collection.
 - Loosely coupled, that is a service that is not associated with a specific dataset or dataset collection.
- Applications, that is an information resource that is hosted on a specific set of hardware and accessible over a network

These are the mandatory information resources that are managed by the catalogue service. According to ISO19115:2003, additional resources might be described and thus be managed by the catalogue service.

The catalogue service specifies the HTTP protocol binding and is thus accessible over the Internet. It provides powerful search capabilities, including spatial searches.

One major design goal is to provide interoperability at the highest grade between distinct catalogue services. This level of interoperability is either realised between catalogue services that comply with this application profile, or between any catalogue services that comply with the base specification of CSW 2.0.

Date: March 22, 2004
CSW 2.0

ISO19115/ISO19119 Application Profile for

Annex C
(normative)

WSDL Specification CSW Profile