

Testbed-12 Web Integration Service

Table of Contents

1. Introduction	7
1.1. Scope	7
1.2. Document contributor contact points	7
1.3. Change requests	7
1.4. Future Work	7
1.5. Foreword	8
2. References	9
3. Terms and definitions.	10
3.1. attribute <XML>	10
3.2. base URL	10
3.3. client	10
3.4. element <XML>	10
3.5. feature	10
3.6. feature identifier	10
3.7. filter expression	10
3.8. interface	11
3.9. Multipurpose Internet Mail Extensions (MIME) type	11
3.10. namespace <XML>	11
3.11. operation	11
3.12. property	11
3.13. resource	11
3.14. remote resource	11
3.15. request	11
3.16. response	12
3.17. schema	12
3.18. schema <XML Schema>	12
3.19. server	12
3.20. service	12
3.21. service metadata capabilities document	12
3.22. Uniform Resource Identifier	12
4. Abbreviated terms	13
5. UML notation	14
6. Overview	15
7. Requirements	16
7.1. Status Quo	16
7.2. Requirements Statement	16
8. Web Integration Service	17
8.1. GetCapabilities operation	17

8.1.1. Introduction	17
8.1.2. Request	17
8.1.3. Response	18
8.1.4. NoOp operation	19
8.1.5. Security considerations	19
8.1.6. Relationship to CSW	20
8.1.7. Example	20
9. GetAssociations operation	25
9.1. Introduction	25
9.2. Associations	25
9.2.1. Introduction	25
9.2.2. Types of association resources	25
9.2.3. UML	26
9.2.4. XML-encoding	26
9.2.5. Associations container	28
9.2.6. MIME type	29
9.3. Standard identifiers	29
9.4. GetAssociations operation	33
9.4.1. Introduction	33
9.4.2. UML	33
9.4.3. KVP-encoding	34
9.4.4. XML-encoding	35
9.4.5. /associations resource (REST)	35
9.4.6. Parameter discussion	35
9.4.7. Response	36
9.4.8. Example	36
9.5. Harvesting associations into an ebRIM CSW catalogue	37
10. Catalogue extension package	39
10.1. Introduction	39
10.2. Identifiers	39
10.2.1. Introduction	39
10.2.2. Identifiers for OGC web services for ebRIM	39
10.2.3. Identifier for the content of OGC web services for ebRIM	39
10.2.4. Identifiers for OGC documents for ebRIM	40
10.2.5. Identifiers for associations for ebRIM	41
10.2.6. Identifiers for ebRIM slot names	41
10.2.7. Identifiers for data types for ebRIM	43
10.3. Classification schemes for ebRIM	44
10.4. Mapping OGC capabilities documents into ebRIM	44
10.4.1. Introduction	44
10.4.2. ServiceIdentification section	44

10.4.3. ServiceProvider section	45
10.4.4. OperationsMetadata section	46
10.4.5. Content section	46
10.4.6. WIS content	46
10.5. GetAssociations stored query	51
11. Security considerations	52
11.1. Advertising Security	52
11.2. Storing Security information in an ebRIM CSW Catalogue.	52
11.3. Public and Private WIS Service objects	53
12. CompuSult WIS implementation details	55
12.1. Web Integration Service endpoint	55
12.2. Security	55
Appendix A: XML Schema Documents	56
Appendix B: OGC Service Integration ebRIM Package	64
Appendix C: Review comments from TB12 participants	96
Appendix D: Revision History	99

Publication Date: 2017-03-10

Approval Date: 2016-12-07

Posted Date: 2016-10-31

Reference number of this document: OGC 16-043

Reference URL for this document: <http://www.opengis.net/doc/PER/t12-A078>

Category: Public Engineering Report

Editor: Panagiotis (Peter) A. Vretanos

Title: Testbed-12 Web Integration Service

OGC Engineering Report

COPYRIGHT

Copyright © 2017 Open Geospatial Consortium. To obtain additional rights of use, visit <http://www.opengeospatial.org/>

WARNING

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

LICENSE AGREEMENT

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD. THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by

destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Abstract

For many years OGC has been developing a suite of standards defining web services interfaces and encodings for geospatial processing. The suite includes a Web Map Service (WMS), a Web Map Tiling Service (WMTS), a Web Feature Service (WFS), a Web Coverage Service (WCS), a Web Catalogue Service (CSW), the Sensor Web (SWE) suite of services, etc. These service interfaces and their implementations have, more or less, been developed independently of one another resulting in isolation and poor integration between them. For example, consider a map generated by a WMS. A client or user cannot easily determine which source data was used to create the map and how to download that source data through an OGC data service such as WFS or WCS. Furthermore when one considers the Publish-Find-Bind paradigm, OGC can only partially support the full potential of this paradigm. This is because OGC structured catalogues can only register services in isolation of other related services and cannot automatically determine the relationships among services and the resources they offer.

In order to achieve better integration between OGC web services and enhance the publish-find-bind paradigm, this OGC Engineering Report defines and discusses three key elements. These are:

1. Defining a new service, called the Web Integration Service (WIS), which allows for the discovery and access to integrated sets of OGC web services deployed at an endpoint.
2. Specifying a means of discovering and describing associations between web resources (both OGC and non-OGC).
3. Defining extensions to the OGC catalogue to allow the service to harvest and make discoverable a rich set of linked OGC and non-OGC resources.

The Web Integration Service (WIS) is an aggregation service whose only purpose is to provide a list of references to a suite of other, perhaps related OGC services available at an endpoint.

A new operation, named GetAssociations, is defined as an extension such that existing OGC services (WMS, WFS, WCS, etc.) may implement this operation in order to support rich auto-discovery. This operation enables OGC web services to externalize their internal association knowledge about their content and relationships to other OGC and external resources. For example, a WMS would know if the source data for a layer it offers is a Shapefile, or a WFS feature type,

or another WMS layer (i.e. cascading), or if a WMTS layer exists that renders the same information more efficiently. This "internal knowledge" can now be externalized via the GetAssociations operation.

Currently, OGC Catalogues Service instances can harvest the capabilities document of an OGC web service, register that service, register the existence of the individual offerings that the service offers and also register the association between the service and the content it offers. Thus, the entire harvesting process is focused on a single OGC web service and consequently offers a limited scope of discovery. In order to support rich discovery, a catalogue needs to be able to automatically register services found at an endpoint as well as register all known associations among those services, their offerings and other OGC and non-OGC resources. This involves harvesting a service's capabilities document to determine what content the service offers but it also involves further interrogating the service to determine of what (if any) other associations it is aware. Populated with this enhanced knowledge a client may now use a catalogue to, for example, find the description of feature data and then be able to find the WFS that offer that data, a WMS that renders those features into a map, a WMTS that has a tiled representation of that data, etc. In order to support this kind of rich discovery, a new CSW-ebRIM package is specified that defines ebRIM object types, associations, classifications and stored queries that support the description of integrated OGC web service and their artifacts within the catalogue.

Business Value

This OGC Engineering Report is of value because the research and recommended solution detailed in this document would allow the community to break down the OGC web service stovepipes that have emerged over the past decade. With the existence of the Web Integration Service, the GetAssociations operation and the extensions to the OGC CSW-ebRIM Catalogue, the entire publish-bind-find cycle is significantly enhanced because:

- Publishing becomes more automatic as catalogues can now access the web integration service to determine which OGC web services are deployed at an endpoint
- Finding (i.e. Discovery) is significantly enhanced because OGC web services now advertise all the associations they are aware of allowing catalogues to harvest this information and thus provide much richer discovery capabilities

What does this ER mean for the Working Group and OGC in general

The primary subject of this Engineering Report is to define methods and apparatus, via the Web Integration Service, to better integrate a loosely coupled set of OGC services. Such a proposal and related discussion is within the purview of the OGC Architecture DWG because the overarching nature of the work presented in the Engineering Report (ER) is germane to multiple OGC standards.

This ER defines a generic discovery operation named GetAssociations. The function of this operation is to allow services to make accessible their internal knowledge of associated OGC and non-OGC resources resulting in a richer set of metadata describing each service. For maximum effect, this operation needs to be defined for all OGC web services and thus the content of this engineering report needs to be reviewed by the OWS Common 1.2 SWG for inclusion into the next version of that standard.

Finally, this engineering report defines an extension package for CSW-ebRIM (see OGC 07-110r4). The function of the package is to instrument a CSW-ebRIM catalogue with the necessary identifiers, associations and classification schemes to allow catalogues to harvest and properly register the richer set of metadata made available via the web integration service and the GetAssociations operation. These extensions need to be reviewed and commented on by the ebRIM AP of CSW SWG.

How does this ER relate to the work of the Working Group

The Architecture DWG considers overarching architectural issues that are germane to multiple OGC standards, including mechanisms for describing and invoking services in a heterogeneous distributed network.

The work described in this ER is clearly of an overarching nature relating to multiple standards and describing a mechanism for integrating, associating and discovering multiple OGC web services and other OGC and non-OGC resources.

Keywords

WIS, web integration service, associations, GetAssociations, identifiers, catalogue, rich discovery, loosely couples, stovepipe

Proposed OGC Working Group for Review and Approval

This ER shall be submitted to the Architecture DWG, OWS Common 1.2 SWG, and the ebRIM AP of CSW SWG for review and comment.

Chapter 1. Introduction

1.1. Scope

This engineering report defines a new OGC service named the Web Integration Service. A web integration service in a aggregation service whose purpose it to provide, for the purpose of discovery, a list of reference to a suite of other, perhaps related, OGC services.

This engineering report defines methods and apparatus to allow OGC web service to externalize this internal association knowledge. This engineering report describes how associations between OGC and non-OGC resource can be represented in XML. This engineering report further defines a new operation; named GetAssociations, that existing OGC web services can implement to allow client applications to interrogate an OGC web service about its internal association knowledge.

This engineering report defines an extension package for the CSW-ebRIM profile (see OGC 07-110r4). The profile encapsulates all the identifiers, associations and classification schemes necessary to allow an OGC CSW-ebRIM catalogue to register OGC web services and, by means of the GetAssociation operation, register the resources those services offer and the associations between those services, their offerings and other OGC and non-OGC resources.

1.2. Document contributor contact points

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

Table 1. Contacts

Name	Organization
Craig Coffin	Compusult
Mark Lawrence	Compusult
Panagiotis (Peter) A. Vretanos	CubeWerx Inc.

1.3. Change requests

- "Consider making the GetResourceById() operation mandatory for all OGC data services"
 - http://ogc.standardstracker.org/show_request.cgi?id=412

1.4. Future Work

- Define a JSON-LD output format for the GetAssociations operation
- Harmonize the identifiers defined herein with similar identifiers found in other OGC specifications
- Define an additional, optional, conformance class for WIS's that implement the GetResourceById() and/or GetAssociations() operations. The WIS would need to cascade those requests down to the services that it is aggregating.

- Extended security for WIS content
 - Extend the security implementation to determine the visibility of service endpoints in the WIS capabilities based on user authorization. Propose solutions to reconcile authentication and authorization for the WIS, catalog and services.
- Approaches for filtering of WIS content
 - It may be useful for a content provider to expose subsets of services through the WIS. For example, an environmental data provider may wish to provide separate entry points for oceanographic and atmospheric data, along with an additional entry point to expose both data holdings.
 - A direct solution might be to stand up a separate WIS instance for each data holding. However, care must be taken to avoid collision between instances, while minimizing redundancy and resource usage.
 - It may be preferable to extend the specification so that a single instance can support subsets of services. A "passive" approach is for the WIS capabilities response to contain groupings or metadata attributes to categorize services. The client would be responsible for processing the response to identify the groupings. A more active approach is to implement a parameter or operation allowing a client to specify a simple category filter in the service request.
- Software Tools
 - For a service implementation that does more than simply returning a static capabilities response, software tools may aid in configuration of the service content. It may be useful to implement an interface to manage the list of service endpoints and the metadata cache, including the ability to add/remove endpoints, restrict access to content, and configure cache settings.
 - Tools to facilitate the creation of service associations may help speed the adoption of the Web Integration Service. A software library could provide developers with an easy means of implementing the GetAssociations operation. A configuration tool for this library could be used for data providers to quickly configure association metadata for their services.

1.5. Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

Chapter 2. References

The following documents are referenced in this document. 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.

- OGC 06-121r9, OGC® Web Services Common Standard

NOTE: This OWS Common Standard contains a list of normative references that are also applicable to this Implementation Standard.

- OGC 07-110r4, CSW-ebRIM Registry Service - Part 1: ebRIM profile of CSW
- OGC 07-144r4, CSW-ebRIM Registry Service - Part 2: Basic extension package CAT2 ebRIM part2
- OGC 09-026r2, OGC Filter Encoding 2.0 Encoding Standard - With Corrigendum
- OGC 09-110r4, OGC® WCS 2.0 Interface Standard- Core: Corrigendum
- OGC 09-147r3, OGC® Web Coverage Service 2.0 Interface Standard - KVP Protocol Binding Extension - Corrigendum
- OGC 09-148r1, OGC® Web Coverage Service 2.0 Interface Standard - XML/POST Protocol Binding Extension
- OGC 09-025r2, OGC® Web Feature Service 2.0 Interface Standard – With Corrigendum
- OGC 12-168r6, OGC® Catalogue Services 3.0 - General Model v3.0
- OGC 12-176r7, OGC® Catalogue Specification - HTTP protocol binding v3.0
- OGC 06-042, OpenGIS Web Map Service (WMS) Implementation Specification
- OGC 07,057r7, OpenGIS Web Map Tile Service Implementation Standard
- OGC 05-077r7, Web Processing Service

Chapter 3. Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9] shall apply. In addition, the following terms and definitions apply.

3.1. attribute <XML>

name-value pair contained in an [element <XML>](#) (see ISO 19136:2007, definition 4.1.3)

NOTE In this document an attribute is an XML attribute unless otherwise specified.

3.2. base URL

a URL prefix to which additional parameters are appended in order to construct a valid KVP-encoded OGC web service request or the URL to which XML-encoded OGC web service requests may be submitted using the HTTP POST method for processing

3.3. client

software component that can invoke an [operation](#) from a [server](#) (see ISO 19128:2005, definition 4.1)

3.4. element <XML>

basic information item of an XML document containing child elements, [attributes](#) and character data (see ISO 19136:2007, definition 4.1.23)

3.5. feature

abstraction of real world phenomena (see ISO 19101:2002, definition 4.11)

NOTE A feature can occur as a type or an instance. The term "feature type" or "feature instance" should be used when only one is meant.

3.6. feature identifier

identifier that uniquely designates a [feature](#) instance

3.7. filter expression

predicate expression encoded using XML (see OGC 09-026r2, definition 4.11)

3.8. interface

named set of [operations](#) that characterize the behaviour of an entity (see ISO 19119:2005, definition 4.2)

3.9. Multipurpose Internet Mail Extensions (MIME) type

media type and subtype of data in the body of a message that designates the native representation (canonical form) of such data (see IETF RFC 2045:1996)

3.10. namespace <XML>

collection of names, identified by a URI reference which are used in XML documents as [element <XML>](#) names and [attribute <XML>](#) names (see W3C XML Namespaces:1999)

3.11. operation

specification of a transformation or query that an object may be called to execute (see ISO 19119:2005, definition 4.3)

3.12. property

facet or attribute of an object, referenced by a name (see OGC 09-026r2, definition 4.21)

3.13. resource

asset or means that fulfills a requirement (see ISO 19115:2003, definition 4.10)

NOTE

In this International Standard, the resource is a [feature](#), or any identifiable component of a feature (e.g. a property of a feature)

3.14. remote resource

resource that is not under direct control of a system

NOTE

In this International Standard, the system is a web feature service. The resource is not held in any data store that is directly controlled by that service and thus cannot be directly retrieved by the service.

3.15. request

invocation of an [operation](#) by a [client](#) (see ISO 19128:2005, definition 4.10)

3.16. response

result of an [operation](#) returned from a [server](#) to a [client](#) (see ISO 19128:2005, definition 4.11)

3.17. schema

formal description of a model (see ISO 19101:2002, definition 4.25)

NOTE

In general, a schema is an abstract representation of an object's characteristics and relations to other objects. An XML schema represents the relationship between the [attributes](#) and [elements](#) of an XML object (for example, a document or a portion of a document).

3.18. schema <XML Schema>

collection of [schema](#) components within the same target [namespace <XML>](#) (see ISO 19136:2007, definition 4.1.54)

3.19. server

particular instance of a [service](#) (see ISO 19128:2005, definition 4.12)

3.20. service

distinct part of the functionality that is provided by an entity through [interfaces](#) (see ISO 19119:2005, definition 4.1)

3.21. service metadata | capabilities document

metadata describing the [\[operations\]](#) and geographic information available at a [server](#) (see ISO 19128:2005, definition 4.14)

3.22. Uniform Resource Identifier

unique identifier for a resource, structured in conformance with IETF RFC 2396 (see ISO 19136:2007, definition 4.1.65)

NOTE

The general syntax is <scheme>::<scheme-specified-part>. The hierarchical syntax with a [namespace <XML>](#) is: <scheme>://<authority><path>?<query> Conventions

Chapter 4. Abbreviated terms

Table 2. Abbreviations

API	Application Program Interface
COTS	Commercial Off The Shelf
DCE	Distributed Computing Environment
DCOM	Distributed Component Object Model
IDL	Interface Definition Language
CGI	Common Gateway Interface
CRS	Coordinate Reference System
DCP	Distributed Computing Platform
EPSG	European Petroleum Survey Group
FES	Filter Encoding Specification
GML	Geography Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Secure Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
KVP	Keyword-value pairs
MIME	Multipurpose Internet Mail Extensions
OGC®	Open Geospatial Consortium
OWS	OGC® Web Service
SQL	Structured Query Language
SOAP	Simple Object Access Protocol
UML	Unified Modelling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
VSP	Vendor Specific Parameter
WFS	Web Feature Service
WSDL	Web Services Description Language
XML	Extensible Markup Language

Chapter 5. UML notation

Many diagrams that appear in this standard are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of [OGC 06-121r9].

Chapter 6. Overview

This main content of this engineering report is divided into three clauses.

The first clause, titled `Web IntegrationService`, defines the resources, operations and responses of a new OGC service that can be considered an aggregation point — for the purpose of discovery — for a suite of OGC web services.

The second clause, titled `Associations` and the `GetAssociations` operations, defines a new, generic, OGC operation named `GetAssociations`. This clause defines an XML-encoding and KVP-encoding for a `GetAssociations` operation suitable for use with the KVP-GET, XML-POST and SOAP OGC service bindings. This clause also defines an `associations` resource that is suitable for use with a service that implements a REST style architecture.

The third clause, titled `Catalogue extensions`, defines a `CSW-ebRIM` extension package that allows a catalogue to use the web integration service and the `GetAssociations` operation/resource to harvest a much broader set of metadata about a service than is currently available through the capabilities document alone thus support rich discovery.

Chapter 7. Requirements

7.1. Status Quo

There is currently no way for a service provider to advertise which services are available at an endpoint and do so in a way that is compatible with the investments already made implementing OGC web service. This is especially true for OGC catalogues which must be aware of every single endpoint that is offered in order to be able to harvest the service metadata and make it discoverable.

Furthermore, the associations that can be discovered between OGC services the, content they offer and other OGC and non-OGC resources is very limited. Currently, a client can interrogate an OGC web service's capabilities document to discover what content it offers and perhaps an association to a metadata document describing the content.

The inability to discover what OGC web services are available at an endpoint and discover the rich associations that link OGC web services to the content that they offer and to other OGC and non-OGC resources significantly restricts the usefulness of OGC catalogues and especially of CSW-ebRIM catalogue which has the ability to capture all this information.

7.2. Requirements Statement

In order to address the shortcomings of the status quo the following requirements must be satisfied

1. a means of advertising which OGC web services are available at an endpoint must be provided
2. the means of discovering which OGC web services are available at an endpoint should be compatible with the already significant software development investment made implementing OGC web services (i.e. code reuse)
3. a method must be provided that allows a OGC web services to be interrogated to discover associations between the service, the content it offers and other OGC and non-OGC resources
4. the method of discovering associations should also provide a light-weight query capability so the associations between specific resources or of a specific association type can be requested
5. the means of identifying resource must take into account that OGC resource are typically identified by two pieces of information; the base URL of the service offering the resource and a local identifier for the resource
6. an extension package must to be defined for the CSW-ebRIM catalogue to support rich discovery by allowing the catalogue to automatically harvest services and their associations available at an endpoint

Chapter 8. Web Integration Service

8.1. GetCapabilities operation

8.1.1. Introduction

The Web Integration Service (WIS) is an aggregation service whose only purpose is to provide a list of references to a suite of other, perhaps related OGC services available at an endpoint. In keeping with the evolutionary refinement and reuse goal stated in Clause 6.1, the web integration service only defines a single operation, GetCapabilities, and responds using a standard OGC capabilities document whose content section lists the available services accessible at that end point. A web integration service shall implement the KVP-encoding of the GetCapabilities operation. A web integration service may optionally implement the XML-encoding of the GetCapabilities operation.

8.1.2. Request

UML

The following figure defines the schema of the GetCapabilities request.

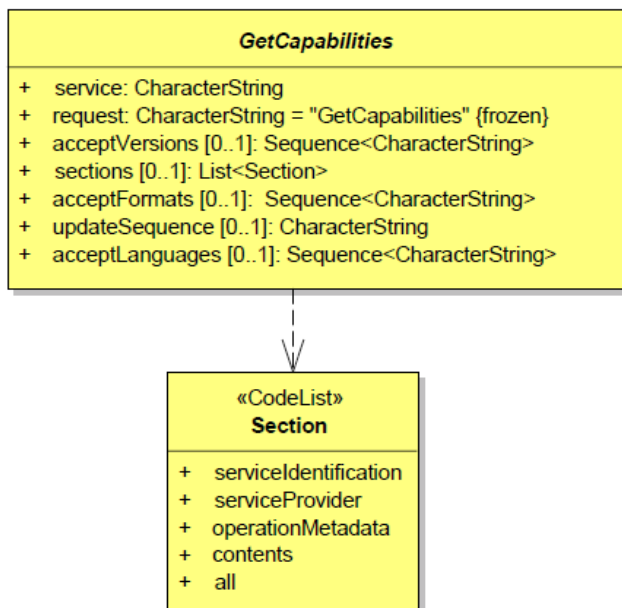


Figure 1. UML class diagram for the GetCapabilities operation

KVP encoding

The KVP encoding of the GetCapabilities request shall be as specified in OGC 06-121r3:2009, 7.2.2. The value of the service parameter shall be fixed to "WIS".

XML encoding

The following XML Schema fragment defines the XML-encoding of the GetCapabilities request.

```

<xsd:element name="GetCapabilities" type="wfs:GetCapabilitiesType"/>
<xsd:complexType name="GetCapabilitiesType">
  <xsd:complexContent>
    <xsd:extension base="ows:GetCapabilitiesType">
      <xsd:attribute name="service" type="ows:ServiceType"
        use="required" fixed="WIS"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

8.1.3. Response

UML

The WIS's response to a GetCapabilities request is essentially a standard OGC capabilities document with a content section composed of zero or more ows:ServiceIdentification sections each describing one OGC web service available at the endpoint.

The following figure defines the schema of the ServiceIdentification section:

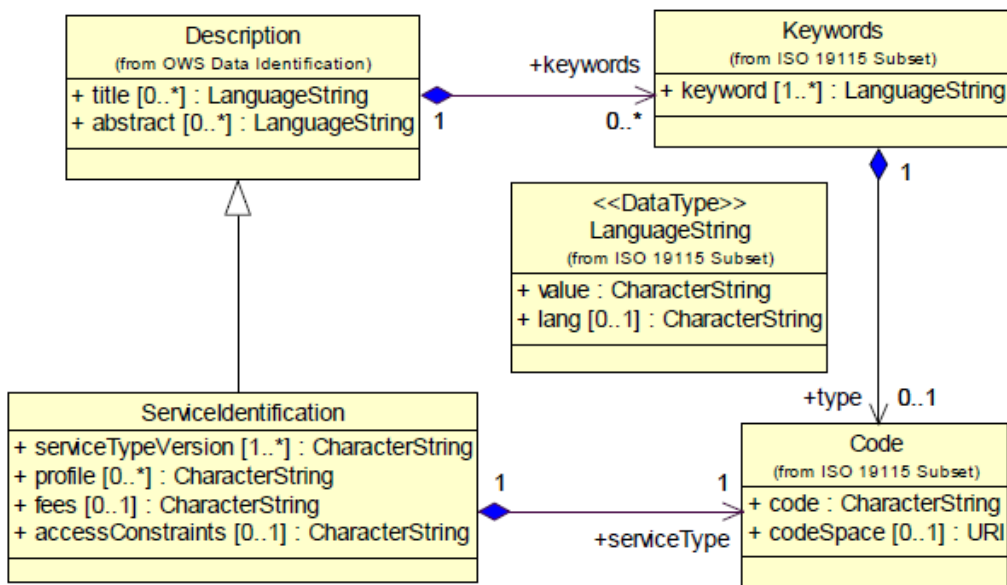


Figure 2. UML class diagram for the ServiceIdentification section

Each service is characterized by its type (i.e. WMS, WMTS, WFS, etc.), the base URL of the service and the standard ServiceIdentification description of the service from Web Service Common (see OGC 06-121r9).

The value of the "type" parameter can be a token – such as the three letter acronyms commonly used by OGC – or a *URI identifier* for the service.

The value of the "baseURL" parameter is the service URL to which GetCapabilities request (see 06-121r9, Clause 7) parameters are appended to form a valid request URL.

XML encoding

The following XML Schema fragment defines the schema for the XML-encoded response to a WIS GetCapabilities request.

```
<xsd:element name="Capabilities" type="wis:WIS_CapabilitiesType"/>
<xsd:complexType name="WIS_CapabilitiesType">
  <xsd:complexContent>
    <xsd:extension base="ows:CapabilitiesBaseType">
      <xsd:sequence>
        <xsd:element ref="wis:Contents"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name="Contents" type="wis:ContentsType"/>
<xsd:complexType name="ContentsType">
  <xsd:sequence>
    <xsd:element ref="wis:Service" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="Service" type="wis:ServiceType"/>
<xsd:complexType name="ServiceType">
  <xsd:sequence>
    <xsd:element ref="ows:ServiceIdentification"/>
  </xsd:sequence>
  <xsd:attribute name="type"
    type="wis:TokenOrAnyURI" use="required"/>
  <xsd:attribute name="baseUrl" type="xsd:anyURI" use="required"/>
</xsd:complexType>
<xsd:simpleType name="TokenOrAnyURI">
  <xsd:union memberTypes="xsd:token xsd:anyURI"/>
</xsd:simpleType>
```

8.1.4. NoOp operation

This engineering report defines an additional WIS operation named "NoOp" that all clients shall ignore. The NoOp operation only exists to satisfy XML validation requirements for capabilities documents imposed by the Web Service Common standard (see OGC 06-121r9, 7.4.6) which requires that OGC web services implement the GetCapabilities operation plus one additional operation. Clearly, the web service common standard did not anticipate the usefulness of a service that only implements the GetCapabilities request. A change request has been posted against OGC 06-121r9 to remove this requirement (see OGC 14-026).

8.1.5. Security considerations

Given the focus on security in Testbed-12, it is worth mentioning that the content of a WIS

capabilities document may be tailored to suite the authentication level of the user. An authenticate user may see more services listed in the capabilities document than a non-authenticated user might.

8.1.6. Relationship to CSW

The question may be raised about the difference between a WIS and a CSW registry service. Unlike a CSW, the WIS provides a light weight, standalone, peer-to-peer discovery mechanism that clients may use to interrogate a providers offerings at an endpoint. The discovery possible using the WIS is not quite as rich as that possible using a CSW but it may be sufficient for many user cases including the harvesting use case described in *Catalogue extension package* clause. From an implementation standpoint, the WIS is certainly a more lightweight implementation.

8.1.7. Example

In response to the following request:

```
http://www.somcorp.com/wis?service=WIS&request=GetCapabilities
```

a web integration service might generate the following capabilities document:

```
<?xml version="1.0"?>
<Capabilities version="1.0.0"
  xmlns="http://www.opengis.net/wis/1.0"
  xmlns:ows="http://www.opengis.net/ows/2.0"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wis/1.0 wis.xsd">
  <ows:ServiceIdentification>
    <ows:Title>SomeSERV</ows:Title>
    <ows:Abstract xml:lang="en">A suite of OGC web services by Some
Corp.</ows:Abstract>
    <ows:ServiceType>SomeSERV</ows:ServiceType>
    <ows:ServiceTypeVersion>7.0.1</ows:ServiceTypeVersion>
  </ows:ServiceIdentification>
  <ows:ServiceProvider>
    <ows:ProviderName>Some Corp.</ows:ProviderName>
    <ows:ProviderSite xlink:href="http://www.somecorp.com"/>
    <ows:ServiceContact>
      <ows:IndividualName>John Smith</ows:IndividualName>
      <ows:PositionName>Senior Developer</ows:PositionName>
      <ows:ContactInfo>
        <ows:Phone>
          <ows:Voice>+1-416-555-1234</ows:Voice>
          <ows:Facsimile>+1-416-555-5678</ows:Facsimile>
        </ows:Phone>
        <ows:Address>
```



```

        <ows:DeliveryPoint>1 Main St., Suite 1</ows:DeliveryPoint>
        <ows:City>SomeCity</ows:City>
<ows:AdministrativeArea>SomeProvince</ows:AdministrativeArea>
        <ows:PostalCode>X1X 2Y2</ows:PostalCode>
        <ows:Country>Canada</ows:Country>
<ows:ElectronicMailAddress>jsmith@somecorp.com</ows:ElectronicMailAddress>
        </ows:Address>
        </ows:ContactInfo>
    </ows:ServiceContact>
</ows:ServiceProvider>
<ows:OperationsMetadata>
    <ows:Operation name="GetCapabilities">
        <ows:DCP>
            <ows:HTTP>
                <ows:Get
xlink:href="http://portal.somecorp.com/somecorp/projects/ows9/someserv.cgi?">
                    <ows:Constraint name="GetEncoding">
                        <ows:AllowedValues>
                            <ows:Value>KVP</ows:Value>
                        </ows:AllowedValues>
                    </ows:Constraint>
                </ows:Get>
            </ows:HTTP>
        </ows:DCP>
        <ows:Parameter name="Format">
            <ows:AllowedValues>
                <ows:Value>application/xml</ows:Value>
                <ows:Value>application/x-bxml</ows:Value>
            </ows:AllowedValues>
        </ows:Parameter>
    </ows:Operation>
    <!-- Clients shall ignore the following operation. -->
    <ows:Operation name="NoOp">
        <ows:DCP>
            <ows:HTTP>
                <ows:Get
xlink:href="http://portal.somecorp.com/somecorp/projects/ows9/someserv.cgi?">
                    </ows:Get>
                </ows:HTTP>
            </ows:DCP>
        </ows:Operation>
</ows:OperationsMetadata>
<Contents>
    <Service type="WMS"
baseUrl="http://portal.somecorp.com/somecorp/projects/ows9/someserv.cgi">
        <ows:ServiceIdentification>
            <ows:Title>SomeSERV WMS</ows:Title>
            <ows:Abstract xml:lang="en">OGC-compliant cascading web map server (WMS)
by Some Corp.</ows:Abstract>
            <ows:ServiceType>WMS</ows:ServiceType>
            <ows:ServiceTypeVersion>1.3.2</ows:ServiceTypeVersion>

```

```

    <ows:ServiceTypeVersion>1.3.1</ows:ServiceTypeVersion>
    <ows:ServiceTypeVersion>1.3.0</ows:ServiceTypeVersion>
    <ows:ServiceTypeVersion>1.1.1</ows:ServiceTypeVersion>
    <ows:ServiceTypeVersion>1.1.0</ows:ServiceTypeVersion>
    <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
  </ows:ServiceIdentification>
</Service>
<Service type="WFS"
baseUrl="http://portal.somecorp.com/somecorp/projects/ows9/someserv.cgi?DATASTORE=OWS9
">
  <ows:ServiceIdentification>
    <ows:Title>The USGS National Map Web Feature Service (subset)</ows:Title>
    <ows:Abstract>This web feature service offers a subset of the USGS
National Map. The National Map is a collaborative effort among the USGS and other
Federal, State, and local partners to improve and deliver topographic information for
the Nation. The geographic information available from The National Map includes
hydrography, boundaries, transportation, structures, and land cover. Other types of
geographic information can be added within the viewer or brought in with The National
Map data into a Geographic Information System to create specific types of maps or map
views. The National Map is a significant contribution to the National Spatial Data
Infrastructure (NSDI) and currently is being transformed to better serve the
geospatial community by providing high quality, integrated geospatial data and
improved products and services including new generation digital topographic
maps.</ows:Abstract>
    <ows:Keywords>
      <ows:Keyword>WFS</ows:Keyword>
      <ows:Keyword>USGS</ows:Keyword>
      <ows:Keyword>TNM</ows:Keyword>
      <ows:Keyword>Web</ows:Keyword>
      <ows:Keyword>Feature</ows:Keyword>
      <ows:Keyword>Service</ows:Keyword>
      <ows:Keyword>hydrography</ows:Keyword>
      <ows:Keyword>boundaries</ows:Keyword>
      <ows:Keyword>transportation</ows:Keyword>
      <ows:Keyword>structures</ows:Keyword>
      <ows:Keyword>land</ows:Keyword>
      <ows:Keyword>cover</ows:Keyword>
    </ows:Keywords>
    <ows:ServiceType>WFS</ows:ServiceType>
    <ows:ServiceTypeVersion>2.0.0</ows:ServiceTypeVersion>
    <ows:ServiceTypeVersion>1.1.0</ows:ServiceTypeVersion>
    <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
    <ows:Fees>None</ows:Fees>
    <ows:AccessConstraints>None</ows:AccessConstraints>
  </ows:ServiceIdentification>
</Service>
<Service type="WFS"
baseUrl="http://portal.somecorp.com/somecorp/projects/ows9/someserv.cgi?DATASTORE=STDS
">
  <ows:ServiceIdentification>
    <ows:Title>The USGS National Map Web Feature Service (subset)</ows:Title>

```

```
<ows:Abstract>This web feature service offers a subset of the USGS National Map. The National Map is a collaborative effort among the USGS and other Federal, State, and local partners to improve and deliver topographic information for the Nation. The geographic information available from The National Map includes hydrography, boundaries, transportation, structures, and land cover. Other types of geographic information can be added within the viewer or brought in with The National Map data into a Geographic Information System to create specific types of maps or map views. The National Map is a significant contribution to the National Spatial Data Infrastructure (NSDI) and currently is being transformed to better serve the geospatial community by providing high quality, integrated geospatial data and improved products and services including new generation digital topographic maps.</ows:Abstract>
```

```
<ows:Keywords>
```

```
<ows:Keyword>WFS</ows:Keyword>
```

```
<ows:Keyword>USGS</ows:Keyword>
```

```
<ows:Keyword>TNM</ows:Keyword>
```

```
<ows:Keyword>Web</ows:Keyword>
```

```
<ows:Keyword>Feature</ows:Keyword>
```

```
<ows:Keyword>Service</ows:Keyword>
```

```
<ows:Keyword>hydrography</ows:Keyword>
```

```
<ows:Keyword>boundaries</ows:Keyword>
```

```
<ows:Keyword>transportation</ows:Keyword>
```

```
<ows:Keyword>structures</ows:Keyword>
```

```
<ows:Keyword>land</ows:Keyword>
```

```
<ows:Keyword>cover</ows:Keyword>
```

```
</ows:Keywords>
```

```
<ows:ServiceType>WFS</ows:ServiceType>
```

```
<ows:ServiceTypeVersion>2.0.0</ows:ServiceTypeVersion>
```

```
<ows:ServiceTypeVersion>1.1.0</ows:ServiceTypeVersion>
```

```
<ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
```

```
<ows:Fees>None</ows:Fees>
```

```
<ows:AccessConstraints>None</ows:AccessConstraints>
```

```
</ows:ServiceIdentification>
```

```
</Service>
```

```
<Service type="WCS"
```

```
baseUrl="http://portal.somecorp.com/somecorp/projects/ows9/someserv.cgi?DATASTORE=OWS9">
```

```
<ows:ServiceIdentification>
```

```
<ows:Title>SomeSERV WCS - OWS9</ows:Title>
```

```
<ows:Abstract xml:lang="en">OGC-compliant web coverage server (WCS) by Some Corp.</ows:Abstract>
```

```
<ows:ServiceType>WCS</ows:ServiceType>
```

```
<ows:ServiceTypeVersion>1.1.2</ows:ServiceTypeVersion>
```

```
<ows:ServiceTypeVersion>1.1.1</ows:ServiceTypeVersion>
```

```
<ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
```

```
</ows:ServiceIdentification>
```

```
</Service>
```

```
<Service type="WCS"
```

```
baseUrl="http://portal.somecorp.com/somecorp/projects/ows9/someserv.cgi?DATASTORE=STDS">
```

```
<ows:ServiceIdentification>
```

```

    <ows:Title>SomeSERV WCS - STDS</ows:Title>
    <ows:Abstract xml:lang="en">OGC-compliant web coverage server (WCS) by
Some Corp.</ows:Abstract>
    <ows:ServiceType>WCS</ows:ServiceType>
    <ows:ServiceTypeVersion>1.1.2</ows:ServiceTypeVersion>
    <ows:ServiceTypeVersion>1.1.1</ows:ServiceTypeVersion>
    <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
  </ows:ServiceIdentification>
</Service>
  <Service type="CSW"
baseUrl="http://portal.somecorp.com/somecorp/projects/ows9/someserv.cgi">
  <ows:ServiceIdentification>
    <ows:Title xml:lang="en-CA">Catalogue Service Web (CSW) by Some
Corp.</ows:Title>
    <ows:Abstract xml:lang="en">The Some Corp. Catalog Service Web (CSW)
implements a web registry service that complies to the Open Geospatial Consortium
Inc.'s Catalogue Service implementation specification V2.0.2 (see 07-006r1) which
specifies the interfaces, bindings, and a framework for defining application profiles
required to publish and access digital catalogues of metadata for geospatial data,
services, and related resource information. Metadata act as generalized properties
that can be queried and returned through catalogue services for resource evaluation
and invocation or retrieval of the referenced resource.</ows:Abstract>
    <ows:Keywords>
      <ows:Keyword xml:lang="en-CA">catalogue</ows:Keyword>
    </ows:Keywords>
    <ows:ServiceType>CSW</ows:ServiceType>
    <ows:ServiceTypeVersion>2.0.2</ows:ServiceTypeVersion>
    <ows:Fees>none</ows:Fees>
    <ows:AccessConstraints>none</ows:AccessConstraints>
  </ows:ServiceIdentification>
</Service>
</Contents>
</Capabilities>

```

Since this capabilities document is composed mostly of standard OGC components it is an easy task to extend an OGC catalogue to consume this document. The catalogue can then automatically determine which services are offered by this provider and harvest each one in turn also maintaining the associations between the web integration service and its child services.

Chapter 9. GetAssociations operation

9.1. Introduction

When a service provider deploys a suite of OGC web services each component service typically exists in isolation and external clients are unaware of the relationships that might exist between these services, their offerings and other OGC and non-OGC web resources. The services themselves, however, typically have much more internal knowledge about such relationships or associations. Services know what data sources were accessed to provide their offerings; they know what other representations might exist, etc. For example, a WMS knows if the source data for a layer it offers is a SHAPE file, or a WFS feature type, or another WMS layer (i.e. cascading), or if a WMTS layer exists that renders the same information more efficiently. In short OGC web services have a lot of internal association knowledge that is currently inaccessible to client applications, especially catalogues. A great deal of added value can be derived if this internal association knowledge can be externalized. Thus, the scope of this clause is to define methods and apparatus to allow OGC web services to externalize this internal association knowledge. This clause describes how associations between OGC and non-OGC resource can be represented in XML. This clause further defines a new operation; named GetAssociations, that existing OGC web services can implement to allow client applications to interrogate an OGC web service about its internal association knowledge.

9.2. Associations

9.2.1. Introduction

This clause defines a general XML Schema for representation associations between two resources.

An association is logically of the form "SOURCE REL TARGET" where the SOURCE resource is the subject of a directional RELationship of the specified type with the TARGET resource (e.g. layer X renders feature type Y).

9.2.2. Types of association resources

This engineering report distinguishes two types of resources that may participate in an association:

1. OGC web service resources
2. URL-addressable resources

OGC web service resources are resources that are only accessible from within an OGC web service. The classic set of OGC resources — WMS map, WFS feature type and WCS coverage — are examples of OGC web service resources since the only way to access instances of each is through the services that offer them. Thus, OGC web service resources are not addressable using a URL alone.

URL-addressable resources are resources which can be accessed directly via URL. Many OGC resources are URL-addressable. Examples include OGC web services, which are accessible via their *base URL* or symbology encoding documents stored in some web-accessible location. For the purposes of this engineering report the only recognized URL-addressable OGC resources are those for which an identifier has been registered with the OGC Naming Authority (OGC NA).

OGC web service resources are distinguished from URL-addressable resources because two components are required for access; the *base URL* of the OGC web service offering the resource and the service-specific identifier which locates the resource within the service.

Example: A WMS layer named Foundation"INWATER_1M and served by an OGC WMS would be identified using the following two components:

1. baseURL of the WMS: <http://www.somecorp.com/wms>
2. service-specific identifier: Foundation.INWATERA_1M

Interaction with an OGC web service can only commence once the *base URL* of the service has been discovered. Communication with the OGC web service commences with the client obtaining the capabilities document of the service by appending the appropriate GetCapabilities request parameters to the previously-discovered *base URL* of the service. The service-specific identifiers for the OGC web service resources the service offers can then be obtained from the capabilities document's content section.

9.2.3. UML

The following UML diagram defines the schema of an association.

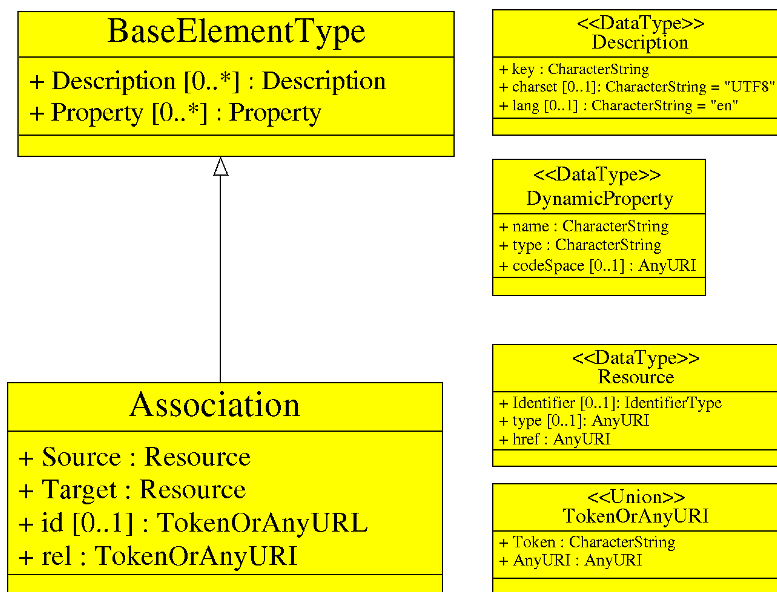


Figure 3. UML class diagram for an Association

9.2.4. XML-encoding

The following XML Schema fragments defines the XML-encoding for an association.

```

<xsd:element name="Association" type="wos:AssociationType"/>
<xsd:complexType name="AssociationType">
  <xsd:complexContent>
    <xsd:extension base="wis:BaseElementType">
      <xsd:sequence>
        <xsd:element name="Source" type="wos:ResourceType"/>
        <xsd:element name="Target" type="wos:ResourceType"/>
      </xsd:sequence>
      <xsd:attribute name="id"
        type="wos:TokenOrAnyURI" use="optional"/>
      <xsd:attribute name="rel"
        type="wos:TokenOrAnyURI" use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

The "Source" parameter shall identify the source resource of the association.

The "Target" parameter shall identify the target resource of the association.

The "rel" parameter shall be used to specify the association type. The canonical value of the rel attribute shall either be a token taken from the IANA Link Relations registry (see RFC 5988) or it shall be a URI registered with the OGC Naming Authority. Vendor or domain specific "rel" values may also be used. However, this engineering report does not assign any specified meaning or interpretation to these values. A association Parser shall not fail if it encounters an associations with an unrecognized "rel" value and is free to ignore that association.

The "id" parameter may used to assign an identifier to the association itself.

The following XML fragment defines a reference to a resource that participates as the source or target of an association.

```

<xsd:complexType name="ResourceType">
  <xsd:sequence>
    <xsd:element ref="assoc:Identifier" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="type" type="assoc:TokenOrAnyURI" use="optional"/>
  <xsd:attribute ref="xlink:href" use="required"/>
</xsd:complexType>
<xsd:element name="Identifier">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="namespace" type="xsd:anyURI"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
<xsd:simpleType name="TokenOrAnyURI">
  <xsd:union memberTypes="xsd:token xsd:anyURI"/>
</xsd:simpleType>

```

If the resource is a *URL-addressable resource* then its URL shall be specified using the "xlink:href" parameter and its type shall be specified using the "type" parameter. The "Identifier" parameter shall be omitted in this case.

If the resource is an *OGC web service resource*, then its *base URL* shall be specified using the "xlink:href" parameter, its type shall be specified using the "type" parameter and the service-specific identifier that locates the resource within the OGC service offering the resource shall be specified using the "Identifier" parameter. If the service-specific resource identifier within the OGC web service is scoped then its namespace shall be specified using the "namespace" parameter. The OGC service type serving the resource shall be deduced from the specified resource type. For example, a WMS layer resource is served by a WMS while a WFS feature type resource is served by a WFS.

9.2.5. Associations container

The following XML Schema fragment defines a container for a set of associations.

```

<xsd:element name="Associations">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="assoc:Association"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="version" type="xsd:string"
      use="required" fixed="1.0.0"/>
  </xsd:complexType>
</xsd:element>

```


The Associations element is a container for zero or more *Association* elements.

9.2.6. MIME type

This engineering report defines the following MIME type to represent an XML document that encodes a set of associations: **application/ogc-as+xml**.

9.3. Standard identifiers

The following tables define standard identifiers for resources and association types used by OGC. This is by no means a comprehensive list but represents the main set of OGC web services and their resources.

Table 3. Service Type Identifiers

Title	Description	HTTP-URI form	URN form
Web Integration Service	An OGC service whose only function is to identify a collection of other OGC services	http://www.opengis.net/serviceType/wis	urn:ogc:serviceType:wis
Web Feature Service	An OGC service for serving and managing features	http://www.opengis.net/serviceType/wfs	urn:ogc:serviceType:wfs
Web Object Service	An OGC service for serving and managing generic (perhaps binary) objects	http://www.opengis.net/serviceType/wos	urn:ogc:serviceType:wos
Web Map Service	An OGC service that serves maps	http://www.opengis.net/serviceType/wms	urn:ogc:serviceType:wms
Web Map Tiling Service	An OGC service that serves map tiles	http://www.opengis.net/serviceType/wmts	urn:ogc:serviceType:wmts
Web Processing Service	An OGC service that exposes Web Processing Services	http://www.opengis.net/serviceType/wps	urn:ogc:serviceType:wps
Web Coverage Service	An OGC service that serves coverages	http://www.opengis.net/serviceType/wcs	urn:ogc:serviceType:wcs

Title	Description	HTTP-URI form	URN form
Catalogue Service	An OGC catalogue service	http://www.opengis.net/serviceType/csw	urn:ogc:serviceType:csw
Web Registry Service	A profile of an OGC catalogue service that uses ebRIM as its information model	http://www.opengis.net/serviceType/wrs	urn:ogc:serviceType:csw
Geosynchronization Service	An OGC service supporting data crowdsourcing.	http://www.opengis.net/serviceType/gss	urn:ogc:serviceType:gss

NOTE

A version number may be appended to the end of the HTTP-URI or URN to reference to a specific version of a service.

Table 4. Content Type Identifiers

Title	Description	HTTP-URI form	URN form
WFS Feature Type	A feature type offered by an OGC Web Feature Service	http://www.opengis.net/resourceType/wfs/featureType	urn:ogc:resourceType:wfs:featureType
WOS Object	A generic object offered by an OGC Web Object Service	http://www.opengis.net/resourceType/wos/object	urn:ogc:resourceType:wos:object
WMS Theme	A collection of WMS layers.	http://www.opengis.net/resourceType/wms/theme	urn:ogc:resourceType:wms:theme
WMS Layer	A map layer offered by an OGC Web Map Service	http://www.opengis.net/resourceType/wms/layer	urn:ogc:resourceType:wms:layer
WMTS Theme	A collection of WMTS layers.	http://www.opengis.net/resourceType/wmts/theme	urn:ogc:resourceType:wmts:theme
WMTS Layer	A tiles map layer offered by an OGC Web Map Tiling Service	http://www.opengis.net/resourceType/wmts/layer	urn:ogc:resourceType:wmts:layer
WPS Process	A processing service offered by an OGC Web Processing Service	http://www.opengis.net/resourceType/wps/process	urn:ogc:resourceType:wps:process

Title	Description	HTTP-URI form	URN form
WCS Theme	A collection of WCS coverages	http://www.opengis.net/resourceType/wcs/theme	urn:ogc:resourceType:wcs:theme
WCS Coverage	A coverage offered by an OGC Web Coverage Service	http://www.opengis.net/resourceType/wcs/coverage	urn:ogc:resourceType:wcs:coverage
Metadata	A document, conforming to a well-known standard that contains metadata about the resource	http://www.opengis.net/resourceType/metadata	urn:ogc:resourceType:metadata
Dublin Core Metadata	Metadata about a resource that uses the Dublin Core set of elements	http://www.opengis.net/resourceType/metadata/dublinCore	urn:ogc:resourceType:metadata:dublinCore
Symbology Encoding Document	A document that encodes symbology in a machine readable format	http://www.opengis.net/resourceType/symbologyEncoding	urn:ogc:resourceType:symbologyEncoding

Table 5. Association Type Identifiers

Title	Description	HTTP-URI form	URN form
Renders	Associates two resources where one resource renders another resources (e.g. a WMS layer that renders a WFS feature type)	http://www.opengis.net/associationType/renderers	urn:ogc:associationType:renders
Tiled By	Associated two resource where one resource is a tiled representation of the other resource (e.g. a WMTS layers that is a tiles representation of a WMS layer)	http://www.opengis.net/associationType/tiledBy	urn:ogc:associationType:tiledBy

Title	Description	HTTP-URI form	URN form
ParentOf	Associates two resources where one resource is the parent of the other resource. This is typically used to register a nested hierarchy of resources	http://www.opengis.net/associationType/parentOf	urn:ogc:associationType:parentOf
Cascades	Associates two services where one OGC service is using the other OGC service as a data source. For examples where one WMS "cascades" a number of other WMS's	http://www.opengis.net/associationType/cascades	urn:ogc:associationType:cascades
CreatedBy	Associates two services where one is responsible for the other creation	http://www.opengis.net/associationType/createdBy	urn:ogc:associationType:createdBy
Symbolizes	Associates the symbology encoding document with a WFS Feature Type	http://www.opengis.net/associationType/symbolizes	urn:ogc:associationType:symbolizes
OperatesOn	Associates two resources where one operates on the other	http://www.opengis.net/associationType/operatesOn	urn:ogc:associationType:operatesOn
DescribedBy	Associates two resources where one resource is a description of the other using some metadata standard	http://www.opengis.net/associationType/describedBy	urn:ogc:associationType:describedBy

9.4. GetAssociations operation

9.4.1. Introduction

As previously discussed, OGC web services have internal knowledge about what associations exist between the service, its offerings and other OGC and non-OGC resources. The GetAssociations operation is the means by which an OGC web service may be interrogated about the associations of which it is aware.

Once externalized in this way, this association information can be automatically harvested and made discoverable from OGC catalogues creating a richer environment for discovery.

It is anticipated that existing OGC web services, wishing to participate as part of an integrated suite of OGC web services, would implement this operation as an add-on to the existing set of operations that the service implements.

It is further anticipated that this operation shall eventually be defined as a mandatory operation for all OGC web services in the OGC Web Services Common standard (see OGC 06-121r9). The document shall be posted as a change request against OGC 06-121r9.

9.4.2. UML

The following UML diagrams define the request and response for the GetAssociations operation.

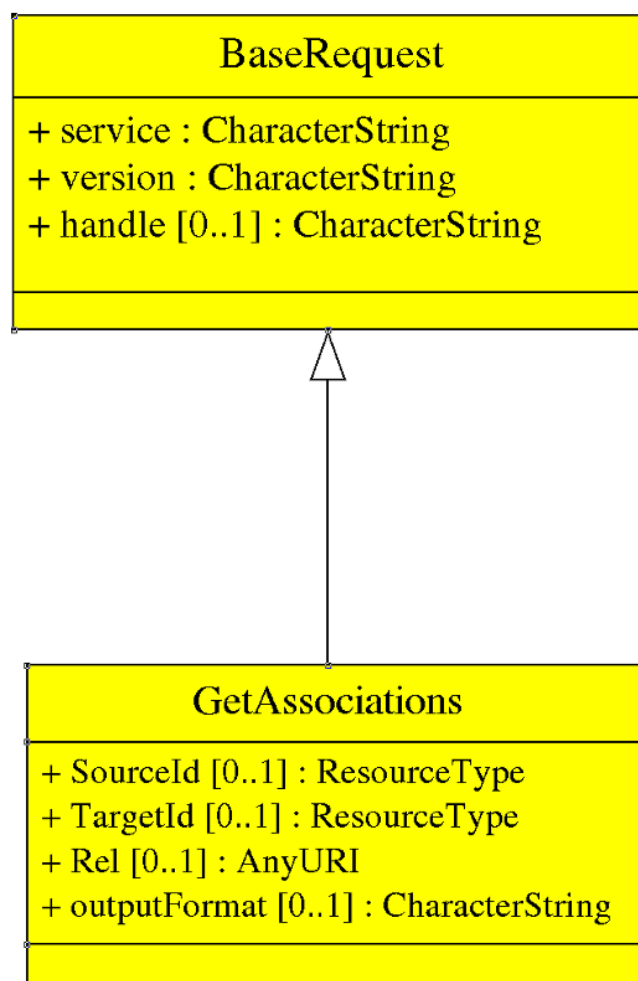


Figure 4. UML class diagram for the GetAssociations operation

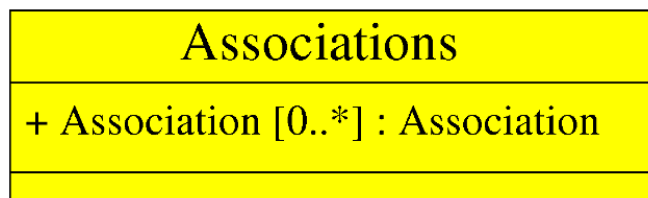


Figure 5. UML class diagram for the GetAssociations response

The mandatory service parameter of the GetAssociations operation shall encode the service identifier of the service being invoked to execute the operation.

The mandatory version parameter of the GetAssociations operation shall be the fixed values of "1.0.0".

The optional contents parameter may be used to interrogate the known association of a specified subset of resources that a service offers. If the parameter is omitted the response document shall contains all knows associations.

The optional outputFormat parameter may be used to request the response in a specific format. The default value is application/ogc-as+xml (see 8.3.5). Other output format values are allowed (e.g. a JSON response) to which this engineering report does not assign any meaning.

9.4.3. KVP-encoding

The following table defines the KVP-encoding for the GetAssociations operation.

Table 6. GetAssociations KVP encoding

URL Component	O/M	Description
REQUEST	M	Fixed value of GetAssociations
SERVICE	M	Identifier for the service offering the operation
VERSION	M	Fixed value of 1.0.0
SOURCEID	O	A URL reference to a source resource (or the base URL of an OGC service) of an association
LOCALSOURCEID	O	The local identifier of a source resource of an association. The base URL of the OGC web service offering the resource shall be specified as the value of the SOURCEID parameter
TARGETID	O	A URL reference to a target resource (or the base URL of an OGC service) of an association
LOCALTARGETID	O	The local identifier of a target resource of an association. The base URL of the OGC web service offering the resource shall be specified as the value of the TARGETID parameter
REL	O	The link relation (or association type) of an association

URL Component	O/M	Description
OUTPUTFORMAT	O	The response format of the operation. The default values is application/ogc-as+xml.

9.4.4. XML-encoding

The following XML Schema fragment defines the XML-encoding for the GetAssociations operation.

```

<xsd:element name="GetAssociations" type="ows:GetAssociationsType"/>
<xsd:complexType name="GetAssociationsType">
  <xsd:sequence>
    <xsd:element ref="ows:ResponseHandler"
      minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="SourceId"
      type="assoc:ResourceType" minOccurs="0"/>
    <xsd:element name="TargetId"
      type="assoc:ResourceType" minOccurs="0"/>
    <xsd:element name="Rel"
      type="xsd:anyURI" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="service"
    type="ows:ServiceType" use="required"/>
  <xsd:attribute name="version" type="xsd:string"
    use="required" fixed="1.0.0"/>
  <xsd:attribute name="outputFormat"
    type="xsd:string" default="application/as+xml"/>
</xsd:complexType>

```

9.4.5. /associations resource (REST)

Servers implementing the REST architectural style shall provide a */associations* resource. The query parameters SOURCEID, LOCALSOURCEID, TARGET, LOCALTARGETID, REL and OUTPUTFORMAT (see the [GetAssociations KVP encoding](#) table) may be appended to this resource path to query specific association information.

9.4.6. Parameter discussion

Without any parameters, an OGC web service implementing the GetAssociations operation shall return a document containing all known associations.

The function of the parameters defined on the GetAssociations operation is to provide a rudimentary query capability. Specifying any of the parameters is equivalent to defining a query predicate composed of AND'ed equality expressions. For example, the following request fragment:

```
... request=GetAssociations&SourceId=urn:uuid:...&Rel=renders ...
```

is equivalent to the query predicate:

```
(SourceId==urn:uuid:...) AND (Rel==renders)
```

In this example, a service implementing the `GetAssociations` operation would return all associations where "Rel==Renders" for the specified `SourceId`.

9.4.7. Response

The response to a `GetAssociations` operation, when the output format is the default value `application/ogc-as+xml`, shall be a set of associations encoded as an XML document that validates against the schema describe in the [Associations](#) clause.

9.4.8. Example

In response to the following request for all known associations:

```
<?xml version="1.0"?>
<ows:GetAssociations
  service="WFS"
  version="1.0.0"
  outputFormat="application/ogc-as+xml"
  xmlns:ows="http://www.opengis.net/ows/2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/ows/2.0
    http://schemas.opengis.net/ows/2.0/owsGetAssociations.xsd">
</ows:GetAssociations>
```

an OGC web map service might generate the following response:

```
<?xml version="1.0"?>
<Associations xmlns="http://www.opengis.net/associations/1.0"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/associations/1.0
    http://schemas.opengis.net/wis/1.0.0/associations.xsd">
  version="1.0.0">
  <Association rel="http://www.opengis.net/associationType/cascades">
    <Source type="http://www.opengis.net/serviceType/wms"
      xlink:href="https://onestop.somecorp.com/somecorp/srvr"/>
    <Target type="http://www.opengis.net/serviceType/wms"
      xlink:href="https://demo.somecorp.com/srvr"/>
  </Association>
  <Association rel="http://www.opengis.net/associationType/cascades">
    <Source type="http://www.opengis.net/resourceType/wms/layer"
      xlink:href="https://onestop.somecorp.com/srvr">
      <Identifier>Demo.Foundation.aerofacp_1m</Identifier>
    </Source>
    <Target type="http://www.opengis.net/resourceType/wms/layer"
```



```

        xlink:href="https://demo.somecorp.com/srvr">
        <Identifier>Foundation.aerofacp_1m</Identifier>
    </Target>
</Association>
<Association rel="http://www.opengis.net/associationType/tiledBy">
    <Source type="http://www.opengis.net/resourceType/wms/layer"
        xlink:href="https://onestop.somecorp.com/srvr">
        <Identifier>Demo.Foundation.aerofacp_1m</Identifier>
    </Source>
    <Target type="http://www.opengis.net/resourceType/wmts/layer"
        xlink:href="https://demo.somecorp.com/somecorp/srvr">
        <Identifier>Foundation.aerofacp_1m</Identifier>
    </Target>
</Association>
<Association rel="http://www.opengis.net/associationType/renderers">
    <Source type="http://www.opengis.net/resourceType/wms/layer"
        xlink:href="https://onestop.somecorp.com/srvr">
        <Identifier>Demo.Foundation.aerofacp_1m</Identifier>
    </Source>
    <Target type="http://www.opengis.net/resourceType/wfs/featureType"
        xlink:href="https://demo.somecorp.com/srvr?DATASTORE=Foundation">
        <Identifier
namespace="http://www.somecorp.com/ns">aerofacp_1m</Identifier>
    </Target>
</Association>
<Association rel="http://www.opengis.net/associationType/renderers">
    <Source type="http://www.opengis.net/resourceType/wms/layer"
        xlink:href="https://onestop.somecorp.com/srvr">
        <Identifier>Demo.Foundation.etopo2</Identifier>
    </Source>
    <Target type="http://www.opengis.net/def/resourceType/ogc/wcs/coverage"
        xlink:href="https://demo.somecorp.com/srvr?DATASTORE=Foundation">
        <Identifier>etopo2</Identifier>
    </Target>
</Association>
</Associations>

```

9.5. Harvesting associations into an ebRIM CSW catalogue

To properly harvest the GetAssociations response it is important that each object returned either as a part of the source or the target also be harvested into the catalogue. This can be accomplished in a number of ways but the following basic algorithm below should be used as a guideline.

- Harvest the service containing the GetAssociations operation first as the generated UUIDs will be needed to create the proper ebRIM Association objects
- For each Association returned from the GetAssociations response
 - Process the Source object

- Use the Identifier value and the generated ebRIM service id to retrieve the Identifier's ebRIM id
- Process the Target object
 - Use the xlink:href parameter to retrieve the associated ebRIM service id for the Target object in the ebRIM database
 - If the service does not exist in the catalogue it will need to be harvested
 - Use the Identifier value and namespace attribute to retrieve the Identifier's ebRIM id
- Using the ebRIM id retrieved for the Source, the Association's rel attribute value and the ebRIM id retrieved for the Target object, create an ebRIM Association object
- Once all associations have been processed the generated ebRIM Association objects should be submitted as part of an ebRIM transaction document

The figure *ebRIM Catalogue View* presents a typical view of what the various ebRIM Associations might look like in an ebRIM database once all the appropriate services and other objects have been successfully harvested.

The ebRIM Catalogue View

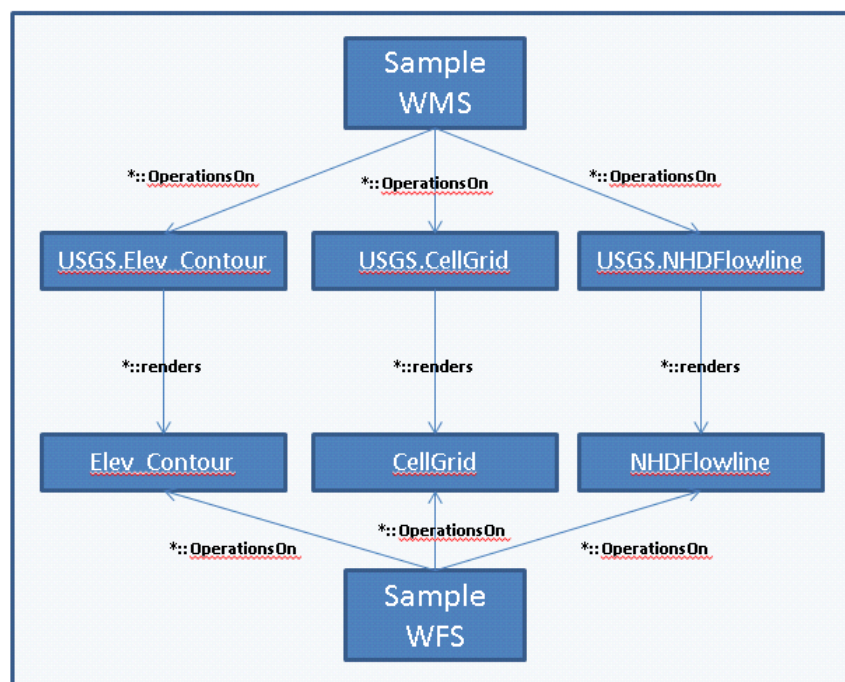


Figure 6. *ebRIM Catalogue View*

Chapter 10. Catalogue extension package

10.1. Introduction

A central component of a deployed suite of integrated OGC web services is a catalogue. The catalogue encapsulates the knowledge of the system making the resources that the system offers and the associations between those resources discoverable to both internal components of the system as well as external clients.

For this Testbed-12 work item, the CSW-ebRIM profile (see OGC 07-110r4), a profile of the OGC catalogue (see OGC 07-006r1) that uses ebRIM (see OASIS regrep-rim-3.0-os), is chosen as the primary catalogue. This choice is motivated by the fact that ebRIM is a flexible meta-model that already includes structures to register resources and the associations between them as well as allow resource to be classified using internal as well as external classification schemes.

The primary means of harvesting or registering an OGC web service into a CSW-ebRIM catalogue is to use the Harvest operation (see OGC 07-110r4, Clause 14) to read the service's capabilities document and then map that document into the catalogue's ebRIM information model.

This clause defines a service integration ebRIM package that encapsulates all the identifiers, associations and classification schemes necessary to allow an OGC CSW-ebRIM catalogue to register OGC web services, the resource those services offer and the associations between those services, their offerings and other OGC and non-OGC resources.

10.2. Identifiers

10.2.1. Introduction

This clause defines a set of identifiers that shall be used to map the capabilities document of an OGC web service into the ebRIM information model of an OGC web registry service (see OGC 07-110r4).

10.2.2. Identifiers for OGC web services for ebRIM

The table *Service Type Identifiers* defines a standard set of service type identifiers for OGC web services. These identifiers are used to define classification nodes within the ebRIM model that allow ebRIM Service records to be classified according to the type of service being represented. The canonical ebRIM service type classification scheme is defined in Annex B.

These identifiers may also be used as the value of a slot, named *resourceType* that may also be used to tag a service record with the type of OGC web service it represents.

10.2.3. Identifier for the content of OGC web services for ebRIM

The table *Content Type Identifier* defined a set of identifier that may be used as the value of the "objectType" parameter within ebRIM to identify the type of OGC web service content being represented by the catalogue record. These identifiers are used to extend the canonical ebRIM object type classification scheme (see Annex B).

10.2.4. Identifiers for OGC documents for ebRIM

The following table lists a set of identifiers for OGC and non-OGC document types commonly used to describe OGC resources. These identifiers are used to define a canonical ebRIM classification scheme named HarvestResourceType (see Annex B).

Table 7. OGC Document Type Identifiers for ebRIM

Identifier	Description
urn:ogc:documentType:capabilities:wis	WIS capabilities document
urn:ogc:documentType:capabilities:wfs	WFS capabilities document
urn:ogc:documentType:capabilities:wms	WMS capabilities document
urn:ogc:documentType:capabilities:wmts	WMTS capabilities document
urn:ogc:documentType:capabilities:wcs	WCS capabilities document
urn:ogc:documentType:metadata:fgdc	Metadata document conforming to the FGDC standard.
urn:ogc:documentType:metadata:anzlic	Metadata document conforming to the ANZLIC standard.
urn:ogc:documentType:metadata:iso19139	Metadata document conforming to the ISO 19139 standard.
urn:ogc:documentType:metadata:iso19119	Metadata document conforming to the ISO 19119 standard.
urn:ogc:documentType:metadata:iso19115	Metadata document conforming to the ISO 19115 standard.
urn:ogc:documentType:ebxml-regrep:classificationScheme	A metadata document containing an ebRIM classification scheme.
urn:ogc:documentType:symbologyEncoding	A symbology encoding document.
urn:ogc:documentType:schema:xml	An XML schema document.

Harvesting OGC web service is accomplished by reading and mapping the service's capabilities document into the information model of the CSW-ebRIM catalogue. A service's capabilities document can be identified by a client providing the complete URL to the capabilities document (e.g. <http://www.someserver.com/wfs?service=WFS&request=GetCapabilities>) or by providing the baseURL of the OGC web service and letting the catalogue form the necessary GetCapabilities request. The latter case gives the catalogue server more control about what to harvest especially in the case of a suite of OGC web services offered through a WIS. The following set of identifiers may be used in the latter case to identify that a base URL is being used to identify the capabilities document of the OGC web service – and thus the catalogue needs to formulate a complete GetCapabilities request in order to obtain the capabilities document. These identifiers will typically be used as the values of the *resourceType* parameter of the CSW Harvest operation.

Table 8. Identifiers for Service Base URLs for ebRIM

Identifier	Description
urn:ogc:baseUrl:service:wis	WIS base URL.
urn:ogc:baseUrl:service:wfs	WFS base URL.
urn:ogc:baseUrl:service:wos	WOS base URL.
urn:ogc:baseUrl:service:wms	WMS base URL.
urn:ogc:baseUrl:service:wmts	WMTS base URL.
urn:ogc:baseUrl:service:wps	WPS base URL.
urn:ogc:baseUrl:service:csw	CSW base URL.
urn:ogc:baseUrl:service:wrs	WRS base URL.
urn:ogc:baseUrl:service:gss	GSS base URL.

10.2.5. Identifiers for associations for ebRIM

The table *Association Type Identifiers* defines a set of identifier for association types that may exist between two resources represented in ebRIM. The identifiers in this table are used to extent the canonical ebRIM association type classification scheme (see Annex B).

These values shall be used as the value of the "associationType" property of the Association object within ebRIM to represent the type of association between registered resources.

The following table lists association identifiers that are useful in the context of OGC service integration and are already defined in the canonical ebRIM AssociationType classification scheme.

Table 9. Additional Association Type Identifiers for ebRIM

Identifier	Description
urn:oasis:names:tc:ebxml-regrep:AssociationType:HasMember	Associates two resources where one resource is a collection and the other resource is a member of that collection (e.g. a collection of WMS layers that share a common theme).
urn:oasis:names:tc:ebxml-regrep:AssociationType:OffersService	Associates two resources where one resource is a web integration service and the other resource is another OGC web service offered by the WIS.

10.2.6. Identifiers for ebRIM slot names

The ebRIM model provides extensible metadata via a slot mechanism. Slots are a name and value list pair that provides additional metadata about a registered object. Among other things, slots are used in this standard to map values found in an OGC's capabilities document into the ebRIM model.

The following table lists a set of identifiers that may be used as the value of the "name" parameter of a slot. These identifiers are used to define a canonical ebRIM SlotName classification scheme.

Table 10. Slot Name Identifiers for ebRIM

Identifier	Description
accessConstraints	Describes any access constraints associated with the resource.
baseTableName	The base table name within an RDBMS used to store a resource.
baseUrl	The base URL of the resource (e.g. the base URL of a service)
contactInstructions	Instruction for contacting a responsible party for the resource.
crs	A CRS associated with the resource.
fees	Any fees associated with obtaining the resource or a copy of the resource.
hasRepositoryItem	A Boolean flag indicating that the catalogue record representing a resource also has an associated repository item entry.
hoursOfService	Any hours of service associated with the resource.
keyword	A list of keywords associated with the resource.
layerLimit	The layer limit associated with a WMS service.
legendUrl	The legend URL for a WMS.
method	HTTP method name for a service binding.
maxHeight	The maximum map height that a WMS can serve.
maxWidth	The maximum map width that a WMS can serve.
onlineResource	May be used to provide a reference to the provider of the resource.
outputFormat	A list of formats indicating the representations available for the resource.
position	The position with an organization if the resource is a person.
queryable	A Boolean indicating whether a WMS map layer is queryable.
resourceType	A subtype for a resource represented in a CSW-ebRIM catalogue. This slot is typically used to differentiate one OGC web service type from another.
role	A role associated with a resource.
supportedVersions	The list of supported version of the resource; typically supported server versions.
scheme	An identifier from a classification scheme that is associated with the resource.
spatialPropertyName	The name of the spatial property of a resource (e.g. the spatial property name of a WFS feature type)

spatialPropertyType	The type of the spatial property of a resource (e.g. the spatial property of a WFS is of type POINT)
supportedOperations	A list of operations supported by the resource.
targetNamespace	A namespace associated with the resource.
title	A human-readable title associated with a resource.
url	A URL associated with a resource. Typically used to provide a link for accessing the resource directly.
where	A location or geometry associated with a resource

NOTE

The complete identifier is formed by appending the value in the Identifier column to the prefix "urn:ogc:ebxml-regrep:slotName:".

10.2.7. Identifiers for data types for ebRIM

The following tables list identifiers for spatial and non-spatial data types. These values may be used to indicate the value type of a slot parameter. The identifiers in Table 11 are defined in the ebRIM standard (see OASIS regrep-rim-3.0-os). The identifiers in Table 12 are defined as part of the Basic Package of the csw-ebRIM standard (see OGC 07-144r4).

Table 11. Datatype Identifiers

Identifier	Description
urn:oasis:names:tc:ebxml-regrep:DataType:Boolean	Boolean
urn:oasis:names:tc:ebxml-regrep:DataType:Date	Date
urn:oasis:names:tc:ebxml-regrep:DataType:DateTime	DateTime
urn:oasis:names:tc:ebxml-regrep:DataType:Double	Double
urn:oasis:names:tc:ebxml-regrep:DataType:Float	Float
urn:oasis:names:tc:ebxml-regrep:DataType:Integer	Integer
urn:oasis:names:tc:ebxml-regrep:DataType:Number	Number
urn:oasis:names:tc:ebxml-regrep:DataType:Raw	Raw
urn:oasis:names:tc:ebxml-regrep:DataType:String	String
urn:oasis:names:tc:ebxml-regrep:DataType:Unknown	Unknown
urn:oasis:names:tc:ebxml-regrep:DataType:URI	URI

Table 12. Spatial Datatype Identifiers

Identifier	Description
urn:ogc:def:dataType:ISO-19107:GM_Point	Point
urn:ogc:def:dataType:ISO-19107:GM_Curve	Line

urn:ogc:def:dataType:ISO-19107:GM_Surface	Polygon
urn:ogc:def:dataType:ISO-19107:GM_MultiPoint	Multi-point
urn:ogc:def:dataType:ISO-19107:GM_MultiCurve	Multi-line
urn:ogc:def:dataType:ISO-19107:GM_MultiSurface	Multi-polygon
urn:ogc:def:dataType:ISO-19107:GM_Aggregate	Geometry collection
urn:ogc:def:dataType:ISO-19107:GM_Object	Geometry object of any type

10.3. Classification schemes for ebRIM

Annex B defines an ebRIM packages for service integration that defines that following classification schemes: ServiceTypes, HarvestResourceTypes, SlotNames.

10.4. Mapping OGC capabilities documents into ebRIM

10.4.1. Introduction

The process of registering an OGC web service into the catalogue involves reading the service's capabilities document and then mapping the contents of that document into ebRIM thus making the information discoverable to catalogue clients.

OGC capabilities documents are composed of a number of sections some of which are common to all services and some of which are specific to a particular service. This clause describes the mapping of the common sections into ebRIM and further describes the mapping of the content sections for the WIS, WMS, WMTS, WFS and WCS services into ebRIM.

10.4.2. ServiceIdentification section

The following table maps the element of the ServiceIdentification section of an OGC capabilities document into ebRIM.

Table 13. Mapping of ServiceIdentification Section into ebRIM

OGC Capabilities XPath	ebRIM XPath
Title	Service/Slot[@name="Title"]/ValueList/Value[1]
Abstract	Service/Description/LocalizedString/@value
Keywords	Service/Slot[@name="keyword"]/ValueList/Value[1..*]
ServiceType	Service/Slot[@name="resourceType"]/ValueList/Value[1]
ServiceTypeVersion	Service/Slot[@name="supportedVersion"]/ValueList/Value[1..*]]
Fees	Service/Slot[@name="fees"]/ValueList/Value[1]
AccessConstraints	Service/Slot[@name="accessConstraints"]/ValueList/Value[1]

10.4.3. ServiceProvider section

The following table maps the elements of the ServiceProvider section of an OGC capabilities document into ebRIM.

Table 14. Mapping of ServiceProvider Section into ebRIM

OGC Capabilities XPath	ebRIM XPath
ProviderName	Organization/Name
ProviderSite	Organization/Slot[@name="onlineResource"]/ValueList/Value[1]
ServiceContact	see Mapping of ServiceProvider/ServiceContact into ebRIM

NOTE Organization/@parent = Service/@lid

Table 15. Mapping of ServiceProvider/ServiceContact into ebRIM

OGC Capabilities XPath	ebRIM XPath
ServiceContact/Individual Name	Person/PersonName/@lastName
ServiceContact/Position Name	Person/Slot[@name="position"]/ValueList/Value[1]
ServiceContact/Role	Person/Slot[@name="role"]/ValueList/Value[1]
ServiceContact/ContactInfo	see Mapping of ServiceProvider/ServiceContact/ContactInfo into ebRIM

NOTE Organization/@primaryContact = Person/@lid

Table 16. Mapping of ServiceProvider/ServiceContact/ContactInfo into ebRIM

OGC Capabilities XPath	ebRIM XPath
OnlineResource	Organization/Slot[@name="onlineResource"]/ValueList/Value[1]
HoursOfService	Organization/Slot[@name="hoursOfService"]/ValueList/Value[1]
ContactInstructions	Organization/Slot[@name="contactInstructions"]/ValueList/Value[1]
Phone/Voice	Person/TelephoneNumber/[@phoneType="Voice"]/number
Phone/Facsimile	Person/TelephoneNumber/[@phoneType="ContactInfo"]/number
Address/DeliveryPoint	Organization/Address/@streetNumber
Address/City	Organization/Address/@city

Address/AdministrativeArea	Organization/Address/@stateOrProvince
Address/PostalCode	Organization/Address/@postalCode
Address/Country	Organization/Address/@country
Address/ElectronicMailAddress	Organization/EmailAddress/@address

10.4.4. OperationsMetadata section

The following table maps the elements of the OperationsMetadata section of an OGC capabilities document into ebRIM.

Table 17. Mapping of OperationsMetadata Section into ebRIM

OGC Capabilities XPath	ebRIM XPath
Operation/@name	ServiceBinding/Name/LocalizedString/@value
Operation/DCP/HTTP/Get/@xlink	ServiceBinding/@accessURI
ServiceBinding/Slot[@name="method"]/ValueList/Value[1]=GET	Operation/DCP/HTTP/Post/@xlink
ServiceBinding/@accessURI	ServiceBinding/Slot[@name="method"]/ValueList/Value[1]=POST

10.4.5. Content section

Introduction

This clause defines the mapping of the content sections of the WIS, WMS, WFS and WCS into the ebRIM model.

10.4.6. WIS content

The content section of a WIS is simply a list of ServiceIdentification sections describing the set of services that the WIS offers. The content section of a WIS is not harvested directly. Rather each service listed in the WIS content section is harvested and an *OffersService* association is registered between the WIS and the harvested child service.

WMS content

WMS theme

The following table maps the metadata describing a WMS layer that contains nested layers into ebRIM. Layers that in turn contain child layers are referred to as themes in this standard.

The relationship between a WMS offering a theme and the theme itself shall be encoded in ebRIM

using the *operatesOn* association.

The relationship between a WMS layer or theme and its child WMS layers shall be encoded in ebRIM using the *HasMember* association.

Table 18. Mapping of WMS Theme metadata into ebRIM

OGC Capabilities XPath	ebRIM XPath
Name	ExtrinsicObject[@objectType="wms:theme"]/Name/LocalizedString/@value
Description/Title	ExtrinsicObject[@objectType="wms:theme"]/Slot[@name="title"]/ValueList/Value[1]
Description/Abstract	ExtrinsicObject[@objectType="wms:theme"]/Description/LocalizedString/@value
Description/Keywords	ExtrinsicObject[@objectType="wms:theme"]/Slot[@name="keyword"]/ValueList/Value[1..*]
CRS	ExtrinsicObject[@objectType="wms:theme"]/Slot[@name="crs"]/ValueList/Value[1]
EX_GeographicBoundingBox or BoundingBox	ExtrinsicObject[@objectType="wms:theme"]/Slot[@name="where" && @type="urn:ogc:def:dataType:ISO-19107:2003:GM_Envelope]/ValueList/Value[1..*]
LegendURL	ExtrinsicObject[@objectType="wms:theme"]/Slot[@name="legendURL"]/ValueList/Value[1]=LegendURL/onlineResource ExtrinsicObject[@objectType="wms:theme"]/Slot[@name="legendURL"]/ValueList/Value[2]=LegendURL/Format ExtrinsicObject[@objectType="wms:theme"]/Slot[@name="legendURL"]/ValueList/Value[3]=LegendURL/width ExtrinsicObject[@objectType="wms:theme"]/Slot[@name="legendURL"]/ValueList/Value[4]=LegendURL/height
Layer (child)	Each child layer is encoded into ebRIM as a sub-theme of <i>WMS layer</i> . The association between a layer and a child layer or theme is encoded into ebRIM using the <i>HasMember</i> association.

WMS layer

The following table maps the metadata describing a WMS layer into ebRIM. The relationship between a WMS offering a layer and the layer itself shall be encoded in ebRIM using the *operatesOn* association.

The relationship between a WMS theme and a WMS layer it contains shall be encoded in ebRIM using the *HasMember* association.

Table 19. Mapping of WMS Layer metadata into ebRIM

OGC Capabilities XPath	ebRIM XPath
------------------------	-------------

Name	ExtrinsicObject[@objectType="wms:layer"]/Name/LocalizedString/@value
Description/Title	ExtrinsicObject[@objectType="wms:layer"]/Slot[@name="title"]/ValueList/Value[1]
Description/Abstract	ExtrinsicObject[@objectType="wms:layer"]/Description/LocalizedString/@value
Description/Keywords	ExtrinsicObject[@objectType="wms:layer"]/Slot[@name="keyword"]/ValueList/Value[1..*]
CRS	ExtrinsicObject[@objectType="wms:layer"]/Slot[@name="crs"]/ValueList/Value[1]
EX_GeographicBoundingBox or BoundingBox	ExtrinsicObject[@objectType="wms:layer"]/Slot[@name="where" && @type="urn:ogc:def:dataType:ISO-19107:2003:GM_Envelope]/ValueList/Value[1..*]
LegendURL	ExtrinsicObject[@objectType="wms:layer"]/Slot[@name="legendURL"]/ValueList/Value[1]=LegendURL/onlineResource ExtrinsicObject[@objectType="wms:layer"]/Slot[@name="legendURL"]/ValueList/Value[2]=LegendURL/Format ExtrinsicObject[@objectType="wms:layer"]/Slot[@name="legendURL"]/ValueList/Value[3]=LegendURL/width ExtrinsicObject[@objectType="wms:layer"]/Slot[@name="legendURL"]/ValueList/Value[4]=LegendURL/height

WMTS content

WMTS themes

The following table maps the metadata describing a WMTS theme into ebRIM. In addition to these mappings, an *operatesOn* association shall be created between the service offering the WMTS layer and the WMTS layer itself.

The relationship between a theme, its sub-themes and any WMTS layers shall be encoded in ebRIM using the *HasMember* association.

Table 20. Mapping of WMTS Theme metadata into ebRIM

OGC Capabilities XPath	ebRIM XPath
Title	ExtrinsicObject[@objectType="wmts:theme"]/Slot[@name="title"]/ValueList/Value[1]
Abstract	ExtrinsicObject[@objectType="wmts:theme"]/Description/LocalizedString[@value="..."]
Keywords	ExtrinsicObject[@objectType="wmts:theme"]/Slot[@name="keyword"]/ValueList/Value[1..*]
Identifier	ExtrinsicObject[@objectType="wmts:theme"]/Name/LocalizedString/@value[1]

LayerRef	This is encoded in ebRIM as a <i>HasMember</i> association between the theme and each child layer.
----------	--

WMTS layers

The following table maps the metadata describing a WMTS layer into ebRIM. In addition to these mappings, an *operatesOn* association shall be created between the service offering the WMTS layer and the WMTS layer itself.

Table 21. Mapping of WMTS Layer metadata into ebRIM

OGC Capabilities XPath	ebRIM XPath
Title	ExtrinsicObject[@objectType="wmts:layer"]/Slot[@name="title"]/ValueList/Value[1]
Abstract	ExtrinsicObject[@objectType="wmts:layer"]/Description/LocalizedString[@value="..."]
Keywords	ExtrinsicObject[@objectType="wmts:layer"]/Slot[@name="keyword"]/ValueList/Value[1..*]
WGS84BoundingBox	ExtrinsicObject[@objectType="wmts:layer"]/Slot[@name="where" && @type="urn:ogc:def:dataType:ISO-19107:2003:GM_Envelope"]/ValueList/Value[1..*]
Identifier	ExtrinsicObject[@objectType="wmts:layer"]/Name/LocalizedString/@value
LegendURL	ExtrinsicObject[@objectType="wmts:layer"]/Slot[@name="legendURL"]/ValueList/Value[1]=href ExtrinsicObject[@objectType="wmts:layer"]/Slot[@name="legendURL"]/ValueList/Value[2]=format ExtrinsicObject[@objectType="wmts:layer"]/Slot[@name="legendURL"]/ValueList/Value[3]=minScaleDenominator ExtrinsicObject[@objectType="wmts:layer"]/Slot[@name="legendURL"]/ValueList/Value[4]=maxScaleDenominator ExtrinsicObject[@objectType="wmts:layer"]/Slot[@name="legendURL"]/ValueList/Value[5]=width ExtrinsicObject[@objectType="wmts:layer"]/Slot[@name="legendURL"]/ValueList/Value[6]=height
Format	ExtrinsicObject[@objectType="wmts:layer"]/Slot[@name="outputFormat"]/ValueList/Value[1..*]

WFS feature types

The following table maps the metadata describing a WFS feature type into ebRIM. In addition to these mappings, an *operatesOn* association shall be created between the service offering the feature type and the feature type itself.

Table 22. Mapping of WFS FeatureType metadata into ebRIM

OGC Capabilities XPath	ebRIM XPath
------------------------	-------------

Name	ExtrinsicObject[@objectType="wfs:featureType"]/Name/LocalizedString/@value
Title	ExtrinsicObject[@objectType="wfs:featureType"]/Slot[@name="title"]/ValueList/Value[1]
Abstract	ExtrinsicObject[@objectType="wfs:featureType"]/Description/LocalizedString/@value
Keywords	ExtrinsicObject[@objectType="wfs:featureType"]/Slot[@name="keyword"]/ValueList/Value[1..*]
DefaultCRS or OtherCRS	ExtrinsicObject[@objectType="wfs:featureType"]/Slot[@name="crs"]/ValueList/Value[1]=DefaultCRS ExtrinsicObject[@objectType="wfs:featureType"]/Slot[@name="crs"]/ValueList/Value[2..*]=OtherCRS
OutputFormats	ExtrinsicObject[@objectType="wfs:featureType"]/Slot[@name="outputFormat"]/ValueList/Value[1..*]
WGS84BoundingBox	ExtrinsicObject[@objectType="wfs:featureType"]/Slot[@name="where" && @type="urn:ogc:def:dataType:ISO-19107:2003:GM_Envelope]/ValueList/Value[1..*]
MetadataURL	ExtrinsicObject[@objectType="wfs:featureType"]/Slot[@name="metadataURL"]/ValueList/Value[1]

WCS Coverage

The following table maps the metadata describing a WCS coverage offering into ebRIM. In addition to these mappings, an *operatesOn* association shall be created between the service offering the coverage and the coverage itself.

Table 23. Mapping of WCS Coverage Offering into ebRIM

OGC Capabilities XPath	ebRIM XPath
Title	ExtrinsicObject[@objectType="wcs:coverage"]/Slot[@name="title"]/ValueList/Value[1]
Abstract	ExtrinsicObject[@objectType="wcs:coverage"]/Description/LocalizedString/@value
Keywords	ExtrinsicObject[@objectType="wcs:coverage"]/Slot[@name="keyword"]/ValueList/Value[1..*]
WGS84BoundingBox or BoundingBox	ExtrinsicObject[@objectType="wcs:coverage"]/Slot[@name="where" && @type="urn:ogc:def:dataType:ISO-19107:2003:GM_Envelope"]/ValueList/Value[1..*]
CoverageId	ExtrinsicObject[@objectType="wcs:coverage"]/Name/LocalizedString/@value
CoverageSubtype	ExtrinsicObject[@objectType="wcs:coverage"]/Slot[@name="resourceType"]/ValueList/Value[1]

10.5. GetAssociations stored query

CSW-ebRIM implementations shall implement a stored query named "GetAssociations" that allows clients to query associations in the catalogue. The prototype of the stored query shall be:

```
getAssociations( sourceId anyURI, localSourceId string,  
                associationType TokenOrAnyURI,  
                targetId anyURI, localTargetId string )
```

The methods shall behave as described for the [_GetAssociations operation_](#).

Chapter 11. Security considerations

11.1. Advertising Security

With so much focus on security in Testbed 12, it seems only natural that the WIS should also discuss how to deal with this concept. Using the guidelines outlined in the engineering document OGC 15-022, we can easily advertise the security constraints in the WIS GetCapabilities document by using the appropriate OWS Common Constraint element in the GetCapabilities operation as outlined below:

```
<ows:Operation name="GetCapabilities">
  <ows:DCP>
    <ows:HTTP>
      <ows:Get xlink:href="http://ogc-
testbed12.compusult.net/wes/wis?request=getCapabilities">
        <ows:Constraint name="urn:ogc:def:tb12:security:authentication">
          <ows:ValuesReference
ows:reference="http://tb12.opengis.net/security/authCodeList#HTTP_BASIC">
            urn:ogc:def:tb12:ietf:2617:basic</ows:ValuesReference>
          </ows:Constraint>
        </ows:Get>
      </ows:HTTP>
    </ows:DCP>
    <ows:Parameter name="Format">
      <ows:AllowedValues>
        <ows:Value>application/xml</ows:Value>
        <ows:Value>application/x-bxml</ows:Value>
      </ows:AllowedValues>
    </ows:Parameter>
  </ows:Operation>
```

The general idea here is that service providers expose a public capabilities document. Then, within that document they advertise the methods that are available for users and/or clients to authenticate against. Any entity that authenticates successfully would then be given access to other services that would not be accessible otherwise.

In the sample above, the 'GetCapabilities' operation is advertising that users and/or clients can authenticate using 'Basic Authentication' but other methods are also available including: client certificate and user name token. A complete list of the available methods can be found here: <http://tb12.opengis.net/security/authCodeList>

11.2. Storing Security information in an ebRIM CSW Catalogue.

When security information is advertised in a service's capabilities document it is very important that this information is captured properly so that it can be accessed and/or displayed later.

Typically, when a service capabilities document is published in a CSW catalogue it creates an ebRIM Service object that is then associated with various applicable objects within the ebRIM data model.

Because a service’s associated security constraints are tied to its operations, it is recommended that a new ebRIM Association object with an association type called urn:ogc:def:ebRIM-AssociationType:OGC:HasConstraint be used to associate each ebRIM service binding to its associated constraints.

Each constraint object should be a new ExtrinsicObject with an object type = urn:ogc:def:ebRIM-ObjectType:OGC:Constraint that has an associated Slot called AllowedValues which contains the list of values permitted for the constraint. The figure, *ebRIM Service Security Association* shows what an ebRIM catalogue might look like after a WIS capabilities document has been successfully harvested.

The ebRIM Service Security Association

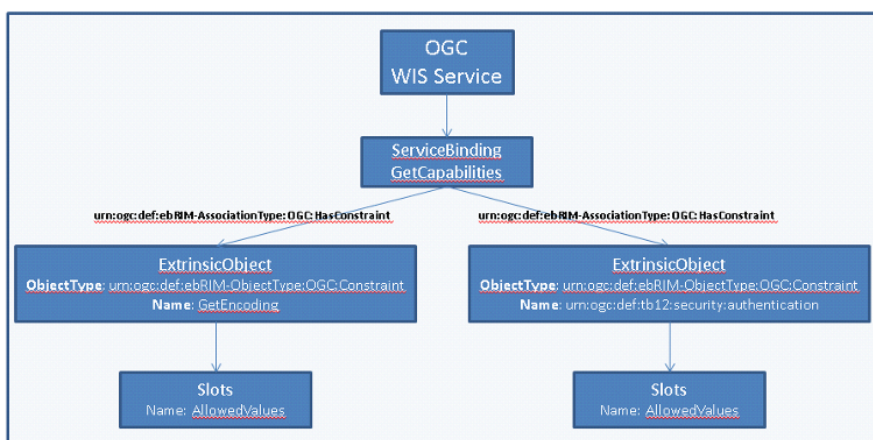


Figure 7. The ebRIM Service Security Association

11.3. Public and Private WIS Service objects

During a typical CSW publishing flow for a Web Integration Service (WIS) it is recommended that the CSW catalogue maintain which of the services listed in the GetCapabilities document have Public access and which ones have Private access. If you recall, the Private services are those returned by the service after the user and/or client has been authenticated.

It is recommended that this be accomplished by classifying each service properly using the new Access Types classification scheme. The classification scheme Access Types is defined as:

```

<rim:ClassificationScheme
  lid="urn:ogc:def:ebRIM-ClassificationScheme:AccessTypes"
  id="urn:ogc:def:ebRIM-ClassificationScheme:AccessTypes"
  isInternal="false" nodeType="urn:oasis:names:tc:ebxml-regrep:NodeType:UniqueCode">
  <rim:Name>
    <rim:LocalizedString charset="ISO-8859-1" xml:lang="en" value="Access Types" />
    <rim:LocalizedString charset="ISO-8859-1" xml:lang="fr" value="Types d'accès" />
  </rim:Name>
  <rim:ClassificationNode lid="urn:ogc:def:ebRIM-ClassificationNode:Private"
    code="Private" id="urn:ogc:def:ebRIM-ClassificationNode:Private">
    <rim:Name>
      <rim:LocalizedString charset="ISO-8859-1" xml:lang="en" value="Private" />
      <rim:LocalizedString charset="ISO-8859-1" xml:lang="fr" value="Privé" />
    </rim:Name>
  </rim:ClassificationNode>
  <rim:ClassificationNode lid="urn:ogc:def:ebRIM-ClassificationNode:Public"
    code="Public" id="urn:ogc:def:ebRIM-ClassificationNode:Public">
    <rim:Name>
      <rim:LocalizedString charset="ISO-8859-1" xml:lang="en" value="Public" />
      <rim:LocalizedString charset="ISO-8859-1" xml:lang="fr" value="Public" />
    </rim:Name>
  </rim:ClassificationNode>
</rim:ClassificationScheme>

```

Figure 8. Classification Scheme of Access Types

Chapter 12. Compusult WIS implementation details

12.1. Web Integration Service endpoint

The WIS endpoint consists of a WIS servlet endpoint, capabilities harvester, and database cache.

To allow flexibility in the content returned by the WIS, the GetCapabilities response is generated on demand from a list of services. The list of service endpoints known to the WIS is specified in an administrator-configurable property.

The WIS invokes a capabilities harvester to retrieve the capabilities for each known service. Metadata parsed from the capabilities responses are timestamped and cached in a database table. The timestamp is used by the WIS to decide whether to use the cached metadata or retrieve a fresh capabilities response.

The WIS cache also records security information for each service. For the purposes of Testbed 12, the security information simply indicates whether or not the service is exposed in the WIS public capabilities. This could be extended in the future to accommodate additional security restrictions.

12.2. Security

Compusult's WIS endpoint has been implemented as a component of Web Enterprise Suite (WES). Access to the WIS endpoint is controlled by WES, requiring authentication via HTTP Basic Auth or PKI client certificate.

In addition to the regular service endpoint, the WIS implementation exposes a publicly-accessible capabilities URL that advertises security annotations as described by OGC 16-048 (OWS Common Security Extension ER).

An unresolved issue is how to reconcile security restrictions for the WIS, its contents, and the catalog. In an operational scenario, the WIS and catalog could have separate authentication credentials. Each service in a WIS could have its own security constraints. Catalog publishing clients will need to be aware of this and handle authentication for each service appropriately. Additionally, a catalog containing anything other than public data will need to enforce security restrictions on the published records.

Compusult's WIS implementation deals with these issues in the following manner:

- With respect to visibility, each service is treated as simply either public or non-public.
- When publishing to the catalog, the same authentication credentials are used for both the catalog and the WIS.
- The public capabilities endpoints are used when publishing services to the catalog.

Appendix A: XML Schema Documents

- Introduction This annex contains the normative schemas for the *Web Integration Service*, the *GetAssociations request* and the *GetAssociations response*.
- **wis.xsd**

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
  targetNamespace="http://www.opengis.net/wis/1.0"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wis="http://www.opengis.net/wis/1.0"
  xmlns:ows="http://www.opengis.net/ows/2.0"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  elementFormDefault="qualified" version="1.0.0">
  <xsd:annotation>
    <xsd:documentation>This XML Schema Document includes and imports,
      directly or indirectly, all the XML Schemas defined by the
      Web Integration Service (WIS) Implementation Specification.

      WIS is an OGC Engineering Report.
      Copyright (c) 2014 Open Geospatial Consortium.
      To obtain additional rights of use,
      visit http://www.opengeospatial.org/legal/ .
    </xsd:documentation>
  </xsd:annotation>
  <!-- =====
    Includes and Imports
    ===== -->
  <xsd:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/xml.xsd"/>
  <xsd:import namespace="http://www.w3.org/1999/xlink"
    schemaLocation="http://www.w3.org/1999/xlink.xsd"/>
  <xsd:import namespace="http://www.opengis.net/ows/2.0"
    schemaLocation="http://schemas.opengis.net/ows/2.0/owsAll.xsd"/>

  <!-- ===== -->
  <!-- =   GETCAPABILITIES Request and Response   = -->
  <!-- ===== -->
  <!-- REQUEST -->
  <xsd:element name="GetCapabilities" type="wis:GetCapabilitiesType"/>
  <xsd:complexType name="GetCapabilitiesType">
    <xsd:complexContent>
      <xsd:extension base="ows:GetCapabilitiesType">
        <xsd:attribute name="service" type="ows:ServiceType"
          use="required" fixed="WIS"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

```

        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<!-- RESPONSE -->
<xsd:element name="Capabilities" type="wis:WIS_CapabilitiesType"/>
<xsd:complexType name="WIS_CapabilitiesType">
    <xsd:complexContent>
        <xsd:extension base="ows:CapabilitiesBaseType">
            <xsd:sequence>
                <xsd:element ref="wis:Contents"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:element name="Contents" type="wis:ContentsType"/>
<xsd:complexType name="ContentsType">
    <xsd:sequence>
        <xsd:element ref="wis:Service" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:element name="Service" type="wis:ServiceType"/>
<xsd:complexType name="ServiceType">
    <xsd:sequence>
        <xsd:element ref="ows:ServiceIdentification"/>
    </xsd:sequence>
    <xsd:attribute name="type" type="wis:TokenOrAnyURI" use="required"/>
    <xsd:attribute name="baseUrl" type="xsd:anyURI" use="required"/>
</xsd:complexType>
<xsd:simpleType name="TokenOrAnyURI">
    <xsd:union memberTypes="xsd:token xsd:anyURI"/>
</xsd:simpleType>
</xsd:schema>

```

- **owsGetAssociations.xsd**

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
    targetNamespace="http://www.opengis.net/ows/2.0"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    elementFormDefault="qualified"
    version="1.0.0">
    <xsd:annotation>
        <xsd:documentation>
            This XML schema document provides a schema for the
            GetAssociations operation which allows a service to
            advertise its associations.
        </xsd:documentation>
    </xsd:annotation>

```

Copyright (c) 2014 Open Geospatial Consortium.

```

    To obtain additional rights of use,
    visit http://www.opengeospatial.org/legal/ .
  </xsd:documentation>
</xsd:annotation>

<!-- =====
Includes and Imports
===== -->
<xsd:import namespace="http://www.w3.org/XML/1998/namespace"
            schemaLocation="http://www.w3.org/2001/xml.xsd"/>
<xsd:import namespace="http://www.w3.org/1999/xlink"
            schemaLocation="http://www.w3.org/1999/xlink.xsd"/>
<xsd:include schemaLocation="../../ows/2.0/owsAll.xsd"/>

<!-- =====
GETASSOCIATIONS REQUEST
===== -->
<xsd:element name="GetAssociations" type="ows:GetAssociationsType"/>
<xsd:complexType name="GetAssociationsType">
  <xsd:sequence>
    <xsd:element name="Content" type="as:StringOrQName"
                minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="service"
                type="ows:ServiceType" use="required"/>
  <xsd:attribute name="version" type="xsd:string"
                use="required" fixed="1.0.0"/>
  <xsd:attribute name="outputFormat"
                type="xsd:string" default="application/as+xml"/>
</xsd:complexType>
<xsd:simpleType name="StringOrQName">
  <xsd:union memberTypes="xsd:string xsd:QName"/>
</xsd:simpleType>

<!-- =====
GETASSOCIATIONS RESPONSE
===== -->
<!-- see associations.xsd -->

</xsd:schema>

```

- **associations.xsd**

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
  targetNamespace="http://www.opengis.net/associations/1.0"
  xmlns:assoc="http://www.opengis.net/associations/1.0"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"

```

```

elementFormDefault="qualified"
version="1.0.0">
<xsd:annotation>
  <xsd:documentation>This XML Schema Document includes and imports,
  directly or indirectly, all the XML Schemas defined by the
  Web Integration Service (WIS) Engineering Report.

  WIS is an OGC Engineering Report.
  Copyright (c) 2010 Open Geospatial Consortium.
  To obtain additional rights of use,
  visit http://www.opengeospatial.org/legal/ .
  </xsd:documentation>
</xsd:annotation>
<!-- =====
      Includes and Imports
      ===== -->
<xsd:import namespace="http://www.w3.org/XML/1998/namespace"
  schemaLocation="http://www.w3.org/2001/xml.xsd"/>
<xsd:import namespace="http://www.w3.org/1999/xlink"
  schemaLocation="http://www.w3.org/1999/xlink.xsd"/>
<!-- ===== -->
<!-- =   BASE TYPES   = -->
<!-- ===== -->
<xsd:complexType name="BaseRequestType" abstract="true">
  <xsd:attribute name="service"
    type="xsd:string" use="required" fixed="WOS"/>
  <xsd:attribute name="version" type="assoc:VersionStringType"
    use="required"/>
  <xsd:attribute name="handle" type="xsd:string"/>
</xsd:complexType>
<xsd:simpleType name="VersionStringType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="1\.0\.\d+"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- ===== -->
<xsd:complexType name="EmptyType"/>
<!-- ===== -->
<xsd:complexType name="DescriptionType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <!-- Key is a handle identifying the kind of description
      (e.g. key="title" or key="abstract"). -->
      <xsd:attribute name="key" type="xsd:string" use="required"/>
      <xsd:attribute name="charset" type="xsd:string"
        use="optional" default="UTF8"/>
      <xsd:attribute ref="xml:lang" use="optional" default="en"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<!-- ===== -->

```

```

<xsd:complexType name="DynamicPropertyType" mixed="true">
  <xsd:sequence>
    <xsd:any minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" use="required"/>
  <xsd:attribute name="type" type="xsd:string" use="required"/>
  <xsd:attribute name="codeSpace" type="xsd:anyURI" use="optional"/>
</xsd:complexType>
<!-- ===== -->
<xsd:complexType name="BaseElementType">
  <xsd:sequence>
    <xsd:element name="Description" type="assoc:DescriptionType"
      minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="Property" type="assoc:DynamicPropertyType"
      minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<!-- ===== -->
<xsd:simpleType name="TokenOrAnyURI">
  <xsd:union memberTypes="xsd:token xsd:anyURI"/>
</xsd:simpleType>
<!-- ===== -->
<xsd:element name="Identifier">
  <xsd:annotation>
    <xsd:documentation>
      An identifier of a particular resource within an OGC web service.
      If the identifier is inherently namespace-qualified (as is the case,
      for example, with WFS feature types), then it should indicate the
      namespace via the "namespace" attribute.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="namespace" type="xsd:anyURI">
          <xsd:annotation>
            <xsd:documentation>
              The namespace (if applicable) of the identifier.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
<!-- === RESOURCE TYPE ===== -->
<xsd:complexType name="ResourceType">
  <xsd:annotation>
    <xsd:documentation>
      A web-accessible resource. A resource may be either
      a URL-addressible resource or an OGC-Web-Service resource.
    </xsd:documentation>
  </xsd:annotation>

```



```

        </xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
</xsd:complexType>
<!-- === ASSOCIATIONS (i.e. collection of Association) ===== -->
<xsd:element name="Associations">
    <xsd:annotation>
        <xsd:documentation>A set of associations.</xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="assoc:Association"
                minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="version" type="xsd:string"
            use="required" fixed="1.0.0"/>
    </xsd:complexType>
</xsd:element>
<!-- === ASSOCIATION ===== -->
<xsd:element name="Association" type="assoc:AssociationType"/>
<xsd:complexType name="AssociationType">
    <xsd:annotation>
        <xsd:documentation>
            An association between two resources. An association is logically
            of the form "SOURCE REL TARGET", where the SOURCE resource is the
            subject of a directional RELationship of the specified type with
            the TARGET resource. E.g., "layer X renders feature type Y".
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
        <xsd:extension base="assoc:BaseElementType">
            <xsd:sequence>
                <xsd:element name="Source" type="assoc:ResourceType">
                    <xsd:annotation>
                        <xsd:documentation>
                            The source resource of the association.
                        </xsd:documentation>
                    </xsd:annotation>
                </xsd:element>
                <xsd:element name="Target" type="assoc:ResourceType">
                    <xsd:annotation>
                        <xsd:documentation>
                            The target resource of the association.
                        </xsd:documentation>
                    </xsd:annotation>
                </xsd:element>
            </xsd:sequence>
            <xsd:attribute name="id"
                type="assoc:TokenOrAnyURI" use="optional">
                <xsd:annotation>
                    <xsd:documentation>

```

An optional identifier for this association. The intent is to be able to assign an identifier to an association when it is considered a first class object.

```
</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
<xsd:attribute name="rel"
               type="assoc:TokenOrAnyURI" use="required">
  <xsd:annotation>
    <xsd:documentation>
      The association type, using a controlled vocabulary.
      (E.g., "renders", "renderedBy", "tiles", "tiledBy",
       "cascades", "cascadedBy", "alternateSource", "describes",
       "describedBy".)

      The vocabulary of this attribute is defined and controlled
      by specific application domains which should specify a set
      of standard values and their meanings, but which shall not
      constrain the use to only these values. Parsers shall not
      fail on unrecognized values of this attribute, and are free
      to ignore associations that specify an unrecognized value
      for this attribute.
    </xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>
```

Appendix B: OGC Service Integration ebRIM Package

- ebRIM Package

The following ebRIM package defines object identifiers, classification schemes and association that may be used by a CSW-ebRIM catalogue to harvest and register the resources described by the capabilities documents of OGC web services.

```
<?xml version="1.0" encoding="UTF-8"?>
<csw:Transaction
  xmlns="http://www.opengis.net/cat/csw/2.0.2"
  xmlns:csw="http://www.opengis.net/cat/csw/2.0.2"
  xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/cat/csw/2.0.2
    http://schemas.opengis.net/csw/2.0.2/CSW-discovery.xsd
    urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0
    http://docs.oasis-open.org/regrep/v3.0/schema/rim.xsd"
  service="CSW" version="2.0.2">
  <!-- ===== -->
  <!-- OGC Association types -->
  <!-- ===== -->
  <csw:Insert>
    <rim:ClassificationNode code="OGC-Association-Types"
      id="urn:ogc:ebxml-regrep:classificationScheme:AssociationTypes"
      parent="urn:oasis:names:tc:ebxml-
regrep:classificationScheme:AssociationType">
      <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="OGC-AssociationType"/>
      </rim:Name>
      <rim:Description>
        <rim:LocalizedString charset="UTF-8"
          value="A sub-tree defining association types defined by OGC"/>
      </rim:Description>
      <rim:ClassificationNode code="Renders"
        id="urn:ogc:associationType:renders">
        <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
          <rim:ValueList>
            <rim:Value>Renders</rim:Value>
          </rim:ValueList>
        </rim:Slot>
        <rim:Slot name="urn:ogc:ebxml-regrep:slotName:reverseAssociationTitle">
          <rim:ValueList>
            <rim:Value>Rendered by</rim:Value>
          </rim:ValueList>
        </rim:Slot>
      </rim:Name>
```

```

    <rim:LocalizedString charset="UTF-8" value="Renders"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="Associates one or more data sources (typically WFS Feature
Types) that are rendered by a WMS or WMTS layers."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="TiledBy"
  id="urn:ogc:associationType:tiledBy">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Tiled By</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:reverseAssociationTitle">
    <rim:ValueList>
      <rim:Value>Tiles</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="Tiled By"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="Associates one or more WMTS layers with the WMS layer that
they tile."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="ParentOf"
  id="urn:ogc:associationType:parentOf">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Parent Of</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:reverseAssociationTitle">
    <rim:ValueList>
      <rim:Value>Child Of</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="Parent Of"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="Associates a registry entry with a parent entry and is
typically used to register a nested hierarchy of objects."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="Cascades"

```

```

id="urn:ogc:associationType:cascades">
<rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
  <rim:ValueList>
    <rim:Value>Cascades</rim:Value>
  </rim:ValueList>
</rim:Slot>
<rim:Slot name="urn:ogc:ebxml-regrep:slotName:reverseAssociationTitle">
  <rim:ValueList>
    <rim:Value>Cascaded By</rim:Value>
  </rim:ValueList>
</rim:Slot>
<rim:Name>
  <rim:LocalizedString charset="UTF-8" value="Cascades"/>
</rim:Name>
<rim:Description>
  <rim:LocalizedString charset="UTF-8"
    value="Associates two services where one OGC web service is using
the other OGC web service as a data source. For examples where one WMS cascades a
number of other WMS's"/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="CreatedBy"
id="urn:ogc:associationType:createdBy">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Created By</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:reverseAssociationTitle">
    <rim:ValueList>
      <rim:Value>Creates</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="Created By"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="Associates two resources where one is responsible for the
other's creation."/>
    </rim:Description>
  </rim:ClassificationNode>
<rim:ClassificationNode code="Symbolizes"
id="urn:ogc:associationType:symbolizes">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Symbolizes</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:reverseAssociationTitle">
    <rim:ValueList>

```

```

        <rim:Value>Symbolized by</rim:Value>
    </rim:ValueList>
</rim:Slot>
<rim:Name>
    <rim:LocalizedString charset="UTF-8" value="Symbolizes"/>
</rim:Name>
<rim:Description>
    <rim:LocalizedString charset="UTF-8"
        value="Associates the symbology encoding document with a WFS Feature
Type."/>
</rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="OperatesOn"
    id="urn:ogc:associationType:operatesOn">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Operates On</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:reverseAssociationTitle">
        <rim:ValueList>
            <rim:Value>Operated On</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="OperatesOn"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="Associates two resource when one resource operates upon the
other resource (e.g. a WFS service operates on feature types)."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="DescribedBy"
    id="urn:ogc:associationType:describedBy">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Described By</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:reverseAssociationTitle">
        <rim:ValueList>
            <rim:Value>Describes</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="DescribedBy"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="Associates two resource where one resource is a description,

```

using some metadata format, of another resource (e.g. an FGDC document might be used to describe a WFS feature type)."/>

```
    </rim:Description>
  </rim:ClassificationNode>
</rim:ClassificationNode>
</csw:Insert>
<!-- ===== -->
<!-- OGC Service Types -->
<!-- ===== -->
<csw:Insert>
  <rim:ClassificationScheme
    id="urn:ogc:ebxml-regrep:classificationScheme:ServiceTypes"
    isInternal="true"
    nodeType="urn:oasis:names:tc:ebxml-regrep:NodeType:UniqueCode">
    <rim:Name>
      <rim:LocalizedString charset="UTF-8" value="ServiceTypes"/>
    </rim:Name>
    <rim:Description>
      <rim:LocalizedString charset="UTF-8"
        value="This is the canonical classification scheme of OGC web service
types."/>
    </rim:Description>
    <rim:ClassificationNode code="wis"
      id="urn:ogc:serviceType:wis">
      <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
          <rim:Value>Web Integration Service</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="wis"/>
      </rim:Name>
      <rim:Description>
        <rim:LocalizedString charset="UTF-8"
          value="An aggregation service that offers other OGC web services."/>
      </rim:Description>
    </rim:ClassificationNode>
    <rim:ClassificationNode code="wfs"
      id="urn:ogc:serviceType:wfs">
      <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
          <rim:Value>Web Feature Service</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="wfs"/>
      </rim:Name>
      <rim:Description>
        <rim:LocalizedString charset="UTF-8"
          value="A service offering features."/>
      </rim:Description>
```



```

</rim:ClassificationNode>
<rim:ClassificationNode code="wos"
  id="urn:ogc:serviceType:wos">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Web Object Service</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="wos"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="A service offering generic (perhaps binary) objects."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="wms"
  id="urn:ogc:serviceType:wms">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Web Map Service</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="wms"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="A service offering maps."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="wmts"
  id="urn:ogc:serviceType:wmts">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Web Map Tiling Service</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="wmts"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="A service offering map tiles."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="wps"
  id="urn:ogc:serviceType:wps">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Web Processing Service</rim:Value>

```

```

        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="wps"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="A service making geo-processes accessible on the web."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="wcs"
    id="urn:ogc:serviceType:wcs">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Web Coverage Service</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="wcs"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="A service offering coverages."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="csw"
    id="urn:ogc:serviceType:csw">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Catalogue Service Web</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="csw"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="An web accessible catalogue service."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="csw-ebRIM"
    id="urn:ogc:serviceType:csw-ebRIM">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>CSW-ebRIM Registry Service - Part 1: ebRIM profile of
CSW</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="csw-ebRIM"/>
    </rim:Name>

```

```

    <rim:Description>
      <rim:LocalizedString charset="UTF-8"
        value="A catalogue profile using ebRIM."/>
    </rim:Description>
  </rim:ClassificationNode>
  <rim:ClassificationNode code="gss"
    id="urn:ogc:serviceType:gss">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
      <rim:ValueList>
        <rim:Value>Web GeoCollaboration Service</rim:Value>
      </rim:ValueList>
    </rim:Slot>
    <rim:Name>
      <rim:LocalizedString charset="UTF-8" value="csw-ebRIM"/>
    </rim:Name>
    <rim:Description>
      <rim:LocalizedString charset="UTF-8"
        value="A service supporting crowd sourced access to transaction OGC
web services."/>
    </rim:Description>
  </rim:ClassificationNode>
</rim:ClassificationScheme>
</csw:Insert>
<!-- ===== -->
<!-- OGC Service Content types. -->
<!-- ===== -->
<csw:Insert>
  <rim:ClassificationNode code="OGC-Service-Content-Types"
    id="urn:ogc:ebxml-regrep:classificationScheme:ContentTypes"
    parent="urn:oasis:names:tc:ebxml-regrep:classificationScheme:ObjectType">
    <rim:Name>
      <rim:LocalizedString charset="UTF-8" value="OGC-ObjectType"/>
    </rim:Name>
    <rim:Description>
      <rim:LocalizedString charset="UTF-8"
        value="A sub-tree defining OGC object types."/>
    </rim:Description>
    <rim:ClassificationNode code="FeatureType"
      id="urn:ogc:contentType:wfs:featureType">
      <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
          <rim:Value>WFS Feature Type</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="WFS Feature Type"/>
      </rim:Name>
      <rim:Description>
        <rim:LocalizedString charset="UTF-8"
          value="A feature type served by an OGC Web Feature Service."/>
      </rim:Description>

```

```

</rim:ClassificationNode>
<rim:ClassificationNode code="Object"
  id="urn:ogc:contentType:wos:object">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>WOS Object</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="WOS Object"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="A generic object (perhaps binary) served by an OGC Web Object
Service."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="WMS Theme"
  id="urn:ogc:contentType:wms:theme">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>WMS Theme</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="WMS Theme"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="A collection of WMS layers."/>
  </rim:Description>
<rim:ClassificationNode code="WMS Layer"
  id="urn:ogc:contentType:wms:layer">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>WMS Layer</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="WMS Layer"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="A map layer served by an OGC Web Map Service."/>
  </rim:Description>
</rim:ClassificationNode>
</rim:ClassificationNode>
<rim:ClassificationNode code="WMTS Theme"
  id="urn:ogc:contentType:wmts:theme">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>

```

```

        <rim:Value>WMTS Theme</rim:Value>
    </rim:ValueList>
</rim:Slot>
<rim:Name>
    <rim:LocalizedString charset="UTF-8" value="WMTS Theme"/>
</rim:Name>
<rim:Description>
    <rim:LocalizedString charset="UTF-8"
        value="A collection of WMTS layers."/>
</rim:Description>
<rim:ClassificationNode code="WMTS Layer"
    id="urn:ogc:contentType:wmts:layer">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>WMTS Layer</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="WMTS Layer"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="A WMTS map layer served by an OGC Web Map Service."/>
    </rim:Description>
</rim:ClassificationNode>
</rim:ClassificationNode>
<rim:ClassificationNode code="WCS Theme"
    id="urn:ogc:contentType:wcs:theme">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>WCS Theme</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="WCS Theme"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="A collection of WCS coverages."/>
    </rim:Description>
<rim:ClassificationNode code="WCS Coverage"
    id="urn:ogc:contentType:wcs:coverage">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>WCS Coverage</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="WCS Coverage"/>
    </rim:Name>
    <rim:Description>

```

```

        <rim:LocalizedString charset="UTF-8"
            value="A coverage served by an OGC Web Coverage Service."/>
    </rim:Description>
</rim:ClassificationNode>
</rim:ClassificationNode>
<rim:ClassificationNode code="Process"
    id="urn:ogc:contentType:wps:process">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>WPS Process</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="WPS Process"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="A process served by an OGC Web Processing Service."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="Unknown"
    id="urn:ogc:resourceType:unknown">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Unknown Resource Type</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="Unknown"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="An artifact or resource of unknown type."/>
    </rim:Description>
</rim:ClassificationNode>
</rim:ClassificationNode>
</csw:Insert>
<!-- ===== -->
<!-- Harvestable resource types. -->
<!-- ===== -->
<csw:Insert>
    <rim:ClassificationScheme
        id="urn:ogc:ebxml-regrep:classificationScheme:HarvestResourceTypes"
        isInternal="true"
        nodeType="urn:oasis:names:tc:ebxml-regrep:NodeType:UniqueCode">
        <rim:Name>
            <rim:LocalizedString charset="UTF-8" value="HarvestResourceType"/>
        </rim:Name>
        <rim:Description>
            <rim:LocalizedString charset="UTF-8"
                value="This is the canonical classification scheme for harvest resource

```

```

types used by the CSW-ebRIM catalogue."/>
</rim:Description>
<!-- Service capabilities documents. -->
<rim:ClassificationNode code="wis"
  id="urn:ogc:documentType:capabilities:wis">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Web Integration Service Capabilities Document</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="wis"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The capabilities document of a WIS."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="wfs"
  id="urn:ogc:documentType:capabilities:wfs">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Web Feature Service Capabilities Document</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="wfs"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The capabilities document of a Web Feature Service"/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="wms"
  id="urn:ogc:documentType:capabilities:wms">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Web Map Service Capabilities Document</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="wms"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The capabilities document of a Web Map Service"/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="wmts"
  id="urn:ogc:documentType:capabilities:wmts">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">

```

```

        <rim:ValueList>
            <rim:Value>Web Map Tiling Service Capabilities Document</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="wmts"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="The capabilities document of a Web Map Tiling Service"/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="wcs"
    id="urn:ogc:documentType:capabilities:wcs">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Web Coverage Service Capabilities Document</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="wcs"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="The capabilities document of a Web Coverage Service"/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="wps"
    id="urn:ogc:documentType:capabilities:wps">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Web Processing Service Capabilities Document</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="wps"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="The capabilities document of a Web Processing Service"/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="csw"
    id="urn:ogc:documentType:capabilities:csw">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Catalogue Service Web Capabilities Document</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="csw"/>

```



```

</rim:Name>
<rim:Description>
  <rim:LocalizedString charset="UTF-8"
    value="The capabilities document of a Catalogue Service Web"/>
</rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="csw-ebRIM"
  id="urn:ogc:documentType:capabilities:csw-ebRIM">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Web Registry Service Capabilities Document</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="csw-ebRIM"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The capabilities document of a Web Registry Service"/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="gss"
  id="urn:ogc:documentType:capabilities:gss">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Web GeoCollaboration Service (formerly GeoSynchronization
Service) Capabilities Document</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="gss"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The capabilities document of a Web GeoCollaboration Service
(formerly GeoSynchronization Service)."/>
  </rim:Description>
</rim:ClassificationNode>
<!-- Service capabilities documents referenced by base URL. -->
<rim:ClassificationNode code="wisBaseUrl"
  id="urn:ogc:baseUrl:service:wis">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Web Integration Service Capabilities Document (accessed
via the server's base URL)</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="wisBaseUrl"/>
  </rim:Name>
  <rim:Description>

```

```

        <rim:LocalizedString charset="UTF-8"
            value="A capabilities document for a Web Integration Service
obtained using a URL created by appending GetCapabilities request parameters to the
server's base URL."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="wfsBaseUrl"
    id="urn:ogc:baseUrl:service:wfs">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Web Feature Service Capabilities Document (accessed via
the server's base URL)</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="wfsBaseUrl"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="A capabilities document for a Web Feature Service obtained
using a URL created by appending GetCapabilities request parameters to the server's
base URL."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="wmsBaseUrl"
    id="urn:ogc:baseUrl:service:wms">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Web Map Service Capabilities Document (accessed via the
server's base URL)</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="wmsBaseUrl"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="A capabilities document for a Web Map Service obtained using
a URL created by appending GetCapabilities request parameters to the server's base
URL."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="wmtsBaseUrl"
    id="urn:ogc:baseUrl:service:wmts">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Web Map Tiling Service Capabilities Document (accessed
via the server's base URL)</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>

```

```

    <rim:LocalizedString charset="UTF-8" value="wmtsBaseUrl"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="A capabilities document for a Web Map Tiling Service obtained
using a URL created by appending GetCapabilities request parameters to the server's
base URL."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="wcsBaseUrl"
  id="urn:ogc:baseUrl:service:wcs">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Web Coverage Service Capabilities Document (accessed via
the server's base URL)</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="wcsBaseUrl"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="A capabilities document for a Web Coverage Service obtained
using a URL created by appending GetCapabilities request parameters to the server's
base URL."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="wpsBaseUrl"
  id="urn:ogc:baseUrl:service:wps">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Web Processing Service Capabilities Document (accessed
via the server's base URL)</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="wpsBaseUrl"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="A capabilities document for a Web Processing Service obtained
using a URL created by appending GetCapabilities request parameters to the server's
base URL."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="cswBaseUrl"
  id="urn:ogc:baseUrl:service:csw">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Catalogue Service Web Capabilities Document (accessed via
the server's base URL)</rim:Value>

```

```

        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="cswBaseUrl"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="A capabilities document for a Catalogue Service Web obtained
using a URL created by appending GetCapabilities request parameters to the server's
base URL."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="csw-ebRIMBaseUrl"
    id="urn:ogc:baseUrl:service:csw-ebRIM">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Web Registry Service Capabilities Document (accessed via
the server's base URL)</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="csw-ebRIMBaseUrl"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="A capabilities document for a Web Registry Service obtained
using a URL created by appending GetCapabilities request parameters to the server's
base URL."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="gssBaseUrl"
    id="urn:ogc:baseUrl:service:gss">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Web GeoCollaboration Service Capabilities Document
(accessed via the server's base URL)</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="gssBaseUrl"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="A capabilities document for a Web GeoCollaboration Service
(formerly GeoSynchronization Service) obtained using a URL created by appending
GetCapabilities request parameters to the server's base URL."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="fgdc"
    id="urn:ogc:documentType:metadata:fgdc">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">

```

```

        <rim:ValueList>
            <rim:Value>Federal Geographic Data Committee's Content Standard for
Digital Geospatial Metadata</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="fgdc"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="The Content Standard for Digital Geospatial Metadata (CSDGM),
Vers. 2 (FGDC-STD-001-1998) is the current US Federal Metadata standard. The FGDC
originally adopted the CSDGM in 1994 and revised it in 1998. According to Executive
Order 12096, all Federal agencies are ordered to use this standard to document
geospatial data created as of January 1995. The standard is often referred to as the
'FGDC Metadata Standard' and has been implemented beyond the federal level with State
and local governments adopting the metadata standard as well."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="anzlic"
    id="urn:ogc:documentType:metadata:anzlic">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>ANZLIC Metadata Profile</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="anzlic"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="This document provides a formal description of the ANZLIC
Metadata Profile which is based on AS/NZS ISO 19115:2005, Geographic information --
Metadata. Changes introduced by ISO 19115:2003/Cor.1:2006, Geographic information --
Metadata -- Technical Corrigendum 1 have also been considered. Implementation of the
Profile is based on ISO/TS 19139:2007, Geographic information -- Metadata -- XML
schema implementation."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="iso19115"
    id="urn:ogc:documentType:metadata:iso19115">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Geographic information -- Metadata -- XML schema
implementation</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="iso19115"/>
    </rim:Name>
    <rim:Description>

```

```

        <rim:LocalizedString charset="UTF-8"
            value="ISO/TS 19139:2007 defines Geographic MetaData XML (gmd)
encoding, an XML Schema implementation derived from ISO 19115."/>
        </rim:Description>
    </rim:ClassificationNode>
    <rim:ClassificationNode code="iso19119"
        id="urn:ogc:documentType:metadata:iso19119">
        <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
            <rim:ValueList>
                <rim:Value>Geographic information â€” Services</rim:Value>
            </rim:ValueList>
        </rim:Slot>
        <rim:Name>
            <rim:LocalizedString charset="UTF-8" value="iso19119"/>
        </rim:Name>
        <rim:Description>
            <rim:LocalizedString charset="UTF-8"
                value="This International Standard provides a framework for platform
neutral and platform specific specification of services that can enable users to
access, process and manage geographic data from a variety of sources, potentially for
various distributed computing platforms."/>
            </rim:Description>
        </rim:ClassificationNode>
        <rim:ClassificationNode code="iso19139"
            id="urn:ogc:documentType:metadata:iso19139">
            <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
                <rim:ValueList>
                    <rim:Value>Geographic information -- Metadata -- XML schema
implementation</rim:Value>
                </rim:ValueList>
            </rim:Slot>
            <rim:Name>
                <rim:LocalizedString charset="UTF-8" value="iso19139"/>
            </rim:Name>
            <rim:Description>
                <rim:LocalizedString charset="UTF-8"
                    value="ISO/TS 19139:2007 defines Geographic MetaData XML (gmd)
encoding, an XML Schema implementation derived from ISO 19115."/>
                </rim:Description>
            </rim:ClassificationNode>
            <rim:ClassificationNode code="ebRIM-ClassificationScheme"
                id="urn:ogc:documentType:ebxml-regrep:classificationScheme">
            <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
                <rim:ValueList>
                    <rim:Value>OASIS ebRIM Classification Scheme XML
Document</rim:Value>
                </rim:ValueList>
            </rim:Slot>
            <rim:Name>
                <rim:LocalizedString charset="UTF-8"
                    value="ebRIM-ClassificationScheme"/>
            </rim:Name>
        </rim:ClassificationNode>
    </rim:ClassificationNode>

```

```

</rim:Name>
<rim:Description>
  <rim:LocalizedString charset="UTF-8"
    value="ClassificationScheme XML documents instances describe a
structured way to classify or categorize registry objects. The structure of the
classification scheme may be defined internal or external to the registry, resulting
in a distinction between internal and external classification schemes. A very common
example of a classification scheme in science is the &quot;Classification of living
things&quot; where living things are categorized in a tree like structure. Another
example is the Dewey Decimal system used in libraries to categorize books and other
publications."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="symbologyEncoding"
  id="urn:ogc:documentType:symbologyEncoding">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>OGC Symbology Encoding Document</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="symbologyEncoding"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="*** Place holder ***"/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="xmlSchema"
  id="urn:ogc:documentType:schema:xml">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>W3C XML Schema Document</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="xmlSchema"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The XML Schema Definition Language, which offers facilities
for describing the structure and constraining the contents of XML documents, including
those which exploit the XML Namespace facility. The schema language, which is itself
represented in an XML vocabulary and uses namespaces, substantially reconstructs and
considerably extends the capabilities found in XML document type definitions
(DTDs)."/>
  </rim:Description>
</rim:ClassificationNode>
</rim:ClassificationScheme>
</csw:Insert>
<!-- ===== -->

```

```

<!-- OGC Slot Names -->
<!-- ===== -->
<csw:Insert>
  <rim:ClassificationScheme
    id="urn:ogc:ebxml-regrep:classificationScheme:SlotNames"
    isInternal="true"
    nodeType="urn:oasis:names:tc:ebxml-regrep:NodeType:UniqueCode">
    <rim:Name>
      <rim:LocalizedString charset="UTF-8" value="SlotName"/>
    </rim:Name>
    <rim:Description>
      <rim:LocalizedString charset="UTF-8"
        value="This is the canonical ClassificationScheme for slot names used
by the OGC CSW-ebRIM catalogue."/>
    </rim:Description>
    <rim:ClassificationNode code="accessConstraints"
      id="urn:ogc:ebxml-regrep:slotName:accessConstraints">
      <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
          <rim:Value>Access Constraints</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="accessConstraints"/>
      </rim:Name>
      <rim:Description>
        <rim:LocalizedString charset="UTF-8"
          value="The name of a slot used to describe any constraints (e.g.
temporal, volumetric) associated with accessing or using an object represented as a
catalogue entry."/>
      </rim:Description>
    </rim:ClassificationNode>
    <rim:ClassificationNode code="baseTableName"
      id="urn:ogc:ebxml-regrep:slotName:baseTableName">
      <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
          <rim:Value>Base Table Name</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="baseTableName"/>
      </rim:Name>
      <rim:Description>
        <rim:LocalizedString charset="UTF-8"
          value="The name of a slot used to store the name of a database table
used to store an object represented as an entry in the catalogue. For example, the
spatial table name used to store data for a WMS layer or WFS feature type."/>
      </rim:Description>
    </rim:ClassificationNode>
    <rim:ClassificationNode code="baseUrl"
      id="urn:ogc:ebxml-regrep:slotName:baseUrl">

```



```

<rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
  <rim:ValueList>
    <rim:Value>Base URL</rim:Value>
  </rim:ValueList>
</rim:Slot>
<rim:Name>
  <rim:LocalizedString charset="UTF-8" value="baseUrl"/>
</rim:Name>
<rim:Description>
  <rim:LocalizedString charset="UTF-8"
    value="A slot used to store the base GetCapabilities URL of an
object represented as an entry in the catalogue. The base GetCapabilities URL is the
URL to which GetCapabilities parameters are appended to form a GetCapabilities
request."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="crs"
  id="urn:ogc:ebxml-regrep:slotName:crs">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Coordinate Reference System</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="crs"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The name of a slot used to store the name or code of a
coordinate reference system associated with a catalogue entry."/>
    </rim:Description>
  </rim:ClassificationNode>
<rim:ClassificationNode code="fees"
  id="urn:ogc:ebxml-regrep:slotName:fees">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Fees</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="fees"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The name of a slot used to describe any fees associated with
accessing or using an object represented as a catalogue entry."/>
    </rim:Description>
  </rim:ClassificationNode>
<rim:ClassificationNode code="hasRepositoryItem"
  id="urn:ogc:ebxml-regrep:slotName:hasRepositoryItem">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">

```

```

        <rim:ValueList>
            <rim:Value>Has Repository Item</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="hasRepositoryItem"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="The name of a slot used to store a boolean that indicates
whether a catalogue entry has an associated item in the catalogue's repository."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="keyword"
    id="urn:ogc:ebxml-regrep:slotName:keyword">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Keyword</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="keyword"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="The name of a slot used to store a list of keywords
associated with a catalogue entry."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="layerLimit"
    id="urn:ogc:ebxml-regrep:slotName:layerLimit">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Layer Limit</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="layerLimit"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="The name of a slot used to store an integer that represents
the maximum number of layers that a client can specify in a WMS request."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="legendUrl"
    id="urn:ogc:ebxml-regrep:slotName:legendUrl">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Legend Graphic URL</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="legendUrl"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="The name of a slot used to store a string that represents
the URL of a legend graphic."/>
    </rim:Description>
</rim:ClassificationNode>

```

```

</rim:Slot>
<rim:Name>
  <rim:LocalizedString charset="UTF-8" value="legendUrl"/>
</rim:Name>
<rim:Description>
  <rim:LocalizedString charset="UTF-8"
    value="The name of a slot used to store the URL of legend graphic
associated with an entry in the catalogue that represents a WMS layer."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="maxHeight"
  id="urn:ogc:ebxml-regrep:slotName:maxHeight">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Maximum Map Height</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="maxHeight"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The name of a slot used to store, in pixels, the maximum map
height that a WMS service generates."/>
    </rim:Description>
  </rim:ClassificationNode>
  <rim:ClassificationNode code="maxWidth"
    id="urn:ogc:ebxml-regrep:slotName:maxWidth">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
      <rim:ValueList>
        <rim:Value>Maximum Map Width</rim:Value>
      </rim:ValueList>
    </rim:Slot>
    <rim:Name>
      <rim:LocalizedString charset="UTF-8" value="maxWidth"/>
    </rim:Name>
    <rim:Description>
      <rim:LocalizedString charset="UTF-8"
        value="The name of a slot used to store, in pixels, the maximum map
width that a WMS service generates."/>
      </rim:Description>
    </rim:ClassificationNode>
    <rim:ClassificationNode code="onlineResource"
      id="urn:ogc:ebxml-regrep:slotName:onlineResource">
      <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
          <rim:Value>Online Resource</rim:Value>
        </rim:ValueList>
      </rim:Slot>
      <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="onlineResource"/>

```

```

</rim:Name>
<rim:Description>
  <rim:LocalizedString charset="UTF-8"
    value="The name of a slot used to store the URL that points to
additional information about the object represented by a catalogue entry."/>
</rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="outputFormat"
  id="urn:ogc:ebxml-regrep:slotName:outputFormat">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Output Format</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="outputFormat"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The name of a slot used to store a list of formats that may
be used to represent the object represented by a catalogue entry."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="queryable"
  id="urn:ogc:ebxml-regrep:slotName:queryable">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Queryable</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="queryable"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The name of a slot used to store a boolean value indicating
whether the object represented by a catalogue entry is queryable or not (usually used
to indicate whether a WMS layer is queryable or not)."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="resourceType"
  id="urn:ogc:ebxml-regrep:slotName:resourceType">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Resource Type</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="resourceType"/>
  </rim:Name>
  <rim:Description>

```

```

        <rim:LocalizedString charset="UTF-8"
            value="The name of a slot used to store the name or code of a
standardized metadata type for the object represented by a catalogue entry."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="reverseAssociationTitle"
    id="urn:ogc:ebxml-regrep:slotName:reverseAssociationTitle">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Reverse Association Title</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8"
            value="reverseAssociationTitle"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="The name of a slot used to store the reverse title for an
association. For example, the reverse title of 'Serves' would be 'Served By'."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="scheme"
    id="urn:ogc:ebxml-regrep:slotName:scheme">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Scheme</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="scheme"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="The name of a slot used to store the name of a scheme
associated with a catalogue entry."/>
    </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="spatialPropertyName"
    id="urn:ogc:ebxml-regrep:slotName:spatialPropertyName">
    <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
        <rim:ValueList>
            <rim:Value>Spatial Property Name</rim:Value>
        </rim:ValueList>
    </rim:Slot>
    <rim:Name>
        <rim:LocalizedString charset="UTF-8" value="spatialPropertyName"/>
    </rim:Name>
    <rim:Description>
        <rim:LocalizedString charset="UTF-8"
            value="The name of a slot used to store the name of the spatial

```

```

property of a catalogue entry (usually representing a feature type)."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="supportedOperations"
  id="urn:ogc:ebxml-regrep:slotName:supportedOperations">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Supported Operations</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="supportedOperations"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The name of a slot used to store a list of operations
supported by a catalogue entry (usually representing a service)."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="title"
  id="urn:ogc:ebxml-regrep:slotName:title">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Title</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="title"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The name of a slot used to store a human-readable title
associated with a catalogue entry."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="where"
  id="urn:ogc:ebxml-regrep:slotName:where">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Where</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="where"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The name of a slot used to store a location or geometry
associated with a catalogue entry"/>
  </rim:Description>
</rim:ClassificationNode>

```

```

<rim:ClassificationNode code="position"
  id="urn:ogc:ebxml-regrep:slotName:position">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>PositionName</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="position"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The name of a slot used to store the role or position of a
responsible person."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="role"
  id="urn:ogc:ebxml-regrep:slotName:role">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Role</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="role"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The name of a slot used to store the function performed by
the responsible party. Possible values of this Role shall include the values and the
meanings listed in Subclause B.5.5 of ISO 19115:2003."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="url"
  id="urn:ogc:ebxml-regrep:slotName:url">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>URL</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="url"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The name of a slot used to store a url associated with the a
catalogue entry."/>
  </rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="hoursOfService"
  id="urn:ogc:ebxml-regrep:slotName:hoursOfService">

```

```

<rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
  <rim:ValueList>
    <rim:Value>Hours of Service</rim:Value>
  </rim:ValueList>
</rim:Slot>
<rim:Name>
  <rim:LocalizedString charset="UTF-8" value="hoursOfService"/>
</rim:Name>
<rim:Description>
  <rim:LocalizedString charset="UTF-8"
    value="The name of a slot used to store the hours of service
associated with the a catalogue entry."/>
</rim:Description>
</rim:ClassificationNode>
<rim:ClassificationNode code="contactInstructions"
  id="urn:ogc:ebxml-regrep:slotName:contactInstructions">
  <rim:Slot name="urn:ogc:ebxml-regrep:slotName:title">
    <rim:ValueList>
      <rim:Value>Contact Instructions</rim:Value>
    </rim:ValueList>
  </rim:Slot>
  <rim:Name>
    <rim:LocalizedString charset="UTF-8" value="contactInstructions"/>
  </rim:Name>
  <rim:Description>
    <rim:LocalizedString charset="UTF-8"
      value="The name of a slot used to store any special contact
instructions associated with the a catalogue entry."/>
  </rim:Description>
</rim:ClassificationNode>
</rim:ClassificationScheme>
</csw:Insert>
</csw:Transaction>

```

- **Consolidated list of identifiers**

For the sake of convenience and clarity, the following consolidated list of identifiers has been compiled from this engineering report:

- urn:ogc:ebxml-regrep:classificationScheme:AssociationTypes
 - urn:ogc:associationType:renders
 - urn:ogc:associationType:tiledBy
 - urn:ogc:associationType:parentOf
 - urn:ogc:associationType:cascades
 - urn:ogc:associationType:createdBy
 - urn:ogc:associationType:symbolizes
 - urn:ogc:associationType:operatesOn

- urn:ogc:associationType:describedBy
- urn:ogc:ebxml-regrep:classificationScheme:ServiceTypes
 - urn:ogc:serviceType:wis
 - urn:ogc:serviceType:wfs
 - urn:ogc:serviceType:wos
 - urn:ogc:serviceType:wms
 - urn:ogc:serviceType:wmts
 - urn:ogc:serviceType:wps
 - urn:ogc:serviceType:wcs
 - urn:ogc:serviceType:csw
 - urn:ogc:serviceType:csw-ebRIM
 - urn:ogc:serviceType:gss
- urn:ogc:ebxml-regrep:classificationScheme:ContentTypes
 - urn:ogc:contentType:wfs:featureType
 - urn:ogc:contentType:wos:object
 - urn:ogc:contentType:wms:theme
 - urn:ogc:contentType:wms:layer
 - urn:ogc:contentType:wmts:theme
 - urn:ogc:contentType:wmts:layer
 - urn:ogc:contentType:wcs:theme
 - urn:ogc:contentType:wcs:coverage
 - urn:ogc:contentType:wps:process
 - urn:ogc:resourceType:unknown
- urn:ogc:ebxml-regrep:classificationScheme:HarvestResourceTypes
 - urn:ogc:documentType:capabilities:wis
 - urn:ogc:documentType:capabilities:wfs
 - urn:ogc:documentType:capabilities:wms
 - urn:ogc:documentType:capabilities:wmts
 - urn:ogc:documentType:capabilities:wcs
 - urn:ogc:documentType:capabilities:wps
 - urn:ogc:documentType:capabilities:csw
 - urn:ogc:documentType:capabilities:csw-ebRIM
 - urn:ogc:documentType:capabilities:gss
 - urn:ogc:baseUrl:service:wis

- urn:ogc:baseUrl:service:wfs
- urn:ogc:baseUrl:service:wms
- urn:ogc:baseUrl:service:wmts
- urn:ogc:baseUrl:service:wcs
- urn:ogc:baseUrl:service:wps
- urn:ogc:baseUrl:service:csw
- urn:ogc:baseUrl:service:csw-ebRIM
- urn:ogc:baseUrl:service:gss
- urn:ogc:documentType:metadata:fgdc
- urn:ogc:documentType:metadata:anzlic
- urn:ogc:documentType:metadata:iso19115
- urn:ogc:documentType:metadata:iso19119
- urn:ogc:documentType:metadata:iso19139
- urn:ogc:documentType:ebxml-regrep:classificationScheme
- urn:ogc:documentType:symbologyEncoding
- urn:ogc:documentType:schema:xml
- urn:ogc:ebxml-regrep:classificationScheme:SlotNames
 - urn:ogc:ebxml-regrep:slotName:accessConstraints
 - urn:ogc:ebxml-regrep:slotName:baseTableName
 - urn:ogc:ebxml-regrep:slotName:baseUrl
 - urn:ogc:ebxml-regrep:slotName:crs
 - urn:ogc:ebxml-regrep:slotName:fees
 - urn:ogc:ebxml-regrep:slotName:hasRepositoryItem
 - urn:ogc:ebxml-regrep:slotName:keyword
 - urn:ogc:ebxml-regrep:slotName:layerLimit
 - urn:ogc:ebxml-regrep:slotName:legendUrl
 - urn:ogc:ebxml-regrep:slotName:maxHeight
 - urn:ogc:ebxml-regrep:slotName:maxWidth
 - urn:ogc:ebxml-regrep:slotName:onlineResource
 - urn:ogc:ebxml-regrep:slotName:outputFormat
 - urn:ogc:ebxml-regrep:slotName:queryable
 - urn:ogc:ebxml-regrep:slotName:resourceType
 - urn:ogc:ebxml-regrep:slotName:reverseAssociationTitle
 - urn:ogc:ebxml-regrep:slotName:scheme

- urn:ogc:ebxml-regrep:slotName:spatialPropertyName
- urn:ogc:ebxml-regrep:slotName:supportedOperations
- urn:ogc:ebxml-regrep:slotName:title
- urn:ogc:ebxml-regrep:slotName:where
- urn:ogc:ebxml-regrep:slotName:position
- urn:ogc:ebxml-regrep:slotName:role
- urn:ogc:ebxml-regrep:slotName:url
- urn:ogc:ebxml-regrep:slotName:hoursOfService
- urn:ogc:ebxml-regrep:slotName:contactInstructions

Appendix C: Review comments from TB12 participants

- **Review comment 1 (C. Heazel)**

On the GetCapabilities() operation – I don't think it is necessary to modify this operation. It is sufficient to have a standard service metadata document with two defined operations; getResourceById() and getAssociations(). GetResourceById() is required by OWS Common. GetAssociations() can do everything you are trying to do in the service metadata. Furthermore, it will avoid further balkanization of the service metadata.

- **Response to comment 1**

The WIS does not modify the GetCapabilities() operation in any way. It is the standard GetCapabilities operation that every other OGC web service uses. The issue is with the response. Currently, OWS common requires that 2 operations be defined in the operations metadata section of a capabilities document. A change request was posted a couple of years ago (OGC 14-026) to remove this restriction. This is a backward compatible change would allow a WIS to be implemented without having to implement any other operation (i.e. simple implementation) and still let existing OGC services operate as they always have.

The GetResourceById operation, as far as I know, is not mandatory for all services. Rather OWS common says: "This typical operation is specified as a base for profiling by specific OWS specifications.". So I believe that OWS common says that if your service includes a GetResourceById operation then it should be derived from the base defined therein.

The WIS handles the 2-operation minimum requirement by defining a dummy no-op. However, the idea of implementing one or both of GetResourceById and/or GetAssociations() is not a bad one. It would impose an additional burden on the implementation of a WIS, since the WIS would need to cascade those requests down to the services that it is aggregating but that extra complexity could be handled by adding an additional, optional, conformance class to the WIS. I have added this to the future work section.

- **Review comment 2 (C. Heazel)**

Consider the value of making a mandatory getResourceById() operation which is defined the same for all OGC services. Since all of the OGC resources are defined by a URL and resource identifier, then the getResourceById URL and the identifier are all we need to retrieve that resource. We can consider this in a CR to OWS Common.

- **Response to comment 2**

Agreed (see: http://ogc.standardstracker.org/show_request.cgi?id=412)

- **Review comment 3 (C. Heazel)**

On identifiers – I suggest using URNs rather than URLs. The URLs imply that the services are only accessible through HTTP and that the identifiers resolve to the identified resource. This will usually

not be the case. Using a URN for the identifier makes a clear distinction between the resource identifier and the URL for accessing that resource.

- **Response to comment 3**

I am not opposed to this but it was my understanding the the OGC directives recommend HTTP-URI identifiers rather than URN identifiers. According to Carl Reid:

"Proposed initially by Simon Cox and presented in early 2010 (https://portal.opengeospatial.org/files/?artifact_id=39331&version=1). The policy was discussed in the OAB and then the broader OGC TC and approved at the June 2010 meetings. The decision is documented in the OGC Policy Directives (06-135r11). The specific wording is:

At the June 2010 meetings, the OGC Members approved a change related to the use of URIs in OGC standards documents. There were two parts to the motion:

1. OGC Technical and Planning Committees directs the OGC-Naming Authority that all new OGC identifiers issued for persistent public OGC resources shall be http URIs, instead of URNs;
2. New standards and new major versions of existing standards shall use http URIs for persistent public OGC resources to replace OGC URN identifiers defined in previous standards and versions, unless the OGC-NA approves an exception.

At the time, the use of http URIs was (a) seen as a way to resolve deployment challenges and (b) provides an opportunity for easier engagement with broader communities. So OGC was asked to consider taking the next step, and mandate the use of http URIs for persistent identifiers in OGC standards.

However, that has not entirely happened. There are still OGC http uri deployment challenges. Further, URNs are still very much in use. The IESG processes a number of new URN proposals each year. Many (most) internet standards use urn's and any number of new revisions or RFC still use urn's. Other standards organizations are also still using urn's.

And of course there was the backwards compatibility issue."

In any case, for all non-CSW eBRIM-specific identifiers in this ER, I have defined both a URN form and an HTTP-URI form — just to cover all the bases!

- **Review comment 4 (C. Heazel)**

Nice – this goal is more likely to be achieved if the `getAssociations()` operation can be implemented without any changes (outside of the additional operation metadata) to the existing service.

Consider a use case from the tactical sensor world. These sensors are not connected to the Internet. Rather they form and re-form mesh networks based on who is currently in range (the sensor can move). Usually one of the participants in the mesh also has up-link capability. `GetAssociations()` can tell me which sensors are currently accessible through the up-link. It could also support a temporal filter so I can discover which sensors have been associated over a period of time. This is useful if I am looking for archived sensor readings. This really calls for a JSON-LD output format.

- **Response to comment 4**

I am not exactly sure how you could implement a `GetAssociations()` operation without modifying the service. The primary reason for the existence of the operation is to allow services to advertise the associations they are usually aware of internally. In static, or pseudo-static case, I suppose you could implement the operation on top of a static `Associations` document without necessarily modifying the service to which the associations pertain. But I don't think that is the general case.

Just like the use case you describe, participants in the mesh network know who they have been and are associated with. The idea is to make this internal or private information available through a public API. In the current testbed a primary use case was to have a catalogue harvest this association information to support rich discovery. However, once exposed through a public API the universe of possibilities for this kind of knowledge is quite large as illustrated by your use case.

I quite like the use case you describe and I am pleased to say that it could be accommodated with the current proposal since associations can have properties and one of those properties could be a time stamp.

I agree that JSON-LD would be useful output format and have added this to the future work section.

Appendix D: Revision History

Table 24. Revision History

Date	Editor	Descriptions
30-OCT-2016	P. Vretanos	Incorporate review comments from OGC reviewers and final edits prior to posting on pending
13-COT-2015	P. Vretanos	Add UML diagrams
16-SEP-2016	P. Vretanos	Add light-weight query capability to GetAssociations
09-SEP-2016	P. Vretanos	Resolve issue of duplicate identifiers
15-JUN-2016	P. Vretanos	Allows associations to have dynamic properties
18-MAY-2016	P. Vretanos	First cut as schemas for wis, associations, etc.
13-APR-2016	P. Vretanos	initial version