Open Geospatial Consortium

Approval Date: 2013-1-18

Posted Date: 2013-03-25

Publication Date: 2013-06-18

Reference number of this document: OGC 12-144

Reference URL for this document: http://www.opengis.net/doctype/per/ows9-aviation-registry

Category: Engineering Report

Editor: David Burggraf

OGC® OWS-9 Architecture - Registry Engineering Report

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.

Document type: OGC® Engineering Report
Document subtype: NA
Document stage: Approved for public release
Document language: English

Abstract

This OGC® Engineering Report provides guidelines for the harvest, registration and retrieval of aviation resources from an OGC web catalogue/registry service (OGC CSW-ebRIM), with particular emphasis on ISO metadata resources. Alternatives for selective and efficient retrieval of such resources are also described along with lessons learned. The OGC CSW-ebRIM registry interface is evaluated against SESAR registry requirements, documented as a gap analysis, to assess whether there are any obstacles to implementing SESAR registry with an OGC CSW-ebRIM interface.

Keywords

ogcdoc, ogc document, ows9, ows-9, registry, architecture, csw, ebrim, aviation

What is OGC Web Services 9 (OWS-9)?

OWS-9 builds on the outcomes of prior OGC interoperability initiatives and is organized around the following threads:

- **Aviation**: Develop and demonstrate the use of the Aeronautical Information Exchange Model (AIXM) and the Weather Exchange Model (WXXM) in an OGC Web Services environment, focusing on support for several Single European Sky ATM Research (SESAR) project requirements as well as FAA (US Federal Aviation Administration) Aeronautical Information Management (AIM) and Aircraft Access to SWIM (System Wide Information Management) (AAtS) requirements.

- **Cross-Community Interoperability (CCI)**: Build on the CCI work accomplished in OWS–8 by increasing interoperability within communities sharing geospatial data, focusing on semantic mediation, query results delivery, data provenance and quality and Single Point of Entry Global Gazetteer.

- **Security and Services Interoperability (SSI)**: Investigate 5 main activities: Security Management, OGC Geography Markup Language (GML) Encoding Standard Application Schema UGAS (UML to GML Application Schema) Updates, Web Services Façade, Reference Architecture Profiling, and Bulk Data Transfer.

- **OWS Innovations**: Explore topics that represent either new areas of work for the Consortium (such as GPS and Mobile Applications), a desire for new approaches to existing technologies to solve new challenges (such as the OGC Web Coverage Service (WCS) work), or some combination of the two.

- **Compliance & Interoperability Testing & Evaluation (CITE)**: Develop a suite of compliance test scripts for testing and validation of products with interfaces implementing the following OGC standards: Web Map Service (WMS) 1.3 Interface Standard, Web Feature Service (WFS) 2.0 Interface Standard, Geography Markup Language (GML) 3.2.1 Encoding Standard, OWS Context 1.0 (candidate encoding standard), Sensor Web Enablement (SWE) standards, Web Coverage Service for Earth Observation (WCS-EO) 1.0 Interface Standard, and TEAM (Test, Evaluation, And Measurement) Engine Capabilities.

The OWS-9 sponsors are: AGC (Army Geospatial Center, US Army Corps of Engineers), CREAF-GeoViQua-EC, EUROCONTROL, FAA (US Federal Aviation Administration), GeoConnections - Natural Resources Canada, Lockheed Martin Corporation, NASA (US National Aeronautics and Space Administration), NGA (US National Geospatial-Intelligence Agency), USGS (US Geological Survey), UK DSTL (UK MoD Defence Science and Technology Laboratory).

License Agreement

Contents

Page

OGC® OWS-9 Architecture - Registry Engineering Report

1 Introduction

1.1 Scope

This OGC® Engineering Report provides guidelines for the harvest, registration and retrieval of aviation resources from an OGC web catalogue/registry service (OGC CSW-ebRIM), with particular emphasis on ISO metadata resources. Alternatives for selective and efficient retrieval of such resources are also described along with lessons learned. The OGC CSW-ebRIM registry interface is evaluated against SESAR registry requirements, documented as a gap analysis, to assess whether there are any obstacles to implementing SESAR registry with an OGC CSW-ebRIM interface.

1.2 Document contributor contact points

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

Name	Organization
David Burggraf	Galdos Systems Inc.

1.3 Revision history

Date	Release	Editor	Primary clauses modified	Description
2012-07-16	0.1	Leif Stainsby	6, 7.2.1.1	
2012-11-21	0.2	David Burggraf	All	

1.4 Future work

The following list of sub-clauses summarizes topics/issues that should be considered for future work related to registries.

1.4.1 Standardize WSDL documents for OWS

In OWS-9 Aviation, ISO 19119/19139 Service Metadata was automatically generated from OWS Capabilities Documents to improve OWS interoperability of service discovery applications. There were several non-OWS component service implementations (and some OWS service implementations) in OWS-9 that only published service descriptions as WSDL documents. However, it was found that the WSDL documents provided:

- Did not follow any common implementation guidelines
- Ranged widely with respect to organization and level of completeness
- Tended to have lighter service descriptions compared to Capabilities and ISO Service metadata
- Most did not pass schema validation.

Therefore, we it is currently not possible to properly automate the creation of ISO Service Metadata from WSDL documents as was done for OWS Capabilities in OWS-9. The proposed future work items shall improve the interoperability of WSDL-capable applications in OWS environments (e.g. SESAR Registry) by:

1. Developing an OWS/ISO profile of the WSDL standard.
2. Developing best practices for the creation/transformation of WSDL documents in/to the OWS/ISO profile of WSDL.
3. The automatic generation of OWS/ISO profiled WSDL directly from OWS Capabilities documents and ISO 19139 Service Metadata.

1.4.2 Harmonize/Bridge Profiles of CSW that support ISO Metadata

Perform a Gap Analysis and Design and Implement an Interface bridge between the two profiles of CSW that support the ISO 19115/19119/19139 Metadata information model, namely CSW-ISO and CSW-ebRIM (loaded with the CIM Registry Package). Both of these OGC CSW profile standards already support the common CSW GetRecord interface, OGC Filter Encoding, and the ISO Metadata information model. The outcomes of this task shall provide:

1. Harmonized Implementation/Deployment Guidelines (OGC Best Practice) to enable interoperability between the CSW ISO and ebRIM-CIM profile implementations.

2. A registry/catalogue Client Component to demonstrate that common requests and transactions of the harmonized Service Interface Standard work with both CSW-ISO and CSW-ebRIM Service implementations.

1.5 Forward

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.

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.

[1] SWIM Registry Concept of Operations V1 (08.03.02.D03)
[2] AIRM-ISRM Requirements (08.03.02.D04)
[3] SESAR Demonstrator Report (08.03.02.D08)
[4] Operational Service and Environment Definition (OSED Step 1) (13.02.02.D01)
[5] CSW-ebRIM Registry Service - Part 1: ebRIM profile of CSW (OGC 07-110r3)
[6] CSW-ebRIM Registry Service - Part 2: Basic extension package (OGC 07-144r3)
[7] Cataloguing ISO Metadata (CIM) draft IS (OGC 07-038r3, v0.1.11, 2009-07-14)
[8] Requirements for Aviation Metadata (OGC 10-195)
[9] Guidance on the Aviation Metadata Profile (OGC 10-196r1)
[10] OWS-9 Metadata & Provenance Engineering Report (OGC 12-145r1)
[11] OGC CIM CR: LI_Lineage (Provenance) Components (OGC 13-013)

3 Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r3] shall apply. In addition, the following terms and definitions apply.

3.1 Association

A logical link between registry objects.

3.2 Entity
A primary concept or object within a SESAR Registry known as a registry object in CSW-ebRIM terminology.

3.3 Identity Management
Management of individual identities and their authentication, authorization, privileges, and permissions within or across system and enterprise boundaries.

3.4 Registry
In a web service context it is a service which manages web-accessible resources. It provides Publication, Discovery, and Management services for both its resource and metadata content.

3.5 Registry Object
A primary concept or metadata object within a CSW-ebRIM Registry known as an Entity in SESAR terminology.

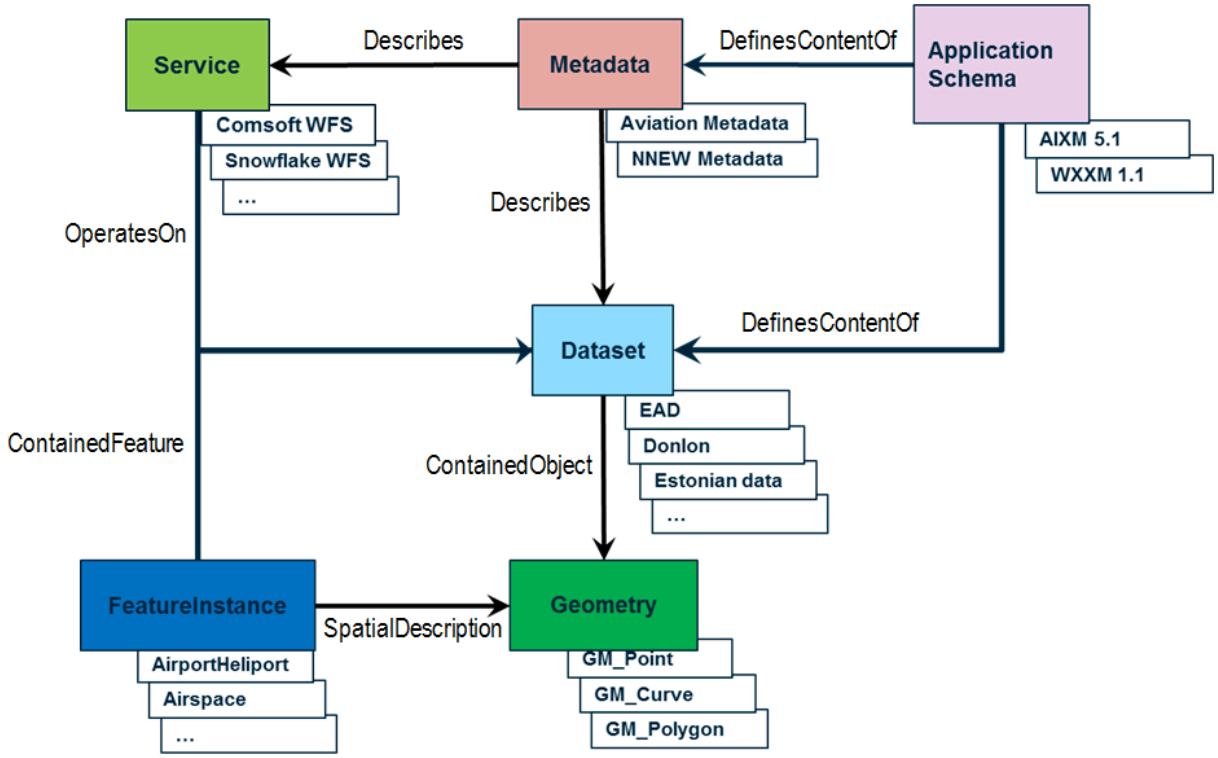
4 Conventions

4.1 Abbreviated terms

AIXM	Aeronautical Information Exchange Model
ATM	Air traffic management
ATS	Abstract Test Suite
CITE	Compliance and Interoperability Test Initiative
CSW	OGC Catalogue Service for the Web
ebRIM	OASIS ebXML Registry Information Model
ETS	Executable Test Suite
GML	Geography Markup Language
ISO	International Organization for Standardization
OGC	Open Geospatial Consortium
PDP	Policy Decision Point (typically in a XACML framework)
PEP	Policy Enforecment Point (typically in a XACML framework)
SESAR	Single European Sky ATM Research Programme
SWIM	System Wide Information Management
XACML	OASIS Extensible Access Control Markup Language
XML	Extensible Markup Language

5 Executive Summary/Overview

Aviation resources such as AIXM features, portrayal rules, schemas, and metadata for datasets and services, were harvested in the OWS-9 project and published into the Galdos INdicio web registry service (OGC CSW-ebRIM) to enable optimal discovery and retrieval of these resources. The SESAR Registry Demonstrator supported only service metadata and so the same service metadata was loaded in both registries (see Section 8.5 for further details on the SESAR Registry Demonstrator). Over one million registry objects representing the aviation resources and associations between them were loaded into the OGC registry service. The associations/relationships between the aviation resources were captured in conformance with the ISO TC211 19101-2001 Domain Reference Model applied to the aviation domain as illustrated in the following annotated diagram. The white boxes that annotate each of the shaded resource type boxes represent sample instances of resources that were deployed and managed.

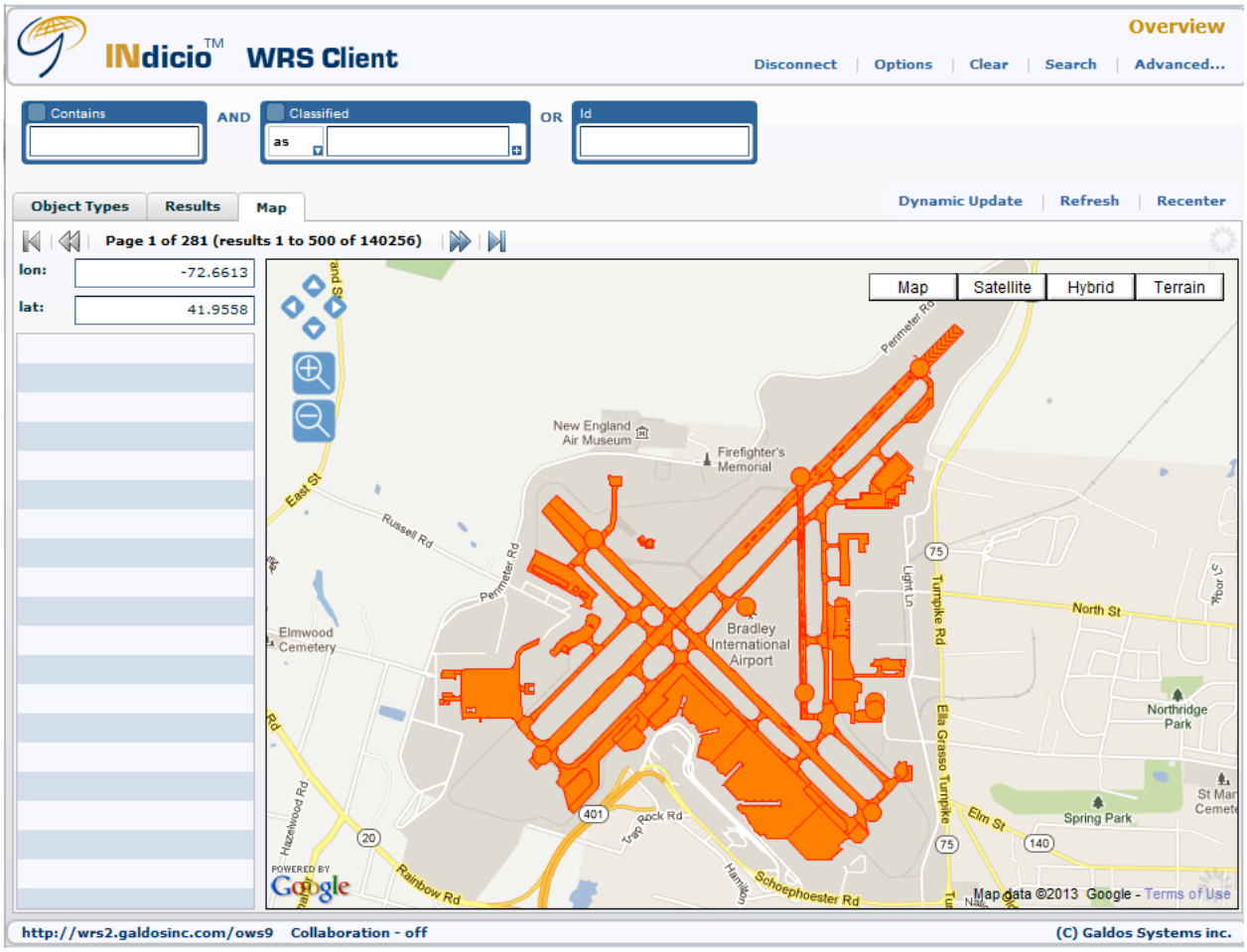


The Domain Reference Model from ISO 19101-2001 (Figure 5), adapted to the aviation domain as shown above, was also discussed in the OWS-9 Metadata & Provenance ER ([10], Section 4.3) from the perspective of modeling of aviation metadata resources. In this Registry ER, the focus is on publishing the aviation resources to registry service implementations in a standardized way to optimize discovery and retrieval.

As part of the registry task in OWS-9, the OGC Cataloguing ISO Metadata (CIM) registry model was used to publish the aviation resources as registry objects. The CIM registry model (currently a draft specification being developed by the OGC CIM SWG)

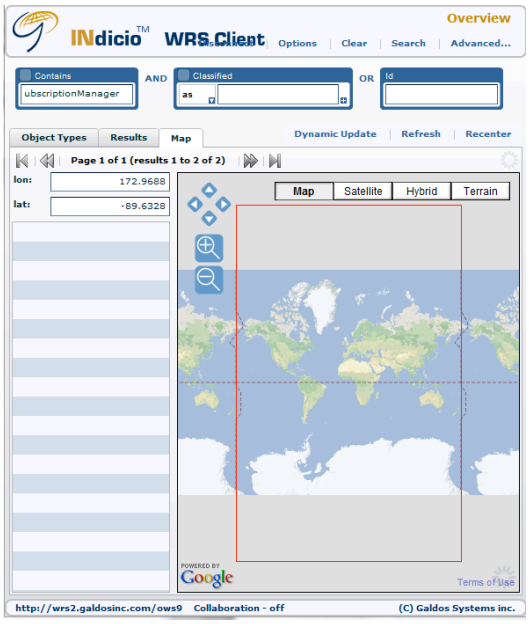
provides a standardized way to represent ISO 19115 and 19119 metadata as registry objects for the discovery of datasets and services. The loaded registry object instances were used to represent the resource types (service, dataset, metadata, application schema, feature instance, and geometry) and the association types (*operates on*, *describes*, *defines the content of*, *contained feature*, *contained object*, and *spatial representation*).

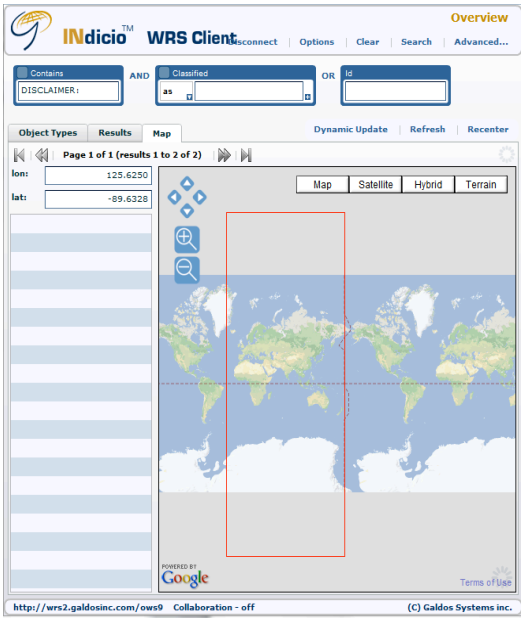
The following registry client screenshot displays the result of a registry query with a spatial filter (using OGC Filter) for metadata in a region surrounding the Bradley International Airport in Connecticut, USA. Dataset metadata from several AIXM features are returned including: the airport/heliport, runways, taxiways, aprons, etc. The geometry was harvested along with the ISO dataset metadata from each of the WFS Services to enable OGC CSW-ebRIM spatial queries. The client view screen is rendered by styling the registry's XML response to graphical form.



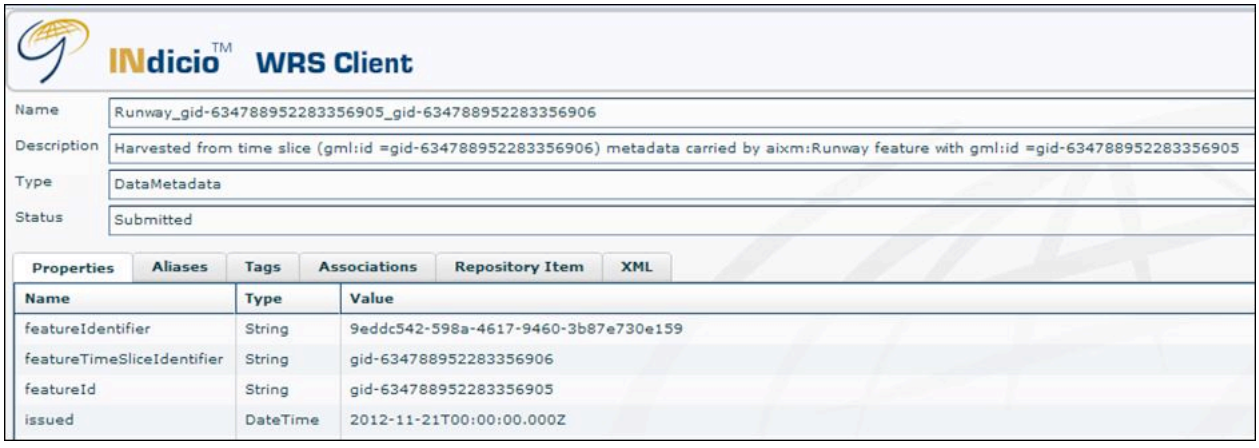
Similarly, spatial searches can be executed on service metadata and the response results can be displayed in the map view showing the bounding box extent of the service's data offering (harvested from the OWS Capabilities documents). The following screenshots from left to right show the map view results of the ifGI OGC Event Service (full global

coverage) and the Snowflake WFS (half globe coverage containing Europe/Asia), respectively:

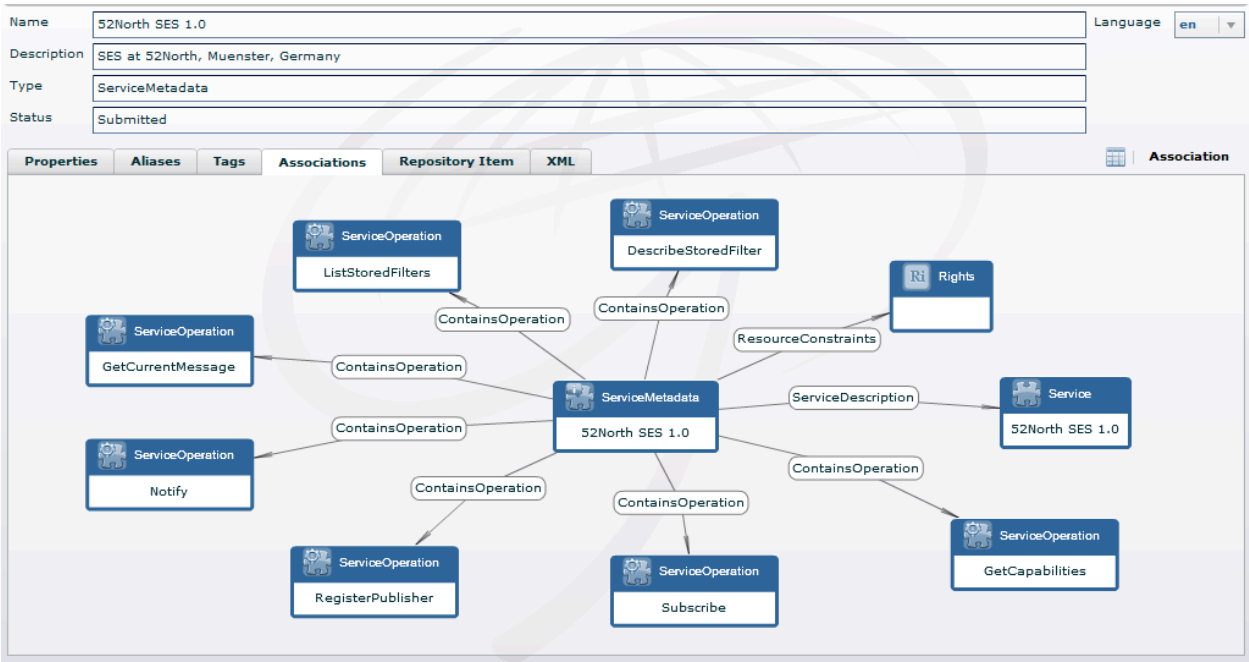




The following registry client screenshot displays the summary of an AIXM runway feature's time slice metadata. In particular, the properties shown on the following landing screen, include the contents of the 'summary' XML view ('brief', 'summary' and 'full' views can be requested from CSW-ebRIM), such as the name, description, type, life-cycle status, issue date, parent runway feature identifier info, and any other 'slot' properties of the object.



Additional information can be seen in the client by clicking on the other tabs such as 'Aliases', 'Tags', 'Associations', 'Repository Item', and 'XML'. Associations can be shown in either graph view or list view. An example screenshot showing the associations in graph view, between a service metadata record and its associated operations and constraints, in the case of the 52North Event Service (SES), is as follows.

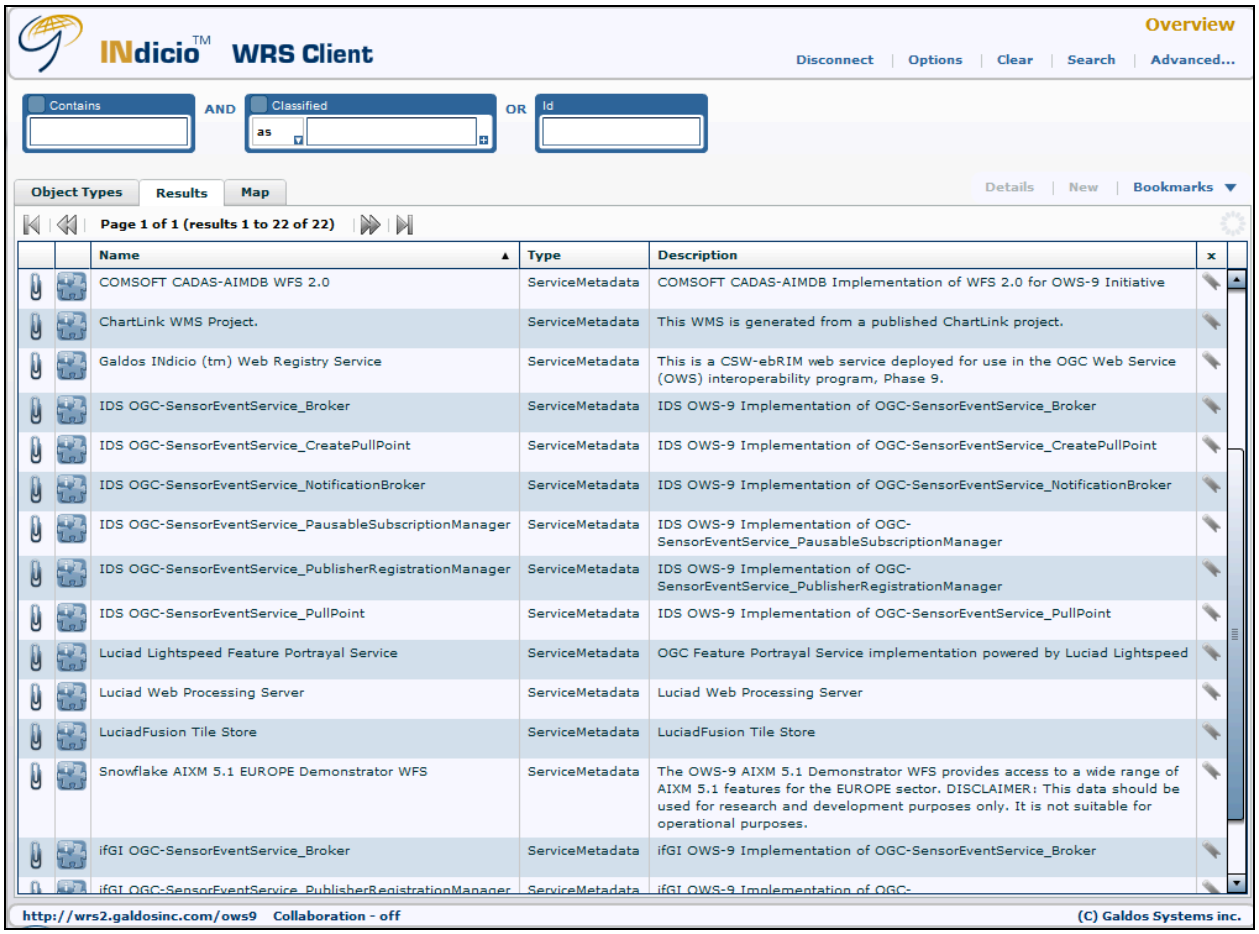


The following OWS services were harvested for service and dataset metadata by using either OWS Capabilities or WSDL documents as inputs (see 1.4.1 and 8.5 for additional details on the use of WSDL inputs).

Service Resources Harvested in OWS-9		
Service Name		Service Description
1	52North SES 1.0	SES at 52North, Muenster, Germany
2	52North WPS 3.1-SNAPSHOT	Service based on the 52North implementation of WPS 1.0.0
3	ATM-TGS Data Management Service	ATM-TGS OWS-9 Implementation of Data Management Service
4	ATM-TGS Dispatch DMS	ATM-TGS OWS-9 Implementation of Dispatch Data Management Service
5	ATM-TGS Ground DMS	ATM-TGS OWS-9 Implementation of Ground Data Management Service
6	Envitia ChartLink WMS	Envitia WMS generated from a published ChartLink project
7	COMSOFT CADAS-AIMDB WFS 2.0	COMSOFT CADAS-AIMDB Implementation of WFS 2.0 for OWS-9 Initiative
8	Galdos INdicio Aviation Web Registry Service (OGC CSW-ebRIM)	This is a CSW-ebRIM web service deployed for use in the OGC Web Service (OWS) interoperability program, Phase 9.
9	IDS OGC SES Broker	IDS OWS-9 Implementation of OGC Sensor Event Service Broker
10	IDS OGC SES Create Pull Point	IDS OWS-9 Implementation of OGC Sensor Event Service Create Pull Point
11	IDS OGC SES Notification Broker	IDS OWS-9 Implementation of OGC Sensor Event Service Notification Broker
12	IDS OGC SES Pausable Subscription Manager	IDS OWS-9 Implementation of OGC Sensor Event Service Pausable Subscription Manager

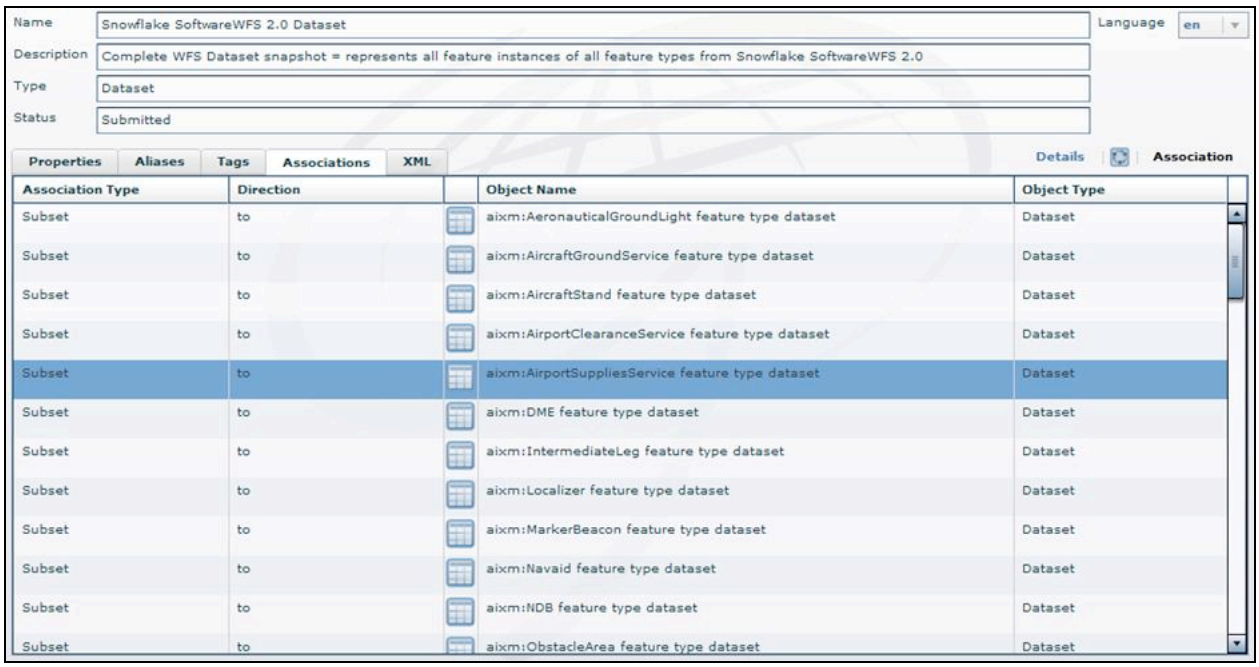
13	IDS OGC SES Publisher Registration Manager	IDS OWS-9 Implementation of OGC Sensor Event Service Publisher Registration Manager
14	IDS OGC SES Pull Point	IDS OWS-9 Implementation of OGC Sensor Event Service Pull Point
15	ifGI OGC SES Service Broker	ifGI OWS-9 Implementation of OGC Sensor Event Service Broker
16	ifGI OGC SES Publisher Registration Manager	ifGI OWS-9 Implementation of OGC Sensor Event Service Publisher Registration Manager
17	ifGI OGC SES Subscription Manager	ifGI OWS-9 Implementation of OGC Sensor Event Service Subscription Manager
18	Luciad Lightspeed FPS	OGC Feature Portrayal Service implementation powered by Luciad Lightspeed
19	Luciad WPS	Luciad Web Processing Server
20	LuciadFusion Tile Store	LuciadFusion Tile Store
21	Snowflake AIXM 5.1 EUROPE Demonstrator WFS	The OWS-9 AIXM 5.1 Demonstrator WFS provides access to a wide range of AIXM 5.1 features for the EUROPE sector. DISCLAIMER: This data should be used for research and development purposes only. It is not suitable for operational purposes.

The OWS Capabilities or WSDL documents were used to automatically generate ISO 19139 service metadata for each service and published into the registry. A request for all services in the registry returns the services as listed in the table above as shown in the following registry client screen shot.

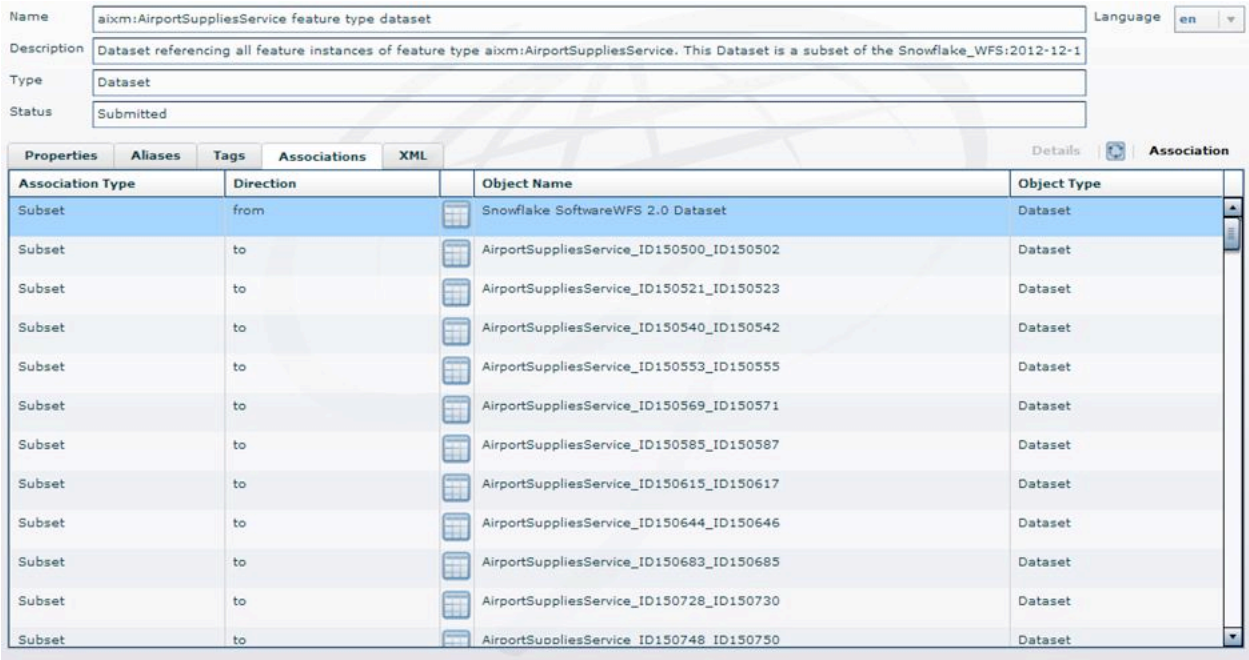


Note that the service metadata corresponding to the list of services shown above was also loaded in the SESAR Registry Demonstrator using WSDL documents as inputs (WSDL is required by the SESAR Registry). In the cases where a WSDL document was not published, it was automatically generated from an OWS Capabilities document.

The complete resource inventory of feature and metadata instances offered by OGC Web Feature Service implementations (by Comsoft GmbH and Snowflake Software Ltd.) was also captured to supplement the feature type service metadata found in OGC WFS capabilities documents. A screen shot showing the result of the request for the entire resource inventory of the Snowflake WFS as a dataset collection is shown in the registry client as follows:



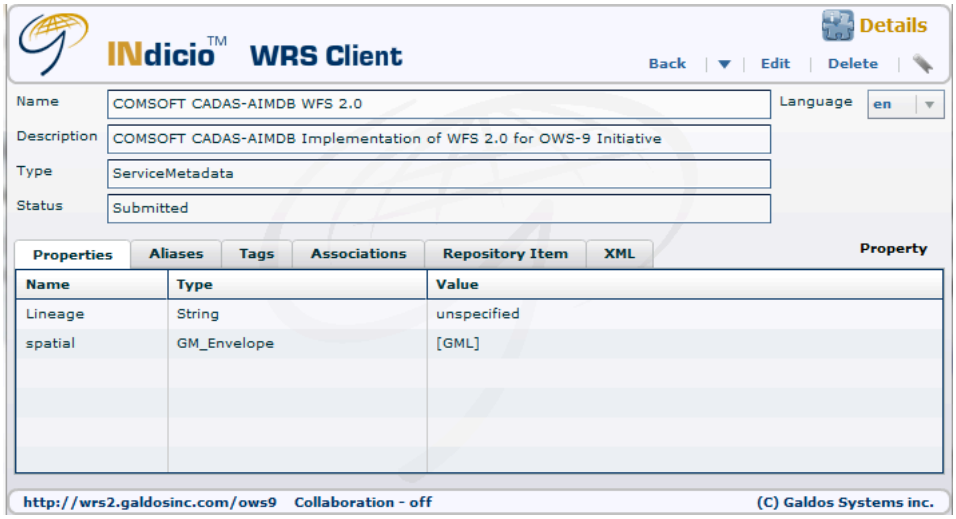
The Snowflake and Comsoft WFS inventory consists of AIXM 5.1 datasets (EAD, Estonia, etc.). Each WFS dataset collection was organized into a hierarchy using the ISO 19115/19139 Dataset Metadata model, in this case, divided into a collection of feature type datasets, which are subsets of the WFS Dataset, shown as 'Subset' associations in the above screenshot. The feature type dataset collection metadata was further divided into feature instance metadata subsets as shown for the `aixm:AirportSuppliesService` feature type dataset in the following registry client screenshot. The 'Subset' association with direction 'from' the WFS dataset, in the highlighted row of the screenshot indicates that the feature type dataset is a sub-collection of the WFS dataset collection. The 'Subset' associations with direction 'to' the `AirportSuppliesService_*` feature instance metadata in the rows following the highlighted row indicate that the feature instance metadata is a subset of the feature type dataset collection.



Since the dataset and service records in the registry are linked in the registry model by the 'OperatesOn' association, a stored query can be issued that finds all services that operate on a dataset record given a feature id (either the `gml:id` or `gml:identifier` values can be used). For example, if we want to find all services that operate on the feature with `gml:id="gid-634788952283357863"`, the following stored query (called ServicesByFeatureId) can be issued in any web browser:

http://wrs2.galdosinc.com/ows9/query/**stored**/**ServicesByFeatureId**?id=gid-634788952283357863

This query returns the COMSOFT CADAS-AIMDB WFS 2.0 Metadata Service record (with id=`urn:uuid:8d6f51ad-cbc1-4672-8fea-4ac2c7d1b77c`), which when retrieved using the registry client has the following screenshot:



The loaded metadata representing dataset collections and dataset instances capture all of the fields specified by the requirements and guidance for the Aviation ISO 19115 metadata profile (OGC 11-171 and OGC 11-172). In addition, the registry captures other useful metadata including feature instance counts, metadata instance counts, and raw XML document size, as summarized in the following table for the Snowflake WFS.

Snowflake EU WFS AIXM 5.1 Data Inventory (2012-12-18T15:12:16-08.00)				
	Feature Types (from Capabilities)	Instance Count	MD_Metadata Count	File Size (KB)
1	AeronauticalGroundLight	2	0	3.96
2	AircraftGroundService	1	0	2.64
3	AircraftStand	1	0	2.70
4	AirportClearanceService	3	0	5.74
5	AirportHeliport	2456	2457	23321.49
6	AirportHeliportCollocation	1	0	2.37
7	AirportSuppliesService	135	135	859.09
8	Airspace	8395	8398	70737.14
9	AirTrafficControlService	0	0	1.03
10	AirTrafficManagementService	0	0	1.03
11	AngleIndication	187	0	202.47
12	ApproachLightingSystem	6	0	9.44
13	Apron	4	4	26.11
14	ApronElement	3	3	22.41
15	ApronLightSystem	0	0	1.03
16	ApronMarking	0	0	1.03
17	ArrestingGear	91	0	135.63
18	ArrivalLeg	0	0	1.03
19	AuthorityForAirspace	527	0	669.64
20	DepartureLeg	10	0	13.05
21	DesignatedPoint	10295	0	14302.08
22	DistanceIndication	130	0	149.21

23	DME	677	0	1650.00
24	FinalLeg	453	0	784.27
25	FireFightingService	1	0	3.46
26	GeoBorder	79	79	1683.52
27	Glidepath	318	0	690.24
28	GroundTrafficControlService	0	0	1.03
29	GuidanceLine	1	1	7.23
30	HoldingPattern	6	0	11.48
31	InformationService	0	0	1.04
32	InitialLeg	394	0	581.94
33	InstrumentApproachProcedure	573	0	2089.59
34	IntermediateLeg	925	0	1524.95
35	Localizer	366	0	843.87
36	MarkerBeacon	353	0	865.25
37	MissedApproachLeg	0	0	1.03
38	Navaid	1841	1843	11550.05
39	NDB	738	0	1817.70
40	ObstacleArea	22	0	27.69
41	OrganisationAuthority	1011	0	1776.42
42	PassengerService	1	0	2.62
43	RadioCommunicationChannel	0	0	1.03
44	Route	11350	0	13374.77
45	RouteSegment	9694	0	32228.91
46	Runway	2527	2527	15075.52
47	RunwayCentrelinePoint	2907	0	6998.19
48	RunwayDirection	4429	4431	24829.80
49	RunwayDirectionLightSystem	19	19	117.93
50	RunwayElement	2	2	14.26
51	RunwayMarking	3	0	5.49
52	RunwayProtectArea	3	0	6.77
53	RunwayVisualRange	93	0	144.55
54	SafeAltitudeArea	229	0	826.13
55	SearchRescueService	0	0	1.03
56	SignificantPointInAirspace	788	0	1048.74
57	SpecialDate	811	0	1023.01
58	StandardInstrumentArrival	14	0	21.75
59	StandardInstrumentDeparture	3	0	8.44
60	StandardLevelColumn	4	0	33.92
61	StandardLevelTable	2	0	3.95
62	TACAN	99	0	265.12
63	Taxiway	14	0	19.83
64	TaxiwayElement	7	7	49.45
65	TaxiwayLightSystem	0	0	1.03

66	TaxiwayMarking	0	0	1.03
67	TouchDownLiftOff	234	0	442.17
68	TouchDownLiftOffMarking	13	0	20.88
69	Unit	2624	0	3926.92
70	VerticalStructure	8304	8307	58572.19
71	VisualGlideSlopeIndicator	498	0	653.75
72	VOR	422	0	1068.02

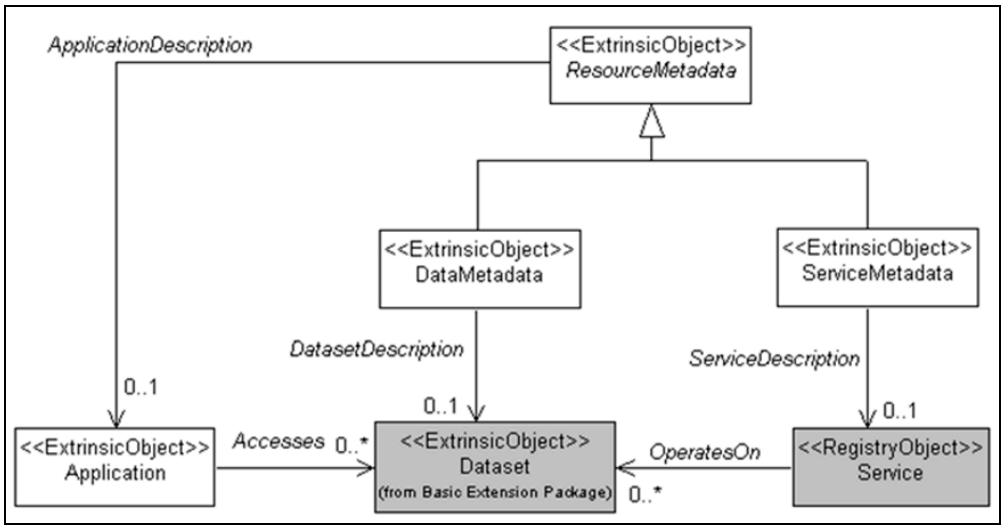
Further details of how CIM was used to publish aviation resources are contained in section 6. The harvest objects are summarized using deployed examples in sections 6.1.1 and 6.2.1 and the corresponding registration process is described in sections 6.1.2 and 6.2.2. A CIM deficiency related to lineage/provenance registration was identified and the proposed CIM extensions to meet the aviation requirements are documented in section 6.1.3. The proposed CIM extensions were a welcome submission to the CIM SWG and the status to date is that they have been accepted without modification for addition to the CIM candidate standard by the CIM SWG chair to be voted on through the OGC SWG consensus process.

The selective and efficient retrieval study, discussed in the Aviation Metadata & Provenance ER, was investigated using the deployed OGC CSW-ebRIM registry service and the results include a fast and iterative Metadata X-Ray retrieval implementation (as described in the OWS-9 Metadata and Provenance ER) in section 7.

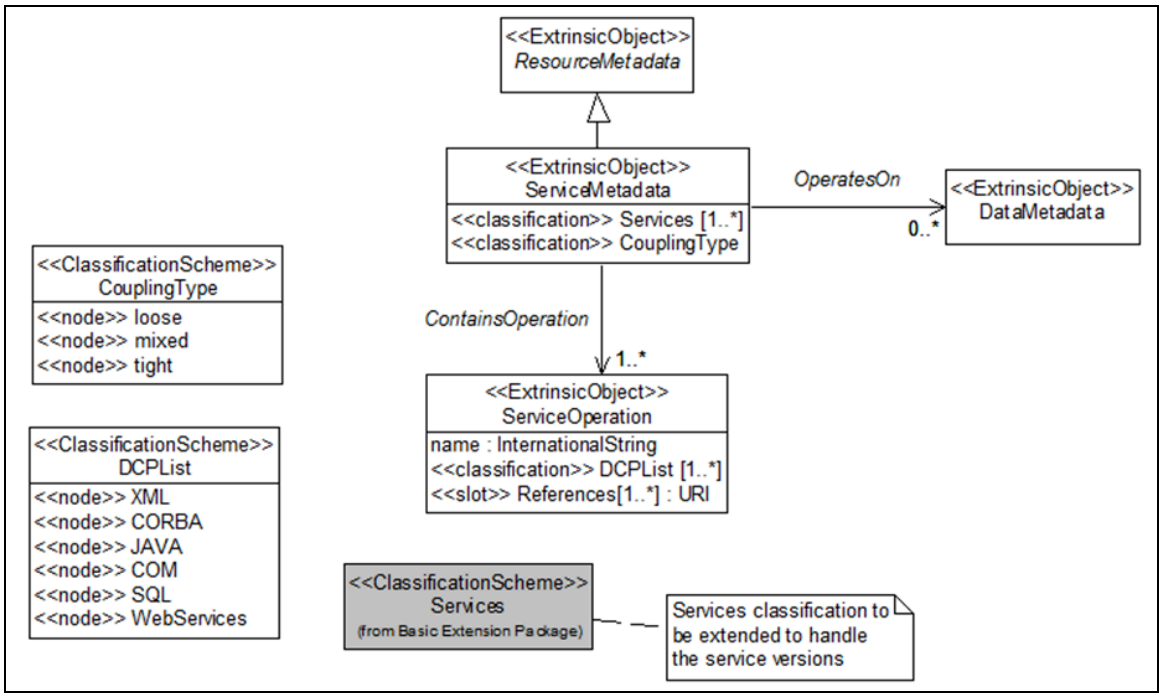
Finally a gap analysis was conducted between the SESAR and CSW-ebRIM registries and the results discussed in section 7.2.1.1.

6 Use of OGC Cataloging ISO Metadata (CIM) Registry Model

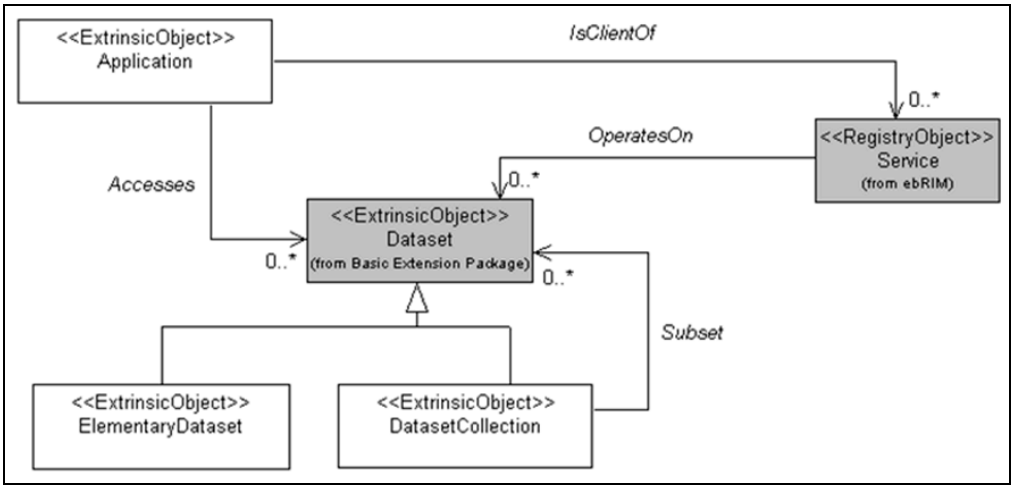
The CIM registry model consists of an XML registry package that can be loaded into any CSW-ebRIM registry service to manage ISO metadata for datasets, services, and associations to other discoverable objects. In CIM, dataset and service metadata are both represented as a type of resource metadata object with an association between them, as depicted in CIM ([7], Fig 16), shown below:



The service metadata object is further described by attributes (slots) and classified by multiple taxonomies (represented by the ebRIM classification and ClassificationScheme stereo-types) for optimal discovery as shown below, from CIM ([7], Fig 8):



Similarly, the CIM dataset metadata object is described by attributes and classifications. The dataset hierarchy can also by represented as a dataset collection containing subsets of more elementary datasets as shown below, from CIM ([7], Fig 15):

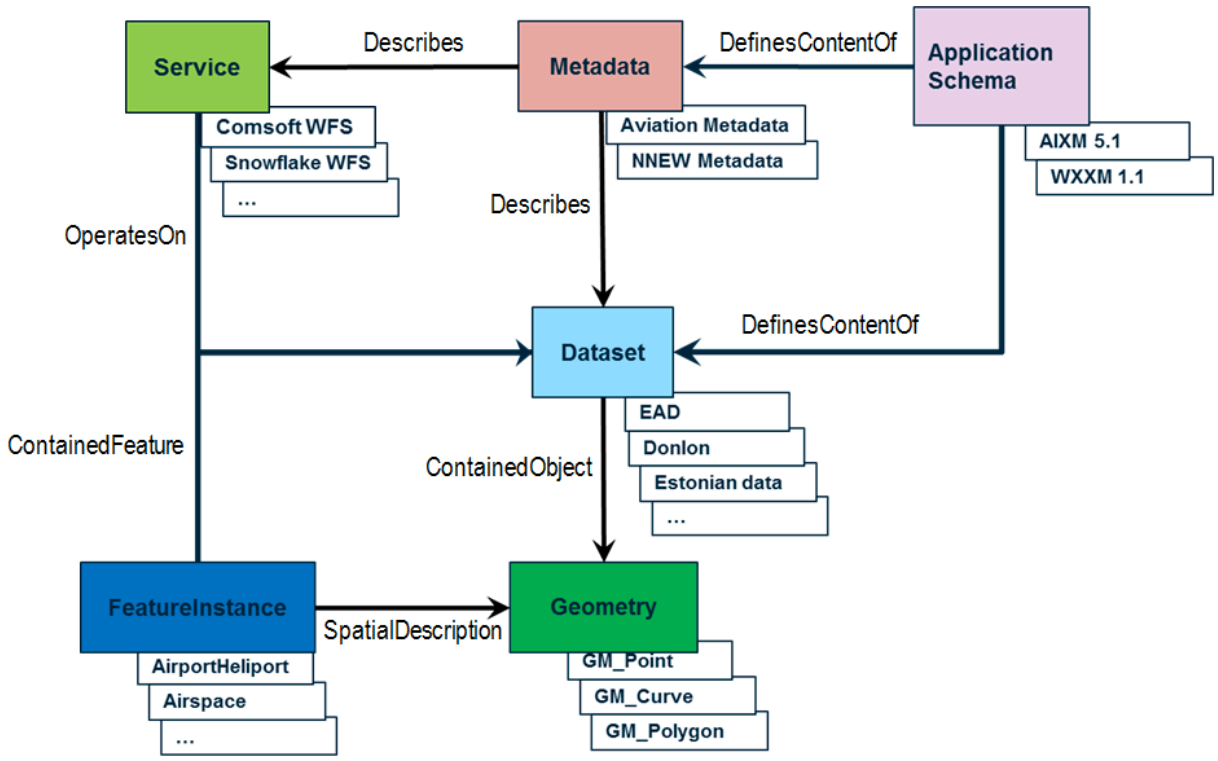


6.1 Dataset Metadata (ISO 19115)

In order to load and manage the complete dataset metadata model meeting the requirements of the Aviation and NNEW metadata profiles using CIM, an extension for the ISO Data Quality component in CIM was required and developed in OWS-9. In particular, the lineage/provenance metadata is a required subcomponent of Data Quality in Aviation. The extension of CIM developed in OWS-9, was contributed to the OGC CIM SWG and is the topic of the formal OGC Change Request ([11]), OGC CIM CR: LI_Lineage (Provenance) Components (OGC 13-013). After deploying the Data Quality extension, documented in Section 6.1.3 (and in [11]), the dataset metadata instances were harvested and published to the Aviation Registry. The steps involved in publishing to the CIM model are described in the next sections.

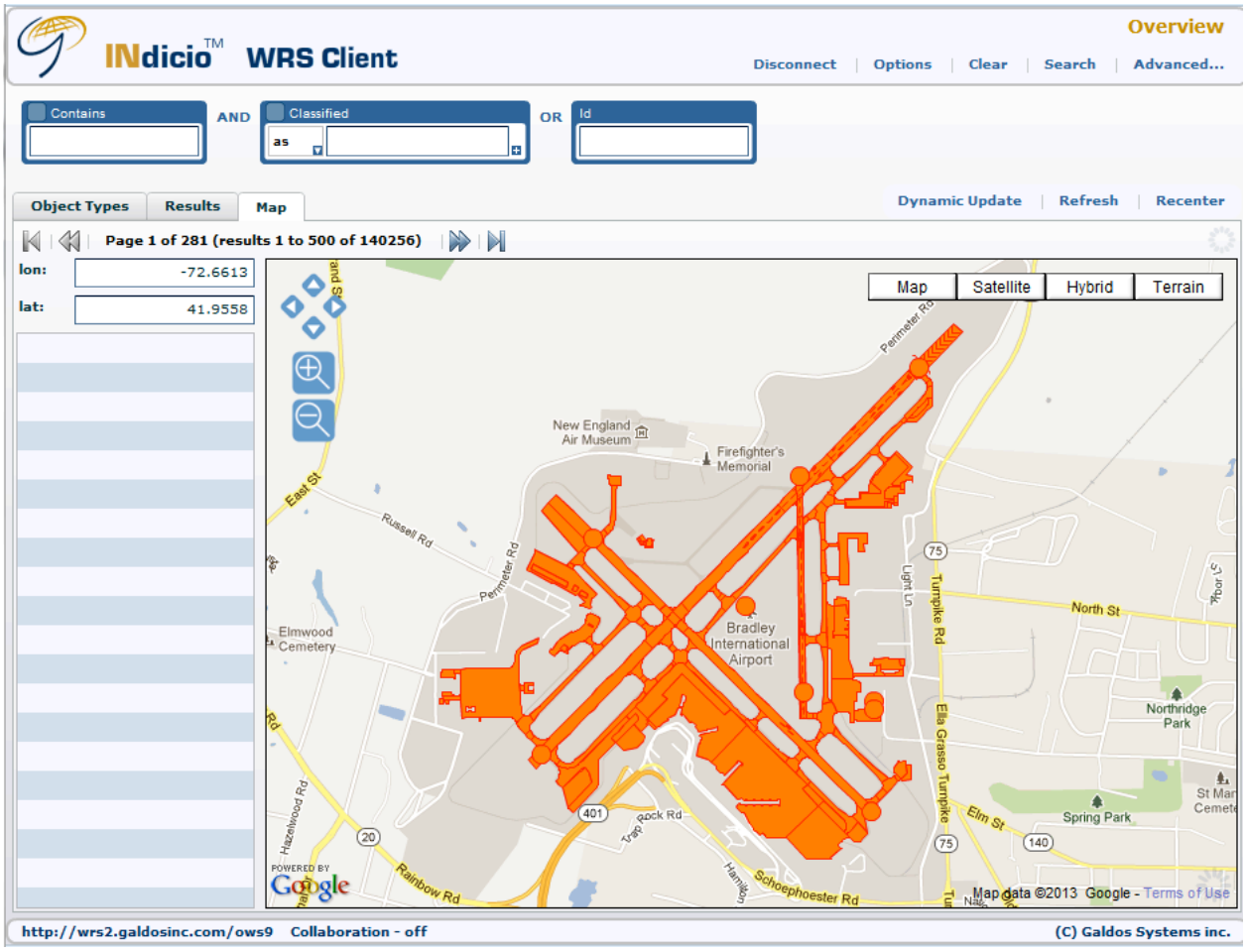
6.1.1 Resource Metadata Harvested in OWS-9

Several types of discovery metadata were harvested into the OWS-9 Aviation Registry including application schemas, ISO 19115/19139 Metadata, feature geometry (to enable discovery by spatial filters) including associations between these resources. Multiple instances (in the case of dataset metadata, several hundred thousand) of each of the following resource types shown in the following aviation domain reference model, were harvested in the OGC CSW-ebRIM registry service for a total of over a million registry objects.



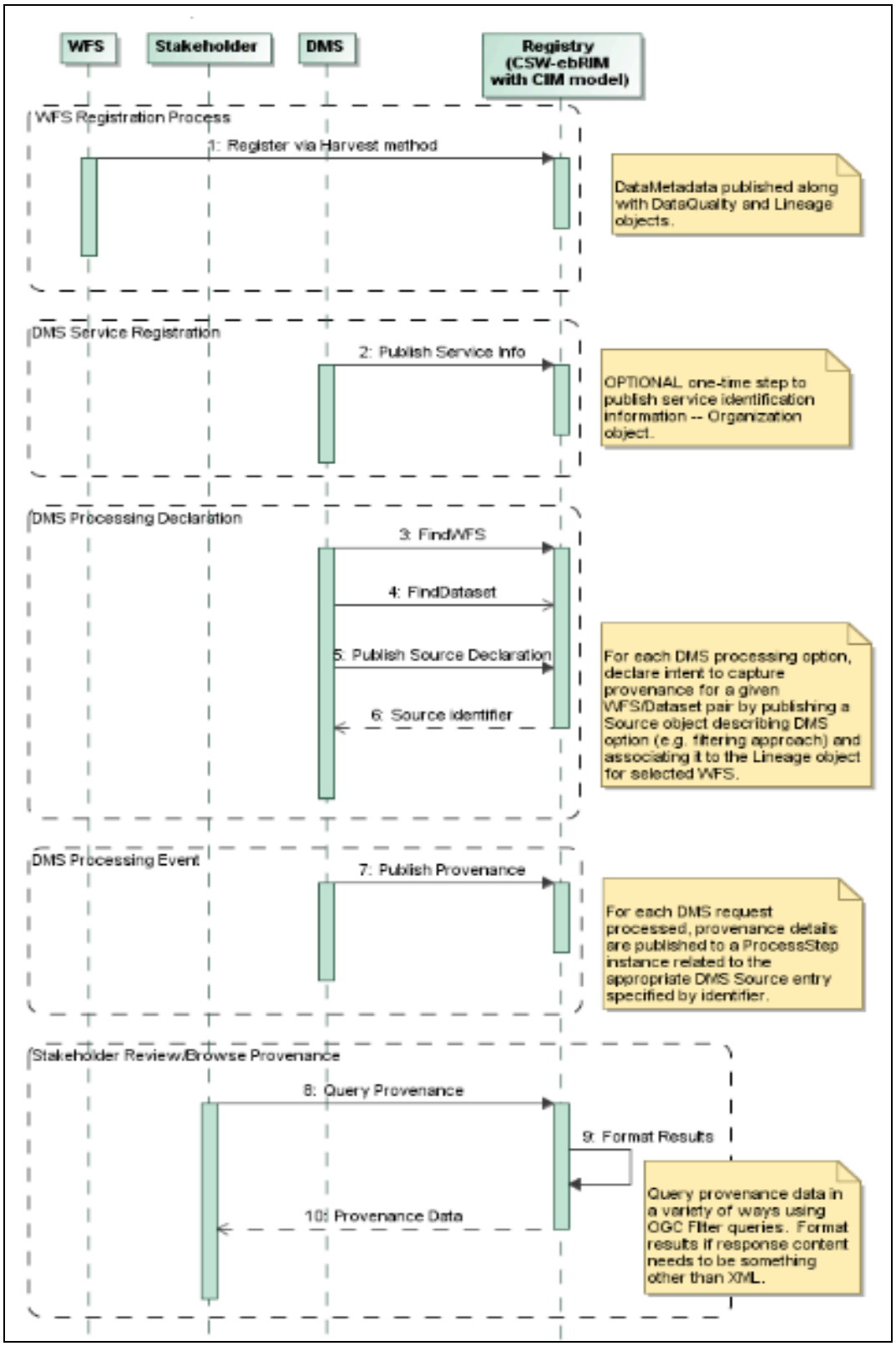
Several examples including some registry client screenshots are shown in Section 5 (Executive Summary/Overview).

Dataset metadata from the WFS service AIXM feature instances were harvested, including: the airport/heliport, runways, taxiways, aprons, etc. If ISO 19139 metadata was not provided with the feature or its time-slice instances, then a minimal encoding was automatically generated during the harvest operation. The feature's geometry was also harvested at the same time as the ISO dataset metadata from every feature instance in each of the WFS Services to enable OGC CSW-ebRIM spatial queries. A sample search result displayed in the 'map' view of the registry client is as follows.



6.1.2 Publication Process

The different components of the CIM model are typically published by different actors, as shown for one OWS-9 aviation scenario illustrated in the following sequence diagram:



6.1.3 Data Quality Extension to CIM

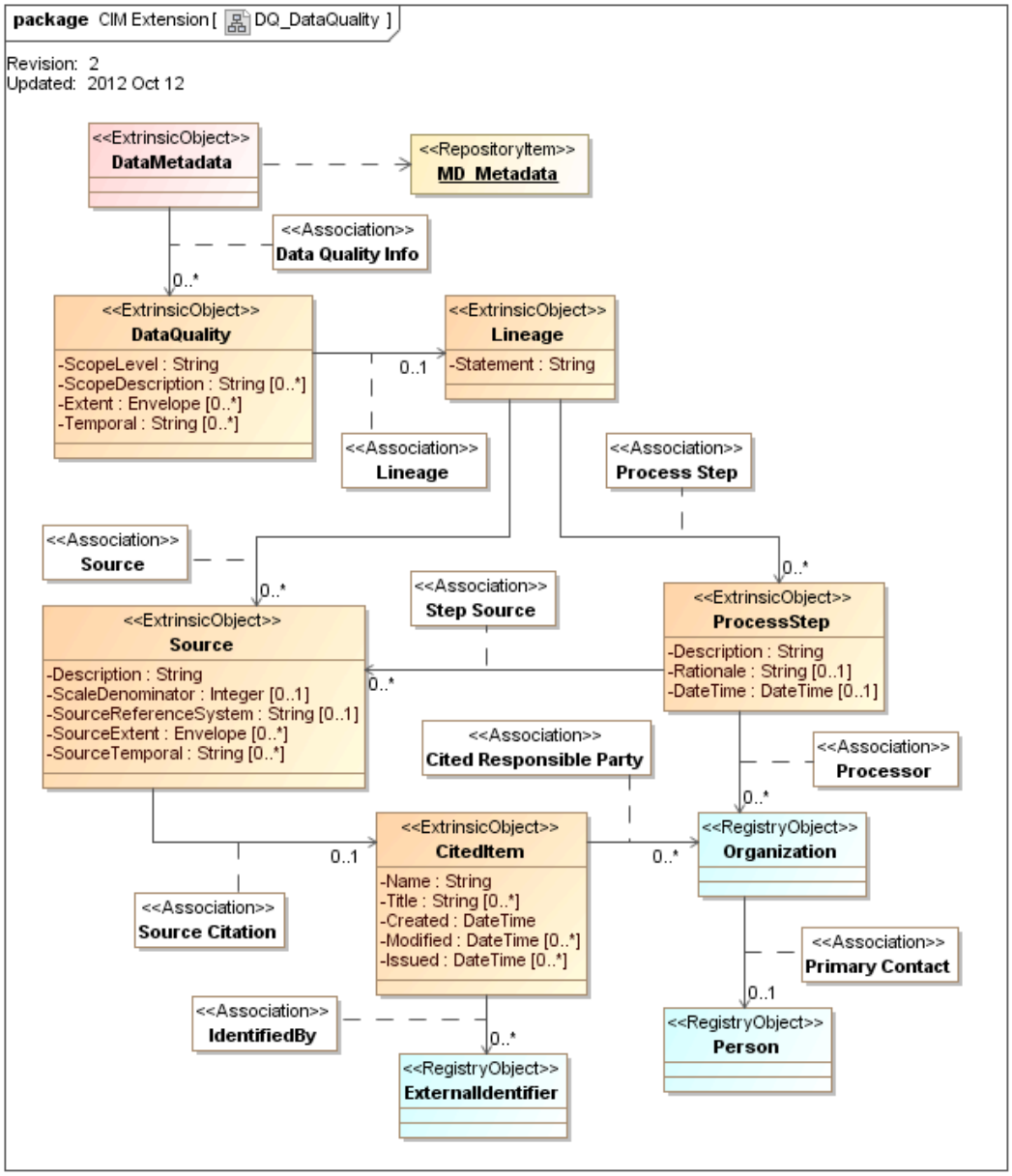
In OWS-9, it was necessary to make use of the ISO 19139 Data Quality model for registration in CSW-ebRIM, in order to properly discover/query for data quality and its subcomponents in the CSW-ebRIM service. The latest version of the Cataloguing ISO Metadata (CIM) document (OGC 07-038r3, v0.1.11, 2009-07-14) had not specified a mapping for the lineage/provenance components contained in ISO 19139 `DQ_DataQuality` (including `LI_Lineage`, `LI_ProcessStep`, and `LI_Source`) to the CIM ebRIM model. The extent of the data quality mapping information in the CIM standard was very limited and is shown in the following table:

CIM Table 39: - From DQ_ConformanceResult to QualityConformanceInformation

ISO 19115/ ISO 19119	CIM	Comments
Title	<<slot>>specTitle	
dateType	<<classification>> DateType	
Date	<<slot>>specDate	
Pass	<<slot>> specPass	

In OWS-9, we used this opportunity to extend the CIM registry model and deployment package to include the necessary lineage/provenance components.

The focus of this proposed extension was to support ISO `DQ_DataQuality` elements relevant to the metadata provenance scenario, including capturing the metadata corresponding to a Data Management Service (DMS) processing a WFS request for re-distribution to the DMS clients.



The details of the CIM registry model extension for data quality are documented in data dictionary and mapping tables in the following sub-sections. The new additions were developed in this extension following the same pattern as the existing CIM mappings and are proposed additions to the CIM standard.

6.1.3.1 Data Dictionary for Data Quality Components

CIM Model Name	Definition	Obligation / Condition	Maximum occurrence	Stereotype	Data type / Target object	Comment

DataQuality	Properties from gmd:DQ_DataQuality	Use obligation from referencing object	Use maximum occurrence from referencing object	<<ExtrinsicObject>>	Specialization of ExtrinsicObject	
ScopeLevel	the level to which the data quality information applies	M	1	<<Slot>>	String	
ScopeDescription	description of the scope	O	N	<<Slot>>	String	
Extent	applicable spatial extent for the data quality information	O	N	<<Slot>>	gml:Envelope	
Temporal	applicable time range for the data quality information	O	N	<<Slot>>	String	follows ISO 8601 Time Interval: {startDateTime} / {endDateTime}
Association type: Report	quantitative quality information for the data specified by the scope	C / lineage not provided ?	N	<<Association>>	----	Ignored; not in scope.
Association type: Lineage	non-quantitative quality information about the lineage of the data specified by the scope	C / report not provided ?	1	<<Association>>	Lineage	See Lineage object defintion below
Lineage	Properties from gmd:LI_Lineage	Use obligation from referencing object	Use maximum occurrence from referencing object	<<ExtrinsicObject>>	Specialization of ExtrinsicObject	
Statement	general explanation of the data producer's knowledge of the lineage of a dataset	C / DataQuality. ScopeLevel = 'dataset' or 'series' ?	1	<< Description >> (core RIM object property, not Slot)	InternationalString	Maximum 1024 characters for an InternationString in ebRIM.
Association type: ProcessStep	information about events in the life of a dataset specified by the scope	C / if Statement and Source not provided	N	<<Association>>	ProcessStep	See ProcessStep object definition below
Association type: Source	information about events used in creating the data specified by the scope	C / if Statement and ProcessStep not provided	N	<<Association>>	Source	See Source object definition below
ProcessStep	Properties from gmd:LI_ProcessStep	Use obligation from referencing object	Use maximum occurrence from referencing object	<<ExtrinsicObject>>	Specialization of ExtrinsicObject	
Description	description of the event, including related parameters or tolerances	M	1	<< Description >> (core RIM object property, not Slot)	InternationalString	Maximum 1024 characters for an InternationString in ebRIM.

Rationale	requirement or purpose for the process step	O	1	<<Slot>>	String	
DateTime	date and time, or range of date and time, on, or over, which the process step occurred	O	1	<<Slot>>	DateTime	
Association type: Processor	identification of, and means of communication with, person(s) and organization(s) associated with the process step	O	N	<<Association>>	Organization	Organization is a core RegistryObject in ebRIM
Association type: StepSource	information about the source data used in creating the data specified by the scope	O	N	<<Association>>	Source	See Source object definition below
Source	Properties from gmd:LI_Source	Use obligation from referencing object	Use maximum occurrence from referencing object	<<ExtrinsicObject>>	Specialization of ExtrinsicObject	
Description	detailed description of the level of the source data	M	1	<< Description >> (core RIM object property, not Slot)	InternationalString	Maximum 1024 characters for an InternationString in ebRIM.
ScaleDenominator	denominator of the representative fraction on a source map	O	1	<<Slot>>	Integer	
SourceReference System	spatial reference system used by the source data	O	1	<<Slot>>	String	
Association type: SourceCitation	recommended reference to be used for the source data	O	1	<<Association>>	CitedItem	See CitedItem object definition below
SourceExtent	information about the spatial extent of the source data	C / Description not provided ?	N	<<Slot>>	gml:Envelope	
SourceTemporal	information about the temporal extent of the source data	C / Description not provided ?	N	<<Slot>>	String	follows ISO 8601 Time Interval: {startDateTime} / {endDateTime}

Association type: SourceStep	information about an event in the creation process for the source data	O	N	<<Association>>	----	Ignored; not in scope.
CitedItem	Properties from gmd:CI_Citation	Use obligation from referencing object	Use maximum occurrence from referencing object	<<ExtrinsicObject>>	Specialization of ExtrinsicObject	
Name	name by which the cited resource is known	M	1	<<Name>> (core RIM object property, not Slot)	InternationalString	Maximum 1024 characters for an InternationString in ebRIM.
Title	short name or other language name of the dataset	O	N	<<Slot>>	String	
Created	date of creation of the resource	O	1	<<Slot>>	DateTime	
Modified	date of revision of the resource	O	N	<<Slot>>	DateTime	
Issued	date of formal issuance (e.g., publication) of the resource	O	N	<<Slot>>	DateTime	
Association type: IdentifiedBy	value uniquely identifying the resource within a namespace	O	N	<<Association>>	ExternalIdentifier	ExternalIdentifier is a core RegistryObject in ebRIM
Association type: CitedResponsibleParty	identification of, and means of communication with, person(s) and organization(s) associated with the resource(s)	O	N	<<Association>>	Organization	Organization is a core RegistryObject in ebRIM

6.1.3.2 CIM Registry Object and Association Types for Data Quality

New CIM model Object types are defined in the Data Dictionary	
ObjectType	URN of ObjectType ClassificationNode
DataQuality	urn:ogc:def:objectType:CIM::DataQuality
Lineage	urn:ogc:def:objectType:CIM::Lineage
Source	urn:ogc:def:objectType:CIM::Source
ProcessStep	urn:ogc:def:objectType:CIM::ProcessStep
AssociationType	URN of AssociationType ClassificationNode

Lineage	urn:ogc:def:associationType:CIM::Lineage
ProcessStep	urn:ogc:def:associationType:CIM::ProcessStep
Source	urn:ogc:def:associationType:CIM::Source
Processor	urn:ogc:def:associationType:CIM::Processor
StepSource	urn:ogc:def:associationType:CIM::StepSource
SourceCitation	urn:ogc:def:associationType:CIM::SourceCitation
IdentifiedBy	Reuse ~ Already defined in CSW-ebRIM Basic Package

6.1.3.3 Mapping ISO Data Quality to CIM Registry model

ISO DQ_DataQuality to CIM DataQuality (ExtrinsicObject)		
ISO 19115/19139 DQ_Quality	**CIM DataQuality Properties**	**Comments**
/gmd:DQ_DataQuality/gmd:scope/gmd:DQ_Scope /gmd:**level**/gmd:MD_ScopeCode	<<Slot>> ScopeLevel	
/gmd:DQ_DataQuality/gmd:scope/gmd:DQ_Scope /gmd:**levelDescription**/gmd:MD_ScopeDescription /gmd:dataset/gco:CharacterString	<<Slot>> ScopeDescription	
/gmd:DQ_DataQuality/gmd:scope /gmd:DQ_Scope/gmd:extent /gmd:EX_Extent/gmd:geographicElement /gmd:EX_GeographicBoundingBox /gmd:**westBoundLongitude**	<<Slot>> Extent (of type gml:Envelope)	For each gmd:**EX_GeographicBoundingBox** element an Extent Slot value (gml:Envelope) is generated. The **WestBoundLongitude** corresponds to the longitude of "lowerCorner" in gml:Envelope.
/gmd:DQ_DataQuality/gmd:scope /gmd:DQ_Scope/gmd:extent /gmd:EX_Extent/gmd:geographicElement /gmd:EX_GeographicBoundingBox /gmd:**eastBoundLongitude**	<<Slot>> Extent (of type gml:Envelope)	The **EastBoundLongitude** corresponds to the longitude of "upperCorner" in gml:Envelope
/gmd:DQ_DataQuality/gmd:scope /gmd:DQ_Scope/gmd:extent /gmd:EX_Extent/gmd:geographicElement /gmd:EX_GeographicBoundingBox /gmd:**southBoundLatitude**	<<Slot>> Extent (of type gml:Envelope)	The **SouthBoundLatitude** corresponds to the latitude of "lowerCorner" in gml:Envelope.
/gmd:DQ_DataQuality/gmd:scope /gmd:DQ_Scope/gmd:extent /gmd:EX_Extent/gmd:geographicElement /gmd:EX_GeographicBoundingBox /gmd:**northBoundLatitude**	<<Slot>> Extent (of type gml:Envelope)	The **NorthBoundLongitude** corresponds to the latitude of "upperCorner" in gml:Envelope
/gmd:DQ_DataQuality/gmd:scope/gmd:DQ_Scope /gmd:extent/gmd:EX_Extent/gmd:temporalElement /gmd:EX_TemporalExtent/gmd:extent /gml32:TimePeriod/gml32:**begin** /gml32:TimeInstant/gml32:timePosition	<<Slot>> Temporal (start date-time portion)	For each gml32:**TimePeriod** element a Temporal slot value is generated. Corresponds to the **startDate** part of the string TimePeriod: {startDate}/{endDate}

/gmd:DQ_DataQuality/gmd:scope/gmd:DQ_Scope /gmd:extent/gmd:EX_Extent/gmd:temporalElement /gmd:EX_TemporalExtent/gmd:extent /gml32:TimePeriod/gml32:**end** /gml32:TimeInstant/gml32:timePosition	<<Slot>> Temporal (end date-time portion)	Corresponds to the **endDate** part of the string TimePeriod: {startDate}/{endDate}
/gmd:DQ_DataQuality/gmd:**lineage**	<<Association>> Lineage	LIneage is mandatory if no Report is provided; for OWS-9 Aviation, Report is out of scope. Association type "**Lineage**" is generated with these properties: *Source object:* **DataQuality** *Target object:* **Lineage**
ISO LI_Lineage to CIM Lineage (ExtrinsicObject)		
ISO 19115 LI_Lineage	**CIM Lineage Properties**	**Comments**
/gmd:DQ_DataQuality/gmd:lineage/gmd:LI_Lineage /gmd:**statement**/gco:CharacterString	<<Slot>> Statement	
/gmd:DQ_DataQuality/gmd:lineage/gmd:LI_Lineage /gmd:**processStep/**gmd:LI_ProcessStep OR: /gmd:**processStep**/@xlink:href	<<Association>> ProcessStep plus optional object: <<ExtrinsicObject>> ProcessStep	For each gmd:**processStep** element. If **LI_ProcessStep** is specified, a corresponding **ProcessStep** instance must also be included. If instead, an @xlink:href is specified, then the **ProcessStep** object is assumed to have been published previously. In either case, an Association of type "**ProcessStep**" is required with these properties: *Source object:* **Lineage** *Target object:* **ProcessStep**
/gmd:DQ_DataQuality/gmd:lineage/gmd:LI_Lineage /gmd:**source**/gmd:LI_Source OR: /gmd:**source**/@xlink:href	<<Association>> Source plus optional object: <<ExtrinsicObject>> Source	For each gmd:**source** element. If **LI_Source** is specified, a corresponding **Source** instance must also be included. If instead, an @xlink:href is specified, then the **Source** object is assumed to have been published previously. In either case, an Association of type "**Source**" is required with these properties: *Source object:* **Lineage** *Target object:* **Source**
ISO LI_ProcessStep to CIM ProcessStep (ExtrinsicObject)		
ISO 19115 LI_ProcessStep	**CIM ProcessStep Properties**	**Comments**
/gmd:DQ_DataQuality/gmd:lineage /gmd:LI_Lineage/gmd:processStep /gmd:LI_ProcessStep/gmd:**description** /gco:CharacterString	<<Description>> Description	
/gmd:DQ_DataQuality/gmd:lineage /gmd:LI_Lineage/gmd:processStep /gmd:LI_ProcessStep/gmd:**rationale** /gco:CharacterString	<<Slot>> Rationale	

/gmd:DQ_DataQuality/gmd:lineage /gmd:LI_Lineage/gmd:processStep /gmd:LI_ProcessStep/gmd:**dateTime** /gco:DateTime	<<Slot>> DateTime	
/gmd:DQ_DataQuality/gmd:lineage /gmd:LI_Lineage/gmd:processStep /gmd:LI_ProcessStep/gmd:**processor** /gmd:CI_ResponsibleParty OR: /@xlink:href	<<Association>> Processor plus optional object: <<RegistryObject>> Organization	For each gmd:**processor** element. If **CI_ResponsibleParty** is specified, a corresponding **Organization** instance must also be included. If instead, an @xlink:href is specified, then the **Organization** object is assumed to have been published previously. In either case, an Association of type "**Processor**" is required with these properties: *Source object:* **ProcessStep** *Target object:* **Organization** (RegistryObject)
/gmd:DQ_DataQuality/gmd:lineage /gmd:LI_Lineage/gmd:processStep /gmd:LI_ProcessStep/gmd:**source** /gmd:LI_Source OR: /@xlink:href	<<Association>> StepSource plus optional object: <<ExtrinsicObject>> Source	For each gmd:s**ource** element. If **LI_Source** is specified, a corresponding **Source** instance must also be included. If instead, an @xlink:href is specified, then the **Source** object is assumed to have been published previously. In either case, an Association of type "**StepSource**" is required with these properties: *Source object:* **ProcessStep** *Target object:* **Source**
ISO LI_Source to CIM Source (ExtrinsicObject)		
ISO 19115 LI_Source	**CIM Source Properties**	**Comments**
/gmd:DQ_DataQuality/gmd:lineage /gmd:LI_Lineage/gmd:source /gmd:LI_Source/gmd:**description** /gco:CharacterString	<<Description>> Description	
/gmd:DQ_DataQuality/gmd:lineage /gmd:LI_Lineage/gmd:source /gmd:LI_Source/gmd:**scaleDenominator** /gmd:MD_RepresentativeFraction /gmd:denominator/gco:Integer	<<Slot>> ScaleDenominator	
/gmd:DQ_DataQuality/gmd:lineage /gmd:LI_Lineage/gmd:source /gmd:LI_Source/gmd:**sourceReferenceSystem** /gmd:MD_ReferenceSystem /gmd:referenceSystemIdentifier/gmd:RS_Identifier /gmd:code/gco:CharacterString	<<Slot>> SourceReferenceSystem	

/gmd:DQ_DataQuality/gmd:lineage /gmd:LI_Lineage/gmd:source /gmd:LI_Source/gmd:**sourceCitation** /gmd:CI_Citation OR: /@xlink:href	<<Association>> SourceCitation plus optional object: <<ExtrinsicObject>> CitedItem	For each gmd:s**ourceCitation** element. If **CI_CitedItem** is specified, a corresponding **Source** instance must also be included. If instead, an @xlink:href is specified, then the **CitedItem** object is assumed to have been published previously. In either case, an Association of type "**SourceCitation**" is required with these properties: *Source object:* **Source** *Target object:* **CitedItem**
/gmd:DQ_DataQuality/gmd:lineage /gmd:LI_Lineage/gmd:source /gmd:LI_Source/gmd:**sourceExtent** /gmd:EX_Extent/gmd:geographicElement /gmd:**EX_GeographicBoundingBox**	<<Slot>> SourceExtent (of type gml:Envelope)	For each gmd:**EX_GeographicBoundingBox** element an Extent Slot value (gml:Envelope) is generated. See mapping defined for DQ_DataQuality ... DQ_Scope ... EX_GeographicBoundingBox
/gmd:DQ_DataQuality/gmd:lineage /gmd:LI_Lineage/gmd:source /gmd:LI_Source/gmd:**sourceExtent** /gmd:EX_Extent/gmd:temporalElement /gmd:EX_TemporalExtent/gmd:extent /gml32:**TimePeriod**	<<Slot>> SourceTemporal	For each gml32:**TimePeriod** element a Temporal slot value is generated. See mapping defined for DQ_DataQuality ... EX_TemporalExtent ... TimePeriod

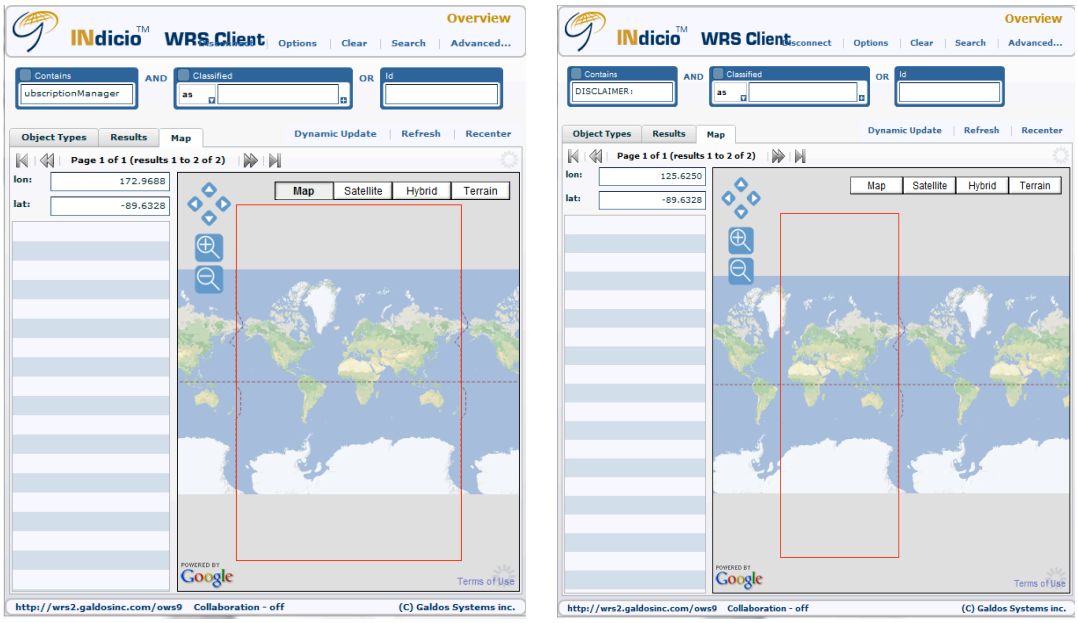
6.2 Service Metadata (ISO 19119)

ISO 19119 Service metadata provides complementary metadata to ISO 19115 dataset metadata and uses the same XML element 19139 `gmd:MD_Metadata` element for its representation. The core information contained in the ISO service metadata are the properties of the service instance that vary among instances of the same service type, such as the name, endpoint URL, supported inputs, operations, and output offerings.

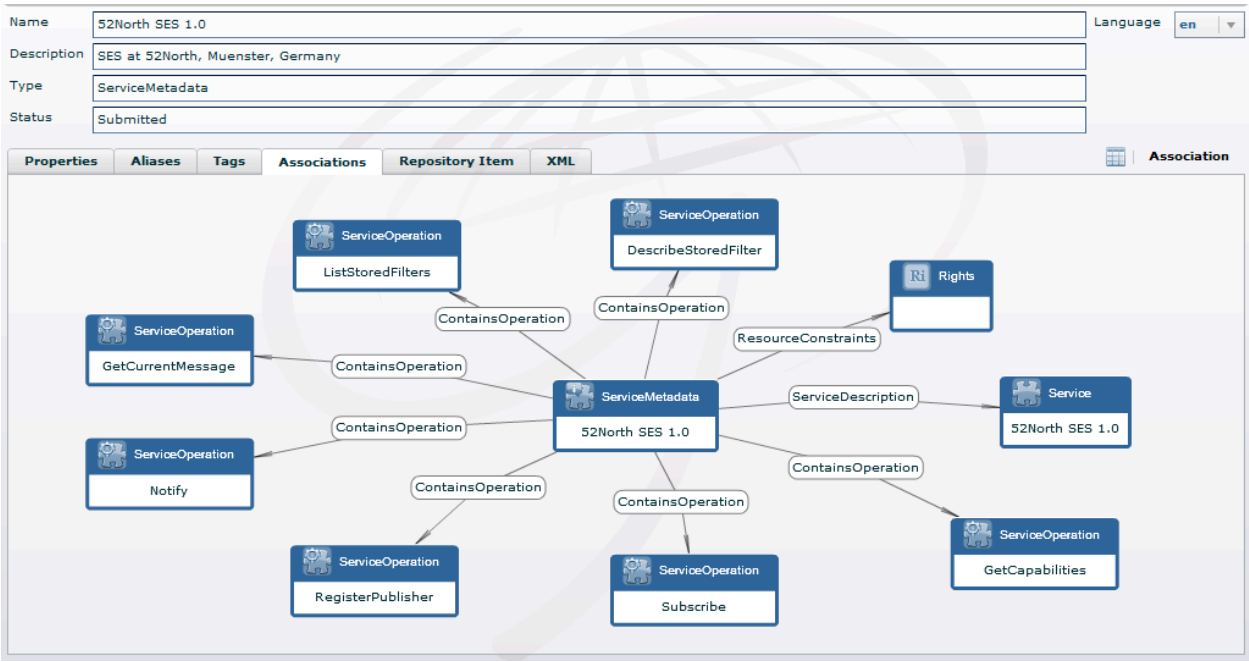
The OGC CIM registry model supports ISO 19119 service metadata. However, the OWS-9 deployed services were predominantly OGC Web Services (OWSs), so the majority of the service metadata content was captured by the mandatory OWS Capabilities documents. Therefore, the mapping of OWS capabilities documents to ISO 19119/19139 service metadata was investigated in OWS-9 and the corresponding transformation was executed during the registry harvest operation. This mapping to ISO service metadata was conducted for the different OWS types (e.g. WFS, FPS, WPS, CSW-ebRIM, etc.) that were deployed in OWS-9. The following sections provide examples of harvested metadata resources and the methodology used to publish and access the resources.

6.2.1 Resource Metadata Examples Harvested in OWS-9

As with dataset metadata, spatial filters in a registry query can be applied to the service metadata record and the response can be displayed in the 'map' view of the registry client showing the bounding box extent of the service's data offering (typically harvested from the OWS Capabilities documents). The following screenshots from left to right show the map view results of the ifGI OGC Event Service (full global coverage) and the Snowflake WFS (half globe coverage containing Europe/Asia), respectively:



Service metadata is associated to other registry objects representing operations supported by the service and constraints/limitations of use. Associations can be shown in registry client in either the graph view or list view. An example screenshot showing the associations in graph view, between a service metadata record and its operations and constraints, in the case of the 52North Event Service (SES), is as follows.

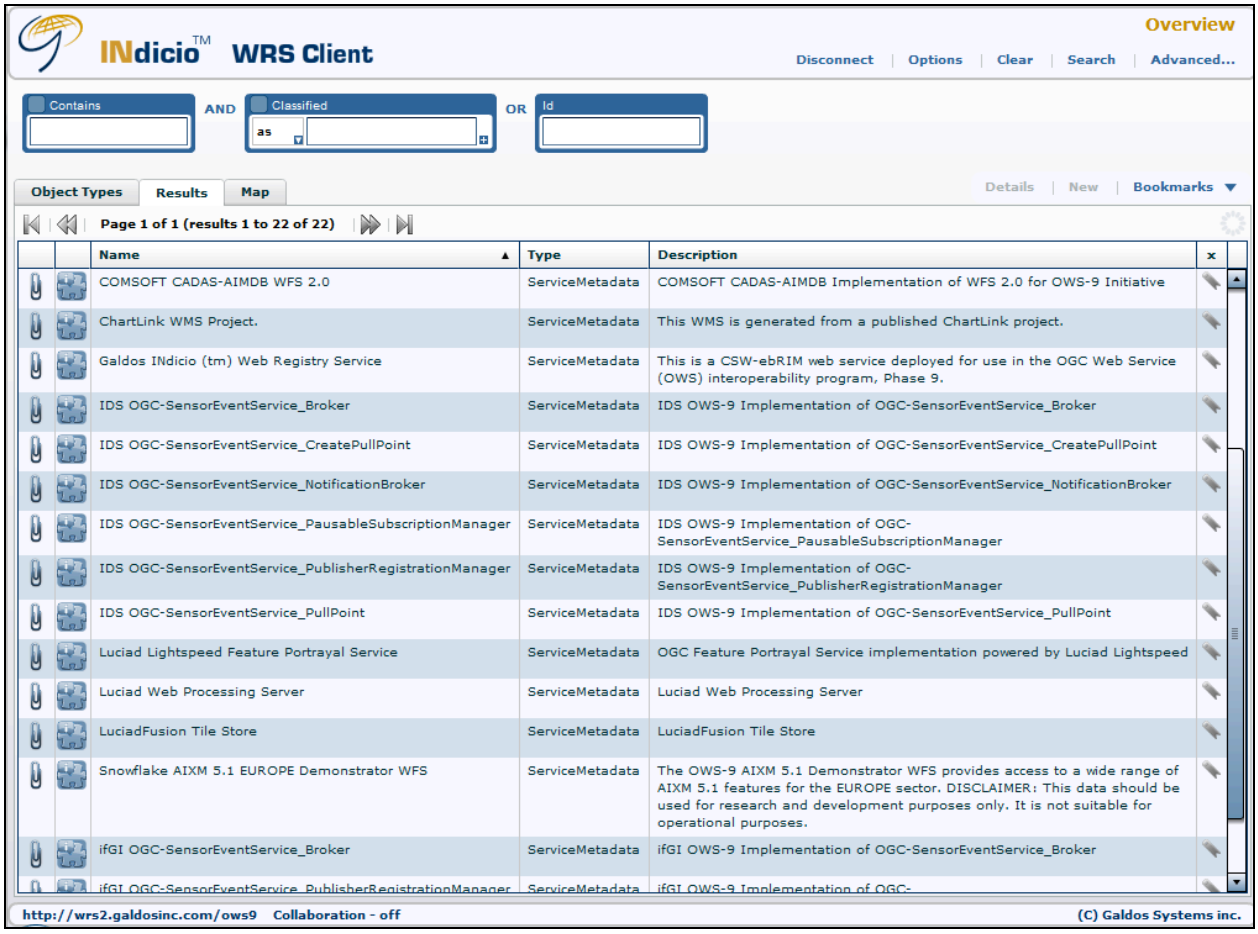


The list view showing the same associations is as follows:

The following OWS services were harvested for service and dataset metadata by using either OWS Capabilities or WSDL documents as inputs (see 1.4.1 and 8.5 for additional details on the use of WSDL inputs).

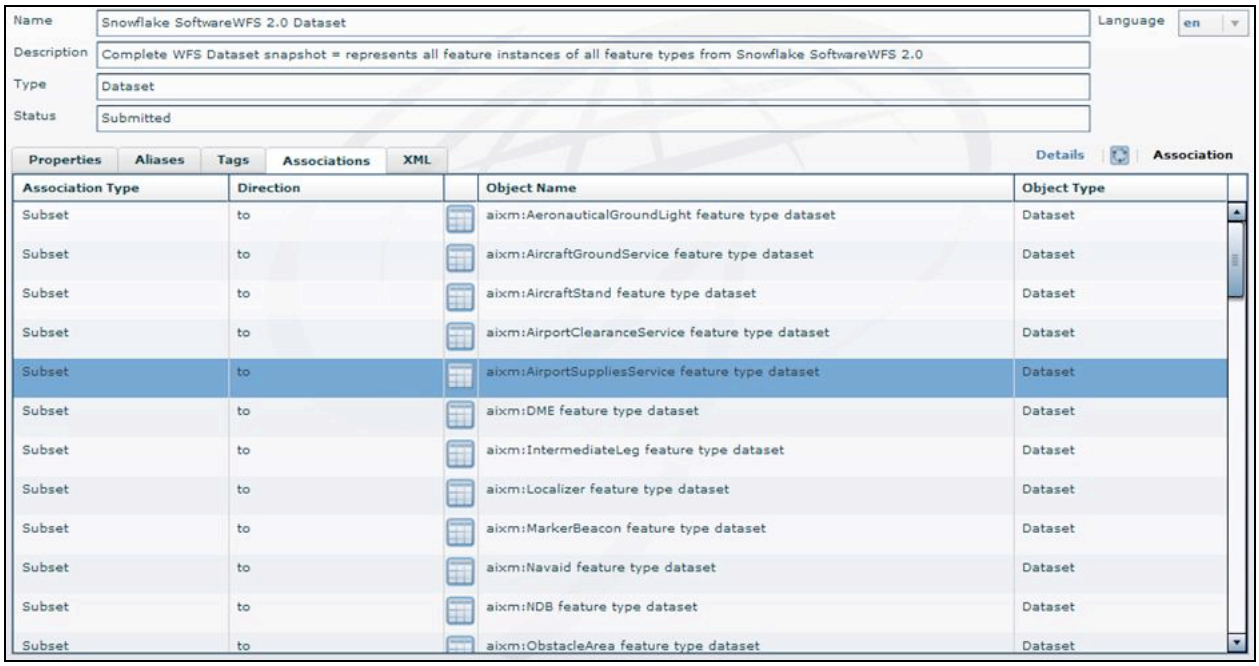
Service Resources Harvested in OWS-9		
Service Name		Service Description
1	52North SES 1.0	SES at 52North, Muenster, Germany
2	52North WPS 3.1-SNAPSHOT	Service based on the 52North implementation of WPS 1.0.0
3	ATM-TGS Data Management Service	ATM-TGS OWS-9 Implementation of Data Management Service
4	ATM-TGS Dispatch DMS	ATM-TGS OWS-9 Implementation of Dispatch Data Management Service
5	ATM-TGS Ground DMS	ATM-TGS OWS-9 Implementation of Ground Data Management Service
6	Envitia ChartLink WMS	Envitia WMS generated from a published ChartLink project
7	COMSOFT CADAS-AIMDB WFS 2.0	COMSOFT CADAS-AIMDB Implementation of WFS 2.0 for OWS-9 Initiative
8	Galdos INdicio Aviation Web Registry Service (OGC CSW-ebRIM)	This is a CSW-ebRIM web service deployed for use in the OGC Web Service (OWS) interoperability program, Phase 9.
9	IDS OGC SES Broker	IDS OWS-9 Implementation of OGC Sensor Event Service Broker
10	IDS OGC SES Create Pull Point	IDS OWS-9 Implementation of OGC Sensor Event Service Create Pull Point
11	IDS OGC SES Notification Broker	IDS OWS-9 Implementation of OGC Sensor Event Service Notification Broker
12	IDS OGC SES Pausable Subscription Manager	IDS OWS-9 Implementation of OGC Sensor Event Service Pausable Subscription Manager
13	IDS OGC SES Publisher Registration Manager	IDS OWS-9 Implementation of OGC Sensor Event Service Publisher Registration Manager
14	IDS OGC SES Pull Point	IDS OWS-9 Implementation of OGC Sensor Event Service Pull Point
15	ifGI OGC SES Service Broker	ifGI OWS-9 Implementation of OGC Sensor Event Service Broker
16	ifGI OGC SES Publisher Registration Manager	ifGI OWS-9 Implementation of OGC Sensor Event Service Publisher Registration Manager
17	ifGI OGC SES Subscription Manager	ifGI OWS-9 Implementation of OGC Sensor Event Service Subscription Manager
18	Luciad Lightspeed FPS	OGC Feature Portrayal Service implementation powered by Luciad Lightspeed
19	Luciad WPS	Luciad Web Processing Server
20	LuciadFusion Tile Store	LuciadFusion Tile Store
21	Snowflake AIXM 5.1 EUROPE Demonstrator WFS	The OWS-9 AIXM 5.1 Demonstrator WFS provides access to a wide range of AIXM 5.1 features for the EUROPE sector. DISCLAIMER: This data should be used for research and development purposes only. It is not suitable for operational purposes.

The OWS Capabilities or WSDL documents were used to automatically generate ISO 19139 service metadata for each service and published into the registry. A request for all services in the registry returns the following results in the registry client.

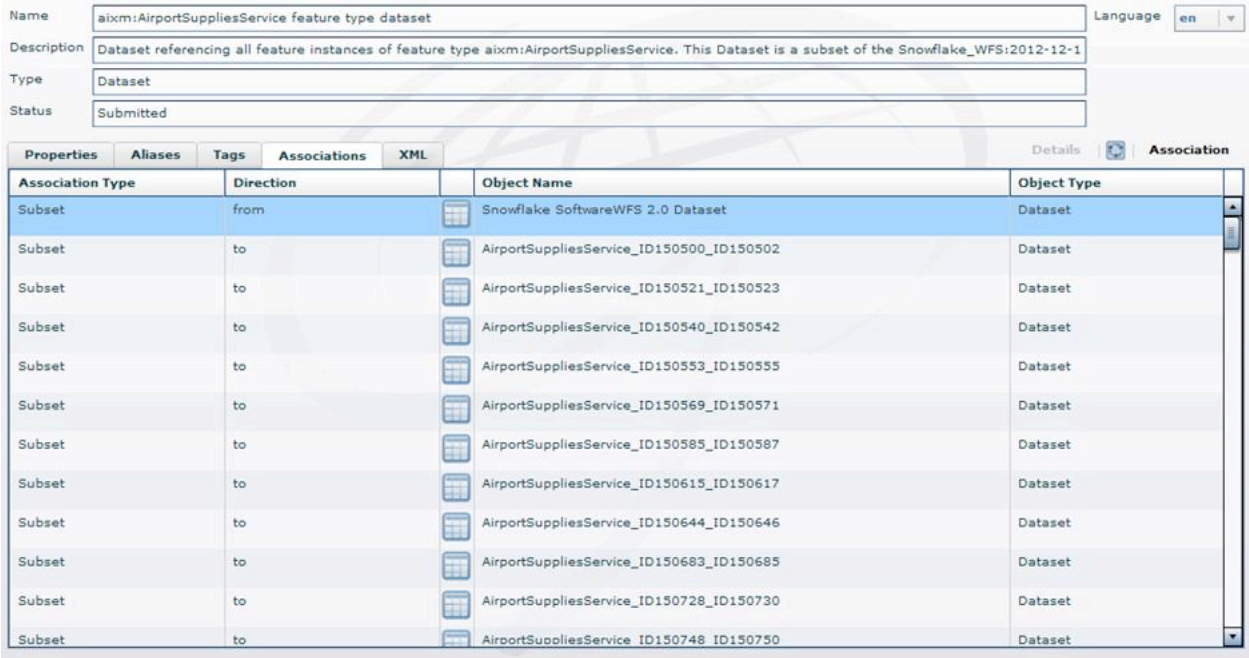


The service metadata corresponding to the list of services shown above was also loaded in the SESAR Registry Demonstrator using WSDL documents as inputs (WSDL is required by the SESAR Registry). In cases where a WSDL document was not provided, it was automatically generated from an OWS Capabilities document.

The complete resource inventory of feature and metadata instances offered by OGC Web Feature Service implementations (by Comsoft GmbH and Snowflake Software Ltd.) was also captured to supplement the feature type service metadata found in OGC WFS capabilities documents. A screen shot showing the entire resource inventory of the Snowflake WFS as a dataset collection is shown in the registry client as follows:



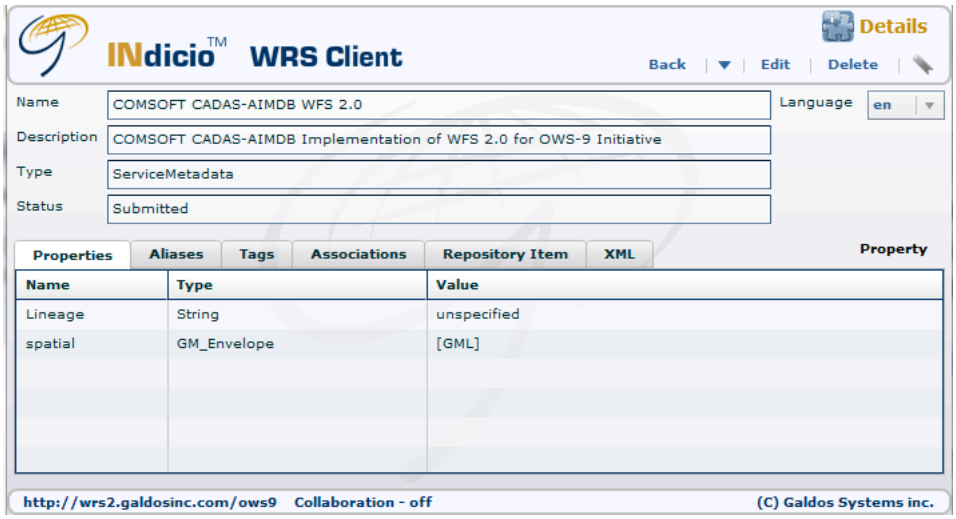
The Snowflake and Comsoft WFS inventory consists of AIXM 5.1 datasets (EAD, Estonia, etc.). Each WFS dataset collection was organized into a hierarchy using the ISO 19115/19139 Dataset Metadata model, in this case, divided into a collection of feature type datasets, which are subsets of the WFS Dataset, shown as 'Subset' associations in the above screenshot. The feature type dataset collection metadata was further divided into feature instance metadata subsets as shown for the `aixm:AirportSuppliesService` feature type dataset in the following registry client screenshot. The 'Subset' association with direction 'from' the WFS dataset, in the highlighted row of the screenshot indicates that the feature type dataset is a sub-collection of the WFS dataset collection. The 'Subset' associations with direction 'to' the `AirportSuppliesService_*` feature instance metadata in the rows following the highlighted row indicate that the feature instance metadata is a subset of the feature type dataset collection.



Since the dataset and service records in the registry are linked in the registry model by the 'OperatesOn' association, a stored query can be issued that finds all services that operate on a dataset record given a feature id (either the `gml:id` or `gml:identifier` values can be used). For example, if we want to find all services that operate on the feature with `gml:id="gid-634788952283357863"`, the following stored query (called ServicesByFeatureId) can be issued in any web browser:

http://wrs2.galdosinc.com/ows9/query/**stored**/**ServicesByFeatureId**?id=gid-634788952283357863

This query returns the COMSOFT CADAS-AIMDB WFS 2.0 Metadata Service record (with id=`urn:uuid:8d6f51ad-cbc1-4672-8fea-4ac2c7d1b77c`), which when retrieved using the registry client has the following screenshot:



The loaded metadata representing dataset collections and dataset instances capture all of the fields specified by the requirements and guidance for the Aviation ISO 19115 metadata profile (OGC 11-171 and OGC 11-172). In addition, the registry captures other useful metadata including feature instance counts, metadata instance counts, and raw XML document size, as summarized in the following table for the Snowflake WFS.

Snowflake EU WFS AIXM 5.1 Data Inventory (2012-12-18T15:12:16-08.00)				
	Feature Types (from Capabilities)	Instance Count	MD_Metadata Count	File Size (KB)
1	AeronauticalGroundLight	2	0	3.96
2	AircraftGroundService	1	0	2.64
3	AircraftStand	1	0	2.70
4	AirportClearanceService	3	0	5.74
5	AirportHeliport	2456	2457	23321.49
6	AirportHeliportCollocation	1	0	2.37
7	AirportSuppliesService	135	135	859.09
8	Airspace	8395	8398	70737.14
9	AirTrafficControlService	0	0	1.03
10	AirTrafficManagementService	0	0	1.03
11	AngleIndication	187	0	202.47
12	ApproachLightingSystem	6	0	9.44
13	Apron	4	4	26.11
14	ApronElement	3	3	22.41
15	ApronLightSystem	0	0	1.03
16	ApronMarking	0	0	1.03
17	ArrestingGear	91	0	135.63
18	ArrivalLeg	0	0	1.03
19	AuthorityForAirspace	527	0	669.64
20	DepartureLeg	10	0	13.05
21	DesignatedPoint	10295	0	14302.08

22	DistanceIndication	130	0	149.21
23	DME	677	0	1650.00
24	FinalLeg	453	0	784.27
25	FireFightingService	1	0	3.46
26	GeoBorder	79	79	1683.52
27	Glidepath	318	0	690.24
28	GroundTrafficControlService	0	0	1.03
29	GuidanceLine	1	1	7.23
30	HoldingPattern	6	0	11.48
31	InformationService	0	0	1.04
32	InitialLeg	394	0	581.94
33	InstrumentApproachProcedure	573	0	2089.59
34	IntermediateLeg	925	0	1524.95
35	Localizer	366	0	843.87
36	MarkerBeacon	353	0	865.25
37	MissedApproachLeg	0	0	1.03
38	Navaid	1841	1843	11550.05
39	NDB	738	0	1817.70
40	ObstacleArea	22	0	27.69
41	OrganisationAuthority	1011	0	1776.42
42	PassengerService	1	0	2.62
43	RadioCommunicationChannel	0	0	1.03
44	Route	11350	0	13374.77
45	RouteSegment	9694	0	32228.91
46	Runway	2527	2527	15075.52
47	RunwayCentrelinePoint	2907	0	6998.19
48	RunwayDirection	4429	4431	24829.80
49	RunwayDirectionLightSystem	19	19	117.93
50	RunwayElement	2	2	14.26
51	RunwayMarking	3	0	5.49
52	RunwayProtectArea	3	0	6.77
53	RunwayVisualRange	93	0	144.55
54	SafeAltitudeArea	229	0	826.13
55	SearchRescueService	0	0	1.03
56	SignificantPointInAirspace	788	0	1048.74
57	SpecialDate	811	0	1023.01
58	StandardInstrumentArrival	14	0	21.75
59	StandardInstrumentDeparture	3	0	8.44
60	StandardLevelColumn	4	0	33.92
61	StandardLevelTable	2	0	3.95
62	TACAN	99	0	265.12
63	Taxiway	14	0	19.83
64	TaxiwayElement	7	7	49.45
65	TaxiwayLightSystem	0	0	1.03

66	TaxiwayMarking	0	0	1.03
67	TouchDownLiftOff	234	0	442.17
68	TouchDownLiftOffMarking	13	0	20.88
69	Unit	2624	0	3926.92
70	VerticalStructure	8304	8307	58572.19
71	VisualGlideSlopeIndicator	498	0	653.75
72	VOR	422	0	1068.02

6.2.2 Publication Process

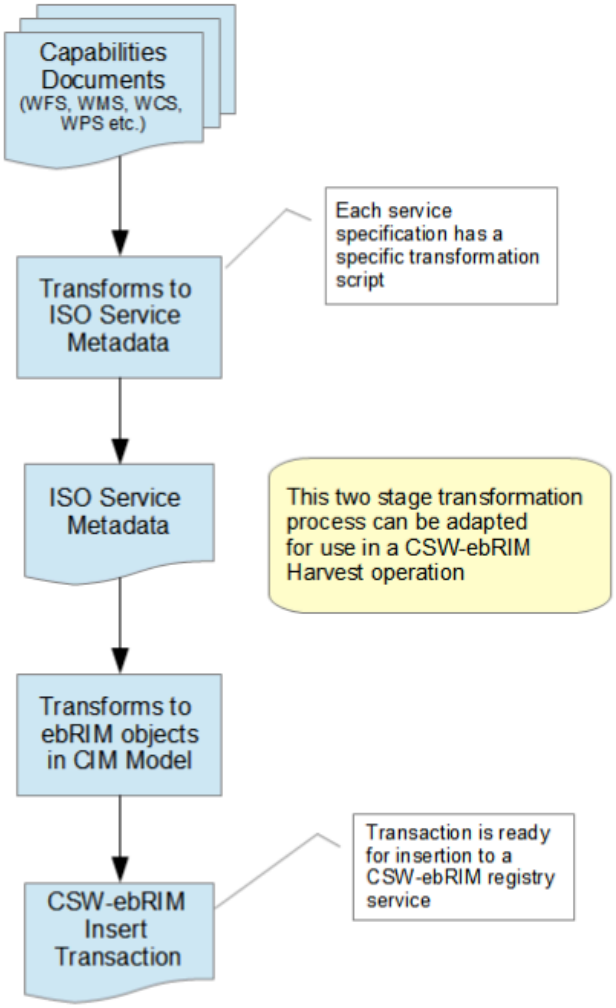
The Harvest operation enables a 'pull' style of registration, whereby ISO 19139 metadata and other resources are retrieved from a remote location and inserted into the registry. Registration of aviation resources can also be pushed into the registry via the standard Insert/Update Transaction operations. Alternatively, an event notification could be sent to a registry service when a registered resource changes, to invoke a pulled harvest update. In OWS-9, the aviation registry implemented the harvest/pull registration of aviation resources.

6.2.2.1 Publishing ISO Service Metadata by Harvesting OWS Capabilities

The service metadata from any OWS can be retrieved or harvested via a GetCapabilities operation (also discussed in the broader context of OWS in Section **Error! Reference source not found.**). The service metadata is then loaded into a CSW-ebRIM registry by transforming the OWS capabilities document response to ISO 19139 service metadata, along with any other metadata useful for the discovery and access/binding to the service.

Starting with the OGC Capabilities document for each OWS type as an input, an XSLT transformation was created to generate the corresponding ISO 19139 service metadata, which gets automatically transformed by the CSW-ebRIM service implementation to the appropriate registry objects in the OGC CIM registry model.

The following diagram illustrates publish/harvest methodology.



Pre-existing associations defined in the CSW-ebRIM standard are also instantiated when service metadata is published, using the CIM registry model as summarized in the following table.

6.2.2.2 Associations related to Service Metadata Objects in CIM

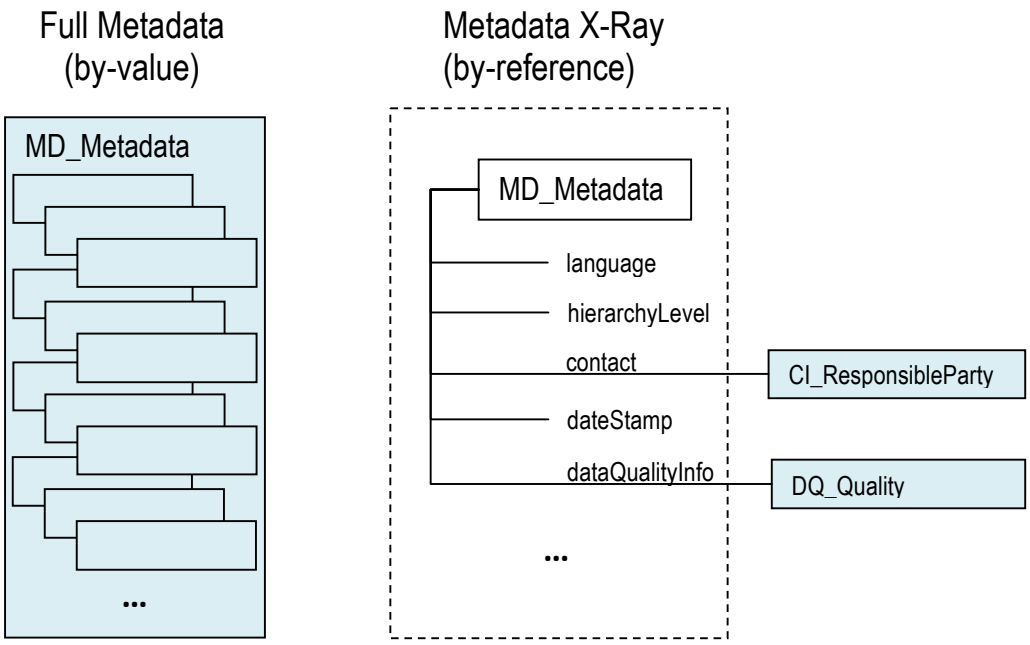
Service Metadata Object Type in CIM	Association Type	Cardinality
ServiceOperation	ContainsOperation	1..*
IdentifiedItem	ResourceReferenceSystem	0..*
Organization	MetadataPointOfContact	0..*
DataMetadata	OperatesOn	0..*
Rights	ResourceConstraints	0..*
Image	GraphicOverview	0..*
MetadataInformation	ResourceMetadataInformation OR ParentMetadataInformation	0..1 (in either case)

7 Results of Selective and Efficient Retrieval of Metadata

The typical problem reported in the aviation threads of past OWS initiatives (Phases 6, 7, 8) was that when a feature's metadata is retrieved encoded as ISO 19139, the full inline encoding of `gmd:MD_Metadata` was retrieved, which is typically a significant portion of the size (~40%) of the feature instance. There is currently no general or common OWS support for the selective and efficient retrieval of ISO 19139 metadata, even though most, if not all, of the metadata was not used by the OWS aviation clients. In order to address the selective and efficient retrieval of metadata, user control over the size of the response should be enabled in the request. Details of the discussion (at a conceptual level) of efficient and selective retrieval of metadata were introduced in the OWS-9 Aviation Metadata & Provenance ER ([10], Section 7). In the Registry ER, efficient and selective retrieval of metadata is demonstrated using OGC CSW-ebRIM registry queries and documented in the following sub-sections 7.1 and 7.2. In Section 7.1, the implementation results of efficiently returning metadata in the 'Metadata X-Ray' encoding are described. In Section 7.2, the implementation results of selectively retrieving sub-components of metadata inline or by-reference are shown, including examples of returning metadata in the 'Data Quality X-Ray' and 'Lineage X-Ray' encodings.

7.1 Efficient Retrieval of Metadata

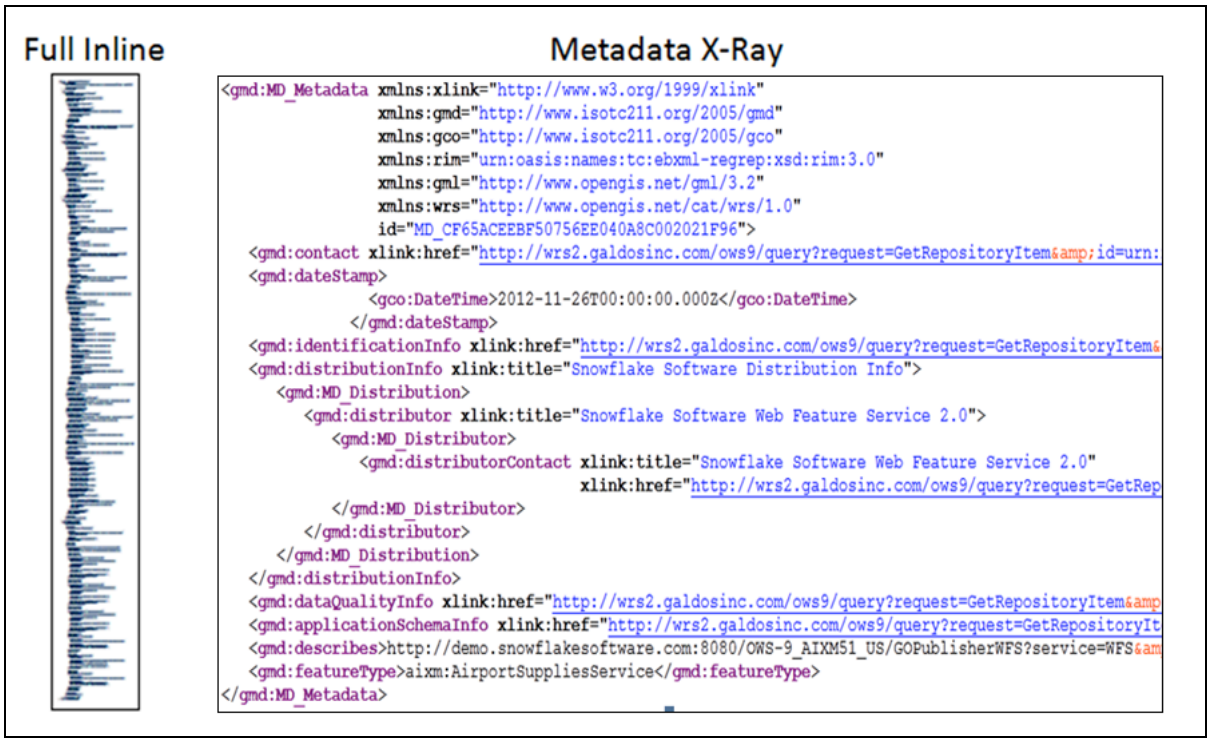
The minimum size of a valid encoding of ISO 19139 metadata occurs when all metadata child elements are encoded by reference – this encoding is referred to as the 'Metadata X-Ray encoding' in the OWS-9 Metadata & Provenance ER ([10]). Note that ISO 19139 allows for many different valid encodings of metadata. An illustrative example of the Metadata X-Ray is as follows:



The relative size comparison of the full inline metadata encoding vs. the Metadata X-Ray encoding is quite significant, the latter being roughly 20 times smaller for a typical aviation metadata instance. The size comparison for typical metadata instances are provided in the following table.

Relative Size Comparison of Full Metadata vs. Metadata X-Ray Encodings			
Entity Counted	Full Inline Metadata	Metadata X-Ray	Reduction Factor
Characters (no spaces)	8481	436	19.5 x
Characters (with spaces)	12275	500	24.6 x
Lines	302	17	17.8 x

A side-by-side comparison of the full inline vs. the Metadata X-Ray encodings of the same metadata instance is illustrated in the following figure.



7.1.1 Implementation Methodology

The full inline encoding of ISO 19139 was first harvested as a repository item in the OGC CSW-ebRIM registry. A basic `GetRepository` item request would return the full inline encoding by default, if a more efficient encoding is not requested. A new parameter, called the `view` parameter was added to the request to further refine the output, i.e. transform the default encoding to an alternate valid response (e.g. the Metadata X-Ray encoding). The value of the `view` parameter is simply the `id` of a registered transformation in the registry, which gets executed on the default response output.

An HTTP GET/KVP query using a `view` parameter has the following form:

http://some.GetRepositoryItem.request**&view=ID**

The ID value above is the repository item id value of the transformation to be used to view the result, in the case of the Metadata X-Ray it is:

urn:x-ows9:def:script:OGC:**MetadataXray**

The registered transformation corresponding to the Metadata X-Ray was an XSLT script (MetadataXRay.xsl) that transforms the `gmd:MD_Metadata` instance in the repository to another valid `gmd:MD_Metadata` instance encoded by-reference everywhere, i.e. with every `xlink:href` attribute value populated by a URI request that points to the content of the child element. The `xlink:href` URI value is set to a registry request that returns the child element. The MetadataXray.xsl transformation is contained in the ouputTransformations.zip archive accompanying this ER

An example request and response are shown in the following sub-sections.

7.1.1.1 Full Inline Metadata Request

The standard GetRepositoryItem request by id returns the full in-line metadata encoding, the default representation.

```
http://wrs2.galdosinc.com/ows9/query
   ?request=GetRepositoryItem
   &id=urn:uuid:8d6f51ad-cbc1-4672-8fea-4ac2c7d1b77c
```

7.1.1.2 Metadata X-Ray Request

A GetRepositoryItem request with the following view parameter returns the Metadata X-Ray encoding (see the following response).

```
http://wrs2.galdosinc.com/ows9/query
   ?request=GetRepositoryItem
   &id=urn:uuid:8d6f51ad-cbc1-4672-8fea-4ac2c7d1b77c
   &view=urn:x-ows9:def:script:OGC:MetadataXray
```

7.1.1.3 Metadata X-Ray Response

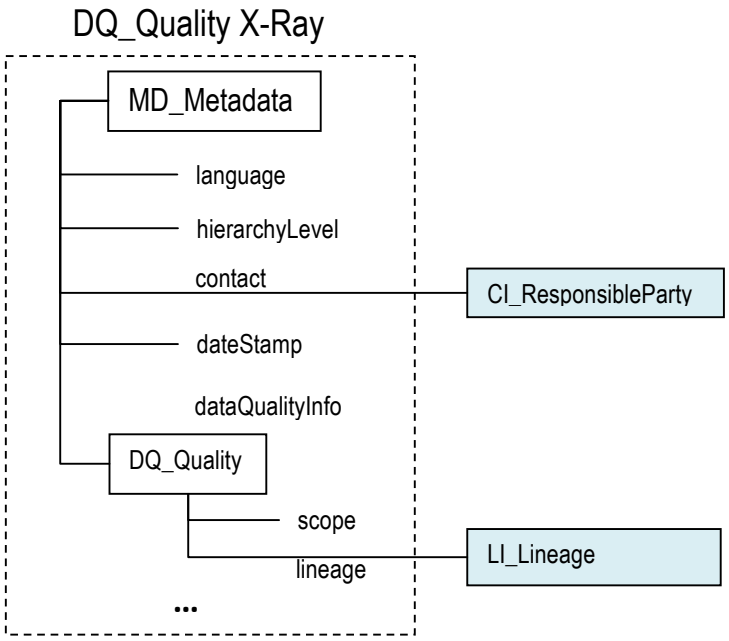
```
<gmd:MD_Metadata xmlns:xlink="http://www.w3.org/1999/xlink"
                 xmlns:gmd="http://www.isotc211.org/2005/gmd"
                 xmlns:gco="http://www.isotc211.org/2005/gco"
                 xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
                 xmlns:gml="http://www.opengis.net/gml/3.2"
                 xmlns:wrs="http://www.opengis.net/cat/wrs/1.0"
                 id="MD_CF65ACEEBF50756EE040A8C002021F96">
   <gmd:contact xlink:href="http://wrs2.galdosinc.com/ows9/query?request=GetRepositoryItem&amp;id=urn:
   <gmd:dateStamp>
              <gco:DateTime>2012-11-26T00:00:00.000Z</gco:DateTime>
            </gmd:dateStamp>
   <gmd:identificationInfo xlink:href="http://wrs2.galdosinc.com/ows9/query?request=GetRepositoryItem&
   <gmd:distributionInfo xlink:title="Snowflake Software Distribution Info">
      <gmd:MD_Distribution>
         <gmd:distributor xlink:title="Snowflake Software Web Feature Service 2.0">
            <gmd:MD_Distributor>
               <gmd:distributorContact xlink:title="Snowflake Software Web Feature Service 2.0"
                                       xlink:href="http://wrs2.galdosinc.com/ows9/query?request=GetRep
            </gmd:MD_Distributor>
         </gmd:distributor>
      </gmd:MD_Distribution>
   </gmd:distributionInfo>
   <gmd:dataQualityInfo xlink:href="http://wrs2.galdosinc.com/ows9/query?request=GetRepositoryItem&amp
   <gmd:applicationSchemaInfo xlink:href="http://wrs2.galdosinc.com/ows9/query?request=GetRepositoryIt
   <gmd:describes>http://demo.snowflakesoftware.com:8080/OWS-9_AIXM51_US/GOPublisherWFS?service=WFS&am
   <gmd:featureType>aixm:AirportSuppliesService</gmd:featureType>
</gmd:MD_Metadata>
```

7.2 Selective Retrieval

To address the *selective* retrieval of metadata, the query should enable the parameter selection of pertinent metadata to be encoded inline, and the remaining metadata compactly encoded with references that a client can resolve in a subsequent step, if desired. Two important categories of selection parameters are: ‘selection by name’ parameters and ‘selection by id type’ parameters.

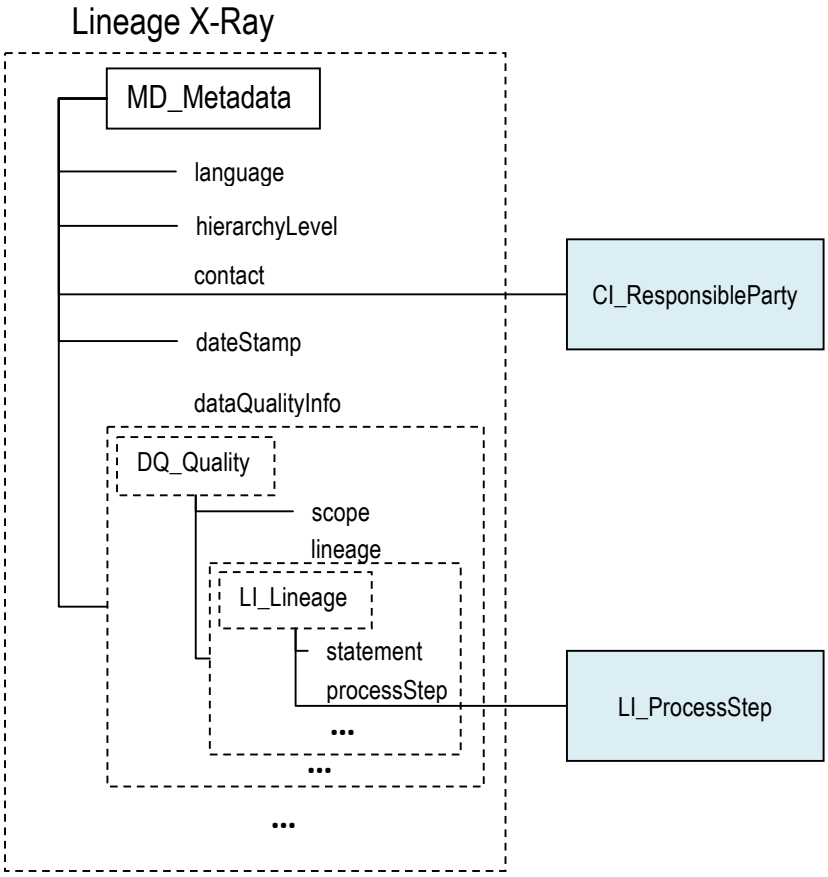
An example of a ‘selection by name’ parameter is illustrated by the DataQuality X-Ray and Lineage X-Ray retrieval mechanisms, where a query returns a result with some selected metadata child objects (e.g. `DQ_DataQuality` and `LI_Lineage`) encoded in-line while others are referenced (introduced in the OWS-9 Metadata & Provenance ER [10]).

The Data Quality X-Ray is the case where ‘selection by name’ parameter is set to the element `gmd:DQ_DataQuality` and is illustrated in the following figure.



See Section 7.2.1.1 for the implementation of the Data Quality X-Ray retrieval.

The Lineage X-Ray is the case where 'selection by name' parameter selects the element names: `gmd:DQ_DataQuality` and `gmd:LI_Lineage` is illustrated in the following figure.



See Section 7.2.1.1 for the implementation of the Lineage X-Ray retrieval.

The 'selection by id type' scheme is to request that all sub-objects in the `MD_Metadata` encoding of metadata that have a populated id value (e.g. the optional `id` and `uuid` attributes in 19139 metadata) to be encoded by-reference; and all objects without an `id`, to be encoded in-line. Note that in OWS-9 a significant number of metadata instances had a mix of objects with and without an `id` attribute. In this case, the metadata author determines which objects get returned in-line by the population of id values on objects.

7.2.1 Implementation Methodology

As described in Section 7.1.1, the CSW-ebRIM GetRepositoryItem request can be used to retrieve ISO 19139 metadata and the `view` parameter identifies a registered transformation to be apply to the output response. A registered transformation, ObjectSkeleton.xsl, selectively encodes, through a parameter input, some element objects inline and some using `xlink:href` with a URI value to a registry request that resolves the object. The ObjectSkeleton.xsl transformation is contained in the ouputTransformations.zip archive accompanying this ER and is invoked by the Lineage

X-Ray representation using a 'selection by name' parameter `ObjectsWantedInline`. The value of the `ObjectsWantedInline` parameter is a list of desired element object names to be encoded in-line. The Lineage X-Ray encoding requires that both the `gmd:DQ_Quality` and `gmd:LI_Lineage` elements are retrieved in-line. The following section 7.2.1.1 shows an example request, where Lineage metadata is selected for in-line retrieval.

7.2.1.1 Data Quality X-Ray Request

In the case that *only* Data Quality metadata is desired to be encoded in-line (i.e. the Data Quality X-Ray representation), the parameter value would be

```
ObjectsWantedInline = 'gmd:DQ_DataQuality'
```

The DataQuality.xsl transformation (in the ouputTransformations.zip archive accompanying this ER) invokes the registered ObjectSkeleton.xsl using the parameter as shown above. A CSW-ebRIM GetRepositoryItem request with the following view parameter returns the Data Quality X-Ray representation.

```
http://wrs2.galdosinc.com/ows9/query
   ?request=GetRepositoryItem
   &id=urn:uuid:8d6f51ad-cbc1-4672-8fea-4ac2c7d1b77c
   &view=urn:x-ows9:def:script:OGC:DataQualityXray
```

7.2.1.2 Data Quality X-Ray Response

The response to the query in 7.2.1.1 is as follows:

```
<MD_Metadata xmlns="http://www.isotc211.org/2005/gmd"
             xmlns:gmd="http://www.isotc211.org/2005/gmd"
             xmlns:srv="http://www.isotc211.org/2005/srv"
             xmlns:gco="http://www.isotc211.org/2005/gco">
   <fileIdentifier>
      <gco:CharacterString>unspecified</gco:CharacterString>
   </fileIdentifier>
   <language>
      <gco:CharacterString>eng</gco:CharacterString>
   </language>
   <hierarchyLevel>
      <MD_ScopeCode codeList="http://www.isotc211.org/2005/resources/Codelist/gmxCodelists.xml#MD_Sc
                    codeListValue="service"/>
   </hierarchyLevel>
   <contact xmlns:xlink="http://www.w3.org/1999/xlink"
            xlink:href="http://wrs2.galdosinc.com/ows9/query?request=GetRepositoryItem&amp;id=urn:x-
   <dateStamp>
      <gco:Date>2012-12-03-08:00</gco:Date>
   </dateStamp>
   <metadataStandardName>
      <gco:CharacterString>OGC Web Feature Service (WFS)</gco:CharacterString>
   </metadataStandardName>
   <metadataStandardVersion>
      <gco:CharacterString>2.0</gco:CharacterString>
   </metadataStandardVersion>
   <identificationInfo xmlns:xlink="http://www.w3.org/1999/xlink"
                       xlink:href="http://wrs2.galdosinc.com/ows9/query?request=GetRepositoryItem&am
   <dataQualityInfo>
      <DQ_DataQuality>
         <scope xmlns:xlink="http://www.w3.org/1999/xlink"
                xlink:href="http://wrs2.galdosinc.com/ows9/query?request=GetRepositoryItem&amp;id=ur
         <lineage xmlns:xlink="http://www.w3.org/1999/xlink"
                  xlink:href="http://wrs2.galdosinc.com/ows9/query?request=GetRepositoryItem&amp;id=
      </DQ_DataQuality>
   </dataQualityInfo>
</MD_Metadata>
```

7.2.1.3 Lineage X-Ray Request

In the case that Lineage metadata is desired to be encoded in-line (i.e. the Lineage X-Ray representation), the parameter value would be

```
ObjectsWantedInline = 'gmd:DQ_DataQuality gmd:LI_Lineage'
```

The LineageXray.xsl transformation (in the ouputTransformations.zip archive accompanying this ER) invokes the registered ObjectSkeleton.xsl using the parameter as shown above. A CSW-ebRIM GetRepositoryItem request with the following view parameter returns the Lineage X-Ray representation.

```
http://wrs2.galdosinc.com/ows9/query
   ?request=GetRepositoryItem
   &id=urn:uuid:8d6f51ad-cbc1-4672-8fea-4ac2c7d1b77c
   &view=urn:x-ows9:def:script:OGC:LineageXray
```

7.2.1.4 Lineage X-Ray Response

The response to the query in 7.2.1.1 is as follows:

```
<MD_Metadata xmlns="http://www.isotc211.org/2005/gmd"
             xmlns:gmd="http://www.isotc211.org/2005/gmd"
             xmlns:srv="http://www.isotc211.org/2005/srv"
             xmlns:gco="http://www.isotc211.org/2005/gco">
   <fileIdentifier>
      <gco:CharacterString>unspecified</gco:CharacterString>
   </fileIdentifier>
   <language>
      <gco:CharacterString>eng</gco:CharacterString>
   </language>
   <hierarchyLevel>
      <MD_ScopeCode codeList="http://www.isotc211.org/2005/resources/Codelist/gmxCodelists.xml#MD_ScopeCode"
                    codeListValue="service"/>
   </hierarchyLevel>
   <contact xmlns:xlink="http://www.w3.org/1999/xlink"
            xlink:href="http://wrs2.galdosinc.com/ows9/query?request=GetRepositoryItem&amp;id=urn:uuid:8d6f51ad-
   <dateStamp>
      <gco:Date>2012-12-03-08:00</gco:Date>
   </dateStamp>
   <metadataStandardName>
      <gco:CharacterString>OGC Web Feature Service (WFS)</gco:CharacterString>
   </metadataStandardName>
   <metadataStandardVersion>
      <gco:CharacterString>2.0</gco:CharacterString>
   </metadataStandardVersion>
   <identificationInfo xmlns:xlink="http://www.w3.org/1999/xlink"
                       xlink:href="http://wrs2.galdosinc.com/ows9/query?request=GetRepositoryItem&amp;id=urn:uui
   <dataQualityInfo>
      <DQ_DataQuality>
         <scope xmlns:xlink="http://www.w3.org/1999/xlink"
                xlink:href="http://wrs2.galdosinc.com/ows9/query?request=GetRepositoryItem&amp;id=urn:uuid:8d6f5
         <lineage>
            <LI_Lineage>
               <statement>
                  <gco:CharacterString>unspecified</gco:CharacterString>
               </statement>
            </LI_Lineage>
         </lineage>
      </DQ_DataQuality>
   </dataQualityInfo>
</MD_Metadata>
```

8 OGC CSW-ebRIM Interface and SESAR Registry Requirements

This section provides a gap analysis between the SESAR Registry Demonstrator and the OGC CSW-ebRIM 1.0.1 registry specification. The purpose of this Gap Analysis is to determine how well aligned these two registry service specifications are currently. The analysis considered various aspects of these registry services, such as: supported entities and relationships types, available service methods, user roles, as well as processing (use cases) and security features.

The primary aspects of the registry services that will be reviewed here are summarized in the following table.

SESAR Registry Service Aspect	Description	CSW-ebRIM Equivalent
Entities	The primary model object-types; typically specific to a domain	RegistryObjects, or ExtrinsicObjects (for extended object types)
Relationships	The connections within the domain model between Entities (or RegistryObjects)	Associations
User Roles	A role identifies a class of user functionality; typically associated with specific privileges, or policies, that help define the scope of the Roles capability	no significant difference
Service Methods	A web service operation by which registry client software can interact with the service	no significant difference
Data Management	Registry functonality related to management of resources	no significant difference
Security	Registry functionality in support of security	no significant difference

8.1 Overview of the Candidate Registries

The SESAR Registry Demonstrator, also called the SWIM Registry, is described in the "SWIM Registry Concept of Operations V1" as follows:

> *The SWIM registry aims at improving the visibility and accessibility of ATM information and services available through SWIM. This enables service providers, consumers, and regulatory authorities to share a common view on SWIM.*
>
> *The SWIM registry is a one stop shop for service information. It describes the complete set of services enabled by SWIM (like the service that allows getting the reference business trajectory for a flight) with qualitative, consolidated and structured information. The registry is also the source of reference for other service related information such as standards, policies and certifications.*
>
> *The SWIM registry will enable direct ATM business benefits to all of its stakeholders by:*
>
> - *Allowing providers (mainly those sharing information over SWIM) to increase visibility (and consequent adoption) of their services. It will also support them in managing their relationship with consumers and managing dependencies with other services, standards or regulations.*

- *Improving the efficiency of consumers (mainly those getting information from other stakeholders over SWIM) in identifying the right provider and reducing their effort in setting up everything required prior to start using a service.*
- *Facilitating a collaborative evolution of services by enabling all relevant stakeholders to share a common view and participate in the lifecycle of these.*

By comparison, the CSW-ebRIM registry is described in "CSW-ebRIM Registry Service - Part 1: ebRIM profile of CSW" as:

> *CSW-ebRIM Registry Service: a profile of the CSW part (Clause 10) of the OpenGIS® Catalogue Service Implementation Specification (v2.0.2, OGC-07-006r1). It applies the CSW interfaces to the OASIS ebXML registry information model (ebRIM 3.0) so as to provide a general and flexible web-based registry service that enables users—human or software agents—to locate, access, and make use of resources in an open, distributed system; it provides facilities for retrieving, storing, and managing many kinds of resource descriptions.*

8.2 Gap Analysis

The following sections provide the tabulated results of the gap analysis. The shaded non-header rows indicate a gap between the SESAR registry requirements and support by the CSW-ebRIM interface standard. A discussion of the gaps is provided in section 8.4.

8.2.1 Entities

SESAR Entity Name	Description	CSW-ebRIM Equivalent Object-Type	Remarks
Service	A descriptive object that specifies the service interface, endpoints, contracts and other service related information.	**Service** object in conjunction with its related sub-types: **ServiceBinding** and **SpecificationLink**.	Additional object-types may be defined and linked, as required, to the Service object via **Associations**.
Standard	These assets can be terms (that provide a common understanding of the different concepts used), data models, and service descriptions (for interoperability, compatibility etc.).	Insert a new "**Standard**" object-type to act as parent to a collection of sub-types which each represent a different type of "Standard" asset. This allows multiple concrete "Standard" sub-types to be modeled within the domain concept of "Standard".	
Policy	Prescribes the conditions and constraints for interacting with a service.	Insert a new "**Policy**" object-type. If required, treat "Policy" as the parent of a collection of "Policy" sub-types.	
Certification	The product of a regulation authority that	Insert a new "**Certification**" object-type. If required, treat	

	describes a level of conformity towards certain criteria and the assessment process to measure it. More concretely, a certification is a set of criteria and a process. I.e. IATA Operational Safety Audit (IOSA) certification.	"Certification" as the parent of a collection of "Certification" sub-**types**.	
Category	Enables the structuring of information based on a common classification structure that facilitates the discovery and management of information in the registry.	Maps directly to the ebRIM **ClassificationScheme** concept which is a tree of **ClassificationNodes**. Each ClassificationNode defines a "category". ClassificationNodes are used in ebRIM to classify any RegistryObject (multiple times, as required).	CSW-ebRIM supports the definition of unlimited schemes which may be flat (1 level deep) or deeply nested, as needed.
Participant	A support entity used to generically refer to all actors relevant to the system: Systems (software agents), Persons, Users (authenticated persons) and Orgranizations.	Use the ebRIM object-types: **Person**, **User** and **Organization**. Users are typically authenticated Persons in ebRIM. Where it is necessary to identify specific sub-groups of Persons, they can be segmented by Classification. A RegistryObject may classified multiple times via the addtion of Classification object.	**Service** object-type can be used to identify software agents, if required.

8.2.2 Relationships

SESAR Registry Relationship Name	Description	CSW-ebRIM Equivalent Association-Type	Remarks
Service Consumption / Provision	This enables to maintain the list of providers and consumers for the different services. This facilitates change management as it is maintained	Define two separate association-types: **Consumes** and **Provides**.	

	the dependency of consumers to a given service, and provides visibility of the services offered by the providers.		
Service Standardization	This establishes a relationship between a service and a particular standard that it uses. This enables service providers to manage their alignment to existing standards and stay informed on the evolution of these standards.	Use existing ebRIM association-type: **Uses**	
Service Dependency	This establishes a relationship between a service and another service that is based on. This enables service providers to efficiently manage service reusability.	Define new association-type: **HasDependency**	
Service Policy Compliance	This establishes a relationship between a service and a particular policy that it complies with. This enables service providers to align to existing policies and stay informed on the evolution of these policies.	Define new association-type: **CompliesWith**	
Certification	These are the records that proof that a particular service or participant has been assessed as conformant to a certification based on the certification assessment done by	Define new association-type: **HasCertification**	

	certification provider.		
Certification Assessment Provision	This enables to maintain the list of certification providers that are entitled to provide certifications (run certification assessments).	Define new association-type: **CertificationProvider**	
Subscription	These are requests from a participant to get notifications for any update performed on a service, standard, policy, or certification	Define a new association-type: **HasSubscription** in conjuction with the core ebRIM object-types: **Subscription** and **Notification**	**Note:** Subscription procedures are not specified nor are they precluded by CSW-ebRIM. The extensibility of ebRIM can support the information objects that could enable the implementation of any subscription procedure. Some registry implementations provide support for subscription.. .
Classification	This represents the specific relationship between a particular set of information (e.g. a service instance) and a particular category.	Use existing ebRIM object-type: **Classification**	A RIM Classification acts like a specialized Association by linking RegistryObjects to ClassificationNodes (e.g. categories). A Classification defines its own links and does not require any additional Associations.

8.2.3 User Roles

The CSW-ebRIM specification does not specify a particular identity management approach for registry implementations, nor does it define any User Roles.

The minimal requirement for CSW-ebRIM registries would be to integrate with a standard identity management component (e.g. for example, an LDAP service) that provides User and Role information and to support its provisioning to the configured Access-Control service. The Access-Control service would perform the policy decision evaluation (e.g. act as XACML PDP) while the policy enforcement (e.g. act as XACML PEP) would need to be performed by the registry sevice, possibly at multiple different processing points.

Therefore the integration requirement for supporting multiple new User Roles is quite low. The primary effort for such integration would be in Policy definition to ensure that each Role was appropriately constrained.

8.2.3.1 High Level SESAR Roles:

SESAR Registry User Role Name	Description
Service Consumer	It is mainly interested in finding information about services in the registry. More specifically it discovers and subscribes to follow the evolution of services in the registry. It is also able to register via the registry for the consumption of services.
Service Provider	It is mainly interested in sharing with other registry users its service offer. It is also able to manage the service lifecycle in the registry including the publication of different versions of a service, the management of dependencies with other services or regulations (e.g. standards, policies…) as well as the management of its different consumers.
Regulation Authority	It regulates the service network with the publication of standards, policies, certifications and categories. It is able to control and enforce via the registry the compliance to these regulations. From the registry point of view, this is just another user of the registry, but from a SWIM perspective this role uses the registry to manage the governance of the SWIM environment.
Certification Provider	It provides the service of certification assessments. This role is granted by a regulation authority responsible for the certification specification. The certification provider conducts assessments (certification provision) of the compliancy of participants or services to certain certifications.
Identity Authority	It is the provider of identities for all users that participate in the service network and this includes the users that interact directly with the registry.
Registry Authority	Equivalent to the role of the regulation authority that is aiming to regulate the services of the network, the registry authority issues standards, policies and certifications for the regulation of the registry.
Registry Manager	This role is responsible for the operation of the registry as well as for the alignment to the regulations set by the regulation authority

8.2.4 Service Methods

The SESAR Registry defines multiple similar Service methods that are specific to an Entity type. The getXXX methods typically differ in method-name and filter criteria. The putXXX methods differ in method-name and input-type parameters. In CSW-ebRIM such groups of similar methods typically map to a single service method that operates on all of the specified Entity types, thus reducing the large number of methods used in SESAR Registry to a much smaller, more manageable number.

8.2.4.1 Common Service Methods

The table below addresses sets of common service methods defined by the SESAR Registry by grouping them and treating each group as a ***class*** of operations with similar behaviour. A description of equivalent CSW-ebRIM behaviour is provided.

SESAR Registry Service Method	Input / Output	CSW-ebRIM Equivalent Functionality	Remarks
getXXXList Eg: getServiceList, getStandardList, getPolicyList	Filter criteria / Entity collection	**GetRecords** request using OGC Filter Encoding; response is RegistryObject collection of specified object-type	
getXXX Eg: getService, getStandard, getCertification	Entity ID plus optional scope (e.g. filter criteria) / Entity instance	Either: **GetRecords** (as above) if scope specfied, Or: **GetRecordById** request, when only ID is specified.	
putXXX Eg: putService, putCertification, putCategory	Entity instance / response code	**Insert**, **Update** or **Delete** transaction	
putXXX[**PropertyName**] Eg: PropertyNames like: Ownership, Keyword, Contact, Approvers, Approver List, Validity Period, Scope, Category, Role	Entity ID and Property value / response code	**Update** transaction	Partial update may be used where a particular registry product has implemented this feature.
putXXXSubscription Eg: putServiceSubscription, putPolicySubscription	Entity ID –OR– Category / response code **Note:** There is an additional parameter to the subscription request which is **implicit** – the **UserId** which would	**Insert**, **Update** or **Delete** transaction to {define, update, remove} a **Subscription** object for the given RegistryObject (or Entity) type and the active **User**.	CSW-ebRIM does not require nor preclude an implementation from addressing the Subscription / Notification requirements. The extensibility of ebRIM can define the information objects that could support the enforcement of any subscription procedure.

	be available to the regsitry service within the request context once the User has been authenticated.		
putServiceLifecycleStatus **Note:** the only Entity within SESAR, that has a Life-cycle status property, is Service. In CSW-ebRIM, all RegistryObjects have a status property.	Service ID / life-cycle status	Use **LifeCycleManager Update** transaction to set a new status value for a Service instance.	
putXXXApproval Eg: Service, Standard, Policy, Certification, Category	Entity ID, User ID / response code	**LifeCycle Update** to status "**Approved**" for the Entity, coupled with an **Insert** of an **Association** between the Entity and the User to record the granting of the approval.	.
putXXXAccessibilityRequest Eg: Service, Standard, Policy, Certification	Entity ID, User ID / response code	**Insert** an Association between the Entity and the specified Particpant. **Note:** in CSW-ebRIM, all inserted RegistryObject have their status set to **Submitted**. This initial status can be used to indicate a request for approval. Subsequently, approval can be granted via application of **Life-cycle Update** to change the status of the Association to **Approved**. See also: **putXXXAccessibilityList**	There is a conceptual gap here, because SESAR does not define a concept of an Entity of type 'request'. However, this method can be readily accommodated in CSW-ebRIM by modeling requests AND grants of accessibility as different states of an Association between an Entity and a Participant.
putXXXAccessibilityList Eg: Service, Standard, Policy, Certification	Entity ID, Participants List / response code	**Insert** Associations for new Participants; perform Life-cycle **Update** to the status **Approved** for all Associations (new or pre-existing) to a specified Particpant.	This service request is coupled with Accessibility Request, above, to provide a Participant access to an Entity instance. Recommend addition of **putAccessibilityApproval**

			to allow adding a single Participant to the list, rather than require an update to the entire list.

8.2.4.2 Entity Specific Methods for Service

SESAR Registry Service Method	Input / Output	CSW-ebRIM Equivalent Functionality	Remarks
putServiceConsumption Request	User ID, Service Contract ID / Service Consumption ID	**Insert**, **Update** or **Delete** transaction for managing a **Consumes** Association between User and a Service entity. **Note:** in CSW-ebRIM, all inserted RegistryObject have their status set to **Submitted**. This initial status can be used to indicate a request for Service Consumption Approval. Subsequently, approval can be granted via application of **Life-cycle Update** to change the status of the Association to **Approved**. See also: **putServiceConsumption Approval**	SESAR does not define a "**consumption**" Entity type or resource. This concept could be modeled as the **Consumes** Association object which links a **User** object to the **Service** of which it is a consumer.
mputServiceConsumption Approval	User ID, Service Contract ID / Service Consumption ID	**Insert**, **Update** or **Delete** the **Consumes** Association that links the User with the Service object; perform Life-cycle **Update** to change the status to **Approved**.	
getServiceConsumption List	Service ID, Service Contract ID / List of service consumption registrations	Selects Users via **GetRecords** request with filtering by Service or Service Contract ID and by traversing the **Consumes** Association. Return type is a list of **User** objects.	SESAR does not define a "**consumption registration**" Entity type or resource. As above, this concept will be modeled as the

			Consumes Association object.

8.2.4.3 Entity Specific Methods for: Standard

There are no specific methods for Entity type "Standard". However, the common service methods provide complete coverage for this entity-type.

8.2.4.4 Entity Specific Methods for: Policy

SESAR Registry Service Method	Input / Output	CSW-ebRIM Equivalent Functionality	Remarks
putPolicyApplicabilityScope	Policy ID, Category ID / response code	**Update** transaction to add the Category to the Policy instance	
putPolicyAssertionAutomation	Policy ID OR Validation script / response code	Validation script would be stored as a **repository-item** related to its Policy.	SESAR does not define a "validation script" Entity type or a resource. In CSW-ebRIM, some RegistryObjects can have related resources, called **repository-items**.
getPolicyAssertionAutomation	Policy ID / Validation script	Use **GetRepositoryItem** request where Validation scripts are **repository-items** related to a Policy instance.	CSW-ebRIM **repository-items** are digital resources related to a RegisryObject and they may be of any mime-type (binary, text, image, script, document etc.). A RegistryObject can be related to zero, or exactly one, **repository-item**.
getPolicyAssertionServiceList Validation	Service List, Category ID / Service list	First, requires selection via **GetRecords** request with filtering by Category. Then selected Services are "validated" and those	

		that pass are returned in the response.	

8.2.4.5 Entity Specific Methods for: Certification

SESAR Registry Service Method	Input / Output	CSW-ebRIM Equivalent Functionality	Remarks
putCertificationProvider	User ID, Certification ID / response code	**Insert**, **Update** or **Delete** transaction for managing Association between User and Certification entity.	
putCertificationRequest	User ID, Certification ID, CertificateType = {Participant | Service}, Object = {User ID | Service ID} / response code	**Insert** an Association between the Certification and the object to be certified (User or Service). **Note:** in CSW-ebRIM, all inserted RegistryObject have their status set to **Submitted**. This initial status can be used to indicate a request for certification. Subsequently, certification can be granted via application of **Life-cycle Update** to change the status of the Association to **Approved**. See also: **putCertificate**	SESAR does not define a "request" Entity type or resource
putCertificate	User ID, Certification ID, CertificateType = {Participant | Service}, Object = {User ID | Service ID}/Certificate ID	**Insert**, **Update** or **Delete** the Association that links the **Certification** with the certified object (User or Service); attach all necessary **Certificate** properties to the **Association**; perform Life-cycle **Update** to change the status to **Approved**. **Note:** if the "Certificate" concept is more than a simple relationship, this approach may need to be revisited. There may be a need to define a **Certificate** entity to properly model the concept. However, if only a few properties need to be captured to represent the **Certificate**, these can be added to the Association, as in CSW-ebRIM all RegistryObjects	SESAR does not define a "certificate" Entity type or resource. Combined with above "request" method to form Request / Approval pair.

		have an extensible set of properties. See also: **putCertificationRequest**	

8.2.4.6 Entity Specific Methods for: Category

SESAR Registry Service Method	Input / Output	CSW-ebRIM Equivalent Functionality	Remarks
putCategoryChild	Child Class ID, Parent Class ID / response code	**Insert**, **Update** or **Delete** transaction for adding Categories to a taxonomy, or **ClassificationScheme**.	
putCategoryMembershipRequest	category ID, {User ID | Service ID } / response code	**Insert** an **Classification** between the **Category** and the specified **Entity** (User or Service). **Note:** in CSW-ebRIM, all inserted RegistryObject have their status set to **Submitted**. This initial status can be used to indicate a request for approval. Subsequently, approval can be granted via application of **Life-cycle Update** to change the status of the **Classification** to **Approved**. See also: **putCategoryMembership Approval**	SESAR does not define a "request" Entity type or resource. However, this method can be readily accommodated in CSW-ebRIM by modeling requests AND approvals of categorization as different states of a **Classification** object which relates a **Category**, or **ClassificationNode**, with a **User** or **Service**.
putCategoryMembershipApproval	category ID, {User ID | Service ID } / response code	Use **Life-cycle Update** to change the status of the **Classification** to **Approved**. See also: **putCategoryMembershipRequest**	SESAR does not define an "approval" Entity type or resource. Combined with above "request" method to form Request / Approval pair.

8.2.4.7 Entity Specific Methods for: Participant

There are no specific methods for Entity type "Participant". The common sevice methods provide complete coverage for this entity-type.

8.3 Use Cases

These SESAR Registry Use Cases summaries are extracted from section 6, "Processes (use cases)", in [08.03.02.D03].

SESAR Registry Use Case	Description	CSW-ebRIM Registry Level of Support
Service Information Publication and Discovery (Design Time) 6.2	The registry as the source of reference for service information. **Mandatory** business process for SWIM.	Fully supported
Service regulation via policies and standards 6.3	The registry as the source of reference for policies and standards. **Mandatory** business process for SWIM.	Fully supported
Service Information quality management via categories and certifications 6.4	The registry as the source of reference for certifications and promotes the usage of categories in order to support service information quality. The registry as an enabler of the following two mechanisms: 1) Categories that enable the consistent structuring of service information and 2) Certifications that enable the validation, by a third party, of service information quality and conformance to regulations. **Mandatory** business process for SWIM.	Fully supported
Service Lifecycle management 6.5	The registry as the consolidated location for service lifecycle management. Also as the location for service publication and discovery, management of service versioning and dependencies, registration of consumers to services, and the classification and certification of services via the registry. **Recommended** business process for	

	SWIM.	
Management of information integrity in the registry 6.6	The registry as an enabler of service information integrity, with controls in place to manage a multi-user shared repository. This implements a consolidated access control to service information. **Mandatory** business process for SWIM.	CSW-ebRIM does not require nor preclude that an implementation be a Policy Enforcement Point (PEP) for access control policies. The extensibility of ebRIM can define the information objects that could support the implementation of any access control system.
Management of information confidentiality in the registry 6.7	The registry as an enabler of service information confidentiality, with controls in place to manage a multi-user shared repository. This implements a consolidated access control to service information. **Mandatory** business process for SWIM.	Same as with 6.6 (above)
Delegated run-time policy enforcement managed at the registry 6.8	This use case considers the registry as the source of reference for policy information for those policy agents responsible for the runtime enforcement (e.g. application firewalls). This enables a consolidated management and control of policy enforcement information. This requires a high availability of the registry for policy and service information. **Optional** business process for SWIM.	Supported. High availability is not addressed by the CSW-ebRIM specification and will depend on the registry implementation.
Run-time service discovery 6.9	This use case considers the registry as a provider of run-time service information. This enables a loose coupling with services and improves flexibility. This requires a high availability of the registry for service information. **Optional** business process for SWIM.	Supported. High availability is not addressed by the CSW-ebRIM specification and will depend on the registry implementation.

8.3.1 Security

This section addresses the security requirements for the registry from two different perspectives:

- The role of the *registry in support of the SWIM security* framework. In other words, the supporting role of the registry for the implementation of security services in SWIM.
- The requirements to *secure registry held content and access to operations*. This describes what is needed to protect the confidentiality, availability and integrity of the registry (i.e. data and the services that the registry provides).

8.3.1.1 Registry in Support of SWIM Security

Here, the registry plays a supporting role for the implementation of security services in SWIM (e.g. role of the registry in SWIM policy enforcement is to provide policies).

SESAR Registry Security Requirement	Registry Role	CSW-ebRIM Level of Support
Authorization of communications to legitimate service endpoints	Support (e.g. provide policies, user role information etc.)	Fully supported
Authorization of service consumptions to registered service consumers	Support (e.g. provide consumption approval details etc.)	Fully supported
Enforcement of security policies	Support (e.g. provide policies etc.)	Fully supported

8.3.2 Registry Service Security

Here, the registry must provide specific functionality to protect its data content and services.

SESAR Security Aspect	Requirement	CSW-ebRIM Level of Support	Remarks
Confidentiality	The accessibility scope (*public*, *restricted* or *private*) of registry information shall be managed by its owner.	Possible to support. Requires suitable access-control policies	Model accessibility scope as a property of specific entities and write access-control policies accordingly.
	Information owner shall manage the list of users/groups that have access to restricted information.	Possible to support. Requires suitable access-control policies	
Integrity	Information stored in the registry with high integrity requirements shall be digitally signed by the publisher.	Possible to support in the implementation, but outside the scope of the CSW-ebRIM interface standard	External requirement (publisher signing), unless registry involved in verification. Feature may be offered by specific registry implementations.
	Integrity checks on the data stored in the registry shall be performed and exceptions reported.	Possible to support in the implementation, but outside the scope of the CSW-ebRIM interface standard	Feature may be offered by specific registry implementations.
Availability	The registry shall be implemented with the redundancy provided by multiple instances in an active-active configuration.	Possible to support in the implementation, but outside the scope of the CSW-ebRIM interface standard	Feature may be offered by specific registry implementations.
	The registry shall provide failure transparency by masking to a service consumer the failure and possible recovery of one of its instances.	Possible to support in the implementation, but outside the scope of the CSW-ebRIM interface standard	Feature may be offered by specific registry implementations.
Authenticity and Non-Repudiation	Information stored in the registry with high integrity requirements shall be signed by the publisher.	Possible to support in the implementation, but outside the scope of the CSW-ebRIM interface standard	External requirement (publisher signing), unless registry involved in verification. Feature may be offered by specific registry implementations.
	All updates done in the registry will be logged.	Possible to support in the implementation, but outside the scope of the CSW-ebRIM interface standard	Feature may be offered by specific registry implementations.

8.4 Addressing the Gaps – Implementing a SESAR Registry with the OGC CSW-ebRIM Interface

The gaps between the scope of the OGC CSW-ebRIM interface and the SESAR registry requirements fall into two categories: subscription management and access control enforcement, neither of which is prohibited by CSW-ebRIM. The CSW-ebRIM interface standard focuses on broader interoperability of the publication and retrieval of resources by query/filter over the web. The CSW-ebRIM standard does not specify any type of subscription (or registration) procedure and does not preclude any of the choices for specific implementations of such procedures. CSW-ebRIM also does not require nor preclude that an implementation be a Policy Enforcement Point (PEP) for access control policies. Alternatively a CSW-ebRIM implementation could participate in a network portal where access control is enforced external to the registry at a single sign-on point of the portal. The extensibility of ebRIM can define the information objects to support the implementation of any subscription or access control system.

8.5 SESAR Registry Demonstrator

The starting point for loading service metadata in the SESAR Registry Demonstrator was to retrieve/create Web Service Description Language (WSDL) documents for each of the service resources. This is one of the differences with respect to loading service metadata compared to the CSW-ebRIM aviation registry – a combination of OWS Capabilities and WSDL documents were used to harvest metadata from the latter. In the case where a service resource (e.g. Snowflake Software WFS 2.0) did not publish a WSDL document, but rather the OWS required Capabilities document, a WSDL document needed to be generated by some means to load the metadata into the SESAR Registry Demonstrator.

In OWS-9, several services did publish the required WSDL documents, but not all. In OWS-9, the WSDL documents that were produced by service providers, did not follow any particular implementation standard, ranged widely with respect to organization and level of completeness, and most did not pass schema validation. In the absence of WSDL documents, the information in the OWS Capabilities documents was used to generate the WSDL. Note that the ISO Service metadata could also have been used to generate WSDL documents and this could be explored further as part of the future work item on standardizing WSDL for OWS proposed in Section 1.4.1. The WSDL documents that were created from OWS Capabilities documents were hand crafted, however it was noted that it would be possible to develop and automate the process to generate WSDL in a consistent and comprehensive manner with respect to both organization and level of completeness. In section 6.2.2 (Publication Process), it was described how ISO 19119/19139 Service Metadata was automatically generated from OWS Capabilities Documents to improve OWS interoperability of discovery applications. However due to the lack of standardized structure of WSDL, it is not possible to properly automate the

creation of ISO Service Metadata from WSDL documents as was done for OWS Capabilities. The proposed future work items listed in section 6.2.2 will go a long way to improving the interoperability of WSDL-capable applications in OWS environments. This list includes: developing an OWS/ISO profile of the WSDL standard, developing best practices for the creation/transformation of WSDL documents in/to the OWS/ISO profile of WSDL, and automatic generation of OWS/ISO profiled WSDL directly from OWS Capabilities documents and ISO 19139 Service Metadata. As a starting point the following WSDL structure for OWS is proposed:

WSDL Definition Structure
- Types (required)
  - Schemas (required)
- Messages (optional)
- Port Type (required)
- Binding (required)
  - Operations (required)
- Service (required)

The following services were harvested into the SESAR registry demonstrator using the input WSDL documents:

Service Resources Harvested in OWS-9		
Service Name		Service Description
1	52North SES 1.0	SES at 52North, Muenster, Germany
2	52North WPS 3.1-SNAPSHOT	Service based on the 52North implementation of WPS 1.0.0
3	ATM-TGS Data Management Service	ATM-TGS OWS-9 Implementation of Data Management Service
4	ATM-TGS Dispatch DMS	ATM-TGS OWS-9 Implementation of Dispatch Data Management Service
5	ATM-TGS Ground DMS	ATM-TGS OWS-9 Implementation of Ground Data Management Service
6	Envitia ChartLink WMS	Envitia WMS generated from a published ChartLink project
7	COMSOFT CADAS-AIMDB WFS 2.0	COMSOFT CADAS-AIMDB Implementation of WFS 2.0 for OWS-9 Initiative
8	Galdos INdicio Aviation Web Registry Service (OGC CSW-ebRIM)	This is a CSW-ebRIM web service deployed for use in the OGC Web Service (OWS) interoperability program, Phase 9.
9	IDS OGC SES Broker	IDS OWS-9 Implementation of OGC Sensor Event Service Broker
10	IDS OGC SES Create Pull Point	IDS OWS-9 Implementation of OGC Sensor Event Service Create Pull Point
11	IDS OGC SES Notification Broker	IDS OWS-9 Implementation of OGC Sensor Event Service Notification Broker
12	IDS OGC SES Pausable Subscription Manager	IDS OWS-9 Implementation of OGC Sensor Event Service Pausable Subscription Manager

13	IDS OGC SES Publisher Registration Manager	IDS OWS-9 Implementation of OGC Sensor Event Service Publisher Registration Manager
14	IDS OGC SES Pull Point	IDS OWS-9 Implementation of OGC Sensor Event Service Pull Point
15	ifGI OGC SES Service Broker	ifGI OWS-9 Implementation of OGC Sensor Event Service Broker
16	ifGI OGC SES Publisher Registration Manager	ifGI OWS-9 Implementation of OGC Sensor Event Service Publisher Registration Manager
17	ifGI OGC SES Subscription Manager	ifGI OWS-9 Implementation of OGC Sensor Event Service Subscription Manager
18	Luciad Lightspeed FPS	OGC Feature Portrayal Service implementation powered by Luciad Lightspeed
19	Luciad WPS	Luciad Web Processing Server
20	LuciadFusion Tile Store	LuciadFusion Tile Store
21	Snowflake AIXM 5.1 EUROPE Demonstrator WFS	The OWS-9 AIXM 5.1 Demonstrator WFS provides access to a wide range of AIXM 5.1 features for the EUROPE sector. DISCLAIMER: This data should be used for research and development purposes only. It is not suitable for operational purposes.

