Open Geospatial Consortium Inc.

Date: 2004-08-02

OpenGIS® Reference number: **04-051**

Version: 0.5

Category: OpenGIS® Interoperability Program Report

Editor: George Percivall, Open Geospatial Consortium, Inc.

OWS1.2 Image Handling Design

Copyright notice

Warning

This document is not an OGC Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard.

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

Document type:	OpenGIS® Discussion Paper
Document subtype:	Interoperability Program Report – OWS1.2
Document stage:	Publicly Available
Document language:	English

Contents

Figures

i. Preface

This document was developed as part of the Image Handling Thread of OGC Web Services 1.2.

ii. Submitting organizations

The following organizations submitted this document to the Open GIS Consortium Inc.:

- NASA Geospatial Interoperability Office

iii. Document Contributor Contact Points

All questions regarding this submission should be directed to the editor or the submitters:

CONTACT	COMPANY	ADDRESS	PHONE/FAX	EMAIL
George Percivall	NASA / GST, Inc.	NASA Goddard Space Flight Center, Code 900.2	1+301-286-4073	percivall@gsfc.nasa.gov
Arliss Whiteside	BAE Systems Mission Solutions	10920 Technology Place, San Diego, CA 92127-1874	858-592-1608	Arliss.whiteside@baesystems.com
John Evans	OGC / GST, Inc.	7855 Walker Drive, Suite 200 Greenbelt, MD 20771 USA	301-474-9696	evans@gst.com
Stephane Fellah	PCI Geomatics	490 rue St. Joseph, Suite 400 Hull, Quebec J8Y 3Y7 CANADA	819-770-0022 x223	Fellah@pcigeomatics.com
John Vincent	Intergraph Mapping and GIS Solutions	[To be supplied]	256-730-7767	jtvincen@intergraph.com
Liping Di	George Mason University	9801 Greenbelt Road, Suite 316-317, Lanham MD 20706 USA	301-552-9496	lpd@rattler.gsfc.nasa.gov

iv. Revision history

Date	Release	Author	Paragraph modified	Description
2002-08-13	0.0.1	Percivall	All	Initial outline of document
2002-08-20	0.1	Percivall	All	Initial draft of document
2002-9-30	0.2	Percivall	Multiple	Interim version with comments received from OWS1.2 participants and during September TC meeting.
2002-12-23	0.3	Percivall	Multiple	Revised based on OWS 1.2 participant comments; Revised to be consistent with other OWS 1.2 DIPRs; Added section on service composition consistent with OWS Common Architecture; Update Image Archive interfaces based on consensus of OWS 1.2 Image Handling group Added definitions for key terms. Design issues formatted for review.
2003-1-20	0.4	Percivall	Multiple	Incorporated comments on 0.3 from OWS1.2 participants. Final editing as an IPR for February 2003 OGC TC meeting.
2004-8-2	0.5 04-051	Percivall; based on comments by J. Evans	Multiple	Changes to make this paper consistent with Web Coverage Service, 1.0 which was released between v0.4 and v0.5: • Updated ref. [2] for WCS • Changed "describeCoverageType" to "describeCoverage" to match the WCS 1.0 • Added note regarding operation name (Figs. 2 and 7; Table 1); • Removed the first Design Issue in 6.2.1 (WCS Interface).

v. Changes to the OpenGIS® Abstract Specification

The OpenGIS© Abstract Specification may require changes to accommodate the technical contents of this document.

The following is a list of the required changes:

- Reuse of interfaces needs to be clarified in OGC AS Topic 12,
- Review of operations defined in OGC AS Topic 6 - The Coverage Type

vi. Future Work

In several places the document lists “Design Issues.” These are highlighted in a box and in italics. Some design issues will only be resolved based on enterprise policies, e.g., metadata management, others issues are anticipated to be resolved in concurrent or future OGC activities.

Foreword

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

The Image Handling Design IPR is prepared as part of OWS1.2

Introduction

Image Handling is a thread in the OGC Web Services 1.2 (OWS1.2). This document defines the system design for Image Handling in OWS1.2. The system design responds to the requirements in [1][1]. The system design specifies two main services: Image Archive Service and Image Catalogue Service. Interfaces for these two services are defined using previously defined OWS service interfaces.

[1] References are listed in Clause 3.

OWS1.2 Image Handling Design

1 Scope

The OWS1.2 Image Handling Design responds to the OWS1.2 Image Handling Requirements through the use of interfaces defined in several OWS services.

2 Conformance

Not required for an IP IPR, DIPR, or Discussion Paper.

3 References[2]

The following documents contain provisions that, through reference in this text, constitute provisions of this document.

[1] "OWS 1.2 Image Handling Requirements," Version: 0.1.3, OGC Document 02-053, Date: 2002-11-19

[2] "Web Coverage Service (WCS)", Version 1.0, OGC Document 03-065r6, Date: 2003-10-16.

[3] "OWS1 Registry Service (WRS)," Version: 0.7.2, OGC Document 02-050r5, Date: 2002-11-18

[4] "Web Object Service (WOS)" Version: 0.0.2, OGC Document 02-049, Date: 2002-07-30

[5] "Web Map Service Implementation Specification," Version: 1.1.1, Date: 2002-01-16, OGC Document 01-068r3

[6] "OGC Open Web Services 1.2 Common Architecture." Version: 0.2, Date: 2002-08-25, OGC document 02-056r1

[2] Note that this clause has been changed from the template version which is titled "normative references" and contains different introductory text.

[7] “OWS 1.2 Image Handling Implementation,” Date January 2003, OGC Document 03-019

[8] “OGC Messaging Framework (OMF),” Version: 0.0.1, Date: 2002-08-14

4 Terms and definitions

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

operation
specification of a transformation or query that an object may be called to execute [OGC AS 12]

interface
named set of **operations** that characterize the behavior of an entity [OGC AS 12]

service
distinct part of the functionality that is provided by an entity through **interfaces** [OGC AS 12]

service instance
server
actual implementation of a **service** or conceptual role as recipient of an operation request.

client
software component that can invoke an **operation** from a **server** or conceptual role as originator of an operation request.

feature
abstraction of real world phenomena

coverage
feature that acts as a function to return one or more feature attribute values for any direct position within its spatiotemporal domain

EXAMPLE Examples include a raster image, polygon overlay, or digital elevation matrix.

image
coverage whose attribute values are a meaningful representation of a physical parameter derived from sensed data

Note: definition derived from the description of imagery in the ISO 19124 review summary and B5.12 of ISO 19115.

georectified gridded coverage
coverage with a gridded domain that is regularly spaced in a geographic or map coordinate system such that any cell can be uniquely geolocated, given the grid cell spacing, origin and orientation.

Note: definition based on MD_Georectified of ISO 19115 and general principles of ISO definitions - the template refers to ISO 19104 - that a definition cannot be multiple sentences. Changed initial "geospatial data" to "coverage".

georeferenced gridded coverage
coverage with a gridded domain whose cells are irregularly spaced in any geographic or map coordinate system and can be geolocated using geolocation information supplied with the data but not from the grid properties alone

Note: definition adapted from ISO 19115 definition of MD_Georeferenceable. Changed initial "geospatial data" to "coverage".

ungeoreferenced gridded coverage
coverage with a gridded domain whose cell geographic coordinates cannot be determined

Note: Ungeorectified is redundant; if it can't be gelocated it is by definition ungeorectified. The key point isn't lack of accompanying information; it's lack of any information.

5 Conventions

5.1 Symbols (and abbreviated terms)

ISO	International Organization for Standardization
OGC	Open GIS Consortium
OWS	OGC Web Services
UML	Unified Modeling Language
WCS	Web Coverage Service
WMS	Web Mapping Service
WOS	Web Object Service
WRS	Web Registry Service

5.2 UML Notation

5.2.1 UML Static Model

The diagrams that appear in this document are presented using the Unified Modeling Language (UML) static structure diagram. The UML notations used in this document are described in the diagram below.

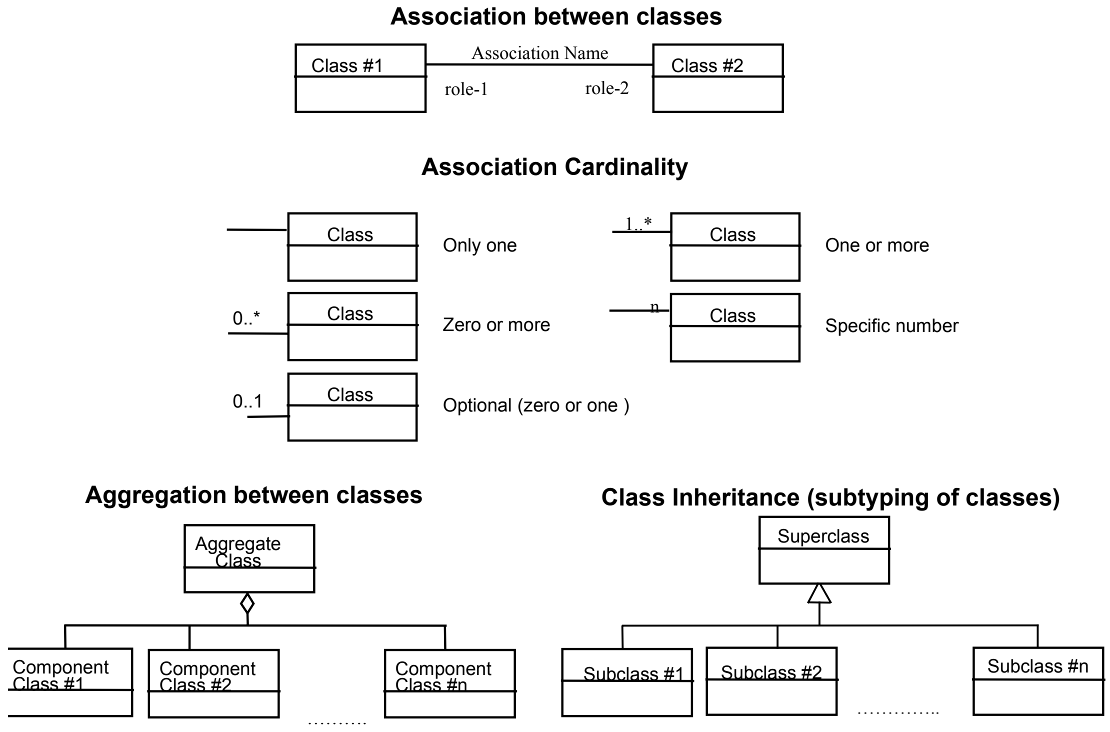

Figure 1— UML notation

In Figure 1, the following three stereotypes of UML classes are used:

a) <<Interface>> A definition of a set of operations that is supported by objects having this interface. An Interface class cannot contain any attributes.

b) <<DataType>> A descriptor of a set of values that lack identity (independent existence and the possibility of side effects). A DataType is a class with no operations whose primary purpose is to hold the information.

c) <<CodeList>> is a flexible enumeration that uses string values for expressing a list of potential values.

5.2.2 UML Dynamic Model

The diagrams that appear in this document are presented using the Unified Modeling Language (UML) dynamic structure diagram. The UML notations used in this document are described in the diagram below.

To represent dynamic models in this document, UML sequence diagrams are used. A sequence diagram shows object interactions arranged in time sequence. In particular it shows the objects participating in an interaction and the sequence of messages exchanged.

A sequence diagram has two dimensions: the vertical dimension represents time; the horizontal dimension represents objects participating in the interaction. Often, only the sequence of messages is important, but in real-time applications, the time axis can be an actual metric. There is no significance to the horizontal ordering of the objects.

6 Image Handling System Design

6.1 Overview

The main services of the Image Handling system design are the Image Archive and the Image Catalogue. An Image Archive provides persistence storage of images and associated metadata. An Image Catalogue provides persistence storage of image metadata. The Image Archive and the Image Catalogue implement interfaces defined in other OWS services.

6.2 OWS Interfaces Applicable to Image Handling

OGC has previously defined several key interfaces, e.g., Web Coverage Service. The design for Image Handling in OWS1.2 is accomplished through reuse of OGC interfaces. This section introduces the interfaces and indicates any requirements particular to the image handling design. Subsequent sections provide the image handling design using these interfaces.

6.2.1 WCS Interface

The Web Coverage Service (WCS) supports the networked interchange of geospatial data as "coverages" containing values or properties of geographic locations. WCS provides access to intact (unrendered) geospatial information, as needed for client-side rendering, multi-valued coverages, and input into scientific models and other clients beyond simple viewers. WCS as defined in [2] defines one interface with three operations as shown here in Figure 2.

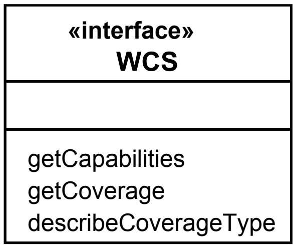

Figure 2 — WCS Interface
(note describeCoverageType was changed to describeCoverage in WCS version 1.0)

Requirements specific to Image Handling [1] that may be above and beyond those currently designed in the WCS interface are as follows[3]:

[3] Need to confirm that WCS interface meets these Image Handling Requirements

5.2.2 Image categories
5.2.3 Image types
5.2.4 Image sizes and formats
5.2.5 Image metadata structure
5.2.8 Web service interfaces

The paragraph number from [1] is indicated in the list above.

Design Issue: Potential changes to the WCS interface to meet Image Handling needs have been identified and are discussed below:

~~*- Allow retrieving a useful subset of a grid coverage that is georeferenced (and thus not georectified).*~~

~~*The getCoverage request allows specifying what Coordinate Reference System is used in the request. It needs to be clarified that the "request CRS" may be in a ground CRS and/or in an image CRS. For the case of a ground CRS in the request posed against a coverage that has a georeferenced coverage (e.g. native coverage CRS is not geographic), some coverage servers may be able to convert to the request CRS into the native CRS and respond to the getCoverage. Also, some servers may support a getCoverage request CRS that is the image CRS native to the coverage, in which case the request can be processed. These cases for georeferenced data have begun to be clarified in the WCS Request For Comment process. (In Abstract Specification Topic 2 (document 01-102), image pixels are located in an image coordinate reference system (which is a specified specialization of an engineering coordinate reference system). The XML schema for this, in document 02-036r3, uses the XML element named ImageCRS).*~~

(The design issue listed above, now with strikethrough font, was addressed in WCS1.0 which allows for a getCoverage request in a georeferenced CRS, e.g., image coordinates.)

- WCS response might provide result that optionally includes metadata describing the coverage subset actually returned.

The GetCoverage operation sometimes returns a somewhat different subset of the grid coverage, than is directly specified in the GetCoverage request. When that occurs, the client needs to receive metadata describing the coverage subset actually returned. Suppose that you are requesting a format that returns 3 payloads in the message (3 bands in separate file for example). Metadata is needed about the role of each payload in the message. Note the problem is also raised for transaction. The metadata can be sent as an attachment to the message. Multi-part messages was investigated in OWS as reported in [8].

- Need to clarify handling of identifiers for coverages that are retrieved from a catalog (and not contained in a Capabilities document).

Every image stored with WOS will have a unique identifier. That unique identifier is assigned by the archive when the coverage is input. Once the identifier is obtained by the user via WRS, it is used in the GetCoverage request. The WCS LayerID could be redefined to be the coverage unique identifier, with a URI value, suggest changing this parameter name to "CoverageURI" or something similar.

- Need to allow handling video images (timed sequence of non-georectified grid coverages), with different georeferencing and footprint for each frame.

This case was part of the UAV portion of the OWS1.2 Demostration. For more details see Image Archive Service Implementation DIPR (reference [7]).

- May need to adapt Filter Encoding for use in a GetCoverage request.

Need to specify the requirements that would require Filter Encoding as this is not a trivial change. Most coverage access is via rectangular "slices" --particularly when dealing with the grid case. When considering other kinds of coverages, priorities may change. May define new operators in the Filter that operate on the range that allows to navigate in the N-dimensional range space.

6.2.2 WMS Interface

A Web Map Service (WMS) produces maps of georeferenced data. A "map" is a visual representation of geodata; a map is not the data itself. WMS as defined in [5] contains one interface with three operations shown here in Figure 3.

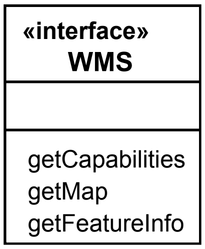

Figure 3 — WMS Interface

6.2.3 WRS Interfaces

The Web Registry Service (WRS) provides interfaces for querying and managing a metadata repository. A registry service is a web services profile of the OpenGIS Catalog Interface Implementation Specification. The Web Registry Service (WRS) as defined in [3] defines six interfaces of which three interfaces are used here. The relevant WRS interfaces for image handling are shown in Figure 4

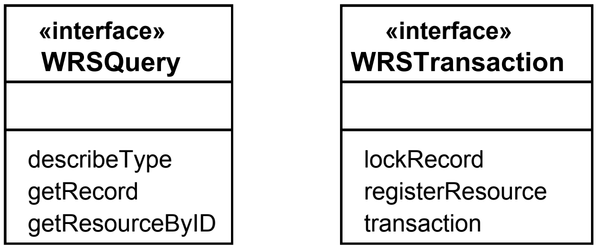

Figure 4 — WRS Interfaces relevant to Image Handling[4]

Requirements specific to Image Handling [1] that may be above and beyond those currently designed in the WRS interfaces are as follows[5]:

5.2.2 Image categories
5.2.3 Image types
5.2.4 Image sizes and formats
5.2.5 Image metadata structure
5.2.6 Multiple servers, para. (a)
5.2.8 Web service interfaces

The paragraph number from [1] is indicated in the list above.

Query refinement is a key capability as there may be millions of images in the catalogs and archives.

Design Issue: During the OWS1.2 the topic of query refinement was discussed, but it was unresolved how the query will be persisted between getRecord operations. It shall not be requirement of the actor to persist or recreate the query.

6.2.4 WOS Interface

The Web Object Service (WOS) provides operations for the management and access to generic objects that do not require a specialized interface. The WOS as defined in [4] contains one interface and five operations shown here in Figure 5

[4] WRS version 0.7.2 moves getCapabilities to an OWS Common interface class.

[5] Need to confirm that WRS interfaces meets these Image Handling Requirements

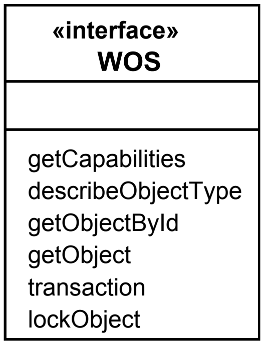

Figure 5 — WOS Interface

Requirements specific to Image Handling [1] that may be above and beyond those currently designed in the WRS interfaces are as follows[6]:

5.2.2 Image categories
5.2.3 Image types
5.2.4 Image sizes and formats
5.2.5 Image metadata structure
5.2.8 Web service interfaces

The paragraph number from [1] is indicated in the list above.

Reference [8] defines a messaging framework to conduct communications between OGC web services. It is independent of any transport protocol and any messaging encoding. The OGC Messaging Framework was used to support insert of image packages using WOS. Information in addition to the image itself needs to be inserted and retrieved from the image archive. OMF provides a method to define an package of imagery and associated information suitable for transfer via OGC Web Services.

6.2.5 OWS Common Interface

OWSCommon defines operations that are common to all OWS services. The OWSCommon getCapabilities operation is used to request the capability profile for a service; this operation is the common mechanism by which any OWS service describes itself. The OWS Common Interface contains one operation as shown here in Figure 6

[6] Need to confirm that WOS interface meets these Image Handling Requirements

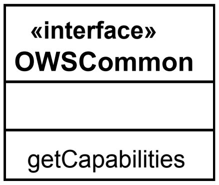

Figure 6 — WOS Interface

6.3 Interface Reuse for Image Handling

6.3.1 OWS Service Composition

The OWS 1.2 Common Architecture [6] defines three methods for defining a service that reuses multiple interfaces. The Image Archive Service and the Image Catalogue are defined using the Service Composition method for interface reuse.

In service composition interfaces presently implemented by two or more existing services are combined to form a new service. The getCapabilities is implemented once in the new service, but provides information about all of the capabilities and content of the new service. In the case of service composition, the interfaces composing the new service might have no independent existence.

6.3.2 Image Handling Interfaces

Interfaces for the Image Archive and for the Image Catalogue are defined through service composition using the interfaces defined in the previous section. Figure 7 provides a UML model showing the realization of interfaces in the image archive and image catalogue class. Table 1 provides a summary of which interfaces are mandatory, optional or not used in the Image Archive and Image Catalogue for OWS1.2.

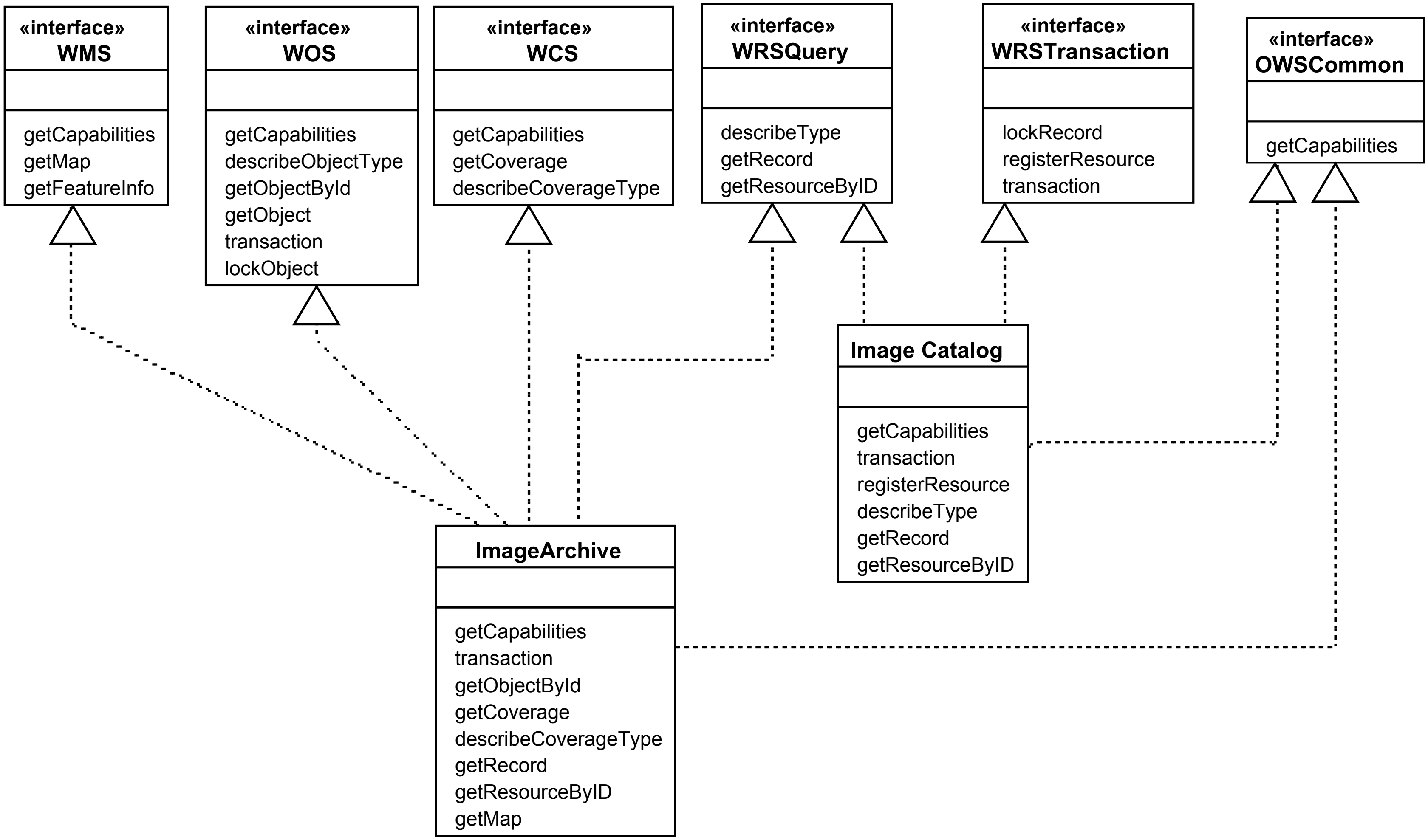

Figure 7— Class Diagram: Image Handling Services and Interfaces

(note describeCoverageType was changed to describeCoverage in WCS version 1.0)

Table 1 - Summary of IAS Interface Inheritance

Interface	Operation	Interface Reuse in IAS
OWSCommon	getCapabilities	**Mandatory**
WOS	getCapabilities	No. getCapabilities from owsCommon used instead
WOS	describeObjectType	No. May be used in future version of IAS.
WOS	getObjectById	**Mandatory**
WOS	getObject	No. May be used in future version of IAS.
WOS	transaction	**Mandatory**. Only INSERT is allowed. UPDATE and DELETE may be used in future version of IAS.
WOS	lockObject	No. May be used in future version of IAS.
WRSQuery	describeType	No. May be used in future version of IAS.
WRSQuery	getRecord	**Optional**
WRSQuery	getResourceByID	**Optional**
WRSTransaction	lockRecord	No. May be used in future version of IAS.
WRSTransaction	registerResource	No. May be used in future version of IAS.
WRSTransaction	transaction	No. Imagery metadata is provided in the WOS:transaction, therefore a separate metadata transaction is not needed.
WCS	getCapabilitties	No. getCapabilities from owsCommon used instead
WCS	getCoverage.	**Mandatory**
WCS	describeCoverage	**Optional**
WMS	getCapabilitties	No. getCapabilities from owsCommon used instead
WMS	getMap	**Optional**
WMS	getFeatureInfo	No. May be used in future version of IAS.

Through the inheritance shown in Figure 7, the following image handling requirements are met (note that the paragraph numbers are from [1]):

5.3.2 Get service metadata The archive function interface(s) shall provide one or more operations that together meet the following service metadata retrieval requirements:

5.3.3 Get schema The archive function interface(s) shall provide one or more operations that together meet the following data schemas retrieval requirements:

5.3.4 Input and delete object The archive function interface(s) shall provide one or more operations that together meet the following data input and management requirements:

5.3.5 Get object The archive function interface(s) shall provide one or more operations that together meet the following data and metadata retrieval requirements:

5.4.2 Get service metadata The catalog function interface(s) shall provide one or more operations that together meet the following service metadata retrieval requirements:

5.4.3 Get schema The catalog function interface(s) shall provide one or more operations that together meet the following metadata schemas retrieval requirements:

5.4.4 Input and delete metadata The catalog function interface(s) shall provide one or more operations that together meet the following metadata input and management requirements:

5.4.5 Query metadata The catalog function interface(s) shall provide one or more operations that together meet the following metadata query and retrieval requirements:

5.3.7 Get map (optional) The image archive function interface(s) can provide one or more operations that together meet the following coverage data portrayal requirements, if such operation(s) are specified:

5.3.6 Get coverage (optional) The image archive function interface(s) shall provide one or more operations that together meet the coverage data retrieval following requirements:

Design issue: Should the Image Catalog and Image Archive services include WRS:lockRecord and/or WOS:lockObject operations? The answer might be different for an Image Catalog and for an Image Archive. In considering these design questions, the following questions arise:

1. What client actions on an object (or record) are prohibited by a lock (not held by that client). For example, are GetObject and GetSingleObject prohibited on a locked object?

2. When would a lockRecord operation be used on an Image Catalog? When would inputting a new version that supersedes the previous version not work, or not be preferable? What is the use case(s) for catalog record locking?

3. When would a lockObject operation be used on an Image Archive? When would inputting a new version that supersedes the previous version not work, or not be preferable? What is the use case(s) for archive object locking?

6.4 Image Archive and Catalogue Relationships

Two additional classes are shown in Figure 8: Image Source and Image Client. The Image Source inserts imagery metadata into the Image Archive and may insert metadata into the Catalogue. The Image client can search either the catalogue or the archive to discover images of interest. The Image Client accesses the image archive for full or partial images.

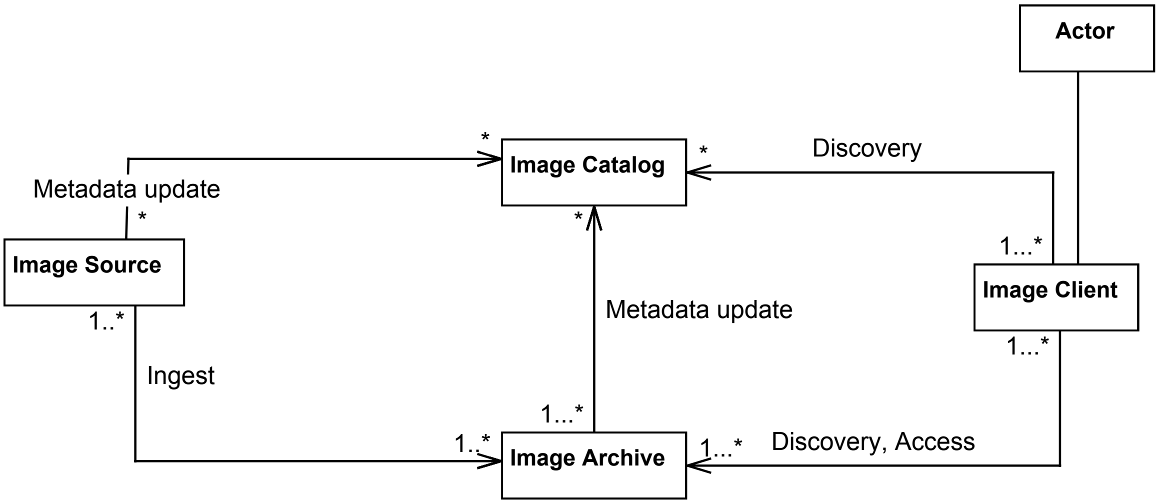

Figure 8 — Class Diagram: Image Handling Relationships

Through the relationships shown in Figure 8, the following image handling requirements are met (note that the paragraph numbers are from [1]):

5.2.6 Multiple servers a) The archiving and cataloging functions shall use interfaces and metadata that, when implemented by different web server instances, shall allow multiple catalog services to easily work with multiple archive services.

5.2.6 Multiple servers b) One catalog server instance shall be able to handle metadata for data handled by multiple archive server instances, when those servers are separately implemented.

5.2.6 Multiple servers c) Multiple catalog server instances shall be able to handle metadata for different data handled by one archive server instance, when those servers are separately implemented. In addition, multiple catalog server instances should be able to handle metadata for the same data handled by one archive server instance.

Note that for OWS1.2, two methods for managing metadata are under study:

1) The Image Source supplies metadata to one or more Image Archives and to one or more Image Catalogs. In this first case, the Image Source is responsible for consistency of metadata between the catalogue and archive.

2) The Image Source supplies metadata to one or more Image Archive. The Image Archive supplies metadata to the Image Catalog. In this second case the Image Archive is responsible for updating the Image Catalogue.

It is recognized that other metadata management approach are possible, i.e., this is not an exhaustive list of metadata management approaches.

Design Issue: Multiple approaches to metadata management. Two approaches for metadata management are described in this design. The approaches are not mutually exclusive and may both be required based on enterprise operational policies. For example, the OWS 1.2 RFQ mentions a potential time delay between when a new image is submitted to the archive and when it is available from the archive service for downloads. Assuming that the registry/catalog can bring the metadata on-line sooner than the IAS can bring the image on-line, the potential for making imagery requests before they can be processed seems to be significant.

If the Image Archive (automatically) inputs search metadata into Image Catalogs, additional requirements can be derived from the requirements stated in Subclause 5.2.5 of [1]. These derived requirements on the interface designs tentatively include:

1. An Archive shall automatically input search metadata into a set of one or more Catalogs whenever new or modified data is input into that Archive. An Archive should also be able to automatically input search metadata into a set containing zero Catalogs.

2. An Archive should allow a client (Image Client or Image Source) to initiate automatic input of search metadata into a set of one or more additional Catalogs for data previously input into that Archive.

3. An Archive shall use information from the client specifying the set of Catalogs into which to automatically input search metadata. This set of Catalogs shall be allowed to be different for each new data set that is input into that Archive.

4. An Archive shall automatically remove or deprecate search metadata from all the Catalogs into which it put search metadata whenever the previously input data is removed or deprecated from that Archive.

Consider using the registerResourcee:

'push' through the transaction() operation, specifying the appropriate data manipulation commands (insert, update, delete) with included registry objects. This operation can be invoked by a dedicated registry client or perhaps by another service functioning as a client.

'pull' through the registerResource() operation your 'manual' mode seems to be 'push' through a dedicated registry client that presents a user-friendly interface

6.5 Image Archive Service

In order to support the interfaces identified above, the Image Archive will provide services on the imagery that are not uniquely defined simply by responding to the interfaces. For example some Image Archives will provide access to images in coordinate reference system (CRS) other than the native CRS of the image when it is inserted in the archive, e.g., the archive may rectify the imagery based on metadata provided with the image. An archive may compute statistics on the images, e.g., histogram, and supply that as metadata – metadata that was not provide on the insert. These services will be variable by implementation and there has been no agreement in the testbed as to mandatory value-added services provided by image archive service.

In addition to supporting the interfaces listed in Table 1, an Image Archive must provide persistent storage. No other service is specifically required of an Image Archive. An example Image Archive is shown in Figure 9. The example provides an InsertManager that may perform checks on an inserted image, may compute image statistics, and may provide some measure of transaction management in placing an image package in the PersistentStorage. The level of query against the metadata in the archive will vary and may include queries against the image data, i.e., data mining. To satisfy the getCoverage operation the Image Archive may provide coordinate transformation and interpolation services providing images that differ from the inserted image to meet image client needs. The variation of services provided by an Image Service will vary by implementation.

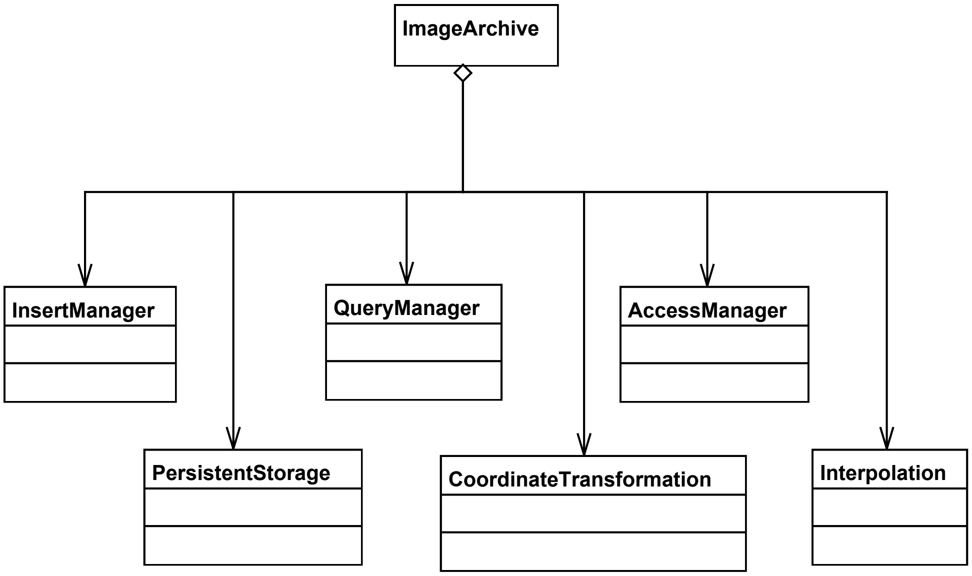

Figure 9 — Image Archive with Example Services

6.6 Image Catalogue Service

Requirements met by the Image Archive Service are: TBD

7 Sequence Diagrams

7.1 Search / discover image metadata

An actor uses an Image Client to search for images of interest, i.e., that are fit to the users use. The search can be done against an Image Catalog (see Figure 10) or an Image

Archive (see Figure 11). The interaction patterns are similar but the search results may differ. An Image Catalog may search metadata for images from multiple archives either because the Catalog contains the metadata or the Catalog propagates a distributed search. The records returned from a query include a URL for image type description document.

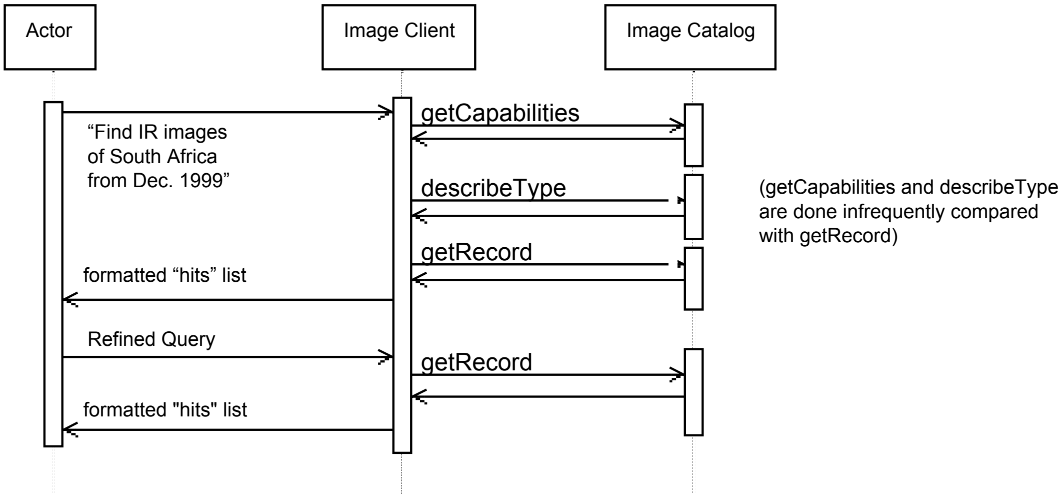

Figure 10 — Sequence Diagram: Search / discover in Image Catalog

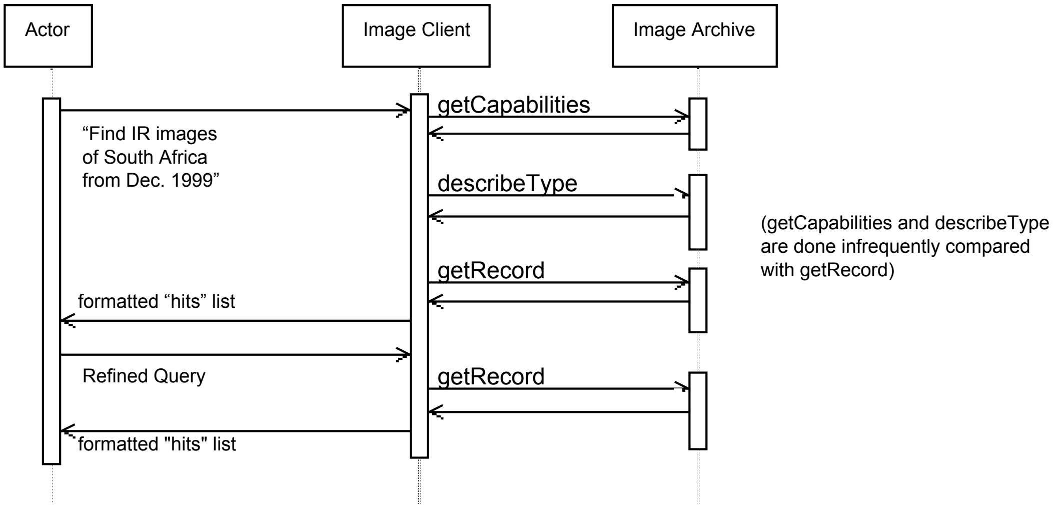

Figure 11 — Sequence Diagram: Search / discover in Image Archive

Discussion about the registry interface:

Question: how do I reuse a Registry response in a subsequent GetCoverage request?

- Answer: Registry gives you a URL; you can reuse that URL as a layer name in GetCoverage. *A Design Issue remains whether the registry returns parts of the request that would be valuable in creating the getCoverage request. Therefore when the client uses the URL against the archive for access it may need to insert that information, eg. Bounding box in getCoverage request*.

Question: How do I associate a particular piece of data with the appropriate operations needed to access it? (More specifically: how does the registry tell me to use GetCoverage for access to particular imagery?)

- Answer: The registry lists services, data, and associations between them. So you can search on data, and then find services that give you access to that data. (Or vice-versa)

Question: If I make a spatially-bounded registry search, should the registry give me complete (e.g., GetCoverage) data-access requests instead of just reporting data and their associations with services?

- Answer: that would definitely be value-added functionality – above and beyond the expected functions of a registry.

7.2 Fetch whole images

This sequence assumes that the Actor has an identifier for an image. The Actor uses the Image Client to retrieve the entirety of the image. The Image Client may then process the image producing some result for the Actor, e.g, calculate statistics about the image, or render the image on a display.

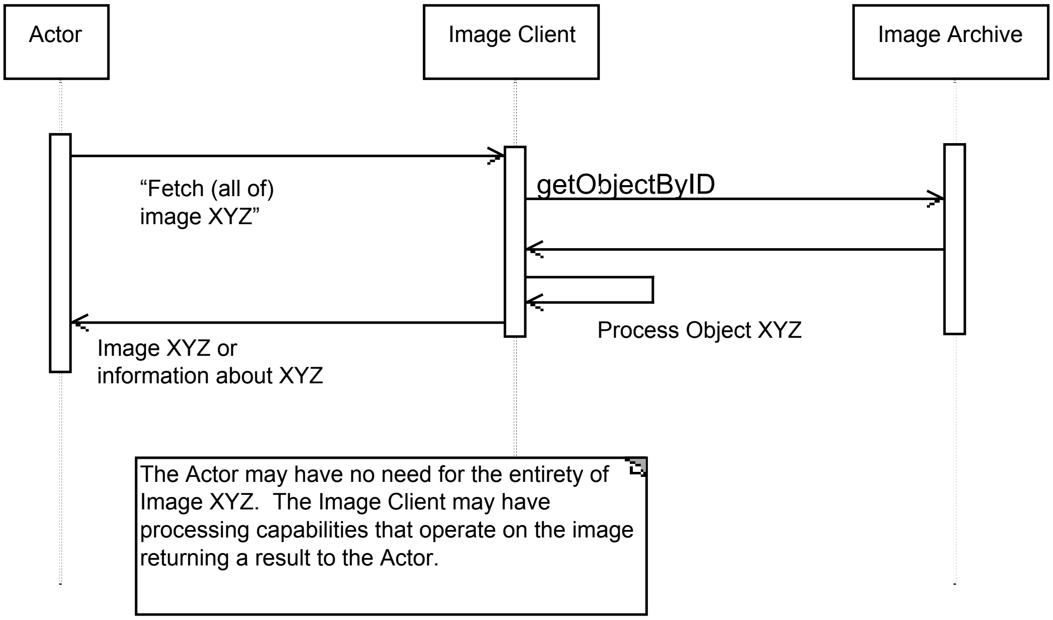
Actor
Image Client
Image Archive
"Fetch (all of) image XYZ"
getObjectByID
Process Object XYZ
Image XYZ or information about XYZ
The Actor may have no need for the entirety of Image XYZ. The Image Client may have processing capabilities that operate on the image returning a result to the Actor.

Figure 12 — Sequence Diagram: Fetch whole images

7.3 Fetch imagery as coverage extracts

This sequence assumes that the Actor has an identifier for an image. The Actor uses the Image Client to retrieve a portion of the image. The Image Client may then process the image producing some result for the Actor, e.g, calculate statistics about the image, or render the image on a display.

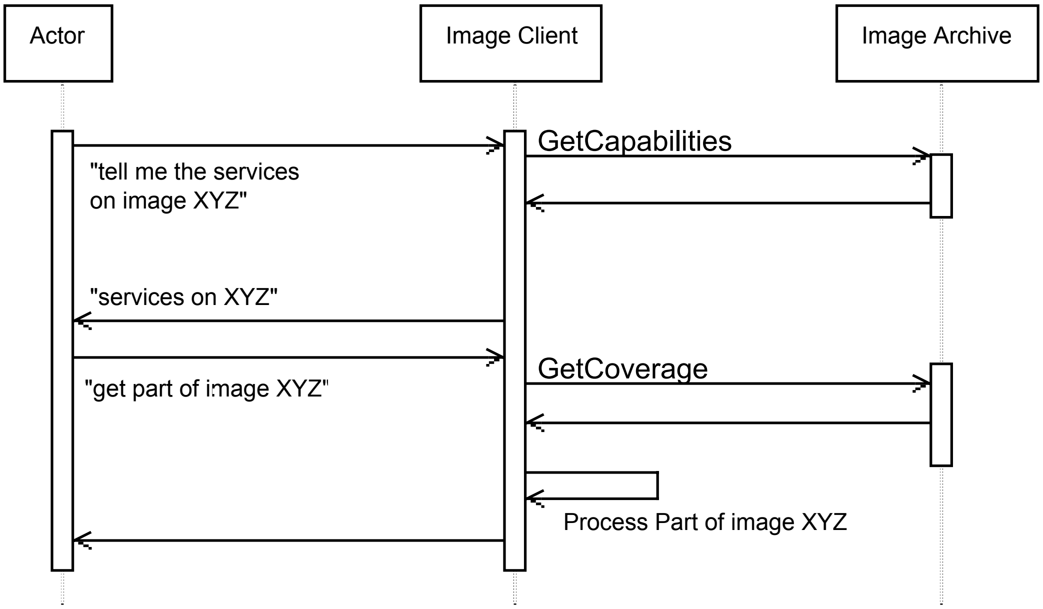

Figure 13— Sequence Diagram: Fetch imagery as coverage extracts

7.4 Insert new image: Source manages metadata

This is the first of two approaches to of inserting an image in an archive. In this first approach, the Image Source is responsible for putting the metadata in one or more Image Catalogs after the image is in an Image Archive.

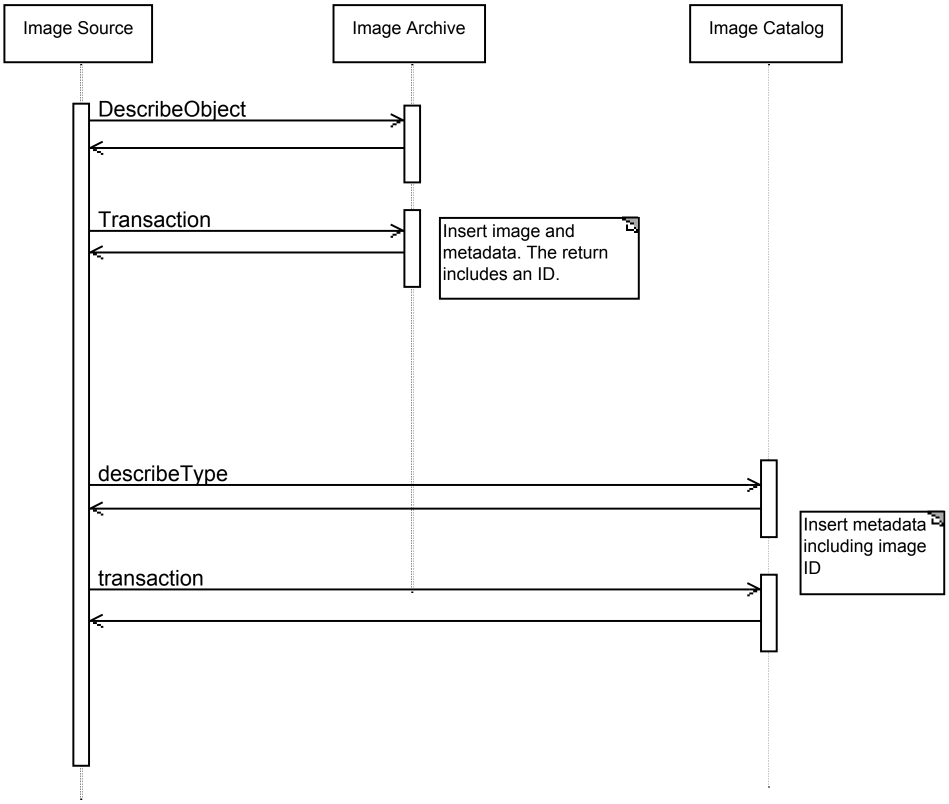
Image Source
Image Archive
Image Catalog
DescribeObject
Transaction
Insert image and metadata. The return includes an ID.
describeType
Insert metadata including image ID
transaction

Figure 14— Sequence Diagram: Insert image, source manages metadata

7.5 Insert new image: Archive manages metadata

This is the second of two approaches of inserting an image in an archive. In this second approach, the Image Archive is responsible for putting the metadata in one or more Image Catalogs after the image is in an Image Archive.

(Need to add a step that the archive registers itself in the registry.)

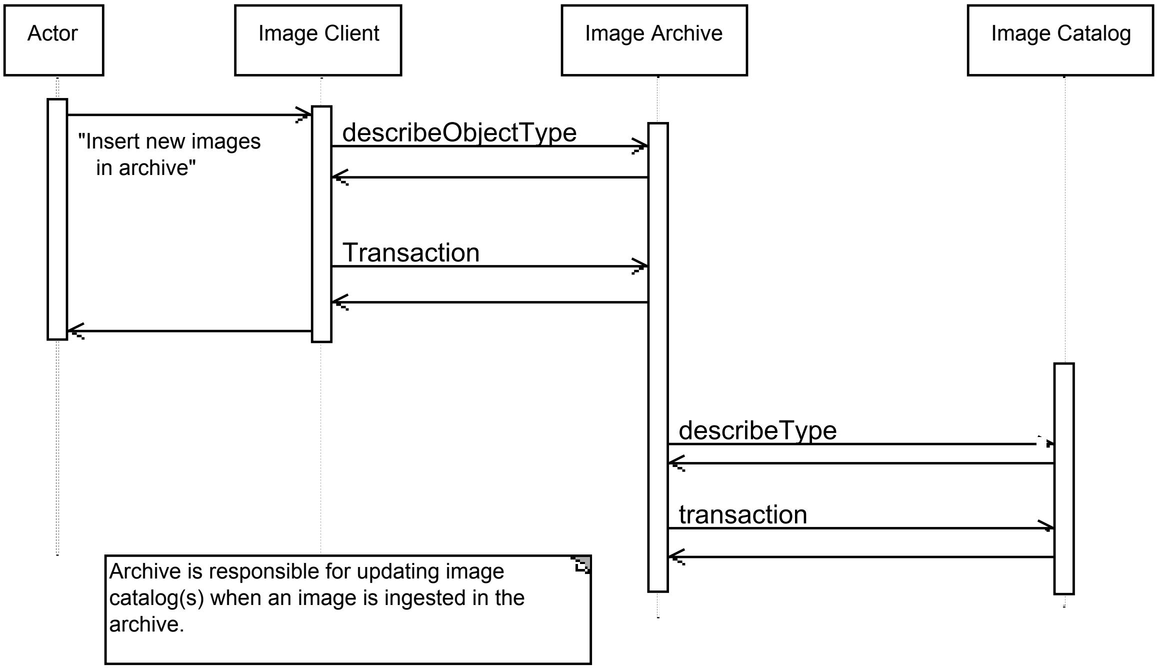
Actor
Image Client
Image Archive
Image Catalog
"Insert new images in archive"
describeObjectType
Transaction
describeType
transaction
Archive is responsible for updating image catalog(s) when an image is ingested in the archive.

Figure 15— Sequence Diagram: Insert new image: Archive manages metadata

7.6 Add image metadata to existing image

This sequence allows for update of the metadata after an image is in an archive. For example an analyst may discover an image, review the image and determine some quality attribute about the image. This sequence provides for the analyst to add metadata to the existing metadata in a catalog.

Design Issue: discussion is needed about metadata management for this sequence. Which metadata store does the client update: catalog or archive? Which service is responsible for metadata management: Image Client, Image Catalog, Image Archive? Only one option is shown here.

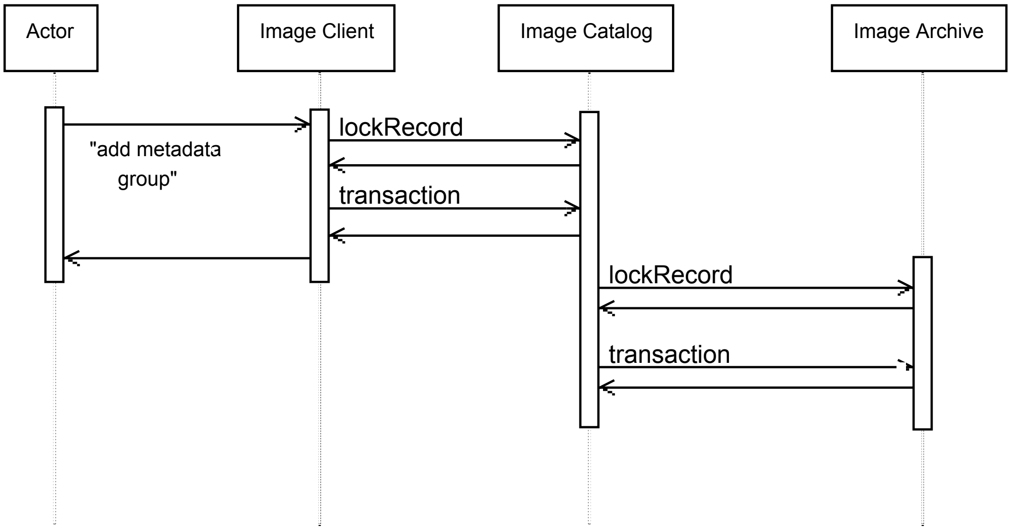

Figure 16— Sequence Diagram: Add image metadata

7.7 Sequence Diagram using WMS

This sequence assumes that the actor has determined an image of interest and wants to view a picture, i.e., reduced resolution from the image. The Image Client may provide the Actor with additional layers to create a map with the picture of the image along with other layers.

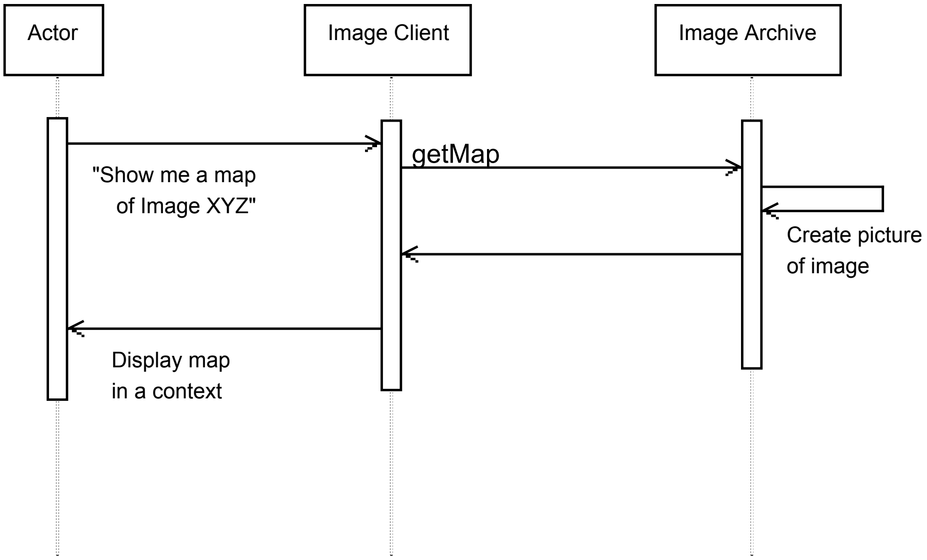

Figure 17 — Sequence Diagram: Get a picture of the image

ANNEX - Requirements Traceability

This annex maps the requirements of “OWS 1.2 image handling requirements,” (OGC document 02-053r1, Date: 2002-11-19, Version: 0.1.3) to the clauses of the Image Handling Design, i.e., this document.

Requirements Clause	**Requirements Text**	**Cross reference to design**
5 Requirements		
5.1 Introduction		
5.2 Common requirements		
5.2.1 Introduction		
5.2.2 Image categories	The image metadata, image archive interface(s), and image catalog interface(s) shall all meet the following image category requirements:	To be satisfied by interfaces: 6.2.1 WCS Interface 6.2.3 WRS Interfaces 6.2.4 WOS Interface
5.2.3 Image types	The image metadata, image archive interface(s), and image catalog interface(s) shall all be able to handle a wide variety of image types, including:	To be satisfied by interfaces: 6.2.1 WCS Interface 6.2.3 WRS Interfaces 6.2.4 WOS Interface
5.2.4 Image sizes and formats	The image metadata, image archive interface(s), and image catalog interface(s) shall all be able to handle a wide variety of image sizes and formats, including::	To be satisfied by interfaces: 6.2.1 WCS Interface 6.2.3 WRS Interfaces 6.2.4 WOS Interface
5.2.5 Image metadata structure	The image metadata, image archive interface(s), and image catalog interface(s) shall together meet the following metadata structure requirements:	To be satisfied by interfaces: 6.2.1 WCS Interface 6.2.3 WRS Interfaces 6.2.4 WOS Interface

Requirements Clause	Requirements Text	Cross reference to design
5.2.6 Multiple servers	The image archive interface(s), image catalog interface(s), and image metadata shall together meet the following multiple server instance requirements:	
	a) The archiving and cataloging functions shall use interfaces and metadata that, when implemented by different web server instances, shall allow multiple catalog services to easily work with multiple archive services.	Multiplicity of catalogs and archives, see 6.4 Image Archive and Catalogue Relationships Catalogue enables distributed search 6.2.3 WRS Interfaces
	b) One catalog server instance shall be able to handle metadata for data handled by multiple archive server instances.	Multiplicity of catalogs and archives, see 6.4 Image Archive and Catalogue Relationships Sequence diagram for metadata updates: 7.4 Insert new image: Source manages metadata
	c) Multiple catalog server instances shall be able to handle metadata for different data handled by one archive server instance. In addition, multiple catalog server instances should be able to handle metadata for the same data handled by one archive server instance.	Multiplicity of catalogs and archives, see 6.4 Image Archive and Catalogue Relationships Sequence diagram for metadata updates: 7.4 Insert new image: Source manages metadata
	d) The transition from image discovery, using an image catalog service, to image exploitation, using an image archive service, should be as smooth and easy as practical.	Sequence diagrams for discovery and access: 7.1 Search / discover image metadata 7.2 Fetch whole images 7.3 Fetch imagery as coverage extracts
5.2.7 Enterprise scalability		
5.2.8 Web service interfaces	The service interfaces specified to support the image archive and image catalog functions shall meet the following web service interface requirements:	To be satisfied by interfaces: 6.2.1 WCS Interface 6.2.3 WRS Interfaces 6.2.4 WOS Interface

Requirements Clause	**Requirements Text**	**Cross reference to design**
5.3 Image archive		
5.3.1 Introduction		
5.3.2 Get service metadata	The archive function interface(s) shall provide one or more operations that together meet the following service metadata retrieval requirements:	Satisfied through interface inheritance: 6.2.5 OWS Common Interface
5.3.3 Get schema	The archive function interface(s) shall provide one or more operations that together meet the following data schemas retrieval requirements:	Satisfied through interface inheritance: 6.2.5 OWS Common Interface
5.3.4 Input and delete object	The archive function interface(s) shall provide one or more operations that together meet the following data input and management requirements:	Satisfied through interface inheritance: 6.2.5 OWS Common Interface
5.3.5 Get object	The archive function interface(s) shall provide one or more operations that together meet the following data and metadata retrieval requirements:	Satisfied through interface inheritance: 6.2.5 OWS Common Interface
5.3.6 Get coverage (optional)	The image archive function interface(s) shall provide one or more operations that together meet the coverage data retrieval following requirements:	Satisfied through interface inheritance: 6.2.5 OWS Common Interface
5.3.7 Get map (optional)	The image archive function interface(s) can provide one or more operations that together meet the following coverage data portrayal requirements, if such operation(s) are specified:	Satisfied through interface inheritance: 6.2.5 OWS Common Interface
5.4 Image catalog		
5.4.1 Introduction		

Requirements Clause	**Requirements Text**	**Cross reference to design**
5.4.2 Get service metadata	The catalog function interface(s) shall provide one or more operations that together meet the following service metadata retrieval requirements:	Satisfied through interface inheritance: 6.2.5 OWS Common Interface
5.4.3 Get schema	The catalog function interface(s) shall provide one or more operations that together meet the following metadata schemas retrieval requirements:	Satisfied through interface inheritance: 6.2.5 OWS Common Interface
5.4.4 Input and delete metadata	The catalog function interface(s) shall provide one or more operations that together meet the following metadata input and management requirements:	Satisfied through interface inheritance: 6.2.5 OWS Common Interface
5.4.5 Query metadata	The catalog function interface(s) shall provide one or more operations that together meet the following metadata query and retrieval requirements:	Satisfied through interface inheritance: 6.2.5 OWS Common Interface
5.5 Image metadata		
5.5.1 Introduction		
5.5.2 Catalog metadata	*(No requirements applicable to system design)*	
5.5.3 Archive metadata	*(No requirements applicable to system design)*	