# Heterogeneous Mission Accessibility - Follow-On - Online Data Access (HMA-FO_ODA)

## Design Definition File (DDF) - Software Design Document (SDD)

| | |
|---|---|
| EOX Doc.Ref: | HMA-FO_ODA-DDF-SDD_EOX |
| Issue: | 1.0 |
| Date: | 2010-11-26 |
| Authors: | C.Schiller, S.Krause, S.Meissl, S.Smolders |
| Responsible for Document: | C.Schiller |

Prime Contractor

Subcontractors

Thurngasse 8/4
1090 Wien
Österreich / Austria

office@eox.at
www.eox.at

IBAN AT171400024510821897
BIC BAWAATWW

Registered Vienna
Commercial chamber Vienna
FN 312068z
UID ATU64265057

| Approved by | Organisation | Date | Signature |
|---|---|---|---|
| Gerhard Triebnig | EOX | 2010-11-26 | |
| Pier-Giorgio Marchetti | ESA | | |
| | | | |

## Release Track

| **Release No.** | **Date** | **Remarks** |
|---|---|---|
| 1.0 | 2010-11-26 | Initial release as part of HMA-FO_ODA- |
| | | |
| | | |
| | | |
| | | |

**Distribution List:**

| | |
|---|---|
| Authors: | All |
| EOX: | G. Triebnig |
| ESA: | P.G. Marchetti |
| GIM: | S. Desender |
| JUB: | P. Baumann |
| SPOT: | D. Giacobbo, P.Merigot, A. Robin |
| Spacebel: | Y. Coene |

# Table of Content

# List of Tables

## List of Figures

# 1 Introduction

This document defines the architectural design for the Reference Implementation of an Online Data Access (ODA) system designed and developed in the HMA-Follow-on (HMA-FO) Task 3 project.

This document will be updated during the HMA-FO project taking into account the updates on WCS EO AP that will be performed in OGC's WCS.SWG .

The ODA system shall provide easy online access to EO datasets utilizing the new WCS 2.0 OGC standard and the WCS EO Application Profile. Thereby is shall demonstrate the possibilities and the use of the enhanced functionalities (e.g. subsetting in space and time) available when compared to simple FTP access.

The HMA-FO Reference Implementation (hereafter also called ODA system) shall demonstrate the following main functionalities:

- provide online access to EO datasets
- allow the subsetting of EO datasets
- allow the reprojection of EO dataset
- allow to access the EO datasets in different formats
- provide metadata for accessed EO datasets
- utilize OGC's WCS 2.0 standard
- utilize OGC's WCS EO AP (in its current draft version)
- prove the suitability of Open Source software tools

However, it has to be noted that extension for the WCS 2.0 are currently missing (e.g. interpolation) or are in preparation (e.g. netCDF and JPEG200 format encoding) and it may therefore be not possible to demonstrate the respective functionality in the Reference Implementation. For some missing extensions (e.g. CRS) the solutions defined for WCS 1.0.0 or WCS 1.1.0 will be used in the implementation of the WCS 2.0 software tool.

The above functionalities will be provided by supporting the following WCS 2.0 and WCS EO AP interfaces:

- GetCapabilities
- DescribeCoverage
- DescribeEOCoverageSet
- GetCoverage

# 2 Applicable and reference documents

## 2.1 Applicable documents

[AD1] ECSS Standard: Space Engineering – Software, Ref. ECSS-E-ST-40C, 6 March 2009

[AD2] HMA-FO_ODA Requirements Baseline Document - Software System Specification (HMA-FO_ODA-RB-SSS_EOX, ver. 1.4, 2010-11-03)

[AD3] HMA-FO_ODA Requirements Baseline Document – Technical Note (HMA-FO_ODA-RB-TN_EOX, ver.1.1, 2010-05-27)

[AD4] ODA_TN: Specification Dependencies – Technical Note (hmafo-tn-0001-spb-v12-draft.doc, 2010-03-19)

[AD5] OGC 09-146r1, GML Application Schema Coverages , ver. 1.0.0, 2010-10-27

[AD6] OGC 09-110r3, WCS Interface Standard: Core, ver. 2.0.0, 2010-10-27

[AD7] OGC 09-147r1, Web Coverage Service 2.0 Interface Standard - KVP Protocol Binding Extension, ver. 1.0.0, 2010-10-27

[AD8] OGC 09-148r1, Web Coverage Service 2.0 Interface Standard - XML/POST Protocol Binding Extension, version 1.0.0, 2010-10-27

[AD9] OGC 07-118r8, User Management Interfaces for Earth Observation Services, version 1.0, 2010-09-08

[AD10] OGC 07-036, Geography Markup Language (GML) Encoding Standard, v.3.2.1, 2007-08-27

[AD11] HMA-FO_ODA, Technical Specification - Software Requirements Specification, (HMA-FO_ODA-TS-SRS_EOX), ver. 1.2, 2010-11-26

[AD12] OGC 07-140, OGC WCS 2.0 Application Profile - Earth Observation, draft, 2010-10-27

[AD13] OGC 10-147, OGC Web Coverage Service (WCS) 2.0 Interface Standard GeoTIFF Encoding Format Extension, ver. 0.0.1, 2010-07-06

[AD14] WCS 2.0 Format Encoding Extension, Presentation at 74th OGC Technical Committee, Toulouse, France, Peter Baumann, September 21, 2010

[AD15] OGC 10-157, Earth Observation Metadata profile of Observations & Measurements, ver. 0.2.0, 2010-11-11

[AD16] Invitation to Tender ESRIN/AO/1-5949/09/I-LG – HMA Follow on activities, Frascati, 19th of January 2009

[AD1] OGC 06-126r2, Compliance Test Language (CTL), ver. 0.6, 31/03/2009

## 2.2 Reference documents

[RD1] http://www.mapserver.org/, MapServer Homepage

[RD2] http://geoserver.org/display/GEOS/Welcome, GeoServer Open Source

[RD3] http://www.deegree.org/, deegree Homepage

[RD4] http://www.unidata.ucar.edu/projects/THREDDS/, THREDDS Data Server Homepage

[RD5] http://www.osgeo.org/, OSGeo Homepage

[RD6] http://www.osgeo.org/node/812, News MapServer Incubation Graduation

[RD7] http://mapserver.org/introduction.html, MapServer Introduction

[RD8] http://www.swig.org/, SWIG Homepage

[RD9] http://www.python.org/, Python Homepage

[RD10] http://svn.osgeo.org/mapserver/trunk/, MapServer development source (svn)

[RD11] http://subversion.tigris.org/, Subversion Homepage

[RD12] http://trac.osgeo.org/mapserver/, MapServer issue tracker (trac)

[RD13] http://trac.edgewall.org/, Trac Homepage

[RD14] http://www.mapserver.org/community/lists.html, MapServer mailings lists

[RD15] http://www.gnu.org/software/mailman/, Mailman Homepage

[RD16] http://www.mapserver.org/community/irc.html, MapServer IRC

[RD17] http://sourceforge.net/, SourceForge

[RD18] http://code.google.com/, Google Code

[RD19] http://zeus.pin.unifi.it/projects/wcsClientLite/, WCS Client of the University of Florence

[RD20] http://www.python.org/dev/peps/pep-0333/, WSGI

# 3 Terms, definitions and abbreviated terms

| | |
|---|---|
| AP | Application Profile |
| CDS | Coordinated Data access System (GSCDA) |
| EO | Earth Observation |
| EP | Extension Package |
| ESA | European Space Agency |
| GCM | GMES Contributing Mission |
| GDAL | Geospatial Data Abstraction Library |
| GMES | Global Monitoring for Environment and Security |
| GML | Geographic Markup Language (OGC) |
| GSC-DA | GMES space component - data access |
| HMA | Heterogeneous Mission Accessibility |
| HMA-E | HME – ESA |
| HMA-FO | HMA – Follow On |
| HMA-I | HMA - Initial |
| HME-T | HMA- Testbed |
| ICD | Interface Control Document |
| KVP | Key-Value Pair |
| HMI | Human Machine Interface |
| ODA | Online Data Access |
| OGC | Open Geospatial Consortium |
| OGR | OGR Simple Features Library |
| OSGeo | Open Source Geospatial Foundation |
| OSS | Open Source Software |
| RB | Requirements Baseline |
| Reference Implementation | A reference implementation is a fully functional implementation of a specification in reference to which other implementations can be evaluated. |
| SOAP | Simple Object Access Protocol, |
| SOS | Sensor Observation Service (OGC) |
| SSE | Service Support Environment |
| TDS | THREDDS Data Server |
| THREDDS | Thematic Realtime Environmental Distributed Data Services (TDS) |
| TN | Technical Note |
| TS | Technical Specification |
| WCPS | Web Coverage Processing Service (OGC) |
| WCS | Web Coverage Service (OGC) |
| WCS EO AP | WCS Earth Observation Application Profile |

| | |
|---|---|
| WCS-T | Web Coverage Service – Transactional (OGC) |
| WCTS | Web Coordinate Transformation Service (OGC) |
| WFS | Web Feature Service (OGC) |
| WMS | Web Mapping Service(OGC) |
| WPS | Web Processing Service (OGC) |
| XML | Extended Markup Language |

# 4 Software design overview

This chapter provides a general overview of the system specifying:

- The context in which the system is operating
- The background of the project and relationships with other projects
- The static and dynamic architecture
- The model of the main information items handled by the system
- The notations and methodologies followed for the design of the system

## 4.1 Software static architecture

This section reports the main components of the system, the identified relationships, the different statuses in which the system is operating as well as the model of the main information items handled .

### 4.1.1 Components Overview

The Reference Implementation of the ODA system shall demonstrate an easy online access to EO datasets utilizing the new WCS 2.0 OGC standard and the WCS EO AP.

The Reference Implementation ODA system include the following components:

- Web Server, which provides the access point to the successive components and performs a basic authentication of a user
- Server for WCS EO AP provides the first handling of the incoming request and supervises the request processing accordingly
- geo-RDBMS, a geo-enabled RDBMS which contains the coverage metadata and the access instructions to the datasets (e.g. contained in a Storage-Backend)
- mapping Server enhanced with WCS 2.0 functionality
- Storage-Backend represents the actual location of the datasets. This can internally (accessible via OS function, e.g. cpio) or an external archive location accessible via HTTP, FTP, or WCS protocol.
- Libraries for raster and vector data manipulation, provide backend functionalities to perform format conversions as well as CRS transformations

The following component diagram (see 4.7 for UML notations) shows the main software components of the ODA system

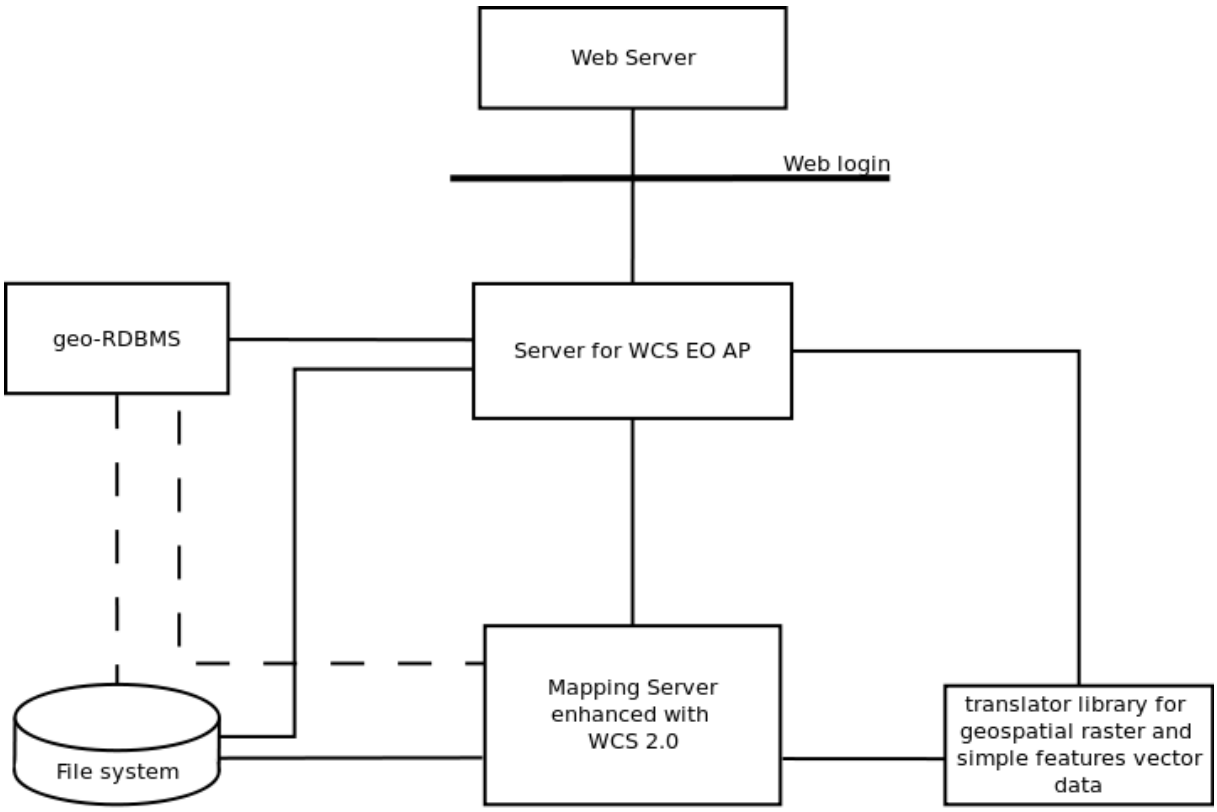


Figure 1: ODA system component diagram

### 4.1.2 Relationship with other systems

A relationship to external systems are not necessarily required. The Reference Implementation ODA system can also function as a stand-alone entity.

However, two external systems may be connected to a ODA system.

- Storage-Backend system:

  The connection to external Storage-Backend systems (e.g. EO archives) are possible. The Storage-Backend system would then hold the offered datasets and allow access for the ODA system to retrieve requested datasets on demand. The location of the datasets in the Storage-Backend's file system, together with metadata parameters like Bounding box, Date and Time, etc. is stored in the ODA systems geo-RDBMS. The interfaces specification to achieve such connection are described in section 4.4. Details about the geo-RDBMS as well as the database's data-model is provided in section 5.4.5.

- ODA Admin Client:

  The ODA Admin Client allows to access the ODA system in order to perform Administration Tasks. These Tasks include:

- configuration of coverage dataset series
- data ingestion
- update (after data ingestion or configuration)
- DB Management

### 4.1.3 Information Model Overview

The ODA system utilizes the coverage data model of OGC defined in [AD10], as well as any changes introduced in [AD5], [AD6], [AD12].

The applied information model is depicted in the Figure 2, 3, and 4 as well as in the Tables 1, 2, 3, 4, 5, 6,and 7.

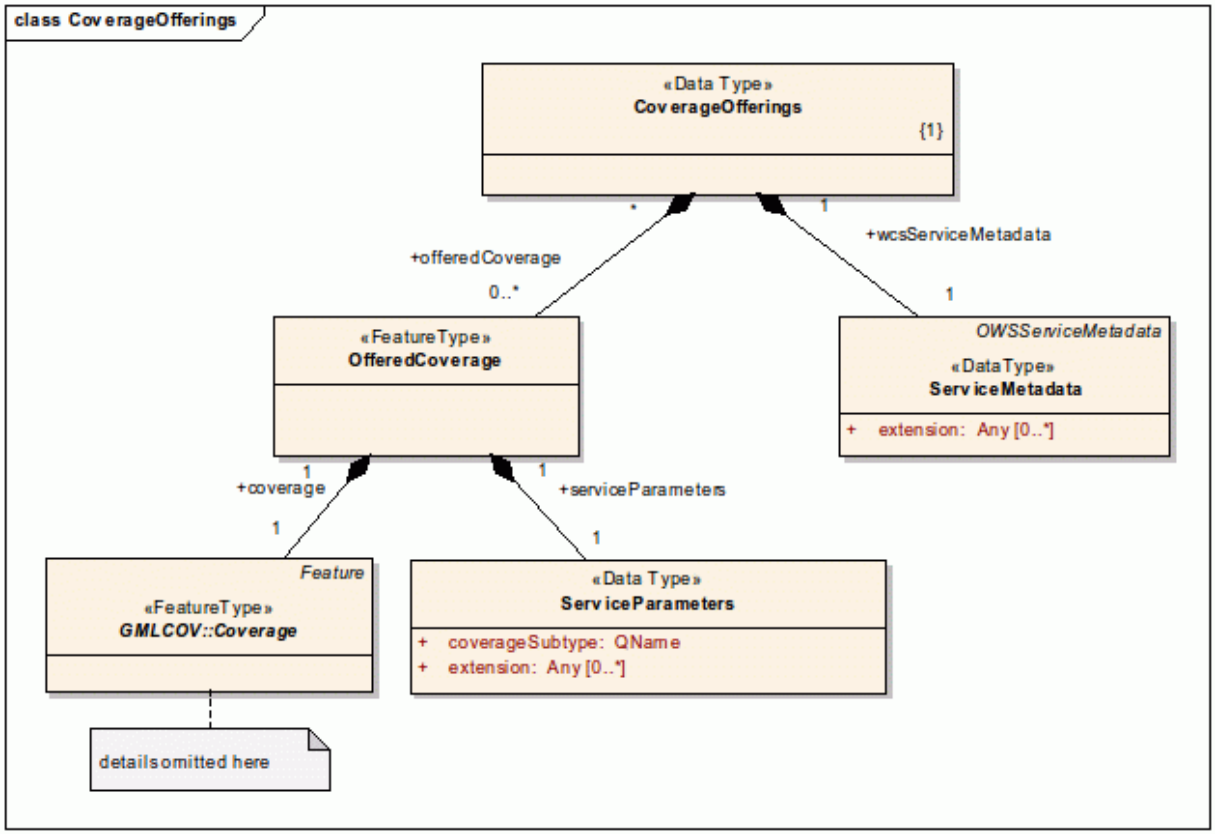


Figure 2: CoverageOfferings UML class diagram (Source: [AD6])

Figure 2 describes the basic *CoverageOfferins* as a UML diagram.

*Note: Many components of the core GML structure CoverageOfferings are left underspecified (e.g., in terms of multiplicity of the elements or proper semantics and use of a component). Any item thus underspecified can be handled arbitrarily by implementations – among others, a server is free to deliver optional elements or not, and a client is free to ignore optional elements when present.*

Each offering of the ODA system is described by a single instance of type *CoverageOfferings* which contains the components listed in Table 1.

| Name | Definition | Data type | Multiplicity |
|---|---|---|---|
| *offeredCoverage* | Set of coverages offered by this service | *OfferedCoverage* | zero or more (optional) |
| *serviceMetadata* | Information specific to this WCS service as a whole | *ServiceMetadata* | one (mandatory) |
| *coverage* | The coverage | *GMLCOV::Coverage* | one (mandatory) |
| *serviceParameters* | Service parameters individual for the coverage on hand | *ServiceParameters* | zero or more (optional) |

Table 1: Components of CoverageOfferings

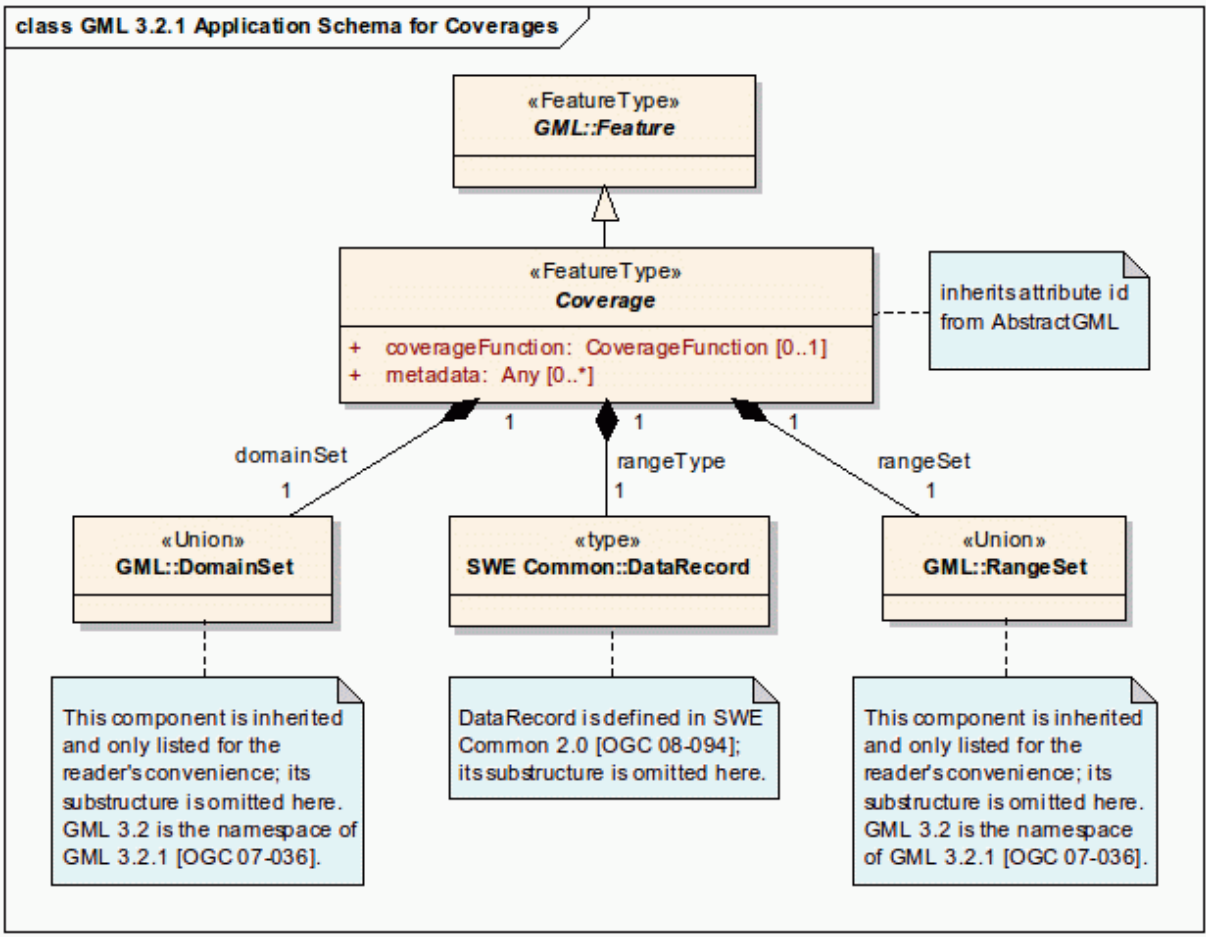


Figure 3: The Coverage data structure. (Source: [AD5])

| Name | Definition | Data type | Multiplicity |
|---|---|---|---|
| *coverage-Function* | GML 3.2.1 coverage function to describe how range values at coverage locations can be obtained | *GML:: Coverage-Function* | Zero or one (optional) |
| *metadata* | Application specific metadata | *Any* | Zero or more (optional) |
| *domainSet* | GML 3.2.1 Definition of coverage domain | *GML:: DomainSe* | One (mandatory) |
| *rangeType* | Structure definition of the coverage range values | *SWE:: DataRecord* | One (mandatory) |
| *rangeSet* | GML 3.2.1 Coverage range values | *GML:: RangeSet* | One (mandatory) |

Table 2: The *Coverage* data structure (Source: [AD5])

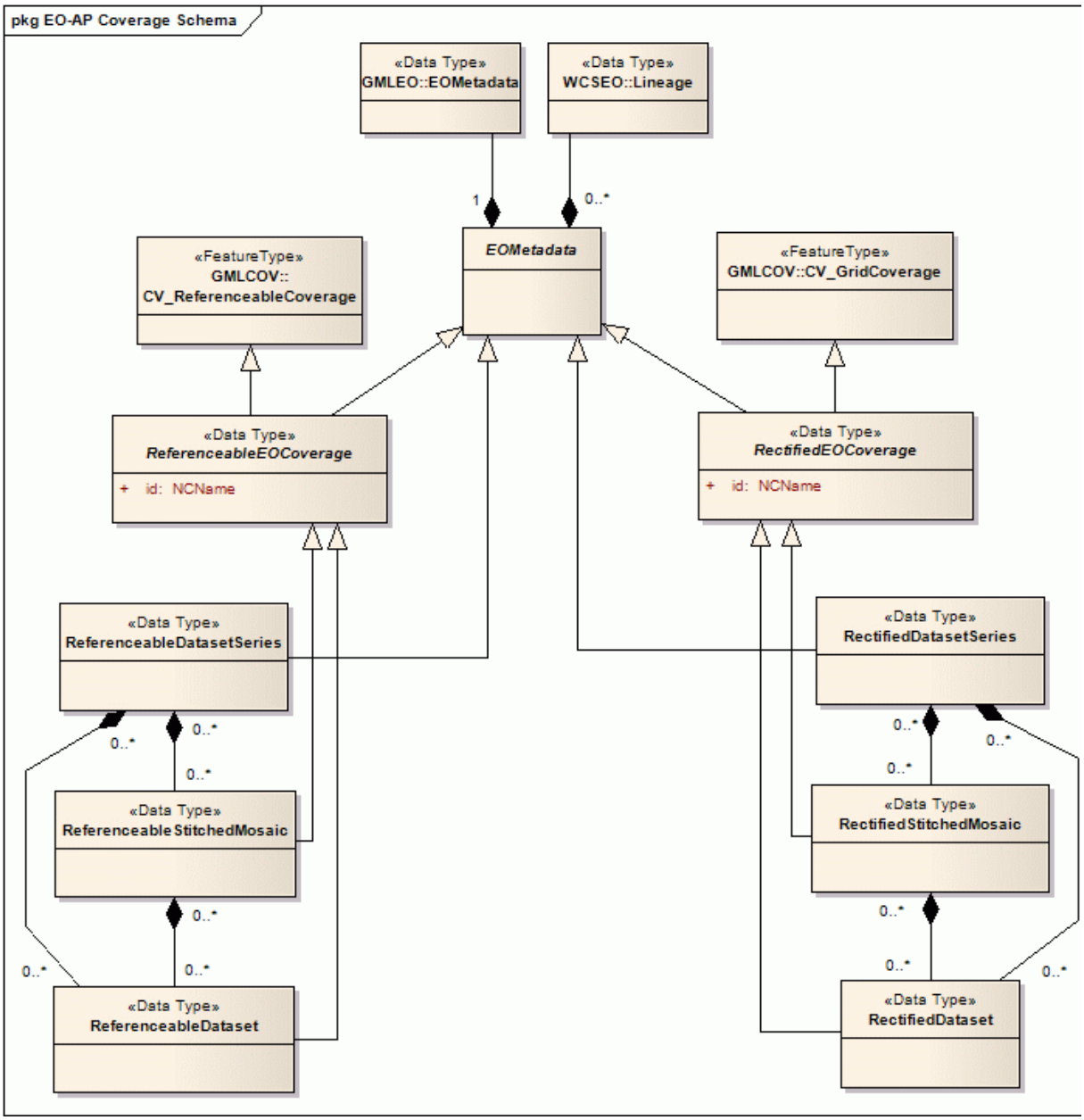


Figure 4: UML Model of EO Application Schema. (Source: [AD12])

| Name | Definition | Data type | Multiplicity |
|---|---|---|---|
| *lineage* | TBD | *WCSEO::Lineage* | zero or more (optional) |
| *metadata* | EO metadata record for this coverage object | *GMLEO::EOMetadata* | one (mandatory) |

Table 3: Components of WCSEO::EOMetadata structure (Source: [AD12])

| Name | Definition | Data type | Multiplicity |
|---|---|---|---|
| eoMetadata | EOMetadata describing this Stitched Mosaic, in particular: its spatio-temporal extent | *GMLEO::Metadata* | one (mandatory) |
| dataset | Dataset referred to by the Stitched Mosaic on hand | *WCSEO::ReferenceableDataset* | zero or more (optional) |

Table 4: Components of WCSEO::ReferenceableStitchedMosaic structure (Source: [AD12])

| Name | Definition | Data type | Multiplicity |
|---|---|---|---|
| *eoMetadata* | EOMetadata describing this Stitched Mosaic, in particular: its spatio-temporal extent | *GMLEO::Metadata* | one (mandatory) |
| *dataset* | Dataset referred to by the Stitched Mosaic on hand | *WCSEO::RectifiedDataset* | zero or more (optional) |

Table 5: Components of WCSEO::RectifiedStitchedMosaic structure (Source: [AD12])

| Name | Definition | Data type | Multiplicity |
|---|---|---|---|
| *eoMetadata* | EO Metadata describing this Dataset Series, in particular: its spatio-temporal extent | *GMLEO::Metadata* | one (mandatory) |
| *stitchedMosaic* | Stitched Mosaic contained in the coverage on hand | *WCSEO::ReferenceableStitchedMosaic* | zero or more (optional) |
| *dataset* | Dataset contained in the coverage on hand | *WCSEO::ReferenceableDataset* | zero or more (optional) |

Table 6: Components of WCSEO:ReferenceableDatasetSeries structure (Source: [AD12])

| Name | Definition | Data type | Multiplicity |
|---|---|---|---|
| *eoMetadata* | EO Metadata describing this Dataset Series, in particular: its spatio-temporal extent | *GMLEO::Metadata* | one (mandatory) |
| *stitchedMosaic* | Stitched Mosaic contained in the coverage on hand | *WCSEO::Rectified-StitchedMosaic* | zero or more (optional) |
| *dataset* | Dataset contained in the coverage on hand | *WCSEO::RectifiedDataset* | zero or more (optional) |

Table 7: Components of WCSEO:RectifiedDatsetSeries structure (Source: [AD12])

## 4.2 Software dynamic architecture

The ODA system Reference Implementation is basically an implementations of OGC Web Services, especially of the new OGC standard for the Web Coverage Service (WCS 2.0) [AD6], a protocol extension [AD7], two format extensions [AD14]and [AD13], and the implementation of the WCS 2.0 Application Profile – Earth Observation [AD9] (WCS EO AP).

In order to handle incoming HTTP request, according to the above mentioned standards, a Server to handle WCS EO AP (see Error: Reference source not found) and WCS 2.0 compliant requests is needed. This Server for WCS EO AP is implemented following the MVC (Model-View-Controller) paradigm as Python Wrapper scripts utilizing the Django and GeoDjango Framework. The MVC view of the components of this Server for WCS EO AP is depicted in Figure 1.

The "Views" are accepting the incoming HTTP requests and forward them, according to the protocol, to the respective service handler. Multiple handlers, one for each protocol (e.g. WCS 2.0, WCS 1.0.0, WMS 1.0.0. WMS 1.1.0, etc.) exist. The incoming request are processed and common functionalities are provided by a set of core functions and by the MapServer. The core functions are also responsible for the communication with a geo-RDBMS and the Storage-Backend system as well as for the ODA system configuration and the communication with the ODA Admin Client (not shown in Figure 1).

The used MapServer has been extended, during the HMA-FO project, to be compliant with the new WCS 2.0 standard. The code of this extension has been submitted to the MapServer community [RD1] and is included in the code trunk of the next MapServer release (MapServer v. 6.0).

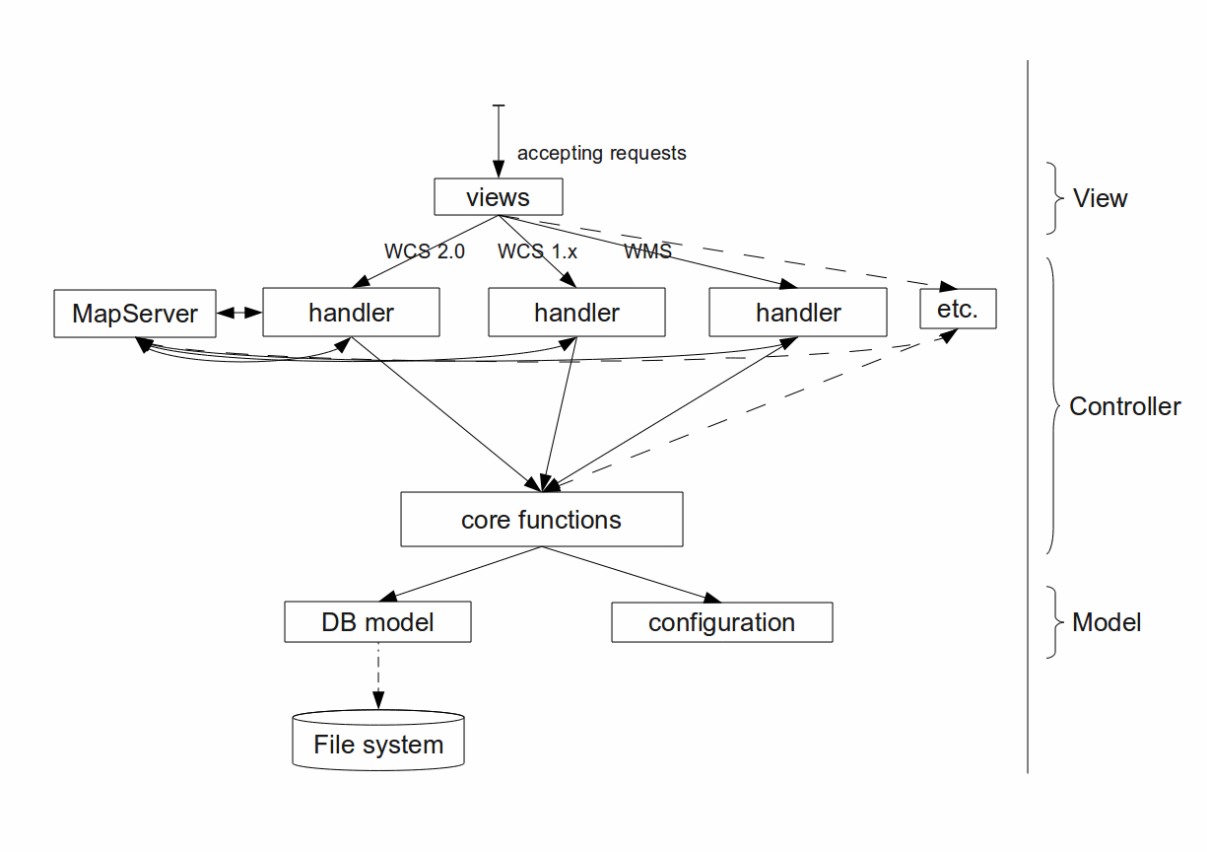


Figure 5: Components of the Server for WCS EO AP

## 4.3 Software behavior

N/A

## 4.4 Interfaces context

Figure 6 and Figure 7 show the context where the ODA system (HMA-FO Reference Implementation) is placed and where entities having a relationship with it.

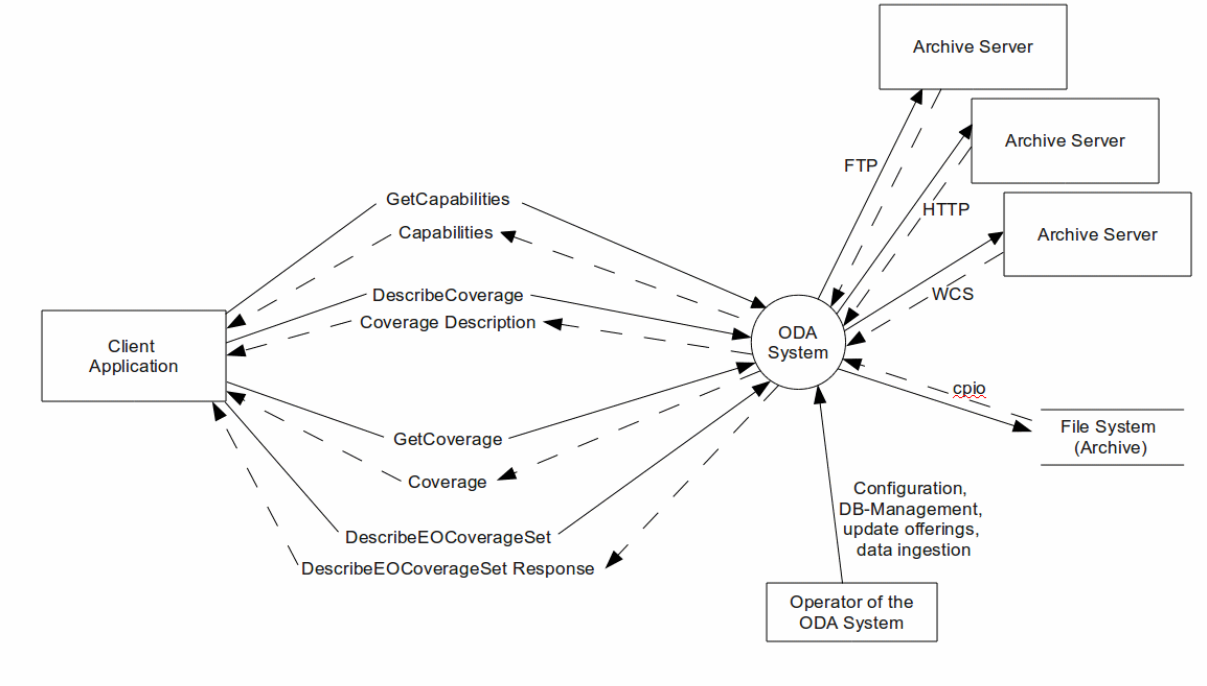


Figure 6: ODA system context diagram

As shown inFigure 6 the ODA system interacts with the following entities:

- WCS Client application, able to send WCS 2.0 [AD6] and WCS EO AP [AD12] compliant request and handle the responses correctly.
- File System containing the datasets (Archive). This represents the possibility that a file system is directly connected (e.g. cpio).
- Archive Server, these represent external Archives which can be accessed utilizing different protocols (HTTP, FTP, WCS) to access the datasets
- Operator of the ODA system is in charge of the operation and configuration of the system and for updating the dataset offerings, as well as for data ingestion into the ODA systems internal data storage.

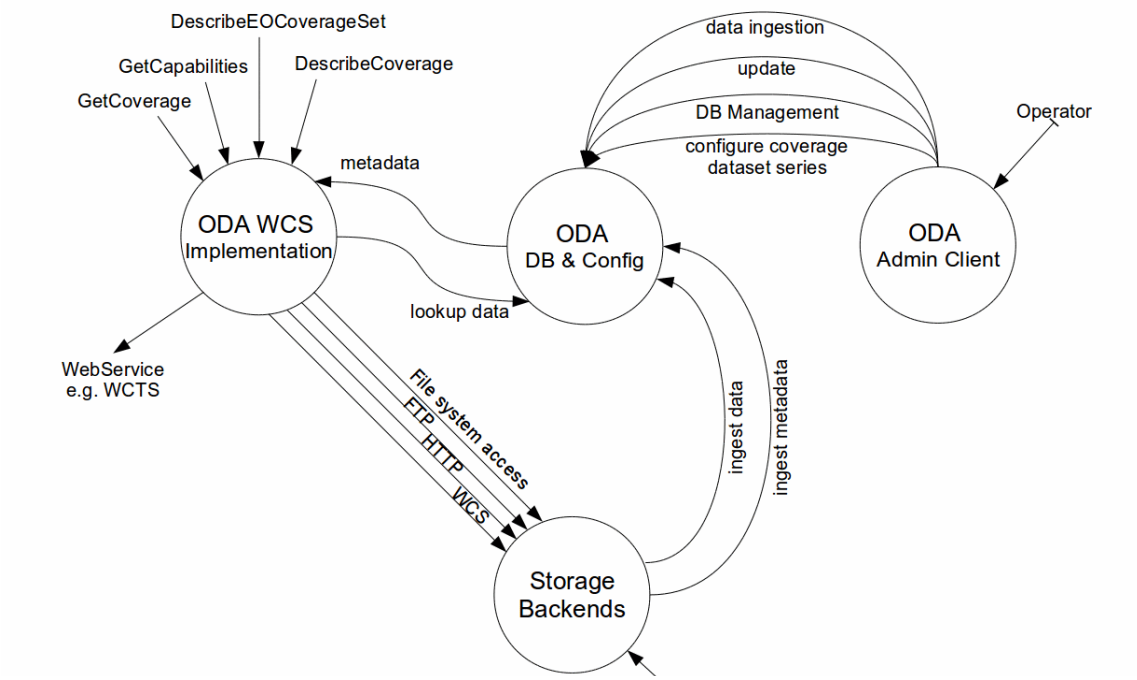


Figure 7: ODA system interface context diagram

The interfaces for communication (Figure 7) between the listed components and entities are:

- Client Application vs. ODA system
  - WCS 2.0 [AD6] and WCS EO AP [AD12] act as ICD which define the request and responses available
  - The User Management Interface [AD9] is not mentioned in the above diagram since for the HMA-FO Reference Implementation, described in this document, it shall not be applied.
- ODA system vs. Archive Server
  - To access external Archive Servers interfaces for FTP, HTTP, or WCS protocol are provided for communication and data access
- ODA system vs. File System
  - The ODA system also foresees that datasets are available on an internal or external File System which may be accessed by calls directly to the Operating System (e.g. cpio).
- Operator vs. ODA system
  - The Operator is in charge for the operation, configuration, and the DB-management of the ODA system. In addition, dataset may be ingested into an inherent data storage.

The following subsections describe all external interfaces shown in the context diagram above. For each interface the following information is provided:

- Name: identifying the interface.
- Description: high level description of the interface.
- Type: specifies the exchange mechanism and the type of the exchanged data.
- Data Structure Description: it provides the formal description of the exchanged data.

### *4.4.1 GetCapabilities*

**Description:**

This operation allows a client to request information about the server's capabilities and coverages offered

**Type:**

Communication over the Internet using HTTP GET with KVP encoding or HTTP POST with xml encoding

**Data Structure description:**

See [AD6], [AD7], and [AD12]

### *4.4.2 DescribeCoverage*

**Description:**

This operation allows a client to submit a list of coverage identifiers and receive, for each identifier, a description of the coverage.

**Type:**

Communication over the Internet using HTTP GET with KVP encoding or HTTP POST with xml encoding

**Data Structure description:**

See [AD6], [AD7], and [AD12]

### *4.4.3 DescribeEOCoverageSet*

**Description:**

A *DescribeEOCoverageSet* request submits an EO coverage identifier together with a spatio-temporal subsetting criterion ("bounding box"). The response to a successful request consists of a set of Dataset descriptions.

**Type:**

Communication over the Internet using HTTP GET with KVP encoding or HTTP POST with xml encoding

**Data Structure description:**

See [AD6], [AD7], and [AD12]

### *4.4.4 GetCoverage*

**Description:**

This operation allows a client to request a coverage comprised of selected range properties at a selected set of spatio-temporal locations, expedited in some coverage encoding format

**Type:**

Communication over the Internet using HTTP GET with KVP encoding or HTTP POST with xml encoding

**Data Structure description:**

See [AD6], [AD7], and [AD12]

### 4.4.5 Data ingestion

**Description:**

This operation involves physically moving the data and metadata files to the Storage-Backend. Subsequently, the ODA system must be configured to present the data and metadata using WCS or WCS EO AP using the ODA Admin Client.

**Type:**

Filesystem operation

**Data Structure description:**

N/A

### 4.4.6 Configure coverage dataset

**Description:**

This operation allows the operator to configure a coverage or a dataset series to publish data and metadata present on a Storage-Backend using WCS or WCS EO AP. The ODA Admin Client provides a Web GUI that allows to configure the system with information about the location and metadata files and possibly additional metadata. The information is ingested into the ODA system database.

**Type:**

Communication over the Internet using HTTP POST

**Data Structure description:**

For the database model, see section 5.4.2.9

### 4.4.7 Update

**Description:**

This operation allows the operator to update information and metadata stored in the ODA system database. The ODA Admin Client provides a Web GUI that allows to configure the system with information about the location and metadata files and possibly additional metadata. The information is ingested into the ODA system database.

**Type:**

Communication over the Internet using HTTP POST

**Data Structure description:**

For the database model, see section 5.4.2.9

### 4.4.8 DB Management

**Description:**

The ODA Admin Client provides a Web GUI that allows to configure database settings independent of specific coverages or dataset series.

**Type:**

Communication over the Internet using HTTP POST

**Data Structure description:**

For the database model, see section 5.4.2.9

### 4.4.9 HTTP

**Description:**

HTTP is one option for connecting the Storage-Backend to the ODA system. Needed data and metadata files are retrieved using the HTTP GET method.

**Type:**

Communication over the Internet using HTTP GET

**Data Structure description:**

N/A

### 4.4.10 FTP

**Description:**

FTP is one option for connecting the Storage-Backend to the ODA system. Needed data and metadata files are retrieved using FTP GET.

**Type:**

Communication over the Internet using FTP

**Data Structure description:**

N/A

### 4.4.11 WCS

**Description:**

Cascading web services are another possibility for connecting the Storage-Backend to the ODA system. Needed coverage data and metadata is retrieved using WCS GetCoverage requests. Possible protocol bindings are KVP over HTTP GET, XML over HTTP POST and SOAP over HTTP POST.

**Type:**

Communication over the Internet using WCS over HTTP

**Data Structure description:**

N/A

### 4.4.12 Cpio

**Description:**

Direct file system access is another option for connecting the Storage-Backend to the ODA system.

**Type:**

Operating System call

**Data Structure description:**

N/A

## 4.5 Long lifetime software

ODA system server developments are made available as OpenSource COTS. These include the enhancements of the MapServer to WCS 2.0 as well as the Server components for WCS EO AP developed as Python Wrapper scripts. The current developments are created for Linux Operating System.

### 4.5.1 ODA system Server

The Reference Implementation of the ODA system server is based on the following software resources:

- Operating System: Linux, preferentially Debian lenny/squeeze/sid (or Ubuntu)
- Apache HTTP server >= 2.2.0
- MapServer 5.6.5
- PostgreSQL >= 8.1 & PostGIS extension >= 1.3.0
- Django Framework >= 1.2 (also including GeoDjango)
- GDAL/OGR >= 1.4

- Python 2.6
- mod_wsgi 3.3
- various libraries like libxml2, libtiff, libgeotiff, etc.

### 4.5.2 WCS client Software

The following auxiliary software should be installed on the machine before deploying the WCS client war file

- OS: Windows, Solaris or Linux (CentOS or Debian or equivalent – both 32 & 64 bits versions supported)
- Servlet container (e.g. Tomcat 6)
- JDK 6
- JAI: JAI version 1.1.3 and JAI imageIO 1.1.0
- RDBMS like Oracle or PostGreSQL/PostGIS for storing the map configurations
- GDAL Version 1.4.0

## 4.6 Memory and CPU budget

### 4.6.1 ODA system Server

The Reference Implementation of the ODA system server needs the following hardware resources:

- CPU: low-end server machine, x86 based PC
- RAM: 4 to 8 GB
- Graphic Adapter: no specific needs
- Disks: RAID-10 Array with a "hot spare" (size depending on test data set storage location - TBD)

### 4.6.2 WCS client Software

We recommend the following minimal hardware configuration for running the WCS client

- CPU: 3.7 GhZ dual Core
- RAM: 4 GB
- Disk Size 70GB RAID

## 4.7 Design standards, conventions and procedures

This section describes the methods applied for the architectural design of the ODA system. Within the project analysis and design phase, both structured and object-oriented modeling and design methods are applied:

- The first levels of the system analysis and decomposition (Architectural Design) are performed by applying structured methods and using Data flow diagrams for representation

- Then the identified leaf components are further decomposed (Detailed Design) using object oriented methods based on UML
- The dynamical behavior of system is described using UML sequence diagrams

### 4.7.1 UML notation

UML uses several kinds of models to describe a system. For this document the following diagrams are considered:

**Deployment diagram**

A deployment diagram models the physical deployment of "artifacts" on "nodes". A deployment diagram could show what hardware components ("nodes") exist and what software components ("artifacts") run on each node and how the different pieces are connected.

**Component diagram**

A component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. A component diagram describes only the static relationship between software components.

**Class diagram**

Class diagrams show the static structure of the model by showing the system's classes, their attributes, and the relationships between the classes. The classes in a class diagram represent both the main objects and or interactions in the application and the objects to be programmed.

**Sequence diagram**

Is a kind of interaction diagram that shows how processes operate with one another and in what order. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

In the following paragraphs the notations used in this document are explained.

### 4.7.2 Deployment diagrams notations

A deployment diagram shows processors, devices, and connections. Each model contains a single deployment diagram that shows the connections between its processors and devices, and the allocation of its processes to processors. The picture below shows an example of deployment diagram where one server and one workstation is connected via a network. The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes.

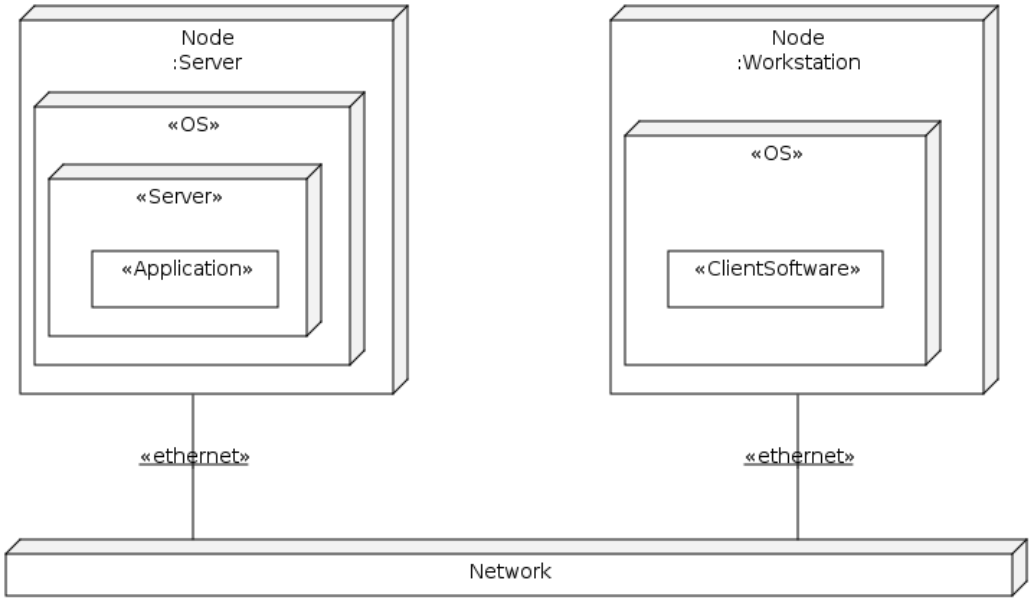


Figure 8: Deployment diagram notation

## 4.7.3 Component diagrams notation

Component diagrams provide a physical view of the current model. A component diagram shows the organizations and dependencies among software components, including source code components, binary code components, and executable components.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.

Component diagrams contain:

**Component packages**

Component packages represent clusters of logically related components, or major pieces of your system. Component packages are analogue to the role played by logical packages for class diagrams. They allow to partition the physical model of the system.

**Components**

A component represents a software module (source code, binary code, executable, DLL, etc.) with a well-defined interface. The interface of a component is represented by one or several interface elements that the component provides. Components are used to show compiler and run-time dependencies, as well as interface and calling dependencies among software modules. They also show which components implement a specific class.

**Interfaces**

An interface specifies the externally visible operations of a class and/or component, and has no implementation of its own. An interface typically specifies only a limited part of the behavior of a class or component.

**Dependency relationships**

The dependency relationship indicates that one entity in a component diagram uses the services or facilities of another. Dependencies in the component diagram represent compilation dependencies. The dependency relationship may also be used to show calling dependencies among components, using dependency arrows from components to interfaces on other components.

The Figure 9 illustrates the basic elements of a component diagram, where an assembly connector bridges a component's required interface (Component A) with the provided interface of another component (Component B); this allows one component to provide the services that another component requires.

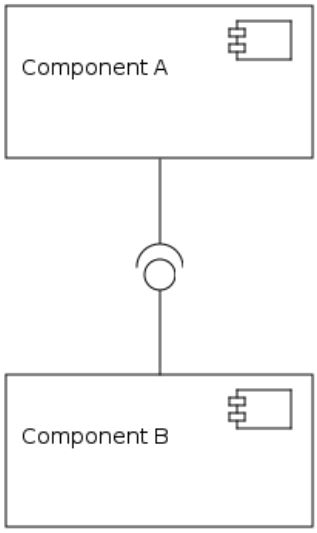


Figure 9: Component diagram notation

## 4.7.4 Class diagrams notations

A class diagram provides a generic description of possible systems. They are being used both for general conceptual modeling of the systematics of the application, and for detailed modeling translating the models into programming code. The classes in a class diagram represent both the main objects and or interactions in the application and the objects to be programmed. In the class diagram these classes are represented with boxes which contain three parts:

- the upper part holds the name of the class
- the middle part contains the attributes of the class

- the bottom part gives the methods or operations the class can take or undertake

Class diagrams contain icons representing classes, interfaces, and their relationships. In particular, class diagrams may contain:

**Logical packages**

Packages serve to partition the logical model of a system. They are clusters of highly related classes that are themselves cohesive, but are loosely coupled relative to other such clusters. You can use packages to group classes, interfaces, and other packages.

**Classes**

A class captures the common structure and common behavior of a set of objects. A class is an abstraction of real-world items. When these items exist in the real world, they are instances of the class, and referred to as objects.

**Interfaces**

An interface specifies the externally visible operations of a class and/or component, and has no implementation of its own. An interface typically specifies only a limited part of the behavior of a class or component.

**Parameterized classes**

A parameterized class is a template for creating any number of instantiated classes that follow its format. A parameterized class declares formal parameters. Other classes, types, and constant expressions can be used as as parameters. The parameterized class itself can not be used as a parameter. Before its objects can be created the parameterized class has to be instantiated.

In its simplest form, parameterized classes can be used to build container classes.

**Instantiated Classes**

An instantiated class is a class formed from a parameterized class by supplying actual values for parameters. An instantiated class is created by supplying actual values for the formal parameters of the parameterized class.

**Association relationships**

An association represents a semantic connection between two classes, or between a class and an interface. Associations are bi-directional; they are the most general of all relationships and the most semantically weak.

**Aggregate relationship**

The aggregate relationship is more specific than association and is used to show a "part-whole" or "part-of" relationship between two classes.

The class at the client end of the aggregate relationship is sometimes called the aggregate class. An instance of the aggregate class is an aggregate object. The class at the supplier end of the aggregate relationship is the part whose instances are contained or owned by the aggregate object.

The aggregate relationship is used for showing that the aggregate object is physically constructed from other objects or that it logically contains another object. The aggregate object has ownership of its parts.

**Generalize/Inherits relationships**

A generalize relationship between classes shows that the subclass shares the structure or behavior defined in one or more super-classes. A generalize relationship is used to show

an "is- a" relationship between classes.

**Instantiates relationships**

An instantiates relationship represents the act of substituting actual values for the parameters of a parameterized class or parameterized class utility to create a specialized version of the more general item. In most cases a uses relationship between the instantiated class and another concrete class that is used as an actual parameter will also be drawn.

**Dependency relationships**

A dependency relationship between two classes, or between a class and an interface, is provided to show that the client class depends on the supplier class/interface to provide certain services, such as:

- the client class accesses a value (constant or variable) defined in the supplier class/interface.
- operations of the client class invoke operations of the supplier class/interface.
- operations of the client class have signatures whose return class or arguments are instances of the supplier class/interface.

Figure 10 depicts the above described items of a class diagram.

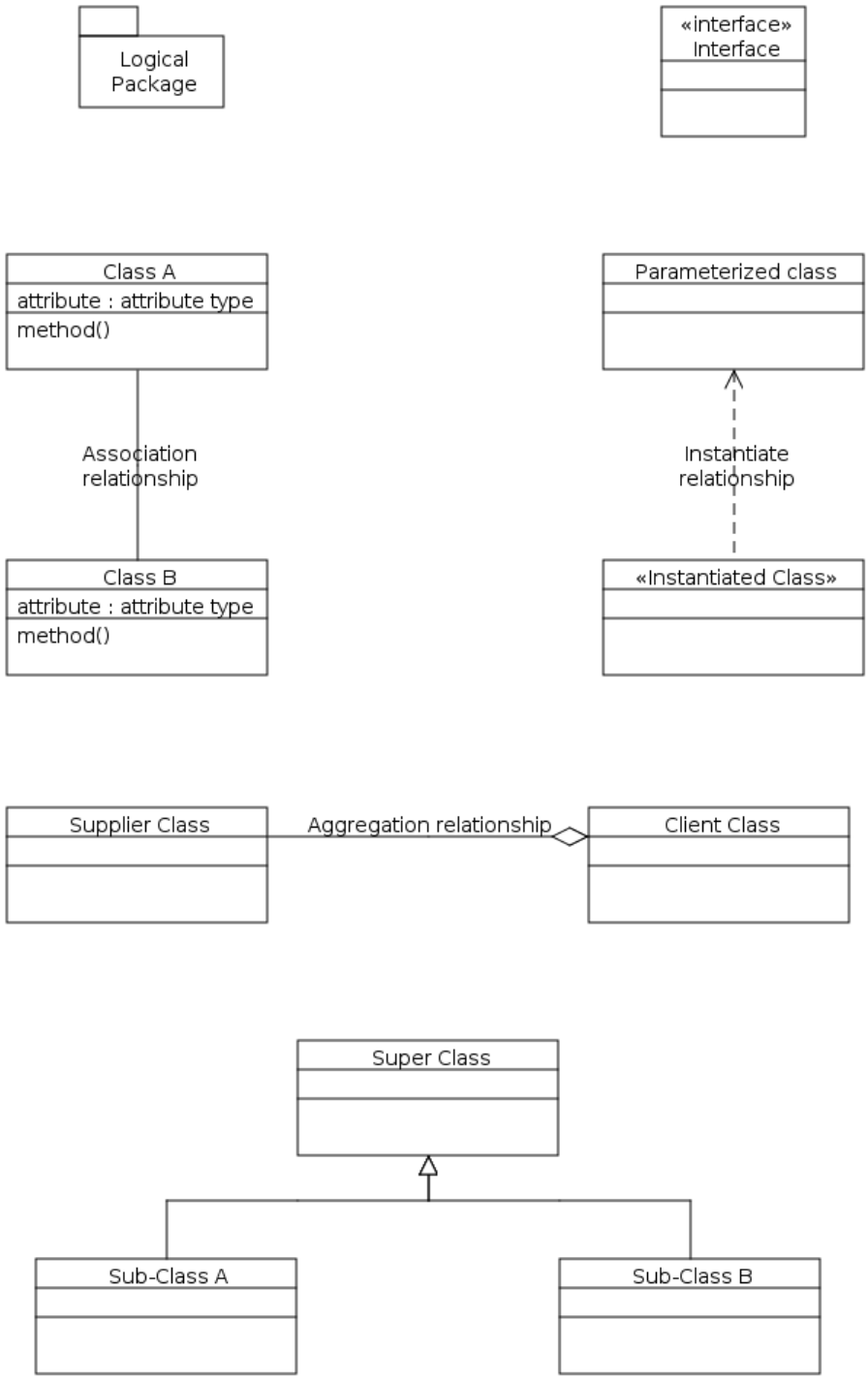


Figure 10: Class diagram notation

### 4.7.5 Sequence diagrams notation

Sequence diagrams are a representation of an interaction between objects. A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

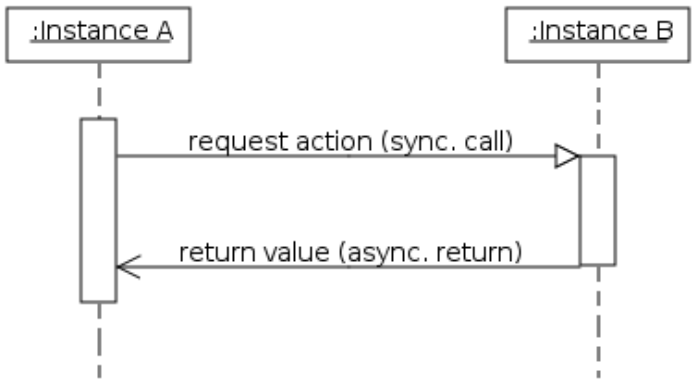


Figure 11: Sequence diagram notation

### 4.7.6 Data Flow diagrams notation

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. It shows what kinds of data will be input to and output from the system, and where the data will come from and go to, as well as where the data will be stored.

In a data flow diagram the following items are considered:

- **Process:** is a process or activity in which data is used or generated
- **External Entity:** represents an external source, user or depository of the data
- **Data Store:** represents an internal physical or electronic repository of data, into and out of which data is stored and retrieved
- **Data Flow** (connector): represents how data flows through the system, in physical or electronic form

Figure 12 illustrates a basic data flow diagram.

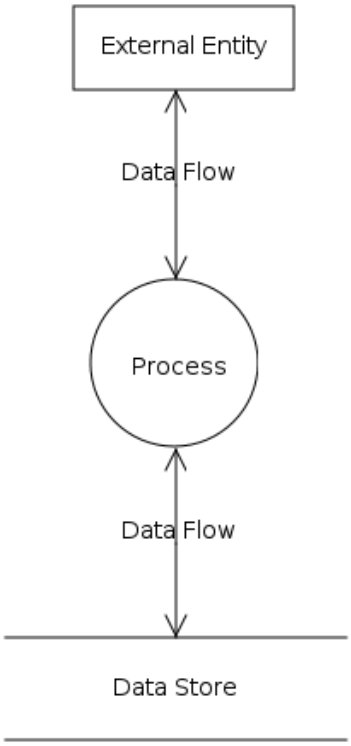


Figure 12: Data flow diagram notation

# 5 Software design

## 5.1 General

This section provides the architectural decomposition of ODA system Reference Implementation describing:

- The different subcomponents function and processing
- The relationships and the interfaces between the different subcomponents
- The dynamical behavior of the system

A system overview is presented in Error: Reference source not found.

## 5.2 Overall architecture

The following, Figure 13 provides a Components overview and Figure 6 a Context diagram of the ODA system Reference Implementation.

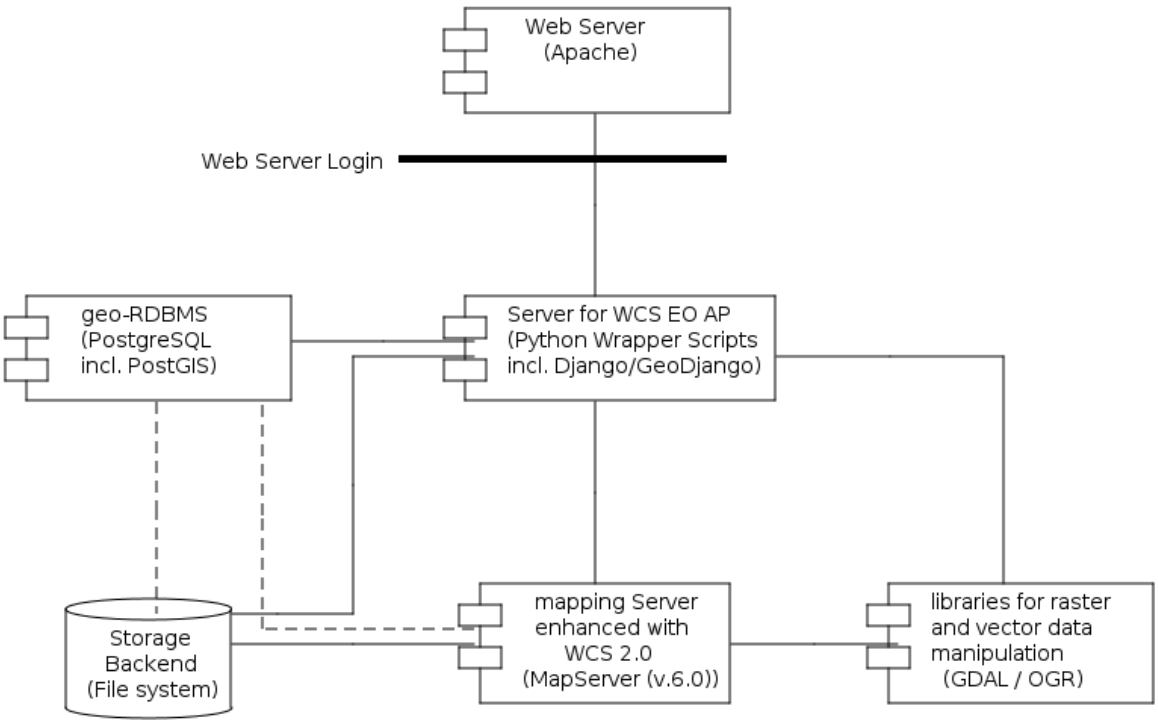


Figure 13: Functional components of the ODA system Reference Implementation

## 5.3 Software components design - General

The ODA system is structured in the following subcomponents:

- Web Server
  - access point to the ODA system
  - provides basic Authentication
- ODA WCS implementation
  - this functional unit listens to incoming HTTP requests and process them according to the requested protocol. The following operations, defined in the WCS 2.0 standard as well as defined in the are WCS EO AP are managed:
    - GetCapabilities
    - DescribeCoverage
    - DescribeEOCoverageSet
    - GetCoverage
  - upon requests arrival a data lookup is performed in the ODA DB
  - metadata is provided in response to data lookup
  - access to Storage-Backends to retrieve the datasets. The I/F supports the following protocols:
    - HTTP
    - FTP
    - WCS
    - direct file system access utilizing Operation System calls i.e. cpio
  - the ODA system provides an I/F to connect to other Web Services (e.g. to WCTS) and may provide data either *byReference* or *byValue*
- ODA DB & Configuration
  - contains the system configuration information which is managed by the operator via the ODA Admin Client
  - may possibly hold offered datasets
- ODA Admin Client
  - client to f be utilized by the ODA operator to configure the ODA system. The HMI provides the following functionalities:
    - management of the ODA RDBMS
    - configuration of the coverage dataset series
    - update mechanism (e.g. after ingestion of new datasets or after configuration changes)
    - ingestion of datasets
- Storage-Backends (internal or external, which are not part of the ODA system)
  - holding the datasets

## 5.4 Software components design - Aspects of each component

### 5.4.1 Web Server

#### 5.4.1.1 Type

Sub-system (package)

#### 5.4.1.2 Purpose

Provides the general access point to the ODA system.
Provides a basic authentication mechanism for the users

#### 5.4.1.3 Function

Delivers the content (XML documents, coverages) provided by the ODA system to the requesting client over the Internet, using the Hypertext Transfer Protocol (HTTP).

#### 5.4.1.4 Subordinates

None

#### 5.4.1.5 Dependencies

None

#### 5.4.1.6 Interfaces

Common Gateway Interface (CGI) / Fast-CGI; WSGI

#### 5.4.1.7 Resources

see sections 4.5 and 4.6

#### 5.4.1.8 References

None

#### 5.4.1.9 Data

User "database" for basic authentication

### 5.4.2 Server for WCS EO AP

#### 5.4.2.1 Type

Sub-system (Python Wrapper scripts)

#### 5.4.2.2 Purpose

The Python wrapper scripts provide the application logic of the ODA system. They bring together configuration metadata stored in the ODA system database, coverage processing routines provided by MapServer and libraries for raster and vector data manipulation, and the actual data residing on the Storage-Backends.

#### 5.4.2.3 Function

Python wrapper scripts are called by the web server through its WSGI interface. Internally, the request is forwarded to handler objects which belong to three levels. The first level are service handlers, which process every incoming request to a given OWS interface (e.g. WCS, WMS). They forward the request to a version handler which provides processing methods common to all operations of a protocol version (e. g. exception handling). The third level are operation handlers; that's where the specific logic for request handling resides.

Handlers use a set of core functions which provide common functionality, e.g. for the communication with a geo-RDBMS and the Storage-Backend system as well as for the ODA system configuration and the communication with the ODA Admin Client.

There are handlers for the following WCS 2.0 EO AP operations and their respective WCS 1.0 and 1.1 counterparts (if they exist):

- *GetCapabilities*
- *DescribeCoverage*
- *DescribeEOCoverageSet*
- *GetCoverage*

Furthermore there handlers for WMS 1.0, 1.1 and 1.3 for the following operations

- *GetCapabilities*
- *GetMap*

#### 5.4.2.4 Subordinates

- Service Handlers – WCS, WMS
- Version Handlers – WCS 2.0, 1.0, 1.1, WMS 1.0, 1.1, 1.3
- Operation Handlers – WCS *GetCapabilities*, WCS *DescribeCoverage*, WCS 2.0 EO AP *DescribeEOCoverageSet*, WCS *GetCoverage*; WMS *GetCapabilities*, WMS *GetMap*
- Core Functions

#### 5.4.2.5 Dependencies

The Web Server shall be up and running

The Database shall be up and running.

The Mapping Server shall be available

The Libraries for Raster and Vector Data Manipulation shall be available.

#### 5.4.2.6 Interfaces

- WCS *GetCapabilities*
- WCS *DescribeCoverage*
- WCS 2.0 EO AP *DescribeEOCoverageSet*
- WCS *GetCoverage*
- WMS *GetCapabilities*
- WMS *GetMap*

#### 5.4.2.7 Resources

see sections 4.5 and 4.6

#### 5.4.2.8 References

[AD6], [AD7], [AD12], [AD13], [AD14]

#### 5.4.2.9 Data

Configuration data and DB model

### 5.4.3 Mapping Server

#### 5.4.3.1 Type

Sub-system (package)

#### 5.4.3.2 Purpose

MapServer is a software for handling OWS requests.

#### 5.4.3.3 Function

Within the scope of the HMA system, MapServer is used to deliver coverages according to WCS 2.0. It is configured by the Python wrapper scripts through its MapScript API.

#### 5.4.3.4 Subordinates

None

#### 5.4.3.5 Dependencies

Libraries for Raster and Vector Data Manipulation shall be available.

#### 5.4.3.6 Interfaces

None

#### 5.4.3.7 Resources

see sections 4.5 and 4.6

#### 5.4.3.8 References

#### 5.4.3.9 Data

Configuration data from the ODA system database; raster data retrieved from the Storage-Backends.

### 5.4.4 Libraries for Raster and Vector data manipulation

#### 5.4.4.1 Type

Sub-system (package)

#### 5.4.4.2 Purpose

The ODA System uses GDAL/OGR as library for raster and vector data manipulation. It is used by the ODA WCS EO AP Server and by MapServer to support raster data I/O.

#### 5.4.4.3 Function

GDAL/OGR offers a common API for accessing different raster and vector formats. It is used by the Python wrapper scripts to examine raster data and by MapServer for I/O operations on raster data.

#### 5.4.4.4 Subordinates

None

#### 5.4.4.5 Dependencies

None

#### 5.4.4.6 Interfaces

None

#### 5.4.4.7 Resources

see sections 4.5 and 4.6

#### 5.4.4.8 References

#### 5.4.4.9 Data

For a list of supported raster data formats, see http://www.gdal.org/formats_list.html.

### 5.4.5 geo-RDBMS

#### 5.4.5.1 Type

Sub-system (package)

#### 5.4.5.2 Purpose

The ODA system uses PostgreSQL/PostGIS as its geospatially enabled database backend. It is used for the storage of the configuration and basic coverage metadata.

#### 5.4.5.3 Function

The ODA WCS EO AP Server (Python wrapper scripts) call the database through its native API and its Python bindings using the Django framework.

#### 5.4.5.4 Subordinates

None

#### 5.4.5.5 Dependencies

None

#### 5.4.5.6 Interfaces

None

#### 5.4.5.7 Resources

see sections 4.5 and 4.6

#### 5.4.5.8 References

None

#### 5.4.5.9 Data

The Database Model used for the ODA system is shown in Figure 14.

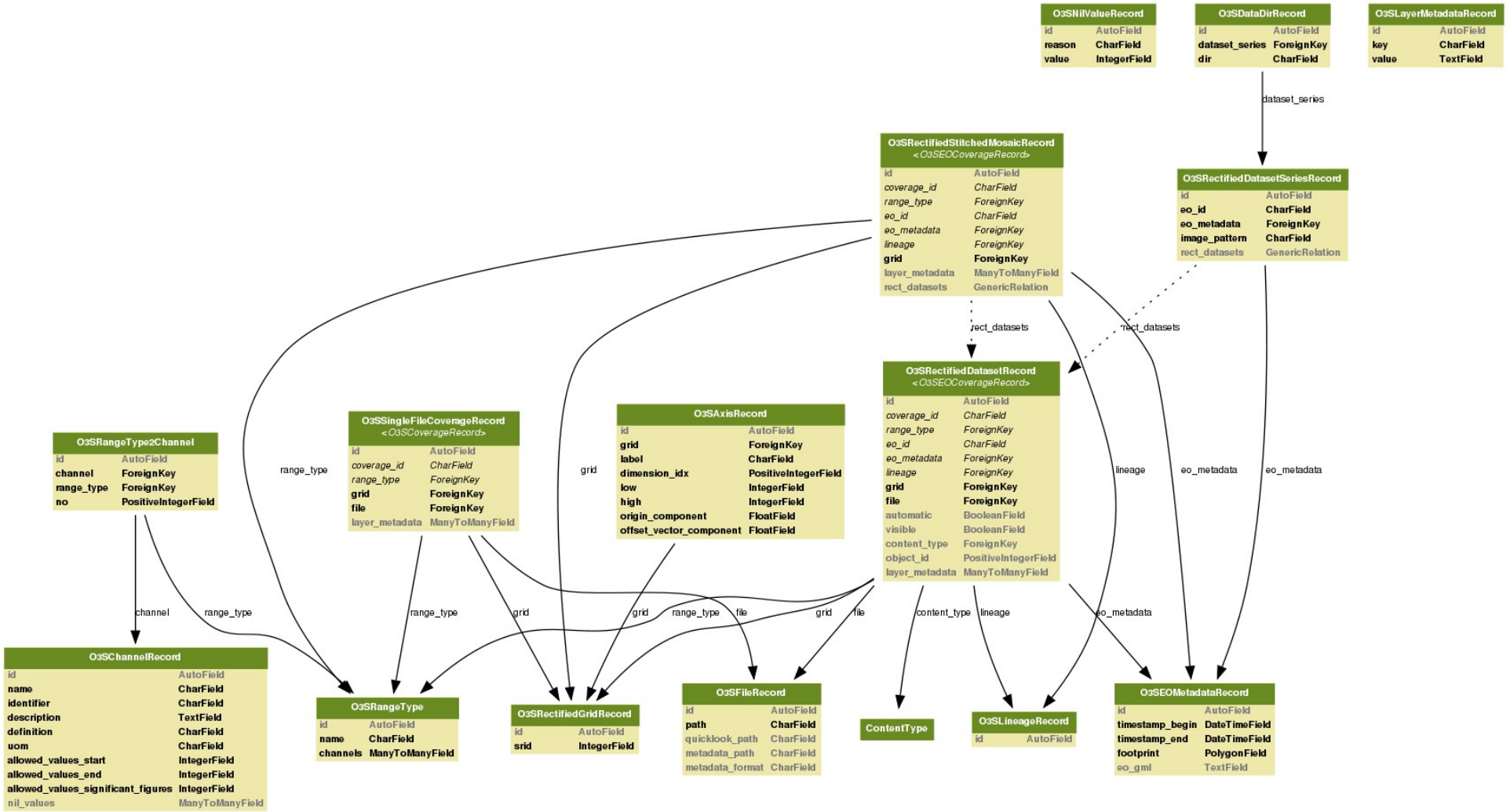


Figure 14: ODA system database model as UML class diagram

# 5.5 Dynamical Model

This section summarizes the main scenarios for accessing and operating the ODA system Reference Implementation of the ODA system.

## 5.5.1 Accessing the ODA system utilizing WCS 2.0 compliant requests

The interactions between a User Client and the ODA system, as defined in [AD6], can be summarized as follows:

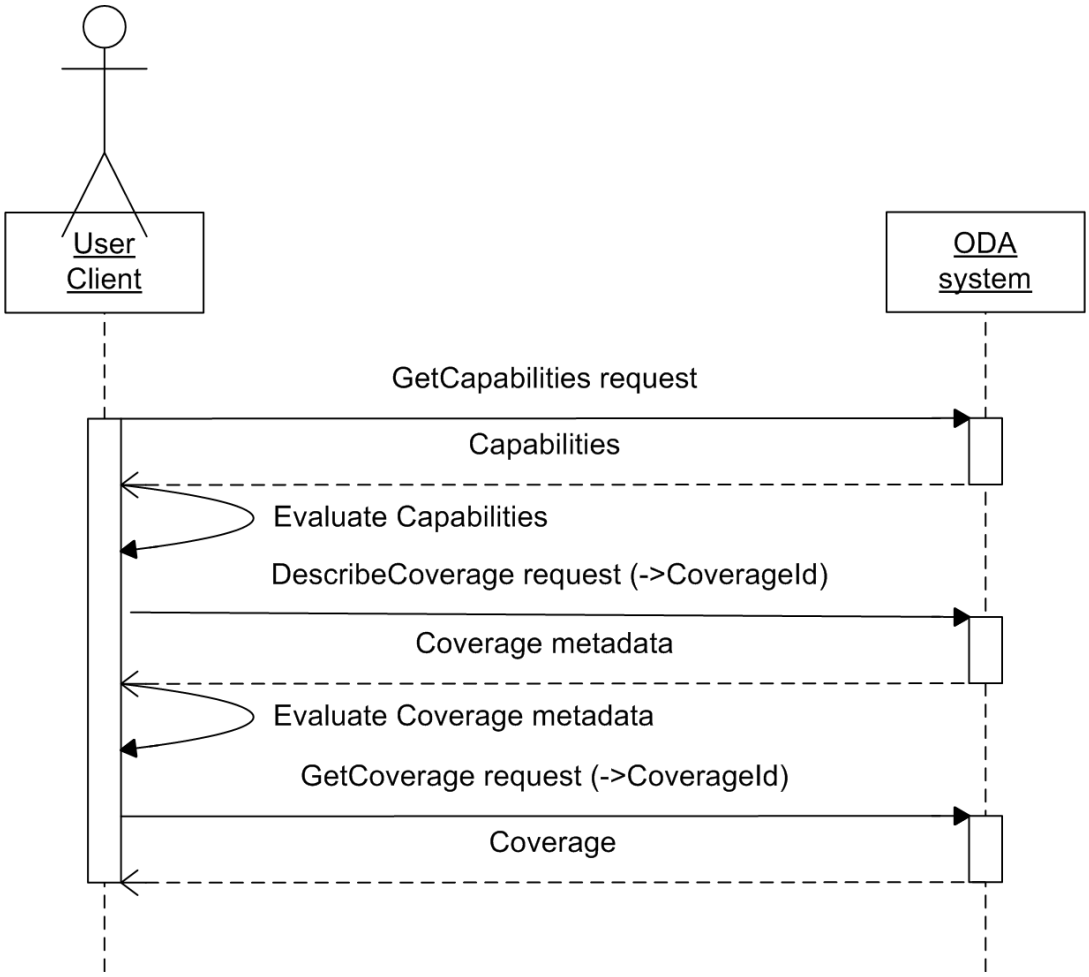


Figure 15: General request scenario to access the ODA system in a WCS 2.0 compliant way

The user disposes WCS 2.0 compliant requests to the ODA system. The ODA system honors valid requests with WCS 2.0 standard conform results.

The required syntax to issue a valid request is described in detail in [AD6] and in the

currently existing WCS 2.0 extensions supported by the Reference Implementation i.e. [AD7], [AD5], [AD13].

- The User client issues a valid *GetCapabilities* request
- The ODA system processes the request and returns a Capabilities document
- The User evaluates the received Capabilities document
- The User selects a *CoverageID* of interest and submits a *DescribeCoverage* request providing the *CoverageID*
- The ODA system processes the request and returns the Coverage metadata for the respective *CoverageID*
- The User evaluates the received Coverage metadata
- The User submits a valid *GetCoverage* request providing the *CoverageID*
- The ODA system processes the request and return the respective coverage to the user

### 5.5.2 Accessing the ODA system utilizing WCS EO AP compliant requests to access a Dataset Series

This scenario summarizes the interactions between a User Client and the ODA system for valid requests utilizing the WCS EO AP as currently described in [AD12].

The WCS EO AP provides, besides the standard WCS 2.0 conformant request types, an addition request type named *DescribeEOCoverageSet*. This request type allows to query Dataset Series as defined in [AD12] and provides *CoverageDescription* on the items queried. Based on this information the user can issue valid *GetCoverage* requests to access the datasets contained or referenced in the Dataset Series. It therefore provides a mechnism to access e.g. coverages representing a Time-Series.

- The User client issues a valid *GetCapabilities* request
- The ODA system processes the request and returns a Capabilities document
- The User evaluates the received Capabilities document
- The User selects a *CoverageID* of interest and submits a *DescribeCoverage* request providing the *CoverageID*
- The ODA system processes the request and returns the Coverage metadata for the respective *CoverageID*
- The User evaluates the received Coverage metadata
- The User selects a *eoID* of a Dataset Series of interest and submits a valid *DescribeEOCoverageSet* providing the the *eoId.* The *DescribeEOCoverageSet* request further allows to specify spatial and temporal limits (I.e AOI and TOI) as constrains (see [AD12] for more details)
- The ODA system processes the request and provides *CoverageDescriptions* for the datasets described by *eoId* and matching the the provided limits
- The User evaluates the received *CoverageDescription*

- The User repeatedly submits valid *GetCoverage* requests for each *CoverageID*
- The ODA system processes each request and returns the respective coverage to the user.

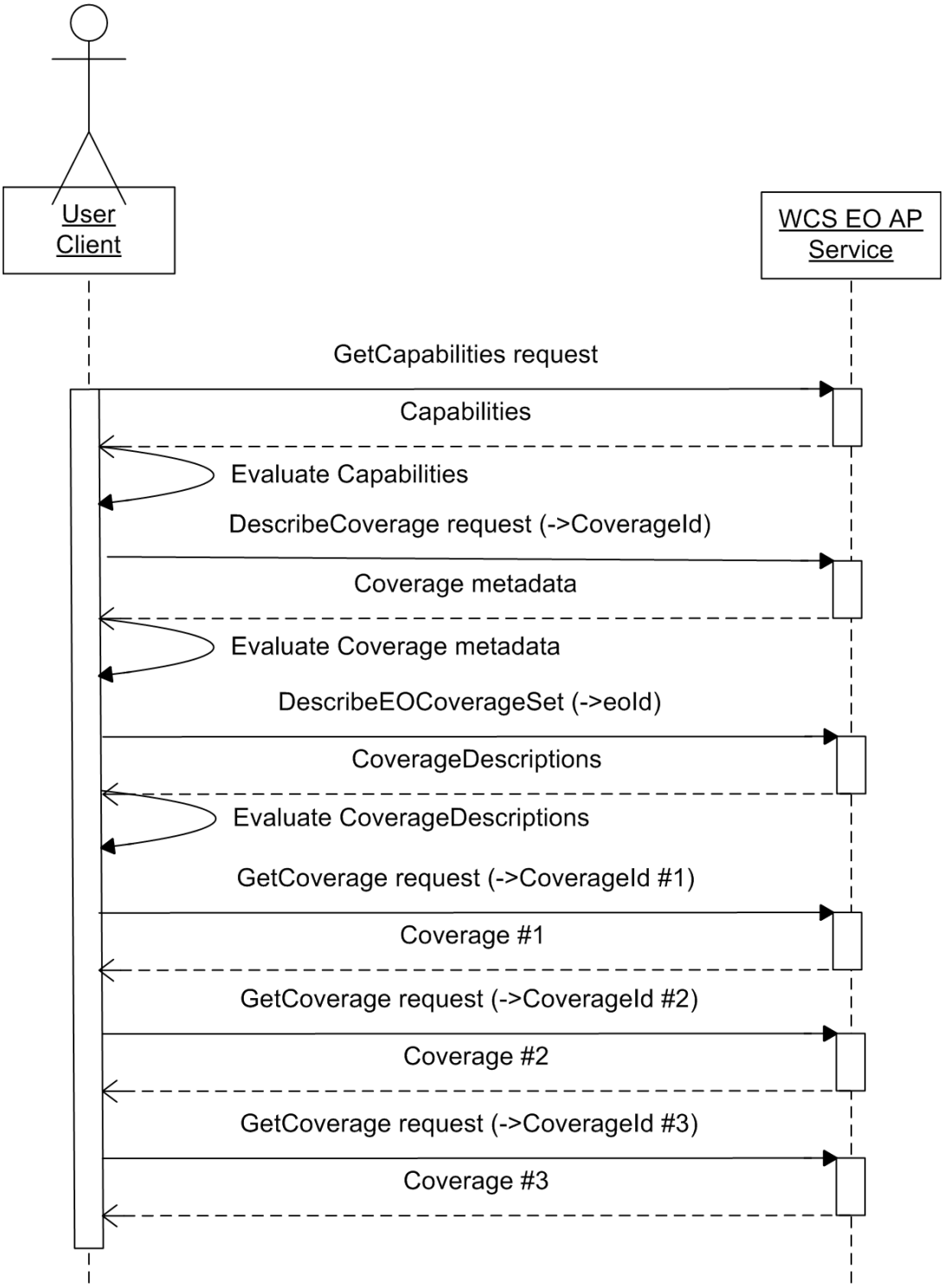


Figure 16: General request scenario to the ODA system to access a Dataset Series in a WCS EO AP compliant way

### 5.5.3 ODA system configuration and management

This scenario summarizes the configuration and management functionalities provided by the ODA system and accessible via the ODA Admin Client.

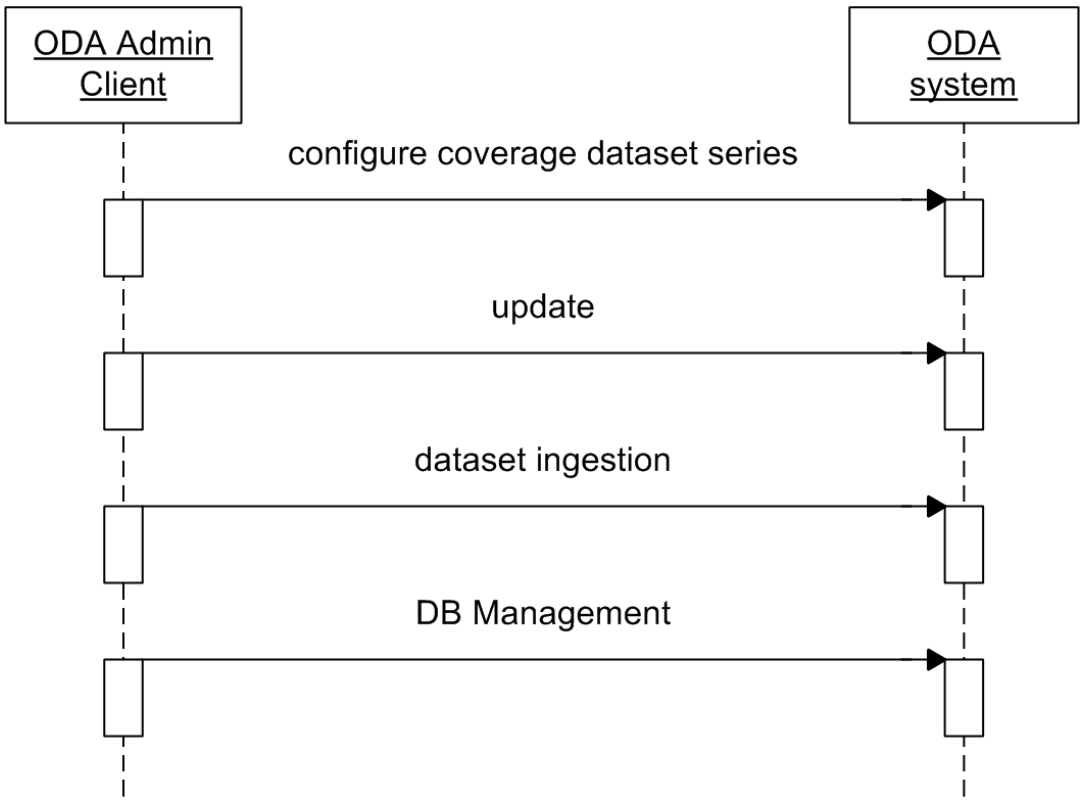


Figure 17: Configuration and Management scenario

The ODA Admin Client provides functionalities to:

- manage the geo-RDBMS
- ingest datasets into the ODA system
- update the information regarding the provided datasets
- configure coverage dataset series

## 5.6 Internal interface design

In the following section all the ODA system internal interfaces presented in Figure 13 are listed and described.

For each interface the following information are provided:

- Name: identifies the interface
- Type: specifies the exchange mechanism and the type of the exchanged data.

- Description: high level description of the interface.

### 5.6.1 Web Server vs. WCS EO AP Server

#### 5.6.1.1 Type

API Invocation

#### 5.6.1.2 Description

The Web Server component invokes the WCS EO AP Server using the Python Web Server Gateway Interface (WSGI; [RD20]). Thus, the Web Server provides the application (WCS EO AP Server) with environment variables containing the request details.

### 5.6.2 WCS EO AP Server vs. geo-RDBMS

#### 5.6.2.1 Type

Exchange of SQL and database record data using API Invocation

#### 5.6.2.2 Description

The WCS EO AP Server calls the geo-RDBMS PostgreSQL/PostGIS using the Django framework which uses the underlying Python binding of the native database API. The WCS EO AP Server sends SQL queries to the database which responds with database record data structures.

### 5.6.3 WCS EO AP Server vs. Mapping Server

#### 5.6.3.1 Type

API Invocation

#### 5.6.3.2 Description

The WCS EO AP Server calls the MapServer component using the Python binding of its MapScript API. The WCS EO AP Server first configures the coverage and then dispatches a customized OWS request to MapServer, which responds with the appropriate OWS response.

### 5.6.4 WCS EO AP Server vs. Libraries for Raster and Vector Data Manipulation

#### 5.6.4.1 Type

API Invocation

#### 5.6.4.2 Description

The WCS EO AP Server calls the libraries for raster and vector data manipulation in order to examine raster data, e.g. for retrieving domain set and range type information. The WCS EO AP Server uses the Python bindings of the GDAL/OGR API to achieve this.

## 5.6.5 WCS EO AP Server vs. Storage-Backend

### 5.6.5.1 Type

Data Exchange using operating system functions, Internet protocols

### 5.6.5.2 Description

The WCS EO AP Server retrieves data from the Storage-Backend in order to examine it or to pass it on to MapServer for coverage computation and image rendering. The Storage-Backend access mechanisms available are:

- FTP
- HTTP
- cpio (file system access)

For FTP and HTTP, the data has to be cached locally in order to allow operations on it.

## 5.6.6 Mapping Server vs. Storage-Backend

### 5.6.6.1 Type

Data Exchange using operating system functions, Internet protocols

### 5.6.6.2 Description

The MapServer component retrieves data from the Storage-Backends in order to compute coverages or to render maps. The Storage-Backend access mechanisms available are:

- cpio (file system access)
- WCS

Data from remote Storage-Backends has to be cached locally for MapServer to operate on it.

## 5.6.7 Mapping Server vs. Libraries for Raster and Vector Data Manipulation

### 5.6.7.1 Type

API Invocation

### 5.6.7.2 Description

The MapServer component accesses geospatial raster data using the GDAL/OGR C API.

# 6 Requirements to design components traceability

| WCS server requirements | | |
|---|---|---|
| Identifier | Description | Software Component |
| SR_ODA_GEN_010 | The Online Data Access System (ODA) shall be accessible over the network. | Web Server |
| SR_ODA_GEN_011 | The ODA system shall be implemented using standardized interfaces. | Mapping Server, Server for WCS EO AP |
| SR_ODA_GEN_050 | The ODA system shall respect the WCS EO AP as far as currently defined.<br>Note: The definition of the WCS EO AP is part of this project and is currently under revision by OGC's WCS.SWG. | Server for WCS EO AP |
| SR_ODA_GEN_070 | The ODA system shall be able to be configured to work in a stand-alone mode.<br>Note: Stand-alone mode, in this respect, means it should be configurable to work in a restricted environment with a limited access to data resources. | geo-RDBMS, Storage Backend |
| SR_ODA_GEN_130 | The ODA system shall provide access to the data content with at least one set of metadata.<br>Note: The amount and the content of metadata provided for a Dataset depends on the type of Dataset. | Mapping Server<br><br>EO metadata? → Server for WCS EO AP |
| SR_ODA_GEN_140 | The ODA system shall provide at least one interface to be integrated with an End User environment.<br>Note: The data is accessible directly out of End User's application (e.g. a GIS system) using either Web Service, API, or Add-on depending on the respective client. | Mapping Server, Server for WCS EO AP |
| SR_ODA_CAP_010 | The ODA system shall allow the download of full datasets. | Mapping Server |
| SR_ODA_CAP_020 | The ODA system shall allow the selection of AOIs (sub-datasets). | Mapping Server |

| | | |
|---|---|---|
| SR_ODA_CAP_030 | The ODA system shall allow the download of sub-datasets (i.e. trim functionality - extract data from files and mosaics). | Mapping Server |
| SR_ODA_CAP_040 | The ODA system shall allow the selection of time-slots. | Server for WCS EO AP |
| SR_ODA_CAP_041 | The ODA system shall allow the download of selected time-slots. | Mapping Server |
| SR_ODA_CAP_050 | The ODA system shall allow the selection of time series.<br>Note: sub-setting within a time range e.g files between a start and an end date set by the use | Server for WCS EO AP |
| SR_ODA_CAP_051 | The ODA system shall allow the download of time series. | Mapping Server |
| SR_ODA_CAP_070 | The ODA system shall allow to download the datasets in different projections and datums. | Mapping Server |
| SR_ODA_CAP_080 | The ODA system shall allow to download the datasets in various file formats.<br>Note: In particular one of GeoTIFF, netCDF, or JPEG2000 shall be supported at least. | Mapping Server |
| SR_ODA_CAP_090 | The ODA system shall allow the download of different channels (bands) of a dataset (e.g. cloud coverage, and other masks) | Mapping Server |
| SR_ODA_CAP_100 | When used as a view service the ODA system shall allow the access to different channels of a dataset (e.g. bands, bitmasks). | Mapping Server |
| SR_ODA_CAP_110 | The ODA system shall be able to access Grid coverages (more precisely, quadrilateral grid coverages). | Mapping Server |
| SR_ODA_CAP_140 | The ODA system shall allow access to the data in its original coordinate reference system. | Mapping Server |
| SR_ODA_CAP_150 | The ODA system shall support the use of CRS defined by official EPSG codes (version included in Proj 4.7.0) (for transformations during the download of | Mapping Server |

| | | |
|---|---|---|
| | datasets). | |
| SR_ODA_CAP_160 | Downloading shall allow transfer with CRS transformation. The product downloaded is transformed into another CRS than that of the source. The product is transformed "on the fly". | Mapping Server |
| SR_ODA_CAP_190 | The ODA system shall be able to be reconfigured (e.g. based on DB input) to enable the instantiation of the "rolling archive" (flowing dataset e.g. FIFO) concept. | Server for WCS EO AP |
| SR_ODA_CAP_210 | The system shall be designed to be able to hold multi-layer raster datasets, e.g. multispectral EO ortho-imagery, error layers, bitmasks. | Mapping Server |
| SR_ODA_CAP_230 | The ODA system shall handle data at various processing levels as defined by CEOS. | Mapping Server |
| SR_ODA_CAP_250 | The ODA system shall not break the general usage e.g. GetCoverage of the implemented solution.<br>Note: Results shall be viewable in a client e.g. Web-Browsers | Mapping Server, Server for WCS EO AP |
| SR_ODA_CAP_260 | It shall be possible for an ODA system to reuse available EO metadata (e.g. from a catalog). | Server for WCS EO AP |
| SR_ODA_CAP_270 | The ODA system shall be able to add a new datasets (file/mosaic) to a repository. | Server for WCS EO AP (ODA Admin Client) |
| SR_ODA_CAP_380 | Downloading shall allow the transfer of an extraction or of the complete dataset with a minimum set (TBD) of metadata.<br>Note: The End User discovers and visualizes the product. The End User wants either a complete download of the product or only an extract for final publishing off line. This final publishing should be a paper map or a backdrop display on a device (PDA/Mobile...). The minimum metadata required shall provide copyright, acquisition date and CRS. | Mapping Server |

| | | |
|---|---|---|
| SR_ODA_CAP_390 | The ODA system shall allow the download of the data in its original unmodified data type. | Mapping Server |
| SR_ODA_CAP_400 | The ODA system shall support dataset collections. | Server for WCS EO AP |
| WCS client requirements | | |
| Identifier | Description | Software Component |
| SR_ODA_CAP-CLI_010 | The WCS client shall be able to visualize datasets that are offered via Web Coverage Services. | WebMapViewer |
| SR_ODA_CAP-CLI_020 | Datasets shall be shown in the clients layer manager.<br>Note: The following general layer management functions shall be available for datasets:<br>Hide/Show dataset layers<br>Change transparency<br>Re-order layers<br>Zoom to extent of layers | WebMapViewer |
| SR_ODA_CAP-CLI_030 | The layer manager shall contain a button that allows the visualization of the dataset metadata that is offered via the DescribeCoverage operation. Both presentation in HTML and downloading in XML shall be provided.<br>Note: The information that is returned in XML shall be rendered to HTML using a fixed XSLT style-sheet based upon the metadata specified in the WCS EO Application Profile. | WebMapViewer |
| SR_ODA_CAP-CLI_040 | The dataset layer shall be visualized on the WCS client's map whereby the WCS client will request the appropriate extent and resolution from the WCS.<br>Note: In other words, the WCS client will only download the part of the image that is required for presentation of the result. | WebMapViewer |
| SR_ODA_CAP-CLI_050 | The WCS client shall offer a "Coverage Parameters" form where the user can specify several options with respect to the visualization of the datasets. | WebMapViewer |

| | | |
|---|---|---|
| SR_ODA_CAP-CLI_060 | The "Coverage Parameters" form shall allow the user to specify the bands which shall be mapped to Red, Green and Blue values<br>Note: If the dataset is to be presented in (true or false) color. | WebMapViewer |
| SR_ODA_CAP-CLI_070 | The "Coverage Parameters" form shall allow the user to specify the dataset band to be used if the image is to be presented in gray scales. | WebMapViewer |
| SR_ODA_CAP-CLI_080 | The "Coverage Parameters" form shall allow the user to specify the (background) color that shall be made transparent. | WebMapViewer |
| SR_ODA_CAP-CLI_090 | The "Coverage Parameters" form shall allow the user to specify simple "image enhancements" (histogram stretches). | WebMapViewer |
| SR_ODA_CAP-CLI_100 | The "Coverage Parameters" form shall allow the user to specify the time of interest in case a Dataset Series that contains multiple times. | WebMapViewer |
| SR_ODA_CAP-CLI_110 | The "Coverage Parameters" form shall allow the user to download the dataset, by specifying:<br>the extent of the dataset to be downloaded via choosing either the full dataset extent, via specifying an AOI on the map or alternatively by the specification of the current map view<br>the output format (from the list of formats that the dataset supports)<br>the resolution in X and Y direction (default values presented in non-exponential notation and given in the native CRS)<br>or alternatively to the resolution, the width and height expressed in pixels<br>the output CRS (default value being the native CRS)<br>the interpolation method<br>the time of interest | WebMapViewer |

| | | |
|---|---|---|
| SR_ODA_CAP-CLI_120 | The user shall be offered the possibility to add dataset layers to the WCS client by using the add layer functionality of the layer manager.<br>Note: A form shall be presented to the user where he needs to provide the base URL of the Service, set the Service Type to WCS, select the appropriate WCS Version and select the appropriate WCS profile or extension. | WebMapViewer |
| SR_ODA_CAP-CLI_130 | If the WCS Service exposes the individual datasets in its capabilities, the WCS client shall list these to the user, to allow the selection of one or more datasets which shall be added to the map. | WebMapViewer |
| SR_ODA_CAP-CLI_140 | If the WCS Service exposes a link to an EO Extension Package of ebRIM CSW catalog (, ) in its capabilities, the WCS client should allow the user to call the appropriate catalog client to allow the election of the appropriate dataset.<br>Note: This requirement is to be confirmed in function of decisions taken when developing the WCS EO AP. The integration mechanism that shall be adopted is the same as used for linking the services catalog to a WCS client and assumes the availability of an appropriate EO extension Package of ebRIM CSW catalog on another Web Portal server (to avoid session sharing between two WCS client instances). The two portal servers need to be installed within the same domain to avoid JavaScript security issues. | WebMapViewer |
| SR_ODA_CAP-CLI_150 | The Service provider shall be offered the same dataset functionality as the end user when setting up a WCS client configuration. | WebMapViewer |
| SR_ODA_CAP-CLI_160 | The Service provider shall be equipped with a JavaScript API that allows the service provider to configure an Web Portal Service instance in order to add dataset at run time. | WebMapViewer |

| | | |
|---|---|---|
| | To be used for instance from within a CIM (services or data) or EO EP of ebRIM CSW catalog (, ). | |
| SR_ODA_CAP-CLI_170 | The Service provider shall be equipped with a mechanism that allows the service provider to configure an Web Portal Service installation to add datasets upon start-up of the WCS client viewer.<br>Note: To be used for instance to allow display of datasets that are the result of Web Coverage Processing Services. | WebMapViewer |
| Performance | | |
| Identifier | Description | Software Component |
| SR_ODA_PE_010 | The total response time of a GetCapabilities request shall not exceed 5 seconds. | Mapping Server, Server for WCS EO AP |
| SR_ODA_PE_020 | The total response time of a DescribeCoverage request shall not exceed 5 seconds. | Mapping Server, Server for WCS EO AP |
| SR_ODA_PE_030 | Since the total response time of a DescribeEOCoverageSet request strongly depends on the size of the targeted Dataset Series a total response time can not be provided. However, it is recommended that the response time should not exceed 20 seconds. | Server for WCS EO AP |
| Interface requirements | | |
| Identifier | Description | Software Component |
| SR_ODA_IF_010 | The ODA system shall support the access via HTTP KVP . | Server for WCS EO AP |
| SR_ODA_IF_050 | The ODA system shall support the following output formats:<br>GeoTIFF<br>according to the OGC WCS 2.0 file format extensions | Mapping Server |
| SR_ODA_IF_060 | The ODA system shall support the following output formats:<br>GML/Multipart MIME | Mapping Server |

| | | |
|---|---|---|
| SR_ODA_IF_070 | The ODA system shall allow to access Archive Servers using the following protocols:<br>FTP<br>HTTP<br>WCS | Server for WCS EO AP |
| SR_ODA_IF_080 (SR_ODA_GEN_050) | see SR_ODA_GEN_050 in section 5.2.1<br>The ODA system shall support the requests defined in WCS EO AP . | Server for WCS EO AP |
| SR_ODA_IF_090 (ODA_DEM_030) | The ODA system shall support the requests defined in WCS 2.0 | Mapping Server |
| Human-Machine-Interface (HMI) | | |
| Identifier | Description | Software Component |
| SR_ODA_IF-CLI_030 | The WCS client shall support WCS 1.0.0 with the HTTP Get Binding. | WebMapViewer |
| SR_ODA_IF-CLI_040 | The WCS client should support WCS 2.0 Core with the HTTP Get Binding. | WebMapViewer |
| SR_ODA_IF-CLI_050 | The WCS client should support WCS 2.0 Core with the SOAP 1.2 Binding. | WebMapViewer |
| SR_ODA_IF-CLI_060 | The WCS client shall support the following image formats:<br>GeoTIFF (as per the limitations of SUN JAI libraries in handling TIFFs)<br>Other coverage formats as supported by the Open Source GDAL library<br>GDAL forms an optional component of the WCS client installation. | WebMapViewer |
| SR_ODA_IF-CLI_070 | The WCS client should support GML Rectified Grids as format in combination with the WCS 2.0 Core. | WebMapViewer |
| SR_ODA_IF-CLI_080 | The WCS client should support the WCS EO Application Profile<br>Note: Extent of support to be confirmed after definition of extension if all aspects of this EO Extension can be covered. | WebMapViewer |
| Operational requirements | | |
| Identifier | Description | Software Component |
| SR_ODA_SOR_070 | The ODA system shall log received request and provided responses. | Web Server, Server for WCS EO AP |

| | | |
|---|---|---|
| SR_ODA_SOR_080 | The ODA system shall provide the operator with an interface allowing checking of logs of received requests | Server for WCS EO AP (ODA Admin Client) |
| SR_ODA_SOR_090 | The configuration data needed for the ODA system shall be kept within text files. | Server for WCS EO AP |
| SR_ODA_SOR_100 | The ODA system shall provide the operator with an interface allowing the configuration and update of the ODA system | Server for WCS EO AP (ODA Admin Client) |
| SR_ODA_SOR_110 | The ODA system shall provide the operator with an interface allowing the upload and ingestion of datasets | Server for WCS EO AP (ODA Admin Client) |
| SR_ODA_SOR_150 | The ODA system shall provide the operator with an interface allowing DB Management | Server for WCS EO AP (ODA Admin Client) |
| Resources requirements | | |
| Identifier | Description | Software Component |
| SR_ODA_SOR_010 | The ODA system should be made available utilizing an instantiation of Debian based Linux (possibly Linux-HA - TBC) . | All |
| SR_ODA_SOR_020 | The PostgreSQL DB with the PostGIS extension shall be available . | geo-RDBMS |
| SR_ODA_SOR_030 | The GDAL library shall be available. | Mapping Server, Server for WCS EO AP |
| SR_ODA_SOR_040 | Python programming language shall be available. | Server for WCS EO AP |
| SR_ODA_SOR_050 | GCC compiler and Linux header files shall be available. | Mapping Server |
| SR_ODA_SOR_060 | A series of EPSG codes to be supported, and which GDAL must be able to handle, shall be agreed upon. | Mapping Server, Server for WCS EO AP |
| SR_ODA_SOR_120 | The ODA system shall run on a low end server machine, x86 based PC | All |
| SR_ODA_SOR_130 | The ODA system shall run on a PC with at least 4GB RAM (preferentially 8GB) | All |

| | | |
|---|---|---|
| SR_ODA_SOR_140 (ODA_CRR_010) | The ODA system shall run on a PC with a RAID-10 Disk Array with a "hot spare" (size is dependent on data holding for the demonstration - TBD) | All |
| SR_ODA_SSOR_010 | The platform where the ODA System is installed shall be connected to the Internet. | All |
| SR_ODA_SSOR_020 | The platform where the ODA System is installed shall be secured with a firewall (TBC). | All |
| SR_ODA_SSOR_040 | The platform where the ODA System is installed should operate an intrusion-detection software (TBC). | All |
| SR_ODA_SSOR_050 | The platform where the ODA System is installed should operate a system-wide logging system (TBD). | All |
| SR_ODA_SSOR_060 | The platform where the ODA System is installed should operate a logging analyzes tool (TBD). | All |
| Design requirements and implementation constraints | | |
| Identifier | Description | Software Component |
| SR_ODA_DRC_010 | OGC standards shall be applied where applicable. | All |
| SR_ODA_DRC_050 | A WebMapViewer client software shall be used as WCS client for the demonstration setup.<br>Note: The use of this client application is a prerequisite of the project as stated in | WebMapViewer |
| SR_ODA_DRC_060 | The protocols used for ODA should not require additional ports to be opened. | Server for WCS EO AP |
| SR_ODA_DRC_070 | The enhancements needed for the implementation of the Reference Implementation shall be based on OpenSource software tools. | Mapping Server |
| SR_ODA_DRC_080 (ODA_GEN_060) | The Reference Implementation shall be based on OpenSource software tools | Mapping Server |
| Security and privacy requirements | | |
| Identifier | Description | Software Component |

| | | |
|---|---|---|
| SR_ODA_SER_040 | The data accessible online shall be protected for illegal downloading.<br>Note: The End User shall have accesses to the data for discovery and visualization purpose. When he wants to process or download the data, the platform shall identify the user and check his rights as defined on SLA. | Web Server |
| SR_ODA_SER_050 | The security constraint shall not be a barrier to the commercial activity.<br>Note: The End User accesses the data online and the security is applied only when necessary with a non significant impact of ease access to the data for authorized users. | Web Server |
| SR_ODA_SER_060 | The security level shall be in line with the threat.<br>Note: The End User accesses free online data (free sub sampling data of commercial product), no security applied for normal use. Nevertheless, in case of massive request that demonstrate no real user connected, the platform should apply a minimum security process as authoritarian disconnection and blacklisted user action. | Web Server |
| Software quality requirements | | |
| Identifier | Description | Software Component |
| SR_ODA_QR_010 | The ODA system shall be developed to operate under a Linux Operating System | All |
| Software maintainability requirements | | |
| Identifier | Description | Software Component |
| SR_ODA_SMR_010 | The developers and maintainers require a user account on the installation platform. | All |
| SR_ODA_SMR_020 | The system shall provide access via a SSH connection for the developers and maintainers. | All |
| SR_ODA_SMR_030 | Root access or superuser (sudo) access rights will be required for the developers and maintainers (at least during initial | All |

| | | |
|---|---|---|
| | setup and operations). | |
| Data definition and database requirements | | |
| Identifier | Description | Software Component |
| SR_ODA_DDD_010 (ODA_SOR_020) | PostgreSQL with the PostGIS extension shall be used for the Reference Implementation | geo-RDBMS |
| Test Platform Validation Requirements | | |
| Identifier | Description | Software Component |
| SR_ODA_VA_120 | The ATPl shall include tests against the following product collections (TBC):<br>SPOT GeoTIFF (DIMAP)<br>Envisat ASAR Level 1 and above products<br>Note: In general all product formats supported by GDAL are potentially possible (but some might need extra integration effort) | Server for WCS EO AP |
| SR_ODA_VA_130 | A test platform as defined by the requirements SR_ODA_SOR_120 and SR_ODA_SOR_130 in section 5.6 should be used | Server for WCS EO AP |

Table 8: Traceability of Requirements to Design Components

**Heterogeneous Mission Accessibility - Follow-On**
**Design Definition File - Software Design Document**

---- End of Document ----