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

Design Justification File (DJF) - Software Reuse File (SRF)

EOX Document Reference:	HMA-FO_ODA-DJF-SRF_EOX
Issue:	1.0
Date:	2010-11-26
Authors:	C.Schiller, S.Krause, S.Meissl
Responsible for Document:	C.Schiller

Prime Contractor:

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

Subcontractors:

Approved by	Organisation	Date	Signature
Gerhard Triebnig	EOX	2010-11-26	

Release Track

Release No.	**Date**	**Remarks**
1.0	2010-11-26	Document released

Distribution List:

Authors:	All
EOX:	G. Triebnig
ESA:	Pier-Giorgio Marchetti
GIM:	S. Desender
SPOT:	D. Giacobbo, P.Merigot
Spacebel:	Y. Coene

Table of Content

1 Introduction..4
2 Applicable and reference documents 5
3 Terms, definitions and abbreviated terms 7
4 Background analysis of Open Source software packages to be used to implement the WCS enhancements...9
5 Decision...13
6 Software development approach...14

List of Tables

Table 1: Trade-off Criteria for existing OWS OS Software Implementations..................11

List of Figures

Figure 1: Performance comparison between MapServer and GeoServer at the FOSS4G 2009 conference in Sysdney.
(Source: http://www.slideshare.net/gatewaygeomatics.com/wms-performance-shootout)..13
Figure 2: MapServer Architecture from [RD7] 15

1 Introduction

This document defines the Software Reuse File (SRF) for the HMA-Follow-on Task-3 (HMA-FO) Online Data Access (ODA) system Reference Implementation.

The HMA-FO ODA system Reference Implementation shall demonstrate the usage and the advantages of a Web Coverage Service (WCS) to access EO datasets. More specifically, the concepts introduced with the new OGC WCS 2.0 standard and the functionalities of the new WCS EO Application Profile (WCS EO AP) shall be demonstrated for the suitability in various online data access scenarios.

The SRF documents the information used during the background analyzes to evaluate the possibilities to reuse existing software products. Further the decision to use a specific software product is documented.

2 Applicable and reference documents

2.1 Applicable documents

[AD1] ECSS Standard: Space Engineering – Software, Ref. ECSS-E-ST40C, 6 March 2009

[AD2] ECSS Standard: Space product assurance – Software product assurance, Ref. ECSS-Q-ST-80C, 6 March 2009

[AD3] HMA-FO_ODA Requirements Baseline Document - Software System Specification (HMA-FO_ODA-RB-SSS_EOX, ver. 1.4, 2010-11-03)

[AD4] HMA-FO_ODA Requirements Baseline Document – Technical Note (HMA-FO_ODA-RB-TN_EOX, ver.1.1, 2010-05-27)

[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 07-140, OGC WCS 2.0 Application Profile - Earth Observation, v.0.1.0, 2010-10-27

[AD10] OGC 10-147, OGC Web Coverage Service (WCS) 2.0 Interface Standard GeoTIFF Encoding Format Extension, ver. 0.0.1, 2010-07-06

[AD11] 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

[AD12] OGC 06-080, GML Application Schema for EO Products, ver. 0.9.3, 2008-07-21

[AD13] OGC 07-011 The OpenGIS Abstract Specification Topic 6, Schema for coverage geometry and functions.

[AD14] HMA-FO_ODA, Technical Specification - Software Requirements Specification, (HMA-FO_ODA-TS-SRS_EOX), ver. 1.2, 2010-11-26

[AD15] HMA-FO_ODA, Design Definition File - Software Design Document, (HMA-FO_ODA-DDF-SDD_EOX) ver. 1.0, 2010-11-26

2.2 References

[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

3 Terms, definitions and abbreviated terms

AP	Application Profile
CDS	Coordinated Data access System (GSCDA)
DJF	Design Justification File
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)
GSCDA	GMES Space Component - Data Access
HMA	Heterogeneous Mission Accessibility
HMA-FO	HMA – Follow On
ICD	Interface Control Document
KVP	Key-Value Pair
HMI	Human Machine Interface
ODA	Online Data Access
ODA system	HMA-FO Reference Implementation for Online Data Access utilizing WCS 2.0 and the WCS EO AP
OGC	Open Geospatial Consortium
OGR	OGR Simple Features Library
OSGeo	Open Source Geospatial Foundation
OSS	Open Source Software
RB	Requirements Baseline
SOAP	Simple Object Access Protocol,
SSE	Service Support Environment
TN	Technical Note

TS	Technical Specification
TS-SRS	Technical Specification- Software Requirements Specification
WCPS	Web Coverage Processing Service (OGC)
WCS	Web Coverage Service (OGC)
WCS EO AP	Web Coverage Service Earth Observation Application Profile
WCS-T	Web Processing Service – Transactional (OGC)
WCTS	Web Coordinate Transformation Service (OGC)
WMS	Web Mapping Service(OGC)
WPS	Web Processing Service (OGC)
XML	Extended Markup Language

4 Background analysis of Open Source software packages to be used to implement the WCS enhancements

At Kickoff of the HMA-FO project it has been decided that the Online Data Access (Task 3) shall use the newly to be developed WCS 2.0 specification as the basis for the development of the WCS EO Application Profile (WCS EO AP) as well as for the development of a compliant test implementation using an Open Source Software Tool.

Therefore a analyzes of Open Source Software (OSS) packages implementing earlier WCS Interface Specification is performed. We are providing an evaluation of the pros and cons of each platform based on implementation requirements concerning the handling and storage of coverages and mosaics.

We are comparing these packages with respect to various criteria, most importantly:

- their support for WCS 1.1
- their flexibility to handle dynamically changing datasets
- their support for mosaic and image pyramid handling
- performance criteria

Whereas the first and last criterion is obvious, the other two need explanation: The ODA software implementation must ensure that the WCS server configuration for available coverages and their metadata can be changed dynamically to handle incoming WCS-T transaction requests.

As EO product collections often contain many terabytes of EO product data, efficient manipulation and storage of large raster data collections and/or mosaics is crucial for productive use of a WCS(-T) implementation. For flexible and high performance access to raster data, large coverages usually are stored as tile sets or image pyramids.

A tile set is obtained by splitting the original coverage at equally spaced perpendicular grid lines into smaller image files. An image pyramid consists of tile sets built from the same raster data resampled to ever diminishing image resolutions. Pyramids are especially useful to deliver imagery at a variety of different zoom levels with high performance. Tiles and pyramids can exist inside files (e.g. GeoTIFF, JPEG200) or at filesystem level (i.e. tiles for each pyramid level are stored separately in the file system).

The technical pyramid metadata (often called tile index), i.e. the resolutions and locations of tiles within the file system, must be stored in a way enabling fast geospatial queries for high performance coverage access. In practice, there are two solutions for the storage of tile indexes: ESRI shapefiles and spatially enabled databases.

Shapefiles are the more commonly supported storage method, whereas spatial databases allow for more flexibility and more high performance geospatial indexing. The most comprehensive and elaborate open source spatial RDBMS implementation is PostgreSQL with its PostGIS extension, which serves as reference for the following comparison.

In Table 1.1.2 the basic properties of the following four software packages are listed:

- MapServer [RD1]
- GeoServer [RD2]
- deegree [RD3]
- THREDDS Data Server [RD4]

Compared Aspect	**MapServer**	**GeoServer**	**deegree WCS**	**THREDDS Data Server**
Maintaining Entity	Community project with backing of the active Open Source Geospatial Foundation (OSGeo)	Community project led by the company OpenGeo, USA	Community project led by lat/lon GmbH and University of Bonn, Germany	Unidata/UCAR
License	MIT-style	GPL Version 2	LGPL	LGPL
Implementation language	C	J2EE	Java	Java
Language bindings	Various through SWIG e.g. Python, Perl, Java, .NET, etc. and PHP	-	-	-
Database bindings	PostgreSQL/PostGIS, MySQL, Oracle	PostgreSQL/PostGIS, MySQL, DB2, Oracle, ArcSDE, SQL Server	Oracle	None
Raster formats	Various through GDAL e.g. GeoTiff, JPEG2000, HDF, netCDF, etc.	Native Java support for GeoTIFF, GTOPO30, ArcGrid, WorldImages, ImageMosiacs, and Image Pyramids Support for MrSID, ECW, JPEG2000, DTED, Erdas Imagine, and NITF through GDAL ImageIO Extension	GeoTIFF, ECW, GIF, TIFF, BMP, JPEG, PNG	GeoTIFF (only for output), NetCDF
Vector formats	PostGIS and various through OGR e.g. Shapefile, GML, SQLite, MySQL, Oracle Spatial, etc.	PostGIS, Shapefile, ArcSDE, DB2 and Oracle	-	-
OWS compliance	WMS (1.0.0, 1.0.7,	WMS 1.1.1, WFS (1.0	WCS 1.0 (Reference	WCS 1.0

Compared Aspect	MapServer	GeoServer	deegree WCS	THREDDS Data Server
	1.1.0, 1.1.1), WFS (1.0.0, 1.1.0), WCS (1.0, 1.1), GML (2.1.2, 3.1.0 Level 0 Profile), WMC (1.0.0, 1.1.0), Filter Encoding (1.0.0), SLD (1.0.0), SOS (1.0.0), OM (1.0.0), SWE (1.0.1), OWS (1.0.0, 1.1.0)	and 1.1, transactions and locking) and WCS (1.0 and 1.1) specifications, as tested by the CITE conformance tests. GeoServer additionally serves as Reference Implementation for WCS 1.1 and WFS 1.0 and 1.1 GML (2.1.2, 3.1.1), SLD	implementation)	

Table 1: Trade-off Criteria for existing OWS OS Software Implementations

The **MapServer** development is now in the hands of the Open Source Geospatial Foundation (OSGeo) [RD5] and [RD6] OSGeo is an open development community that applies a responsible project governance model. The OSGeo seal provides added confidence in the viability and safety of the project. MapServer implements the WCS 1.1 interface.

MapServer excels at supporting a broad range of raster data sources by means of GDAL, an open source library for raster data manipulations written in C. It allows for a wide range of coordinate transformations by using the open source PROJ.4 library. Furthermore, it provides scripting language bindings (Python, PHP, and others) that allow to easily extend the functionality of the basic software.

MapServer supports various database bindings, most importantly to the geospatially enabled open source RDBMS PostgreSQL with its PostGIS extension. MapServer can be extended to use rasdaman, the raster image database developed by JUB.

MapServer's biggest drawback is that dynamic configuration is a relatively tricky thing to do. It relies on static Mapfiles for the configuration of an OGC Web Service which can be manipulated by means of the scripting language bindings. Database storage of dynamic configuration items, most importantly the metadata and file system paths of coverages uploaded with WCS-T, is not directly supported. It can be implemented using scripts or by extending the MapServer C code, however. MapServer's advantage is the support of fast-cgi, which is a protocol for interfacing interactive programs with a web server. This support provides a highly efficient implementation allowing MapServer to serve data, at least via WMS, much faster then GeoServer.

GeoServer is a software package with capacities similar to MapServer implemented in Java. It is supported by an open source community as well. GeoServer is built on **Geotools**, an open source Java GIS toolkit. GeoServer is the reference implementation for WCS 1.1, but compliance with this standard has not yet been certified by the OGC.

GeoServer supports an equally wide range of raster formats and database backends as MapServer does. Among its benefits is an easy-to-use administration interface, which greatly simplifies configuration of data sources and service properties. Furthermore, it comes with an optional plug-in for image pyramid creation and manipulation which could be used for storing mosaics.

On the downside, GeoServer does only allow to store coverage and pyramid metadata (like the location of the tiles on the file system, and references to the original coverages a mosaic is composed of) in shapefiles on the file system. It needs to be extended to include metadata information stored in databases, which would be the method of choice for large collections of coverages or mosaics for reasons of performance, scalability and flexibility.

The **deegree** framework – written in Java, too – was the OGC Reference Implementation for WCS 1.0. However, the current version of the software does not yet support WCS 1.1, and thus we have to dismiss it from further consideration.

The **THREDDS Data Server** (TDS) is mentioned here for completeness, although it is not suitable for the purposes of the ODA project, as it provides only a very restricted implementation of WCS 1.0.

5 Decision

All these considerations leave us with two viable open source software solutions for the implementation of online data access with respect to the criteria mentioned above: MapServer in combination with PostgreSQL/PostGIS for mosaic and pyramid storage and GeoServer.

- given that coverage and mosaic handling using a combination of MapServer and PostgreSQL/PostGIS allows for great performance and flexibility;
- given MapServers performance advantage due to fast-cgi;
- given that GeoServer does not support this without modifications to the code of the distribution;
- given that EOX has ample experience with MapServer open source software set-up and has successfully used it to implement raster data stores accessible through web interfaces;

We conclude that an approach combining MapServer and PostgreSQL/PostGIS is our first choice for implementation, GeoServer the second. Additionally, we propose to use rasdaman, an Open Source raster server usable as extension or side component to MapServer like GDAL, for the demonstration of advanced concepts.

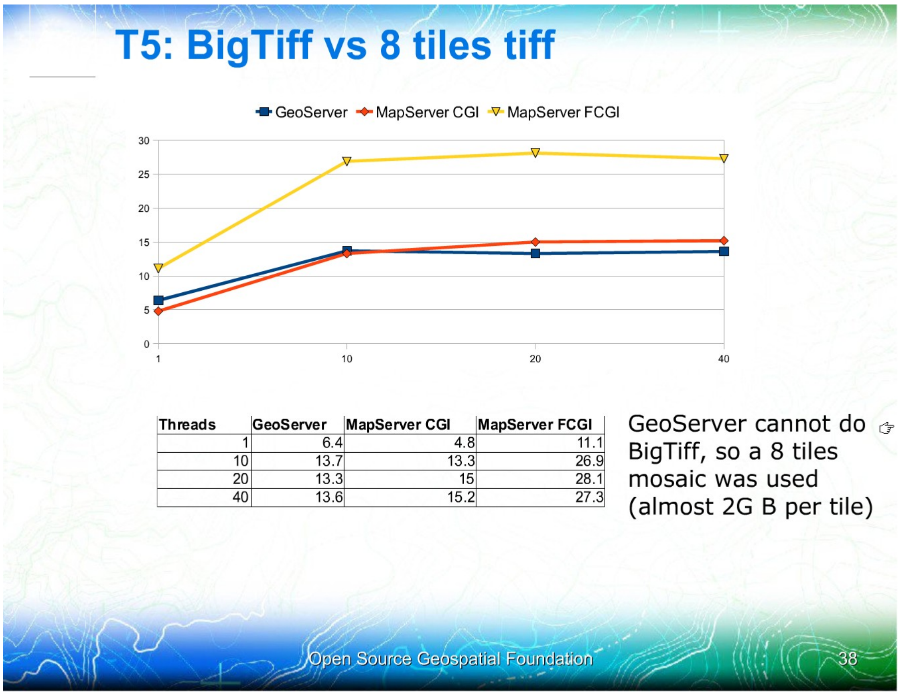

Figure 1: Performance comparison between MapServer and GeoServer at the FOSS4G 2009 conference in Sysdney.

(Source: http://www.slideshare.net/gatewaygeomatics.com/wms-performance-shootout).

6 Software development approach

We propose to implement changes by enhancing the source-code to support WCS2.0 and WCS EO AP and trying to feed patches back to the MapServer community.

Figure 2 shows the high level architecture of MapServer. The part with brown background at the top of the figure shows the data input whereas the bottom parts in blue show the possible output.

The implementation task will focus on the compliance to WCS2.0 and the WCS Earth Observation Application Profile. However, as the Application Profile is still under development the extend can currently not be defined.

The profile will especially reflect on the new WCS-T specification which will be used to address the Use Case scenarios US4_1 to US4_3. as defined in the Requirement Baseline Document (HMA-FO_ODA-RB-SSS_EOX). From Figure 2 it can be seen that the data processed with MapServer currently flows only from the top to the bottom. The WCS-T specification foresees the data flow in the opposite direction. Currently no available Open Source implementation of the WCS-T specification is known to the consortium.

Currently there is no implementation of the WCS-T specification foreseen in the MapServer community. Due to the change of the target (WCS 1.1.1 → WCS 2.0) extending the MapServer with the WCS-T interface, as stated in the proposal, will not anymore be the first implementation task.

The Transaction operation accepts a description of input coverages as parameter (where applicable), that contains references to the actual metadata and image files to be ingested. The operation can be synchronous or asynchronous; it returns an XML response confirming that the transaction has been processed by the server, or an exception report message in the case of failure.

The raster image data and metadata may be either attached to the request (e.g. via HTTP POST) or merely referenced to by a URL. In the latter case, the resources have to be made available by the user or the service that issues the request, using some file transfer mechanism (e.g. HTTP, FTP). Currently, the reference is expected to point to the data and metadata files themselves, but not to another WCS server.

Note that the terms client and server in that context do only refer to the roles of the participants with respect to the WCS-T protocol. If the data is not attached to the request, it has to be retrieved by the receiver of the WCS-T request by some other file transfer mechanism such as HTTP or FTP. In that case the originator of the request (the WCS-T client) usually will act as a server – the roles are reversed for actual data transfer.

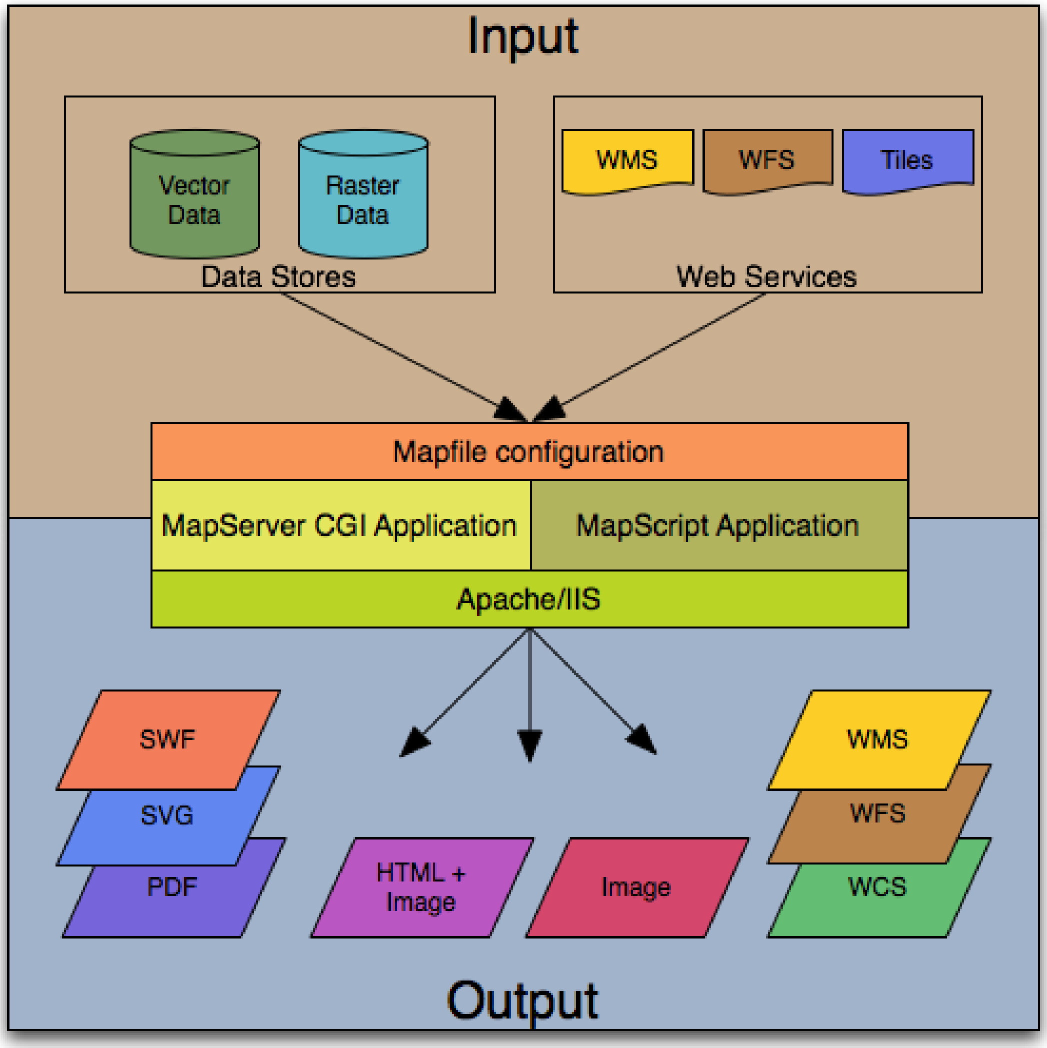

Figure 2: MapServer Architecture from [RD7]

The implementation will support insertion, update, partial update, and deletion of coverages on a server accessible via WCS to the extent the WCS specification allows it. Among the data formats supported will be at least GeoTIFF and if possible JPEG2000. For these formats the implementation will be tested and demonstrated. However, various other file formats like HDF, NetCDF, JPEG, PNG, etc. could also be made easily available through the Geospatial Data Abstraction Library (GDAL).

The WCS interface is available at the output side, but not on the input. This means that MapServer can not act as WCS client. This functionality, however, which will be available with rather small effort when WCS-T is implemented. The consortium will propose this extension to the MapServer community after finishing the WCS-T implementations.

Besides the possibility to extend the C source-code of the OSS MapServer there exists the MapScript scripting interface. MapScript allows to directly create and access MapServer objects instead of interacting with MapServer through its Common Gateway Interface (CGI) and Mapfiles. The MapScript interface is generated by the Simplified Wrapper Interface Generator (SWIG) [RD8] library which provides language bindings to scripting languages like Python [RD9].

The current WCS specification also foresees SOAP over HTTP as optional encoding. If it is decided to use these SOAP bindings within the WCS EO Application Profile we propose to use the afore mentioned scripting interface for the implementation. This ensures the most efficient and flexible way to implement the support of the additional communication protocol layer SOAP.

MapServer currently supports WCS versions 1.0 and 1.1. The new developments for the MapServer based on WCS 2.0 will need, due to the increased development volume, to get a much closer involvement of the MapServer developer community to open a new development branch. Also the EO Application Profile will be based on WCS 2.0.

Special attention is drawn at the presentation of accompanying metadata. This includes client side implementations as well as the WCS server side which has to support the delivery of requested metadata be it in the GetCapabilities or DescribeCoverage response

Additional functionalities or at least configuration needs arising from the Application Profile which are not available in the current MapServer implementation. It has to be decided on a case by case basis in conjunction with the ESA technical officer if this is better implemented in the source-code or by using the MapScript scripting interface to MapServer.

MapServer is an official project of the Open Source Geospatial Foundation (OSGeo) [RD5] since it has graduated from incubation on 16th December 2008 [RD6] Thus the major software configuration management tools are provided by OSGeo. These tools include a Subversion (SVN) repository [RD10], [RD11], an Issue and Bug Tracker and Wiki utilizing the software Trac [RD12], [RD13], and some mailing lists provided by the software Mailman [RD14], [RD15] For online discussions, which would be very ineffective via email, the MapServer community uses the Internet Relay Chat (IRC) [RD16]. All these software tools are licensed as Open Source software and are also used by EOX routinely.

EOX will actively participate in the community process to optimize the possible impact of the developed extensions. The developments and extensions will be fed into the main development branch of MapServer, if agreed by the community, hosted at OSGeo. Otherwise EOX will set-up an freely available SVN at their own premises with a development branch starting from the latest stable release of MapServer to develop and integrate the new extensions and also provide it as OSS. EOX will provide on a best effort basis upstream releases of the extensions as the MapServer community releases new stable versions. Patches for MapServer including the stable releases of the extensions will be made available either on SourceForge [RD17] or GoogleCode [RD18]. The choice is almost only a matter of taste since for software development the installation at EOX will be used and only stable patches will be fed to the chosen repository. However, the completely preferred solution is a close cooperation with the community which includes the usage of OSGeo repository and other tools.

Since GeoServer also is a community project comparable configurations can be found and equivalent mechanisms used if, in consultation with the responsible ESA technical officer, decision will be made to not use MapServer.

Based on the facts presented in the above comparison the decision has been in favour of MapServer as the OSS tool for the developments of the HMA-FO Reference Implementation.

Heterogeneous Mission Accessibility Follow-On -
Design Justification File - Software Reuse File

---- End of Document ----