



D21.3 Registry Installation and User Manual

Work package	WP21	Services/Toolkits Development and Adaptation
Task	T21.2 RepInfo Registry Service Development	
Author (s)	Christian Ceprani	ACS
Author (s)	Manuela Sagona	ACS
Author (s)	Andrea Colapicchioni	ACS
Authorized by		
Reviewer		
Doc Id		
Dissemination Level	CONFIDENTIAL/PUBLIC	
Issue	1.4	
Date	28/02/2014	

Abstract:

This document represents the Installation, Deployment and User Manual for the Representation Information Registry Service developed in the frame of SCIDIP-ES project. This document contains relevant information on how to install, configure and use the Registry Service.

Document Log

Date	Author	Changes	Version	Status
11/02/2014	Manuela Sagona	Base document created from D21.4. Updated for M30 Release.	1.3	Draft
28/02/2014	Manuela Sagona, Andrea Colapicchioni, Shirley Crompton	Minor editorial changes. Replaced Manifest and RepInfoLabel examples with those using new xsd schemas.	1.4	Released

TABLE OF CONTENTS

1	INTRODUCTION.....	6
1.1	PURPOSE AND SCOPE.....	6
1.2	WHO SHOULD READ THIS DOCUMENT.....	6
1.3	SYSTEM CONTEXT.....	6
1.4	RELEASE NOTES.....	7
2	INSTALLATION GUIDE.....	7
2.1	OVERVIEW.....	7
2.2	PREREQUISITES.....	7
2.2.1	SOFTWARE PREREQUISITES.....	7
2.2.2	HARDWARE PREREQUISITES.....	7
2.3	DOWNLOAD INFORMATION.....	7
2.4	LICENSE INFORMATION AND TERMS OF USE.....	8
2.5	REGISTRY OVERALL ARCHITECTURE.....	8
2.5.1	THE SERVICE GATE.....	8
2.5.2	THE REGISTRY ENGINE.....	8
2.5.3	APACHE SOLR.....	9
2.5.4	OPENSEARCH REQUESTHANDLER.....	9
2.5.5	INSERT.....	10
2.5.6	SEARCH.....	10
2.6	THE SCIDIP-ES REPOSITORY.....	10
2.7	OSS/COTS INSTALLATION.....	11
2.8	REGISTRY AND REPOSITORY INSTALLATION.....	11
2.9	UNINSTALLATION.....	17
3	USING SCIDIP-ES REPINFO REGISTRY AND REPOSITORY SERVICES.....	18
3.1	SCIDIP REGISTRY.....	18
3.1.1	GETTING STARTED.....	18
3.1.2	GETTING A REPINFOLABEL/MANIFEST BY CPID FROM THE REGISTRY SERVICE.....	18
3.1.3	FILTERING RESULTS FROM THE REGISTRY SERVICE.....	20
3.1.4	INSERTING A NEW REPINFOLABEL/MANIFEST INTO THE REGISTRY.....	20
3.1.5	DELETING A REPINFOLABEL/MANIFEST FROM THE REGISTRY.....	22
3.1.6	EXPORTING/IMPORTING REGISTRY CONTENT.....	22
3.2	SCIDIP REPOSITORY SERVICE.....	23
3.2.1	GETTING STARTED.....	23
3.2.2	INSERTING A NEW DIGITALOBJECT INTO THE REPOSITORY SERVICE.....	23
3.3	GETTING A DIGITALOBJECT BY CPID FROM THE REPOSITORY SERVICE.....	24
3.4	DELETING A DIGITAL OBJECT BY CPID FROM THE REPOSITORY SERVICE.....	24
4	AUTHENTICATION AND AUTHORIZATION IN SCIDIP-ES.....	26
4.1	CREATE USERS.....	26
4.2	CREATE GROUPS.....	26
4.3	CONFIGURE APACHE.....	27
4.4	UPDATE USERS AND GROUPS.....	29

4.4.1	USERS MANAGEMENT	30
4.4.2	GROUPS	30
4.5	CURL COMMANDS UPDATE	30
5	SCIDIP REGISTRY AND REPOSITORY BACKUP/RESTORE	30
5.1	REGISTRY	30
5.2	REPOSITORY	31
6	REFERENCE MANUAL	31
6.1	KEYBOARD SHORTCUTS	31
6.2	COMMAND-LINE COMMANDS	32
6.3	PUBLIC APIS	32
6.3.1	API LOGIC	32
6.3.2	IF-SCIDIPE-REGISTRYSEARCH	32
6.4	AUTHENTICATION/AUTHORIZATION - GETTING STARTED	34
6.4.1	CREATE USERS	35
6.4.2	CREATE GROUPS	35
6.4.3	CONFIGURE APACHE	35
6.5	REGISTRY - GETTING STARTED	37
6.5.1	GETTING A REPINFOLABEL/MANIFEST BY CPID FROM THE REGISTRY SERVICE	38
6.5.2	INSERTING A NEW REPINFOLABEL/MANIFEST INTO THE REGISTRY	38
6.5.3	DELETING A REPINFOLABEL/MANIFEST FROM THE REGISTRY	39
6.5.4	EXPORTING/IMPORTING REGISTRY CONTENT	39
6.6	SCIDIP REPOSITORY - GETTING STARTED	40
6.6.1	INSERTING A NEW DIGITALOBJECT INTO THE STORAGE SERVICE	40
6.6.2	GETTING A DIGITALOBJECT BY CPID FROM THE STORAGE SERVICE	41
6.6.3	DELETING A DIGITALOBJECT BY CPID FROM THE STORAGE SERVICE	41
7	TROUBLESHOOTING COMMON ISSUES	42
8	ANNEX – ATOM SCHEMA DESCRIPTION	43
9	ANNEX – OPENSEARCH SCHEMA DESCRIPTION	48
10	ANNEX – RIL SCHEMA DESCRIPTION	58
11	ANNEX – MANIFEST SCHEMA DESCRIPTION	61
12	ANNEX – RIL AND MANIFEST EXAMPLES	63
12.1	EXAMPLE MANIFEST FOR MERIS N1 SCIENTIFIC DATA	63
12.2	EXAMPLE RIL FOR MERIS N1 SCIENTIFIC DATA	63
13	FIGURES AND TABLES	64
13.1	LIST OF FIGURES	64
13.2	LIST OF TABLES	64

1 Introduction

1.1 Purpose and Scope

This document provides an overview of the Registry/Repository Services focusing in particular on its installation and usage.

1.2 Who should read this document

Users who wish to understand, install and use the Registry/Repository Service.

1.3 System Context

The Representation Information (ReplInfo) Registry service, the core component of the SCIDIP-ES e-infrastructure, allows the capture of networks of representation information needed to preserve digital objects.

The Registry Service is a repository for ReplInfo objects which are defined in OAIS¹ as the additional information that maps a data object into more meaningful concepts and allows it to be understood and used correctly.

ReplInfo objects stored in the Registry Repository Service are described by Manifest objects which are searchable to facilitate discovery and access of ReInfo objects.

The ReplInfo provided by the Registry Service can cover the following topics:

1. The **structure** of the digital object
2. The **meaning** of that digital object
3. The processing and rendering information
4. The location of the digital object

The information are intended to help a user to access the digital object, to understand and use it.

Each piece of ReplInfo is kept in the Registry as an Archive Information Package (AIP). An AIP is encoded as a “Manifest” which points to the ReplInfo (normally as a ReplInfoLabel), to the data object itself, and also points to or contains the Preservation Description Information (PDI) for the data object. PDI is also provided for a ReplInfoLabel, since this is archived as a data object.

All data object within SCIDIP-ES is accompanied by a Curation Persistent IDentifier (CPID) pointing to the ReplInfo which provides additional information to facilitate its interpretation and use. Given a CPID associated with a data object, one can request the ReplInfoLabel from the Registry as fully described in D21.3 Master Document.

¹ <http://public.ccsds.org/publications/archive/650x0m2.pdf>

1.4 Release Notes

The Registry/Repository Service release allows the user to store and retrieve Representation Information by means of their Manifest and RepInfoLabel. This is achieved via a RESTful interface based upon OpenSearch protocol as described in Sections 3.1.2 to 3.1.2.

2 Installation Guide

2.1 Overview

There are three main steps for the Registry and Repository services installation and setup:

1. PEAR packages installation
2. Registry and Repository packages installation and setup
3. MogileFS installation and setup

2.2 Prerequisites

2.2.1 Software prerequisites

Software	CentOS release 6.4 Apache Tomcat6 MySql (Ver 14.14)
----------	--

2.2.2 Hardware prerequisites

Specs	Ram: 4GB Ram swap: 8GB Disk: 50GB Vcpu: 2
-------	--

2.3 Download Information

The software can be downloaded from the SCIDIP-ES interactive platform, at the address

<http://int-platform.scidip-es.eu/>

2.4 License Information And Terms of Use

The SCIDIP-ES Registry and Repository services are licensed under the Apache License, Version 2.0 (the "License");

You may not use this software except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

2.5 Registry Overall Architecture

The SCIDIP-ES Registry comprises the Service Gate (Section 2.5.1) and Registry Engine (Section 2.5.2). The former is a REST² wrapper built on top of the Registry Engine which is built on Apache SOLR³

2.5.1 The Service Gate

The Service gate is responsible for receiving the OpenSearch Request and to route the request to SOLR as well as performing the preliminary validity check and taking care of the logging activities.

The Service Gate module provides the interface functionalities:

- Open Search support
- Additional services (Import, Export)
- Metadata validation and ingestion service

The Service gate is implemented in PHP. See the following chapters for a description of the Service Interfaces.

2.5.2 The Registry Engine

The Registry Engine provides the core functionality of the SCIDIP-ES Registry.

The **Registry Engine** component implements the actual index/search functionalities built on SOLR and a custom RequestHandler plugin which is responsible for handling the OpenSearch format input/output.

The **SOLR** component is provided by Apache SOLR (see Section 2.5.3).

² <http://en.wikipedia.org/wiki/REST>

³ <https://lucene.apache.org/solr/>

The **OpenSearchRequestHandler** component is a plugin of SOLR implementing the OpenSearch requests conversion to and from the SOLR syntax.

2.5.3 Apache SOLR

Apache SOLR (SOLR hereafter) is an open-source search platform built upon the Apache Lucene library, the *de-facto* standard for Java indexing and search technology. SOLR provides a number of valuable features on top of the low-level APIs provided by Lucene: full-text search, near-real time indexing, dynamic clustering, distributed indexing, scalability and geospatial search.

SOLR exposes the functionalities of its core (Indexing/Search) through a REST HTTP interface which simplifies integration with external components.

In addition, SOLR provides an extensible plugin architecture easily allowing *ad-hoc* customization (see Section 2.5.4 below).

The following diagram provides a high-level view of the internal architecture of SOLR

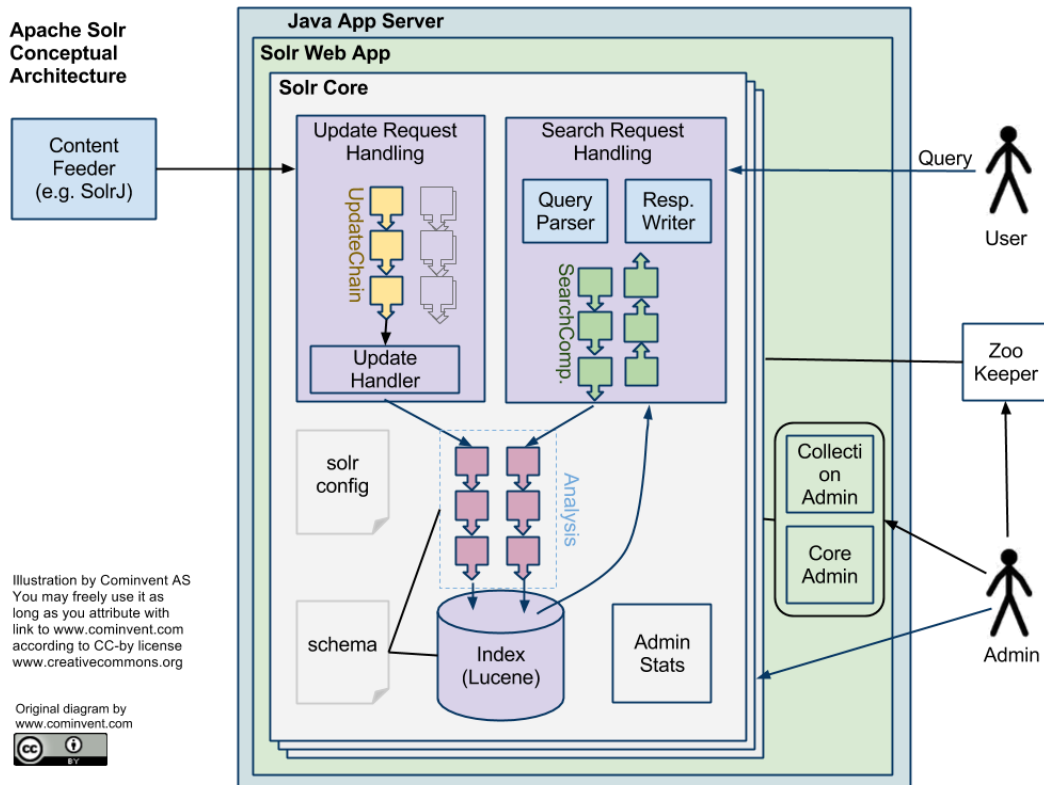


Figure 1 SOLR High Level Internal Architecture

2.5.4 OpenSearch RequestHandler

The request handler is responsible of the translation of the incoming request to and from the internal SOLR syntax: the “OpenSearch RequestHandler” translates the OpenSearch format into SOLR syntax.

The OpenSearch protocol is implemented by means of an XSLT transformation applied dynamically on the result of the SOLR request.

2.5.5 Insert

The process of adding a document into the index is called “indexing” and it involves a number of steps to extract terms from input data and to update the index: a high level overview of the whole process is shown in Figure 2.

Input data, i.e. “Documents”, are represented as an unordered set of “Fields” each one containing multiple “Terms”. A Field can be declared in multiple ways to determine whether it should be analysed, indexed or simply stored.

2.5.6 Search

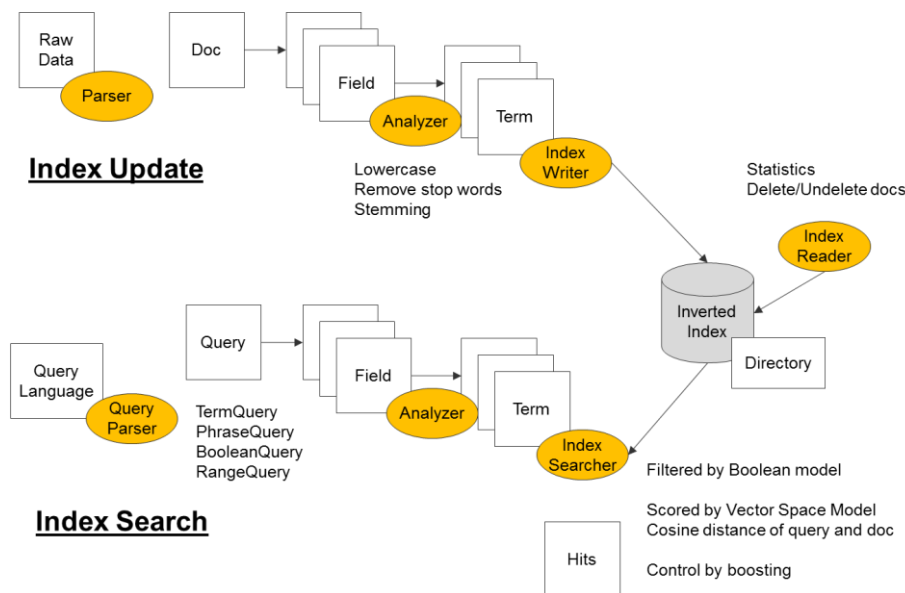


Figure 2 SOLR Index Update and Index Search - High Level Overview

2.6 The SCIDIP-ES Repository

The Repository is responsible for storing and retrieving the digital objects pointed to by the manifests. It is built on Mogilefs⁴, an open source distributed filesystem and a PHP library implementing the REST service interfaces.

Mogilefs features include:

- Application level -- no special kernel modules required.

⁴ <https://code.google.com/p/mogilefs/>

- No single point of failure -- all three components of a MogileFS setup (storage nodes, trackers, and the tracker's database(s)) can be run on multiple machines, so there's no single point of failure. (It is possible to run trackers on the same machines as storage nodes, too, so there is no need to maintain 4 machines...) A minimum of 2 machines is recommended.
- Automatic file replication -- files, based on their "class", are automatically replicated between enough unique storage nodes as to satisfy the minimum replica count as requested by their class. For instance, for a photo hosting site you can make original JPEGs have a minimum replica count of 3, but thumbnails and scaled versions only have a replica count of 1 or 2. If you lose the only copy of a thumbnail, the application can just rebuild it using the original JPEGs. In this way, MogileFS (without RAID) can save money on disks that would otherwise be storing multiple copies or representations of data unnecessarily.
- "Better than RAID" -- in a non-SAN RAID setup, the disks are redundant, but the host isn't. If you lose the entire machine, the files are inaccessible. MogileFS replicates the files between devices which are on different hosts, so files are always available.
- Flat Namespace -- Files are identified by named keys in a flat, global namespace. You can create as many namespaces as you'd like, so multiple applications with potentially conflicting keys can run on the same MogileFS installation.
- Shared-Nothing -- MogileFS doesn't depend on a pricey SAN with shared disks. Every machine maintains its own local disks.
- No RAID required -- Local disks on MogileFS storage nodes can be in a RAID, or not. It's cheaper not to, as RAID doesn't buy you any safety that MogileFS doesn't already provide.
- Local filesystem agnostic -- Local disks on MogileFS storage nodes can be formatted with your filesystem of choice (ext3, XFS, etc..). MogileFS does its own internal directory hashing so it doesn't hit filesystem limits such as "max files per directory" or "max directories per directory". You can use what you're comfortable with.

2.7 OSS/COTS Installation

Assuming a Linux platform, as root user, copy the file ScidipRegistryRepository.tgz into a temporary directory (referred as <installation_dir>) and run the following commands:

```
tar xfv ScidipRegistryRepository.tgz  
cd <installation_dir>/PEARpackages  
rpm -ivh *
```

2.8 Registry and Repository Installation

To install the Registry and Repository PHP modules run the following commands:

```
cd <installation_dir>/RegistryPackages

rpm -ivh *

cd /etc/httpd/conf.d/

mv scidip.conf scidip.conf.orig

mv scidip_storage.conf scidip_storage.conf.orig

touch scidip.conf
```

copy the following lines into scidip.conf file where <server_name> is the name of the local installation server:

```
<VirtualHost *:80>
  ServerName <server_name>
  ServerAdmin admin@<server_name>

  ErrorLog logs/registry_error.log
  CustomLog logs/registry_access.log common

  Alias /ScidipRegistry "/var/www/html/acs-apache-solr_scidip/web"

  RewriteEngine On
  RewriteCond %{REQUEST_URI} /ScidipRegistry/(.*)$
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteRule .* /acs-apache-solr_scidip/web/main.php [L,QSA]

  Alias /ScidipRepository "/var/www/html/Scidip_Storage/web"

  RewriteCond %{REQUEST_URI} /ScidipRepository/(.*)$
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteRule .* /Scidip_Storage/web/main.php [L,QSA]

  RewriteCond %{REQUEST_URI} /FindingAidGui/(.*)$
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d

</VirtualHost>
```

Deploy solr4.war under webapps directory in tomcat6 and restart tomcat6 and httpd services by running the following commands:

```
cp /opt/solr4/apache-solr-4.0.0.war /var/lib/tomcat6/webapps/solr4.war
service httpd restart
service tomcat6 restart
```

MogileFS Installation and setup

A MogileFS installation package kit is provided. The list of the provided rpms is as follows:

- lighttpd-1.4.28-2.el6.rf.x86_64.rpm
- mogilefsd-2.60-5.el6.noarch.rpm
- MogileFS-Utills-2.23-1.el6.noarch.rpm
- mogstored-2.60-5.el6.noarch.rpm
- mogstored-backend-apache-2.60-5.el6.noarch.rpm
- mogstored-backend-lighttpd-2.60-5.el6.noarch.rpm
- mogstored-backend-none-2.60-5.el6.noarch.rpm
- mogstored-backend-perlbal-2.60-5.el6.noarch.rpm
- Perlbal-1.75-1.el6.noarch.rpm
- perl-common-sense-3.0-1.el6.rf.x86_64.rpm
- perl-Danga-Socket-1.61-1.el6.rf.noarch.rpm
- perl-Gearman-1.09-1.el6.noarch.rpm
- perl-Gearman-Client-Async-0.94-3.el6.noarch.rpm
- perl-Gearman-Server-1.09-1.el6.noarch.rpm
- perl-IO-AIO-3.6-1.el6.rf.x86_64.rpm
- perl-IO-Socket-SSL-1.31-2.el6.noarch.rpm
- perl-IO-stringy-2.110-10.1.el6.noarch.rpm
- perl-MogileFS-Client-1.16-1.el6.noarch.rpm
- perl-Net-LibIDN-0.12-3.el6.x86_64.rpm
- perl-Net-Netmask-1.9015-4.el6.remi.noarch.rpm
- perl-Net-SSLeay-1.35-9.el6.x86_64.rpm
- perl-Perlbal-XS-HTTPHeaders-0.19-2.el6.x86_64.rpm
- perl-String-CRC32-1.4-9.el6.x86_64.rpm

perl-Sys-Syscall-0.23-1.el6.rf.noarch.rpm

Check your Linux Distribution for the presence of the following packages:

- lua-5.1.4-4.1.el6.x86_64.rpm
- php-mysql-5.3.3-22.el6.x86_64.rpm
- php-soap-5.3.3-22.el6.x86_64.rpm
- sysstat-9.0.4-20.el6.x86_64.rpm

Go into the Mogile installation kit directory and check that all the dependencies are solved by running the following command:

```
cd <installation_dir>/MogileInstallationKit  
rpm -ivh * --test
```

Expected output:

```
warning: lighttpd-1.4.28-2.el6.rf.x86_64.rpm: Header V3 DSA/SHA1 Signature, key  
ID 6b8d79e6: NOKEY  
warning: perl-IO-Socket-SSL-1.31-2.el6.noarch.rpm: Header V3 RSA/SHA256  
Signature, key ID f21541eb: NOKEY  
warning: perl-Net-Netmask-1.9015-4.el6.remi.noarch.rpm: Header V3 DSA/SHA1  
Signature, key ID 00f97f56: NOKEY  
Preparing... ##### [100%]
```

Run the command:

```
rpm -ivh *
```

Expected output:

```
warning: lighttpd-1.4.28-2.el6.rf.x86_64.rpm: Header V3 DSA/SHA1 Signature, key  
ID 6b8d79e6: NOKEY  
warning: perl-IO-Socket-SSL-1.31-2.el6.noarch.rpm: Header V3 RSA/SHA256  
Signature, key ID f21541eb: NOKEY  
warning: perl-Net-Netmask-1.9015-4.el6.remi.noarch.rpm: Header V3 DSA/SHA1  
Signature, key ID 00f97f56: NOKEY  
Preparing... ##### [100%]  
 1:mogstored-backend-none ##### [ 4%]  
 2:mogstored ##### [ 8%]  
 3:perl-Sys-Syscall ##### [ 13%]  
 4:perl-Danga-Socket ##### [ 17%]  
 5:perl-String-CRC32 ##### [ 21%]  
 6:perl-Gearman ##### [ 25%]  
 7:perl-Net-SSLeay ##### [ 29%]  
 8:perl-Net-Netmask ##### [ 33%]  
 9:perl-Net-LibIDN ##### [ 38%]  
10:perl-IO-Socket-SSL ##### [ 42%]  
11:perl-IO-stringy ##### [ 46%]  
12:perl-MogileFS-Client ##### [ 50%]  
13:perl-common-sense ##### [ 54%]  
14:perl-IO-AIO ##### [ 58%]  
15:perl-Perlbal-XS-HTTPHea##### [ 63%]  
16:Perlbal ##### [ 67%]  
17:lighttpd ##### [ 71%]  
18:mogstored-backend-light##### [ 75%]  
19:mogstored-backend-perlb##### [ 79%]  
20:MogileFS-Utills ##### [ 83%]  
21:mogilefsd ##### [ 88%]  
22:perl-Gearman-Client-Asy##### [ 92%]  
23:perl-Gearman-Server ##### [ 96%]  
24:mogstored-backend-apach##### [100%]
```

MYSQL Setup

Check mysql status by running the command:

```
service mysqld status
```

If the service is inactive start it and set root password:

```
service mysqld start  
mysqladmin -u root password 12qwas  
mysql -pmysql.12qwas
```

In mysql, create the mogilefs db and mogile user:

```
mysql> CREATE DATABASE mogilefs;  
mysql> GRANT ALL ON mogilefs.* TO 'mogile'@'%';  
mysql> SET PASSWORD FOR 'mogile'@'% ' = OLD_PASSWORD( 'mogile' );  
mysql> FLUSH PRIVILEGES;  
mysql> quit
```

Database Configuration

```
mogdbsetup --dbhost=localhost --dbrootpass=12qwas --dbname=mogilefs --  
dbuser=mogile --dbpassword=mogile --yes  
  
mysql -u mogile -pmogile -e"alter table mogilefs.file change length length  
bigint(20) UNSIGNED NULL;"
```

Trackers Configuration

Create a configuration file at /etc/mogilefs/mogilefsd.conf with the following content:

```
db_dsn DBI:mysql:mogilefs:localhost  
db_user mogile  
db_pass mogile  
conf_port 6001  
listen 0.0.0.0:7001  
node_timeout 5  
rebalance_ignore_missing 1  
listener_jobs 5
```

Modify tracker configuration file /etc/mogilefs/mogilefs.conf with the following setting:

```
trackers=localhost:7001
```

Repository Server Configuration

Create the repository directory (make sure it has access permissions for the user you will use to run mogstored)

```
mkdir -p /var/mogdata && chown -R mogstored:mogstored /var/mogdata
```

Configure the Repository Node by creating a configuration file at /etc/mogilefs/mogstored.conf with the following content:

```
httplisten=0.0.0.0:7500  
mgmtlisten=0.0.0.0:7501  
docroot=/var/mogdata
```

and start the mogile tracker daemons:

```
service mogilefsd start  
service mogstored start
```

NB: Ignore for the time being the following error

"Starting MogileFS tracker daemon: ERROR: Need to be root to increase max connections"

Set the services to be started on boot

```
chkconfig mogstored on  
chkconfig mogilefsd on
```

Add your host by running the command:

```
mogadm host add mogilestorage --ip=<yourIP> --port=7500 --status=alive
```

You can get confirmation that your host has been added with the following command:

```
mogadm host list
```

Expected output:

```
mogilestorage [1]: alive  
IP: <yourIP>:7500
```

Add devices BUT MAKE sure you NEVER re-use the same device ID: i.e. if /var/mogdata/devX exists, to create a new device you MUST create /var/mogdata/dev(X+1).

```
mogadm device add mogilestorage 1
```


This means "mogilestorage" has a host id of "1" and it has one device named "dev1" on it and that device is in the "alive" state.

Create the directory for the device with the proper permissions:

```
mkdir /var/mogdata/dev1 && chown -R mogstored:mogstored /var/mogdata
```

Finally, get confirmation that your devices are configured:

```
mogadm device list
```

Expected output:

```
mogilestorage [1]: alive
                    used(G) free(G) total(G)
dev1: alive         2.793   23.024  25.817
```

Create a domain (i.e. a namespace):

```
mogadm domain add scidip
```

Add a class to the domain:

```
mogadm class add scidip scidip-objects --mindevcount=1
```

Check domain and class creation:

```
mogadm domain list
```

Expected output:

domain	class	mindevcount	replpolicy	hashtype
scidip	default	2	MultipleHosts()	NONE
scidip	scidip-objects	1	MultipleHosts()	NONE

2.9 Uninstallation

Registry and Repository PHP modules

```
cd /etc/httpd/conf.d/
rm scidip.conf
mv scidip.conf.orig scidip.conf
mv scidip_storage.con.orig f scidip_storage.conf
```

```
rpm -e Scidip_Storage acs-apache-solr_scidip acs-apache-solr ACSPhpLib
```

3 Using SCIDIP-ES RepInfo Registry and Repository Services

3.1 Scidip Registry

3.1.1 Getting Started

The Registry Service REST API's is available from the url:

`http://<server_name>/ScidipRegistry`

The main access to the Registry is available at the URL:

`http://<server_name>/ScidipRegistry/RILProduct`

An example of Registry service is available at the integration platform URL:

<http://registry.scidip-es.eu/ScidipRegistry/RILProduct>

What you will get is:

```
<atom:feed xsi:schemaLocation="http://www.w3.org/2005/Atom ./atom.xsd
http://scidip.eo.esa.int/schema/catalogue ./ScidipRegistrySearchService.xsd
http://a9.com/-/spec/opensearch/1.1/
./OpenSearch.xsd"xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/"
xmlns:cs="http://scidip.eo.esa.int/schema/catalogue"
xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<opensearch:totalResults>0</opensearch:totalResults>
<opensearch:startIndex>0</opensearch:startIndex>
<opensearch:itemsPerPage>100</opensearch:itemsPerPage>
<opensearch:Query role="request" searchTerms="" startPage="0"/>
<atom:author>
<atom:name>Scidip Catalogue Server</atom:name>
</atom:author>
<atom:generator>Scidip Catalogue Server</atom:generator>
<atom:id>http://registry.scidip-es.eu/ScidipRegistry/RILProduct</atom:id>
<atom:title>Atom 1.0 Feed</atom:title>
<atom:updated/>
</atom:feed>
```

3.1.2 Getting a RepInfoLabel/Manifest by CPID from the Registry Service

Registry Service provides a RESTful interface based upon OpenSearch protocol. Each request can be made using plain HTTP 1.1 protocol.

The GET request to get a RepInfoLabel (or a Manifest) from the Registry is in the form:

http://<server_name>/ScidipRegistry/RILProduct?q=cpid:<CPID>

where:

<server_name> is your server name and <CPID> is the CPID of the RepInfoLabel or of the Manifest to be returned. Here is an example of a GET request to get RepInfoLabel with CPID "RIL-PDF-A_in_English" from the integration platform Registry:

[http://scidip-es.acsys.it/ScidipRegistry/RILProduct?q=cpid: RIL-PDF-A_in_English](http://scidip-es.acsys.it/ScidipRegistry/RILProduct?q=cpid:RIL-PDF-A_in_English)

The response to the get RepInfoLabel request is an atom:feed element including one atom:entry element. The response is encoded according to the API described in section 6.3. The formal definition of the interface is reported in section 6.3.

Here is an example of response for the GET request reported above:

```
<atom:feed xsi:schemaLocation="http://www.w3.org/2005/Atom ./atom.xsd
http://scidip.eo.esa.int/schema/catalogue ./ScidipRegistrySearchService.xsd
http://a9.com/-/spec/opensearch/1.1/
./OpenSearch.xsd"xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/"
xmlns:cs="http://scidip.eo.esa.int/schema/catalogue"
xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<opensearch:totalResults>1</opensearch:totalResults>
<opensearch:startIndex>0</opensearch:startIndex>
<opensearch:itemsPerPage>100</opensearch:itemsPerPage>
<opensearch:Query role="request" searchTerms="q=cpid:RIL-PDF-
A_in_English" startPage="0"/>
<atom:author><atom:name>Scidip Catalogue Server</atom:name></atom:author>
<atom:generator>Scidip Catalogue Server</atom:generator>
<atom:id>http://registry.scidip-es.eu/ScidipRegistry/RILProduct</atom:id>
<atom:title>Atom 1.0 Feed</atom:title>
<atom:updated/>
<atom:entry>
<atom:id>RIL-PDF-A_in_English</atom:id>
<atom:link rel="alternate" type="application/atom+xml" href="http://registry.sci
dip-es.eu/ScidipRegistry/RILProduct"/>
<atom:published/>
<atom:title>RIL-PDF-A_in_English</atom:title>
<atom:updated/>
<reinfo xsi:schemaLocation="http://preservedigital.info/ril
http://registry.scidip-
es.eu/repository/SCIDIP/ril.xsd" xmlns="http://preservedigital.info/ril"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<cpid type="RIL-PDF-A_in_English" category="Other/Software/Binary">RIL-PDF-
A_in_English</cpid>
<creator>SCIDIP-ES project</creator>
<lastmodified>2013-01-01T00:00:00Z</lastmodified>
<displayname>PDF-A in English</displayname>
<description>PDF-A in English</description>
<structuralRILList>
<cpid type="RIL-
English_text_in_ASCII" category="Structure/Formats/Specification">MAN-
definition_of_PDF-A
```

```
</cpid>
</structuralRILList>
<semanticRILList>
<cpid type="RIL-
English_text_in_ASCII" category="Semantic/Language/HumanWritten">
MAN-Placeholder_definition_of_English
</cpid>
</semanticRILList>
<otherRILList>
</otherRILList>
</repinfo>
</atom:entry>
</atom:feed>
```

3.1.3 Filtering results from the Registry Service

In order to filter results according to specific rules, a new query parameter has been added: 'fdq='

A composite query will have the following form:

```
http://<server_name>/ScidipRegistry/RILProduct?q=<solr_field_1>:<searched_value>
&fdq=<solr_field_2>:<filtering_value>
```

For example the following request:

http://registry.scidip-es.eu/ScidipRegistry/RILProduct?q=registryItemType:repinfo&fq=list_cpid_s:MAN-IPF_MERIS_processor

will provide you with a list of RepInfoLabel referencing manifest MAN-IPF_MERIS_processor.

For a detailed list of SOLR fields see Table Table 6-1 - Indexed fields

Note: among the SOLR fields, the "version" keyword one plays a particular role with the following rules:

1. If no version is specified the **last version** is returned
2. If **version:ALL** is specified all the versions of the required object(s) are returned
3. In order to have a specific version the corresponding field have to be specified in the fq parameter: **fq=version:<id_version>**

3.1.4 Inserting a new RepInfoLabel/Manifest into the Registry

To insert a new RepInfoLabel or a Manifest into the Registry, the HTTP POST operation is used.

HTTP POST operation, with xml file as payload, is addressed to:

```
http://<server_name>/ScidipRegistry/feed
```

by using the following CURL statement:

```
curl --data-binary @<full_xml_path> -H 'Content-Type: text/xml'  
http://<server_name>/ScidipRegistry/feed
```

For example, the following curl command inserts a new RIL (specified in file RIL-BEAM_software.xml) into the Registry:

```
curl --data-binary @ RIL-BEAM_software.xml -H 'Content-Type: text/xml'  
http://registry.scidip-es.eu/ScidipRegistry/feed
```

Where the RIL-BEAM_software.xml file is located in the local directory in which the command is launched and its content is reported here:

```
<?xml version="1.0" encoding="UTF-8" ?>  
<repinfo xmlns="http://preservedigital.info/ril"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://preservedigital.info/ril http://registry2.scidip-  
es.eu/repository/SCIDIP/ril_1.1.xsd">  
  
  <!-- An echo of the CPID requested for which this RIL describes -->  
  <cpid>RIL-BEAM_software</cpid>  
  <version>1393286400</version>  
  <lastmodified>2014-02-27T15:25:50</lastmodified>  
  <displayname>BEAM software</displayname>  
  <description>Software for viewing, analysing and processing of remote  
sensing raster data</description>  
  <structuralRILList>  
    <cpid type="RIL-English_text_in_ASCII" category="Structure">MAN-  
Missing_Structure_RepInfo_-_to_be_supplied_later</cpid>  
  </structuralRILList>  
  <semanticRILList>  
    <cpid type="RIL-XHTML" category="Semantic/Document">MAN-  
BEAM_user_manual</cpid>  
  </semanticRILList>  
  <otherRILList>  
    <or>  
      <cpid type="RIL-MAC_OS_executable" category="Other/Software">MAN-  
BEAM_MAC_OS_installation</cpid>  
      <cpid type="RIL-MS_Windows_executable"  
category="Other/Software">MAN-BEAM_MS_Windows_installation</cpid>  
    </or>  
  </otherRILList>  
</repinfo>
```

As illustrated in the above XML, the RepInfoLabel of the BEAM software class contains four more CPIDs (MAN-Missing_Structure_RepInfo_-_to_be_supplied_later, MAN-BEAM_user_manual, MAN-BEAM_MAC_OS_installation and MAN-BEAM_MS_Windows_installation), which link it to other four RepInfoLabels indirectly via their Manifest or directly via the CPID type attribute. The Manifest for MAN-BEAM_MAC_OS_installation is shown below:

```
<?xml version="1.0" encoding="UTF-8" ?>  
<manifest xmlns="http://preservedigital.info/manifest"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:schemaLocation="http://preservedigital.info/manifest
http://registry2.scidip-es.eu/repository/SCIDIP/manifest_1.1.xsd">
<cpid>MAN-BEAM_MAC_OS_installation</cpid>
<version>1393286400</version>
<rilcpid>RIL-MAC_OS_executable</rilcpid>
<categories>
  <categoryList>Other/Software</categoryList>
</categories>
<ppid>NONE</ppid>
<name>BEAM MAC OS installation</name>
<location>http://registry2.scidip-
es.eu/repository/demo1/MERIS/Software/BEAM/beam\_4.11\_macosx\_installer.dmg
</location>
</manifest>
```

The location field contains the address where the digital object is stored. In the above example, it is the Repository service available at the url:

<http://registry2.scidip-es.eu/repository/>

See Section 3.2 for more informaton.

3.1.5 Deleting a RepInfoLabel/Manifest- from the Registry

Deleting objects from Registry is allowed by accessing to the solr4 backend:

```
curl http://server_name:8080/solr4/products/update?withCommit=true -H "Content-
Type: text/xml" --data-binary '<delete><query>cpid:RepInfoLabelCPID (or
ManifestCPID)</query></delete>'

curl http://server_name:8080/solr4/products/update -H "Content-Type: text/xml" -
-data-binary '<commit/>'
```

NB. Remember to commit your changes all the time.

In order to “clean” the Registry you can run a delete query over all data by running the commands:

```
curl http://server_name:8080/solr4/products/update?withCommit=true -H "Content-
Type: text/xml" --data-binary '<delete><query>*:*/</query></delete>'

curl http://server_name:8080/solr4/products/update -H "Content-Type: text/xml"
--data-binary '<commit/>'
```

3.1.6 Exporting/Importing Registry content

Exporting the content Registry is allowed by running the following curl command:

```
curl -o filename.xml -H "Content-Type: application/xml"
'http://<server_name>/ScidipRegistry/export?q=*:*'
```

The exported file should have the following format:

```
<?xml version="1.0" encoding="utf-8"?>
<RegistryContent>
<item>
<repinfo> (or <manifest>)
...
</repinfo> (or </manifest>)
</item>
<item>
...
</item>
.
.
.
<item>
</item>
</RegistryContent>
```

You can reimport the dumped file into the registry with following command:

```
curl --data-binary @filename.xml -H "Content-Type: application/xml"
'http://<server_name>/ScidipRegistry/import'
```

3.2 Scidip Repository service

3.2.1 Getting Started

The Repository is accessible at the URL:

http://<server_name>/ScidipRepository/

The service allows user to store/download/delete digital objects from the MogileFS storage. The different functionalities are available at different endpoints as reported here below.

The Repository, available at the URL:

http://<server_name>/ScidipRepository/download/<DigitalObjectCPID>

allows user to download digital objects.

An example of Registry/Repository service is available on the integration platform URL:

<http://registry.scidip-es.eu/ScidipRepository/download/>

Append the CPID of the required digital object to download it into your browser.

3.2.2 Inserting a new DigitalObject into the Repository Service

The corresponding URL is: <http://<server name>/ScidipRepository/store> and the complete curl query is:

```
curl --data-binary @full_data_path -H 'Content-Type: <mime type>'
http://<server_name>/ScidipRepository/store/<digitalObjectCPID>
```

where **<mime type>** depends on the digital object type (usually application/pdf, text/plain, text/xml or application/octet-stream is used)

An example:

```
curl --data-binary @MERIS_Calibration_Paper_20120917.pdf -H 'Content-Type: application/pdf' http://registry.scidip-es.eu/ScidipRepository/store/MAN-MERIS_Calibration_Paper_20120917
```

NB. Note that the *url* used in the CURL statements above maps to the value in the “location” field of the corresponding object Manifest as shown in the Figure 3-1 and Error! Reference source not found..

3.3 Getting a Digital Object by CPID from the Repository Service

There two ways to get data from repository service: via the command line or web download:

By running the curl query:

```
curl -o outfile http://<server_name>/ScidipRepository/download/<objectCPID>
```

By opening a browser to the same url:

```
http://<server_name>/ScidipRepository/download/<objectCPID>
```

Example:

```
curl -o MERIS_Calibration_Paper_20120917.pdf http://esaregistry.scidip-es.eu/ScidipRepository/download/MAN-MERIS_Calibration_Paper_20120917
```

3.4 Deleting a Digital Object by CPID from the Repository Service

A curl query will delete the requested data:

```
curl http://<server_name>/ScidipRepository/delete/<objectCPID>
```



```

<semanticRIList>
  <and>
    <cpid type="RIL-PDF-A_in_English">MAN-ENVI01</cpid>

    <cpid type="RIL-PDF-A_in_English">MAN-ENVI93</cpid>

    <cpid type="RIL-PDF-A_in_English">MAN-MERIS_Calibration_Paper_20120917</cpid>

    <cpid type="RIL-PDF-A_in_English">MAN-MERISRRsecondreprocessing-V2</cpid>
  </and>
</semanticRIList>
  
```

Figure 3-1 Semantic part of the MERIS N1 data RIL

```

<manifest xmlns:schemaLocation="http://www.esa.int/inf/manifest http://registry2.scidip-es.eu/repository/SCIDIP/manifest.xsd" version="SCIDIP/0.1">
  <cpid>MAN-MERIS_Calibration_Paper_20120917</cpid>
  <rilocpid>RIL-PDF-A_in_English</rilocpid>
  <type>RIL-PDF-A_in_English</type>
  <ppid>00000000-0000-0000-0000-000000000000</ppid>
  <name>Documentation for MERIS N1 data</name>
  <location>http://esaregistry.scidip-es.eu/ScidipRepository/download/MAN-MERIS_Calibration_Paper_20120917</location>
</manifest>
  </atom:entry>
</atom:feed>
  
```

Figure 3-2 Manifest of MERIS Calibration Paper

4 Authentication and Authorization in SCIDIP-ES

This section describes the procedure to protect Scidip Registry and Repository by means of APACHE Basic Authentication and Authorization. The following procedure has to be done as **root** user (or a user with root privileges). Steps to be done are:

1. User creation
2. Groups creation
3. Apache configuration

A detailed description of each steps are described in the following sub-sections.

4.1 Create users

In order to create users you need to create a password file. We'll use a text password file. This file should be placed somewhere not accessible from the web. For example, put the password file `<passwords_file>` in `/etc/`

To create the file, use the ***htpasswd*** utility that came with Apache. This should be located in the bin directory of wherever you installed Apache. To create the file, type:

```
htpasswd -c <passwords_file> <username>
```

htpasswd will ask you for the password, and then ask you to type it again to confirm it. Start by creating the password file with the user "*useradmin*":

```
htpasswd -c /etc/basicusers useradmin
```

You will be asked:

```
New password: <your_password>  
Re-type new password: <your_password>
```

Add password for user *useradmin*.

In order to *add* other users use the same ***htpasswd*** utility without "*-c*" option.

Now create users "*scidipuser*" and "*guest*" with desired password:

```
htpasswd /etc/basicusers scidipuser  
htpasswd /etc/basicusers guest
```

4.2 Create Groups

Now you have to create the groups to implement an authorization policy. Create the file named "*basicgroups*" in `/etc/`

Each line of the group file contains a groupname followed by a colon, followed by the member usernames separated by spaces:

```
<groupname>: <username1> <username2> <username3>
```

The content of Scidip group file will be:

```
admin: useradmin  
user: scidipuser  
guest: guest
```

4.3 Configure Apache

Next, you'll need to configure the server to request a password and tell the server which users are authorized and which services they are allowed to access.

Apache configuration is done via the scidip.conf file under /etc/httpd/conf.d/

For each protected area (Directory section) you have to specify:

```
AuthType <auth_type>  
AuthName <auth_name>  
AuthBasicProvider <auth_source>  
AuthUserFile <passwords_file>  
AuthGroupFile <groups_file>  
Require group <group name>
```

Where:

<auth_type> is the authentication type i.e. *Basic*

<auth_name> is the name of your protected area i.e. *"Protected Area"*

<auth_source> is the source of authentication i.e. *file*

<passwords_file> is the whole path to the users and passwords file i.e. */etc/basicusers*

<groups_file> is the whole path to the groups file i.e. */etc/basicgroups*

<group name> is/are the allowed group/s

For each Directory you can authorize different groups to access to specific services by inserting the above mentioned directives inside a <Location> section.

The following table describes the authorization policy adopted for the defined groups:

		admin	guest	user
FindingAid	Find	X	X	X

Registry	Search	No Authentication is required		
	Feed	X		
	Import/Export	X		
Repository	Download	X		X
	Upload	X		

Table 4-1 Authorization policy

In line with Table 1, edit the scidip.conf and modify it as follows:

```
<VirtualHost *:80>
  ServerName <server_name>
  ServerAdmin admin@<server_name>

  ErrorLog logs/registry_error.log
  CustomLog logs/registry_access.log common

  <Directory "/var/www/html/acs-apache-solr_scidip/web">
    AllowOverride All
    Order allow,deny
    Allow from all
  </Directory>

  Alias /ScidipRegistry "/var/www/html/acs-apache-solr_scidip/web"

  <Location ~ ^(/ScidipRegistry/feed| ScidipRegistry/import|
ScidipRegistry/export)$>
    AuthType Basic
    AuthName "Protected Area"
    AuthBasicProvider file
    AuthUserFile /etc/basicusers
    AuthGroupFile /etc/basicgroups
    Require group admin
  </Location>

  RewriteEngine On
  RewriteCond %{REQUEST_URI} /ScidipRegistry/(.*)$
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteRule .* /acs-apache-solr_scidip/web/main.php [L,QSA]

  <Directory "/var/www/html/Scidip_Storage/web">
    AllowOverride All
    Order allow,deny
    Allow from all
    AuthType Basic
    AuthName "Protected Area"
    AuthBasicProvider file
    AuthUserFile /etc/basicusers
    AuthGroupFile /etc/basicgroups
    Require group admin
  </Directory>
```

```
Alias /ScidipRepository "/var/www/html/Scidip_Storage/web"

<Location ~
^(/ScidipRepository/download|/ScidipRepository/retrieve|/ScidipRepository/locate)$>
    AuthType Basic
    AuthName "Protected Area"
    AuthBasicProvider file
    AuthUserFile /etc/basicusers
    AuthGroupFile /etc/basicgroups
    Require group admin user
</Location>

RewriteCond %{REQUEST_URI} /ScidipRepository/(.*)$
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule .* /Scidip_Storage/web/main.php [L,QSA]

<Location ~ /FindingAidGui/(.*)$>
    AuthType Basic
    AuthName "Protected Area"
    AuthBasicProvider file
    AuthUserFile /etc/basicusers
    AuthGroupFile /etc/basicgroups
    Require group admin user guest
</Location>

RewriteCond %{REQUEST_URI} /FindingAidGui/(.*)$
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
#RewriteRule .* /FindingAidGui/index.html?q= [L,QSA]

</VirtualHost>
```

Restart apache:

```
service httpd restart
```

Go to

http://<server_name>/ScidipRegistry/export

A login pop-up will open, insert username and password (the user must be in "admin" group) and verify the authentication/authorization process.

4.4 Update Users and Groups

In order to update users and groups you have to edit the *basicusers* and *basicgroups* files.

4.4.1 Users Management

Add a user by running the command:

```
htpasswd <passwords_file> <username>
```

Delete a user by running the command:

```
htpasswd -D <passwords_file> <username>
```

NB: When deleting a user remember to update also the groups file: remove the corresponding username from the group he belongs to.

4.4.2 Groups

Groups update is done by simply editing the corresponding file /etc/basicgroups.

4.5 Curl commands update

When apache authentication/authorization is turned on all the curl commands previously mentioned have to be modified taking into account the user credentials, they will be modified as follows:

```
curl --basic --user <username>:<password> ...
```

An example is provided for the registry feeding:

```
curl --basic --user useradmin:12qwaszx --data-binary @MAN-test_new.xml --verbose -H  
"Content-Type: application/xml" 'http://<server_name>/ScidipRegistry/feed'
```

5 Scidip Registry and Repository Backup/Restore

5.1 Registry

Backup functionality consists simply in exporting the whole content of the Registry in XML format file.

Restore functionality is provided by the Import functionality applied to the exported XML file.

See Section 3.1.6 for a detailed Export/Import commands syntax description.

5.2 Repository

The *backup* process consists in 2 steps:

Mysql mogilefs database dump:

It is provided by the mysqldump utility. The syntax is here reported:

```
mysqldump -u root -p[root_password] [database_name] > [dump_filename.sql]
```

Run the following command (**as root**):

```
mysqldump -u root -p[root_password] mogilefs > mogilefs_dump.sql
```

Mogilefs storage device backup:

Compress all the mogile data into a tgz file:

```
cd /var && tar zcfv mogdata.tgz mogdata
```

The archive mogdata.tgz will contain all the mogile content.

The *restore* functionality consists of the mysql dump restore and the restore of the archive file in the proper directory:

```
mysql -u root -p[root_password] mogilefs < mogilefs_dump.sql  
cd /var && tar zxfv mogdata.tgz
```

NB: Note that the Storage Backup/Restore procedure is valid for mogile installation configuration described in Section 2.8.

6 Reference Manual

6.1 Keyboard shortcuts

None

6.2 *Command-line commands*

None

6.3 *Public APIs*

6.3.1 *API logic*

The current Registry API's are used by the core framework library to store/retrieve/browse registry objects (RIL and Manifest).

All these requests are HTTP GET returning XML payload.

6.3.2 *IF-SCIDIPES-RegistrySearch*

This interface allows a client to perform search and retrieve operation from the SCIDIP-ES RepInfo Registry Service.

This interface has been split in several sub-interfaces:

- **RILDescriptionDocument**: it returns the Open Search Description Document (OSDD), needed for issuing an Open Search query against the RIL stored in the dataset the OSDD is related to.
- **RILSearch**: it allows finding the RILs matching the search criteria specified.
- **RILRetrieve**: it allows retrieving the full set of information about a precisely identified RIL record.

The following sub-sections describe, for each of the above sub-interfaces, the request and response messages describing the tailoring / extension with respect to the Open Search [OS] protocol.

Applicable standards

This interface has been defined referring to the following standards:

[HTTP]: RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1

[OS]: OASIS OpenSearch - Search Web Services searchRetrieve Operation: Binding for OpenSearch Version 1.0 <http://docs.oasis-open.org/search-ws/v1.0/opensearch-v1.0.html>

Protocol

HTTP 1.1 GET

The response is an XML message conveyed in the HTTP response body.

RILDescriptionDocument

The RIL and Manifest templates are provided below:

```
<?xml version="1.0" encoding="UTF-8"?>
```



```
<repinfo xmlns="http://preservedigital.info/ril"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://preservedigital.info/ril http://registry2.scidip-
es.eu/repository/SCIDIP/ril_1.1.xsd">
  <cpid type="CPID of the RIL type Manifest">Cpid of the RIL</cpid>
  <version>CPID version</version>
  <creator>Creator Name</creator>
  <lastmodified>Creation/lastModified Date</lastmodified>
  <displayname>the name to be displayed</displayname>
  <description>A description</description>
  <structuralRILList>
    <cpid type="object type" category="object category(ies)">CPID of the
object Manifest</cpid>
  </structuralRILList>
  <semanticRILList>
    <cpid type="object type" category="object category(ies)">CPID of the
object Manifest</cpid>
  </semanticRILList>
  <otherRILList>
    <cpid type="object type" category="object category(ies)"> CPID of the
object Manifest</cpid>
  </otherRILList>
</repinfo>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest xmlns="http://preservedigital.info/manifest"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://preservedigital.info/manifest
http://registry2.scidip-es.eu/repository/SCIDIP/manifest1_1.xsd"
version="SCIDIP/0.1">
  <cpid>CPID of the Manifest</cpid>
  <version>CPID version</version>
  <rilcpid>CPID of the RIL describing the digital object</rilcpid>
  <type>CPID of the RIL describing the object type</type>
  <ppid>provenance ID</ppid>
  <category>object category(ies)</category>
  <location>URL where the digital object is stored</location>
</manifest>
```

RIL/ManifestSearch

This operation is an Open Search query for searching the RIL matching the specified filtering criteria.

This operation complies with the Open Search specification [OS].

RIL/ManifestSearch Request

The following table reports the tailoring of OpenSearch V1.1 parameters to perform RILSearch request.

Key	Value	Tailoring
q	searchTerms	The specified keyword is searched (LIKE) on the following RIL attributes: registryItemType cpid

		<pre> version cpid_version lastmodified displayname description category list_cpid_s list_type_cpid_s list_category_cpid_s list_version_cpid_s For the Manifest: cpid registryItemType categories version cpid_version rilcpid ppid location </pre>
--	--	--

Table 6-1 - Indexed fields

RILSearch Response

The response to the RILSearch request is an atom:feed element including one atom:entry element for each RIL record matching the search filter.

The response is built according to:

- The encoding of OpenSearch Response element;
- The encoding of RIL.

6.4 Authentication/Authorization - Getting Started

Here follows the main steps to protect Scidip Registry and Repository by means of APACHE Basic Authentication and Authorization. The following steps can be done only by **root** user (or as user having root privileges). The authorization policy adopted for the defined groups is summarized in the following table:

		admin	guest	user
FindingAid	Find	X	X	X
Registry	Search	No Authentication is required		
	Feed	X		
	Import/Export	X		
Repository	Download	X		X

	Upload	X		
--	---------------	---	--	--

6.4.1 Create users

We use a text password file. Put the password file <passwords_file> in /etc/
To create the file, use the *htpasswd* utility:

htpasswd -c <passwords_file> <username>

Start by creating the password file with the user "useradmin":

```
htpasswd -c /etc/basicusers useradmin
```

You will be asked:

New password: <your_password>

Re-type new password: <your_password>

Add password for user *useradmin*.

In order to *add* other users use the same *htpasswd* utility without "-c" option.

Now create users "scidipuser" and "guest" with desired password:

```
htpasswd /etc/basicusers scidipuser  
htpasswd /etc/basicusers guest
```

6.4.2 Create Groups

Create the file named "basicgroups" in /etc/

Each line of the group file contains a groupname followed by a colon, followed by the member usernames separated by spaces:

<groupname>: <username1> <username2> <username3>

The content of Scidip group file will be:

admin: useradmin

user: scidipuser

guest: guest

Add all the created users, if any

6.4.3 Configure Apache

Next, you'll need to configure the server to request a password and tell the server which users are authorized and which services they are allowed to access to.

Accordingly to the previous table, edit the `/etc/httpd/conf.d/scidip.conf` file and modify it as follows:

```
<VirtualHost *:80>
  ServerName <server_name>
  ServerAdmin admin@<server_name>

  ErrorLog logs/registry_error.log
  CustomLog logs/registry_access.log common

  <Directory "/var/www/html/acs-apache-solr_scidip/web">
    AllowOverride All
    Order allow,deny
    Allow from all
  </Directory>

  Alias /ScidipRegistry "/var/www/html/acs-apache-solr_scidip/web"

  <Location ~ ^(/ScidipRegistry/feed| ScidipRegistry/import| ScidipRegistry/export)$>
    AuthType Basic
    AuthName "Protected Area"
    AuthBasicProvider file
    AuthUserFile /etc/basicusers
    AuthGroupFile /etc/basicgroups
    Require group admin
  </Location>

  RewriteEngine On
  RewriteCond %{REQUEST_URI} /ScidipRegistry/(.*)$
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteRule .* /acs-apache-solr_scidip/web/main.php [L,QSA]

  <Directory "/var/www/html/Scidip_Storage/web">
    AllowOverride All
    Order allow,deny
    Allow from all
    AuthType Basic
    AuthName "Protected Area"
    AuthBasicProvider file
    AuthUserFile /etc/basicusers
    AuthGroupFile /etc/basicgroups
```

```
Require group admin
</Directory>

Alias /ScidipRepository "/var/www/html/Scidip_Storage/web"

<Location ~ ^(/ScidipRepository/download|/ScidipRepository/retrieve|/ScidipRepository/locate)$>
  AuthType Basic
  AuthName "Protected Area"
  AuthBasicProvider file
  AuthUserFile /etc/basicusers
  AuthGroupFile /etc/basicgroups
  Require group admin user
</Location>

RewriteCond %{REQUEST_URI} /ScidipRepository/(.*)$
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule .* /Scidip_Storage/web/main.php [L,QSA]

<Location ~ /FindingAidGui/(.*)$>
  AuthType Basic
  AuthName "Protected Area"
  AuthBasicProvider file
  AuthUserFile /etc/basicusers
  AuthGroupFile /etc/basicgroups
  Require group admin user guest
</Location>

RewriteCond %{REQUEST_URI} /FindingAidGui/(.*)$
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
#RewriteRule .* /FindingAidGui/index.html?q= [L,QSA]

</VirtualHost>
```

Restart apache:

```
service httpd restart
```

6.5 Registry - Getting Started

The Registry Service REST API's is available from the url

http://<server_name>/ScidipRegistry

The main access to the Registry is available at the URL

http://<server_name>/ScidipRegistry/RILProduct

An example of Registry service is available at the integration platform url

<http://registry.scidip-es.eu/ScidipRegistry/RILProduct>

6.5.1 Getting a RepInfoLabel/Manifest by CPID from the Registry Service

Registry Service provides a RESTful interface based upon OpenSearch protocol.

Each request can be made using plain HTTP 1.1 protocol.

The GET request to get a RepInfoLabel (or a Manifest) from the Registry is in the form:

http://<server_name>/ScidipRegistry/RILProduct?q=cpid:<CPID>

where:

<server_name> is your server name and **<CPID>** is the CPID of the RepInfoLabel or of the Manifest to be returned.

Here is an example of a GET request to get RepInfoLabel with CPID "RIL-PDF-A_in_English" from the integration platform Registry:

http://scidip-es.acsys.it/ScidipRegistry/RILProduct?q=cpid:RIL-PDF-A_in_English

6.5.2 Inserting a new RepInfoLabel/Manifest into the Registry

To insert a new RepInfoLabel or a Manifest into the Registry, the HTTP POST operation shall be used.

HTTP POST operation, with xml file as payload, shall be addressed to

http://<server_name>/ScidipRegistry/feed

by using the following CURL statement, using the basic authentication option:

```
curl --basic --user <username:password> --data-binary @<full_data_path> -H 'Content-Type: text/xml'  
http://<server_name>/ScidipRegistry/feed
```

For example, the curl following command inserts a new RIL (specified in file RIL-PDF-A_in_English.xml) into the Registry:

```
curl --data-binary @RIL-PDF-A_in_English.xml -H 'Content-Type: text/xml' http://registry.scidip-es.eu/ScidipRegistry/feed
```

Where the RIL-PDF-A_in_English.xml file is located in the local directory where the command is launched.

6.5.3 Deleting a RepInfoLabel/Manifest from the Registry

Deleting objects from Registry is allowed by accessing to the solr4 backend:

```
curl http://<server\_name>:8080/solr4/products/update?withCommit=true -H "Content-Type: text/xml" --data-binary '<delete><query>cpid:<RepInfoLabelCPID> </query></delete>'  
  
curl http://<yourSERVER>:8080/solr4/products/update -H "Content-Type: text/xml" --data-binary '<commit/>'
```

NB. Remember any time to commit your changes.

In order to “clean” the Registry you can run a delete query over all the data by running the commands:

```
curl http://<yourSERVER>:8080/solr4/products/update?withCommit=true -H "Content-Type: text/xml" --data-binary '<delete><query>*:*</query></delete>'  
  
curl http://<yourSERVER>:8080/solr4/products/update -H "Content-Type: text/xml" --data-binary '<commit/>'
```

6.5.4 Exporting/Importing Registry content

Exporting the content Registry is allowed by running the following curl:

```
• curl --basic --user <username:password> -o filename.xml -H "Content-Type: application/xml" 'http://<yourSERVER>/ScidipRegistry/export?q=*:*'
```

the exported file has the following format:

```
<?xml version="1.0" encoding="utf-8"?>  
<RegistryContent>  
<item>  
<repinfo> (or <manifest>)  
...  
</repinfo> (or </manifest>)  
</item>
```

```
<item>
...
</item>
.
.
.
<item>
</item>
</RegistryContent>
```

You can re-import into the registry the dumped file with following command:

```
• curl --basic --user <username:password> --data-binary @<filename.xml> -H "Content-Type:
application/xml" 'http://<yourSERVER>/ScidipRegistry/import'
```

6.6 Scidip Repository - Getting Started

The Repository, available at the URL

http://<server_name>/ScidipRepository/

allows the user to store/download/delete digital objects to the installed MogileFS storage depending on the endpoint used. Available functionality are here below reported.

6.6.1 Inserting a new DigitalObject into the Storage Service

The corresponding url is: <http://<server name>/ScidipRepository/store>

```
curl --basic --user <username:password> --data-binary @<full_data_path> -H 'Content-Type: <mime type> '
http://<server name>/ScidipRepository/store/<dataCPID>
```

where **<mime type>** depends on the digital object type (usually *application/pdf*, *text/plain*, *text/xml* or *application/octet-stream* is used)

An example:

```
curl --basic --user <username:password> --data-binary @MERIS_Calibration_Paper_20120917.pdf -H 'Content-
Type: text/plain' http://registry.scidip-es.eu/ScidipRepository/store/MAN-
MERIS_Calibration_Paper_20120917
```

NB. Note that the mentioned url is just the one placed in the “location” field of the corresponding object Manifest as shown in the Figures 3-1 and 3-2.

6.6.2 Getting a DigitalObject by CPID from the Storage Service

There are two ways to get data from storage service: via command line or via web download.

By running the curl query:

```
curl --basic --user <username:password> -o <outfilename>  
http://<server_name>/ScidipRepository/download/<objectCPID>
```

Or just by opening a browser to the same url:

```
http://<server_name>/ScidipRepository/download/<objectCPID>
```

Example:

```
curl --basic --user <username:password> -o MERIS_Calibration_Paper_20120917.pdf http://esaregistry.scidip-  
es.eu/ScidipRepository/download/MAN- MERIS_Calibration_Paper_20120917
```

6.6.3 Deleting a DigitalObject by CPID from the Storage Service

```
curl --basic --user <username:password> http://<server_name>/ScidipRepository/delete/<dataCPID>
```

7 Troubleshooting Common Issues

No Common Issues know.

8 Annex – Atom schema description

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
targetNamespace="http://www.w3.org/2005/Atom" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation>
      This version of the Atom schema is based on version 1.0
of the format specifications,
      found here
http://www.atomenabled.org/developers/syndication/atom-format-spec.php.
    </xs:documentation>
  </xs:annotation>
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/03/xml.xsd"/>
  <xs:annotation>
    <xs:documentation>
      An Atom document may have two root elements, feed and entry,
as defined in section 2.
    </xs:documentation>
  </xs:annotation>
  <xs:element name="feed" type="atom:feedType"/>
  <xs:element name="entry" type="atom:entryType"/>
  <xs:complexType name="textType" mixed="true">
    <xs:annotation>
      <xs:documentation>
        The Atom text construct is defined in section 3.1 of the
format spec.
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:any namespace="http://www.w3.org/1999/xhtml"
minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="type">
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="text"/>
          <xs:enumeration value="html"/>
          <xs:enumeration value="xhtml"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attributeGroup ref="atom:commonAttributes"/>
  </xs:complexType>
  <xs:complexType name="personType">
    <xs:annotation>
      <xs:documentation>
        The Atom person construct is defined in section 3.2 of
the format spec.
      </xs:documentation>
    </xs:annotation>
    <xs:choice maxOccurs="unbounded">
      <xs:element name="name" type="xs:string"/>
      <xs:element name="uri" type="atom:uriType" minOccurs="0"/>
    </xs:choice>
  </xs:complexType>
</xs:schema>
```

```
<xs:element name="email" type="atom:emailType" minOccurs="0"/>
<xs:any namespace="##other"/>
</xs:choice>
<xs:attributeGroup ref="atom:commonAttributes"/>
</xs:complexType>
<xs:simpleType name="emailType">
  <xs:annotation>
    <xs:documentation>
      Schema definition for an email address.
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
    <xs:pattern value="\w+@(\w+\.)+\w+"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="feedType">
  <xs:annotation>
    <xs:documentation>
      The Atom feed construct is defined in section 4.1.1 of
the format spec.
    </xs:documentation>
  </xs:annotation>
  <xs:choice minOccurs="3" maxOccurs="unbounded">
    <xs:element name="author" type="atom:personType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="category" type="atom:categoryType"
minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="contributor" type="atom:personType"
minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="generator" type="atom:generatorType"
minOccurs="0"/>
    <xs:element name="icon" type="atom:iconType" minOccurs="0"/>
    <xs:element name="id" type="atom:idType"/>
    <xs:element name="link" type="atom:linkType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="logo" type="atom:logoType" minOccurs="0"/>
    <xs:element name="rights" type="atom:textType" minOccurs="0"/>
    <xs:element name="subtitle" type="atom:textType"
minOccurs="0"/>
    <xs:element name="title" type="atom:textType"/>
    <xs:element name="updated" type="atom:dateTimeType"/>
    <xs:element name="entry" type="atom:entryType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:choice>
  <xs:attributeGroup ref="atom:commonAttributes"/>
</xs:complexType>
<xs:complexType name="entryType">
  <xs:annotation>
    <xs:documentation>
      The Atom entry construct is defined in section 4.1.2 of
the format spec.
    </xs:documentation>
  </xs:annotation>
  <xs:choice maxOccurs="unbounded">
    <xs:element name="author" type="atom:personType" minOccurs="0"
maxOccurs="unbounded"/>
```

```

        <xs:element name="category" type="atom:categoryType"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="content" type="atom:contentType"
minOccurs="0"/>
        <xs:element name="contributor" type="atom:personType"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="id" type="atom:idType"/>
        <xs:element name="link" type="atom:linkType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="published" type="atom:dateTimeType"
minOccurs="0"/>
        <xs:element name="rights" type="atom:textType" minOccurs="0"/>
        <xs:element name="source" type="atom:textType" minOccurs="0"/>
        <xs:element name="summary" type="atom:textType"
minOccurs="0"/>
        <xs:element name="title" type="atom:textType"/>
        <xs:element name="updated" type="atom:dateTimeType"/>
        <xs:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:choice>
    <xs:attributeGroup ref="atom:commonAttributes"/>
</xs:complexType>
<xs:complexType name="contentType" mixed="true">
    <xs:annotation>
        <xs:documentation>
            The Atom content construct is defined in section 4.1.3
of the format spec.
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="type" type="xs:string"/>
    <xs:attribute name="src" type="xs:anyURI"/>
    <xs:attributeGroup ref="atom:commonAttributes"/>
</xs:complexType>
<xs:complexType name="categoryType">
    <xs:annotation>
        <xs:documentation>
            The Atom cagegory construct is defined in section 4.2.2
of the format spec.
        </xs:documentation>
    </xs:annotation>
    <xs:attribute name="term" type="xs:string" use="required"/>
    <xs:attribute name="scheme" type="xs:anyURI" use="optional"/>
    <xs:attribute name="label" type="xs:string" use="optional"/>
    <xs:attributeGroup ref="atom:commonAttributes"/>
</xs:complexType>
<xs:complexType name="generatorType">
    <xs:annotation>
        <xs:documentation>
            The Atom generator element is defined in section 4.2.4
of the format spec.
        </xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
        <xs:extension base="xs:string">

```

```

        <xs:attribute name="uri" type="xs:anyURI"
use="optional"/>
        <xs:attribute name="version" type="xs:string"
use="optional"/>
        <xs:attributeGroup ref="atom:commonAttributes"/>
    </xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="iconType">
    <xs:annotation>
        <xs:documentation>
            The Atom icon construct is defined in section 4.2.5 of
the format spec.
        </xs:documentation>
    </xs:annotation>
</xs:simpleContent>
    <xs:extension base="xs:anyURI">
        <xs:attributeGroup ref="atom:commonAttributes"/>
    </xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="idType">
    <xs:annotation>
        <xs:documentation>
            The Atom id construct is defined in section 4.2.6 of the
format spec.
        </xs:documentation>
    </xs:annotation>
</xs:simpleContent>
    <xs:extension base="xs:anyURI">
        <xs:attributeGroup ref="atom:commonAttributes"/>
    </xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="linkType" mixed="true">
    <xs:annotation>
        <xs:documentation>
            The Atom link construct is defined in section 3.4 of the
format spec.
        </xs:documentation>
    </xs:annotation>
    <xs:attribute name="href" type="xs:anyURI" use="required"/>
    <xs:attribute name="rel" type="xs:string" use="optional"/>
    <xs:attribute name="type" type="xs:string" use="optional"/>
    <xs:attribute name="hreflang" type="xs:NMTOKEN" use="optional"/>
    <xs:attribute name="title" type="xs:string" use="optional"/>
    <xs:attribute name="length" type="xs:positiveInteger"
use="optional"/>
    <xs:attributeGroup ref="atom:commonAttributes"/>
</xs:complexType>
<xs:complexType name="logoType">
    <xs:annotation>
        <xs:documentation>
            The Atom logo construct is defined in section 4.2.8 of
the format spec.
        </xs:documentation>
    </xs:annotation>
</xs:simpleContent>
    <xs:extension base="xs:anyURI">

```

```

        <xs:attributeGroup ref="atom:commonAttributes"/>
    </xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="sourceType">
    <xs:annotation>
        <xs:documentation>
            The Atom source construct is defined in section 4.2.11
of the format spec.
        </xs:documentation>
    </xs:annotation>
    <xs:choice maxOccurs="unbounded">
        <xs:element name="author" type="atom:personType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="category" type="atom:categoryType"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="contributor" type="atom:personType"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="generator" type="atom:generatorType"
minOccurs="0"/>
        <xs:element name="icon" type="atom:iconType" minOccurs="0"/>
        <xs:element name="id" type="atom:idType" minOccurs="0"/>
        <xs:element name="link" type="atom:linkType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="logo" type="atom:logoType" minOccurs="0"/>
        <xs:element name="rights" type="atom:textType" minOccurs="0"/>
        <xs:element name="subtitle" type="atom:textType"
minOccurs="0"/>
        <xs:element name="title" type="atom:textType" minOccurs="0"/>
        <xs:element name="updated" type="atom:dateTimeType"
minOccurs="0"/>
        <xs:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:choice>
    <xs:attributeGroup ref="atom:commonAttributes"/>
</xs:complexType>
<xs:complexType name="uriType">
    <xs:simpleContent>
        <xs:extension base="xs:anyURI">
            <xs:attributeGroup ref="atom:commonAttributes"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="dateTimeType">
    <xs:simpleContent>
        <xs:extension base="xs:dateTime">
            <xs:attributeGroup ref="atom:commonAttributes"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:attributeGroup name="commonAttributes">
    <xs:attribute ref="xml:base"/>
    <xs:attribute ref="xml:lang"/>
    <xs:anyAttribute namespace="##other"/>
</xs:attributeGroup>
</xs:schema>

```

9 Annex – OpenSearch schema description

XML unofficial Schema:

```
<?xml version="1.0" encoding="utf-8"?>
<!-- thanks to http://weblogs.asp.net/wkriebel/archive/2008/02/04/opensearch-
xsd.aspx -->
<xs:schema xmlns="http://a9.com/-/spec/opensearch/1.1/"
xmlns:mstns="http://a9.com/-/spec/opensearch/1.1/"
xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://a9.com/-
/spec/opensearch/1.1/" elementFormDefault="qualified">
  <xs:annotation>
    <xs:documentation>Based on
http://www.opensearch.org/Specifications/OpenSearch/1.1</xs:documentation>
  </xs:annotation>
  <xs:element name="Query" type="mstns:QueryType">
    <xs:annotation>
      <xs:documentation>
        Defines a search query that can be performed by search clients.
Please see the OpenSearch Query element specification for more information.
OpenSearch description documents should include at least one Query element of
role="example" that is expected to return search results. Search clients may use
this example query to validate that the search engine is working properly.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="totalResults" type="xs:long"/>
  <xs:element name="startIndex" type="xs:long"/>
  <xs:element name="itemsPerPage" type="xs:long"/>
  <xs:element name="OpenSearchDescription">
    <xs:complexType>
      <xs:choice minOccurs="3" maxOccurs="unbounded">
        <xs:element name="ShortName">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="16"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Description">
          <xs:annotation>
            <xs:documentation>Contains a human-readable
text description of the search engine.</xs:documentation>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="1024"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Url" maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>Describes an interface by
which a search client can make search requests of the search engine.OpenSearch
provides support for both index-based and page-based search engines. By default,
both the first search result and the first page of search results are numbered
"1". Search engines can use the "indexOffset" and "pageOffset" attributes to
inform search clients of different starting values.</xs:documentation>
          </xs:annotation>

```



```

<xs:complexType>
  <xs:attribute name="template"
use="required">
    <xs:annotation>
      <xs:documentation>Contains the
search URL template to be processed according to the OpenSearch URL template
syntax. </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="type" use="required">
    <xs:annotation>
      <xs:documentation>Contains the
MIME type of the search result format.Restrictions: The value must be a valid
MIME type.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="rel" type="xs:string"/>
  <xs:attribute name="indexOffset"
use="optional" default="1">
    <xs:annotation>
      <xs:documentation>Contains the
index number of the first search result.</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction
base="xs:integer"/>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="pageOffset"
use="optional" default="1">
    <xs:annotation>
      <xs:documentation>Contains the
page number of the first set of search results.</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:restriction
base="xs:integer"/>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
</xs:element>
<xs:element name="Contact" type="xs:string"
minOccurs="0">
  <xs:annotation>
    <xs:documentation>Contains an email address
at which the maintainer of the description document can be reached.
Restrictions: The value must conform to the requirements of Section 3.4.1 "Addr-
spec specification" in RFC 2822.
  </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Tags" minOccurs="0">
  <xs:annotation>
    <xs:documentation>
Contains a set of words that are used as keywords to identify and categorize
this search content. Tags must be a single word and are delimited by the space
character (' '). Restriction: The value must not contain HTML or other markup.
  </xs:documentation>
  </xs:annotation>

```

```

        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="256"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="LongName" minOccurs="0">
        <xs:annotation>
            <xs:documentation>
Contains an extended human-readable title that identifies this search engine.
Search clients should use the value of the ShortName element if this element is
not available. Restrictions: The value must not contain HTML or other markup.
            </xs:documentation>
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="48"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="Image" minOccurs="0"
maxOccurs="unbounded">
        <xs:annotation>
            <xs:documentation> Contains a URL that
identifies the location of an image that can be used in association with this
search content. Image sizes are offered as a hint to the search client. The
search client will chose the most appropriate image for the available space and
should give preference to those listed first in the OpenSearch description
document. Square aspect ratios are recommended. When possible, search engines
should offer a 16x16 image of type "image/x-icon" or "image/vnd.microsoft.icon"
(the Microsoft ICON format) and a 64x64 image of type "image/jpeg" or
"image/png".
            Restrictions: The value must be a URI.
        </xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:anyURI">
                    <xs:attribute name="height"
type="xs:nonNegativeInteger" use="optional">
                        <xs:annotation>
                            <xs:documentation>
Contains the height, in pixels, of this image. </xs:documentation>
                        </xs:annotation>
                    </xs:attribute>
                    <xs:attribute name="width"
use="optional">
                        <xs:annotation>
                            <xs:documentation>
Contains the width, in pixels, of this image.</xs:documentation>
                        </xs:annotation>
                    </xs:simpleType>
                    <xs:restriction
base="xs:nonNegativeInteger"/>
                        </xs:simpleType>
                    </xs:attribute>
                    <xs:attribute name="type"
use="optional">
                        <xs:annotation>

```

```
<xs:documentation>
Contains the the MIME type of this image. Restrictions: The value must be a
valid MIME type. </xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="Query" type="mstns:QueryType"
minOccurs="0" maxOccurs="unbounded">
<xs:annotation>
<xs:documentation> Defines a search query
that can be performed by search clients. Please see the OpenSearch Query element
specification for more information. OpenSearch description documents should
include at least one Query element of role="example" that is expected to return
search results. Search clients may use this example query to validate that the
search engine is working properly. </xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="Developer" minOccurs="0">
<xs:annotation>
<xs:documentation> Contains the human-
readable name or identifier of the creator or maintainer of the description
document. The developer is the person or entity that created the description
document, and may or may not be the owner, author, or copyright holder of the
source of the content itself. Restrictions: The value must not contain HTML or
other markup. </xs:documentation>
</xs:annotation>
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:maxLength value="64"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="Attribution" minOccurs="0">
<xs:annotation>
<xs:documentation> Contains a list of all
sources or entities that should be credited for the content contained in the
search feed. Restrictions: The value must not contain HTML or other
markup.</xs:documentation>
</xs:annotation>
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:maxLength value="256"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="SyndicationRight" default="open"
minOccurs="0">
<xs:annotation>
<xs:documentation> Contains a value that
indicates the degree to which the search results provided by this search engine
can be queried, displayed, and redistributed. </xs:documentation>
</xs:annotation>
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="open">
<xs:annotation>
```

```
<xs:documentation> The
search client may request search results. The search client may display the
search results to end users. The search client may send the search results to
other search clients. </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="limited">
    <xs:annotation>
      <xs:documentation> The
search client may request search results. The search client may display the
search results to end users. The search client may not send the search results
to other search clients. </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="private">
    <xs:annotation>
      <xs:documentation> The
search client may request search results. The search client may not display the
search results to end users. The search client may not send the search results
to other search clients. </xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="closed">
    <xs:annotation>
      <xs:documentation> The
search client may not request search results.</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
</xs:restriction>
</xs:simpleType>
</xs:element>
  <xs:element name="AdultContent" type="xs:string"
default="false" minOccurs="0">
  <xs:annotation>
    <xs:documentation> Contains a boolean value
that should be set to true if the search results may contain material intended
only for adults. As there are no universally applicable guidelines as to what
constitutes "adult" content, the search engine should make a good faith effort
to indicate when there is a possibility that search results may contain material
inappropriate for all audiences.The values "false", "FALSE", "0", "no", and "NO"
will be considered boolean FALSE; all other strings will be considered boolean
TRUE.
  </xs:documentation>
  </xs:annotation>
</xs:element>
  <xs:element name="Language" type="xs:string" default="*"
minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation> Contains a string that
indicates that the search engine supports search results in the specified
language. An OpenSearch description document should include one "Language"
element for each language that the search engine supports. If the search engine
also supports queries for any arbitrary language then the OpenSearch description
document should include a Language element with a value of "*". The "language"
template parameter in the OpenSearch URL template can be used to allow the
search client to choose among the available languages.
Restrictions: The value must conform to the XML 1.0 Language Identification, as
specified by RFC 3066. In addition, the value of "*" will signify that the
```

search engine does not restrict search results to any particular language.</xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="InputEncoding" type="xs:string"
default="UTF-8" minOccurs="0" maxOccurs="unbounded">

<xs:annotation>

<xs:documentation>Contains a string that indicates that the search engine supports search requests encoded with the specified character encoding. An OpenSearch description document should include one "InputEncoding" element for each character encoding that can be used to encode search requests. The "inputEncoding" template parameter in the OpenSearch URL template can be used to require the search client to identify which encoding is being used to encode the current search request. Restrictions: The value must conform to the XML 1.0 Character Encodings, as specified by the IANA Character Set Assignments. </xs:documentation>

</xs:annotation>

</xs:element>

<xs:element name="OutputEncoding" type="xs:string"
default="UTF-8" minOccurs="0" maxOccurs="unbounded">

<xs:annotation>

<xs:documentation> Contains a string that indicates that the search engine supports search responses encoded with the specified character encoding. An OpenSearch description document should include one "OutputEncoding" element for each character encoding that can be used to encode search responses. The "outputEncoding" template parameter in the OpenSearch URL template can be used to allow the search client to choose a character encoding in the search response. Restrictions: The value must conform to the XML 1.0 Character Encodings, as specified by the IANA Character Set Assignments.

</xs:documentation>

</xs:annotation>

</xs:element>

<xs:any namespace="##other" minOccurs="0"

maxOccurs="unbounded"/>

</xs:choice>

</xs:complexType>

</xs:element>

<xs:complexType name="QueryType">

<xs:attribute name="role" use="required">

<xs:annotation>

<xs:documentation> role - Contains a string identifying how the search client should interpret the search request defined by this Query element. Role values: A role value consists of an optional prefix followed by the local role value. If the prefix is present it will be separated from the local role value with the ":" character. All role values are associated with a namespace, either implicitly in the case of local role values, or explicitly via a prefix in the case of fully qualified role values. Role extensibility: The role attribute may take on values beyond those specified in this document provided they are fully qualified with a prefix and associated with a declared namespace. Clients that encounter unrecognized role values should continue to process the document as if the Query element containing the unrecognized role value did not appear. Role prefix: A role prefix associates a local role name with a namespace. All prefixes must be previously declared as an XML namespace prefix on the containing Query element or ancestor elements. Local role values: Local role values are not preceded by a prefix. Local role values are associated with the OpenSearch 1.1 namespace. The following role values are identified with the OpenSearch 1.1 namespace. The list is exhaustive; only the role values

listed below may appear in the OpenSearch 1.1 namespace. Role values:

```
</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="request">
        <xs:annotation>
          <xs:documentation>
            Represents the search query that can be performed to
retrieve the same set of search results.
          </xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="example">
        <xs:annotation>
          <xs:documentation>
            Represents a search query that can be performed to
demonstrate the search engine.
          </xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="related">
        <xs:annotation>
          <xs:documentation>
            Represents a search query that can be performed to
retrieve similar but different search results.
          </xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="correction">
        <xs:annotation>
          <xs:documentation>
            Represents a search query that can be performed to
improve the result set, such as with a spelling correction.
          </xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="subset">
        <xs:annotation>
          <xs:documentation>
            Represents a search query that will narrow the current
set of search results.
          </xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="superset">
        <xs:annotation>
          <xs:documentation>
            Represents a search query that will broaden the current
set of search results.
          </xs:documentation>
        </xs:annotation>
      </xs:enumeration>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="title" use="optional">
  <xs:annotation>
    <xs:documentation>
```

Contains a human-readable plain text string describing the search request.

```
</xs:documentation>
</xs:annotation>
<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:maxLength value="256"/>
  </xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="totalResults" use="optional">
  <xs:annotation>
    <xs:documentation>
```

Contains the expected number of results to be found if the search request were made.

```
</xs:documentation>
</xs:annotation>
<xs:simpleType>
  <xs:restriction base="xs:nonNegativeInteger"/>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="searchTerms" use="optional">
  <xs:annotation>
    <xs:documentation>
```

Contains the value representing the "searchTerms" as an OpenSearch 1.1 parameter.

```
Restrictions: The value must be URL-encoded.
</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="count" use="optional">
  <xs:annotation>
    <xs:documentation>
```

Contains the value representing the "count" as a OpenSearch 1.1 parameter.

Search clients should anticipate that the value of the "count" parameter may not be honored by the search engine, and should rely exclusively on the contents of the "itemsPerPage" response element in calculating actual page size.

```
</xs:documentation>
</xs:annotation>
<xs:simpleType>
  <xs:restriction base="xs:nonNegativeInteger"/>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="startIndex" use="optional">
  <xs:annotation>
    <xs:documentation>
```

Contains the value representing the "startIndex" as an OpenSearch 1.1 parameter.

Replaced with the index of the first search result desired by the search client.

Default: The value specified by the "indexOffset" attribute of the containing Url element.

```
</xs:documentation>
</xs:annotation>
<xs:simpleType>
  <xs:restriction base="xs:integer"/>
</xs:simpleType>
```

```
</xs:attribute>
<xs:attribute name="startPage" use="optional">
  <xs:annotation>
    <xs:documentation>
      Contains the value representing the "startPage" as an
OpenSearch 1.1 parameter.
      Replaced with the page number of the set of search results
desired by the search client.
      Default: The value specified by the "pageOffset" attribute of
the containing Url element.
    </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:integer"/>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="language" type="xs:string" use="optional"
default="*">
  <xs:annotation>
    <xs:documentation>
      Contains the value representing the "language" as an
OpenSearch 1.1 parameter.
      Replaced with a string that indicates that the search client
desires search results in the specified language.
      An OpenSearch description document should include one
"Language" element for each language that the search engine supports. If the
search engine also supports queries for any arbitrary language then the
OpenSearch description document should include a Language element with a value
of "*". The "language" template parameter in the OpenSearch URL template can be
used to allow the search client to choose among the available languages.
      Restrictions: The value must conform to the XML 1.0 Language
Identification, as specified by RFC 3066. In addition, a value of "*" will
signify that the search client desires search results in any language.
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
<xs:attribute name="inputEncoding" type="xs:string" use="optional"
default="UTF-8">
  <xs:annotation>
    <xs:documentation>
      Contains the value representing the "inputEncoding" as an
OpenSearch 1.1 parameter.
      Replaced with a string that indicates that the search client
is performing the search request encoded with the specified character encoding.
      An OpenSearch description document should include one
"InputEncoding" element for each character encoding that can be used to encode
search requests. The "inputEncoding" template parameter in the OpenSearch URL
template can be used to require the search client to identify which encoding is
being used to encode the current search request.
      Restrictions: The value must conform to the XML 1.0 Character
Encodings, as specified by the IANA Character Set Assignments.
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
<xs:attribute name="outputEncoding" type="xs:string" use="optional"
default="UTF-8">
  <xs:annotation>
    <xs:documentation>
```


Replaced with a string that indicates that the search client desires a search response encoding with the specified character encoding.

An OpenSearch description document should include one "OutputEncoding" element for each character encoding that can be used to encode search responses. The "outputEncoding" template parameter in the OpenSearch URL template can be used to allow the search client to choose a character encoding in the search response.

Restrictions: The value must conform to the XML 1.0 Character Encodings, as specified by the IANA Character Set Assignments.

```
</xs:documentation>
  </xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:schema>
```

10 Annex – RIL schema description

Here is the ReplonfoLabel schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:jxb="http://java.sun.com/xml/ns/jaxb"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xjc="http://java.sun.com/xml/ns/jaxb/xjc"
  jxb:extensionBindingPrefixes="xjc"
targetNamespace="http://preservedigital.info/ril" elementFormDefault="qualified"
  jxb:version="2.0" attributeFormDefault="unqualified"
xmlns="http://preservedigital.info/ril">

  <xs:annotation>
    <xs:appinfo>
      <jxb:schemaBindings>
        <jxb:package name="eu.scidipes.common.framework.api.ril" />
        <jxb:nameXmlTransform>
          <jxb:elementName prefix="JAXB" />
        </jxb:nameXmlTransform>
      </jxb:schemaBindings>
    </xs:appinfo>
    <xs:documentation>RepInfo Label XML definition</xs:documentation>
  </xs:annotation>

  <xs:element name="repinfo">
    <xs:annotation>
      <xs:documentation>Root element of the Rep Info
Label</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="cpid" />
        <xs:element name="version" type="xs:positiveInteger">
          <xs:annotation>
            <xs:documentation>Version of this RIL. Versions, once on
a registry, are set and any alterations
to the RIL will result in a new version being
recorded by the registry</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="lastmodified" type="xs:dateTime">
          <xs:annotation>
            <xs:documentation>Date/time stamp that this version of
thsi RIL was uploaded</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="displayname" type="xs:string">
          <xs:annotation>
            <xs:documentation>Human readable name for this Rep Info
Label</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="description" type="xs:string">
          <xs:annotation>
            <xs:documentation>Detailed description on this Rep Info
Label and its intended usage
          </xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```

        </xs:annotation>
    </xs:element>
    <xs:element ref="structuralRILList" />
    <xs:element ref="semanticRILList" />
    <xs:element ref="otherRILList" />
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="structuralRILList">
    <xs:annotation>
        <xs:documentation>Structural Rep Info</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:complexContent>
            <xs:extension base="listType" />
        </xs:complexContent>
    </xs:complexType>
</xs:element>

<xs:element name="semanticRILList">
    <xs:annotation>
        <xs:documentation>Semantic Rep Info</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:complexContent>
            <xs:extension base="listType" />
        </xs:complexContent>
    </xs:complexType>
</xs:element>

<xs:element name="otherRILList">
    <xs:annotation>
        <xs:documentation>Other Rep Info</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:complexContent>
            <xs:extension base="listType" />
        </xs:complexContent>
    </xs:complexType>
</xs:element>

<xs:element name="or">
    <xs:annotation>
        <xs:documentation>Logical 'Or' Rep Info group</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:complexContent>
            <xs:extension base="andOrType" />
        </xs:complexContent>
    </xs:complexType>
</xs:element>

<xs:element name="and">
    <xs:annotation>
        <xs:documentation>Logical 'And' Rep Info group</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:complexContent>

```

```

        <xs:extension base="andOrType" />
    </xs:complexContent>
</xs:complexType>
</xs:element>

<xs:element name="cpid">
    <xs:annotation>
        <xs:documentation>The Curation Persistent Identifier of the resource
to be linked to.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="type" type="xs:string" use="optional">
                    <xs:annotation>
                        <xs:documentation>The Curation Persistent Identifier
of the RIL associated with the linked resource.</xs:documentation>
                    </xs:annotation>
                </xs:attribute>
                <xs:attribute name="category" type="xs:string"
use="optional">
                    <xs:annotation>
                        <xs:documentation>The category hierarchy which best
classifies this use of the linked resource</xs:documentation>
                    </xs:annotation>
                </xs:attribute>
                <xs:attribute name="version" type="xs:positiveInteger"
use="optional">
                    <xs:annotation>
                        <xs:documentation>The exact version of the resource
to link to</xs:documentation>
                    </xs:annotation>
                </xs:attribute>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>

<xs:complexType name="listType">
    <xs:annotation>
        <xs:documentation>Identifier or group directly under Structural,
Semantic or Other RI element
    </xs:documentation>
    </xs:annotation>
    <xs:choice minOccurs="0">
        <xs:element ref="and" />
        <xs:element ref="or" />
        <xs:element ref="cpid" />
    </xs:choice>
</xs:complexType>

<xs:complexType name="andOrType">
    <xs:annotation>
        <xs:documentation>At least two identifiers or groups to model RI
relationships within Structural, Semantic
or Other RI elements
    </xs:documentation>
    </xs:annotation>
    <xs:choice minOccurs="2" maxOccurs="unbounded">

```

```

        <xs:element ref="and" />
        <xs:element ref="or" />
        <xs:element ref="cpid" />
    </xs:choice>
</xs:complexType>

</xs:schema>

```

11 Annex – Manifest schema description

Here is the Manifest schema:

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:jxb="http://java.sun.com/xml/ns/jaxb"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xjc="http://java.sun.com/xml/ns/jaxb/xjc"
  jxb:extensionBindingPrefixes="xjc" jxb:version="2.0"
xmlns="http://preservedigital.info/manifest"
targetNamespace="http://preservedigital.info/manifest"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xs:annotation>
    <xs:appinfo>
      <jxb:schemaBindings>
        <jxb:package name="eu.scidipes.common.framework.api.manifest"/>
        <jxb:nameXmlTransform>
          <jxb:elementName prefix="JAXB" />
        </jxb:nameXmlTransform>
      </jxb:schemaBindings>
    </xs:appinfo>
    <xs:documentation>Manifest XML definition</xs:documentation>
  </xs:annotation>

  <xs:element name="manifest">
    <xs:annotation>
      <xs:documentation>Root element of the manifest</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="cpid" type="xs:string">
          <xs:annotation>
            <xs:documentation>Manifest Persistent
Identifier</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="version" type="xs:positiveInteger">
          <xs:annotation>
            <xs:documentation>Version of this Manifest. Versions,
once on a registry, are set and any alterations to the manifest will result in a
new version being recorded by the registry.</xs:documentation>
          </xs:annotation>

```

```

        </xs:element>
        <xs:element name="rilcpid" type="xs:string">
          <xs:annotation>
            <xs:documentation>RIL Persistent
Identifier</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element ref="categories" minOccurs="0" />
        <xs:element name="ppid" type="xs:string">
          <xs:annotation>
            <xs:documentation>Provenance Persistent Identifier (to
be defined)</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="name" type="xs:string" minOccurs="0">
          <xs:annotation>
            <xs:documentation>(Optional) Associated textual name of
object. Content is context dependent
(e.g. qualified classname for Java byte code).
          </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="location" type="xs:string">
          <xs:annotation>
            <xs:documentation>Object location. Usually a form of
URI</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="categories">
    <xs:annotation>
      <xs:documentation>Categories which this Manifest's object can be
used to represent</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="categoryList" type="xs:string">
          <xs:annotation>
            <xs:documentation>Category Name which this object can be
used to represent</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

12 Annex – RIL and Manifest examples

Here are two examples from the MERIS EO ESA's dataset model stored at the Registry at <http://registry2.scidip-es.eu>.

12.1 Example Manifest for MERIS N1 Scientific Data

```
<?xml version="1.0" encoding="UTF-8" ?>
<manifest xmlns="http://preservedigital.info/manifest"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://preservedigital.info/manifest
http://registry2.scidip-es.eu/repository/SCIDIP/manifest_1.1.xsd">
  <cpid>MAN-MERIS_N1_data_example</cpid>
  <version>1393286400</version>
  <rilocpid>RIL-2MERIS_N1_data</rilocpid>
  <categories>
    <categoryList>Other/ExampleData</categoryList>
  </categories><ppid>NONE</ppid>
  <name>MERIS N1 data example</name>
  <location>http://registry2.scidip-es.eu/repository/demo1/MISSING</location>
</manifest>
```

12.2 Example RIL for Meris N1 Scientific Data

```
<?xml version="1.0" encoding="UTF-8" ?>
<repinfo xmlns="http://preservedigital.info/ril"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://preservedigital.info/ril http://registry2.scidip-
es.eu/repository/SCIDIP/ril_1.1.xsd">

  <!-- An echo of the CPID requested for which this RIL describes -->
  <cpid type="RIL-2MERIS_N1_data">RIL-2MERIS_N1_data</cpid>
  <version>1392742488</version>
  <lastmodified>2013-01-01T00:00:00Z</lastmodified>
  <displayname>MERIS N1 data</displayname>
  <description>MERIS N1 data</description>
  <structuralRILList>
    <and>
      <cpid type="RIL-2PDF_1-5_in_English" category="Semantic/Document"
      >MAN-2MERIS_product_handbook</cpid>
      <cpid type="RIL-2PDF_1-4_in_English"
      category="Other/Software/Documentation" >MAN-2Issue_sheet</cpid>
      <cpid type="RIL-2PDF_1-2_in_English" category="Semantic/Document"
      >MAN-2Product_specification</cpid>
    </and>
  </structuralRILList>
  <semanticRILList>
    <and>
      <cpid type="RIL-2PDF_1-3_in_English" category="Semantic/Document"
      >MAN-2MERIS_Vol1</cpid>
```

```

<cpid type="RIL-2PDF-A_in_English" category="Semantic/Document"
>MAN-2DOC_MERIS_N1</cpid>
<cpid type="RIL-2PDF_1-4_in_English" category="Semantic/Document"
>MAN-2Resolution_Imaging_Spectrometer_details</cpid>
<cpid type="RIL-2PDF_1-4_in_English" category="Semantic/Document"
>MAN-2MERIS_Detailed_Instrument_Description</cpid>
<cpid type="RIL-2PDF_1-2_in_English" category="Semantic/Document"
>MAN-2MERIS_2nd_reprocessing:_Changes_Description</cpid>
<cpid type="RIL-2PDF_1-4_in_English" category="Semantic/Document"
>MAN-2MERIS_INSTRUMENT_CALIBRATION</cpid>
</and>
</semanticRILList>
<otherRILList>
  <or>
    <cpid type="RIL-2MS_Windows_executable" category="Other/Software"
    >MAN-2BEAM_MAC_OS_installation</cpid>
    <cpid type="RIL-2MAC_OS_executable" category="Other/Software"
    >MAN-2BEAM_MS_Windows_installation</cpid>
  </or>
</otherRILList>
</repinfo>

```

13 Figures and Tables

13.1 List of Figures

Figure 1 SOLR High Level internal Architecture	9
Figure 2 SOLR Index Update and Index Search - High Level Overview	10
Figure 3-1 Semantic part of the MERIS N1 data RIL.....	25
Figure 3-2 Manifest of the MERIS N1 data RIL.....	25

13.2 List of Tables

Table 4-1 Authorization policy	28
Table 6-1 - Indexed fields	34

14 Terminology

ACRONYM	DESCRIPTION
AIP	Archival Information Package
ARK	Archival Resource Key
CDMI	Cloud Management Interface
DOI	Digital Object Identifier
ES	Earth Science
GIS	Gap Identification Service
KB	Knowledge Base
OS	Orchestration Service
OWL	Web Ontology Language
PI	Persistent Identifier
PNM	Preservation Network Model
PURL	Persistent Uniform Resource Locator
RDF	Resource Description Framework
RepInfo	Representation Information
SNIA	Storage Networking Industry Association
SWKM	Semantic Web Knowledge Middleware
VM	Virtual Machine
WP	Work Package
XAM	eXtensible Access Method
XML	eXtensible Mark-up Language