**ECP-2007-DILI-537003**

**PEER**

# D3.1 Guidelines for publishers and repository managers on deposit, assisted deposit and self-archiving

| | |
|---|---|
| **Deliverable number/name** | *D-3.1* |
| **Dissemination level** | *Public* |
| **Delivery date** | *May 2009* |
| **Status** | *Final v.8.3* |
| **Author(s)** | *Magchiel Bijsterbosch, Foudil Brétel, Natasa Bulatovic, Dale Peters, Maurice Vanderfeesten, Julia Wallace* |
| **Internal Review** | *Christoph Bruch* |

***e*Content*plus***

This project is funded under the *e*Content*plus* programme[1], a multiannual Community programme to make digital content in Europe more accessible, usable and exploitable.

1 OJ L 79, 24.3.2005, p. 1.

# Table of Contents

# Tables, Figures & Appendices

## Tables



## Figures



## Appendices

# 1 Scope and Purpose

PEER (Publishing and the Ecology of European Research), supported by the EC eContentplus programme, will investigate the effects of the large-scale, systematic depositing of authors' peer-reviewed manuscripts (so called Green Open Access or stage-2[1] research output) on reader access, author visibility, and journal viability, as well as on the broader ecology of European research.

Guidelines documenting the procedures for publisher deposit; for author assisted deposit and self-archiving, and for transfer to participating PEER repositories are presented here by Work Package (WP) 3: *Repository Management and Reporting,* following extensive consultation with both target groups. In addition, the title implies the anticipated deposit by authors to repositories. No consultation with this group is foreseen in the project, with the intent to limit interference with established practise, and with the methodology of behavioural research in WP4. Both publishers and repository managers are thus advised in this document to refer authors to a generic helpdesk to be established in this work package.

The Guidelines set out in this document are based on workflows set out by WP2: *Repository Interface Framework*, in D2.1 *Draft report on the provision of usage data and manuscript deposit procedures for publishers and repository managers*. The workflow is illustrated in Figure 1 below. The agreed principles which underpin those procedures have been extrapolated from D2.1 and listed here in a user-friendly manner for ease of reference. These Guidelines should therefore be read in conjunction with D2.1

## 1.1 Publishers

The PEER project content comprises the contribution of approximately 11 publishers, who have agreed to participate in the project which aims to make available stage-2 outputs for 200 journal titles, in a research observatory. During the project more publishers will be invited to join the project to increase the number of journals to approximately 300 journal titles. To ensure that sufficient content is made available as a research sample to validate the research process, the publishers have agreed to collectively deposit 50% of the outputs on behalf of the authors. For the other 50%, publishers will invite the authors to self-archive their current manuscripts, and any previous manuscripts from participating journals.

## 1.2 Repositories

Authors are expected to follow their established practice of deposit in an institutional or subject-specific repository. Failing such practice, deposit in one or more of the PEER designated repositories is recommended. The Repository Task Force listed below was established in WP 2, comprising qualified representatives from the active repository community from EU countries. These six repositories, to which these Guidelines are primarily addressed, form an important research sample in PEER.

- Max-Planck-Gesellschaft zur Förderung der Wissenschaften e.V. (MPG)
- HAL, Institut National de Recherche en Informatique et en Automatique (INRIA)
- Göttingen State and University Library (UGOE)
- BiPrints, Universität Bielefeld (UNIBI)
- Kaunas University of Technology, Lithuania

1 The stage-2 version is the author's accepted manuscript which includes all the changes made as part of the peer-review process, but is not the final published version.

- University Library of Debrecen, Hungary

In accordance with DRIVER recommendations on the designation of a dedicated preservation agency, the e-Depot at the Koninklijke Bibliotheek in The Netherlands acts as a closed preservation repository, without participation in the usage measurement.

## 1.3 Workflow

A diagram of the PEER workflow shows the expected parallel paths of publisher deposit and author deposit, and structures the Guidelines set out in this document.

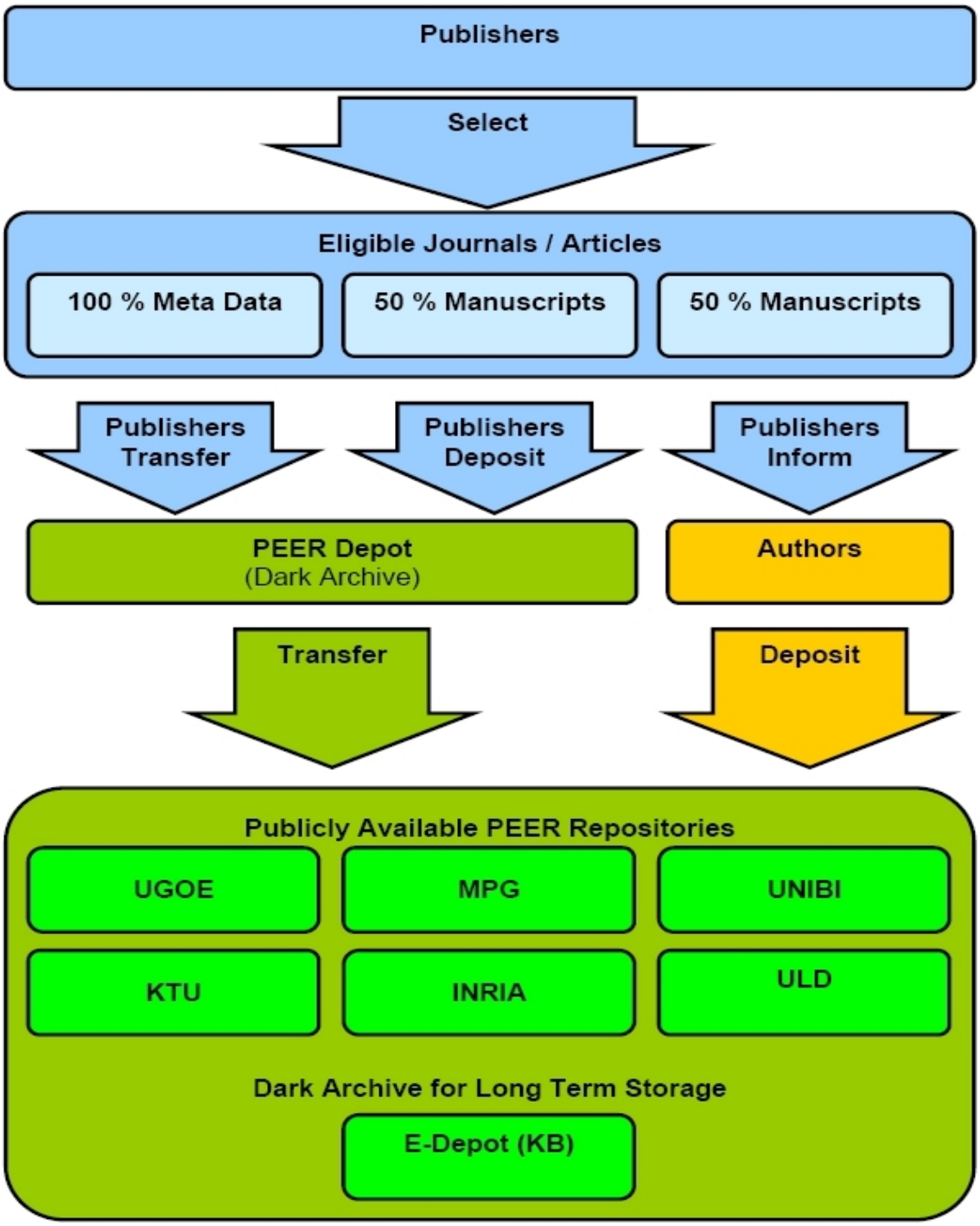


*Figure 1: PEER workflow*

# 2 Guidelines for Publishers

## 2.1 PEER Depot

The PEER Depot is established as a closed (dark) archive to receive the publisher deposit in the form of both 50% of the full-text outputs, as well as 100% of the metadata outputs, to serve as a base line control for the research process. The PEER Depot will conduct a preliminary pilot for a period of 4-6 weeks in May and June 2009, prior to full implementation of the project workflow, illustrated in Figure 1. In summary, the PEER Depot is a:

- **closed archive** (not accessible, nor searchable from the public internet)
- hosted at INRIA
- centralised point of collection for publisher deposits
- receives 100% metadata and 50% full-text outputs
- distributes 50% full-text outputs to all participating repositories

## 2.2 Transfer of Content to PEER Depot

Each Publisher has compiled a **profile** as a mechanism to monitor deposit rates at the PEER Depot. The profiles contain confidential information elements and are available for internal use only.

The following information is included in the profile:

- list of journals contributed to PEER
- schedule of publication frequency, number of issues per volume, etc.
- estimated total number of articles to be submitted per journal
- applicable metadata schemas
- submission file formats
- any specific issues that fall outside of the agreed deposit procedures

### 2.2.1 Deposit procedures

Publisher submissions to the PEER Depot will be conducted as follows:

- on a daily basis, as articles become available continuously
- files indicating failed conversion to .PDF-format are excluded
- submission by FTP/S[1] transmission or by SWORD protocol[2]
- ingest to a dedicated directory, one directory per publisher (not on journal level)
- transmission as zip files, one per article[3]
- file naming convention as [PublisherArticleId]_[yymmddhhmmss].zip[1]

---

1 FTP/SSL is a secure way to transfer files. The opensource command line tool **cURL** can be used as a FTPS client.

2 Recommended mechanism to alert content providers to failed deliveries.

3 A single zip file is essential to enable the PEER Depot to identify clearly each article, i.e. the material is not spread into many files that need to be gathered together.

- submission accompanied by an md5 checksum[2]
- metadata file contained in the zip file should include the name of the full-text file, with *.xml extension
- the zip package must contain only one obvious full-text file
- to identify PEER articles in repository logfiles, metadata file and full-text file will be renamed in the PEER Depot to `PEER_stage2_[urlencoded-DOI].pdf` before transfer to repositories
- three options for the submission of 50% full-text files to the PEER Depot include:
    - all required metadata are submitted at stage-2 deposit
    - only a subset of metadata is provided during the first deposit *including a publisher-article-id*; the rest is provided in a second deposit during the embargo period *including a publisher-article-id*[3]
    - all the metadata updated by the publisher at stage-3 is submitted again, in replacement of the stage-2 deposit (except the document, which remains stage-2)
- a submission is considered complete when the full-text and all required metadata are provided
- backfiles with only a DOI are currently unexploitable

### 2.2.2 Monitoring of publisher deposit rate

Publisher deposit rates are monitored biannually against the profile submitted. If after the first six months the overall submission proves insufficient as a research sample, additional content will be included.

- monitored against publisher profile
- conducted biannually by MPG
- sample size adjusted by STM if necessary

### 2.2.3 Full-text format

A wide range of file formats are identified in author submissions to publishers. The agreed submission format submitted by publishers to the PEER Depot is as follows:

- PDF/A-1
- PDF acceptable
- Conversion from single source file[4] undertaken at PEER Depot
- files indicating failed PDF conversion prior to transfer are excluded

---

1 The PublisherArticleId may not be the same article-id as in the metadata, but it must be some kind of unique alphanumerical identifier. 'yymmddhhmmss' is the date in the form: year in 2 digits, month, day, hour, minutes, seconds.
2 Each zip file is delivered along with its checksum file.
3 New metadata overwrite the previous version.
4 Tex, LaTex or Word *a priori*.

## 2.3 Metadata

Publisher profiles indicate a wide range of metadata schema deployed. Derived from the DRIVER Guidelines[1], the minimum required set of metadata elements common to all publisher submissions includes

- mandatory elements: Title, Creator, Date, Identifier and Type
- additional recommended elements (listed below) as available
- dc:description element, e.g. abstract, not routinely held in the publishers' metadata set but nevertheless recommended
- identical metadata formats for both 50% metadata plus full-text submission and for 50% metadata-only submissions
- input as NLM DTD (Journal Publishing Tag set) preferred
- input in XML required, with metadata file named with *.xml extension
- backfiles with only a DOI currently unexploitable

| DublinCore-like name | Comment |
|---|---|
| Title* | Article Title |
| Creator* | Corresponding Author's name: Last Name, First Name |
| AuthorEmail | Corresponding Author's email address |
| Description | Abstract |
| Date* | Date of Publication |
| Identifier* | DOI or PublisherArticleId |
| Coverage | Geographic location of the Contributing Author: ISO 3166-1-A2[2] |
| Journal | Journal Title[3] |
| Affiliation | multi-tier organisation list: Country, Organization, Laboratory |
| ISSN | These elements are not mandatory to electronic publication, and can be derived from CrossRef after DOI is provided, and may therefore not be provided by publishers. Possible use of CrossRef for DOI resolution. |
| Volume | |
| Issue | |
| Page | |

1 DRIVER Guidelines: http://www.driver-repository.eu/DRIVER-Guidelines.html.
2 Mapping to ISO 3166 provided by PEER Depot where required.
3 Conforms with titles in Journal table provided by publishers. See 11.3 in the PEER Description of Work.

| | |
|---|---|
| Type* | Default value = article. Mapped to *info:eu-repo/semantics/ article, info:eu-repo/semantics/acceptedVersion* |
| Subject | Subject headings; Scientific classification (defaults to what is provided in the Journal table)[1] |
| Language | ISO 639-1 (defaults to 'eng') |
| Embargo | Embargo Period (defaults to what is provided in Journal table)[2] |

*Table 1: Minimum metadata requirements*

## 2.4 Embargo period

The embargo period differs according to journal.

- Publishers include publication date in metadata set
- Publication date plus embargo period determines the date of distribution from the PEER Depot to participating repositories
- Embargo period determines release of author deposits previously received in participating repositories
- Authors are alerted to embargo period listed in journal table available on PEER helpdesk website

## 2.5 Filtering

Preliminary filtering by publishers is documented in the publisher profile.

- publisher filtering is provided by the journal list which ensures that the global set is 45% European

Further levels of filtering will take place at the PEER Depot:

- monitoring of the European percentage
- filtering by journal title for distribution of 50% full-text outputs to repositories
- filtering by country of European authors of corresponding author
- filtering by type of non-research papers (i.e. letters to the editor) is optional

1 See 11.3 in the PEER Description of Work.
2 See 11.3 in the PEER Description of Work.

# 3 Guidelines for Repository Managers

## 3.1 PEER Depot

The PEER Depot is established as a closed (dark) archive to receive the publisher deposit in the form of both 50% of the full-text outputs, as well as 100% of the metadata outputs, to serve as a base line control for the research process. The PEER Depot will conduct a preliminary pilot for a period of 4-6 weeks in May and June 2009, prior to full implementation of the project workflow, illustrated in Figure 1. In summary, the PEER Depot is a:

- **closed archive** (not accessible, nor searchable from the public internet)
- hosted at INRIA
- centralised point of collection for publisher deposits
- receives 100% metadata and 50% full-texts outputs
- distributes 50% full-text outputs to all participating repositories

## 3.2 Transfer of Content from PEER Depot to Repositories

A wide range of content formats submitted by publishers are normalised by the PEER Depot for transfer to participating repositories. The objective is to achieve a European core set of Repositories which can exchange material, and ultimately accept material directly from publishers. Minimal requirements for participating repositories are set out in the DRIVER Guidelines[1].

- participating repositories opt to set up a dedicated sub-repository exclusively for receipt of PEER content; or to add content to an existing repository
- additional effort in the ingest of PEER content is limited to the implementation of the SWORD protocol (see Appendix B)

### 3.2.1 Transfer procedures

The transfer of 50% full-text content from the PEER Depot will be conducted as follows:

- on a daily basis, as articles are normalised continuously
- submission by FTP/S[2] transmission[3] or SWORD protocol (see Appendix B)
- as zip files, one per article[4]
- the zip package contains only one PDF data file and one metadata file
- in order to identify PEER articles in repository logfiles, full-text file naming convention is as follows:

  `[PEER_stage2_[urlencoded-DOI].pdf]`

1 DRIVER Guidelines v.2.0: http://www.driver-repository.eu/DRIVER-Guidelines.html.

2 FTP/SSL is a secure way to transfer files. The opensource command line tool **cURL** can be used as a FTPS client.

3 FTP pull has two advantages: repositories do not have to install a FTP-server; and they have confirmation of successful ingest.

4 A single zip file is essential to enable the PEER Depot to identify clearly each article, i.e. the material is not spread into many files that need to be gathered together.

- submission accompanied by an md5 checksum[1]
- in the case of FTP/S, an acknowledgement file named

  `ack_PEER_stage2_[urlencoded-DOI].txt`

  comprising only the repository internal identifier in successful ingestion will be returned (void if unsuccessful)

## 3.3 Metadata

Publisher profiles indicate a wide range of metadata schema deployed. Derived from the DRIVER Guidelines[2], the minimum required set of metadata elements common to all publisher submissions, will be transferred to repositories:

- mandatory elements : Title, Creator, Date, Identifier and Type
- additional recommended elements (listed in Figure 1) as available
- PEER Depot transforms received metadata to TEI (see Appendix A)
- PEER Depot exports TEI files, or mapping to Dublin Core, as per repository preference

## 3.4 Embargo period

The embargo period differs according to each journal. A list of journal titles and corresponding embargo periods will be made publically available on the Helpdesk site, in the Journal table provided by publishers.[3]

- Publication date plus embargo period determines the date of distribution from the PEER Depot to participating repositories. This includes 100% metadata and 50% full-text comprising publisher deposit.
- Repositories hold any content previously received by author deposit until matching metadata is received from PEER Depot.
- Matching metadata determines the release of author deposits in participating repositories, following expiration of the embargo period.

## 3.5 Usage data provision

Participating repositories need to provide a minimum set of data to enable the analysis of the usage data provision:

- Logfiles and description of their structure
- Identifier mapping files

### 3.5.1 Logfiles and structure

As described in D2.1[1], various participating repositories provide various formats for their usage statistics. It has been agreed that each participating repository delivers the usage statistics as raw data[2] and in the NSCA combined logfile format (see Table 2).

1 Each zip file is delivered along with its checksum file.
2 DRIVER Guidelines: http://www.driver-repository.eu/DRIVER-Guidelines.html.
3 See 11.3 in the PEER Description of Work.

| Log entry fields | Mandatory /Optional | Example | Description |
|---|---|---|---|
| host | Mandatory | 125.125.125.125 | The IP-address or host/subdomain name of the HTTP client that made the HTTP resource request. |
| rfc931 | Optional | - | The identifier used to identify the client making the HTTP request. If no value is present, a "-" is substituted. |
| username | Optional | Jdoe user:37676 | The username, (or user ID) used by the client for authentication. If no value is present, a "-" is substituted. |
| date:time | Mandatory | 10/Oct/1999:21:15:05 +0500 | The date and time stamp of the HTTP request. |
| Request | Mandatory | "GET /peer01.pdf HTTP/1.0" | The HTTP request. The request field contains three pieces of information. The main piece is the requested resource (index.html). The request field also contains the HTTP method (GET) and the HTTP protocol version (1.0). |
| statuscode | Optional | 200 | The status is the numeric code indicating the success or failure of the HTTP request. |
| Bytes | Optional | 1043 | The bytes field is a numeric field containing the number of bytes of data transferred as part of the HTTP request, not including the HTTP header. |
| referer | Mandatory | http://www.google.com | The URL which linked the user to the site (optional). |
| user_agent | Optional | "Mozilla/5.0“ | The Web browser and platform used by the visitor to your site (optional). |
| cookies | Optional | ("USERID=XXX;IMPID =01234") | Cookies take the form KEY = VALUE. Multiple cookie key-value pairs are delineated by semicolons (;). |

*Table 2: Description of the NCSA Combined logfile format*

Participating repositories may have deviations to format expressed in Table 2, e.g.:

- optional fields are not populated or anonymised for reasons of data privacy
- the URL forms of requests have diverse structures

---

1 PEER D2.1 Draft report on logfile harvesting systems and manuscript deposit procedures for publishers and repository managers, http://www.peerproject.eu/reports/

2 Whenever participating repositories are anonymising the username or IP-addresses for reasons of data privacy, this is still considered as raw data.

Therefore the logfile description is standardised as follows:

- provide the pattern of the log entry fields used, e.g.

  LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined
  CustomLog log/access_log combined

- provide short description for each field in the pattern, with an example
- for **request** field in the log entry provide the pattern of the URL corresponding to a retrieval of a repository item and download of a file attached to the repository item
- the pattern must clearly describe what part of it is used as identifier of the repository item or as an identifier of the attached file respectively
- based on an input from the research team, participating repositories should provide description of URL patterns used for other particular types of request[1].

### 3.5.2 Identifier mapping files

In case when logfiles do not provide a PEER identifier of the resource in the request field or in other log entry fields, repositories provide additional information to enable to map the identifier of the repository resource to the appropriate PEER identifier. For each PEER resource that is created in the repository, the mapping file contains the following entries:

- Peer resource ID (`PEER_stage2_[urlencoded-DOI]`)
- Repository identifier value used in the logfile
- pattern in which the repository identifier value is used

1 At present this input is not yet known as the research team is not yet established.

# 4 PEER Helpdesk

Based on the precedent set in the DRIVER Support site[1], these Guidelines should be disseminated to the relevant stakeholder communities identified in the publishers, authors and repository managers. The PEER website[2] provides a suitable mechanism to achieve this dissemination through collaboration between WP 3 & WP 8.

## 4.1 PEER Helpdesk functions

A PEER Helpdesk, linked from the PEER website, will be developed in the implementation phase of the project, and promoted as an authoritative source of information. This is envisaged as a key point of contact for all the stakeholder communities participating in PEER. As all these actors − publishers, authors and repository managers − are likely to call upon a support facility, a telephone hotline would overly burden the support team.

The Helpdesk as an online interface will facilitate outreach and information provision activities and will in particular provide advice and support on the implementation of these Guidelines, and questions of deposit and transfer, as described in D2.1.

The Helpdesk will offer direct support by means of an online query and mediated response service throughout the project duration. Current investigation of automated sytems is based on the following project criteria:

- meeting the diverse needs of three identified stakeholder communities
- efficient query handling and response mechanisms
- handling of specific query behavior on predetermined information-seeking tasks
- documentation of query results for future reference, in the form of frequently asked questions (FAQs).

The Helpdesk system also provides a mechanism of passive interaction for those seeking assistance, but unwilling to ask – a notable online query behaviour pattern. Given the increased importance attached to usability features therefore, systems evaluation will be conducted jointly by members of WP 3 & WP 8.

Technically, the support facility may be implemented in the form of a ticket system (such as *Trac* or *Request Tracker/RT*).

- frequently asked questions (FAQs) will be developed and published on the Helpdesk site, based on that established in DRIVER[3]
- a ticketing system is highly effective since the questions and answers are well documented
- the results can be published, and the participants are able to review issues as they arise
- where the ticketing system is made public, the “wisdom of crowds”-principle can be applied to gain more efficient response to complex problems

1 DRIVER Support website: http://www.driver-support.eu/.
2 PEER website: http://www.peerproject.eu/.
3 DRIVER Helpdesk: http://helpdesk.driver.research-infrastructures.eu/.

## 4.2 Helpdesk for Publishers and Repository Managers

Publishers will deposit both 50% of the full-text outputs, as well as 100% of the metadata outputs from eligible journals at the PEER Depot (see Ch. 2). The 50% full-text outputs will be pushed from the PEER Depot to the repositories participating in PEER (see Ch. 3).

Following extensive consultation with both publisher and repository communities, it is expected that implementing these Guidelines will be straightforward. The Helpdesk will, however, support the consistent explanation and information on guiding publishers and repository managers through the deposition process.

The support for publishers is provided by experts resp. representatives of INRIA and the KB, and is expected to cover queries regarding:

- metadata schema
- transfer procedures
- deviations from profile submitted, etc.

The support for repository managers is provided by expert representatives from SURF and MPG, and is expected to cover queries regarding:

- how to obtain the “NSCA combined” logfile format, if not directly available
- this might entail the provision of scripts for mapping from other formats
- use of the PEER-filename in the repository
- advice on the corresponding interface to implement the SWORD protocol, etc.

## 4.3 Helpdesk for Authors

### 4.3.1 Guidance for authors on deposit procedures

For reasons of data privacy, the participating publishers are not able to make available the contact details of eligible authors, and no direct communication is envisaged. The Guidelines outlined here are therefore not directed at the author community directly, but rather they reflect the considered opionion of the work package in consultation with the publisher community on recommended practice in offering assistance to authors.

Two main objectives of the guidance aimed at authors are:

- self-selection of European authors
- self-archiving by two anticipated routes:
    - by following established practice of deposit in an institutional or subject-specific repository
    - failing such practice, by deposit in one or more of the recommended PEER designated repositories

The PEER Helpdesk will not only offer guidance to publishers and repository managers, often already involved in large-scale archiving, but also to authors, who may need guidance in self-archiving for the first time. A recent study by Swan indicates that a substantial proportion of the author population (36%) are unaware of the possibility of providing Open Access to their work by self-archiving, and that only 49% of the author population have self-archived in some way. Of relevance to the PEER Helpdesk is the observation that authors have frequently expressed reluctance to self-archive because of the perceived time

required and possible technical difficulties in carrying out this activity. However, similar findings suggest that only 20% of authors found some degree of difficulty with the first act of depositing an article in a repository, and that this dropped to 9% for subsequent deposits.[1]

The PEER Helpdesk aims to:

- provide general information on the objectives of the PEER project
- limit disruption of established author practice in self-archiving
- limit confusion on self-selection of authors based in the European Union
- reduce level of perceived technical difficulty in self-archiving
- provide simple explanations to authors on repository submission procedures
- host an interactive demonstration of a step-by-step deposit procedure
- provide basic explanations of authors' intellectual property rights
- encourage adherence to varied embargo periods, as determined by publishers according to each journal

### 4.3.2 Author communication

Authors eligible for participation in the PEER project will be notified via the relevant publisher. The author deposit procedure is envisaged in alignment with the normal points of contact between publishers and authors.

- On submission of a manuscript to an eligible journal, authors are informed by the publisher about PEER and its objectives (see Appendix C).
- On acceptance of their article, the author receives an invitation to self-select on the basis of EU authors, and to self-archive the stage-2 manuscript in one of the participating PEER repositories (see Appendix C).
- A link to the PEER Helpdesk is included in the invitation.
- The request for deposition includes a plea to inform the project of the target repository, should the author intend to deposit in a repository of choice, other than one of the specified PEER repositories.
- Authors may prefer not to respond to the invitation to self-archive, and/or not to inform the project of that preference.
- Many of the publishers have the right to deposit the stage-2 articles as part of the publishing license with the author. If an author objects to the Open Access publication of his/her article(s) through the publisher deposit workflow, the publisher concerned will consider the objection, and instruct the PEER Depot to remove the respective article(s), if necessary.

### 4.3.3 Monitoring author response

It is not possible to predict the behaviour of authors invited to self-archive a stage-2 manuscript. In terms of the project research activities, deposition will be monitored by the behavioural research undertaken in WP4, and measured against the 100% metadata control managed by the PEER Depot.

---

1 SWAN, A. & BROWN, S. (2005) *Open access self-archiving: An author study.* http://cogprints.org/4385/

However, it is expected that authors will respond immediately to the invitation − or not at all. For this reason, it may be necessary for PEER repositories to monitor:

- the rate of author deposit
- adherence to varied embargo periods, as determined by publishers according to each journal

# 5 Conclusions

These Guidelines serve to distill the procedures set out in D2.1 *Draft report on the provision of usage data and manuscript deposit procedures for publishers and repository managers*. The concise style and bulleted layout are intended to offer easy access, as a quick reference tool. It is therefore recommended that this document is consulted in conjunction with D2.1.

In addition, the report sets out a major advance in the repository practice, in the use of the SWORD protocol, the specification for which is included in Appendix B. This document is presented as a cohesive sub-report, as it is expected that it may become a ready reference tool in its own right.

By their very nature, these Guidelines are expected to be dynamic, and certain points may be revised and additions be made to this document, as agreed procedures are tested in their implementation over the coming months.

# Appendix A. TEI components of a PEER metadata format

This annex describes the output format that will be adopted by the PEER Depot (Partner: INRIA) for distributing the metadata information provided by publishers. It is based on the TEI[1] guidelines, with some additional constraints intended to make the corresponding information structures universally interpretable.

## 1 Overview

The proposed structure combines a global structure (<TEI>[2]), which can potentially integrate any information that can be found in a full-text representation of a paper article, and a sub-structure (<biblStruct>[3]) that specifically contains the bibliographical information of the article. This allows us to process in a uniform way the two following scenarios:

- The PEER Depot receives full-text articles in XML (or retrieves them from repositories such as PMC) and converts them to the TEI format, thus exploiting all its expressive capacities.
- The PEER Depot receives specific metadata information, with possibly some additional content (e.g. abstract). A highly simplified <TEI> structure is created, which is mainly a container for disseminating the bibliographical content.

The remaining part of this document will primarily address the second scenario, which is the one needed for the research to be carried out within the PEER project.

## 2 General structure of a TEI document

The TEI information model is intended to represent both the textual content of a document and the metadata attached to it. This is reflected in the two main parts of a <TEI> root element, namely <TEIHeader> and <text>.

The TEI header is in turn organised in a series of sub-components:

- <fileDesc> gathering the main characteristics of the document (title, author, bibliographic description of the source)
- <profileDesc> providing some information about the content (e.g. languages used in the text, keywords)
- <revisionDesc> providing the history of the document

The <text> element is further decomposed in <front>, <body> and <back>. Where available, abstracts are represented in <front> and full-text content in subsequent elements.

## 3 Skeleton of a full TEI document (as relevant for PEER)

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
    <teiHeader>
        <fileDesc>
            <titleStmt>
                <title level="a" type="main">...</title>
            </titleStmt>
            <publicationStmt>
                <availability>
                    <p>Copyright © The Animal Consortium 2009</p>
```

1 Text Encoding Initiative (www.tei-c.org)
2 http://www.tei-c.org/release/doc/tei-p5-doc/html/ref-TEI.html
3 http://www.tei-c.org/release/doc/tei-p5-doc/html/ref-biblStruct.html

```
                </availability>
                <date>2009</date>
                <authority>The Animal Consortium</authority>
            </publicationStmt>
            <sourceDesc>
                <biblStruct>...</biblStruct>
            </sourceDesc>
        </fileDesc>
        <profileDesc>
            <textClass>
                <keywords>
                    <list>
                        <head>Keywords</head>
                        <item>
                            <term>foetal development</term>
                        </item>
                        <item>
                            ...
                        </item>
                    </list>
                </keywords>
            </textClass>
        </profileDesc>
        <revisionDesc>
            <change when="2008-08-27">Received</change>
            <change when="2008-12-01">Accepted</change>
        </revisionDesc>
    </teiHeader>
    <text>
        <front>
            <div type="abstract">
                <head>Abstract</head>
                <p>...</p>
            </div>
        </front>
        <body/>
        <back/>
    </text>
</TEI>
```

## 4 Representation of bibliographical information

The representation is based on the TEI <biblStruct> element, which is organised as follows:

```
<biblStruct type="article">
    <analytic>
        …
    </analytic>
    <monogr>
        …
        <imprint>
            …
        </imprint>
    </monogr>
    …
</biblStruct>
```

A <biblStruct> is mainly divided into two sub-structures:

- <analytic> to indicate the bibliographical characteristics of an article (title and authors)
- <monogr> to account for the publication details of the journal (journal name, publisher information, ISSN, etc.), and contains in turn an <imprint> element which gathers publication and/or distribution aspects of the article in the corresponding journal (pagination, volume, issue, etc.)

When applicable, additional notes or identifiers can follow, for instance, the DOI, pubmed-central-id or repository-specific-id will appear here:

```
<biblStruct type="article">
    <analytic>…</analytic>
    <monogr>…</monogr>
    <idno type="pmid">12345678</idno>
</biblStruct>
```

### 4.1 The <analytic> element

**Overview**

The title of a journal article is represented by means of the <title> element (with appropriate @level attribute) as follows:

```
<title level="a">Multilocus Analysis of Age Related Macular
Degeneration</title>
```

When necessary a further @type attribute may be used to differentiate between main and subtitles (@type=”main” vs. @type=”subordinate”).

Each author in the <analytic> element is independently described by means of an <author> element. This element contains the author’s name, affiliation and addresses − when available − as presented in the outline below:

```
<author>
     <idno type="...">...</idno>
     <persName>
        <forename>Michael</forename>
        <surname>Dean</surname>
     </persName>
     <affiliation>…</affiliation>
     <email>dean@ncifcrf.gov</email>
</author>
```

**Dealing with affiliations**

The <affiliation> component of <author> is intended to contain any potentially relevant information with regard to the author’s academic situation: research group, laboratory, institution.

```
<affiliation>
   <orgName type="laboratory">CSA Department</orgName>
   <orgName type="institution">Indian Institute of Science</orgName>
   <address>
      <settlement>Bangalore</settlement>
      <postCode>560012</postCode>
      <country>India</country>
      <addrLine type="phone">+91-80-22932386</addrLine>
      <addrLine type="fax">+91-80-23602911</addrLine>
   </address>
</affiliation>
<email>kavitha@csa.iisc.ernet.in</email>
```

### 4.2 The <monogr> element

The <monogr> element gathers journal identification information (journal title and ISSN together with the publishing information contained in its <imprint> sub-element), for instance:

```
<monogr>
   <title level="j" type="main">European Journal of Human Genetics</title>
   <title level="j" type="nlm-ta">Eur J Hum Genet</title>
   <idno type="ISSN">1018-4813</idno>
   <imprint>…</imprint>
</monogr>
```

### 4.3 The <imprint> element

“By imprint is meant all the information relating to the publication of a work: the person or organization by whose authority and in whose name a bibliographic entity such as a book is made public or distributed (whether a commercial publisher or some other organization), the place of publication, and a date. It may also include a full address for the publisher or organization. Full bibliographic references usually specify either the number of pages in a print publication (or equivalent information for non-print materials), or the specific location of the material being cited within its containing publication.”[1]

The <imprint> element is organised as follows:

```
<imprint>
   <pubPlace>Oxford</pubPlace>
   <publisher>Clarendon Press</publisher>
   <date typ=”published” when="1969-02-07"/>
   <biblScope type="vol">3</biblScope>
   <biblScope type="issue">2</biblScope>
</imprint>
```

The possible values for the attribute type on <biblScope> are the following:

- vol: volume
- issue: issue
- fpage: first page
- lpage: last page
- pp: number of pages when the information about full pagination is not available[2]

## 5 <biblStruct> skeleton

The following example provides an overview of the full internal structure of the <biblStruct> element as provided by the PEER Depot within a <TEI> document. Most mandatory PEER metadata fields are illustrated here.

```
<biblStruct type="article">
    <analytic>
        <title level="a" type="main">…</title>
        <author type="corresp">
            <persName>
                <forename>…</forename>
                <surname>…</surname>
            </persName>
            <affiliation>
                <orgName type="">…</orgName>
                <address>…<country>FR</country></address>
            </affiliation>
            <email>…</email>
        </author>
```

1 http://www.stoa.org/projects/epidoc/stable/guidelines/

2 We restrict here the semantic of the recommended value (cf. http://www.tei-c.org/release/doc/tei-p5-doc/html/ref-biblScope.html).

```
    </analytic>
    <monogr>
        <title level="j" type="main">…</title>
        <idno type="ISSN">…</idno>
        <imprint>
            <publisher>…</publisher>
            <pubPlace>…</pubPlace>
            <date when="2009-02-03"/>
            <biblScope type="fpage">…</biblScope>
        </imprint>
    </monogr>
    <idno type="DOI">…</idno>
</biblStruct>
```

## 6 Mapping table

The following table makes explicit the PEER mandatory metadata fields, as found in the TEI based exchange format transferred to PEER repositories.

| Field Name | Path in TEI document | Notes |
|---|---|---|
| Article title | /TEI/teiHeader/fileDesc/sourceDesc/biblStruct/analytic/title[@type='main'] | When applicable additional titles are provided (with specific values of @type). |
| Corresponding author | /TEI/teiHeader/fileDesc/sourceDesc/biblStruct/analytic/author[@type='corresp'] | Additional authors are provided as siblings of this element in further <author> elements. |
| Author name | ./persName | The following elements are used for describing author's name: <forename>, <surname>, <roleName>, <nameLink>, <genName> |
| Author email | /TEI/teiHeader/fileDesc/sourceDesc/biblStruct/analytic/author/email | |
| Abstract | /TEI/text/front/div[@type='abstract'] | Further elements may be found in the abstract, most notably:<br><head> for abstract title<br><p> for paragraphs<br><hi> for additional rendering (e.g. <hi rend="italic">) |
| Publication date | /TEI/teiHeader/fileDesc/sourceDesc/biblStruct/monogr/imprint/date/@when | Expressed in conformance to ISO 8601:2004 (i.e. yyyy-MM-dd) |
| DOI of published article | /TEI/teiHeader/fileDesc/sourceDesc/biblStruct/idno[@type='DOI'] | When applicable, further identifiers maybe provided with additional <idno> elements. |
| Country of contributing authors | /TEI/teiHeader/fileDesc/sourceDesc/biblStruct/analytic/author/affiliation/address/country | Expressed in conformance to ISO 3166-1-A2 (e.g. FR). |
| Journal title | /TEI/teiHeader/fileDesc/sourceDesc/biblStruct/monogr/title[@type='main'] | Additional titles (e.g. abbreviated) may appear with publisher specific @type values. |
| Affiliation | /TEI/teiHeader/fileDesc/sourceDesc/biblStruct/analytic/author/affiliation | Main components here are expressed in <orgName> and <address> elements |
| ISSN | /TEI/teiHeader/fileDesc/sourceDesc/biblStruct/monogr/idno[@type='ISSN'] | @type value may be ISSN (generic) pISSN (printed version) or eISSN (electronic version) |
| Volume | /TEI/teiHeader/fileDesc/sourceDesc/biblStruct/monogr/imprint/biblScope[@type=vol] | |

| | | |
|---|---|---|
| Issue | /TEI/teiHeader/fileDesc/sourceDesc/biblStruct/monogr/imprint/biblScope[@type=issue] | |
| First page | /TEI/teiHeader/fileDesc/sourceDesc/biblStruct/monogr/imprint/biblScope[@type='fpage'] | |
| Last page | /TEI/teiHeader/fileDesc/sourceDesc/biblStruct/monogr/imprint/biblScope[@type='lpage'] | |
| Type | /TEI/teiHeader/fileDesc/sourceDesc/biblStruct/@type | Possible values are: **article**, inproceeding, inbook, book, thesis, report |
| Subject headings | /TEI/teiHeader/profileDesc/textClass/keywords | When available, provided as a <list> of <item> for each keyword<br>(defaults to what is provided in the Journal table)[1] |
| Language | /TEI/teiHeader/profileDesc/langUsage/language/@ident | ISO 639-1 (defaults to 'en') |
| Embargo | /TEI/teiHeader/fileDesc/publicationStmt/availability | Note that the information is rarely provided.<br>(defaults to what is provided in the Journal table)[2] |

*Table 3: Mapping of TEI*

## References

http://www.inera.com/EML2002Rosenblum01.pdf

1 See 11.3 in the PEER Description of Work.
2 See 11.3 in the PEER Description of Work.

# Appendix B. SWORD Protocol

## 1. Introduction

In the PEER project, selected stage-2 material from publishers is being transferred to or deposited into the PEER Depot after which the content is being transferred from the depot to multiple, publicly available repositories.

The stage-2 material will be transferred in a Submission Information Package (SIP) containing the full-text publication, metadata and the complementary stage-2 source files. The SWORD AtomPub profile contains specific features that allows for an application-level deposit of material into repositories.

The PEER information model can be mapped onto the OAIS Reference Model and the DRIVER object model for Enhanced Publications.

Implementers may set up their own server conforming to these Guidelines using one of repository specific implementations available from SourceForge, or write their own custom implementation either using the generic Java library, also available from SourceForge, begin their implementation from scratch.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

It is assumed that the reader of this document has knowledge of the PEER D2.1 report[1], SWORD profile v1.3[2] , the OAIS[3] Reference Model[4] and the DRIVER[5] II Enhanced Publication object model and Functionalities[6].

### 1.1 SWORD overview

The SWORD AtomPub Profile is an application profile of the Atom Publishing Protocol (APP) (RFC 5023)[7] that contains specific features that allow for an application-level deposit of material into repositories.

The APP is based on the HTTP transfer of Atom-formatted representations. It is easy to think of APP as a way of publishing just Atom Syndication Format feeds. While it is true that APP provides the means to publish Atom Syndication Format Entries to collections (such as blogs), it also provides a mechanism for the publishing of binary formatted data called Media Resources in APP context (Internet Engineering Task Force 2007). While in the blog scenario this mechanism may be used to add attachments to a blog post (i.e. images, audio, video, documents), SWORD exploits this for the publishing (or deposit) of material into repositories, usually in some form of content packaging in which data and descriptive metadata are being held together in one container (see Figure 2).

---

1 PEER D2.1 Draft report on logfile harvesting systems and manuscript deposit procedures for publishers and repository managers, http://www.peerproject.eu/reports/.

2 ALLINSON, J et al (2008) *SWORD AtomPub Profile version 1.3*, viewed 25 March 2009. http://www.swordapp.org/docs/sword-profile-1.3.html.

3 Open Archival Information System.

4 Consultative Committee for Space Data Systems 2002, *OAIS Reference Model*, http://public.ccsds.org/publications/archive/650x0b1.pdf.

5 Digital Repository Infrastructure Vision for the European Region.

6 VERHAAR, P & PLACE, T (2008) *Report on Object Models and Functionalities*, DRIVER II D4.2.

7 Internet Engineering Task Force 2007, *The Atom Publication Protocol*, RFC 5023, Internet Engineering Task Force, http://tools.ietf.org/html/rfc5023.

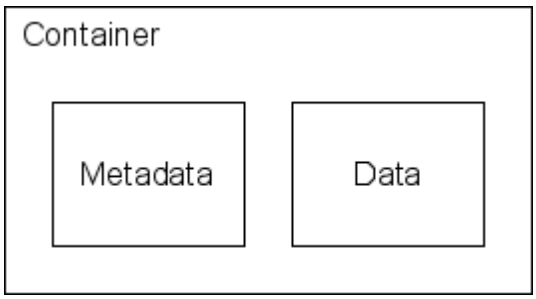


*Figure 2: Content Package or Container*

An example of an implementation of such a container would be a zip file containing a full-text manuscript in the PDF/A-1 format and descriptive metadata in the NLM-XML format.

The container is being submitted by a client to a SWORD interface service (server) as a bit stream using a HTTP POST request consisting of a header containing information about authorisation and the bit stream (type and format of the container) in order for the server to be able to interpret the bit stream properly, and a body part containing the bit stream itself (see Figure 3). Upon reception, the server sends a HTTP response back to the client – again consisting of a header and a body part – with the header containing a HTTP status code indicating a success or failure of the attempted deposit according to regular HTTP semantics, and a response document containing additional APP/SWORD specific information about the deposit being made.

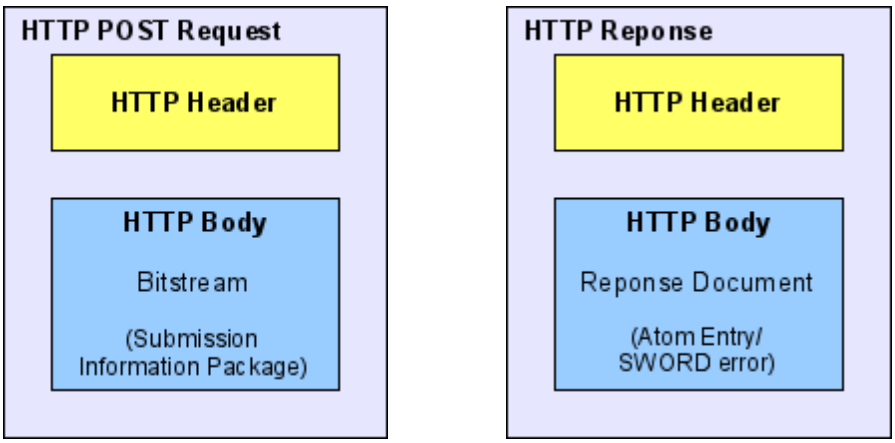


*Figure 3: HTTP request and response structure in the SWORD context*

### 1.2 Use of SWORD in PEER

In the PEER workflow there are two scenarios of deposits into the PEER repositories specified: deposit made by PEER and deposit made by authors (see Figure 4)

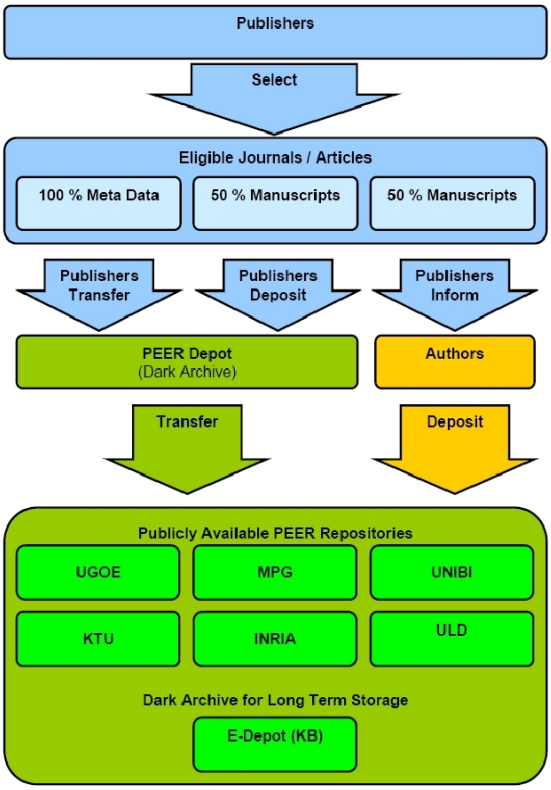


*Figure 4: PEER Workflow*

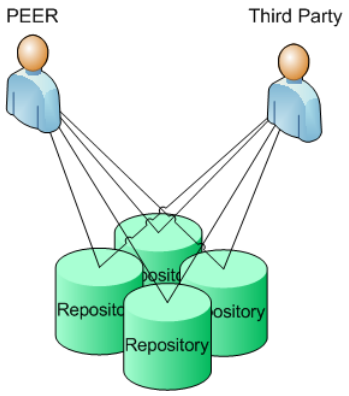


*Figure 5: Deposit situation*

This results in an n:n-relation between repositories and deposit sources either the PEER Depot or third party services operated by an author (see Figure 5). To prevent multiple tailored solutions and implementations it is important to define a standard process for the deposit of material into repositories.

The processes may be categorised into two types of mechanisms: **push and pull**. An example of the **pull** mechanism is the KB's mechanism of the eDepot harvesting repositories through OAI-PMH and pulling content using a webclient (see Figure 6) which downloads the objects specified in the location entries in the metadata.

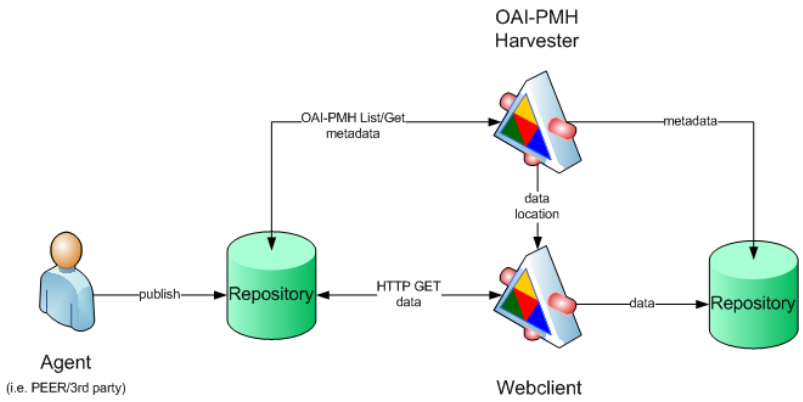


*Figure 6: OAI-PMH data harvest*

An example of the **push** mechanism is the SWORD deposit mechanism where the data is being pushed by an agent (i.e. a webservice or desktop application representing a user) to the SWORD interface of a repository which then accepts or rejects the deposit (see Figure 7).

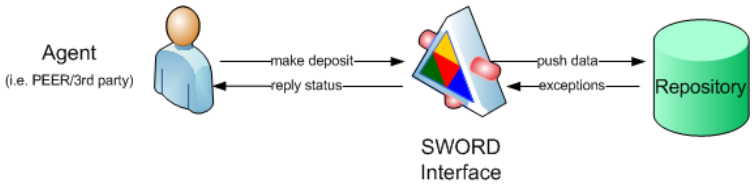


*Figure 7: SWORD data deposit*

Finally, a third, hybrid mechanism can be created by setting up an FTP server to which deposits can be uploaded (pushed) by an agent. A repository may then pull the FTP content which is then being pulled into the repository (see Figure 8).

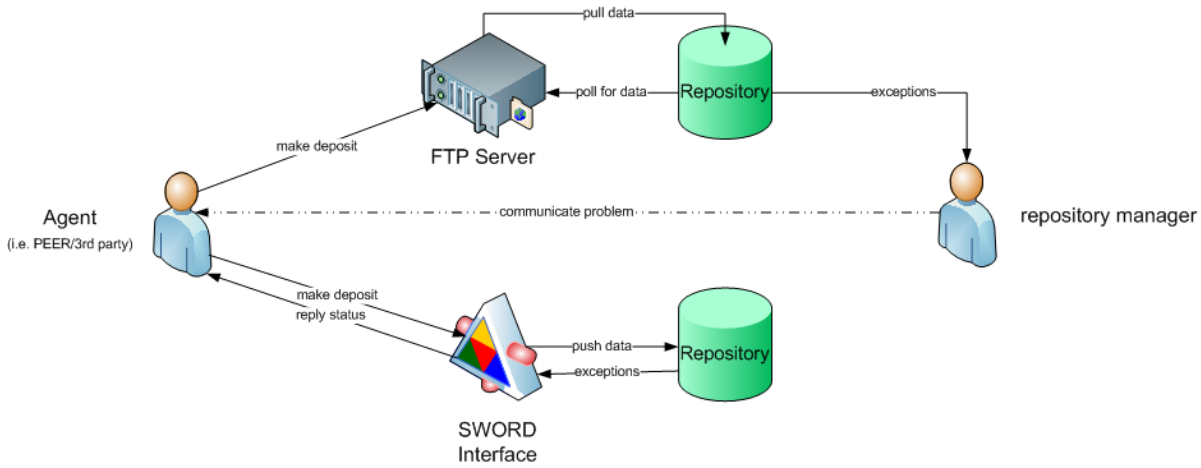


*Figure 8: SWORD vs FTP*

A disadvantage of this mechanism is that this only provides direct feedback to the agent about status of the upload, not of the status of the actual deposit into the repository. This may lead to the situation when an agent successfully uploads data to the FTP server, but the data is being rejected by the repository afterwards because it does not adhere to rules the repository enforces on its contents without the agent being informed about this rejection – something that is not the case when using SWORD.

Figure 9 provides a schematic overview of the use of SWORD in the PEER deposit scenario. Here a publisher transfers manuscripts and metadata into the PEER Depot where the manuscripts and metadata are being converted and crosswalked to the formats specified for the PEER deposit process. The converted and crosswalked manuscripts and metadata are then being packaged into a container and sent to the SWORD interface service of a repository where the contents are being unpacked from the container. Upon reception these MAY be converted and crosswalked into an internal storage format before they are being archived into the repository.

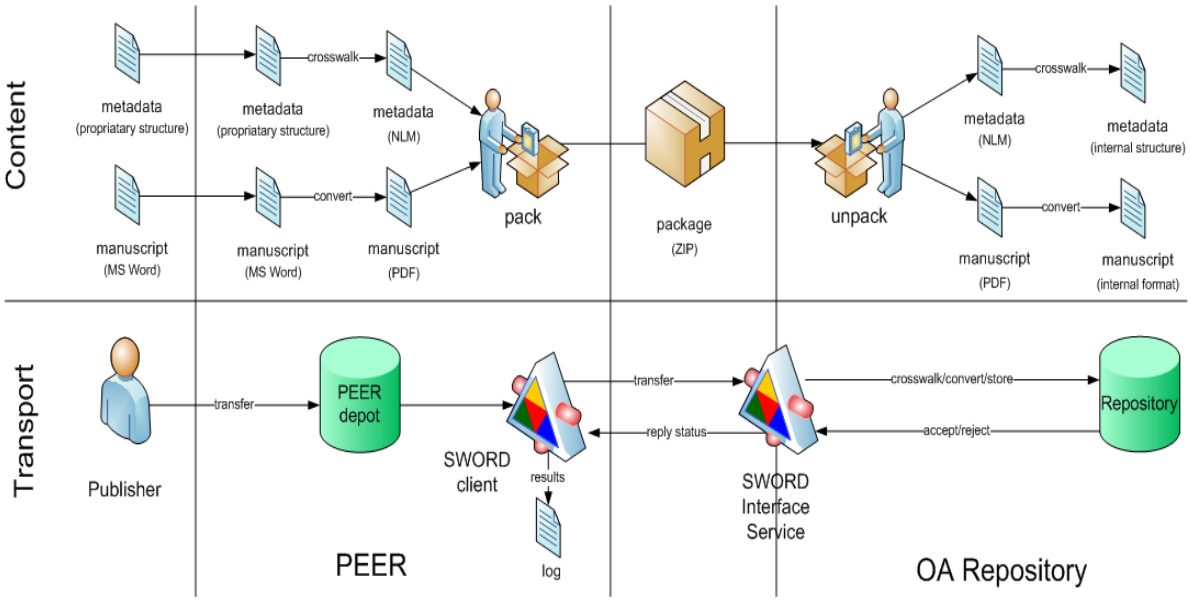


*Figure 9: SWORD use in PEER for PEER Depot*

# 2 Use of SWORD features

## 2.1 About this section

This section will describe the use of the SWORD profile in the context of the PEER project. The contents are organised according and supplementary to the document SWORD Atom Pub Profile version 1.3 part A. If a SWORD profile section or feature is omitted, implementations MUST behave as defined in SWORD profile.

## 2.2 Package Support

The PEER Submission Information Package (SIP) MAY be expressed using (a combination of) different formats (i.e. XML containers or RFC 1951 compliant zip archives) and/or serialised using different structural models (i.e. DIDL, METS, ORE, TEI, NLM, MODS, DC). The mappings between the SIP, its components and the formats and structures will be defined and expressed using specialised application profiles developed in the PEER context.

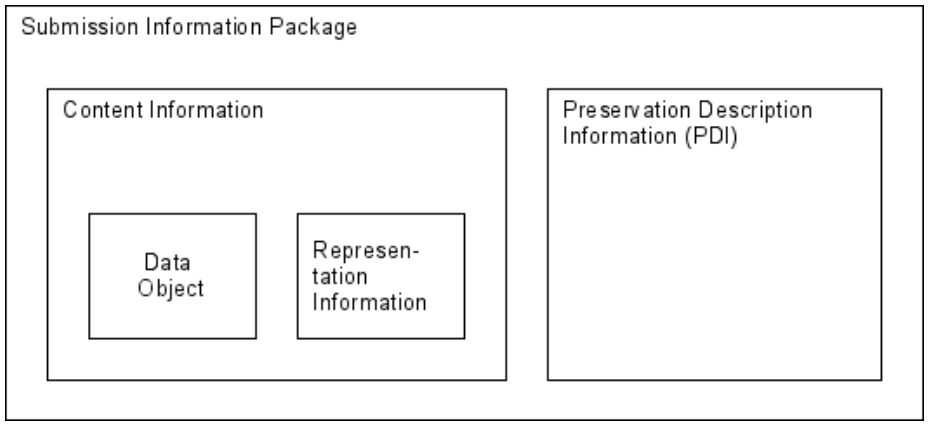


*Figure 10: Submission Information Package structure*

The SWORD profile offers the possibility to enumerate multiple packaging formats in the Service Document and supply a Quality Value attribute indicating a preference and level of support for a designated package format.

**Package support in Service Description**

The server MAY support multiple packaging formats with varying quality values according to the support of the PEER Submission Information Package (SIP).

The server MUST support at least one package format with Quality Value “1.0”, indicating full support where all components supplied within the SIP will be processed and understood when using the designated package format.

All supported formats MUST be listed in the Service Document.

All formats listed in the Service Document MUST have a Quality Value attribute assigned.

The value used in the <sword:accepted Packaging> element MUST NOT overload any values enumerated in the SWORD Content Package Types.

The server MAY use the <sword:service> element in the Service Document to indicate the existence of other service interfaces supporting additional package formats.

The server SHOULD NOT accept a specific package format across multiple interfaces with different levels of support as indicated by the Quality Value attribute in the Service Document.

**Package Support during Resource Creation**

If a server receives a POST request with a format that is not listed as an accepted format in the Service Document, the server MUST reject the package by returning an HTTP status code of 415 (unsupported media type).

**Package description in entry documents**

When describing packaged resources in Media Entry documents, the server SHOULD add sword:packaging elements to the entry.

### 2.3 Mediated Deposit

The following paragraph is considered informative, but is included for clarity in the use of the SWORD profile outside the PEER project.

The PEER workflow offers two ways a manuscript can be deposited into one of the publicly available PEER repositories: either by publisher deposit (through the PEER Depot) or by author deposit (where the publisher informs the author who deposits his/her article(s) in the actual publicly available repository[1]).

For the author deposit, the author MAY make the deposit by proxy through a web service (i.e. by filling in a form to provide the metadata and upload a file containing the full-text material) after which the web service is making the actual deposit. The web service MAY not be used for the PEER project exclusively in which case the web service MAY use its own credentials to authenticate at the server (at the repository side).

Figure 11 depicts an example of the use of this mechanism in the PEER context. Note that the greyed out parts of the figure are considered outside the scope of the PEER project.

1 PEER D2.1 Draft report on logfile harvesting systems and manuscript deposit procedures for publishers and repository managers, p.8, http://www.peerproject.eu/reports/.

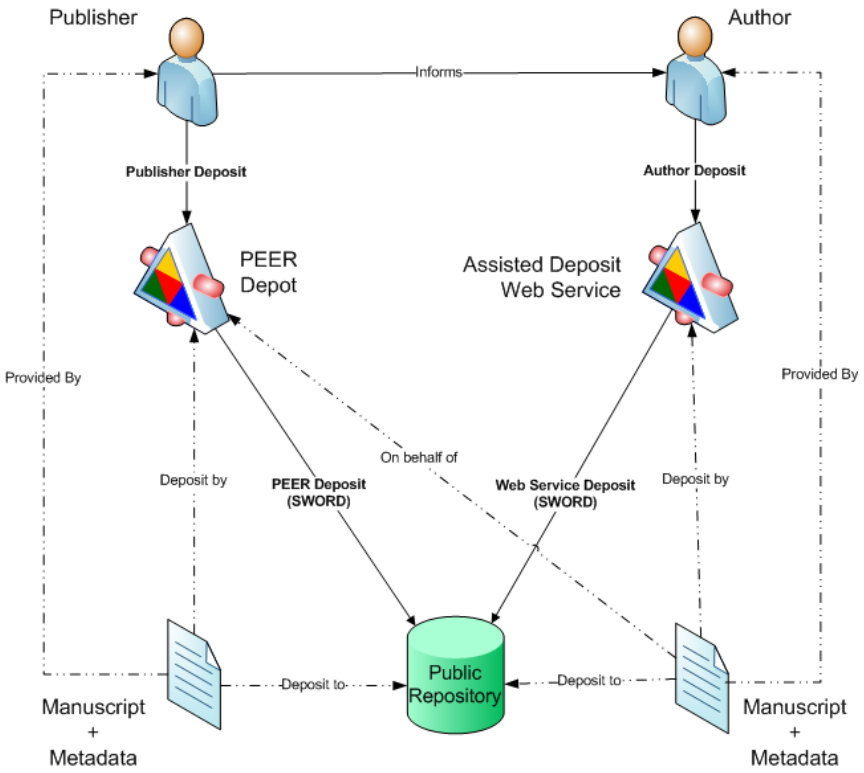


*Figure 11: PEER deposit workflow*

It is recognized that the repository MAY want to keep track of data that is being deposited within the PEER context by creating a single user account to the PEER Depot. This then covers the publisher deposit workflow, but does not provide for a solution for the case of author deposit through another web service which MAY use different credentials.

A possible solution MAY be the use of mediated deposit where a client authenticates using its assigned credentials on behalf of another known user (e.g. a web service authenticates using its own credentials and makes the deposit on behalf of the PEER user which is used by the PEER Depot).

This method MAY also be used to authenticate on behalf of other users (i.e. authors, librarians, data stewards, research assistants, etc.) that already have a valid user account at the repository.

The use of mediated deposit is considered OPTIONAL and is currently not implemented in the application of the SWORD profile within the PEER project.

### Mediation in Service Description

Servers supporting mediated deposit MUST indicate this by including a SWORD:mediation element with a value of “true” in the Service Document as defined in the SWORD profile version 1.3 section 2.1.

For servers that do not include a SWORD mediation element in the Service Document, a default value of “no” SHOULD be assumed by clients.

### 2.4 Auto-discovery

AtomPub makes no recommendations on the discovery of Service Documents.

The SWORD profile states that it is RECOMMENDED that server implementations use an <html:link rel="sword" href="[Service Document URL]"/> element in the head of a relevant HTML document to assist with service discovery.

In addition, it is RECOMMENDED to also include an <atom:link rel="sword" type="application/atomsvc+xml" href="[Service Document URL]"/> element in relevant response documents such as Error Documents.

### 2.5 Nested Service Descriptions

Nested Service Descriptions MAY be used to specify alternative collections for both organisational (i.e. generic collection with a nested PEER specific collection) and technical purposes (i.e. a specific interface or service instance to cater for specific types of content packaging).

## 3. Use of APP features

The contents of the following section are organised according and supplementary to the document SWORD Atom Pub Profile version 1.3 part B. If a SWORD profile section or feature is omitted, implementations MUST behave as defined in the SWORD profile.

### 3.1 Securing the Atom Publishing Protocol

The SWORD profile states servers SHOULD support the use of HTTP Basic Authentication over TLS. Because from a trust perspective it is important to confirm the identity of the PEER Depot during the deposit process, this statement is considered insufficient for the purposes of the PEER project.Therefore this requirement has been restated as follows:

Servers implementing SWORD MUST support HTTP Basic Authentication (RFC 2617) over TLS (RFC 2818).

### 3.2 Creating and Editing Resources

When depositing resources using SWORD, resources are created by a server when a client makes an HTTP POST request with the resource in the HTTP request body. If the deposit is made successfully, the server then gives a HTTP response with the HTTP 201 Status code in the header of the response indicating the resource has been successfully created at the repository side.

Servers returning a HTTP 201 status code after a deposit MUST preserve the resource deposited.

Clients receiving a HTTP 201 status code MUST consider the resource deposited as being accepted for storage by the repository.

#### 3.2.1 Asynchronous treatment of resources

It MAY however be the case that the repository implements an additional asynchronous validation process after which a resource MAY or MAY NOT be accepted. This for instance is the case when a repository uses an intermediate repository where resources deposited through the SWORD interface are temporarily stored, after which they will be moved to a final location within the repository when they are properly validated by a repository manager. When a resource is then being rejected by the repository during the validation process after the server has sent an HTTP 201 response to the client, the situation MAY arise where the client considers the resource as being successfully deposited into the repository, while in fact the resource is NOT being stored into the repository. This situation is viewed as undesirable.

Servers implementing an asynchronous validation process MUST return an HTTP 202 Accept response code indicating the request has been accepted for processing, but the processing has not been completed.

Clients receiving a HTTP 202 status code upon deposit of a resource MUST consider the resource deposited as NOT being stored into the repository.

RFC2616 states that there is no facility for the re-sending of status codes. Therefore, a client will not receive a notification of the outcome of the processing carried out by the server. In order to allow clients to retrieve the outcome of the deposit, the sword:treatment element MAY contain the status of the processing of the deposited resource.

Servers implementing HTTP 202 status codes MUST supply a permanent link to the Atom Entry document of the response.

Servers implementing HTTP 202 status codes MUST update the sword:treatment element of the Atom Entry document of the resource with the status of the processing of the deposited resource.

Client SHOULD implement a mechanism to confirm the successful deposit by periodically checking back at the server with an HTTP GET request to the permanent link supplied by the server, in order to check the contents of the sword:treatment element of the Atom Entry describing the deposited resource when a HTTP 202 status code has been received upon deposit.

## 4. PEER Object Model

In order to future proof agreements and guidelines for a technical model, it is important to detach the technical implementation from the abstract object and information model. Furthermore it is important to keep this abstract model aligned with other developments in the area the model will be used in. For PEER, there are two of such developments:

- OAIS Reference Model for its use by the KB
- DRIVER object model for Enhanced Publications for its use in DRIVER context

In PEER, manuscripts and metadata will be transferred between authors, publishers, the PEER Depot, Open Access repositories and an LTP repository exploited by the KB.

This results in a PEER object consisting of a manuscript object which is being described by one or more metadata objects (see Figure 12).

The D2.1 report provides an exhausting metadata field set to be used in the PEER project (see Table 3, below).

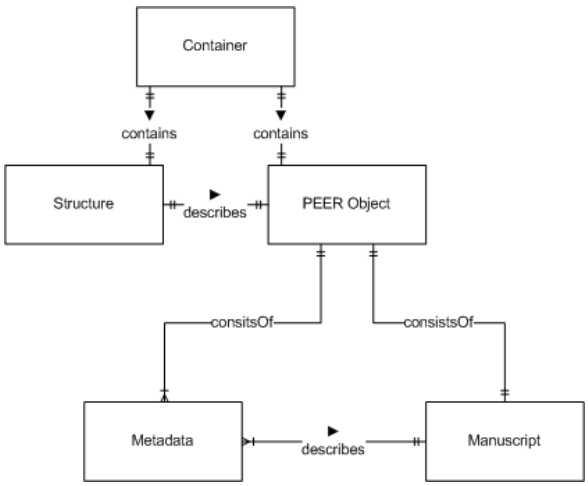


*Figure 12: PEER Object model ERD*

| Field Name | Semantics | Syntax |
|---|---|---|
| **Title** | Article Title | |
| **Creator** | Corresponding author's name | Last name, first name |
| AuthorEmail | Corresponding author's e-mail address | |
| Description | Abstract | |
| **Date** | Date of publication | ISO 8601:2004 ; yyyy-mm-dd |
| **Identifier** | DOI of published article | |
| Coverage | Geographic location of the contributing Author | ISO 3166-1-A2 |
| Journal | Journal title | |
| Affiliation | multi-tier organisation list | Country, Organization, Laboratory |
| ISSN | | |
| Volume | | |
| Issue | | |
| Page | | |
| **Type** | Semantic type of the publication | info:eu-repo/semantics/article info:eu-repo/semantics/acceptedVersion defaults to article. |
| Subject | Subject headings; Scientific classification (defaults to what is provided in the general STM Journal table[1]) | |
| Language | Language of the publication | ISO 639-1 (defaults to 'eng') |
| Embargo | Embargo Period (defaults to what is provided in the general STM Journal table[5]) | |

*Table 4: PEER information model*

For deposit the PEER object will be packaged into a container. The OAIS reference model specifies the Submission Information Package (SIP) as a specialised Information Package (IP) – which is used by the KB in the eDepot – for submission purposes (see Figure 13).

1 See Appendix A.

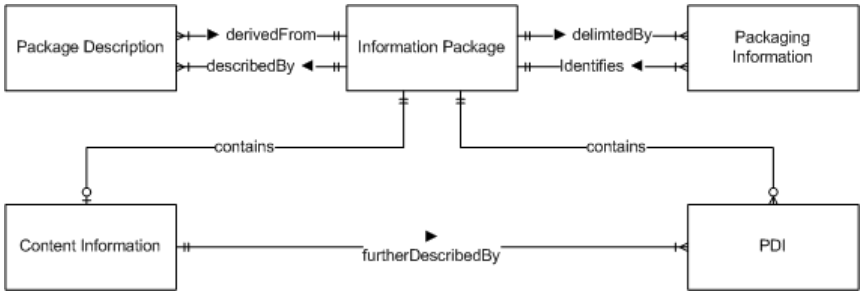


*Figure 13: OAIS Information Package ERD*

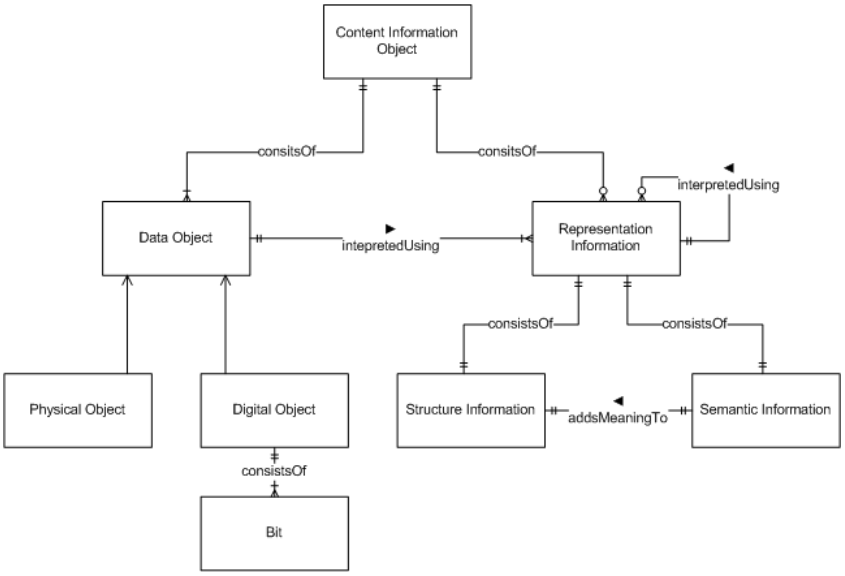


*Figure 14: OAIS Content Information Object ERD*

An IP consists of content information that is being described by Package Description Information (PDI).

The content information object is defined as a data object (i.e. PDF file) interpreted using representation information (i.e. mime-type, encoding version, etc.) (see Figure 14). Note the structure information being a part of the representation information.

The PDI (see Figure 15) contains Reference Information (i.e. bibliographic descriptions and persistent identifiers), Provenance Information (i.e. information about the conversion process), Context Information (i.e. reference to the research project a publication is based on) and Fixity Information (i.e. a checksum).

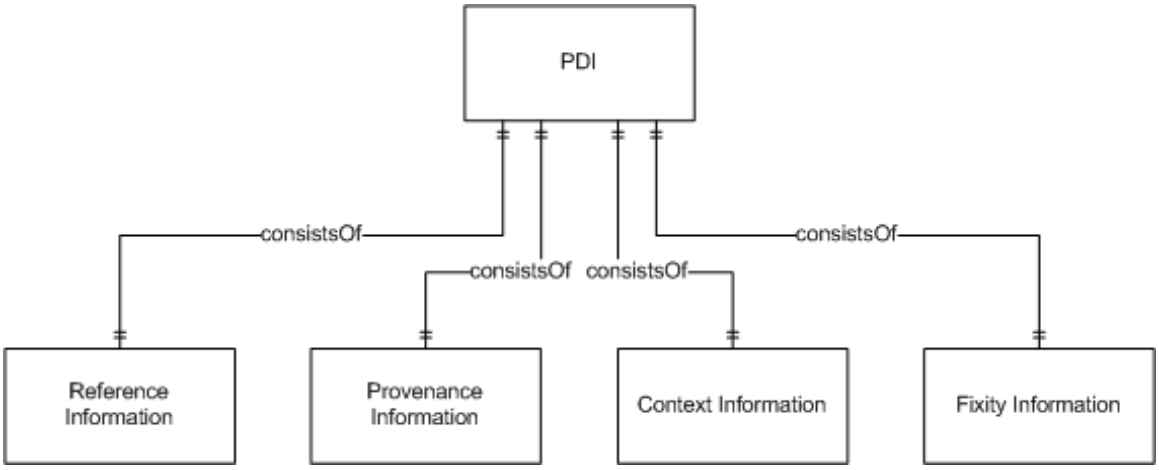


*Figure 15: OAIS Package Description Information ERD*

The PEER information model can be mapped onto the OAIS Reference Model as depicted in Figure 16. Here the structure object containing the structural information is being added to the PEER object model.

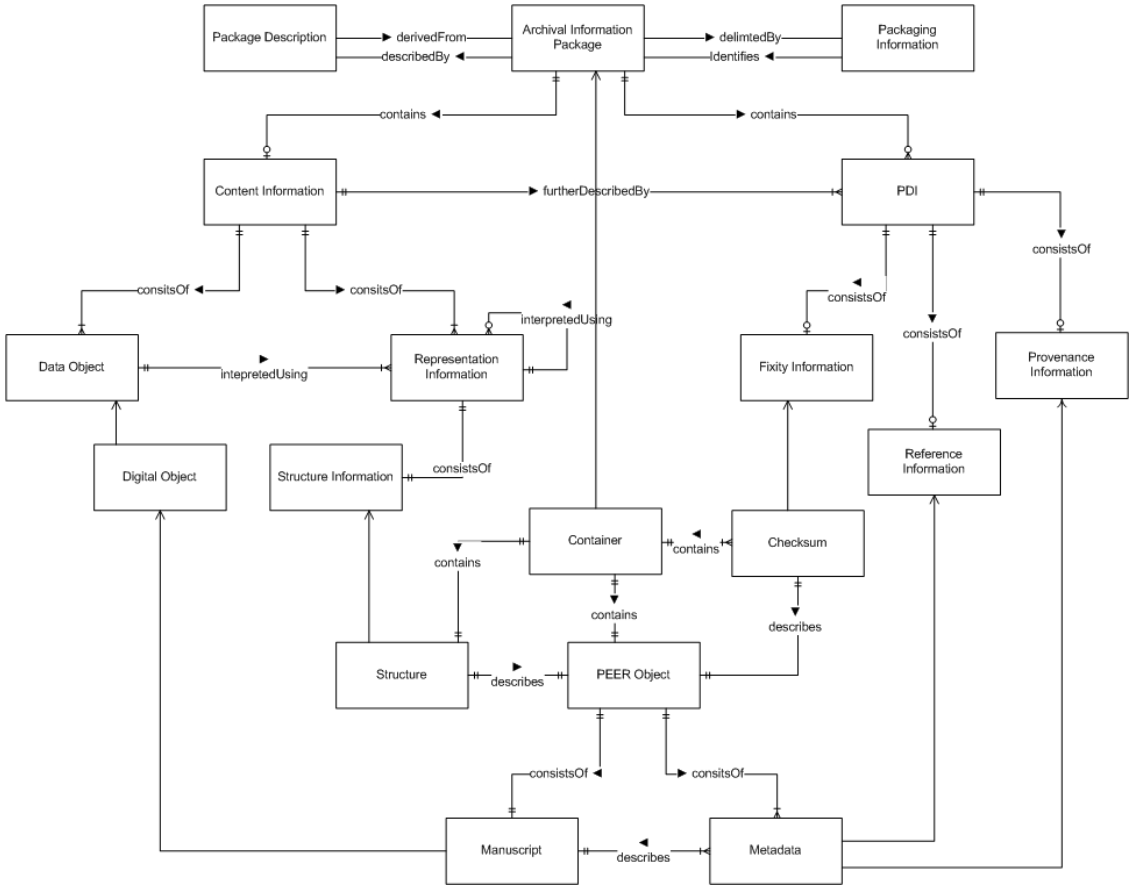


*Figure 16: OAIS Reference Model-PEER Information Mapping*

Figure 17 depicts a technical mapping of the PEER object model. The structure is expressed using the ORE abstract data model which is serialised as an Atom feed. This

Atom feed is to be contained in an XML file. The metadata is serialised in TEI, again contained in an XML file. The manuscript is encoding in PDF/A, contained in a PDF file. The XML file containing the ORE Atom feed, the XML file containing the TEI document and the PDF file containing the manuscript are then being packaged in an RFC 1951 compliant zip file. Upon deposit using SWORD, the zip file is being placed into the body of the HTTP POST request. The Header contains an MD5 checksum and MAY contain authorisation information (see Figure 18).

The HTTP POST request is then being sent to the SWORD Interface Service as described in paragraph 1.2.

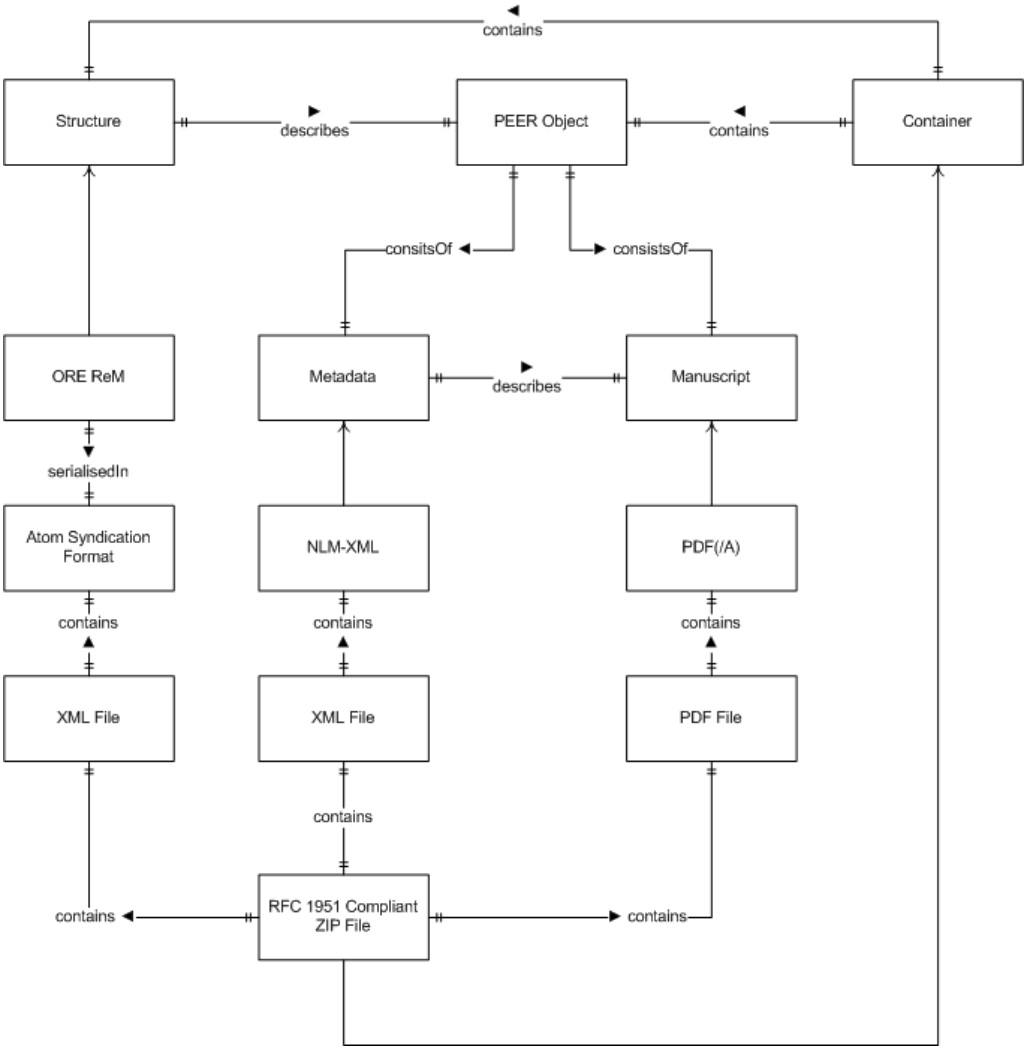


*Figure 17: Technical Mapping of the PEER model*

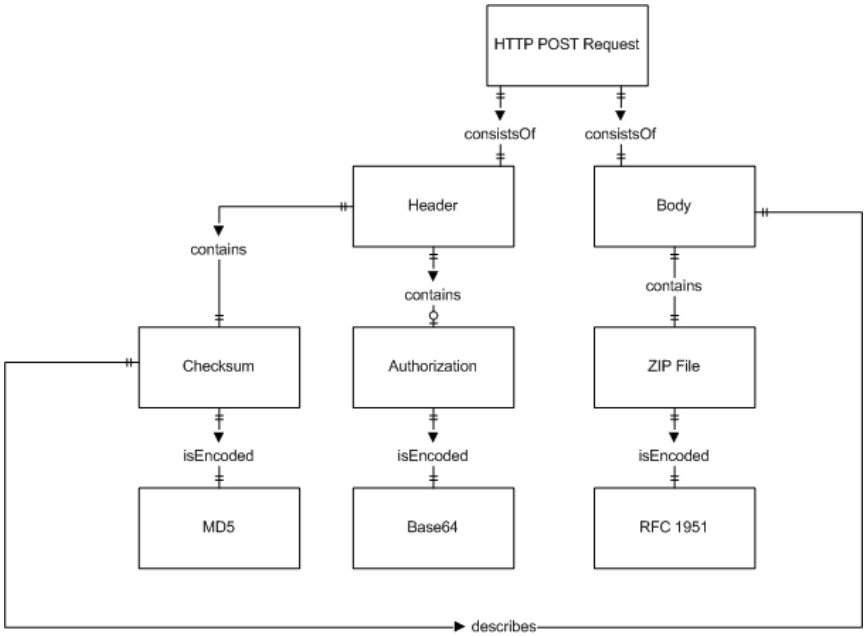


*Figure 18: HTTP Mapping of the Technical Model*

# 5 Implications on Repository level

## 5.1 Overview of the technical process

Generally speaking, the technical process of deposit can be broken in sequential order to the serialisation and deposit request sub-processes on the client side (i.e. the PEER Depot) and the de-serialisation, response and store sub-processes on the repository side (i.e. publicly available repository).

### Serialisation – Client side

The serialisation processes involve the serialisation of the metadata from an internal storage to a specific (agreed upon standard) metadata field set and structure (i.e. DC) and the packaging of the metadata and object file(s) into a content package (i.e. MPEG-21 DIDL XML containers or RFC 1951 compliant zip archives) which MAY include adding a manifest describing contents and their correlation (i.e. the relation between an XML file containing the descriptive metadata of a full-text publication and a PDF file containing the actual full-text publication) to a bit stream.

### Deposit Request – Client side

The deposit request process includes a client posting the data to a service (i.e. the HTTP POST request in SWORD) and the server receiving the data and placing it into a temporary storage (either in memory or on disk).

### De-serialisation – Repository side

In the de-serialisation process the receiving server tries to interpret (decode) the bit stream again, in essence validating the contents. This MAY include the unpacking (when using zip archives) or decoding (when using XML containers) of the bit stream to be able to interpret the individual contents. It MAY also include the mapping or crosswalking of the metadata structure to an internal (proprietary) metadata field set and/or structure. This process MAY not necessarily be taking place in the actual interface service; it MAY include the sending of

the bit stream to an internal storage service which then indicates a success or failure to the deposit service.

**Response – Repository side**

After the contents are being de-serialised successfully and the server confirms the contents of the received bitstream, the server MUST reply its status to the client (when using SWORD this means reporting the appropriate HTTP status code and correct Atom/SWORD response document). If the de-serialisation process has failed for whatever reason, the server SHOULD reject the deposit request to indicate an unsuccessful deposit to the client, or accept the deposit with an HTTP 4xx status code and appropriate exception message indicating a partial successful deposit.

**Store – Repository side**

The final step of the deposit process includes the storage of the received (meta)data into the internal (meta)data store. This part of the process is implementation specific and considered outside the scope of this document.

### 5.2 Functional Requirements

A repository implementing a SWORD interface service in the PEER context MUST be able to:

- authenticate a user
- receive, process and respond to an HTTP POST request as specified in this document
- interpret and store a PEER Submission Information Package as specified by the PEER project

### 5.3 Implementation Steps

The implementation steps can be broken down into the implementation and exposure of the web service to the outside world and interface with the repository on the inside.

Depending on specific needs, an implementer of the SWORD profile may either choose to make an implementation by using one of the repository specific implementations available for DSpace, ePrints and Fedora on Sourceforge[1] or to write a custom SWORD server implementation (optionally by using the generic Java library also available from Sourceforge).

For the repository specific option please refer to the documentation provided with the designated packages.

The second option either involves writing a service from scratch or use the source code available from the SWORD Java library. This library contains ready to implement code for writing servers and clients.

In addition to creating the web service which behaves according to the guidelines specified in this document, special attention should be paid to the creation of crosswalk rules to map the expression(s) of the PEER SIP to the internal repository data structure and semantics.

1 SWORD Project, *SourceForge.net: SWORD − Project Web Hosting − Open Source Software*, viewed on 25 March 2009, http://sword-app.sourceforge.net/.

# Appendix C. Author communication texts

The PEER project aims to monitor the effects of systematic archiving over time, with the intent to limit interference with established practise, in support of the behavioural research methodology envisaged in WP4. Both publishers and repository managers are therefore provided with generic texts to communicate sufficient and consistent information to authors, both at established points of contact with publishers, and in an online Helpdesk established in this work package.

## 1 Publisher websites and author communications

### 1.1 Publisher acknowledgement of submission

#### 1.1.1 Publisher deposit

"The journal which you are submitting to/have submitted to [as appropriate] is participating in the PEER project. This project, which is supported by the European Union EC eContent*plus* programme <link to http://ec.europa.eu/information_society/activities/econtentplus/index_en.htm> aims to monitor the effects of systematic self-archiving (author deposit in repositories) over time. If your submission is accepted, your accepted manuscript may be archived by [Publisher name] on your behalf, as part of this project. The project will develop models to illustrate how traditional publishing systems may coexist with self-archiving. For further information please visit the PEER project website. <a href> http://www.peerproject.eu/ </a href>"

#### 1.1.2 Author deposit

"The journal which you are submitting to/have submitted to [as appropriate] is participating in the PEER project. This project, which is supported by the European Union EC eContent*plus* programme <link to http://ec.europa.eu/information_society/activities/econtentplus/index_en.htm>, aims to monitor the effects of systematic self-archiving (author deposit in repositories) over time. If your submission is accepted, and you are based in the EU, you may be invited to deposit your accepted manuscript in a repository as part of this project. The project will develop models to illustrate how traditional publishing systems may coexist with self-archiving. For further information please visit the PEER project website.
<a href> http://www.peerproject.eu/ </a href>"

### 1.2 Publisher acceptance

"This journal is participating in the PEER project, which aims to monitor the effects of systematic self-archiving (author deposit in repositories) over time. PEER is supported by the EC eContent*plus* programme <link to http://ec.europa.eu/information_society/activities/econtentplus/index_en.htm>.

As your manuscript has been accepted for publication by [Journal name], you may be eligible to participate in the PEER project. If you are based in the European Union, you are hereby invited to deposit your accepted manuscript in an institutional or subject-based repository of your choice, or in one of the participating PEER repositories. When depositing your manuscript in a repository, please set an embargo period of X months from the date of publication of the journal article for the public release of your accepted manuscript. For further information please visit the PEER project website.
<a href> http://www.peerproject.eu/ helpdesk/</a href>"

## 2 PEER project website

### 2.1 PEER Helpdesk: Guidelines for Authors

“Invited authors, based in the European Union, may self-archive their accepted manuscripts in the PEER project, via deposit in an institutional or subject-based repository of choice, or in one of the participating PEER repositories. The following repositories are available for invited PEER deposits:

- Max-Planck-Gesellschaft zur Förderung der Wissenschaften e.V. (MPG)
- HAL, Institut National de Recherche en Informatique et en Automatique (INRIA)
- Göttingen State and University Library (UGOE)
- BiPrints, Universität Bielefeld (UNIBI)
- Kaunas University of Technology, Lithuania
- University Library of Debrecen, Hungary

Authors who intend to deposit their accepted manuscript in a repository other than one of the PEER repositories above, are kindly requested to notify the project by inserting the URL of the alternative repository of choice: “

Repository URL: [ ]

### 2.2 Recommended Helpdesk FAQ

1. What is self-archiving?
   Self-archiving refers to the practice of scholars depositing copies of their research papers in electronic repositories or ‘open archives’.
   (http://www.sparceurope.org/resources/hot-topics/institutional-repositories)
2. What is a stage-2 manuscript?
   The stage-2 version is the author’s accepted manuscript which includes all the changes made as part of the peer-review process, but is not the final published version.
3. How do I self-archive (deposit) my accepted manuscript for PEER?
   Locate the institutional repository at your library, research organisation, or subject discipline (eg.arXiv for physics). <a href> http:www.arxiv.org </ahref>
   Follow the deposit procedures provided, or view demo.

## 3 Repository websites

### 3.1 Repository retrieval

“This is the author’s accepted manuscript. This version has been peer-reviewed, prior to publication. This article has been published in final, edited form in [journal title, ISSN, URL of journal]. DOI: XXX [if available].
[repository name] is designed to allow users to access the research output of the PEER project. Consult the project website for further details:
<a href> http://www.peerproject.eu/ </a href>

Any correspondence concerning this service should be sent to the Repository Administrator: [e-mail]”