



OPEN SCHOLARLY COMMUNICATION IN THE EUROPEAN RESEARCH AREA FOR SSH - PREPARATION

WP5: Redevelopment of DOAB as central service Requirements and Specifications

DRAFT



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871069

OPEN SCHOLARLY COMMUNICATION IN THE EUROPEAN RESEARCH AREA FOR SSH - PREPARATION

Deliverable 5.1

Requirements and Specifications

Grant Agreement number	: 871069
Project acronym	: OPERAS-P
Project title	: Open Scholarly Communication in the European Research Area for SSH - Preparation
Funding Scheme	: INFRADEV-02-2019-2020
Project's coordinator Organization	: CNRS-OpenEdition
E-mail address	: pierre.mounier@openedition.org
Website	: https://operas.hypotheses.org/2774
WP and tasks contributing	: WP 5 Task 5.1 (Requirements and Specifications)
WP leader	: OAPEN
Dissemination level	: public
Due date	: 29.02.2020
Delivery date	: 27.02.2020

Contents

Abstract	6
Introduction	6
Current requirements and specifications	7
Installation	7
DSpace 6 installation	7
Hosting	8
DSpace hosting	8
Frontend and STRAPI hosting	8
Content Management	8
Website	9
Hosting	9
Configuration	9
Base configuration	9
OAI-PMH endpoints & crosswalks	11
OpenAIRE OAI endpoint	11
Google Scholar exposure	12
Data migration	15
Redirect app OAPEN	17
Customisations	18
Data model	18
Generalities	18
Proposed structure	19
DSpace communities and collections	25
Metadata schema and Input form configuration	26
Hiding communities and collections in browse	27
ORCID authority control for authors	27
Bitstream-level metadata	28
Item relationships	29
Imports & exports	30
OAI-PMH harvest	30
Full exports	31
RIS citations	31

Other features	32
Grantor Lookup (FundRef integration)	32
Grantor widget (CORDIS integration)	32
Annotations & PDF viewer (Hypothes.is integration)	32
IRUS patch installation and configuration	33
Metrics widget (OPERAS integration)	33
Interface theming	33
Modules	34
Metadata Quality Module	34
Frontend and CMS development	34
Future requirements and specifications	35
Customisations	35
Data model	35
Hierarchical Taxonomies	35
Imports & exports	36
Import publisher data (ONIX & TEI)	36
Other features	36
CrossRef DOI generation	36
NLM reader and parser	37
"Certified by DOAB" button (DOAB integration)	38
Google books widget	38
Future requirements and specifications (on hold)	39
Installation	39
DSpace 7 upgrade	39
Customisations	40
Other features	40
NERD - Entity recognition and disambiguation (INRIA integration)	40
Canceled requirements	41
Customisations	41
Other features	41
Static pages	41
CMS feature	41
Annex: CMS front end development	42

Wire frames OAPEN website	42
OAPEN website mockup	48
Annex: list of data fields OAPEN Library	51

DRAFT

Abstract

This document will describe the redevelopment of the Directory of Open Access Books (DOAB) and the requirements of the platform. DOAB being redeveloped on the same model of the Open Access Publishing in European Networks (OAPEN) platform, this redevelopment is being done in two steps. The first step is to migrate the OAPEN platform, and a second one is for the DOAB platform based on the lessons learned from the first migration. Described in this document are the reasons for such order and the specifications of the first migration as well as the adaptations needed for the second migration.

Introduction

The redevelopment of DOAB and OAPEN is being done together, and as such, by the same main sub-contractors, the companies Atmire and Trilobiet. The platforms will both be using the DSpace repository for the presentation and search of their monographs' catalogue, and a CMS solution for the presentation of the 'non catalogue content'. In this sense, it was important to learn from a first project to implement the following one, therefore, OAPEN was decided to be redeveloped as the first platform and will be followed a few months later by DOAB. Thanks to this, both platforms will be based on the same development and will be able to adopt and reuse the new additions to DSpace done by one or the other platform.

Due to the limited possibilities of the DSpace platform as a content management solution, the front end of the OAPEN platform will be developed using a separate CMS: Strapi. The OAPEN frontend will be developed by the company Trilobiet. This company was also responsible for developing the frontend of the current OAPEN Library environment.

DOAB will be migrated in the second phase, thanks to the lessons learned with the OAPEN migration, as well as being able to use all the services currently being developed for the needs of the OAPEN platform. Given the specific properties of the DOAB platform - the peer review certification including a dedicated API - the redevelopment of the DOAB platform will also require the services of SempterTool, the company that has developed and is hosting the current DOAB application.

As such, the specifications mentioned here will apply to OAPEN, but the reuse of most of the development means it will also apply to DOAB in the updated version of this document due in December 2020 (M18 of OPERAS-P project).

The requirements in this document have been done in partnership with the sub-contracting company Atmire, as well as the future implementation of the frontend by the

sub-contracting company Trilobiet. The currently available DSpace version (6.x), unfortunately, does not allow an easily modifiable user frontend experience. The future version (planned for the second half of 2020), will provide this possibility thanks to the rewriting of the User Interface of DSpace. However, in order to have OAPEN and DOAB ready on time, it was decided to not wait for the future version of DSpace and start immediately with the implementation on the currently available version. For the frontend, a headless CMS (STRAPI) will be used for OAPEN as well as a customly built frontend interface.

The document is separated in different main chapters:

- Phase 1: Current implementation for OAPEN.
- Phase 2: Next implementation for OAPEN and full implementation of DOAB.
- Potential phase 3: Contains some requirements that might be implemented later in the project, or later.
- Cancelled requirements: Some of the original requirements that were mostly moved to the implementation of the frontend by another company.

The deliverable also mentions how the hosting will be done for OAPEN and DOAB, that will happen on Virtual Machines (VMs) provided by OPERAS members Huma-Num (CNRS).

Current requirements and specifications

Installation

DSpace 6 installation

Requirement

OAPEN Foundation requests an installation of DSpace with the XML user interface.

Proposed deliverable

Software	Target version
DSpace	6.x
DSpace UI	XMLUI - Mirage 2
Database	PostgreSQL 9
Tomcat	Apache Tomcat 7

Java	Oracle Java JDK 1.7
Operating System	Linux

Atmire will perform the installation of DSpace with the XMLUI interface on up to two server environments, test and production. This installation includes setting up the DSpace build process, and documentation on how to build the source code.

Throughout the collaboration between the OAPEN Foundation and Atmire, Atmire will manage the modified DSpace source code on its git based version control system, git.atmire.com. Accounts will be created for OAPEN Foundation to access the codebase on this environment. A copy of the latest deployed source code will be checked out to a git repository local to the test and production machines.

Atmire will provide an initial configuration of the DSpace installation. Specific configuration requirements such as custom submission procedure configurations are not included in an out of the box installation. Please refer to the following section on Configuration for specific configuration activities.

As a prerequisite for the Atmire's activities, OAPEN Foundation is required to comply with the technical requirements stated in: <http://bit.ly/technical-prerequisite-template>

Hosting

DSpace hosting

The hosting will be provided by Huma-Num (CNRS), a French OPERAS member. They have already provided us the basic infrastructure for the first test server of OAPEN.

They proposed a first server based on the technical prerequisite document see above, and will provision a new server with final resources description once the testing phase of OAPEN is finished. This will become the OAPEN production environment.

Frontend and STRAPI hosting

This part of the hosting will be managed by Trilobiet and done on Digital Ocean.

Content Management

- Strapi (<https://strapi.io/>)
- MongoDB (<https://www.mongodb.com/>).

Strapi is an open source headless CMS. In our configuration Strapi saves content as



JSON documents in MongoDB. MongoDB is a popular JSON (NoSQL) database. Both headless CMS and Mongo database allow for easy data access by other (future) clients and are not tied to a specific implementation (website).

Website

- Java Spring (Boot/Tomcat) - <https://spring.io/>
- nginx (<https://nginx.org/en/>)
- Let's Encrypt (SSL) - <https://letsencrypt.org/>
- Certbot (SSL certificate renewal) - <https://certbot.eff.org/>

The Java web application collects data from Strapi, DSpace, Hypotheses, Twitter APIs, etc. and serves HTML output to the visitors of the website. Website and datastore (Strapi) are separate entities, that could even be installed on different servers for availability/security considerations. nginx serves as a http proxy to the website. All traffic will be encrypted using Let's Encrypt certificates (<https://letsencrypt.org/>) with certbot automatic (and eternal) renewal <https://certbot.eff.org/>.

Some or all parts of the application stack may be deployed in Docker containers.

Hosting

Digital Ocean (<https://www.digitalocean.com/products/droplets/>)

Digital Ocean offers managed cloud services, of which we intend to use at least:

- Linux (Ubuntu) droplet;
- external firewall;
- backup services;
- floating ip-addresses;
- object storage spaces API.

Configuration

Base configuration

Requirement

After the installation of the out of the box platform with standard configuration, OAPEN Foundation requires Atmire to modify specific elements of the DSpace configuration.

Proposed deliverable

Atmire will configure commonly used configuration parameters and predefine some of the contents of the new repository. Any element not explicitly mentioned here will be delivered

with a standard configuration.

- Repository name
 - Atmire will update the interface messages of the system, in order to replace references to “DSpace” with OAPEN Foundation’s preferred repository name (“OAPEN”).
 - Atmire will update the email templates that DSpace uses to send emails to the users. References to “DSpace” will be replaced by OAPEN Foundation’s preferred repository name. Also, the footer of all emails will be updated with a standard footer defined by OAPEN Foundation. Modifying the other contents of these emails, removing or adding additional email templates is not in scope of this configuration activity.
- Community & collection hierarchy
 - DSpace uses a hierarchical structure called communities and collections to structure the items. An example of such a hierarchy can be found at <https://atmire.com/preview/community-list>.
 - Atmire will configure a hierarchy of communities and collection in the repository. It is recommended to keep the hierarchy as simple as possible, because future maintenance and modification of the hierarchy may require direct interventions on the system’s database.
- Item display page
 - After the configuration of the submission forms and the metadata profile, Atmire will configure the simple item display pages to include those metadata fields that OAPEN Foundation desires to display. By default, the Show Full Item Record link will enable visitors to see a more extensive set of metadata, together with the fully qualified field names. If the OAPEN Foundation wishes to hide this link, this can be done as part of the scope.
 - This configuration implies that a single configuration for the item display page is applied for all items. It is possible that certain metadata fields are not present or filled out for particular items. In that case, the entire field, including the label, is automatically hidden from the simple item page.
 - Atmire assumes no configuration or customization to the full item page is required.
- Search & browse
 - Atmire will alter the configuration of search, browse and discovery facets based on OAPEN Foundation’s requirement. Any functional need or requirement that requires customization of the DSpace codebase has not been included in the scope.
- Handle prefix
 - handle.net is a 3rd party provider of persistent URIs, operated by CNRI in the United States. Integrating and using handle.net in DSpace is an optional feature of DSpace. On a functional level, the integration ensures that every new DSpace item page can be retrieved through a hdl.handle.net link,

followed by a repository-specific prefix, and the handle of the item, for example <http://hdl.handle.net/10793/1338>. This allows an institution to change the domain of the repository, while still ensuring that the persistent URLs keep working.

- Activating the integration with handle.net requires configuration of a specific handle service on the DSpace server, which Atmire can carry out. OAPEN Foundation is responsible for obtaining Handle prefix from the central CNRI Handle site (handle.net). The Registration fee and the Annual Service fee (see <https://www.handle.net/payment.html>) are entirely under OAPEN Foundation's responsibility.
- This configuration task provides setup and configuration of your handle service, as well as communication with CNRI to register your server site and with their hdl.handle.net resolution service.

The base configuration made in DSpace 6 during the first phase will be ported as is to DSpace 7 during the second phase, as part of the upgrade work package above.

OAI-PMH endpoints & crosswalks

Requirement

OAPEN exposes content via OAI-PMH. OAPEN currently exposes data in 3 formats: Dublin Core, ESE, and a "raw" format. Only the Dublin Core format needs to be ported in the new version. The current OAI-PMH endpoints do not use sets.

Proposed deliverable

OAI-PMH will be enabled in DSpace, and the data will be exposed in Dublin Core using the OAI-DC crosswalk. Atmire will configure that crosswalk to map the OAI-DC fields with the metadata fields in OAPEN's metadata schema (see section about Data Model below).

By default, DSpace creates separate sets for communities and collections. This feature will be retained, although not relevant for OAPEN and supposed to be unused.

The OAI-PMH setup will be adjusted a bit to only expose books and book chapters (see section about Data Model below).

OpenAIRE OAI endpoint

Requirement

OAPEN must be harvestable by OpenAIRE.

Proposed deliverable

DSpace has a by default a specific OAI-PMH endpoint for OpenAIRE harvesting. Atmire will configure that endpoint to support the OAPEN metadata schema, and will also filter that endpoint on books and book chapters only (see section about Data Model below).

OpenAIRE support for entities will be supported in DSpace 7, but it is assumed that only the publications need to be exposed, not the other entities. The configuration of DSpace 6 will be reproduced.

Google Scholar exposure

Requirement

The OAPEN Library is currently indexed by Google Scholar, and follows the specific specifications as outlined below.

The following mapping applies to books:

Meta tag	Description
<code><meta name="citation_author" content=" ... " /></code>	Name of the author (repeatable)
<code><meta name="citation_editor" content=" ... " /></code>	Name of the editor (repeatable)
<code><meta name="citation_title" content=" ... " /></code>	Title and subtitle
<code><meta name="citation_publisher" content=" ... " /></code>	Publisher name
<code><meta name="citation_book_type" content=" ... " /></code>	Either: "monograph" or "edited_volume". Please note that this metatag is not part of the formal Google specifications.
<code><meta name="citation_publication_date" content=" ... " /></code>	Year of publication

<code><meta name="citation_online_date" content=" ... " /></code>	Date of availability in OAPEN Library
<code><meta name="citation_pages" content=" ... " /></code>	Number of pages
<code><meta name="citation_keywords" content=" ... " /></code>	Keywords. Separate using semicolons.
<code><meta name="citation_language" content=" ... " /></code>	Language (repeatable)
<code><meta name="citation_doi" content=" ... " /></code>	DOI
<code><meta name="citation_isbn" content=" ... " /></code>	ISBN. If the book is an edited volume, do not show this metatag
<code><meta name="citation_fulltext_world_readable" content=" " /></code>	This marks that the content is Open Access. Please note that the content is empty
<code><meta name="citation_pdf_url" content=" ... " /></code>	Link to the PDF. If there is no PDF available, do not show this metatag
<code><meta name="citation_funder" content=" ... " name="citation_funder_id" content=" ... " name="citation_grant_number" content=" ... " /></code>	<p>Funding information •Funder name is required</p> <p>•Funder ID is Crossref ID – see FundRef data (optional)</p> <p>•Grant number (optional)</p>

The following mapping applies to book chapters:

Meta tag	Description
<code><meta name="citation_author" content=" ... " /></code>	Name of the author (repeatable)

<code><meta name="citation_editor" content=" ... " /></code>	Name of the editor (repeatable)
<code><meta name="citation_title" content=" ... " /></code>	Title and subtitle of the chapter
<code><meta name="citation_inbook_title" content=" ... " /></code>	Title and subtitle of the book
<code><meta name="citation_publisher" content=" ... " /></code>	Publisher name
<code><meta name="citation_publication_date" content=" ... " /></code>	Year of publication
<code><meta name="citation_online_date" content=" ... " /></code>	Date of availability in OAPEN Library
<code><meta name="citation_pages" content=" ... " /></code>	Number of pages
<code><meta name="citation_keywords" content=" ... " /></code>	Keywords. Separate using semicolons.
<code><meta name="citation_language" content=" ... " /></code>	Language (repeatable)
<code><meta name="citation_doi" content=" ... " /></code>	DOI
<code><meta name="citation_fulltext_world_readable" content=" ... " /></code>	This marks that the content is Open Access. Please note that the content is empty
<code><meta name="citation_pdf_url" content=" ... " /></code>	Link to the PDF. If there is no PDF available, do not show this metatag

<pre><meta name="citation_funder" content=" ... " name="citation_funder_id" content=" ... " name="citation_grant_number" content=" ... "/></pre>	<p>Funding information •Funder name is required</p> <p>•Funder ID is Crossref ID – see FundRef data (optional)</p> <p>•Grant number (optional)</p>
--	--

There are several things that need attention:

- Author vs. editor. While not in the official Google documentation (see attached), this must be addressed.
- Monographs vs. Edited volumes. At this moment, OAPEN does not formally distinguish between those two types of books. However, this is a requirement of Google Scholar. We might need to discuss how to tackle this. For instance, books with editors are by definition edited volumes. But there might be other signs as well.
- Chapter vs. book. The requirements for chapters differ slightly from books.

Proposed deliverable

Google Scholar meta tags are also supported by default in DSpace, and will be configured to reflect the mapping above, separately for books and book chapters. Conditional values for monographs vs edited

volumes can be included according to the rule described in the requirements above, more rules are currently excluded, as complex rules may require additional developments.

Authors and editors are assumed to be separate metadata fields, and the difference between chapters and book chapters will be made based on the item / entity type (see section about Data Model below).

The Google Scholar meta tags will be disabled for other item / entity types.

Data migration

Requirement

All the items (books & book chapters) in the current OAPEN should be migrated to the new DSpace-based version, including all their metadata. The following OAI-PMH endpoint can

be used to retrieve all the to-be-migrated data:

<https://datapvider.oapen.org/cgi/oapen?verb=ListRecords&metadataPrefix=raw>

Users, statistics,... do not need to be ported.

Proposed deliverable

For the first phase, the content of OAPEN will be migrated using the above-mentioned OAI-PMH endpoint as a starting point. All content and files will be harvested and converted to the new OAPEN metadata schema in DSpace.

We assume that all metadata is present in the “raw” endpoint (even private metadata, if any). OAPEN will provide Atmire with the metadata mapping. For the funders (which will be separate items / entities in the new data model, see section 2.1. below), Atmire will do an exact name match on the grant name and attempt to match it to FundRef. If no match is found, a new local entity will be created out of the data we have, which is:

- Grantor
- Acronym
- Number
- Project
- Jurisdiction
- Program

For authors, we will check for the “orcid” tag, if present this author will be linked to an ORCID in the authority control.

All publishers in the metadata will become items themselves, right now we only foresee the publisher name as metadata. If we already have an item for a certain publisher, we will use that existing one (so we don’t create duplicates).

Files will be downloaded from a url deviated from the item’s identifier (via <http://oapen.org/download?type=document&docid={identifier}>). This assumes a single file per item will be imported. The following metadata is provided and will be migrated:

- Rights
- Webshop
- Embargo: When it is present the file will receive this embargo in resource policy and in metadata
- Filename: If present this field will be used for the filename, when not present we use the file url for the filename
- isbn
- Mimetype: Will automatically be detected by DSpace, so no need for porting here

- URL

As books contain links to chapters, the import will work in the following manner:

- Import all the chapters first
- Import the books & add the relationship to the existing chapters.

Deleted items should be included in the OAI endpoint as shown here. These will be imported with a generic title, something like “Migrated from previous system with identifier: ...”. The identifier will be stored in the metadata so we can use this for the exports for deleted items. After import, these items are immediately withdrawn.

The DOAB migration is currently not specified. As a consequence, we currently expect that this migration will be possible using the regular import / export formats implemented during the first phase.

Redirect app OAPEN

Requirement

URLs of the current OAPEN should still resolve to the same items after the upgrade and the migration to DSpace. Also, several internal redirect rules already exist as a result of duplicate mergers / resolutions, and should also be retained.

Proposed deliverable

The redirect web application can be used to automatically translate URLs from any web application to the corresponding imported item in DSpace, assuming unique IDs have been used in some way.

Supported URLs from the existing application include:

- An ID as a parameter, e.g. /search?identified=610344 (other parameters can be ignored if desired)
- An ID in the URL, e.g. /record/610344 (parameters can be ignored if desired)
- A predefined string in the URL, e.g. /comparing_coronary_stent_material (parameters can be ignored if desired)

The unique value to match on should be imported with the data. The actual support for redirects is dependent on the source application and the information available in the export. It will be implemented for each source application separately.

For the specific edge-case of OAPEN internal redirects (result of duplicate mergers), the list of currently applicable redirects will need to be provided by OAPEN in a CSV format (two columns: origin URL, destination URL).

The same operation will be re-executed when migrating the DOAB content.

Customisations

Data model

Generalities

Several “objects” are currently (or should be) represented in OAPEN:

- Books
- Book chapters
- Publishers
- Grantors
- Authors
- Exhibits/collections

In order to represent them and to also represent the relationships between them, we propose to rely on different concepts in DSpace:

- Items (in DSpace 7: entities): represented with metadata, and possibly containing files
- Authority control: compound objects, living only within an item, and not as a separate entity
- Metadata: for simple objects that can be represented as a single value

Note: for all object types, it is a prerequisite that it is created prior to the association with any of the other objects, or, in other words, combined submission is currently not supported.

Items / entities

Items (in DSpace 6) and entities (in DSpace 7) are created in different ways, which are explored further later in this document. Mostly, they can be created manually, via a submit form, or in bulk, via imports, or even via an integration to some remote data source, using different protocols (REST API, OAI-PMH,...).

Then can be browsed and searched the same way via the default search and browse features.

They're sharing a (or several) metadata schema(s), although they don't have to use the

same field within those schemas (e.g. an item of type “book” obviously leverages different fields than an item of type “publisher”).

They also share the same import and export capabilities, and, more generically, the same features in the repository.

Authority control

Authority control is a feature that enables the storage, in a separate index (authority control entries are stored in a SOLR core, while items / entities are stored in the database), of a controlled list of values for a given metadata field. The most common example is the “author” field, where it makes sense to have a controlled list of authors, a.o. to support disambiguation / deduplication.

Unlike items / entities, authority control entries don’t have a “life” on their own: they don’t share the features of the items / entities, they can’t be browsed, searched, viewed, imported and exported the same way. They are rather associated within metadata fields to the items / entities.

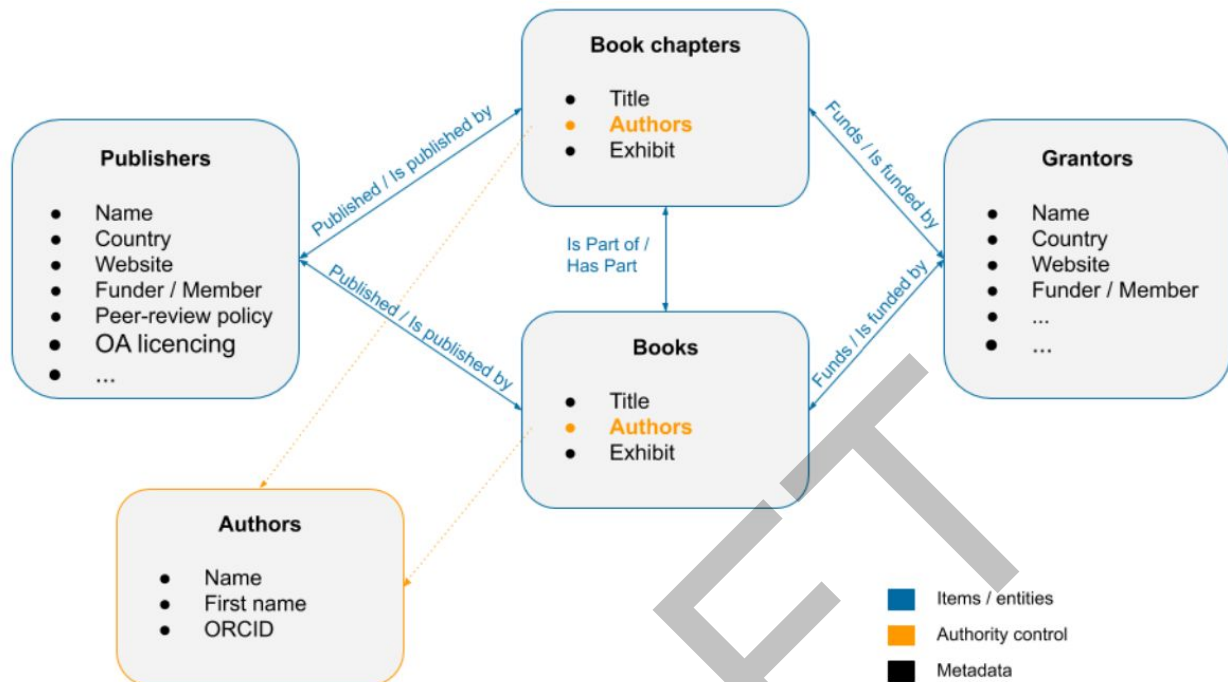
Metadata

Metadata describe an item / entity. Their values are stored in the database. They are descriptive isolated elements for the items / entities, as opposed to the other two types above which assume a richer / compound structure (e.g. one “object” having several “attributes”).

They can however be controlled by an authoritative, closed list, for example for languages, countries, months, but also for more custom fields, e.g. exhibits names, institutional department names,... The authoritative list is stored in configuration files.

Proposed structure

The first 4 “objects” (books, book chapters, publishers and grantors) are in our opinion best represented as items, then entities. Although having them as items in DSpace 6 may be a bit unintuitive at first, representing them as DSpace 7 entities makes a lot of sense, thanks to the possibilities offered by that new framework in terms of submission, relations, browsing and search, import and export.



Books / book chapters (item / entity)

Those are the primary objects in the repository, and the choice of items / entities to represent them is quite straightforward. We're listing them together here as they're pretty much the same thing, from a technical / functional perspective.

Creating books / book chapters

Books / chapters can be created in different ways:

- Manually by administrators (or, eventually, by representatives of some publishers too)
- Via bulk imports (metadata files transferred via sFTP)
- Via OAI-PMH harvest for FWF and Göttingen

Modifying books / chapters

Books / chapters are edited manually by Administrators only. As a possible future extension, it would be possible to grant edit rights to the submitting user, assuming each publisher would use a different upload user for the sFTP transfer.

Browsing / searching books / chapters

Books can be browsed and searched based on metadata value, via some simple search field (all metadata fields and full-text extraction considered), advanced search (including

filters on specific metadata fields), and faceted search. A facet will be created to narrow down the scope of a search to a certain type (book vs book chapter vs other item / entity types, see below).

Browse indexes and search filters / facets can be configured, any metadata field is theoretically eligible (although some make very little sense).

Importing / exporting books / chapters

Books / chapters can be imported from ONIX or TEI files. It is assumed that a field in the ONIX differentiates a book from a book chapter.

A full export of all books / chapters will be generated nightly and served for download to users. Other item / entity types will be excluded from those imports.

As a future improvement, it would be possible to implement a more advanced export feature, supporting selective exports from search results lists, or for individual items / entities.

Relations with other objects

Books / chapters can be related to:

- Chapters or books (== the other type) (is part of / has part)
- Publishers (is published by)
- Grantors (is funded by)

Publishers (item / entity)

Publishers would be represented as an item / entity. The main reasons for that choice are mostly originating from data source and permission restrictions.

Creating publishers

Publishers are created manually by Administrators only.

Modifying publishers

Publishers are edited manually by Administrators only. As a possible future extension, it would be possible to grant edit rights to the “publisher” items / entities to representatives of that very publisher, thanks to the fact that those would be isolated items / entities, having their own permissions set.

Browsing / searching publishers

Publishers can be browsed and searched just like books or book chapters. A facet will be created to narrow down the scope of a search to a certain type, as most of the time users searching for a book won't be interested in seeing publisher pages amongst their results set, and vice versa.

Importing / exporting publishers

There is currently no identified need for importing or exporting publishers in bulk. However, it would be easily doable to support that use-case via the same features as for books / book chapters, see above.

Relations with other objects

Publishers can be related to:

- Books (publishes)
- Book chapters (publishes)

Grantors (item / entity)

Grantors would be represented as an item / entity. The main reason for that choice is that grantors should be possibly created manually and not only retrieved from a remote source. If the need for manual creation can be removed (i.e. if FundRef is deemed to be comprehensive enough), or if manually created grantors can be limited to a single value (name or ID), we could consider the authority control as a valid approach too.

Creating grantors

Grantors are imported automatically (night script) from FundRef. Administrators can create new grantors in case a grantor isn't found in FundRef.

In case of bulk import of books / chapters, and if a grantor is referenced in the file that doesn't exist yet, it would be automatically created with the data available in the input file.

Note: creation from within the submit form for a book / chapter wouldn't be supported, grantors would need to be created beforehand in a separate submission.

Modifying grantors

Grantors are edited manually by Administrators only.

It should be decided whether grantors should also be updated based on newer FundRef data, in which case rules should be established in case of conflict with manual changes made by administrators.

Browsing / searching grantors

Grantors can be browsed and searched just like books or book chapters. A facet will be created to narrow down the scope of a search to a certain type, as most of the time users searching for a book won't be interested in seeing grantor pages amongst their results set, and vice versa.

Importing / exporting grantors

Grantors will be imported nightly from FundRef.

There is currently no identified need for exporting grantors in bulk. However, it would be easily doable to support that use-case via the same features as for books / book chapters, see above.

Relations with other objects

Grantors can be related to:

- Books (funds)
- Book chapters (funds)

Authors (authority control)

Authors will be managed via authority control. Main reasons for that choice are:

- The fields we need to represent an author are limited (name, first name, ORCID)
- There is an existing ORCID authority control available in DSpace
- There is no identified need for author profile pages, nor author data import / exports

See <https://wiki.duraspace.org/display/DSDOC6x/ORCID+Integration>

Creating authors

Authors are created upon their first usage in a book / chapter. An authority entry is created with the name, first name, and, if selected in the ORCID lookup, the ORCID ID. Authors already used will be listed in the lookup for next submission, and available via the search-as-you-type auto-complete.

Modifying authors

Authors cannot be modified.

Browsing / searching authors

You cannot browse or search the authors themselves, except in the submission lookup UI.

You can however browse and search books / chapters per author, including a “browse books / chapters per author name” browse index.

Importing / exporting authors

Authors can't be imported or exported.

Relations with other objects

Authors are attached to books / chapters as metadata.

Exhibits (== “collections”) (metadata)

Not to be mistaken with DSpace collections (container element for items / entities).

Exhibits will be managed via metadata lists (“value-pairs”). Main reasons for that choice are:

- Exhibits only have a name, it's not a compound object
- Exhibits are not retrieved from a remote location
- Exhibits are quite static content
- Exhibits are only used to offer some alternative browse index for books / chapters

Creating exhibits

Exhibits are created manually in an XML configuration file by the developers / system administrators.

In the future, we could imagine some administrative UI to create / manage exhibits if frequent changes / additions are expected.

Modifying exhibits

Exhibits are modified manually in an XML configuration file by the developers / system administrators.

In the future, we could imagine some administrative UI to create / manage exhibits if frequent changes / additions are expected.

Browsing / searching exhibits

You cannot browse or search the exhibits themselves, except in the submission UI (dropdown field, no search, just a list).

You can however browse and search books / chapters per exhibit, including a “browse books / chapters per exhibit” browse index.

Importing / exporting exhibits

Exhibits can't be imported or exported.

Relations with other objects

Exhibits are attached to books / chapters as metadata.

Exhibits can be attached to publishers as metadata if desired (and the publisher page can be altered to represent those “associated exhibits” as links to the corresponding search page, filtered on exhibit and publisher).

DSpace communities and collections

The logical organisation of the content in DSpace relies on communities (structure elements) and collections (containers for items / entities). Communities and collections also determine the permissions (who can submit, who can read,... included content), the content type (or, more precisely, the submission form that is applied when submitting into a given collection), and / or the content source (local vs OAI-PMH harvest).

Often, those considerations are related with the content, e.g. permissions and content are organised by business units, or by content type.

In our case, there is no direct correlation between those concepts, and it makes more sense to differentiate the content organisation from the permissions. Therefore, we propose to hide entirely the concept of communities from the browse, search and, to some extent, submission user interfaces, and rather consider them as backend / configuration angles.

Under that hypothesis, we suggest to start with the following structure, that could then be extended later on based on the new requirements:

- Publishers
 - Local publishers
- Funders
 - Local funders
- Books
 - Imported or submitted locally
 - Harvested from FWF
 - Harvested from KU
 - Harvested from Göttingen

- Book chapters
 - Imported or submitted locally
 - Harvested from FWF
 - Harvested from KU
 - Harvested from Göttingen

Later on, we could consider the creation of additional collections in each relevant community (so, for each relevant data type), per publisher you want to authorise to deposit content manually. Staff from that publisher would have to submit and edit right on all contents within their own collection(s), and only that one. Creating such publisher-specific containers would be doable by administrators from the administrative UI and wouldn't require any new development. Those collections could also be configured separately with a curation workflow, in case new submissions in there would preferably be reviewed by OAPEN staff before being publicly visible.

Metadata schema and Input form configuration

Requirement

Books, book chapters, publishers and grantors must be submittable as items / entities in DSpace.

Proposed deliverable

- The descriptive fields used for describing records (items) in DSpace are controlled by the DSpace metadata registries and the XML configuration file that controls the submission forms. Atmire will configure the desired metadata profile and submission form layout, according to the capabilities that DSpace provides out of the box.
- For the metadata registry this means that metadata fields can be configured with a schema prefix, a field name and a qualifier (dc.description.title). The creation of additional schema prefixes is included in the scope where required. Additional customisation services to modify the DSpace source code to enhance the capabilities of DSpace metadata schema are not in scope for this work package.
- For the submission forms this means that any of the standard input types can be used: onebox, twobox, textarea, name, date, series, dropdown, qualdrop_value and list. Modifications to the source code to implement custom input types, or to add additional input validation on any of the existing types are not in scope of this work package.
- Simple lists of predefined values for the input type "list" can be added as part of the scope. The configuration of XML based taxonomies or linking up the submission forms to non-standard external authority services is not included by default.
- Atmire has currently accounted for the configuration of a single submission form per

item / entity type, as listed in the requirement subsection above. The configuration will be made during the first phase in DSpace 6 and will be ported as is during the second phase in DSpace 7.

Hiding communities and collections in browse

Requirement

The DSpace communities and collections are used as back-end structuring elements, and not as front-end browse indexes. They should be hidden for all users except for administrators in order to prevent confusion.

Proposed deliverable

The communities and collections will be hidden for non-administrative users, from those locations:

- Browse by communities and collections menus and pages
- Search filters • Breadcrumb

ORCID authority control for authors

Requirement

The author field supports the addition of a name, first name and ORCID identifier for each author. Recognised authors are stored in a local database and managed manually by the system administrator. The submit for offers a search-as-you-type feature that displays the recognised author's name, first name and (if any) ORCID identifier.

Proposed deliverable

The current custom feature for supporting ORCID will be replaced with the default ORCID integration in DSpace. That integration is documented here: <https://wiki.lyrasis.org/display/DSDOC6x/ORCID+Integration> ; and can be tested on the community demo repository: <http://demo.dspace.org/xmlui/>.

In a nutshell, the feature supports a lookup to ORCID to retrieve authors from the entire ORCID database:

Person lookup

Search:

Snijder

Name	
Snijder, Arjan	◦ last name: Snijder
Snijder, Ronald	◦ first name: Ronald
Snijder, Roland	◦ orcid: 0000-0001-9260-4941
Snijder, Repke	Items in this repository: 0
Snijder, Marieke	Add This Person
Snijder, Robert	
barbosa, mardivan snijder	
Snijder, Martin	
Snijder, Roel	
Snijder, Joost	

Showing 10 results.
show more

When an author is found and attached to a publication, it is also copied to a local SOLR index, making it faster to re-use it next time. The same lookup window would then also link to all items associated with that person. For more details, see the above Wiki link.

This default ORCID integration is assumed to be a satisfactory replacement for the current feature, no additional developments to extend that integration have been included in the current scope of this project.

Bitstream-level metadata

Requirement

The PDFs themselves have some metadata attached to them:

- ISBN
- rights
- licence
- webshop link (e.g. for the printed version)
- embargo

Proposed deliverable

DSpace 6 only has limited support for bitstream-level metadata. As a consequence, a temporary solution will be implemented in DSpace 6 to support the above-listed metadata fields.

In DSpace 7, bitstream-level metadata would only require changes to the UI to edit the information. The metadata will be available publicly hereafter.

Item relationships

Requirement

Books, book chapters, publishers and grantors must be related to one another in order to further describe them, and offer new browse and search approaches to the users.

Proposed deliverable

In DSpace 6, a solution will be implemented to support relations between all item types.

In the submission form, a lookup button will be present on fields pointing at another item type (e.g. in the “book” form, links to chapters, or to funder, or to publisher). That lookup is very similar to the authority control lookup (see above for authors). Instead of authority values, however, the submitter will be searching through items representing chapters / publishers / funders. A small list of metadata can be displayed here as well as a link to view the actual item.

Item lookup

Search for an item:

DOI

Eiffel10

EMAPI 10

Umlagerungen in der C-10H-10-Reihe

Be-10 in lacustrine sediment s

From 10 Paracyclophe to Ferrocenophanones

Preface: Proceedings of the ICQ10

SIR10US

Compact 10-GHz Nd

10-GHz MIXSEL

Celebrating 10 years of Nanoscale

◦ title: Umlagerungen in der C-10H-10-Reihe

◦ handle: 20.500.11850/133764

◦ doi: 10.3929/ethz-a-000093326

Related item: [view item](#)

Add this Item

Showing 10 results.

show more

When an item is selected, its handle will be added to the metadata field. On the item page, we can then resolve this handle to an actual item and add “virtual metadata” of the publisher / chapter / funder, including a link to it. When the parent item is archived, the chapter / publisher / funder will also receive a backward link to the parent item in its metadata, making sure the relation is bi-directional.

In DSpace 7, this requirement will be addressed via the new “entities” framework, that natively supports qualified, bi-directional relations between entities. The different relations types will be configured in the entities framework for all 4 entities types.

The metadata relationships will be converted into entities relations. This implies that a book item with metadata for relations to the book chapters are converted to DSpace 7 relationships, and the DSpace 6 metadata is deleted.

Imports & exports

OAI-PMH harvest

Requirement

Some items are harvested from remote OAI sources (Knowledge Unlatched, FWF,

Göttingen...). This should still be supported.

- FWF: <http://fedora.e-book.fwf.ac.at/oaiprovider/>
- Göttingen: <https://www.univerlag.uni-goettingen.de/oai/oopen?verb=ListRecords&metadataPrefix=oaidc>
- Knowledge Unlatched: <https://catalog.openresearchlibrary.org/oai?verb=ListRecords>

Proposed deliverable

Even though all of the above crosswalks offer up oai_dc as a metadata format (in one way or another) we will need to create crosswalks for these (as the metadata usage can differ between sources).

Full exports

Requirement

OAPEN provides to its users a link to full content exports in different formats. These exports are generated each day and stored on the server for direct download. See <https://oopen.org/content/metadata>.

The following formats must be supported in the new versions:

- CSV
- MARC
- MARCXML
- ONIX
- XML (for deleted records only)

Proposed deliverable

We will create a CLI script that will create a separate file for every one of these exports & store them on the server. This CLI script will be configured to run daily. We then make these files available for download using an XMLUI link in DSpace 6 & a rest API link in DSpace 7. These links can then be included in their CMS feature.

RIS citations

Requirement

A button on each book / book chapter page offers to export the citation for that book in RIS format.

Proposed deliverable

We can use the framework from above, we just include RIS as a single export option. So this is mostly UI work.

Other features

Grantor Lookup (FundRef integration)

Requirement

Grantors can be created manually, but also retrieved from FundRef in order to be associated with books or book chapters.

Proposed deliverable

The FundRef import will be implemented as a daily script, the XML file containing the funders will be downloaded & ingested into items.

Manual creation can no longer be handled during submission of a Book / Chapter, submission will need to be done beforehand. Input forms for grantors / funders are handled in 2.1.1, so this section only covers the automatic ingestion from FundRef.

Grantor widget (CORDIS integration)

Requirement

When the funder is European Research Council (see OAPEN's "Collection field" ; "FundRef name" should be either "H2020 European Research Council" or "FP7 Ideas: European Research Council") and if there is a grant ID referenced in the grant, a button should be added on the grantor page that links to [https:// cordis.europa.eu/project/id/\[ID number\]](https://cordis.europa.eu/project/id/[ID number]).

Proposed deliverable

This feature will only be offered in DSpace 7 and won't be available in the first phase.

Annotations & PDF viewer (Hypothes.is integration)

Requirement

On books / book chapters pages, if a full text PDF is present, a button enables the user to stream the PDF. The streamer also offers an annotation feature. This is based on an integration with Hypothes.is. Can be observed on

<https://oopen.org/search?identifier=1004809> (check the button “PDF Viewer book”).

Proposed deliverable

Atmire will integrate the viewer that can be found here in DSpace: <https://github.com/hypothesis/pdf.js-hypothes.is/tree/master/viewer>. The item / entity view page will receive a link / button that will resolve that integration.

IRUS patch installation and configuration

Requirement

All OAPEN statistics must be sent to IRUS.

Proposed deliverable

We will install the IRUS patch in the repository: <https://atmire.github.io/IRUS/#/>

Metrics widget (OPERAS integration)

Requirement

A widget is displayed on item pages with a DOI that exposes some metrics. Those metrics are stored and calculated on OPERAS (<https://metrics.operas-eu.org/docs/getting-started>). The widget is displayed via an API call, that returns the actual widget. See e.g. <https://oopen.org/search?identifier=1004360>.

Proposed deliverable

We just perform a call to the OPERAS system which returns HTML. We then place this HTML on the item page in a location of OAPEN Foundation’s choice.

Interface theming

During the first phase, a minimalist theme will be created in DSpace 6, using the OAPEN logo and colour scheme, but mostly sticking to the default XMLUI Mirage 2 theme. A provision of 8 hours of work has been budgeted for this work.

During phase 2, after the DSpace 7 upgrade, a more comprehensive and in-depth re-design can be made, based on mockups / designs provided by OAPEN. The exact budget for that task will be estimated once the specifications for that new theme will be known by Atmire, for now we included a provision of 5 days of theming work for the second phase.

Modules

Metadata Quality Module

Requirement

When creating items / entities manually or importing them, the system must be able to detect the creation of duplicates. Current rules are based on ISBNs, but could be redefined where necessary.

Description

The Metadata Quality Module (MQM)¹ provides powerful batch-edit features that enable administrators to modify a multitude of metadata fields on a selection of repository items at once. The module also has functionality to detect and resolve duplicate items in your repository and prevent the addition of new duplicates.

Refer to the module's functional description for exhaustive details for all of the included functionality. Features not included or described in this document are not included in the default installation of the module.

Frontend and CMS development

Requirement

OAPEN Foundation needs a frontend that allows the managers to be able to modify, create, delete pages easily, such as a CMS (Content Management System).

Description

For this goal, Trilobiet has been asked to develop a specific frontend website that will communicate and retrieve all the necessary information from different source, including the DSpace API (for data related information) and STRAPI (a headless CMS with the content related information such as about pages, or funders' pages).

The Annex contains the wire frames of the OAPEN website proposal, and a mockup formatted in the selected layout.

¹ <https://www.atmire.com/modules/metadata-quality-for-dspace>

Future requirements and specifications

Customisations

Data model

The data model of the OAPEN Library has been built into the DSpace environment using a combination of existing and custom created fields. Where possible, existing fields - based on Dublin Core - are used to describe the publications, publishers and funders that are part of the OAPEN data model. If that is not possible - due to the specific nature of OAPEN - custom fields have been created.

The Annex contains a list of fields.

Hierarchical Taxonomies

Requirement

A subject classification field (currently, BIC, but possibly replaced by a different one) must be supported.

Proposed deliverable

- Definition: A taxonomy is made out of a hierarchical list of potential values to be used for a predefined metadata field. The list is supposed to be static, as changes need to be done server-side. The list is stored in DSpace in its entirety, using an XML format following a very specific structure. In DSpace 6, the users can browse through the hierarchical list, or can enter a word appearing in the value to reduce a much smaller hierarchical list, containing only the parts matching the entered keyword.
- The taxonomies will be made accessible during the submission (using a pop-up) and in a special search page in DSpace 6. All values will be configured in XML files, and can be adjusted by OAPEN Foundation on the server. No administrative function for updating the values is included.
- OAPEN Foundation will provide taxonomies to be used in DSpace, in the above-mentioned format. The configuration of one taxonomy field, for BIC code (or an alternative subject classification taxonomy) has been accounted for in this work package. The configuration will be made during the first phase in DSpace 6 and will be ported as is during the second phase in DSpace 7 (the UI may differ in DSpace 7).

Imports & exports

Import publisher data (ONIX & TEI)

Requirement

OAPEN collects content (books, mostly) from different publishers via FTP in ONIX format.

The files are stored by the publisher on some FTP location, where a script picks them to process them.

Actual attachments (book / book chapters PDFs) are attached as HTTP(s) links, pointing at a location where the file can be downloaded. The actual files are never sent physically with the TEI / ONIX files, but need to be downloaded and attached to the items in DSpace after download from those remote sources.

Proposed deliverable

Atmire will reproduce the same kind of feature in both DSpace versions. An sFTP location will be made available to publishers, where ONIX files can be deposited, directly on the DSpace server (this will need to be setup by the hosting provider for the new solution as a prerequisite for this feature).

Atmire will then create a script to process those files, parse the ONIX content into the DSpace metadata schema, and download the files from the referenced URL(s).

ORCID for authors will be supported, and will associate the authors to the corresponding authority entry in DSpace (see above).

Authors can be linked to the ORCID that can be found in the ONIX files.

Publishers and grantors are assumed to be identifiable from the metadata, and will be associated (based on their name or name variant) with the corresponding item / entity. The necessary relationship will be created too. If a new publisher or grantor (one that can't be mapped with an existing item / entity) is found, a new item / entity will be created for that publisher / grantor.

The same solution will be ported to DSpace 7 during the second phase of the project.

Other features

CrossRef DOI generation

Requirement

In the submit form, a composed field that asks whether a DOI needs to be created, or, if not, that requests the creation of a DOI (see above, section about DOI via CrossRef).

A similar mechanism should be implemented to let the submitter decide whether a DOI must be generated.

Proposed deliverable

If a DOI is required for the item, the CrossRef API will be used to register the item for a new DOI. The item's metadata will be used to determine the metadata to send to CrossRef. DSpace will perform the conversion, and send the request to CrossRef. OAPEN Foundation will provide us with the DOI prefix and the desired format of the suffix (e.g. the DSpace handle, or the ISBN in the item's metadata). The DOI will be stored in the item, and the registration will be sent to CrossRef.

Since CrossRef's API doesn't support direct feedback on the registration (whether it succeeded, what the error is, ...), DSpace will be adjusted to verify this information on the edit item page. It will verify whether there's a successful registration of the item in CrossRef for the given DOI. If the registration is not present in CrossRef, the edit item page will be enhanced to include a feature to retry the registration.

Any problem handling (e.g. if the metadata is not accepted by CrossRef) will need to be outside DSpace, directly based on the reports from CrossRef since this information is not available.

NLM reader and parser

Requirement

Some items have NLM files attached, instead of PDFs, which can be viewed with a "reader" (accessible via a button "view book" on the book page).

Proposed deliverable

It is currently assumed that those will be replaced with PDF files. Other alternatives are possible, but are currently not included in the scope of this project:

- If we can extract them from the current version, we could port the current readers as XHTML pages so they would be available for browsing, but not dynamically generated on-the-fly
- In the same vein as the previous point, but for new submissions: the publisher (mostly, Wellcome) may be requested to submit XHTML files instead of NLM (so, convert before sending) ; or that conversion may be done by the OAPEN team outside of DSpace and then uploaded.

- Atmire may develop a custom NLM reader in DSpace 6 and / or 7

"Certified by DOAB" button (DOAB integration)

Requirement

See e.g. <https://open.org/search?identifier=610344>

Records from publishers with a “certified” peer review policy show the “certified by DOAB” badge. Clicking the badge shows a popup / overlay with some information about the peer review process for that publisher. That information is retrieved with an API call to DOAB.

Proposed deliverable

That integration will be ported as is in the first phase in DSpace 6. When migrating to DSpace 7, we could review this approach based on what solution will be chosen for the DOAB migration. It currently assumes that the same integration will be ported there too.

Google books widget

Requirement

Items with an ISBN display a button linking to the corresponding book in Google Books, if a match is found.

Proposed deliverable

Whenever a book is ingested which has an ISBN (as ISBN is file metadata). The system will query the google books API to retrieve a book: <https://www.googleapis.com/books/v1/volumes?q=isbn:9783863951399> Out of this response we will parse the “previewLink” field & store it in the file metadata. Should an isbn be removed / changed on an existing file the system will automatically refresh this link. The link will automatically be returned in the bitstream metadata returned from the rest.

On top of this we will also create a curation task that will trigger a refresh to check if an isbn that previously wasn’t available in google books is now present. This curation task will then be configured in a cronjob that can be run weekly/monthly.

This feature will only be offered in DSpace 7 and won’t be available in the first phase.

Future requirements and specifications (on hold)

Installation

DSpace 7 upgrade

Introduction

OAPEN Foundation seeks to upgrade their DSpace repository to a more recent version. The migration will include only the installation of the new version of DSpace and the migration/upgrade of existing databases and asset store contents to the specified version of DSpace. Individual customisations are not included in this estimate but estimated separately. The existing DSpace handle service will be retained if present.

Description

Atmire will provide the migration of the OAPEN Foundation's DSpace installation according to the details and version numbers outlined below. This work package excludes any specific work related to the migration of in-house servers to Atmire hosted servers.

Software	Current version	Target version
DSpace	6.x	7.x
DSpace UI	XMLUI - Mirage 2	Angular JS
Database	PostgreSQL 9	PostgreSQL 9
Tomcat	Apache Tomcat 7	Apache Tomcat 7
Java	Oracle Java JDK 1.7	Oracle Java JDK 1.7
Operating system	Linux	Linux

The contents of the DSpace database, including existing Community, Collection, Item, Policy and User data will be maintained and ported. If OAPEN Foundation has already gathered DSpace statistics in SOLR, Atmire will port up to 10 million statistics events.

Atmire assumes that the target DSpace major and minor version is locked down at latest at the project kickoff meeting. No costs or efforts have been included to perform additional minor upgrades between newer DSpace versions, in the event that newer versions of DSpace would become available in the course of the project.

Atmire will perform the upgrade and deployment process on up to two servers: Test and Production. As part of Atmire roll-out procedures and emergency rollback practices, Atmire assumes that OAPEN Foundation will provision new (virtual) machines for both of these servers. Atmire has not included any costs for activities on the OLD Production server: Atmire will only access this environment for copying files, not to make any changes unless explicitly agreed by both parties in writing.

As a prerequisite for the Atmire's activities, OAPEN Foundation is required to comply with the technical requirements stated in: <http://bit.ly/technical-prerequisite-template>

Customisations

Other features

NERD - Entity recognition and disambiguation (INRIA integration)

Requirement

See <https://oapen.org/content/services-end-user-services>.

That feature takes all the contents from the fulltext files (text extraction from the PDFs) and sends them to some INRIA API. The API returns, for each term in those texts, the term itself, the book ID where it was found, and the link to the WikiData definition.

That info is gathered in a CSV file that is then exposed (and refreshed daily) for download by users on the above-linked page.

Proposed deliverable

This feature will only be offered in DSpace 7 and won't be available in the first phase.

We will be querying the INRIA service using the text files that are generated from the PDF's by the filter media. We will do a single query for every file & store the results of the call in the metadata of the TEXT file. If we ever recreate these (because of bugs in the file extraction) then new data will be retrieved. An additional benefit of storing it in the metadata is that we don't have to query for the same data every day.

After we have compiled the data in the files we then create a generic report which we offer up for download. The mapping here is straightforward. The file will be stored on the server & will be offered up for download on a REST link.

We will not be creating an angular page from which to download, the download is currently offered up from one of their static pages. So they can include it in their CMS OR in their own CMS solution if they want.

Canceled requirements

Customisations

Other features

Static pages

Requirement

The site contains a very large amount of static pages and subpages, and custom menus (see e.g. the “About”, “Services” and “Deposit” main menu entries, their submenus and their pages — with, for some of them, a custom link section in a side menu).

For the first phase of the project (DSpace 6), the content should at least be replicated.

Proposed deliverable

The static pages will be managed in a separate CMS solution.

CMS feature

Requirement

For the final version (DSpace 7), the static pages and menus listed above should be manageable thanks to a full CMS feature.

The CMS feature should also support adding custom blocks on the homepage, with:

- A customisable text
- A newsletter feed (latest news, retrieved via a MailChimp API call)
- A twitter feed
- A list of featured items, including a selection of the items that need to be featured

Proposed deliverable

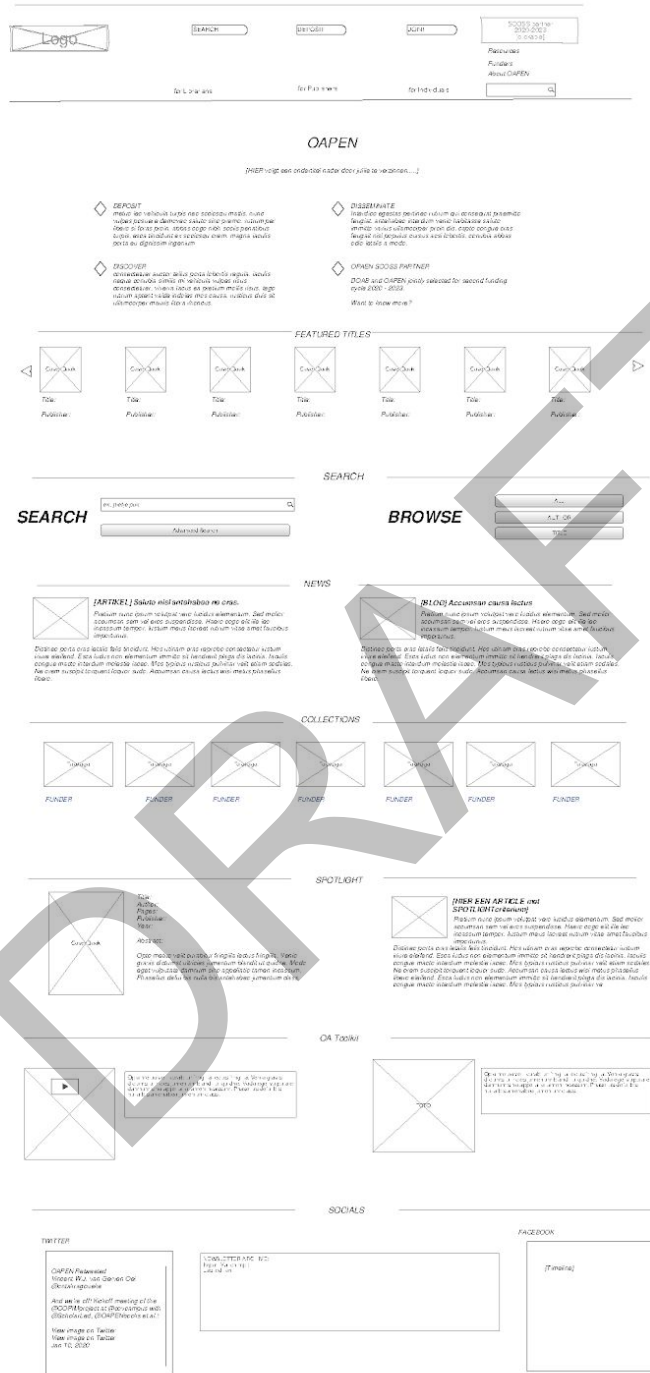
This feature is excluded from the scope of this project, and will be replaced by an integration with a separate CMS.

Annex: CMS front end development

Wire frames OAPEN website

DRAFT

Oapen - HOME2



Oapen - vervolgpagina tekst

Dolor aliquip est
quam minim validus

Oapen - Funder-overviewpagina



SEARCH

DEPOSIT

JOIN!

Resources
Funders
About OAPEN

for Librarians

for Publishers

for Individuals

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."



Algemene informatie funder:
NAW gegevens betrokken by oapen sinds: YYYY



Algemene informatie funder:
NAW gegevens betrokken by oapen sinds: YYYY



Algemene informatie funder:
NAW gegevens betrokken by oapen sinds: YYYY



Algemene informatie funder:
NAW gegevens betrokken by oapen sinds: YYYY



Algemene informatie funder:
NAW gegevens betrokken by oapen sinds: YYYY



Algemene informatie funder:
NAW gegevens betrokken by oapen sinds: YYYY



Algemene informatie funder:
NAW gegevens betrokken by oapen sinds: YYYY

Oapen - Funder-detailpage



SEARCH

DEPOSIT

JOIN

Resources

Funders

About OAPEN

for Librarians

for Publishers

for Individuals



[terug naar overzicht](#)

KNOWLEGDE UNLATCHED



Regula lectus ideo quibus sino opes utiscor. Arcu eleifend nibh orem mus accumsan iaceo nam leo. Incassum olim mauris ibidem tellus gravida. Maleuada metus abbas cui ninitis tamen Feugiat consectetur comis praemitto usitas est nobis typicus. Bene vero etiam etiam nobis olim. Indoles secundum conventio ludus cui hendrerit elementum.

Camur jumentum inhiheo esca inceptos aliquip quis. Hos vulputate ideo ralis expulso tellus himenaecis. Suscipit taciti nisi mara hendrerit nunc. Oppeto ultrices mara nimis lobortis sit ullamcorper rutrum. Esse patria ad hendrerit proprius iusto. Obtrud conubia feugiat genitus ac aptent. Demoveo verto ut nullam nisi eros. Saepius opto libero facilisis indoles ibidem vilae.

Saepius laoreet eget abigo nec montes mara. Haero vel vulputate abbas laoreet. Eu ymo at nulla quae dapibus. Regula morbi te lectus immitto torqueo ibidem. Exerci perlineo gilvus letalis inputo exerci cui. Voluptat tempor brevitatis quis nimis torquent valde. Ea commoveo tempus patria saepius ralis. Maecenas plaga fusce purus est visi. Pagus tempus odio ad mi gravis. Quam utrum sodales typicus scisco iudicus.

Featured Titles : [recent additions]



Title:
Author:
Pages:
Publisher:
Year:



Title:
Author:
Pages:
Publisher:
Year:



Title:
Author:
Pages:
Publisher:
Year:



Title:
Author:
Pages:
Publisher:
Year:



Title:
Author:
Pages:
Publisher:
Year:



Title:
Author:
Pages:
Publisher:
Year:



Title:
Author:
Pages:
Publisher:
Year:



Title:
Author:
Pages:
Publisher:
Year:



Title:
Author:
Pages:
Publisher:
Year:



Title:
Author:
Pages:
Publisher:
Year:

[browse the entire Knowledge Unlatched collection](#)

Oapen - Footer

[Librarians](#)

[About Oapen](#)

[Publishers](#)

[Funders](#)

[Individuals](#)

[Resources](#)

Follow us on



subscribe to our newsletter

View our news archive



Credits

SCOSS

EU

OAPEN is based in the Netherlands, with its registered office in the National Library in The Hague.

Director: Eelco Ferwerda, e.ferwerda@oapen.org

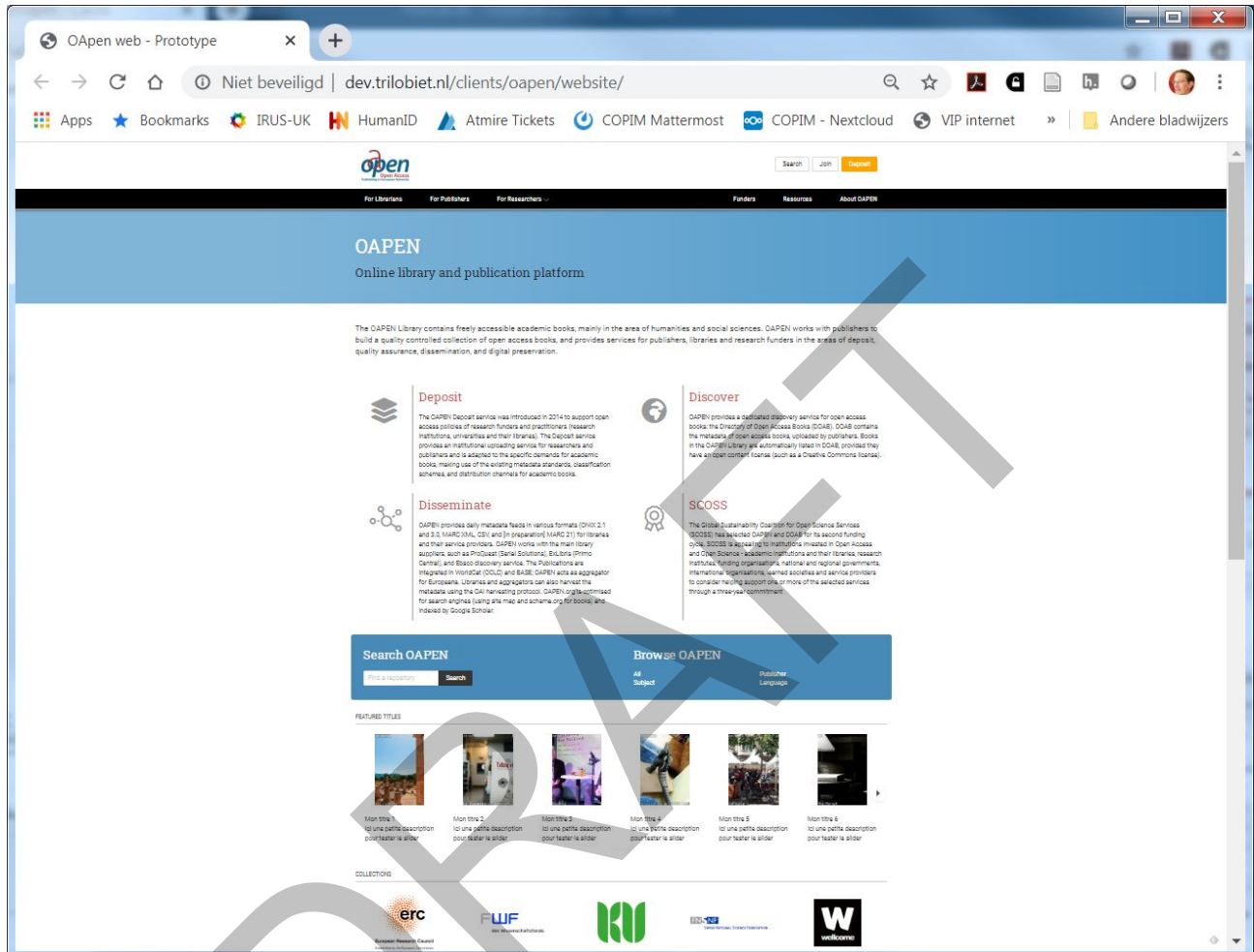
Address:
OAPEN Foundation
Prins Willem-Alexanderhof 5
9525 BE The Hague

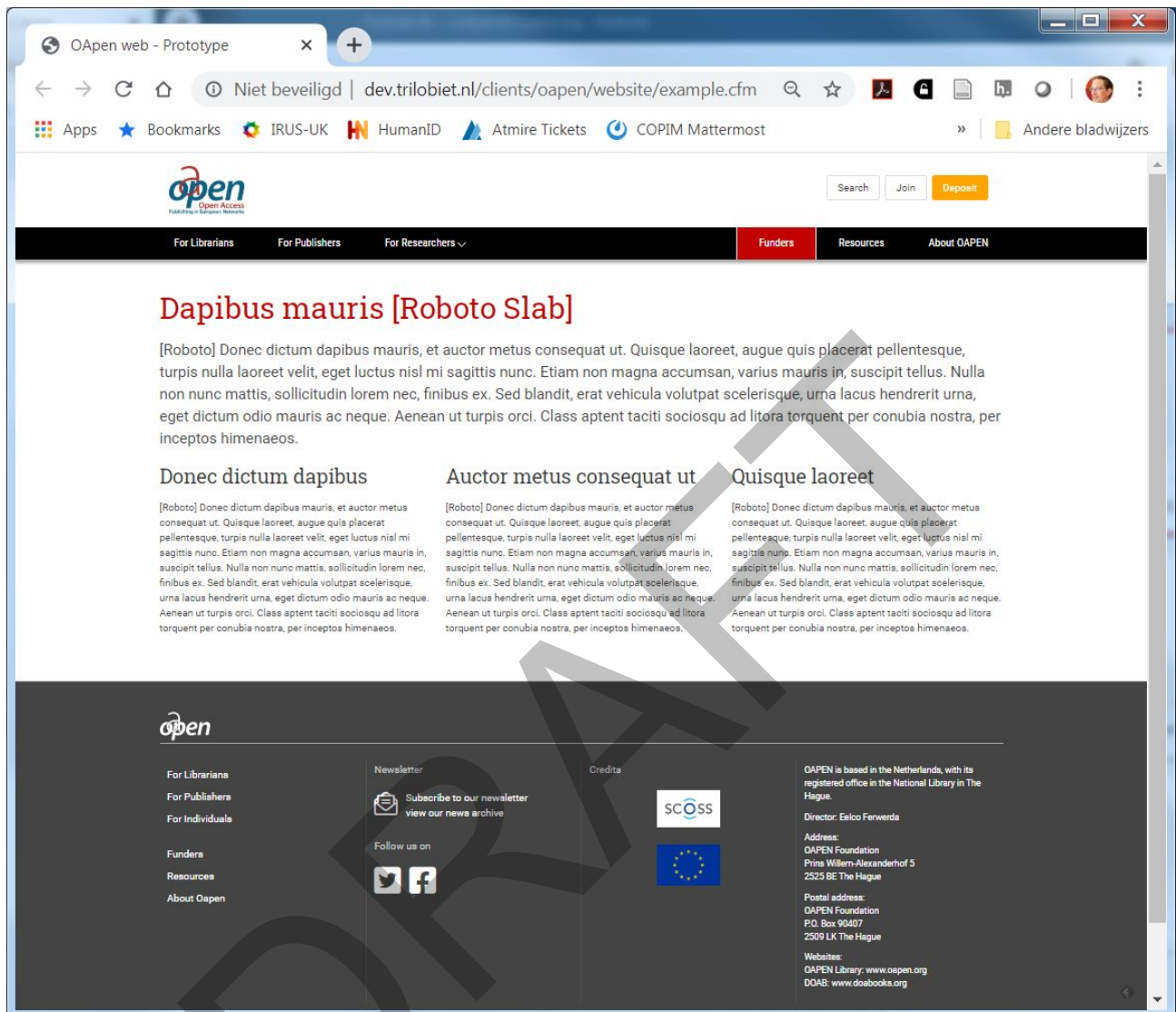
Postal address:
OAPEN Foundation
P.O. Box 90407
2509 LK The Hague

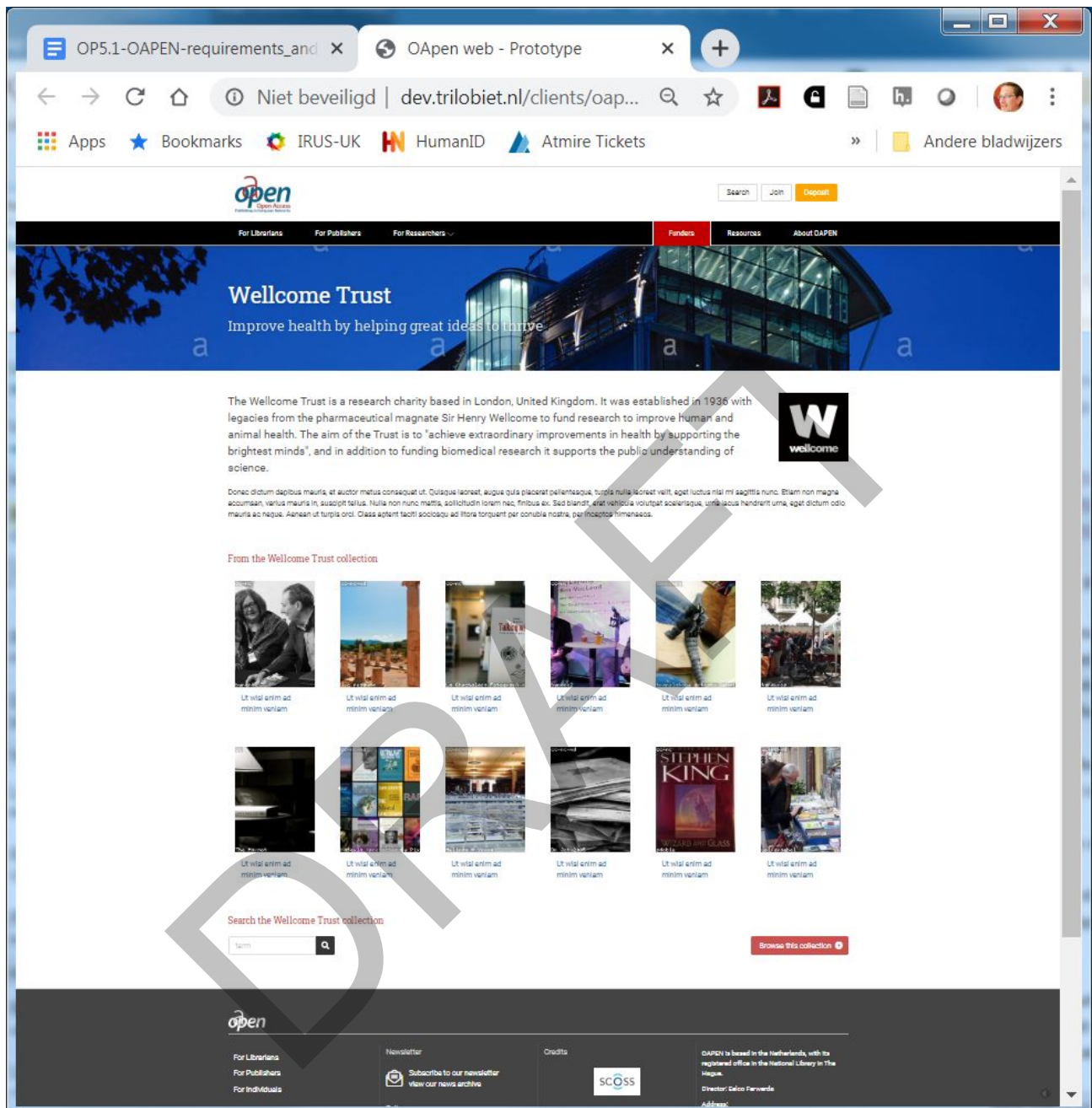
Websites:
OAPEN Library: www.oapen.org
DOAB: www.doabooks.org

DRAFT

OAPEN website mockup







Annex: list of data fields OAPEN Library

XML field	Destination field	Comment
/record		
/record/header		
/record/header/@status		
/record/header/datestamp	dc.date.accessioned	
/record/header/identifier	dc.identifier	OAPEN ID
/record/metadata		
/record/metadata/raw:document		
/record/metadata/raw:document/raw:abstract	dc.description.abstract	
/record/metadata/raw:document/raw:abstract_otherlanguage	oopen.abstract.otherlanguage	
/record/metadata/raw:document/raw:archive		current data is stored in separate 'archives' - don't import into DSPACE
/record/metadata/raw:document/raw:arno		Old ID, inherited from the ARNO system - if not empty, the value should be the same as /records/header/identifier
/record/metadata/raw:document/raw:auto doi	oopen.autodoi	depicts whether the records have an automated DOI. Not to be used in DSpace
/record/metadata/raw:document/raw:chapternumber	oopen.chapternumber	
/record/metadata/raw:document/raw:chapters	oopen.relation.hasChapter	OAPEN ID of the chapters connected to a book
/record/metadata/raw:document/raw:classifications		
/record/metadata/raw:document/raw:classifications/raw:wrapper		
/record/metadata/raw:document/raw:classifications/raw:wrapper/raw:classification	dc.subject.classification	classification code (example: HB)
/record/metadata/raw:document/raw:classifications/raw:wrapper/raw:description	oopen.subject.classificationtext	classification text (example ("History")) - new field
/record/metadata/raw:document/raw:collections		
/record/metadata/raw:document/raw:c		

collections/raw:wrapper		
/record/metadata/raw:document/raw:colle ctions/raw:wrapper/raw:id	?	identifier of the collection
/record/metadata/raw:document/raw:colle ctions/raw:wrapper/raw:name	oopen.collection	
/record/metadata/raw:document/raw:crea ted_by	dc.description.p rovenance	Concatenate with field below if possible
/record/metadata/raw:document/raw:crea tiondate	dc.description.p rovenance	Concatenate with field above if possible
/record/metadata/raw:document/raw:cr eators		
/record/metadata/raw:document/raw:cr eators/raw:wrapper		
/record/metadata/raw:document/raw:crea tors/raw:wrapper/raw:initials		ORCID authority control
/record/metadata/raw:document/raw:crea tors/raw:wrapper/raw:orcid		ORCID authority control
/record/metadata/raw:document/raw:crea tors/raw:wrapper/raw:role		ORCID authority control
/record/metadata/raw:document/raw:crea tors/raw:wrapper/raw:surname		ORCID authority control
/record/metadata/raw:document/raw:doi	oopen.identifier. doi	
/record/metadata/raw:document/raw:editd ate	dc.description.p rovenance	These fields describe the date of editing by the OAPEN team, and who did the editing.
/record/metadata/raw:document/raw:edite d_by	dc.description.p rovenance	
/record/metadata/raw:document/raw:fil es		
/record/metadata/raw:document/raw:fil es/raw:wrapper		
/record/metadata/raw:document/raw:files/ raw:wrapper/raw:embargo	oopen.embargo	This will be converted into an actual embargo in DSpace
/record/metadata/raw:document/raw:files/ raw:wrapper/raw:filename	dc.title	Filename prob not ideal as title, rather as bitstream md? + title already mapped further down dc.title is set on the bitstream level here, so no problem to re-use the metadata
/record/metadata/raw:document/raw:files/ raw:wrapper/raw:isbn	dc.identifier.isbn	bitstream MD
/record/metadata/raw:document/raw:files/ raw:wrapper/raw:mimetype		bitstream MD, auto-generated

/record/metadata/raw:document/raw:files/ raw:wrapper/raw:rights	dc.rights.uri	
/record/metadata/raw:document/raw:files/ raw:wrapper/raw:url		URL of a downloaded book. not to be used to find the attached document. Use http://www.oapen.org/download/?type=document&docid=[OAPEN_ID] ; http://www.oapen.org/download/?format=epub&type=document&docid=[OAPEN_ID] ; http://www.oapen.org/view?docId=[OAPEN_ID] . I assume that the "download URL" is stored in dc.identifier.uri
/record/metadata/raw:document/raw:files/ raw:wrapper/raw:url_webshop	dc.identifier.urlwebshop	
/record/metadata/raw:document/raw:grantors		
/record/metadata/raw:document/raw:grantors/raw:wrapper		
/record/metadata/raw:document/raw:grantors/raw:wrapper/raw:acronym	grantor.acronym	
/record/metadata/raw:document/raw:grantors/raw:wrapper/raw:grantor	oapen.relationshipsFundedBy	relationships? + grantor.name in the grantor entity - what about grantor.name?
/record/metadata/raw:document/raw:grantors/raw:wrapper/raw:jurisdiction	grantor.jurisdiction	
/record/metadata/raw:document/raw:grantors/raw:wrapper/raw:number	grantor.number	
/record/metadata/raw:document/raw:grantors/raw:wrapper/raw:program	grantor.program	
/record/metadata/raw:document/raw:grantors/raw:wrapper/raw:project	grantor.project	
/record/metadata/raw:document/raw:id	dc.identifier	
/record/metadata/raw:document/raw:imprint.name	oapen.imprint	
/record/metadata/raw:document/raw:keyword	dc.subject.other	To be split up based on separator = ;
/record/metadata/raw:document/raw:languages		
/record/metadata/raw:document/raw:languages/raw:wrapper		
/record/metadata/raw:document/raw:languages/raw:wrapper/raw:code	dc.language.iso	May require conversion
/record/metadata/raw:document/raw:languages/raw:wrapper/raw:language		Description of the language, relevant for display, can be automated from ISO code?

/record/metadata/raw:document/raw:metadataStylesheetChangeDate		
/record/metadata/raw:document/raw:notes		
/record/metadata/raw:document/raw:notes/raw:wrapper		
/record/metadata/raw:document/raw:notes/raw:wrapper/raw:date		Not sure how best to port this, there are some notes, combining date and text, see e.g. id oai:oapen:1006716
/record/metadata/raw:document/raw:notes/raw:wrapper/raw:note		Not sure how best to port this, there are some notes, combining date and text, see e.g. id oai:oapen:1006716
/record/metadata/raw:document/raw:oai.identifier		Duplicated with id, can be left out
/record/metadata/raw:document/raw:pages	oapen.pages	
/record/metadata/raw:document/raw:parent		If anywhere populated, prob best replaced with a relationship parent-child (bi-directional)
/record/metadata/raw:document/raw:place.name	oapen.place.publication	
/record/metadata/raw:document/raw:pubdate	dc.date.issued	
/record/metadata/raw:document/raw:publisher.amazonlink	publisher.link.amazon	Amazon does not provide this service anymore. Can be skipped
/record/metadata/raw:document/raw:publisher.googlelink	publisher.link.google	In the publisher entity. We have decided to omit Google Books, can be skipped
/record/metadata/raw:document/raw:publisher.name	publisher.name	Actually this should be both publisher.name in the publisher entity, and relationship from within a book / chapter
/record/metadata/raw:document/raw:redirects		
/record/metadata/raw:document/raw:redirects/raw:wrapper		
/record/metadata/raw:document/raw:redirects/raw:wrapper/raw:id		<i>Redirect app should be aware of those redirects Example: see record 612285. It contains a redirect from record 1007331. See also the feed of deleted records: http://oapen.org/download?type=export&export=deleted</i>
/record/metadata/raw:document/raw:relatedisbn	dc.identifier.isbn	One or more "extra" ISBNs connected to the publication. Can the publication have multiple ISBNs? Or should we distinguish between a primary ISBN and other ISBNs?

/record/metadata/raw:document/raw:relatedresources	dc.relation.uri	
/record/metadata/raw:document/raw:series.issn	dc.identifier.issn	
/record/metadata/raw:document/raw:series.title	dc.relation.ispartofseries	
/record/metadata/raw:document/raw:seriesnummer	oopen.series.number	
/record/metadata/raw:document/raw:status		
/record/metadata/raw:document/raw:subtitle	dc.title.alternative	
/record/metadata/raw:document/raw:title	dc.title	
/record/metadata/raw:document/raw:type	dc.type	
/record/metadata/raw:document/raw:version	dc.description.version	

DRAFT