CENDARI Documentation

Release 0.3.0-273

CENDARI Project

July 15, 2016

This is the official documentation of CENDARI infrastructure. This includes user guides to various tools provided and example use cases as well as answers to frequently asked questions

The documentation further includes the technical documentation of the different components and their respective installation instructions.

Note: These docs are maintained by the CENDARI project team. They are work in progress and extended during the project's life.

Learn about CENDARI and what it can do.

1.1 Overview

CENDARI (Collaborative European Digital Archive Infrastructure) is a research collaboration aimed at integrating digital archives and research resources on medieval and modern European history.

This document describes the tools and applications developed during the project's lifetime. It is aimed both at users of the infrastructure and administrators or developers who either want to set up an own instance of the tools or intend to enhance the tools.

1.2 Components

CENDARI consists of several components.

The following applications are visible to the end user.

- Notes VRE, see Working with the NTE
- TRAME, see TRAME

For power users, CENDARI provides the

- AtoM Archival Director, see CENDARI Archival Directory

The following components are back end applications, a user does not interact with.

- CENDARI Data API
- Litef
- Pineapple
- RDF converter
- NERD

All CENDARI curated data is stored in the

- CKAN repository

This section contains a collection of guides and HowTos.

2.1 Authors

- Carsten Thiel, UGOE (thiel@sub.uni-goettingen.de)
- Hanna-Lena Meiners, UGOE (meiners@sub.uni-goettingen.de)
- Stefan Buddenbohm, UGOE (buddenbohm@sub.uni-goettingen.de)
- Kathleen Walker-Meikle, CERL (kathleen.walker-meikle@cerl.org)
- Stijn van Rossem, CERL (stijn.vanrossem@cerl.org)
- Jörg Lehmann (joerg.lehmann@fu-berlin.de)

2.2 Introductory Video

The introductory Video to the NTE has been published on Youtube.

2.3 Working with the NTE
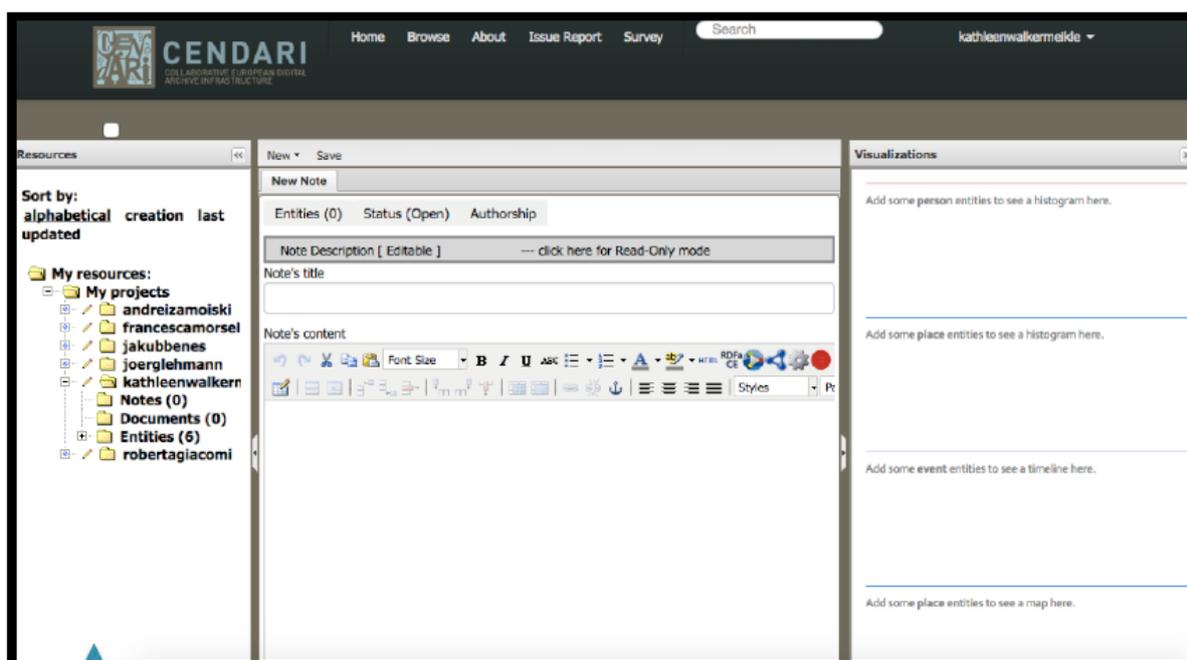
The Note Taking Environment consists of three columns. On the left are your resources, where you can see your notes, documents and related entities. The middle column consists of your working space. If you click on a file in the left column, it will appear in the middle column and you can either work on it (in 'Edit mode') or read it (in 'Read mode'). The right-hand column has visualisations of contents of your files, such as names and places.

2.3.1 1. Creating a new project

Establishing new projects

If you want to create a new project, click on the *New*-tab on top of the middle pane of the NTE's main window; select *project* from the drop-down menu and click it. You are directed to a page where you have to provide for a name of this new project. Please be aware that you will be shown an error message if a project with the identical name already exists. After having provided the name of the project, click *Save*. Afterwards this project can be shared with every user available in the list of usernames on the project edit page; the latter can be reached by clicking on the pencil in front of the project name.

Aliases for projects

If you like, you can add a new name for a project. Go to the project edit page by clicking on the pencil in front of the project name. Scroll down on this page until you reach the very bottom. You will find there the label *New Alias:* and beneath it a field where you can provide for a new name (alias) of this project. Afterwards click on the *Add* button below the field for the new name. The alias is provided and a link containing this alias of the project is being displayed.

Publishing your project

If you want to make your project publicly accessible – i.e. enable every user of the internet to access it without prior log-in – you have to go to the project edit page by clicking on the pencil in front of the project name. Scroll down on this page until you reach the bottom. You will find there the label *Is public:* and beneath it a check box. If you tick it and click on *Save* afterwards, the project will be displayed in the *Public projects* folder available at the bottom of the resources panel (west pane).

2.3.2 2. Create and edit content

Create a Note

To create a new note, click *New* in the Menu Bar and then *Note*. Write your file's title into the field *Note's title*. Write some content in the main body of the note (this can be a short as one word such as *test*, as a note without content cannot be saved). Click on *Save* (a green bar should appear), and you can continue working. Please be aware that you will be shown an error message if a note with the identical name already exists.

Creating a table of contents

To structure your note, type the headings and elements of your note into the text. Then mark each heading one by one and choose the appropriate heading from the editor's drop-down menu. Keep to the correct order: heading 2 follows heading 1 and so on. Click on *Save* and a table of contents will be created. You can view it in the *Read mode*. For example, many of the CENDARI ARGs have the following table of contents:

- Title, Authors, Abstract
- Introduction
- Major Research Questions
- Primary Sources
- Secondary sources
- Additional sources

Creating a table

Go to the point in the text where you plan to insert a table. Click on the *Insert → Edit Table* button (second row, far left of the editor). This will bring up a pop-up menu in which you can select how many rows and columns are needed, in which position the table should be placed (for example, to the right or left of the text). Click on *Insert* to create your table. If your cursor is over the table, you can select other options from the editor (second row) such as *Insert row before* or *Insert row after*. For borders and other options on your table, cluck again on the *Insert → Edit Table* button and choose the *Advanced* table. In this drop-down menu you can select the border in the line *frame*, select the colour and then click on *Update*. Save again and chose *Read mode* and all of the selected options will be displayed.

Saving Work

Please save all of your work and log yourself out after finishing a session. It is a good idea to regularly save work in the NTE.

HTML

If you have experience with editing in HTML Source Code, you can click on the *html* button in the first row of the editor. In the pop-up window you can edit code and save it with *Update*.

Image Viewer

The image viewer is meant to transcribe scans; it can be used after having established a document in your project. The image viewer is to be found on the bottom of the middle column (split screen). You can use it if you click on the `Scans' tab and upload and select one of the image presented there by clicking on it. Afterwards this image is presented in the Image Viewer below. You may need to scroll down and open the *Viewers* tab. The image can also be viewed in a separate browser tab, by clicking on the expand button (on the top right corner of the *Image* viewer pane). Via the *Scans*-tab several multimedia file formats can be uploaded: png, jpeg, tif, tiff, mp3, avi, doc, pdf.

Faceted Search

On top of the main page of the Note Taking Environment NTE there is a search box. By typing a query in the search field, results from within the Note Taking Environment, from dbpedia entries and from TRAME, a database for medieval sources will be shown. There is a check box underneath the *include TRAME* checkbox, called *Search Current Project* for limiting the search to the current project. On the results page, the faceted search options are available. Please do filter the results according to your needs.

2.3.3 3. Insert functions

Inserting text

You can simply type text into the editor as in any word processing programme (using the buttons in the first row of the editor). You can also copy and paste text from another text file or from a wiki. You can layout your text by marking it and choosing the layout options in the first line of the editor. At the moment no images are accepted in the Note Taking Environment, although this is due to the software being in a development phase and this feature will soon be available. If you have copied and pasted text into the editing window, you can automatically detect links in the text by clicking on the big red button in the top row of the editor (called *markcreativework.desc*). You can also insert links manually by selecting a section of text, and then clicking on the *chain* button in the editing menu (called *Insert → Edit Link*).

Insert images

You can add images to your notes by moving the cursor to the place in the text where you want to insert the picture, and then click on the small icon with the tree (in the second line of the menu of the editor, to the very left). A window opens; to the right of *Image URL* there is an icon *Browse*. By clicking it you can upload images from your computer. In the tab *Appearance* you can tell the system to put the image to the right or left side and to display it smaller than the original size (f.ex. you put the value 300 into the left box and keep the *Constrain Proportions* box ticked, then the image will be displayed smaller - in this case 300px, which is around 10cm). The NTE will also provide users with the possibility to upload images in a separate folder, under *Documents*. To add a document go to *New* and click on *Document*. Then type in the description of the document you want to add and click 'Save'. The description will become the title of your document. Next you can add your document by going to the `scan' tab at the top of the page and choosing the right file to upload. After uploading the scan, save your document again. In the near future, it will be possible to use the viewer function at the bottom of the page, allowing you to examine your scans closely and for example to work on a transcription in the NTE.

**Insert Footnotes/References**

Go to the end of your text. If there is no reference section there, create it first by marking it as you did when creating headings. Click on the button with an anchor and provide a name of the anchor. Make sure that this is either one word, words with no spaces in between, or words with hyphens or underscores. Click *Insert*. Then go to the section in the text which you wish to link to the footnote. Put the number of the footnote in brackets (i.e. [1]), highlight the number and then click on the chain symbol (*Insert → Edit Link*). Select the named anchor from the *Anchors* drop-down menu. Click *Insert* and then *Save*. In the Read mode the link will work. You can do the same vice versa so that the readers will be taken back to the point in the text from the reference.

Screenshot of the Editor bar

2.3.4 4. Search functions

The Faceted Search within the CENDARI infrastructure can be accessed through the Search bar in the main window in the top right corner of the Note Taking Environment.
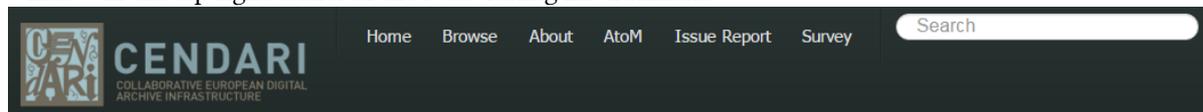
The search interface

After having typed a search term into the search box, the user will be present the main search interface in a new tab.

Users can type precise combinations of search terms by using the following symbols: + signifies AND; I signifies OR; - negates a single token; " " can be put around a phrase; * can be put as a placeholder at the end of a term for prefix; the symbols (and) signify precedence. Users can thus combine search terms and symbols according to their needs. Example: The combination of prisoners+of+war will render more precise search terms than the three words prisoners of war.

Using the facets

On the left side of the screen, the facets of the Faceted Search are being presented to the user. The user can narrow the results of his initial search by filtering them through the facets. Any facet can be opened by clicking the plus symbol. Afterwards the user can select the filter he wants to choose by clicking on it. In the screenshot below the user can choose whether the results from dbpedia, from the CENDARI repository (everything available there), results from the Note Taking Environment, or results from archives (only results in the international standard format Encoded Archival Description) should be displayed to him.

The same applies to the other filters like f.ex. Artifact (results being a note or a document within the Note Taking Environment), Date, Format, Person etc. Filters can also be applied subsequently in order to narrow search results.

Finally, a user can determine whether or not to include the results of the medieval database TRAME or whether or not to limit the search to the current project from where the search was started. This is being done by ticking the check box in the appropriate field.

In order to return to the start, a user can click the reset button on top of the Facets panel.

Facets (reset)

Saving the results of a search

Generally, the results of a search will be displayed to the user as a list of links (displayed in blue with a check box in front of it). Users are provided with a link to the main repository and, if applicable, a link to the data provider. Beneath each link a snippet from the content can be found, i.e. the beginning of the information provided for the result.

Users can choose the results they consider relevant for their search by ticking the check box in front of the result.

☑ **John Coombe papers**

GB 206 Liddle Collection EP 012 John Coombe papers

A user can also decide to select all the results displayed on a page, or none of them, by clicking one of the fields displayed below.

Select All   Select None

After having selected all the results she is interested in, the user can copy them to his clipboard and f.ex. paste them into a note or a .doc-file.

Copy Selection to Clipboard

Viewing a result

A single search result can be viewed by clicking on the link. Depending on the result, this can be a dbpedia page, a file or a note or document in the Note Taking Environment, or, if applicable, a link to the provider.

An example for a result from the German Bundesarchiv; the archival description is stored within the CENDARI repository:

2.3.5 5. Named Entity Recognition and Disambiguation (NERD)

CENDARI provides for a multi-language Named Entity Recognition and Disambiguation service. This service locates and classifies entities in the text provided in notes and documents such as the names of persons, places, organizations, events. Using the service is quite simple: After having provided a text in a note or document, simply press the *Smiley* button in the toolbar of the editor. The service will process the text and find the possible entities contained. The results are being displayed in the text: Recognized entities receive a colored underlining. Please save the results by pressing the *Save* button. The recognized entities will be shown in the resources tree in the west panel. Afterwards you can go through your text and resolve entities manually or assign the appropriate entity class if they were not recognized correctly.

2.3.6 6. Entity Management

Tagging of entities

Highlight the entity you want to add to your resource. Then right-click with your mouse and choose from the context menu the appropriate entity, which can be a person's name, a place, an event or an organisation. Then click on *Save*. Your entity will be listed in the *entities* section on the left column, and will shown in the column on a right. You can also view the entity by clicking on the *entities* tab in your working space (top left-hand side). Tagging entities allows them to be recognized by the system and linked to the available database. You can add an explanatory description to your entity if you right-click with your mouse, as a drop-down menu will appear, offering different options (for example *son of*).

Resolving of entities

If you want your entities to be tagged right-click the entity with your mouse and choose from the context menu the appropriate entity, as mentioned above. Then *Save*. To resolve it, you can either click on the tab *Entities* in your working space section or choose the *Entities* section in the left column. After selected and clicking on the

entity, you have the option of a *Resolve* field. In this field you need to provide a link. Open a new tab in your Chrome browser and search for the entity you are looking for in dbpedia. If the entity you are looking for is simple – for example Paris –, the easiest way is to guess the page's URL by typing it after http://dbpedia.org/page/ (i.e. http://dbpedia.org/page/Paris). Click on the Enter bar of your keyboard, and press *Save*. The resolved entity will then be shown on a map to your right. If the desired entity is a complex entity (for example a transcribed name), the easiest solution is to go to the corresponding Wikipedia page choose the correct entry and then copy and paste the link into the field *Resolve* in the NTE. Afterwards replace the beginning of the link *http:// en.wikipedia.org/wiki* with *http://dbpedia.org/ resource/* → *Save*. In the right-hand column the number of all related documents and notes will be shown, and you see these documents by clicking on the tabs in the middle of your working space (*Related notes*, *Related documents*, *Related entities*).
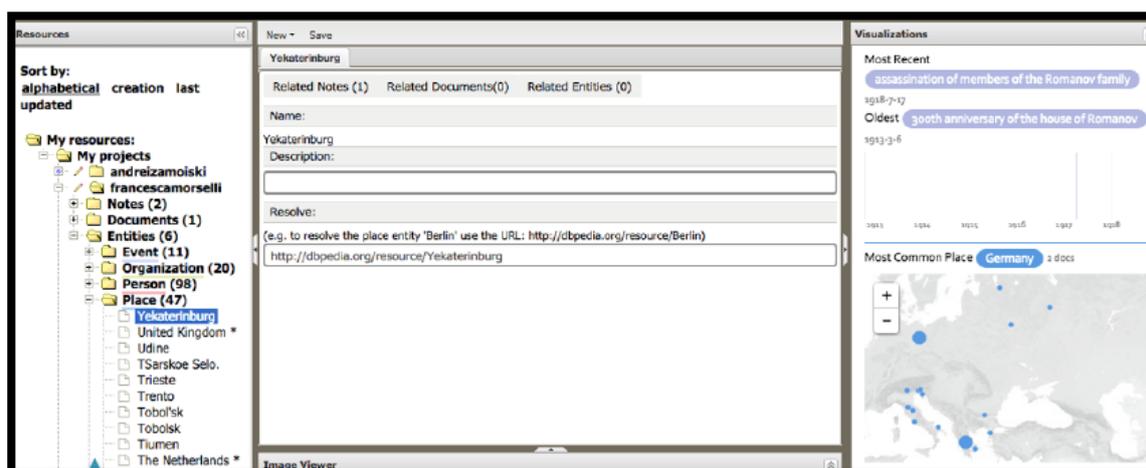
Screenshot (after clicking and resolving an entity).

Auto-completion for the resolution of entities

If you hover with the cursor over an entity, its colour changes to orange or yellow. If you click on it, a window opens where you can resolve or delete the entity (*Edit Schema.org Entity*); you can also change the entity type here. In order to have events resolved, you should provide an entity name in the first field. Over there an auto-completion service has been provided to help in the entity resolution process. If you type the first few characters of the entity name in the *Resolve* field, the NTE will suggest some dbpedia entries and fetch the url of the selected suggestion automatically. If the service does not find what you are looking for, you can always copy-paste a dbpedia link in the form http://dbpedia.org/resource/ENTITY. Another way to resolve entities is to go to the *entities* section in the west panel. There is also an auto-completion service to help in the entity resolution process has been provided.

Manual Resolution of Events

Provide a date for the event between square brackets in this format `mm/dd/yyyy', that is month, then day followed by the year. For example, in a note or document, tag *Battle of Albuera [05/16/1811]* as an event. Then, press save. The NTE will reload the page and show the date of this battle on the timeline. To verify this, you can open the events folder from the resources tree (west panel) and locate the entity *Battle of Albuera [05/16/1811]*. Hovering over this event will highlight its date in the timeline. Note that resolved event entities do not have an asterisk (*) next to their names in the resources tree. If you do not provide a date between brackets [] or not in the correct format during the tagging process, the date will not be recognized and the entity will not be resolved.

Automatic Resolution of Events

Create an event entity from a document or a note as before. You do not need to provide a date between brackets. Open the event you created (e.g.:menuselection:*battle of Wilkomierz*) from the resources tree. In the *Resolve* field, type the first few letters of the event and pick a suitable result from the suggested list. Press *save*. When resolving events automatically, the user has an option to check the date for an event from dbpedia after having found the event via the suggestions or after having provided the dbpedia link. This is done by clicking the *Search for dates* button in the entity form. If a date has been found then the user has the option to keep or discard this date. The NTE will try to fetch the date automatically from dbpedia. If a date exists and if its format is processable, then the entity will be resolved and its date assigned to the timeline. If there is no available date from dbpedia or the

retrieved format is not in a recognisable format, a warning message is issued. This is a potential candidate for manual resolution (as above). Please note that events which have lasted for more than one day are not being displayed on the timeline. Also, hovering over the label for the date field shows a tooltip describing the date formats to be used for tagging dates in notes, documents and transcripts (i.e. %d/%m/%Y and %m/%d/%Y)

Clustering

Aliases are different names for the same entities. For example, the person *Lawrence of Arabia*, *T.E. Lawrence* and *Thomas Edward Lawrence* is the same individual. Similarly the terms *First World War*, *Great War* and *World War One* represent the same event. Another example would be different spellings for the same place in different languages and scripts, such as *Vilnius*, *Вильнюс*, *Вільня* or *ווילנע*. So that the system knows that these different names point to the same entity, users can group them into a cluster. This can be done in two ways. First, after having marked up these as entities in a note or document as an entity, go to the list of resources in the left panel, and open the appropriate entity. Then resolve all of the entities by giving them the same dbpedia URL. For the T.E. Lawrence example, all three entities should be resolved with the URL http://dbpedia.org/resource/T._E._Lawrence. Entities with the same dbpedia URL will be automatically merged into an alias cluster. The second possible option for giving aliases for all entities involves resolving only one of the entities with the appropriate URL. After doing this, click on *Save → Create a new cluster containing this topic* which should be in the middle panel under the URL's field. Please note that you have to enable pop-up windows in your browser to access this page. A new page will open with the entity for which you want to create a cluster on top, followed by a list of all the other entities below. In order to select all the entities which should be grouped in the cluster (for example *Vilnius*, *Вильнюс*, etc.), you have to tick the boxes in front of these entities. Remember to click *Save* afterwards. Both operations will result in the clustering of all related entities in a folder in the list of resources in the left-hand panel. You can edit clusters there if desired. Access the edit page of a cluster by opening one of the entity pages of the cluster and click on the cluster name link in the middle panel, which is just below the field for the URL. Subsequently after opening the page, you can edit the name of the cluster or add and remove entities from the cluster. Always click on *Save* after having selected the appropriate boxes.

2.3.7 7. Versioning

Versioning and Reversion History

Open a note or a document, next to the save button you will find the *version history* button. Click on this button and you will see the latest versions for this document or note. You can check each of the old versions of these documents by clicking on the links.

Reversion History: the date of each history version of a document or a note is available now. Open a document, click on *Version History* button from the menu bar. The revision history is displayed in a new page. Select an old version, the content of this document version will be opened in the editor. You will notice next to *Document Description* the date (between brackets) when this version was created. The *Revert to this version* link next to *Document's description* allows for reverting the document to the selected history version. Note that revision history also works for notes.

2.3.8 8. Collaboration and Sharing

2.3.9 9. Reporting issues

If something does not work or if the system crashes, you can click on the button *Issue report* on the very top of the page.

2.3.10 10. Frequently Asked Questions

How can I invite other persons to work on an ARG together?

What happens with my feedback given via the issue report or the survey?

What happens to my data in several years? Will all my work be stored and how long will it be stored?

2.4 CENDARI Archival Directory

The CENDARI Archival directory is implemented through AtoM, which stands for Access to Memory. It is a web-based, open source application for standards-based archival description and access in a multilingual, multi-repository environment. This software was chosen because it adheres to international standards and as a user friendly tool to produce archival descriptions.

2.4.1 1. Starting your search

To start with a simple search like *manuscript*, enter your search terms in the search field. You can browse the results by the proposed list (*Archival descriptions*, *Archival institutions*, *Subjects* and *Places*) or you narrow your search by facets. You will get a list of records and record creators that match your search criteria. These search results also show which archival institutions hold the proper records.

The lists provided show all records from all archives, according to CENDARIs priority to provide transnational access and multilingual content. Narrow down your search results by using the facets provided on the left side. Choose from the proposed level of description (collections, fonds, etc.); archival institutions with relevant holdings; creators (if it differs from archival institutions); place (choose by country); and by subject (however, not all archival descriptions contain subject access points). Choose keywords carefully, and spell them correctly. When you're deciding what keywords to put in the search field, try to choose words that are likely to appear on the record you're looking for.

You can narrow down search results for complex searches by using the **Advanced Search** page. You will reach it by clicking on the search field or you go directly to the Advanced Search page. Enter your search terms in the field *Search*. Narrow it down by using the dropdown menu to the right of the *Any field* button, choosing the fields that should be searched (*title*, *reference code*, etc.). You can always add new criteria and combine them with logical connectives (*or*, *and*, *not*). You can filter your search results to find exactly what you are searching. Search filters on the left side of the Advanced search page will help you to narrow down your search. Use such search filters as *repository*, *level of description* and etc. If you are searching for sources on a certain topic, please bear in mind that the archival holdings described in AtoM cannot be seen as being exhaustive, even if there are several dozen entries available for an archive. Archival descriptions have been produced according to the needs of the topical Archival Research Guides and not with the aim to provide a comprehensive overview of all sources available in an archive. Please, generally search independently from the language within the collections. If you are looking for results in certain language, type keywords in this language (and in the appropriate script, f.ex. cyrillic) into the search box.

2.4.2 2. Disclaimer

Archival institutions have been chosen for description according to the following criteria, according to CENDARIs case studies:

World War I

- to have all countries which have participated in the First World War covered

- to give special attention to East European and South East European countries and to describe so-called *hidden archives* which did not dispose of a digital representation so far

- to describe archival institutions with significant holdings with priority to central national archives, national military archives, national war museums

- archives with a wide scope of digitized finding aids

Medieval Culture

- to cover most Western countries, including USA and Australia

- to give special attention to East European and South East European countries

- to cover collections of interest for the scientific community according to their relevance in the academic bibliography

- to cover archival institutions with an advanced digitization level

Descriptions of archival holdings: Please bear in mind that the archival holdings described in the CENDARI Archival Directory cannot be seen as being exhaustive, even if there are several dozen entries available for an archive. Archival descriptions have been produced according to the needs of the topical Archival Research Guides and not with the aim to provide a comprehensive overview of all sources available in an archive. The selection on sources depends on the research question and therefore exhaustive descriptions of archival sources can never be reached. Since descriptions of archival holdings can be added by users registered as contributors, and since they can be made publicly available by users registered as editors, the Archival Directory can be seen as work in progress, never complete, and depending on the contribution of users.

2.4.3 3. Frequently Asked Questions

Are users allowed to add descriptions and what kind of descriptions are users being allowed to add?

Users which are not logged in (reseachers) have *view-only* access to the CENDARI Archival Directory. Users are allowed to add archival descriptions once they have received a DARIAH-account, are authenticated and have been granted contributor's rights, i.e. to read, create, edit archival descriptions. Users with contributor's rights are not allowed to delete anything or to create archival institutions. Archival descriptions can be added according to the needs of these users while observing the standards of the International Council on Archives. If users want to have their archival descriptions being published and thus freely accessible for everybody, they should ask a user with editor rights (usually their mentor) to check for the quality of the descriptions and afterwards change the status of the archival description from *draft* to *published*. If users want archival institutions to be created, they should ask their mentor.

What kind of agreements does CENDARI have with Archives and Libraries regarding the sharing of records?

CENDARI does not have any specific agreements with Cultural Heritage Institutions. The Archival Directory AtoM will be open for Cultural Heritage Institutions a s well as for users to add descriptions according to their needs while observing the standards of the International Council on Archives. If users want to publish archival descriptions and the content of finding aids within CENDARIs Archival Directory, they should first talk to the archives whether or not they are allowed to do so.

Will there be any translating of the institutional and archival descriptions be done?

Unfortunately, no translations will be provided. Users are kindly requested to generally search independently from the language within the collections. If you are looking for results in certain language, type keywords in this language (and in the appropriate script, f.ex. cyrillic) into the search box.

Will there be a browse by period function?

Currently, it is foreseen to provide a browse by period function; it will be enabled only later. The reason why it is not yet there is simply that the majority of archival holdings described have been produced in the same period, i.e. 1914-1918.

Why do labels provided in the Subjects and Places categories not correspond to all the archival holdings available?

All the entries have been produced manually. Therefore it depended on the decision of the describing person to add subjects and places labels where she or he decided this to be appropriate. Therefore, the categories *Subjects* and *Places* correspond not to archival description, only to a part of the descriptions produced by the MM teams and WW1 historians. In many cases these categories are not filled at all. If users think these descriptions would be helpful, they can fill in these fields by themselves, where appropriate. This can be done by entering an archival description after having logged in, click on *edit* at the foot of the page, click on *Access points*, start writing the appropriate Subject or Place in the field given (note: you have to wait for some one or two seconds after having started typing since the subjects and places already available in the system show up), and after having chosen or typed manually the appropriate subject or place click on *Save* at the foot of the page.

Why is there no comprehensive search tree in a hierarchical order?

Hierarchised searches need pre-fixed taxonomies, on which the possible users of the Archival Directory should have agreed upon. In a system like the one provided over here, such a pre-fixed taxonomy cannot be foreseen. Instead, CENDARI chose a more open approach where users are free to choose their subjects and places according

to their needs without having them pre-ordered in a hierarchical tree. Users can even add subjects and places which have not yet added to the Archival Directory (compare here the FAQ ``Why do labels provided in the Subjects and Places categories ...'' on how to do this).

Why is there no hierarchy in the archival descriptions according to the level of fonds, subfonds, items etc.?

The software of the Archival Directory does not allow to link fonds, collections, items etc. to each other if they are being produced separately, i.e. by several contributors. That is why to arrange a comprehensive search tree in a hierarchical order is problematic.

2.4.4 4. Manual for the use of the Archival Directory AtoM

Within the CENDARI project, an electronic Archive Directory of sources for medieval and modern history was set up to establish descriptions of holdings, collections, records, and single items. The software used was AtoM or Access to Memory. It aims to be user-friendly and checks the quality of the descriptions thanks to mandatory fields. Explanations guide users through the process of creating an archival description. It stores the content in the standardized format Encoded Archival Description (EAD), which can be later exported. This archival standard is used in almost all European and many non-European countries. It is approved by the International Council on Archives (ICA), the institution that collaborated in the development of the AtoM software. AtoM is free web-based application, and as a multilingual multi-repository interface and database for archival descriptions, all elements in AtoM can be translated into multiple languages (using the translation interface).

The CENDARI project has a strong transnational focus, and one of its goals was to include many archives and institutions which are little known or rarely used by researchers. It is focused on archives and libraries containing documentation for historical research on the Middle Ages and World War One. The CENDARI Archival Directory AtoM can be accessed both directly from the website and indirectly via the link in the Note-Taking Environment (NTE). The CENDARI Archival Directory AtoM contains entries for archival descriptions and institutions. A basic FAQ is available to assist users who are browsing and searching for archival descriptions and collections. Information is also provided on creating and editing entries.

What is an archival description?

For all historical research, any historical source needs to have indications on the context in which it was created and distributed. Moreover, it should have information regarding its history – its use, storage and authenticity. Usually, cultural heritage institutions (archives, libraries and museums) possess the legal authority to give an account on the credibility and reliability of source material they are responsible for. Their catalogues and finding aids contain the necessary information on the subject of single holdings and record groups, integrating essential explanations on how to use any given inventory and how to interpret its contents.

AtoM uses the standardized format Encoded Archival Description (EAD) to present information about an archival unit, which can range from a huge collection to an individual item in an archival institution. The International Council on Archives (ICA) defines an International Standard Archival Description (General) (ISAD(G)) as "The creation of an accurate representation of a unit of description and its component parts, if any, by capturing, analyzing, organizing and recording information that serves to identify, manage, locate and explain archival materials and the context and records systems which produced it. This term also describes the products of the process."

This standard can contain twenty-six data elements, of which six are mandatory in the creation of a compliant archival description. Following ICA rules, the obligatory elements are:

- **Reference code**: identifies uniquely the unit of description and provides a link to the description that represents it. To assist international standardisation, it needs to contain the country code, the repository code and a specific local reference code, control number, or other unique identifier.

- **Title**: this is the name of the unit of description. It can either be the formal title or an abridged title (as long as the latter follows national conventions and the rules of multilevel descriptions.

- **Date(s)**: this records the date(s) of the unit of description. It can be either a single date or a range of dates, as appropriate. The date(s) is either the data when the records were collected or when they were first created.

- **Level of Description**: identifies the level of arrangement of the unit of description. For example, it is a single item, a file, sub-series, series, sub-fonds or fonds.

- **Extent and Medium of the Unit of Description (quantity, bulk, size)**: this element identifies and describes the physical or logical extent of the unit in Arabic numerals (and unit of measurement) and the medium of the unit of description. For example: one folder, containing 73 items.

- **Name of creator**: identifies the creator or creators of the described unit.

- **Repository**: This is not an official mandatory field in the ISAD(G) standard, but it is a mandatory field in the standard developed and used in the Archival Directory.

Here is an **example** of these mandatory fields in an archival description in the Archival Directory:

Reference code: GB Add MS 49683-49962

Title: The Balfour Papers

Date(s): 1868-1936

Level of Description: Collection

Extent and medium: 280 items

Name of creator: Arthur James Balfour

Repository: The British Library

This archival description also contains non-mandatory elements such as Scope and Content, Language of material, Finding aids, Name Access Points and Archivist's note.

The ISAD(G) standard is not obligatory and it does not reflect practice in all archival institutions. When the idea of an international descriptive standard was first put forward, there was debate on whether such a standard was really necessary. The advent of computerised description, digitisation and digital finding aids that cover multiple archival institutions has strengthened the idea that a standardised methodology for describing archival units is very useful.

The International Council on Archives Committee that developed the ISAD(G) recognised that all archival institutions are very different, and that their arrangement and subsequent archival descriptions will reflect different needs, whether these are organisational, national, or cultural. The ISAD(G) should be used in conjunction with national standards or as a basis for further development of existing national standards.

Useful links:

- International Council on Archives on ISAD(G) (it also includes accepted translations of all of the terms).

- International Council on Archives Committee on Descriptive Standards.

- UNESCO Archives Portal (provides information on archival institutions all over the world).

- International Standard Archival Authority Record. For Corporate Bodies, Persons and Families http://www.icacds.org.uk/eng/ISAAR%28CPF%292ed.pdf.

- ISDIAH: International Standard For Describing Institutions with Archival Holdings https://wiki.ica-atom.org/RS-3#4.7.

What is an archival institution?

Every archival description in the Archival Directory is linked to an archival institution entry. An archival institution has been defined as "an institution holding legal and physical custody of noncurrent documentary materials determined to have permanent or continuing value. Archives and manuscript repositories are archival institutions." This includes all institutions that manage, preserve and/or store historical source materials. The International Council on Archives has developed an international standard to describe archival institutions: ISDIAH (International Standard for Describing Institutions with Archival Holdings). It has also developed a standard to describe corporate bodies, persons and families as units within an archival descriptive system: ISAAR (International Archival Authority Record for Corporate Bodies, Persons and Families). Detailed documentation on these standards can be found on the ICA's website.

The holdings can cover a diverse range of materials, including:

- **Documentary material, which can be divided into:**

1. Printed material: most printed material is not kept in archival institutions but in libraries.For various reasons, such as rarity, historic importance, limited print run, and age, such material may be preserved in archival institutions (for example: pamphlets)

2. Manuscripts: All archival material that is written by hand or typewritten. Can be bound or unbound, and cover a wide variety of genres (letters, handwritten books, etc)

- Audiovisual material: sound recordings, which might appear in a variety of formats

- Iconographic material: this can cover maps, plans, drawings, sketches, paintings, postcards, posters, and advertising material.

- Objects

Unpublished archival material can be found both in archives and special collections in libraries. For example, both might contain bound and unbound manuscript material. Along with libraries and archives, museums are also considered to be memory institutions. While they may not hold documentary material, they hold objects which are historical sources.

Regarding medieval archival materials, due to various historical factors related to the dispersion of materials, individual bound manuscripts are usually found in special collections in libraries (example), while documentary material, often unbound, (for example, court rolls, accounts, charters, etc) are found in archives. Regardless, both are considered technically archival institutions in the Archival Directory.

The Archival Directory was set up as a component of the CENDARI research infrastructure in order to pursue two goals. Firstly, to give visibility and accessibility to many archival institutions and their holdings which did not have a digital representation along with enabling the CENDARI team to establish entries for archival descriptions and institutions according to the methodology and criteria established by the team members. And secondly, to allow further users of the CENDARI virtual research environment to easily create archival descriptions with the help of a user-friendly tool and thus broaden the basis of digitally available descriptions of archival content.

How the CENDARI Archival Directory AtoM can be used

The CENDARI Archival Directory AtoM can be a very useful tool for historians researching in any archival institution. As AtoM provides an intuitive user interface, its archival forms can be used as a sort of registry, where the researcher can record information on the archival institutions and collections. However, as the CENDARI AtoM is a web-based application, the institution where the historian is researching needs to have an internet connection.

An increasing number of archival institutions offer researchers electronic finding aids. These describe specific collections, including information related to both the archive's structure and content. Common descriptive fields are creator, date of creation, medium and extent, languages and system of arrangement. Other collection guides can be internal archival guides (focusing on collections of special interest) or shared spreadsheets. If none of these sources are available, it is best to request advice directly from the archivist, who would have detailed knowledge of the collections available in the archive as well as their structure and order. The archivist might also give the researcher access to resources that are only available off-line or after submitting a written request. It is highly recommended to contact the archivist when researching in a cultural heritage institution and before creating archival descriptions in the CENDARI Archival Directory.

There is a distinction between modern and medieval sources in the *Archival Description* field in AtoM. Modern sources are usually grouped in collections or fonds, as most modern sources such as documents or registries are grouped in large numbers. Medieval sources are usually grouped in collections or fonds when they involve documentary material and usually found in archives. Medieval sources of authored texts, in manuscript format, are usually found in special collections in specialist libraries. An example of the former can be found in the ARG Dispersed Medieval Monastic Archives in Florence, which documents manuscript materials in archives. Meanwhile the ARG Italian Vernacular Bibles details individual manuscripts in codex format in libraries (Special Collections).

Apart from using AtoM to search and create archival descriptions and collections, the CENDARI Notes Taking Environment (NTE) allows researchers to upload images such as scans or photographs of the documents. These can be analysed in a private virtual working space, with the possibility to annotate and describe them. This allows the user to access, in a digital working environment, both the reference to the examined archive and collections, and the photographed items, which can be transcribed and tagged in the NTE.

This is possible because all the data – manually established data in AtoM as well as data provided by cultural heritage institutions – were brought together in the repository. In other words: All the data contained in AtoM is migrated to the repository on a regular basis. The automatic indexation of the named entities contained in the archival descriptions, like names, places and dates, enhances the search possibilities offered to the users in the Note Taking Environment. As an example: the results of a search for person in the repository will show all related hits from both manually and automatically ingested data, along with all hits from the ontologies. These search results would included the people, places and dates related to the individual subject of the search.

Remember that in uploading photographs, scans or digitized objects into the CENDARI Notes Taking Environment, users have to stick to the terms and conditions provided by the CENDARI project; moreover, it is good practice to discuss this intention with the archivist, and if necessary to request a formal permission to the archive.

Please note that CENDARI does not have any specific agreements with any cultural heritage institution (archives, libraries and museums). Cultural Heritage Institutions and users can add descriptions according to their needs to the Archival Directory, as long as they observe International Council on Archives standards. If users wish to publish their archival descriptions and their search results from Archival Directory, they should discuss relevant permissions with the archive in question. Similarly, it is strongly recommended that there should be a clear agreement between the archive and the researcher before any material is uploaded to the Note Taking Environment.

Adding archival institutions to the CENDARI Archival Directory

Important information can be found in the descriptions of archival institutions. These descriptions consist of seven fields (called *areas*):

Identity area Contact area Description area Access area Services area Control area Access points

The Identity area lists the institution's formal names. The Contact area provides contact information as well as street addresses, locations of reading rooms, etc. Official websites of archival institutions can give further information, and the researcher can also directly contact the archivists (via mail, telephone call or on site in the reading room). This information is available in the Access areas, for example information on *Opening times*.

The **Description area** of archival institutions contains information on the history of the archival institution ("History"), their most important collections ("Archival and other holdings"), information on finding aids, guides and publications (in the corresponding fields). Additionally researchers can find information about the geographical and cultural context of the holdings and on the archival institution's records management and collecting policy

(in the corresponding field).

Within the **Services area** a user can learn more about research services provided by the institution. If a user aims at providing information in this field, he can write down information over here like, for example, whether you can speak English with the archivist, who can advise you on-site, computer facilities, etc. In the field *reproduction services* information can be provided on facilities are available to researchers such as the possibility of photocopying or scanning materials, fee information, internal archival regulations, etc.

The sub-area **Public area** provides useful information to facilitate research and the time spent in the archival institution. For example, if the internet connection is free or there is a charge, existence of cafeterias or restaurants, or if cash machines are available inside or nearby the institution.

Due to lack of existing data, many archival institutions have empty fields in their descriptions in the Archival Directory. CENDARI users are welcome to contribute updated information on the archival institutions to the aforementioned fields.

The Archival Directory AtoM provides users with the possibility to edit and publish new archival institutions and archival descriptions. However, these possibilities are limited by the users' rights. This has been done in order to ensure the quality of the archival descriptions. A user can only add archival descriptions once they have a DARIAH-account, been authenticated and granted contributors' rights. Users who have not logged into the CENDARI Archival Directory ("researchers") have view-only access. Contributors' rights allow a user to read, create, or edit archival descriptions. However they do not allow the user the ability to delete any archival description nor create a record for a new archival institution. This can only be done by users with editors' rights. A user with editor rights might be the mentor (a historian, librarian, archivist or someone with similar experience) of a user with contributors' rights, who is not yet familiar with the software AtoM or with the establishment of archival descriptions.

A user with editor rights can create records for archival institutions. To add a new archival institution, click on the *Add button* (the plus sign symbol) in the top right hand side of the Archival Directory AtoM. When you start filling the fields explanations will appear below each field. Information should be placed into the field in the accordance with the international standard ISDIAH, and users are requested to stick to them.

Adding archival descriptions

Archival descriptions should be added in accordance with the International Council on Archives ISAD(G) standard. In order to their archival descriptions freely accessible to all Archival Directory viewers, the user would need to contact a user with editor rights. The user with editor rights will check the quality of the submitted archival descriptions and then can change the archival descriptions' status from *draft* to *published.*

To create an archival description, click on the *Add* button on the top right (represented by a plus symbol) and select *Archival Description.* The user can name their archival description, which is initially designated 'Untitled' at the top of the page until changed by the user.

The areas are:

Identity Area, which includes:

- Identifier [mandatory] This should be a specific local reference code, control number, or other unique identifier. The country and repository code will be automatically added from the linked repository record to form a full reference code. Alternative identifiers can be added if desired (http://www.worldcat.org/webservices/registry/xsl/identifiers/).

Title [mandatory]:

- **This can either be a formal title or a concise title according to the rules of multilevel description and national conventi**

    - Date(s) [mandatory]
    - Level of description [mandatory]
    - Add new child levels [optional]
    - Extent and medium [mandatory]

Context Area, which includes:

- Name of creator [mandatory]

- Repository [optional]

- Archival History [optional]

- Immediate Source of Acquisition or Transfer [optional]

Content and Structural Area

Conditions of Access and Use area

Allied Materials Area

Notes area

Access points

Description Control area

Rights Area

Administrator Area

Links to digitized objects

Multilinguality of the Archival Directory

Many archival institutions in the Directory were described with names in Russian, Ukrainian, Lithuanian, etc. The original name can be seen in the identity area (the field "Authorized form of name"). The translated name of the same institution (usually in English) is available in the next field "Parallel form(s) of name". This field can also contain different forms of authorized names. For example, there can be provided abbreviations and acronyms of institutional names. For example, in a search, typing the Latin letter form "GARF" will bring up as a result the State Archive of the Russian Federation. Typing in the Cyrillic "ГАРФ", will bring up the same institution (Государственный архив Российской Федерации)

In the title field for archival collection descriptions in some Eastern European archival institutions, the name of the archival unit is available in both English and Russian. For example:

- Fonds 601 - Nicholas II of Russia / Николай II

When adding new descriptions, it is best to enter the original language the institution's proper title and the descriptions of their archival holdings, along with inputting an English translation of the titles. This will help future searches.

Using the Notes area

The *Notes area* allows the user to add annotations according to their individual research interests. Users can inform the CENDARI community about important details regarding the archival collections. For example, the researcher's experience of working with this collection, on-site or personal discoveries which could be useful and interesting for other users, etc. Every user can open a new note for his annotations by clicking on "Add new". The notes should be accurate and understandable for other researchers. Editors monitoring the content may not accept notes if they do not correspond to the relevant archival descriptions.
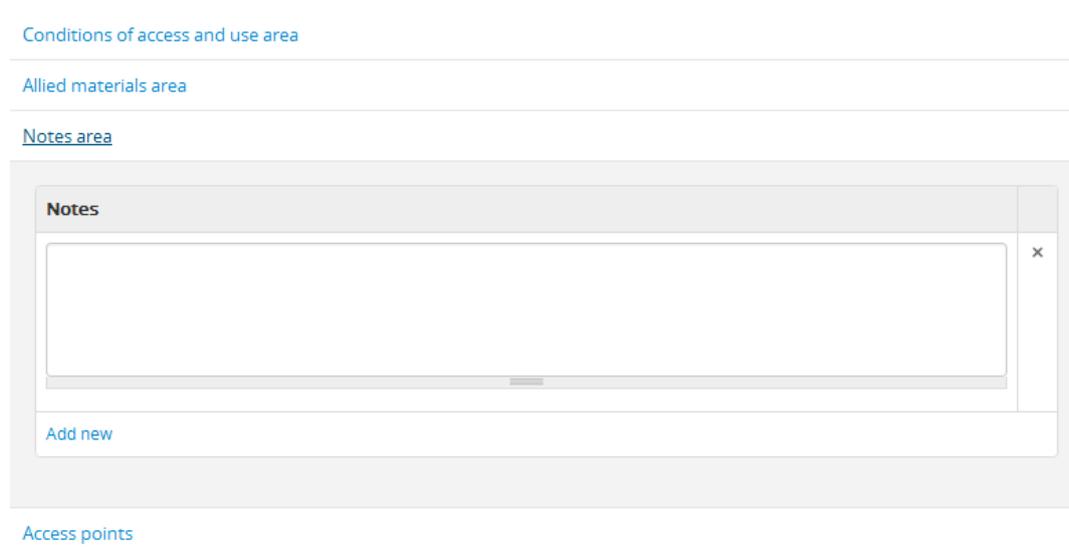
Uniting dispersed sources of the same institutions or persons

The dispersion of sources poses a major problem for historical research. Different tools are available, so that extensive travel might now always be necessary in order to find and describe sources. There on-line archival catalogues, databases, blogs, email expert advice, and projects such as as CENDARI. Archival sources can be found in unexpected places, for example, among the papers of secret services, state organizations or private papers, which might be in the process of being described and digitised. Some previously top-secret files may be declassified.

To discover archival collections, the researcher should be familiar with the history of its creators. Help can be provided by both primary sources (published collections) and secondary sources (encyclopedic entries, thematic monographs and articles).

The case of Jewish archives: the American JDC archive

Jewish sources are scattered across different countries and various cultural heritage institutions (Jewish archives, national and regional archives, national libraries, museums, etc.). Due to wars, migrations and deportations the Jewish archives were dispersed and even split among different countries. For example, the records of the American Jewish Joint Distribution Committee (JDC) can be found in different institutions of Eastern Europe, America, and Israel. This organization operated in over 90 countries. Their main task was to bring urgent aid to Jewish communities as well as to individuals in need, including refugees, orphans, POWs and other groups. The activity of the Jewish Joint Distribution Committee was very important after 1914, when the JDC was involved in charitable activities in the warring countries and later in the newly established states of Central and Eastern Europe (Poland, Lithuania, Hungary and others). In the 1920s the JDC supported programs such as medical assistance, training and resettlement of Jews from shtetls, in the Soviet republics until its activity was stopped by the Soviet regime during the years of the Great Terror (1937-1938).

- American Jewish Joint Distribution Committee, New York Archives
- The case of political parties, organizations and private papers

Many sources might be dispersed or have been relocated. For example, leaders of political parties were mobile people, and their private papers as well as the records of their organizations wandered with them, while another part remained in place. Nowadays many sources are preserved as separate collections or as parts of other collections within various archival holdings. For example, records of the Russian revolutionary leader Leon Trotsky (Leo Trotzki) are held among archival holdings in Russian, Dutch, American, and other archives.

Internationaal Instituut voor Sociale Geschiedenis

- Collection Kautsky-Archiv - Karl Kautsky Papers

Russian State Archive of Socio-Political History

- Fonds 325 - Trotsky (Bronshtein) Lev Davidovich / Троцкий Лев Давидович

Houghton Library, Harvard College Library, Harvard University

- Collection MS Russ 13.1 - Leon Trotsky exile papers

Information on dispersed collections can be consolidated in the Archival Directory:

- To virtually consolidate some collections, use the section *Allied materials area* within an archival description.

- In the field *Existence and location of originals*, the user can mention other archival institutions as well as libraries and museums where original records are located. This field can also provide details on whether the originals do not exist anymore or if their present location is unknown. For example, for records lost or destroyed during the Second World War. It would be best if the user provides control numbers.

- In the field *Existence and location of copies*, the user can add information on the location of copies. Nowadays many archival institutions run programs to exchange their data. National archives in many countries are keen to receive original records or copies from foreign institutions on some important events relevant for national history or on outstanding compatriots, or the history of the diaspora. Some institutions, like the Central Archive for the History of the Jewish People (CAHJP) or the Yad Vashem archives focus on searching for data on the history of Jewish people in foreign archival holdings.

- In the field *Related units of description*, the user can add information on the units of description from the same repository or from another relevant one. Some archival collections (fonds) have up to now no proper inventories or finding aids that contain archival descriptions. In this way users can enlarge the common knowledge by contributing data.

- The field *Related descriptions* demonstrates the relationship between archival descriptions in CENDARI Archival Directory. The user can choose from proposed variants of archival descriptions or add own ones.

- In the field *Publication notes* the user can enter information on publications, studies or analysis of the sources. Information on published facsimiles or transcriptions can also be provided.

Another way to unite some collections or other units of descriptions (series, files, or even parts of descriptions) is to provide labels and adding new terms.

Users can add labels (subjects and/or places) to any archival description in the Archival Directory AtoM. This can be done both with entries that they have created themselves or those already contained in the system. This can be done by logging in and clicking on the *edit* button at the foot of the appropriate archival description page. Then click on *Access points* and enter the appropriate Subject or Place into the field. After either having typed manually or having chosen from the listed subjects and/or places, click on *Save* at the foot of the page. The categories *Subjects* and *Places* only partially correspond to the descriptions created the Archival Directory. This is due to all entries being entered manually, where it was personal decision of each compiler whether to add a relevant subject or place label to the entries.

The labels produced by users can help to identify proper collections even though they are placed in different countries and in the various holdings. For example, the subject label *Cinema* will list different collections relevant to the history of the film industry and to records of different institutions which were responsible for propaganda and the production of movies during WWI, personal papers of actors, filmmakers, as well as copies of newsreels and movies.

When producing new terms for archival descriptions, the user should consider that similar terms may have been previously created and that it is unnecessary to duplicate them.

The terms must be generally accepted by the scholarly community. Slang, unquotable and misspelled words are unacceptable. In the case of places, only real or previously existing places should be added (cities, states, villages, regions, etc.). It is obvious that imaginary and fictional places (such as Middle-earth from Tolkien's world) are not appropriate.

2.5 TRAME

TRAME (Texts and Manuscript Transmission of the Middle Ages in Europe) is a research infrastructure project for the development and interoperability of web databases about medieval manuscript tradition.
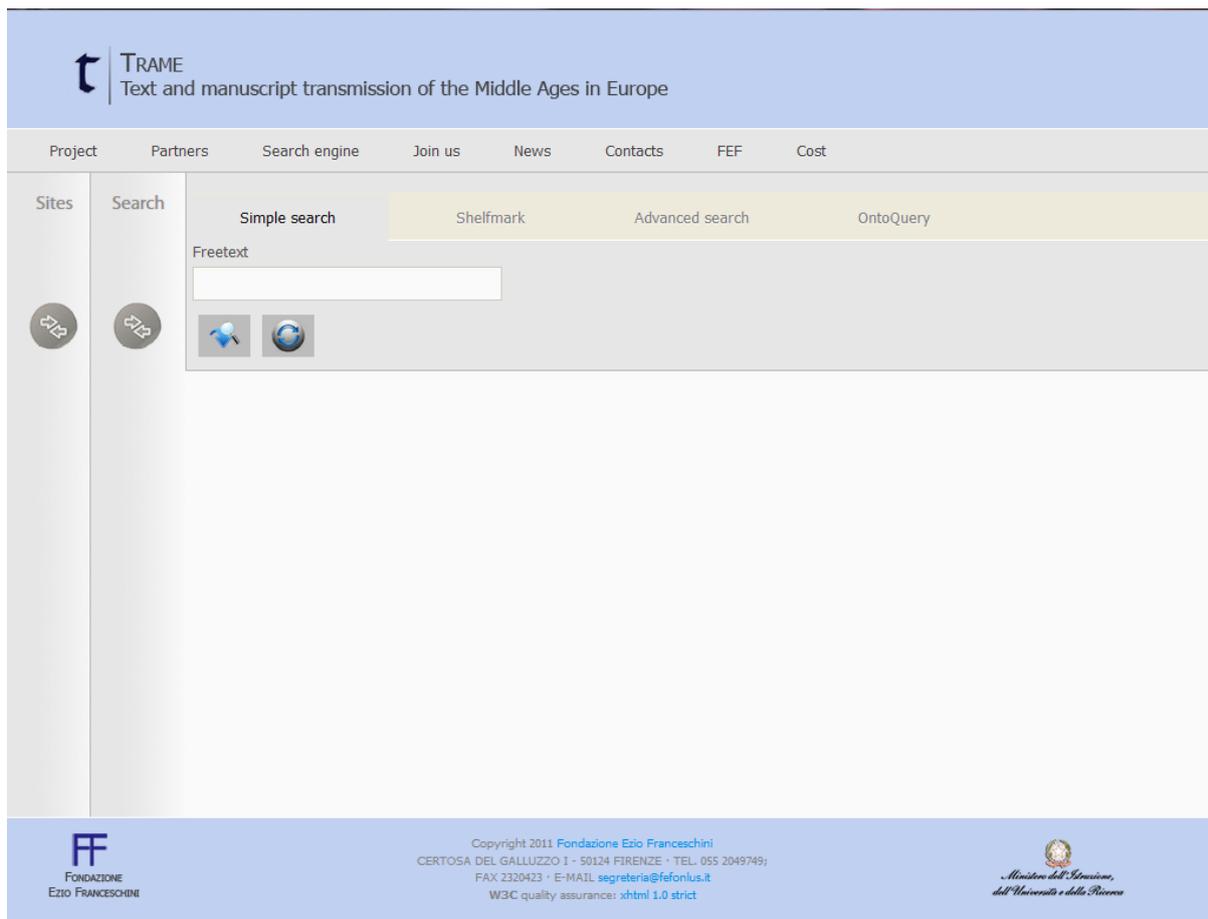
2.5.1 1. What is it?

TRAME is a metasearch tool for medieval manuscripts, hosted by the Fondazione Ezio Franceschini and SISMEL. It aims to assist users in finding primary sources for the study of medieval culture. Originally developed in 2011 to allow a combined search on a group of different Italian repositories and databases, it has increasingly widened its scope to become a global research infrastructure for all scholars of the Middle Ages. Its main aim is to build a research infrastructure that focuses on promoting interoperability and fostering discoverability among the different digital resources available in the medieval digital ecosystem. Since 2014 TRAME has focused primarily on extending the meta-search approach to other web resources, using the user's interaction with the research tool in an attempt to define a medieval manuscripts ontology, and redesigning the portal with the aim of improving the accessibility and usability of the site. Currently it implements a number of features (including simple search, shelf-mark search, and advanced search) on more than 80 selected scholarly digital resources devoted to western medieval manuscripts, authors, and texts across the EU and USA. These resources include digital libraries, research databases and other projects from leading research institutions. TRAME is a research tool rooted in the international medieval scholarly community and an ongoing collaborative international effort. Its development is in line with the Memorandum of Understanding of the COST Action IS1005 "Medieval Europe Medieval Cultures and Technological Resources", representing 260 researchers coming from 39 leading institutions in 24 countries across the European Union.

2.5.2 2. What can I find there?

TRAME allows the user to search simultaneously in library catalogues, project databases and research portals. It combines online resources that are inside the TRAME network as well as external sites. The infrastructure combines both:

Bibliographies pertaining to manuscripts, e.g.

- MEL – Medioevo latino. Bollettino bibliografico della cultura europea da Boezio e Erasmo (secoli VI-XV)
- MEM – Medioevo musicale. Bollettino bibliografico della musica medievale
- TLION mss – Bibliografia dei manoscritti citati in rivista
- BibMan – Bibliografia dei manoscritti in alfabeto latino conservati in Italia

Repertories of texts and manuscripts, e.g.

- LIO – Repertorio della lirica italiana delle Origini
- BAI – Biblioteca Agiografica Italiana
- MAFRA – Repertorio dei manoscritti gallo-romanzi esemplati in Italia
- MAGIS – Manoscritti agiografici dell'Italia del Sud
- IRHT – Jonas : Répertoire des textes et des manuscrits médiévaux d'oc et d'oïl - JONAS
- Ramon Llull Database
- Repertorium Biblicum Medii Aevii
- BHL Biblioteca Hagiographica Latina manuscrita

Catalogues of manuscripts (with images), e.g.

- CODEX – Inventario dei manoscritti medievali della Toscana
- MDI – Manoscritti datati d'Italia
- Manuscriptorium
- Enluminures

Repertories pertaining to the history of traditions, e.g.

- TETRA – La trasmissione dei testi latini del Medioevo
- TLION – Tradizione della letteratura italiana online

Bio-bibliographies

- BISLAM – Bibliotheca Scriptorum Latinorum Medii Recentiorisque Aevi
- CALMA – Compendium Auctorum Latinorum Medii Aevi (500-1500)

Library catalogues, e.g.

- Beinecke Digital Collections
- Bodleian Library
- Bibliothèque nationale de France
- Trinity College
- Bayerische Staatsbibliothek
- Hill Museum & Manuscript Library
- Munchen (BSB) - MDZ
- Early Manuscripts at Oxford University
- Bibliothèque Municipale de Lyon
- Mazarinum - Les collections numeriques de la Bibliotheque Mazarine - Mazarinum

2.5.3 3. How to use TRAME

Site Selection

The introductory Video to TRAME: Site selection has been published on Youtube.

Simple Search

The introductory Video to *TRAME: simple search* has been published on Youtube.

Shelf-mark search

The introductory Video to *TRAME: Shelf-mark search* has been published on Youtube.

Advanced search

The introductory Video to *TRAME: advanced search* has been published on Youtube.

2.5.4 4. TRAME version 2

A completely revised TRAME user interface is currently in development. Not all of the search functions are already operational. In the new version of TRAME 2 you can perform searches as a normal user or as a registered user. The main difference is that as registered user you can build and save your own searches in 'search packages' (meaning your own selection of databases). You will have the opportunity to makes these packages public, so others can use them, or to keep them private. The search options remain the same as in the old version: a simple or freetext search, shelf-mark and advanced search. At this at this stage the only search fully working is the freetext.

Video User interface TRAME 2

The introductory Video to *TRAME 2: Interface* has been published on Youtube.

2.5.5 5. Technical Background

TRAME's development has been influenced by changes regarding the nature of information available in the WWW. TRAME has developed from a basic meta-search approach towards an attempt to establish a Medieval Semantic Knowledge base, by using custom applications for information collection and integration (i.e.: web crawler, data miner).

The application is written in OO-PHP, the design follows the MVC Pattern, the RDBMS is MySql and the front-end combines Xhtml and Javascript.

The search engine scans a set of sources for searched query terms and retrieves links to provide a wide range of information, including simple references, detailed manuscript record, and full-text transcriptions.

Currently, it is possible to perform queries by freetext, shelf-mark, author, title, date, copyist or incipit, on more than 80 selected scholarly digital resources across the EU and the USA.

Advantages of TRAME's search of remote resources:

- TRAME has light and flexible infrastructure, as both data indexes are not stored in a central database. Actually no information is stored except for a few technical metadata.

- TRAME will send the user query across a vast number of repositories and present the results in a single list.

- TRAME can send a user query across a number or remote systems over HTTP protocol, it's also supporting OAI-PMH on selected repositories and (if available) specific APIs

- The results will be divided in groups according to their provenance or type (the original data provider)

- All search results found by TRAME's meta-search engine are accessible via the original provider's web site, with their own policies and licensing methods

- A user query is sent simultaneously over a wide number of connected systems in order to collect a unique list of results. The search results will have all the information needed to identify each individual manuscript, such as localization (City, Library and Holding), shelf-mark and the link to the actual digital resource (URI: uniform resource identifier)

To learn more about the technical background of TRAME and TRAME 2 please have a look at the source code documentation.

2.6 PINEAPPLE

PINEAPPLE is a simple browser for the CENDARI semantic repository.
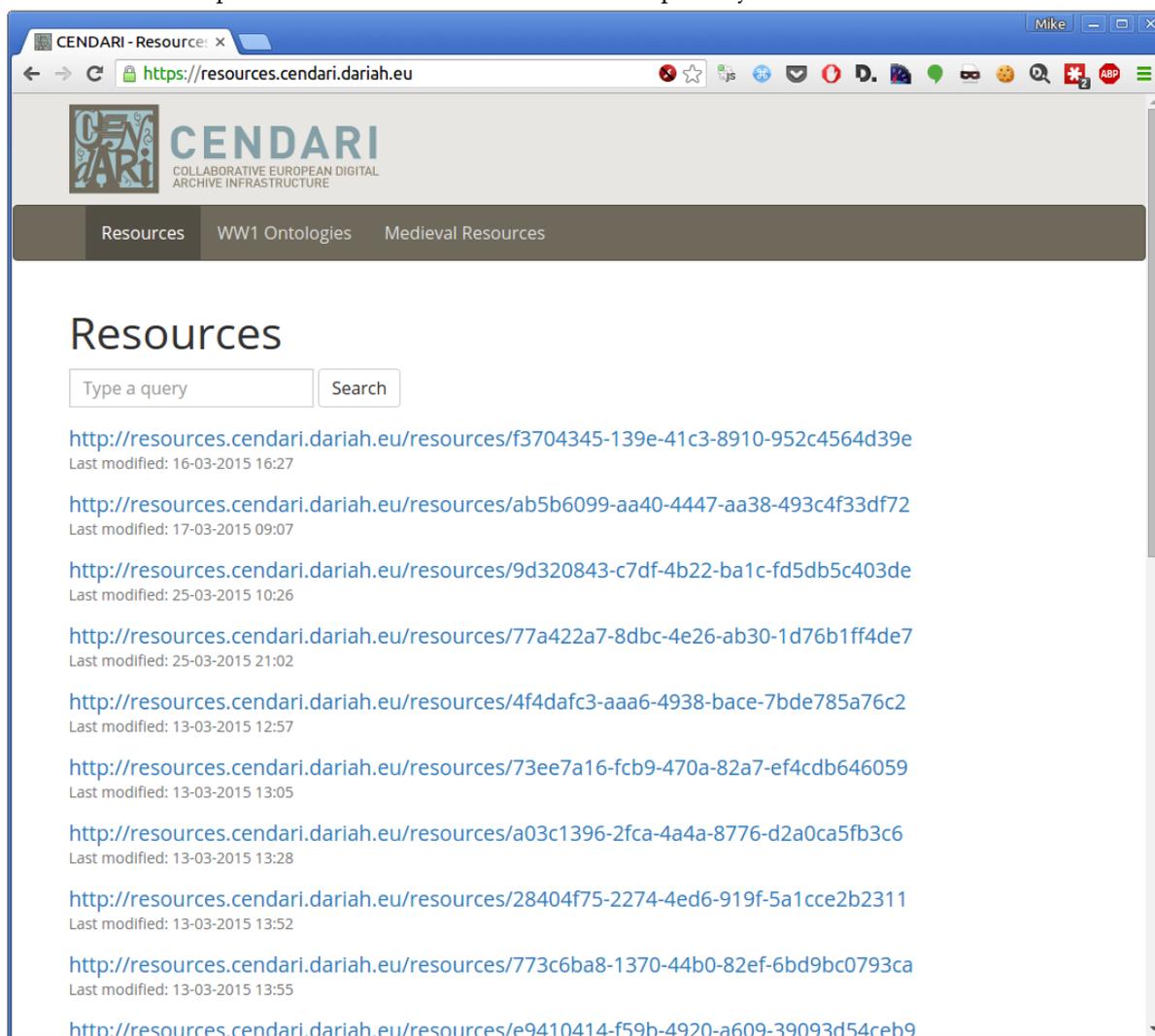
2.6.1 1. What is it?

PINEAPPLE provides a browsing interface for three types of data in CENDARI's semantic repository:

- WW1 archival data
- WW1 ontologies
- Medieval manuscripts

2.6.2 2. How to use PINEAPPLE

Resources search

Supplying a textual query will filter resources by the contents of their plain text field.
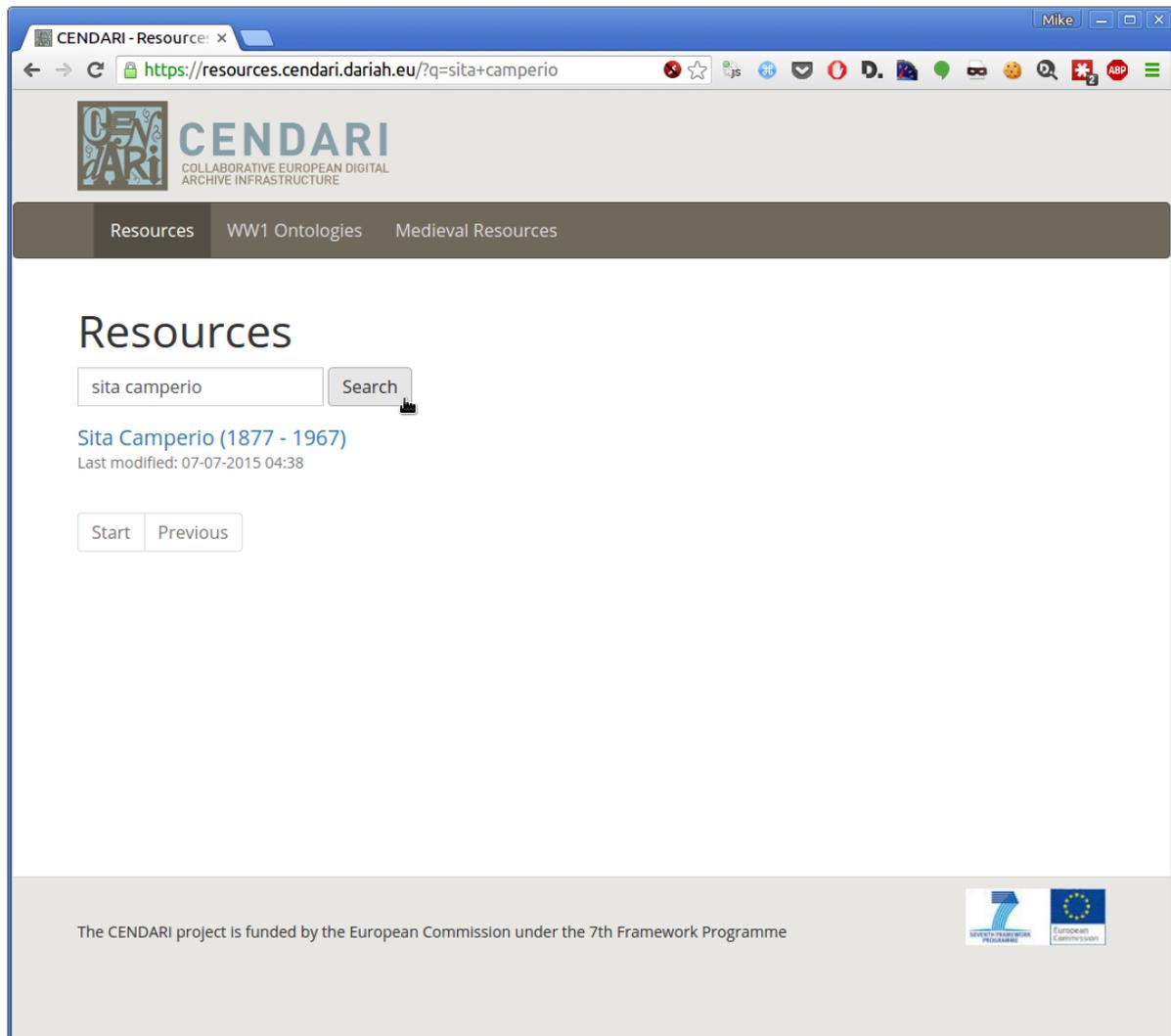
The resource detail view displays its plain text content and semantically-related people, organisations, locations, and events.

Other resources which share access points are also listed below.

WW1 ontologies search

The ontology view allows exploring the WW1 ontologies used by (or developed for) CENDARI. Ontology items can be faceted by item type (person, place, event, concept etc) and the ontology to which they belong. Providing a textual query filtered the ontology items by their (English) preferred label.

The ontology item detail view displays both other ontology items related by semantic links, and CENDARI resources which contain a textual match to the item's preferred label.

The detail view also displays information about the ontology to which the item belongs.

Browsing medieval resources

The medieval resources view allows browsing and faceted search for medieval manuscripts.

Results can be narrowed by filtering on facets such as manuscript author, organisation, and the religious order of the author/organisation.

This section describes how the components are set up and connected.

3.1 Installation

The central CENDARI infrastructure as hosted by UGOE is managed entirely through puppet.
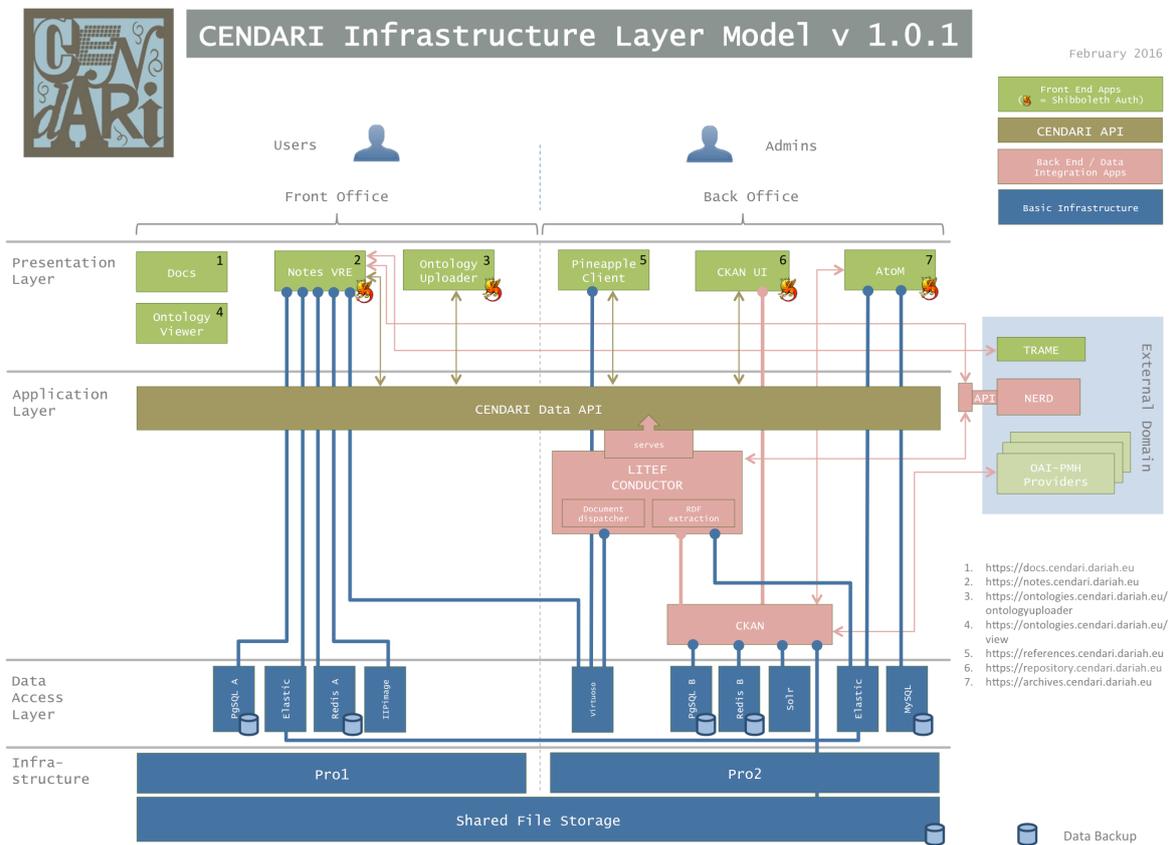
Fig. 3.1: The CENDARI Infrastructure Layer Model

All of the configuration management is based on the the Infrastructure Layer Model, see *Figure*. The setup is split into two Roles, the Front Office comprises the applications intended for all users, while the Back Office consists of those applications used by power users and administrative staff.

3.1.1 Prerequisites

The infrastructure set up by UGOE splits Back Office and Front Office into two servers, which each exist twice. The two setups provide a staging environment to facilitate components integration and a production environment. The duplication allows developers to immediately test the latest versions without impacting the production servers.

CENDARI uses Ubuntu 14.04 on virtualized x86_64 servers. The components have not been tested on other distributions.

The following packages are installed on all servers:

```
package {
  'fabric':            ensure => installed;
  'g++':               ensure => installed;
  'libsasl2-dev':      ensure => installed;
  'nfs-common':        ensure => installed;
  'nodejs':            ensure => installed;
  'nodejs-legacy':     ensure => installed;
  'npm':               ensure => installed;
  'openjdk-7-jdk':     ensure => installed;
  'python-pip':        ensure => installed;
  'python-virtualenv': ensure => installed;
}
```

In addition, some node packages are globally available

```
npm install -g less@1.3.3
npm install -g node-sass@3.2.0
```

3.1.2 Database and Search Indexers

As indicated by the model, individual PostgreSQL databases are used for data storage by both the Notes VRE and the CKAN repository application. Furthermore, AtoM uses MySQL. An elasticsearch cluster is set up between the servers within each environment.

Applications can therefore always access databases and query elasticsearch locally on standard ports.

3.1.3 Back Office

CKAN

Start by following the instructions to set up CKAN, version 2.2.1.

To install CKAN, the following requirements should be met:

```
package{
  'libredis-perl':            ensure => present;
  'postgresql-contrib-9.3':   ensure => present;
  'postgresql-postgis':       ensure => present;
  'postgresql-server':        ensure => present;
  'postgresql-server-dev-9.3': ensure => present;
  'python-pastescript':       ensure => present;
  'redis-server':             ensure => present;
  'solr-tomcat':              ensure => present;
  'tomcat6':                  ensure => present;
}
```

You need two databases for CKAN and the DataStore.

Make sure the config files /etc/solr/conf/schema.xml, /etc/ckan/apache.wsgi, /etc/ckan/production.ini, /etc/ckan/who.ini have the right contents.

```
mkdir -p /var/www/ckan
cd /var/www/ckan
rm -rf *
virtualenv --no-site-packages /var/www/ckan
source /var/www/ckan/bin/activate
pip install -e "git+https://github.com/ckan/ckan.git@ckan-2.2.2#egg=ckan"
pip install -r /var/www/ckan/src/ckan/requirements.txt
deactivate
source /var/www/ckan/bin/activate
cd /var/www/ckan/src/ckan
../../bin/python setup.py install

# BUILD EXTENSIONS

git clone https://github.com/CENDARI/ckanext-dariahshibboleth.git /var/www/ckan/src/ckanext-dariahshibboleth
cd /var/www/ckan/src/ckanext-dariahshibboleth
../../bin/python setup.py install
git clone https://github.com/CENDARI/ckanext-cendari.git /var/www/ckan/src/ckanext-cendari
cd /var/www/ckan/src/ckanext-cendari
../../bin/python setup.py install
git clone https://github.com/okfn/ckanext-archiver.git /var/www/ckan/src/ckanext-archiver
pip install -r /var/www/ckan/src/ckanext-archiver/pip-requirements.txt
cd /var/www/ckan/src/ckanext-archiver
../../bin/python setup.py install
git clone https://github.com/okfn/ckanext-datastorer.git /var/www/ckan/src/ckanext-datastorer
pip install -r /var/www/ckan/src/ckanext-datastorer/pip-requirements.txt
cd /var/www/ckan/src/ckanext-datastorer
../../bin/python setup.py install
git clone https://github.com/okfn/ckanext-harvest.git /var/www/ckan/src/ckanext-harvest
pip install -r /var/www/ckan/src/ckanext-harvest/pip-requirements.txt
cd /var/www/ckan/src/ckanext-harvest
../../bin/python setup.py install
cd /var/www/ckan/src
pip install -e git://github.com/kata-csc/ckanext-kata.git@1.2#egg=ckanext-kata
git clone https://github.com/kata-csc/ckanext-oaipmh.git /var/www/ckan/src/ckanext-oaipmh
cd /var/www/ckan/src/ckanext-oaipmh
../../bin/python setup.py install
```

The following plugins are required:

- DARIAH Shibboleth authentication plugin for CKAN, see also CKAN plugin ``dariahshibboleth''

- CKAN Archiver Extension

- CKAN Datastorer Extension

- Remote harvesting extension for CKAN

- Kata extension for CKAN, version 1.2

- OAI PMH extension to CKAN

Litef Conductor

The Litef Conductor can be built and installed from its source in the litef-conductor repository on GitHub.

To build it, Java 1.7.x, Scala 2.10.4 and SBT are required. Running

```
sbt compile assembly
```

will compile a fat jar, that can be deployed to the server.

Additionally, Litef requires a configuration file /etc/litef/application.conf

```
logger.scala.slick.session=DEBUG

litef {
  namespace = "http://localhost:42042/v1/"
}

litef.ckan {
  home          = "https://host/ckan"
  namespace     = "https://host/ckan/api/3/"
  apiKey        = "***********"

  localStoragePrefix = "/data/resources"

  postgres {
    url      = "jdbc:postgresql://localhost:5432/***********"
    driver   = "org.postgresql.Driver"
    username = "***********"
    password = "***********"
  }
}

litef.virtuoso {
  url = "jdbc:virtuoso://localhost:1111"
  username = "dba"
  password = "***********"
}

litef.nerd {
  namespace = "http://traces1.saclay.inria.fr:8090/"
}

litef.elastic {
  namespace = "http://localhost:9200/"
}

litef.indexer {
  localStoragePrefix = "/var/lib/litef/default/resources"
}

litef.conductor {
  fileSizeLimit = 10000
  plugins = "conductor.plugins.DocumentIndexerPlugin,conductor.plugins.NerdPlugin,conductor.plugins.VirtuosoFee
}

akka {
  log-dead-letters = 10
  log-dead-letters-during-shutdown = on
  loglevel = "INFO"
}

spray {
  can.server {
    request-timeout = 240s
    idle-timeout = 300s
  }
  can.client {
    request-timeout = 240s
    idle-timeout = 300s
  }

  routing {
    file-chunking-threshold-size = 10k
    file-chunking-chunk-size = 5k
```

```
  }
}

logger.scala.slick=ERROR
logger.scala.slick.session=ERROR
```

Litef requires a dedicated CKAN sysadmin, thus after installing CKAN create that sysadmin:

```
source /var/www/ckan/bin/activate
cd /var/www/ckan/src/ckan
paster sysadmin add litef -c /etc/ckan/production.ini
```

Add the new sysadmin's API key to `/etc/litef/application.conf`. Litef needs modifications to the CKAN database.

```sql
CREATE VIEW litef_ckan_resource_view AS
SELECT resource.id,
 resource.resource_group_id,
 resource.url,
 resource.format,
 resource.description,
 resource."position",
 resource.revision_id,
 resource.hash,
 resource.state,
 resource.extras,
 resource.name,
 resource.resource_type,
 resource.mimetype,
 resource.mimetype_inner,
 resource.size,
 resource.last_modified,
 resource.cache_url,
 resource.cache_last_updated,
 resource.webstore_url,
 resource.webstore_last_updated,
 resource.created,
 GREATEST(resource.last_modified, resource.created) AS modified,
 ( SELECT rg.package_id
    FROM resource_group rg
    WHERE (rg.id = resource.resource_group_id)) AS package_id,
 resource.url_type
FROM resource;

ALTER TABLE public.litef_ckan_resource_view OWNER TO ckanuser;

CREATE VIEW litef_ckan_group_resource_view AS
SELECT p.owner_org AS group_id,
 r.id,
 r.resource_group_id,
 r.url,
 r.format,
 r.description,
 r."position",
 r.revision_id,
 r.hash,
 r.state,
 r.extras,
 r.name,
 r.resource_type,
 r.mimetype,
 r.mimetype_inner,
 r.size,
```

```
 r.last_modified,
 r.cache_url,
 r.cache_last_updated,
 r.webstore_url,
 r.webstore_last_updated,
 r.created,
 r.modified,
 p.id AS package_id,
 r.url_type
FROM ((package p
   JOIN resource_group rg ON ((p.id = rg.package_id)))
   JOIN litef_ckan_resource_view r ON ((r.resource_group_id = rg.id)));

ALTER TABLE public.litef_ckan_group_resource_view OWNER TO ckanuser;

CREATE VIEW litef_ckan_user_group_role_view AS
SELECT "user".id AS user_id,
 "user".apikey AS user_apikey,
 member.capacity AS group_role,
 member.group_id,
 member.id,
 member.state
FROM (member
   JOIN "user" ON ((member.table_id = "user".id)))
WHERE (member.table_name = 'user'::text);

ALTER TABLE public.litef_ckan_user_group_role_view OWNER TO ckanuser;

CREATE VIEW litef_ckan_user_group_role_view__through_package_group AS
SELECT "user".id AS user_id,
 "user".apikey AS user_apikey,
 object_role.role AS group_role,
 role.group_id
FROM (("user" "user"
   JOIN user_object_role object_role ON (("user".id = object_role.user_id)))
   JOIN group_role role ON ((object_role.id = role.user_object_role_id)));

ALTER TABLE public.litef_ckan_user_group_role_view__through_package_group OWNER TO ckanuser;

CREATE VIEW litef_ckan_user_package_role_view AS
SELECT "user".id AS user_id,
 "user".apikey AS user_apikey,
 object_role.role AS package_role,
 role.package_id
FROM (("user" "user"
   JOIN user_object_role object_role ON (("user".id = object_role.user_id)))
   JOIN package_role role ON ((object_role.id = role.user_object_role_id)));

ALTER TABLE public.litef_ckan_user_package_role_view OWNER TO ckanuser;

CREATE VIEW litef_ckan_user_package_view AS
SELECT group_r.user_id,
 group_r.user_apikey,
 group_r.group_role AS package_role,
 package.id,
 package.name,
 package.title,
 package.version,
 package.url,
 package.notes,
 package.license_id,
 package.revision_id,
 package.author,
```

```
package.author_email,
package.maintainer,
package.maintainer_email,
package.state,
package.type,
package.owner_org,
package.private
FROM ((litef_ckan_user_group_role_view__through_package_group group_r
  JOIN "group" ON ((group_r.group_id = "group".id)))
  JOIN package ON ((package.owner_org = "group".id)));

ALTER TABLE public.litef_ckan_user_package_view OWNER TO ckanuser;

CREATE TABLE litef_processed_resource (
 resource_id character varying(254) NOT NULL,
 last_processed timestamp without time zone,
 attachment character varying NOT NULL);

ALTER TABLE public.litef_processed_resource OWNER TO ckanuser;

CREATE TABLE litef_resource_attachment (
 resource_id character varying(254) NOT NULL,
 format character varying(254) NOT NULL,
 created timestamp without time zone NOT NULL,
 modified timestamp without time zone NOT NULL,
 content text);

ALTER TABLE public.litef_resource_attachment OWNER TO ckanuser;

ALTER TABLE ONLY litef_processed_resource
 ADD CONSTRAINT litef_processed_resource_pkey PRIMARY KEY (resource_id, attachment);

ALTER TABLE ONLY litef_resource_attachment
 ADD CONSTRAINT litef_resource_attachment_pk PRIMARY KEY (resource_id, format);

CREATE INDEX litef_processed_resource_id_last_processed_idx ON litef_processed_resource USING btree (resource_i

CREATE INDEX litef_processed_resource_last_processed_idx ON litef_processed_resource USING btree (last_processe

CREATE INDEX litef_resource_attachment_resource_id_modified_idx ON litef_resource_attachment USING btree (resou
```

Finally, Litef can be started by

```
java -Dfile.encoding=UTF-8 -cp /etc/litef:/opt/litef/litef-conductor.jar core.Rest
```

Data API

The Data API is served by Litef on port 42042 and can be accessed locally. To provide remote access from the Front Office server, create a dedicated user named `apiuser`. When adding the public keys for connecting to its `.ssh/authorized_keys`, it is recommended to prepend options `from="1.2.3.4",no-pty,no-X11-forwarding` for security.

AtoM

- Follow the instructions to install AtoM from CENDARI's AtoM fork on GitHub.

- The fork includes the the following addtional components as git submodules:

    - Shibboleth authentication plugin

    - the AtoM2CKAN sync scripts

Prerequisites

AtoM is a PHP application served through Apache

```
package {'php5-cli': ensure => installed }
```

It relies on **ElasticSearch** and **MySQL**.

Install

Clone the repository's CENDARI-branch with its submodules

```
git clone --branch stable/2.2.x-cendari https://github.com/CENDARI/atom.git
git submodule update --init --recursive --remote
```

than run

```
OLDDIR=$(pwd)
for dir in arDominionPlugin sfCendariThemePlugin; do
  cd plugins/$dir
  make
  cd $OLDDIR
done

chmod -R og+rX .

chown -R www-data data
chown -R www-data log
chown -R www-data cache
chown -R www-data atom2ckan
```

Use the web installer to set up AtoM (fails if config exists).

After an upgrade, make sure to upgrade the database and clear the cache

```
sudo -u www-data php symfony tools:upgrade-sql
sudo -u www-data php symfony cc
```

If necessary, recreate the Elastic index

```
sudo -u www-data php symfony search:populate
```

Configuration

Verify Language and Timezone in `apps/qubit/config/settings.yml` and add the database settings to `config/config.php` and the ElasticSearch settings to `config/search.yml`.

Add the Shibboleth mapping to `apps/qubit/config/app.yml`:

```
all:
  shibboleth_administrator_groups: 'cendari-archives-admins;cendari-admins'
  shibboleth_editor_groups: 'cendari-archives-editors'
  shibboleth_contributor_groups: 'cendari-archives-contributors;cendari-contributors'
  shibboleth_translator_groups: 'cendari-archives-translators'
```

CKAN upload

To regularly upload all config files to CKAN, install atom2ckan by cloning from GitHub and filling in the settings to `complete_atom_to_ckan_config.php`.

Three satellite tables should be provided in the atom database using the following commands:

```
create table harvester_ead(
    atom_ead_id int not null,
    atom_ead_slug varchar(255),
    atom_eag_slug varchar(255),
    repository_resource_id varchar(40),
    sync_date datetime,
    primary key (atom_ead_id)
);


create table harvester_eag(
    atom_ead_id int not null,
    atom_eag_slug varchar(255),
    repository_resource_id varchar(40),
    sync_date datetime,
    primary key (atom_ead_id)
);

create table harvester_date(
    date datetime
);
```

Also there the last table should be populated with one value before which represents the date before CENDARI project was started, like:

```
insert into harvester_date (date) values ('2010-01-01 01:01:01');
```

The following command should be executed by the `www-data` user periodically, i.e. via cron

```
cd /var/www/atom2/atom2ckan && php complete_atom_to_ckan.php
```

E-Mail report   The script `mail_report.php` sends information on the current number of files in AtoM and the number of transferred files to CKAN. It is part of *atom2ckan*, thus simply fill in the settings to `mail_report_config.php`.

The following command should be executed by the `www-data` user periodically (weekly: Wednesday, 14:00), i.e. via cron

```
cd /var/www/atom2/atom2ckan && php mail_report.php
```

Documentation

The official CENDARI is created using Sphinx. It is build from the code repository on GitHub.

Prerequisites

Make sure you have doxphp installed.

To build the CENDARI theme you need bower and grunt:

```
npm install -g bower
npm install -g grunt-cli
```

To build the PDF version, you need LaTeX installed, i.e.

```
package {
  'texlive-latex-recommended': ensure => installed;
  'texlive-latex-extra':       ensure => installed;
  'texlive-fonts-recommended': ensure => installed;
```

```
    'texlive-fonts-extra':        ensure => installed;
}
```

Preparing the build environemnt

To compile the documentation with parts included directly from submodules, make sure to have the lastest revision

```
git submodule update --init --recursive --remote
```

You may want to define the version and iteration:

```
VERSION=$(grep ^version source/conf.py | sed 's/version = '\''\([0-9.]*\)'\''/\1/g')
ITERATION=0
export CENDARI_RELEASE="${VERSION}-${ITERATION}"
```

Enable the inclusion of the submodules

```
export CENDARI_INCLUDE_SUBMODULE_DOCS="YES"
```

Install the requirements into a dedicated Python Vitualenv:

```
virtualenv venv
source venv/bin/activate
pip install -r requirements.txt
pip install -r dependency-requirements.txt
```

Compile the CENDARI theme in `source/sphinx_rtd_theme` by running

```
npm install

grunt exec:bower_update
grunt sass:dev
grunt clean:build
grunt exec:build_sphinx
```

In the Repositories main directory run

```
/bin/bash submodules-import.sh
```

to prepare the imported sources.

Compiling the Documentation

Finally build the documentation in HTML and PDF format.

```
make html
make latexpdf
cp build/latex/CENDARI.pdf build/html/
```

Make sure to serve the content of the `build/html` directory.

The BackOffice consists of the following components, that need to be installed in order:

1. CKAN

2. Litef Conductor

3. AtoM

4. Documentation

3.1.4 Front Office

Notes VRE

Follow the instructions provided with the editorsnotes repository on GitHub.

The following packages are required to compile the Notes VRE on Ubuntu 14.04

```
package {
  'libcurl4-gnutls-dev': ensure => installed;
  'libjpeg-dev':         ensure => installed;
  'libopenjpeg-dev':     ensure => installed;
  'libpq-dev':           ensure => installed;
  'libtiff5-dev':        ensure => installed;
  'libvips-tools':       ensure => installed;
  'libxml2-dev':         ensure => installed;
  'libxslt1-dev':        ensure => installed;
  'python-dev':          ensure => installed;
  'python-vipscc':       ensure => installed;
  'ruby-dev':            ensure => installed;
  'unixodbc-dev':        ensure => installed;
  'zlib1g-dev':          ensure => installed;
  'iipimage-server':     ensure => installed;
  'imagemagick':         ensure => installed;

}
```

After cloning, run

```
fab setup
fab sync_database
```

Install the package `supervisor` and make sure the file `/etc/supervisor/conf.d/notes.conf` contains

```
directory=/var/www/notes
command=/var/www/notes/bin/python manage.py rqworker
stopasgroup=true
stopsignal=QUIT
stderr_logfile=/var/log/notes.err.log
stdout_logfile=/var/log/notes.out.log
```

To `/etc/odbc.ini` add

```
[VOS]
Description = Virtuoso
Driver = /usr/lib/odbc/virtodbcu_r.so
Server = host.name
address = host.name:1111
Port = 1111
UserName = dba
Locale = en.UTF-8
```

The file `/etc/apache2/mods-available/iipsrv.conf` must contain

```
ScriptAlias /iipsrv/ "/usr/lib/iipimage-server/"

<Directory "/iipsrv/">
  AllowOverride None
  Options None
  Order allow,deny
  Allow from all
  AddHandler fcgid-script .fcgi
</Directory>
```

```
# Set our environment variables for the IIP server
FcgidInitialEnv VERBOSITY "1"
FcgidInitialEnv LOGFILE "/var/log/iipsrv.log"
FcgidInitialEnv MAX_IMAGE_CACHE_SIZE "10"
FcgidInitialEnv JPEG_QUALITY "90"
FcgidInitialEnv MAX_CVT "5000"
FcgidInitialEnv FILESYSTEM_PREFIX '/var/www/notes/uploads/'


FcgidIdleTimeout 0
FcgidMaxProcessesPerClass 1
```

Enable the corresponding Apache module with `a2enmod iipsrv`.

Data API endpoint

To provide local access to the CENDARI Data API, create a passphrase-less ssh keypair and distribute it to the Back Office's `apiuser`. Finally use `autossh` to establish port forwarding:

```
autossh -NL 42042:127.0.0.1:42042 -i /path/to/key apiuser@back.office
```

To set up the FrontOffice, you need:

1. Notes VRE

2. Data API endpoint

3.2 Authentication

CENDARI uses the Authentication and Authorisation Infrastructure currently under development by DARIAH-DE [1], the DARIAH AAI. Consequently, all account management is handled externally by and through DARIAH.

The DARIAH AAI provides SAML [2] authentication using the open source solution Shibboleth [3].

3.2.1 Shibboleth explained

Shibboleth authentication provides a procedure to deferr the actual login to a known third party. From the point of view of an application, the login process is deferred to the Shibboleth service provider, the SP, which runs on the same server that hosts the application. The SP in turn requests the user's identity attributes from the DARIAH identity provider, the IdP. Upon receiving the user's identity and authorisation attributes from the DARIAH IdP, the SP provides the attributes to the application via the web- or application server in the form of headers or environment variables. In the case of CENDARI, the application than sends the Shibboleth attributes to the CENDARI Data API, which returns the CENDARI internal session token and username, see below.

From the user's point of view, when trying to log into a CENDARI service, he is redirected to the familiar DARIAH AAI login page, where he performs the authentication. Upon successful completion, he will instantly be redirected back to the original website and automatically logged in there. Apart from the URL changes, the whole process is entirely transparent to the user.

Should the user not have a DARIAH Homeless account, he can directly apply for an account from the DARIAH login page. If he does so using an institutional mail address, the account will be created automatically and upon accepting the the terms of use for both DARIAH and CENDARI, the user will be able to use CENDARI services.

A very extensive and in-depth explanation of Shibboleth and an AAI federation, see below, is provided by the SWITCH online demo [4].

---

[1] DigitAl Research Infrastructure for the Arts and Humanities
[2] Security Assertion Markup Language
[3] Shibboleth
[4] SWITCH AAI Demo

The user's attributes are exchanged between the SP and the IdP using public-key cryptography. They are sent along with the redirects through the user's brower. To achieve this, the CENDARI SP and the DARIAH IdP have pre-shared their public-keys through a manual verification process.

Therefore the DARIAH IdP can trust that the authentication requests have been started by a trusted party, i.e. CENDARI, and disclose the user's private attributes. Even more importantly, the CENDARI SP can validate the user's attributes it receives as authoritative. The application can thus simply trust the identity it receives from the SP. The exchange between the actual application and the SP is handled entirely by the Apache web server.

3.2.2 Enabling Shibboleth authentication

Within CENDARI, all services have been developed or extended with the use of Shibboleth authentication in mind.

There are two ways Shibboleth authentication is used:

1. Access to the application is permitted only after successful authentication.

2. Public access to the application is possible, but login is required to contribute content.

In both cases, the setup is very similar as the Shibboleth authentication is always enforced directly by Apache. Either the whole application or a dedicated Login-Page are protected by asserting

Alternatively, Apache can enforce the Shibboleth authentication directly for a certain URL, i.e. the application can rely on Apache to make sure the user is authenticated through Shibboleth when accessing a certain path on the server

```
# enforcing Shibboleth
<Location /shibboleth-enforced>
  AuthType shibboleth
  ShibRequestSetting requireSession true
  Require shib-attr isMemberOf cendari-users
</Location>
```

in the Apache config. This way, the application simply checks for the presence of Shibboleth attributes and acts accordingly. In the first case of full shibboleth protecttion, the application simply fails to work without Shibboleth.

The second case, where authentication is optional, rewuires the application to override its internal authentication in the presence of Shibboleth attributes. To make this work, the login has to be done on an individual page with a dedicated URL as opposed to a dropdown menu embedded on all pages.

This login URL is than set up to enforce Shibboleth authentication prior to access. If the login page is called, the application checks for the presence of Shibboleth attributes. If they are present, the corresponding user is logged in directly, without actually prompting for a password. In other words, a session is started and the user is immediately directed back to the applications landing page. If the user is not yet know to the application, the account is created from the Shibboleth data on the fly.

If the login page is called but no shibboleth data provided, as may be the case in development, the application handles authentication through its usual internbal mechanism.

In both cases, the application initiates a session using its own session management after successful authentication either by Shibboleth or internal mechanisms.

This solution has been applied to the CKAN repository and the AtoM Archival directory through dedicated plugins plugins, as well as CENDARI's own NTE.

Closing the Session

To end a Shibboleth session, i.e. log out the current user, the application has to end it's own user session and finally initiate a redirect to

```
https://<fqdn>/Shibboleth.sso/Logout
```

which will close the Shibboleth session on the Server. Unless the user has authenticated within the last 30 minutes with the DARIAH IdP, a new login will be required when next accessing the services.

3.2.3 Accessing user identity

After successful authentication, the CENDARI front end applications can access the user's attributes from the server's environment variables or headers, as listed in the following table.

Attribute	Contents
eppn	the *eduPersonPrincipalName*, the user's unique username
givenName	the user's first name
sn	the user's last name
cn	comma seperated list of the user's full names
mail	comma seperated list of the user's email addresses
isMemberOf	comma seperated list of the user's DARIAH groups

All of these attributes are UTF-8 encoded strings, but some will eventually be reduced to ASCII. At the time of implementing Shibboleth authentication in CENDARI, DARIAH-DE had not finalised the specification of these attributes.

In general, the eppn is the user's unique identifier, ending in `@dariah.eu` for all DARIAH users. While DARIAH is working towards an integration with DFN-AAI Shibboleth federation [5], which in turn is part of the European eduGAIN meta federation [6], due to blocking bugs in the beta phase, CENDARI will not support this, but require all users to register a DARIAH Homeless Account with the DARIAH AAI.

The part of the eppn before the DARIAH domain suffix usually consists of the user's DARIAH login name, but this varies. The characters encountered can be any ascii character, including spaces, quotes and the @ symbol. The acutal CENDARI username and the Data API session key can be requested from the CENDARI Data API by sending the user's eppn, mail and cn to the API, see POST /session

For both the full name and the mail address, some users have more than one entry which are presented to the application as comma seperated list. Consistency in their order is not guaranteed.

The isMemberOf attribute contains a list of groups that the user was added to by DARIAH AAI management. These groups are used to grant access and privileges to users on a per-application basis.

For each application a number of these groups can be mapped to specific access rights. Thus upon login, the application matches this list against the known groups corresponding to its access rights.

3.3 Maintenance

Todo

User changes Mail Adress

Todo

User changes Username

Todo

restore from backup

---

[5] DFN-AAI – Authentifikations- und Autorisierungs-Infrastruktur
[6] eduGAIN – Interconnecting federations to link services and users worldwide

Todo

re-index all data

This section describes the actual implementation of the components. As far as possible, inline source code documentation from the componentes source is used directly to compile the documentation.

Documentation on the implementation of the following CENDARI components is available.

4.1 CENDARI REST API

4.1.1 Server location

The API is available only internally on CENDARI servers and does not accept requests from outside (local connections and ssh tunnels only).

If you want to connect to the API from outside, you can use the server connect.cendari.dariah.eu. This server can be accessed using ssh on the standard port 22 from worldwide using SSH keys, no passwords. Please note that you can only access the API using the account, you do not get real shell access. See the server's page for details on how to get an Account.

4.1.2 Authorization

In order to get any result, you need to provide your CKAN API key as the `Authorization' HTTP request header.

Your CKAN API Key: log in to CKAN, go to your Profile and copy the value for the private API Key in the lower left corner of the page.

In order for an application (e.g. VRE) to get this key for a Shibboleth-authenticated user, it should post Shibboleth attributes to the API. The key will be returned in response, and it should be used for future API requests.

4.1.3 API Requests

GET /status

Response JSON object:

Attribute	Type	Description
status	String	If the API is running, ``OK'' is returned
hostname	String	FQDN of the API server
version	String	API version

Example request:

```
curl http://localhost:42042/v1/status
```

Example response:

```
{
  "status": "OK",
  "hostname": "int2.cendari.dariah.eu",
  "version": "1.1.0-SNAPSHOT"
}
```

POST /session

Request JSON object:

Attribute	Type	Description
eppn	String	User's eduPersonPrincipalName
mail	String	User's e-mail address
cn	String	User's full name

Response JSON object:

Attribute	Type	Description
username	String	User's unique username
sessionKey	String	Authentication key
sysadmin	Boolean	*true* if a user is a sysadmin, otherwise *false*

Example request:

```
curl -H "Content-Type: application/json" -d '{"eppn":"cendari_notes@dariah.eu", "mail":"cendari-admins@gwdg.de"
```

Example response:

```
{
  "username": "cendari_notes",
  "sessionKey": "******",
  "sysadmin": false
}
```

GET /dataspaces

Request filter parameters:

Parameter	Type	Description
visibility (optional)	String	Valid values are: `public`, `private`, and `all`. Default is `all`
origin (optional)	String	If specified, only the resources that have the specified origin will be included
state (optional)	String	Valid values are: ``active'', ``deleted'', and ``all''. Default is ``active''
since (optional)	DateTime (ISO 8601)	If specified, only the resources created or modified after the specified date/time will be included
until (optional)	DateTime (ISO 8601)	If specified, only the resources created or modified before the specified date/time will be included

Response JSON object:

Attribute	Type	Description
data	Array	Array of dataspace objects

Dataspace object:

Attribute	Type	Description
id	String	Unique id of the dataspace object
url	String	Dataspace resource URL
resources	String	URL for the dataspace's resources
sets	String	URL for the dataspace's sets
name	String	Dataspace unique name
title	String	Dataspace title
description	String	Additional information about the dataspace
visibility	String	Visibility can be `private` or `public`
origin	String	Information about origin of the dataspace (NTE, AtoM, ...)
state	String	State can be ``active'' or ``deleted''

Example request:

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/dataspaces
```

Example response:

```
{
  "data": [{
    "id": "c5633d7f-8bb8-4b77-be22-6ee722ff4705",
    "url": "http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705",
    "resources": "http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705/resources",
    "sets": "http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705/sets",
    "name": "cendari-dataspace",
    "title": "CENDARI dataspace",
    "description": "Content of this dataspaces is created within the CENDARI project",
    "visibility": "private",
    "origin": "",
    "state": "active"
  }, ...]
}
```

Example request (with filter parameter `state`):

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/dataspaces?state=all
```

Example request (with filter parameters `since` and `until`):

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/dataspaces?since=2015-05-20T13:00:00\&until=2015
```

Example response when there are no valid results:

```
{
  "end": true
}
```

GET /dataspaces/$id

Parameters:

Parameter	Type	Description
id	String	Unique identifier of the dataspace object

Response JSON object:

Attribute	Type	Description
id	String	Unique identifier of the dataspace object
url	String	Dataspace resource URL
resources	String	URL for the dataspace's resources
sets	String	URL for the dataspace's sets
name	String	Dataspace unique name
title	String	Dataspace title
description	String	Additional information about the dataspace
visibility	String	Visibility can be `private` or `public`
origin	String	Information about origin of the dataspace (NTE, AtoM, ...)
state	String	State can be ``active'' or ``deleted''

Example request:

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/dataspaces/33175d43-6f5c-4099-abbf-8197f9c2df4b
```

Example response:

```
{
  "id": "c5633d7f-8bb8-4b77-be22-6ee722ff4705",
  "url": "http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705",
  "resources": "http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705/resources",
  "sets": "http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705/sets",
  "name": "cendari-dataspace",
  "title": "CENDARI dataspace",
  "description": "Content of this dataspaces is created within the CENDARI project",
  "visibility": "private",
  "origin": "",
  "state": "active"
}
```

POST /dataspaces

Request JSON object:

Attribute	Type	Description
name	String	Unique name of the dataspace (a string between 2 and 100 characters long, containing only lowercase alphanumeric characters, − and _)
title	String	Dataspace title
description	String	Additional information about the dataspace
visibility	String	Visibility can be `private` or `public`. Default value is `private`. If a dataspace is `public`, all registered users have read-only access to it. If a dataspace is `private`, only users with privileges can access it.
origin	String	Information about origin of the dataspace (NTE, AtoM, ...)

Response JSON object:

Attribute	Type	Description
id	String	Unique id of the dataspace object
url	String	Dataspace resource URL
resources	String	URL for the dataspace's resources
sets	String	URL for the dataspace's sets
name	String	Dataspace unique name
title	String	Dataspace title
description	String	Additional information about the dataspace
visibility	String	Visibility can be `private` or `public`
origin	String	Information about origin of the dataspace (NTE, AtoM, …)
state	String	State can be ``active'' or ``deleted''

Example request:

```
curl -H "Authorization: your-apikey" -H "Content-Type: application/json" \
    -d '{"name":"cendari-dataspace", "title":"CENDARI dataspace", "description":"Content of this dataspaces is
    http://localhost:42042/v1/dataspaces
```

Example response:

```
{
  "id": "c5633d7f-8bb8-4b77-be22-6ee722ff4705",
  "url": "http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705",
  "resources": "http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705/resources",
  "sets": "http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705/sets",
  "name": "cendari-dataspace",
  "title": "CENDARI dataspace",
  "description": "Content of this dataspaces is created within the CENDARI project",
  "visibility": "private",
  "origin": "",
  "state": "active"
}
```

PUT /dataspaces/$id

Request JSON object:

Attribute	Type	Description
title	String	Dataspace title
description	String	Additional information about the dataspace
visibility	String	Visibility can be `private` or `public`. If a dataspace is `public`, all registered users have read-only access to it. If a dataspace is `private`, only users with privileges can access it.
origin	String	Information about origin of the dataspace (NTE, AtoM, …)

Response JSON object:

Attribute	Type	Description
id	String	Unique id of the dataspace object
url	String	Dataspace resource URL
resources	String	URL for the dataspace's resources
sets	String	URL for the dataspace's sets
name	String	Dataspace unique name
title	String	Dataspace title
description	String	Additional information about the dataspace
visibility	String	Visibility can be `private` or `public`
origin	String	Information about origin of the dataspace (NTE, AtoM, …)
state	String	State can be ``active'' or ``deleted''

Example request:

```
curl -X PUT -H "Authorization: your-apikey" -H "Content-Type: application/json" \
    -d '{"title":"CENDARI dataspace", "description":"Content of this dataspaces is created within the CENDARI p
    http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705
```

Example response:

```
{
  "id": "c5633d7f-8bb8-4b77-be22-6ee722ff4705",
  "url": "http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705",
  "resources": "http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705/resources",
  "sets": "http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705/sets",
  "name": "cendari-dataspace",
  "title": "CENDARI dataspace",
  "description": "Content of this dataspaces is created within the CENDARI project",
  "visibility": "public",
  "origin": "",
  "state": "active"
}
```

DELETE /dataspaces/$id

Parameters:

Parameter	Type	Description
id	String	Unique identifier of the dataspace object

Example request:

```
curl -X DELETE -H "Authorization: your-apikey" http://localhost:42042/v1/dataspaces/e9d8a1f7-5641-46f7-929f-231
```

GET /dataspaces/$did/resources

Parameters:

Parameter	Type	Description
did	String	Unique identifier of the dataspace object

Request filter parameters:

Parameter	Type	Description
state (optional)	String	Valid values are: ``active'', ``deleted'', and ``all''. Default is ``active''.
since (optional)	DateTime (ISO 8601)	If specified, only the resources created or modified after the specified date/time will be included
until (optional)	DateTime (ISO 8601)	If specified, only the resources created or modified before the specified date/time will be included

Response JSON object:

Attribute	Type	Description
nextPage (optional)	String	URL of the next page of results
currentPage (optional)	String	URL of the current page of results
data (optional)	Array	Array of resource objects that belong to the dataspace
end	Boolean	`false` if there are no more resources to return, otherwise `true`

Resource object:

Attribute	Type	Description
id	String	Unique id of the resource object
url	String	Resource object URL
dataUrl	String	Resource file URL
viewDataUrl	String	URL for CKAN resource preview
name	String	Resource title
description	String	Additional information about the resource
format	String	Resource file format
mimetype	String	Resource file MIME Type
size	Integer	Resource file size (in KB)
created	String	Timestamp created in ISO 8601 format
created_epoch	Integer	Timestamp created
modified	String	Timestamp last modified in ISO 8601 format
modified_epoch	Integer	Timestamp last modified
setId	String	Unique identifier of the set that the resource belongs to
state	String	State can be ``active'' or ``deleted''

Example request (with the next page, resources that belong to a dataspace):

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705,
```

Example response:

```
{
  "nextPage": "http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705/resources/query/results
  "currentPage": "http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705/resources/query/res
  "data": [{
    "id": "fee6284a-154d-4a33-832c-1836c5561658",
    "url": "http://localhost:42042/v1/resources/fee6284a-154d-4a33-832c-1836c5561658",
    "dataUrl": "http://localhost:42042/v1/resources/fee6284a-154d-4a33-832c-1836c5561658/data",
    "viewDataUrl": "https://int2.cendari.dariah.eu/ckan/dataset/5038ef1a-475b-4312-b395-b63a2e2252fc/resource/fe
    "name": "CENDARI-logo.jpg",
    "description": "Updated resource: CENDARI logo 150x150 without text",
    "format": "JPG",
    "mimetype": "application/octet-stream",
    "size": 1,
    "created_epoch": 1427724584090,
    "modified_epoch": 1427724584090,
    "setId": "5038ef1a-475b-4312-b395-b63a2e2252fc",
    "state": "active",
    "created": "2015-03-30T16:09:44Z",
    "modified": "2015-03-30T16:09:44Z"
  }, ...],
  "end": false
}
```

Example response (when there are no more resources in the dataspace to return):

```
{
  "end": true
}
```

Example request (with filter parameter `state`):

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705,
```

Example request (with filter parameters `since` and `until`):

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/dataspaces/c5633d7f-8bb8-4b77-be22-6ee722ff4705,
```

GET /dataspaces/$did/sets

Parameters:

Parameter	Type	Description
did	String	Unique identifier of the dataspace object

Request filter parameters:

Parameter	Type	Description
state (optional)	String	Valid values are: `active`, `deleted`, and `all`. Default is `active`.

Response JSON object:

Attribute	Type	Description
nextPage (optional)	String	URL of the next page of results
currentPage (optional)	String	URL of the current page of results
data (optional)	Array	Array of set objects that belong to the dataspace
end	Boolean	`false` if there are no more sets to return, otherwise `true`

Set object:

Attribute	Type	Description
id	String	Unique id of the set object
url	String	Set resource URL
name	String	Set unique name
title	String	Set title
description	String	Additional information about the set
dataspaceId	String	Unique id of the dataspace that the set belongs to
dataspaceUrl	String	URL for the dataspace that the set belongs to
resources	String	URL for the set's resources
state	String	State can be `active` or `deleted`

Example request (with the next page, resources that belong to a dataspace):

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/dataspaces/099c3cae-9fe2-4acf-970f-b5b149eeae24,
```

Example response:

```
{
  "nextPage": "http://localhost:42042/v1/dataspaces/099c3cae-9fe2-4acf-970f-b5b149eeae24/sets/query/results/AAA
  "currentPage": "http://localhost:42042/v1/dataspaces/099c3cae-9fe2-4acf-970f-b5b149eeae24/sets/query/results/
  "data": [{
    "id": "2bba037f-d5c3-482a-965c-89f057582b37",
    "url": "http://localhost:42042/v1/sets/2bba037f-d5c3-482a-965c-89f057582b37",
    "name": "9200316-bibliographicresource_3000073808381",
    "title": "Aerial photograph, Lombartzyde, Belgium, 1917",
    "description": "",
    "dataspaceId": "099c3cae-9fe2-4acf-970f-b5b149eeae24",
    "dataspaceUrl": "http://localhost:42042/v1/dataspaces/099c3cae-9fe2-4acf-970f-b5b149eeae24",
    "resources": "http://localhost:42042/v1/sets/2bba037f-d5c3-482a-965c-89f057582b37/resources",
    "state": "active"
  }, ...],
  "end": false
}
```

Example response (when there are no more sets in the dataspace to return):

```
{
  "end": true
}
```

GET /sets

Request filter parameters:

Parameter	Type	Description
dataspaceId (optional)	String	Unique identifier of the dataspace object
state (optional)	String	Valid values are: `active`, `deleted`, and `all`. Default is `active`.

Response JSON object:

Attribute	Type	Description
nextPage (optional)	String	URL of the next page of results
currentPage (optional)	String	URL of the current page of results
data (optional)	Array	Array of set objects
end	Boolean	`false` if there are no more sets to return, otherwise `true`

Set object:

Attribute	Type	Description
id	String	Unique id of the set object
url	String	Set resource URL
name	String	Set unique name
title	String	Set title
description	String	Additional information about the set
dataspaceId	String	Unique id of the dataspace that the set belongs to
dataspaceUrl	String	URL for the dataspace that the set belongs to
resources	String	URL for the set's resources
state	String	State can be ``active'' or ``deleted''

Example request:

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/sets
```

Example response:

```
{
  "nextPage": "http://localhost:42042/v1/sets/query/results/AAAAAAAAAAAAAAAFN1yVT1AAAAAAAAAAAKAAAACg==",
  "currentPage": "http://localhost:42042/v1/sets/query/results/AAAAAAAAAAAAAAAFN1yVT1AAAAAAAAAAAAAAAACg==",
  "data": [{
    "id": "2bba037f-d5c3-482a-965c-89f057582b37",
    "url": "http://localhost:42042/v1/sets/2bba037f-d5c3-482a-965c-89f057582b37",
    "name": "9200316-bibliographicresource_3000073808381",
    "title": "Aerial photograph, Lombartzyde, Belgium, 1917",
    "description": "",
    "dataspaceId": "099c3cae-9fe2-4acf-970f-b5b149eeae24",
    "dataspaceUrl": "http://localhost:42042/v1/dataspaces/099c3cae-9fe2-4acf-970f-b5b149eeae24",
    "resources": "http://localhost:42042/v1/sets/2bba037f-d5c3-482a-965c-89f057582b37/resources",
    "state": "active"
  },...],
  "end": false
}
```

Example response when there are no more results to return:

```
{
  "end": true
}
```

Example request with filter parameter `dataspaceId`:

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/sets?dataspaceId=099c3cae-9fe2-4acf-970f-b5b149
```

GET /sets/$id

Parameters:

Parameter	Type	Description
id	String	Unique identifier of the set object

Response JSON object:

Attribute	Type	Description
id	String	Unique id of the set object
url	String	Set resource URL
name	String	Set unique name
title	String	Set title
description	String	Additional information about the set
dataspaceId	String	Unique id of the dataspace that the set belongs to
dataspaceUrl	String	URL for the dataspace that the set belongs to
resources	String	URL for the set's resources
state	String	State can be `active` or `deleted`

Example request:

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/sets/2bba037f-d5c3-482a-965c-89f057582b37
```

Example response:

```
{
  "id": "2bba037f-d5c3-482a-965c-89f057582b37",
  "url": "http://localhost:42042/v1/sets/2bba037f-d5c3-482a-965c-89f057582b37",
  "name": "9200316-bibliographicresource_3000073808381",
  "title": "Aerial photograph, Lombartzyde, Belgium, 1917",
  "description": "",
  "dataspaceId": "099c3cae-9fe2-4acf-970f-b5b149eeae24",
  "dataspaceUrl": "http://localhost:42042/v1/dataspaces/099c3cae-9fe2-4acf-970f-b5b149eeae24",
  "resources": "http://localhost:42042/v1/sets/2bba037f-d5c3-482a-965c-89f057582b37/resources",
  "state": "active"
}
```

POST /sets

Creating sets in a dataspace is allowed only for editors and admins of a dataspace, and sysadmins. A user with a `member` role on a dataspace has read-only access and therefore cannot add sets.

Request JSON object:

Attribute	Type	Description
name	String	Unique name of the set (a string between 2 and 100 characters long, containing only lowercase alphanumeric characters, – and _)
title (optional)	String	Set title
description (optional)	String	Additional information about the set
dataspaceId	String	Unique id of the dataspace that the set belongs to

Response JSON object:

Attribute	Type	Description
id	String	Unique id of the set object
url	String	Set resource URL
name	String	Set unique name
title	String	Set title
description	String	Additional information about the set
dataspaceId	String	Unique id of the dataspace that the set belongs to
dataspaceUrl	String	URL for the dataspace that the set belongs to
resources	String	URL for the set's resources
state	String	State can be `active` or `deleted`

Example request:

```
curl -H "Authorization: your-apikey" -H "Content-Type: application/json" \
    -d '{"name":"9200316-bibliographicresource_3000073808381", "title":"Aerial photograph, Lombartzyde, Belgium
    http://localhost:42042/v1/sets
```

Example response:

```
{
  "id": "2bba037f-d5c3-482a-965c-89f057582b37",
  "url": "http://localhost:42042/v1/sets/2bba037f-d5c3-482a-965c-89f057582b37",
  "name": "9200316-bibliographicresource_3000073808381",
  "title": "Aerial photograph, Lombartzyde, Belgium, 1917",
  "description": "",
  "dataspaceId": "099c3cae-9fe2-4acf-970f-b5b149eeae24",
  "dataspaceUrl": "http://localhost:42042/v1/dataspaces/099c3cae-9fe2-4acf-970f-b5b149eeae24",
  "resources": "http://localhost:42042/v1/sets/2bba037f-d5c3-482a-965c-89f057582b37/resources",
  "state": "active"
}
```

PUT /sets/$id

Set update is allowed only for editors and admins of a dataspace, and sysadmins. A user with a `member` role on a dataspace has read-only access and therefore cannot update sets in that dataspace. Moving a set from one dataspace to another is not allowed.

Request JSON object:

Attribute	Type	Description
title (optional)	String	Set title
description (optional)	String	Additional information about the set

Response JSON object:

Attribute	Type	Description
id	String	Unique id of the set object
url	String	Set resource URL
name	String	Set unique name
title	String	Set title
description	String	Additional information about the set
dataspaceId	String	Unique id of the dataspace that the set belongs to
dataspaceUrl	String	URL for the dataspace that the set belongs to
resources	String	URL for the set's resources
state	String	State can be `active` or `deleted`

Example request:

```
curl -X PUT -H "Authorization: your-apikey" -H "Content-Type: application/json" \
    -d '{"description":"This is a copy of an aerial photograph."}' \
    http://localhost:42042/v1/sets/2bba037f-d5c3-482a-965c-89f057582b37
```

Example response:

```
{
  "id": "2bba037f-d5c3-482a-965c-89f057582b37",
  "url": "http://localhost:42042/v1/sets/2bba037f-d5c3-482a-965c-89f057582b37",
  "name": "9200316-bibliographicresource_3000073808381",
  "title": "Aerial photograph, Lombartzyde, Belgium, 1917",
  "description": "This is a copy of an aerial photograph.",
  "dataspaceId": "099c3cae-9fe2-4acf-970f-b5b149eeae24",
  "dataspaceUrl": "http://localhost:42042/v1/dataspaces/099c3cae-9fe2-4acf-970f-b5b149eeae24",
  "resources": "http://localhost:42042/v1/sets/2bba037f-d5c3-482a-965c-89f057582b37/resources",
  "state": "active"
}
```

GET /sets/$id/resources

Parameters:

Parameter	Type	Description
id	String	Unique identifier of the set object

Request filter parameters:

Parameter	Type	Description
state (optional)	String	Valid values are: `active`, `deleted`, and `all`. Default is `active`.
since (optional)	DateTime (ISO 8601)	If specified, only the resources created or modified after the specified date/time will be included
until (optional)	DateTime (ISO 8601)	If specified, only the resources created or modified before the specified date/time will be included

Response JSON object:

Attribute	Type	Description
nextPage (optional)	String	URL of the next page of results
currentPage (optional)	String	URL of the current page of results
data (optional)	Array	Array of resource objects that belong to the set
end	Boolean	`false` if there are no more resources to return, otherwise `true`

Resource object:

Attribute	Type	Description
id	String	Unique id of the resource object
url	String	Resource object URL
dataUrl	String	Resource file URL
viewDataUrl	String	URL for CKAN resource preview
name	String	Resource title
description	String	Additional information about the resource
format	String	Resource file format
mimetype	String	Resource file MIME Type
size	Integer	Resource file size (in KB)
created	String	Timestamp created in ISO 8601 format
created_epoch	Integer	Timestamp created
modified	String	Timestamp last modified in ISO 8601 format
modified_epoch	Integer	Timestamp last modified
setId	String	Unique identifier of the set that the resource belongs to
state	String	State can be ``active'' or ``deleted''

Example request (with the next page, resources that belong to a dataspace):

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/sets/5038ef1a-475b-4312-b395-b63a2e2252fc/resou
```

Example response:

```
{
  "nextPage": "http://localhost:42042/v1/sets/5038ef1a-475b-4312-b395-b63a2e2252fc/resources/query/results/AAAA
  "currentPage": "http://localhost:42042/v1/set/5038ef1a-475b-4312-b395-b63a2e2252fc/resources/query/results/AA
  "data": [{
    "id": "fee6284a-154d-4a33-832c-1836c5561658",
    "url": "http://localhost:42042/v1/resources/fee6284a-154d-4a33-832c-1836c5561658",
    "dataUrl": "http://localhost:42042/v1/resources/fee6284a-154d-4a33-832c-1836c5561658/data",
    "viewDataUrl": "https://int2.cendari.dariah.eu/ckan/dataset/5038ef1a-475b-4312-b395-b63a2e2252fc/resource/f
    "name": "CENDARI-logo.jpg",
    "description": "Updated resource: CENDARI logo 150x150 without text",
    "format": "JPG",
    "mimetype": "application/octet-stream",
    "size": 1,
    "created_epoch": 1427724584090,
    "modified_epoch": 1427724584090,
    "setId": "5038ef1a-475b-4312-b395-b63a2e2252fc",
    "state": "active",
    "created": "2015-03-30T16:09:44Z",
    "modified": "2015-03-30T16:09:44Z"
  }, ...],
  "end": false
}
```

Example response (when there are no more resources to return):

```
{
  "end": true
}
```

Example request (with filter parameters `since` and `until`):

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/sets/5038ef1a-475b-4312-b395-b63a2e2252fc/resou
```

GET /resources

Request filter parameters:

Parameter	Type	Description
dataspaceId (optional)	String	Unique identifier of the dataspace object
state (optional)	String	Valid values are: ``active'', ``deleted'', and ``all''. Default is ``active''.
since (optional)	DateTime (ISO 8601)	If specified, only the resources created or modified after the specified date/time will be included
until (optional)	DateTime (ISO 8601)	If specified, only the resources created or modified before the specified date/time will be included

Response JSON object:

Attribute	Type	Description
nextPage (optional)	String	URL of the next page of results
currentPage (optional)	String	URL of the current page of results
data (optional)	Array	Array of resource objects
end	Boolean	`false` if there are no more resources to return, otherwise `true`

Resource object:

Attribute	Type	Description
id	String	Unique id of the resource object
url	String	Resource object URL
dataUrl	String	Resource file URL
viewDataUrl	String	URL for CKAN resource preview
name	String	Resource title
description	String	Additional information about the resource
format	String	Resource file format
mimetype	String	Resource file MIME Type
size	Integer	Resource file size (in KB)
created	String	Timestamp created in ISO 8601 format
created_epoch	Integer	Timestamp created
modified	String	Timestamp last modified in ISO 8601 format
modified_epoch	Integer	Timestamp last modified
setId	String	Unique identifier of the set that the resource belongs to
state	String	State can be ``active'' or ``deleted''

Example request (with the next page, resources that belong to a dataspace):

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/resources
```

Example response:

```
{
  "nextPage": "http://localhost:42042/v1/resources/query/results/AAAAAAAAAAAAAAAFNuaKzXwAAAAAAAAAKAAAACg==",
  "currentPage": "http://localhost:42042/v1/resources/query/results/AAAAAAAAAAAAAAAFNuaKzXwAAAAAAAAAAAAAACg==",
  "data": [{
    "id": "fee6284a-154d-4a33-832c-1836c5561658",
    "url": "http://localhost:42042/v1/resources/fee6284a-154d-4a33-832c-1836c5561658",
    "dataUrl": "http://localhost:42042/v1/resources/fee6284a-154d-4a33-832c-1836c5561658/data",
    "viewDataUrl": "https://int2.cendari.dariah.eu/ckan/dataset/5038ef1a-475b-4312-b395-b63a2e2252fc/resource/fe
    "name": "CENDARI-logo.jpg",
    "description": "Updated resource: CENDARI logo 150x150 without text",
    "format": "JPG",
    "mimetype": "application/octet-stream",
    "size": 1,
    "created_epoch": 1427724584090,
    "modified_epoch": 1427724584090,
    "setId": "5038ef1a-475b-4312-b395-b63a2e2252fc",
    "state": "active",
    "created": "2015-03-30T16:09:44Z",
    "modified": "2015-03-30T16:09:44Z"
  }, ...],
  "end": false
}
```

Example response (when there are no more resources in the dataspace to return):

```
{
  "end": true
}
```

Example request (with filter parameters `dataspaceId` and `state`):

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/resources?dataspaceId=c5633d7f-8bb8-4b77-be22-6c
```

Example requests (with filter parameters `since` and `until`):

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/resources?since=2015-05-20T13:00:00\&until=2015-
```

GET /resources/$id

Returns resource metadata

Parameters:

Parameter	Type	Description
id	String	Unique identifier of the resource object

Response JSON object:

Attribute	Type	Description
id	String	Unique id of the resource object
url	String	Resource object URL
dataUrl	String	Resource file URL
viewDataUrl	String	URL for CKAN resource preview
name	String	Resource title
description	String	Additional information about the resource
format	String	Resource file format
mimetype	String	Resource file MIME Type
size	Integer	Resource file size (in KB)
created	String	Timestamp created in ISO 8601 format
created_epoch	Integer	Timestamp created
modified	String	Timestamp last modified in ISO 8601 format
modified_epoch	Integer	Timestamp last modified
setId	String	Unique identifier of the set that the resource belongs to
state	String	State can be ``active'' or ``deleted''

Example request:

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/resources/fee6284a-154d-4a33-832c-1836c5561658
```

Example response:

```
{
  "id": "fee6284a-154d-4a33-832c-1836c5561658",
  "url": "http://localhost:42042/v1/resources/fee6284a-154d-4a33-832c-1836c5561658",
  "dataUrl": "http://localhost:42042/v1/resources/fee6284a-154d-4a33-832c-1836c5561658/data",
  "viewDataUrl": "https://int2.cendari.dariah.eu/ckan/dataset/5038ef1a-475b-4312-b395-b63a2e2252fc/resource/fee
  "name": "CENDARI-logo.jpg",
  "description": "CENDARI logo 150x150 without text",
  "format": "JPG",
  "mimetype": "application/octet-stream",
  "size": 1,
  "created_epoch": 1427724584090,
  "modified_epoch": 1427724584090,
  "setId": "5038ef1a-475b-4312-b395-b63a2e2252fc",
  "state": "active",
  "created": "2015-03-30T16:09:44Z",
  "modified": "2015-03-30T16:09:44Z"
}
```

POST /resources

Creates new resource in the specified set

Request multipart/form-data:

Field	Type	Description
file	File	File for upload
name (optional)	String	Resource title. If not specified, it defaults to file name
format (optional)	String	File format (XML, RDF, JSON, JPG, ...)
description (optional)	String	Additional information about the resource
setId	String	Unique identifier of the set that the resource belongs to

Response JSON object:

Attribute	Type	Description
id	String	Unique id of the resource object
url	String	Resource object URL
dataUrl	String	Resource file URL
viewDataUrl	String	URL for CKAN resource preview
name	String	Resource title
description	String	Additional information about the resource
format	String	Resource file format
mimetype	String	Resource file MIME Type
size	Integer	Resource file size
created	String	Timestamp created in ISO 8601 format
created_epoch	Integer	Timestamp created
modified	String	Timestamp last modified in ISO 8601 format
modified_epoch	Integer	Timestamp last modified
setId	String	Unique identifier of the set that the resource belongs to

Example request:

```
curl -H "Authorization: your-apikey"  \
    -F "file=@cendari-logo.jpg" \
    -F "format=JPG" \
    -F "description=CENDARI logo 150x150 without text" \
    -F "setId=5038ef1a-475b-4312-b395-b63a2e2252fc" \
    http://localhost:42042/v1/resources
```

Example response:

```
{
  "id": "fee6284a-154d-4a33-832c-1836c5561658",
  "url": "http://localhost:42042/v1/resources/fee6284a-154d-4a33-832c-1836c5561658",
  "dataUrl": "http://localhost:42042/v1/resources/fee6284a-154d-4a33-832c-1836c5561658/data",
  "viewDataUrl": "https://int2.cendari.dariah.eu/ckan/dataset/5038ef1a-475b-4312-b395-b63a2e2252fc/resource/fee
  "name": "cendari-logo.jpg",
  "description": "CENDARI logo 150x150 without text",
  "format": "JPG",
  "mimetype": "application/octet-stream",
  "size": 1,
  "created_epoch": 1427724584090,
  "modified_epoch": 1427724584090,
  "setId": "5038ef1a-475b-4312-b395-b63a2e2252fc",
  "state": "active",
  "created": "2015-03-30T16:09:44Z",
  "modified": "2015-03-30T16:09:44Z"
}
```

PUT /resources/$id

Parameters:

Parameter	Type	Description
id	String	Unique identifier of the resource object

Request multipart/form-data:

Field	Type	Description
file	File	File for upload
name (optional)	String	Resource title. If not specified, it defaults to file name
format (optional)	String	File format (XML, RDF, JSON, JPG, ...)
description (optional)	String	Additional information about the resource

Response JSON object:

Attribute	Type	Description
id	String	Unique id of the resource object
url	String	Resource object URL
dataUrl	String	Resource file URL
viewDataUrl	String	URL for CKAN resource preview
name	String	Resource title
description	String	Additional information about the resource
format	String	Resource file format
mimetype	String	Resource file MIME Type
size	Integer	Resource file size
created	String	Timestamp created in ISO 8601 format
created_epoch	Integer	Timestamp created
modified	String	Timestamp last modified in ISO 8601 format
modified_epoch	Integer	Timestamp last modified
setId	String	Unique identifier of the set that the resource belongs to
state	String	State can be ``active'' or ``deleted''

Example request:

```
curl -H "Authorization: your-apikey"  \
    -F "file=@new-cendari-logo.jpg" \
    -F "name=New CENDARI logo" \
    -F "format=JPG" \
    -F "description=Updated resource: New CENDARI logo 150x150 without text" \
    http://localhost:42042/v1/resources/fee6284a-154d-4a33-832c-1836c5561658
```

Example response:

```
{
  "id": "fee6284a-154d-4a33-832c-1836c5561658",
  "url": "http://localhost:42042/v1/resources/fee6284a-154d-4a33-832c-1836c5561658",
  "dataUrl": "http://localhost:42042/v1/resources/fee6284a-154d-4a33-832c-1836c5561658/data",
  "viewDataUrl": "https://int2.cendari.dariah.eu/ckan/dataset/5038ef1a-475b-4312-b395-b63a2e2252fc/resource/fee
  "name": "New CENDARI logo",
  "description": "Updated resource: New CENDARI logo 150x150 without text",
  "format": "JPG",
  "mimetype": "application/octet-stream",
  "size": 0,
  "created_epoch": 1427724584090,
  "modified_epoch": 1427724584090,
  "setId": "5038ef1a-475b-4312-b395-b63a2e2252fc",
  "state": "active",
  "created": "2015-03-30T16:09:44Z",
  "modified": "2015-03-30T16:09:44Z"
}
```

DELETE /resources/$id

Parameters:

Parameter	Type	Description
id	String	Unique identifier of the resource object

Example request:

```
curl -X DELETE -H "Authorization: your-apikey" http://localhost:42042/v1/resources/575fb6e5-5b1c-4a1d-896e-802b
```

GET /users

Response JSON object:

Attribute	Type	Description
data	Array	Array of user objects

User object:

Attribute	Type	Description
id	String	User's unique id
url	String	User resource URL
username	String	User's unique username
fullname	String	User's full name
about	String	Additional information about the user
emailHash	String	MD5-hashed user's e-mail address
state	String	State can be ``active'' or ``deleted''

Example request:

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/users
```

Example response:

```
{
  "data": [{
    "id": "716eeb7e-1fec-4eab-9e99-c35cbe7ae96b",
    "url": "http://localhost:42042/v1/users/716eeb7e-1fec-4eab-9e99-c35cbe7ae96b",
    "username": "cendari_notes",
    "fullname": "CENDARI Notes Admin",
    "about": "",
    "emailHash": "15ec72819c9c5c09104fbb03e23c899f",
    "state": "active"
  }, ...]
}
```

GET /users/$id

Parameters:

Parameter	Type	Description
id	String	User's unique identifier

Response JSON object:

Attribute	Type	Description
id	String	User's unique identifier
url	String	User resource URL
username	String	User's unique username
fullname	String	User's full name
about	String	Additional information about the user
emailHash	String	MD5-hashed user's e-mail address
state	String	State can be ``active'' or ``deleted''

Example request:

```
curl -H "Authorization: your-apikey" http://localhost:42042/v1/users/716eeb7e-1fec-4eab-9e99-c35cbe7ae96b
```

Example response:

```
{
  "id": "716eeb7e-1fec-4eab-9e99-c35cbe7ae96b",
  "url": "http://localhost:42042/v1/users/716eeb7e-1fec-4eab-9e99-c35cbe7ae96b",
  "username": "cendari_notes",
  "fullname": "CENDARI Notes Admin",
  "about": "",
  "emailHash": "15ec72819c9c5c09104fbb03e23c899f",
  "state": "active"
}
```

GET /privileges

Returns an array of user privileges for dataspaces. The authenticated user can only list privileges for the dataspaces whose member he is.

Request filter parameters:

Parameter	Type	Description
dataspaceId (optional)	String	Unique identifier of the dataspace object
userId (optional)	String	Unique identifier of the user object
state (optional)	String	Valid values are: ``active'', ``deleted'', and ``all''. Default is ``active''.

Response JSON object:

Attribute	Type	Description
data	Array	Array of user privileges for dataspaces.

Privilege object:

Attribute	Type	Description
id	String	Unique id of the privilege object
url	String	Privilege resource URL
userId	String	Unique id of the user with the given privilege
userUrl	String	User resource URL
dataspaceId	String	Unique id of the dataspace object
dataspaceUrl	String	Dataspace resource URL
role	String	User's role in the dataspace (``member'', ``editor'' or ``admin'')
state	String	State can be ``active'' or ``deleted''

Example request:

```
curl -H "Authorization: your-apikey"  http://localhost:42042/v1/privileges
```

Example response:

```
{
  "data": [{
    "id": "c6a4d3f7-3ac7-457d-9254-4528861d8816",
    "url": "http://localhost:42042/v1/privileges/c6a4d3f7-3ac7-457d-9254-4528861d8816",
    "userId": "716eeb7e-1fec-4eab-9e99-c35cbe7ae96b",
    "userUrl": "http://localhost:42042/v1/users/716eeb7e-1fec-4eab-9e99-c35cbe7ae96b",
    "dataspaceId": "099c3cae-9fe2-4acf-970f-b5b149eeae24",
    "dataspaceUrl": "http://localhost:42042/v1/dataspaces/099c3cae-9fe2-4acf-970f-b5b149eeae24",
    "role": "member",
    "state": "active"
  }, ...]
}
```

Example request:

```
curl -H "Authorization: your-apikey" "http://localhost:42042/v1/privileges?dataspaceId=099c3cae-9fe2-4acf-970f-
```

Example response:

```
{
  "data": [{
    "id": "f44aa8de-e735-4ee3-af2d-d076e1162715",
    "url": "http://localhost:42042/v1/privileges/f44aa8de-e735-4ee3-af2d-d076e1162715",
    "userId": "494dc503-1766-42f1-83da-8ff173b11af5",
    "userUrl": "http://localhost:42042/v1/users/494dc503-1766-42f1-83da-8ff173b11af5",
    "dataspaceId": "099c3cae-9fe2-4acf-970f-b5b149eeae24",
    "dataspaceUrl": "http://localhost:42042/v1/dataspaces/099c3cae-9fe2-4acf-970f-b5b149eeae24",
    "role": "member",
    "state": "active"
  }, {
    "id": "b1807ea8-da5f-49c7-a455-c28613bd74fe",
    "url": "http://localhost:42042/v1/privileges/b1807ea8-da5f-49c7-a455-c28613bd74fe",
    "userId": "eb92e2f7-e7a8-48f6-9e8d-d03b4b8ad8df",
    "userUrl": "http://localhost:42042/v1/users/eb92e2f7-e7a8-48f6-9e8d-d03b4b8ad8df",
    "dataspaceId": "099c3cae-9fe2-4acf-970f-b5b149eeae24",
    "dataspaceUrl": "http://localhost:42042/v1/dataspaces/099c3cae-9fe2-4acf-970f-b5b149eeae24",
    "role": "admin",
    "state": "active"
  }, {
    "id": "5bb7878e-1c19-42bf-a963-b00c7e889e1f",
    "url": "http://localhost:42042/v1/privileges/5bb7878e-1c19-42bf-a963-b00c7e889e1f",
    "userId": "741219ae-6e2b-4e72-85b5-934b18b120df",
    "userUrl": "http://localhost:42042/v1/users/741219ae-6e2b-4e72-85b5-934b18b120df",
    "dataspaceId": "099c3cae-9fe2-4acf-970f-b5b149eeae24",
    "dataspaceUrl": "http://localhost:42042/v1/dataspaces/099c3cae-9fe2-4acf-970f-b5b149eeae24",
    "role": "editor",
    "state": "active"
  }, {
    "id": "c6a4d3f7-3ac7-457d-9254-4528861d8816",
    "url": "http://localhost:42042/v1/privileges/c6a4d3f7-3ac7-457d-9254-4528861d8816",
    "userId": "716eeb7e-1fec-4eab-9e99-c35cbe7ae96b",
    "userUrl": "http://localhost:42042/v1/users/716eeb7e-1fec-4eab-9e99-c35cbe7ae96b",
    "dataspaceId": "099c3cae-9fe2-4acf-970f-b5b149eeae24",
    "dataspaceUrl": "http://localhost:42042/v1/dataspaces/099c3cae-9fe2-4acf-970f-b5b149eeae24",
    "role": "member",
    "state": "active"
  }]
}
```

GET /privileges/$id

Parameters:

Parameter	Type	Description
id	String	Unique identifier of the privilege object

Response JSON object:

Attribute	Type	Description
id	String	Unique id of the privilege object
url	String	Privilege resource URL
userId	String	Unique id of the user with the given privilege
userUrl	String	User resource URL
dataspaceId	String	Unique id of the dataspace object
dataspaceUrl	String	Dataspace resource URL
role	String	User's role in the dataspace (``member'', ``editor'' or ``admin'')
state	String	State can be ``active'' or ``deleted''

Example request:

```
curl -H "Authorization: your-apikey" http://localhost:42042/v1/privileges/c6a4d3f7-3ac7-457d-9254-4528861d8816
```

Example response:

```
{
  "id": "c6a4d3f7-3ac7-457d-9254-4528861d8816",
  "url": "http://localhost:42042/v1/privileges/c6a4d3f7-3ac7-457d-9254-4528861d8816",
  "userId": "716eeb7e-1fec-4eab-9e99-c35cbe7ae96b",
  "userUrl": "http://localhost:42042/v1/users/716eeb7e-1fec-4eab-9e99-c35cbe7ae96b",
  "dataspaceId": "099c3cae-9fe2-4acf-970f-b5b149eeae24",
  "dataspaceUrl": "http://localhost:42042/v1/dataspaces/099c3cae-9fe2-4acf-970f-b5b149eeae24",
  "role": "member",
  "state": "active"
}
```

POST /privileges

Grant ``member'', ``editor'' or ``admin'' privilege to the user on the dataspace.

Request JSON object:

Attribute	Type	Description
userId	String	User's unique identifier
dataspaceId	String	Unique identifier of the dataspace
role	String	Valid values are ``member'', ``editor'' and ``admin''

Response JSON object:

Attribute	Type	Description
id	String	Unique id of the privilege object
url	String	Privilege resource URL
userId	String	Unique id of the user with the given privilege
userUrl	String	User resource URL
dataspaceId	String	Unique id of the dataspace object
dataspaceUrl	String	Dataspace resource URL
role	String	User's role in the dataspace (``member'', ``editor'' or ``admin'')
state	String	State can be ``active'' or ``deleted''

Example request:

```
curl -H "Authorization: your-apikey" -H "Content-Type: application/json" -d '{"userId":"716eeb7e-1fec-4eab-9e99-
```

Example response:

```
{
  "id": "c6a4d3f7-3ac7-457d-9254-4528861d8816",
  "url": "http://localhost:42042/v1/privileges/c6a4d3f7-3ac7-457d-9254-4528861d8816",
  "userId": "716eeb7e-1fec-4eab-9e99-c35cbe7ae96b",
  "userUrl": "http://localhost:42042/v1/users/716eeb7e-1fec-4eab-9e99-c35cbe7ae96b",
  "dataspaceId": "099c3cae-9fe2-4acf-970f-b5b149eeae24",
  "dataspaceUrl": "http://localhost:42042/v1/dataspaces/099c3cae-9fe2-4acf-970f-b5b149eeae24",
  "role": "member",
  "state": "active"
}
```

DELETE /privileges/$id

Parameters:

Parameter	Type	Description
id	String	Unique identifier of the privilege object

HTTP Status Code:

- 204 - Privilege deleted

- 403 - User not authorized to delete the privilege

- 404 - Privilege with specified id not found

Example request:

```
curl -X DELETE -H "Authorization: your-apikey" http://localhost:42042/v1/privileges/c6a4d3f7-3ac7-457d-9254-452
```

4.2 AtoM Extensions

AtoM (Access to Memory) is used within the CENDARI project for management of Archival descriptions.

Todo

Add description of EAD and EAG

The standard installation of AtoM is used in conjunction with a CENDARI theme. In addition CENDARI developed a Shibboleth authentication extension and a script to upload AtoM archival descriptions in EAD and EAG to the CENDARI repository.

4.2.1 Shibboleth Authentication

AtoM plugin: sfDariahShibUserPlugin

This plugin is used to enable Shibboleth Authentication in AtoM. The code is available on GitHub.

About the plugin

The plugin hooks into the authentication dialog, which is only shown on the dedicated login page `/user/login` and, assuming the webserver requires Shibboleth authentication for this page, upon visiting the url the Shibboleth data is used to authenticate the user to AtoM.

User rights are mapped based on the configuration file `apps/qubit/config/app.yml`. All CENDARI project members are given contributer status, while archivists and historians are granted editor status.

To facilitate the Single-Sign-On experience, a number of upstream templates is overwritten:

- The menu template `_userMenu` is changed to remove the drop-down login menu.

- The user templates `editSuccess` and `indexSuccess` are replaced by simplified versions that remove the password handling from the user interface.

- The user template `loginSuccess` is replaced to remove login dialogs on pages other than the dedicated login page, which could appear on session timeout. In this case the user is redirected to the login page.

Code Documentation

Plugin Configuration
class **sfDariahShibUserPluginConfiguration**

> Main Plugin Class for enabling Shibboleth authentication for AtoM.

> property **$summary**
> > Short summary for display in AtoM interface.

property $version
> Current version.

contextLoadFactories()
> Load additional css file for new login link.

initialize()
> Upon initialization, set plugin to active and inject our login class *sfDariahShibUser* into the config

Extending the login page
class UserLoginAction

> Overwrites upstream class to inject Shibboleth handling and pass request down to authentication class.

execute()
> Renders the login dialog, calls the login action if Shibboleth data is present or POST data is sent as a fall back, redirects the user after successful authentication

>> **Parameters**

>>> • **$request** (*sfWebRequest*) -- The current web request.

>> **Returns** void

Performing login and account creation
class sfDariahShibUser

> User Class handling authentication and user creation from Shibboleth data.

authenticate()
> Performs the actual authentication, calling parent if web request's data is missing

>> **Parameters**

>>> • **$usermail** (*string*) -- the mail address of the user to authenticate (entered or from Shibboleth)

>>> • **$password** (*string*) -- the password entered into the login form, empty in case of Shibboleth

>>> • **$request** (*sfWebRequest*) -- the current web request

createUserFromShibInfo()
> Creates a new AtoM user from Shibboleth data and assignes a random password

>> **Parameters**

>>> • **$request** (*sfWebRequest*) -- the current web request

>> **Returns** QubitUser $user The newly created user.

updateUserFromShibInfo()
> Updates user's access privileges from Shibboleth data

>> **Parameters**

>>> • **$user** (*QubitUser*) -- the current user

>>> • **$request** (*sfWebRequest*) -- the current web request

generateUserNameFromShibInfo()
> Generate a username from the Shibboleth ePPN

>> **Parameters**

>>> • **$request** (*sfWebRequest*) -- the current web request

>> **Returns** string $username the local part of the ePPN as username

---

generateRandomPassword()
>   Generates a random 25 character password. An additonal prepended string ensures compliance with tightend AtoM security policy.

>   **Returns** string $andom String to be used as password.

AtoM is integrated into the DARIAH Shibboleth authentication, using the plugin sfDariahShibUserPlugin developed by CENDARI.

4.2.2 Connection to the CKAN Repository

AtoM-CKAN Synchronisation

The files edited in AtoM are synchronised to the CENDARI CKAN repository once per day. This is achieved by a script collection specifically created during the CENDARI project. The code is availabe on GitHub.

The scripts rely on modifications to the AtoM Database that need to be guessed form the scripts Readme or the code itself.

There are three satelite tables in the atom database:

```
harvester_ead
+----------------------+--------------+------+-----+---------+-------+
| Field                | Type         | Null | Key | Default | Extra |
+----------------------+--------------+------+-----+---------+-------+
| atom_ead_id          | int(11)      | NO   | PRI | NULL    |       |
| atom_ead_slug        | varchar(255) | YES  |     | NULL    |       |
| atom_eag_slug        | varchar(255) | YES  |     | NULL    |       |
| repository_resource_id | varchar(40) | YES  |     | NULL    |       |
| sync_date            | datetime     | YES  |     | NULL    |       |
+----------------------+--------------+------+-----+---------+-------+


harvester_eag
+----------------------+--------------+------+-----+---------+-------+
| Field                | Type         | Null | Key | Default | Extra |
+----------------------+--------------+------+-----+---------+-------+
| atom_eag_id          | int(11)      | NO   | PRI | NULL    |       |
| atom_eag_slug        | varchar(255) | YES  |     | NULL    |       |
| repository_resource_id | varchar(40) | YES  |     | NULL    |       |
| sync_date            | datetime     | YES  |     | NULL    |       |
+----------------------+--------------+------+-----+---------+-------+


harvester_date
+-------+----------+------+-----+---------+-------+
| Field | Type     | Null | Key | Default | Extra |
+-------+----------+------+-----+---------+-------+
| date  | datetime | YES  |     | NULL    |       |
+-------+----------+------+-----+---------+-------+
```

Table harvester_ead contains AtoM id and slug of the description, AtoM slug and ID of the description's repository, CKAN ID of the description and the date of the synchronization.

Table harvester_eag contains AtoM slug and ID of the repository, CKAN ID of the repository and the date of the synchronization.

Table harvester_date contains time of the last execution of this script.

A collection of scripts AtoM-CKAN Synchronisation have been developed by CENDARI.

4.3 CKAN Modifications and Extension

CKAN serves as central repository for all CENDARI content. It directly functions as the backend for the CEN-DARI REST API. Some *modifications* to the database have been made.

Todo

Actually provide the specifications of the DB changes.

4.3.1 Shibboleth Authentication

CKAN plugin ``dariahshibboleth''

This plugin is used to enable Shibboleth Authentication in CKAN. The code is available on GitHub.

About the plugin

The plugin hooks into the authentication dialog. The dialog is extended to include a link to Shibboleth authentication.

If a user vists the login page while being authenticated with Shibboleth, he is logged in to CKAN. In case the user does not yet exists in CKAN, an account is created.

On every login, the user's mail address and full name are matched to the Shibboleth data and updated if neccessary.

To facilitate the Single-Sign-On experience, the login page is usually protected on the webserver level and thus never seen by a user. The CKAN internal user management is disabled so that changes can not be made.

Code Documentation

class dariahshibboleth.plugin.**DariahShibbolethPlugin**(***kwargs*)
    Bases: ckan.plugins.core.SingletonPlugin

    Main plugin class implemeting IConfigurer and IAuthenticator.

    abort(*status_code*, *detail*, *headers*, *comment*)
        Simply passes through an abort.

    get_auth_functions()
        Pass.

    identify()
        Extracts the logged in user from the pylons session.

    login()
        Performs the actual login, if Shibboleth data is found by *get_shib_data()*.

        If the a CKAN user with the ePPN does not exist, he is created. Otherwise full name and mail address are updated if neccessary.

        Finally, a pylons session is created for session management.

    logout()
        Log out the user by destroying the pylons session and redirecting to Shibboleth logout.

    update_config(*config*)
        Add our extended login form template with Shibboleth link to CKAN's toolkit.
dariahshibboleth.plugin.**generate_user_name**(*eppn*)
    Returns a valid username by defaulting to the ePPN's local part. This is not federation-ready!

        Parameters eppn -- The ePPN to extract the username from.

> **Returns** Lower cased local part of ePPN.

dariahshibboleth.plugin.**get_shib_data**(*self*)
> Extracts full name, email address and ePPN from Shibboleth data.

> > **Returns** user_dict containing the data or `None` if no Shibboleth data is found.

dariahshibboleth.plugin.**get_user**(*eppn*)
> Look up CKAN user by ePPN.

> > **Parameters** `eppn` -- String holding the ePPN to look up.

> > **Returns** user_dict of the user or `None`.

dariahshibboleth.plugin.**hash_email**(*email*)
> Create a CKAN style hash from an email.

> > **Parameters** `email` -- The email address to hash.

> > **Returns** hex encoded md5 hash of the normalized email.

CKAN plugin ``cendari''

This plugin is used to enable Shibboleth Authentication including the interfacing with the CENDARI Data API for CKAN. The code is available on GitHub.

About the plugin

The plugin hooks into the authentication dialog. The dialog is extended to include a link to Shibboleth authentication.

If a user vists the login page while being authenticated with Shibboleth, his Shibboleth attributes are sent to the Data API's POST /session endpoint. Thus user creationg is deferred to the CENDARI Data API and the plugin than simply logs in the user returned by the Data API. In the event that the API does not respond, the plugin checks wheter a CKAN user corresponding to the Shibboleth attributes exists and logs him in directly.

The plugin further supports the promoting of users to sysadmin status, should this be indicated by the Shibboleth attributes. To use the feature, add a line to CKAN's configuration in the `[app:main]` section, defining groups to be checked against the user's *isMemberOf* attribute:

```
shibboleth_sysadmin_groups = shib-admins shib-ckan-admins
```

To facilitate the Single-Sign-On experience, the login page is usually protected on the webserver level and thus never seen by a user. The CKAN internal user management is disabled so that changes can not be made.

Code Documentation
class cendari.plugin.**CendariAuthPlugin**(***kwargs*)
> Bases: `ckan.plugins.core.SingletonPlugin`

> Main plugin class implemeting `IConfigurer` and `IAuthenticator`.

> **abort**(*status_code*, *detail*, *headers*, *comment*)
> > Simply passes through an abort.

> **get_auth_functions**()
> > Pass.

> **identify**()
> > Extracts the logged in user from the pylons session.

> **login**()
> > Performs the actual login: takes Shibboleth data found by *get_shib_data()* and sends it to the CENDARI Data API.

The user returned by the API is logged in and the plugin verifies the sysadmin status against the Shibboleth data.

If the API does not respond, but a user with the right mail address exists in CKAN, this user is logged in.

Finally, a pylons session is created for session management.

logout()
Log out the user by destroying the pylons session and redirecting to Shibboleth logout.

update_config(*config*)
Add our extended login form template with Shibboleth link to CKAN's toolkit.

cendari.plugin.**get_shib_data**(*self*)
Extracts full name, email address and ePPN from Shibboleth data.

> **Returns** user_dict containing the data or `None` if no Shibboleth data is found.

cendari.plugin.**verify_sysadmin_status**(*self*, *api_username*)
Checks the shibboleth data for sysadmin privileges and grants or revokes them in CKAN accordingly.

The CKAN web interface is integrated into the CENDARI authentication mechanism through the CKAN plugin CKAN plugin ``cendari''. A pure Shibboleth plugin was also developed by CENDARI, but is no longer in use: CKAN plugin ``dariahshibboleth''.