**COLLABORATIVE EUROPEAN DIGITAL ARCHIVE INFRASTRUCTURE**

| | |
|---|---|
| **Project Acronym:** | CENDARI |
| **Project Grant No.:** | 284432 |
| **Theme:** | FP7-INFRASTRUCTURES-2011-1 |
| **Project Start Date:** | 01 February 2012 |
| **Project End Date:** | 31 January 2016 |

| | |
|---|---|
| **Deliverable No. :** | 7.4 |
| **Title of Deliverable:** | Final releases of toolkits |
| **Date of Finalised Deliverable:** | January 2016 |
| **Revision No.:** | 1 |
| **WP No.:** | 7 |
| **Lead Beneficiary:** | MISANU |
| **Author (Name and email address):** | Alexander Meyer alexander.meyer@inria.fr<br>Alfredo Maldonado maldonaa@tcd.ie<br>Bojan Marinkovic bojanm@mi.sanu.ac.rs<br>Emiliano Degl'Innocenti emiliano.degli.innocenti@gmail.com<br>Fabrizio Butini fab.butini@gmail.com<br>Ivan Čukić ivan@mi.sanu.ac.rs<br>Jun Zhang jun.zhang@kcl.ac.uk<br>Laurent Romary Laurent.Romary@inria.fr<br>Mark Hedges mark.hedges@kcl.ac.uk<br>Michael Bryant michael.bryant@kcl.ac.uk<br>Milica Knezevic knezevic.milica@gmail.com<br>Natasa Bulatovic bulatovic@mpdl.mpg.de<br>Patrice Lopez patrice.lopez@inria.fr<br>Wei Tai wtai@scss.tcd.ie<br>Zdenek Uhlír Zdenek.Uhlir@nkp.cz |
| **Dissemination Level:** | PU = public |

SEVENTH FRAMEWORK PROGRAMME

| | |
|---|---|
| **Nature of Deliverable:** | O |
| **Abstract (approx. 150 words):** | The CENDARI final release of data integration and semantic services toolkit deliverable provides information about the tools and services delivered and adopted for the final CENDARI infrastructure, technologies in use and their status.<br><br>In addition, it provides links to the source code and documentation of the toolkits.<br><br>More details about selected components have already been provided in deliverable **D7.2. Data integration toolkit and repository**. In this document we provide short overview about the data integration components developed within the WP7.<br><br>Software, components and technical documentation for the tools are available at CENDARI GitHub (https://github.com/CENDARI).<br><br>User and administrator guides for components are available at https://docs.cendari.dariah.eu. |

# Table of contents

# 1 Data integration components

There are several software components which have been implemented or customised to serve the data integration and data management workflows in CENDARI. Short details about the components are provided further in the document. More details about components have been provided in CENDARI deliverable **D7.2 Data integration toolkit and repository**.

## 1.1 Cendari Architecture

The CENDARI research infrastructure is a complex system of applications. The infrastructure comprises four main layers (see Figure 1):

1. ***Files storage and servers***
2. ***The Data Access Layers***: includes search engines such as Elastic Search and SOLR; includes the RDF triplestore Virtuoso , a PostgreSQL and a MySQL Database
3. ***The Application Layer***: includes the CENDARI Repository and the CENDARI Data API that constitute the communication layer between the data stores and the presentation layer. The application layer integrates applications that connect to the existing infrastructure, such as TRAME (the harvester and scarper for the medieval data) or the Named Entity Recognition (NERD) services.
4. ***The Presentation layer***: includes the presentation of different functions of the CENDARI data space, such as: Ontology Viewer (Pineapple); Notes VRE; Repository User Interface; Archival directory user Interface.

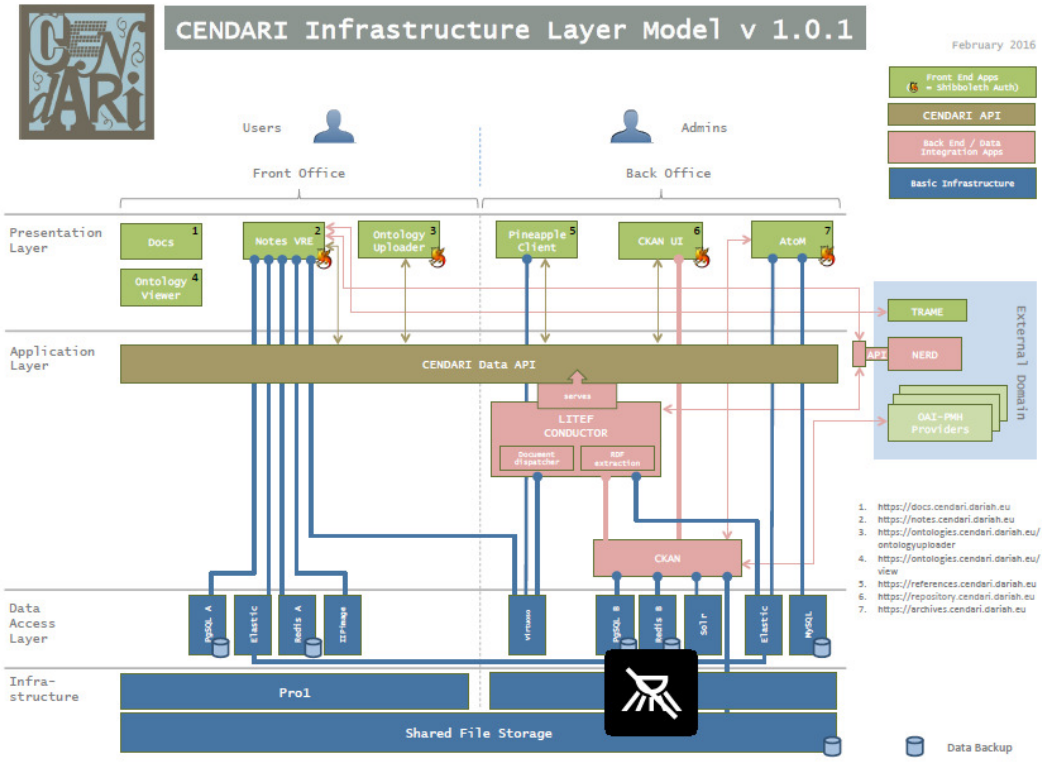


Figure 1: CENDARI Architecture

This document focuses on components developed or customized within the scope of WP7.

## 1.2 CENDARI Repository

| | |
|---|---|
| Component Description | The CENDARI data repository was established to manage content produced and collected in a variety of existing CENDARI workflows. This includes archival descriptions created manually by users, archival descriptions ingested and harvested from external sources and institutions, as well as metadata schemas, ontologies, and all data produced by tools and services within the CENDARI infrastructure. |

| | |
|---|---|
| Component URL | https://repository.cendari.dariah.eu/ |
| Software | Based on CKAN open source solution. CKAN is an open-source data portal software that makes data accessible by providing tools for publishing, sharing, and searching. It's a data management solution for storing raw data and metadata. |
| Software official website | http://ckan.org/ |
| Source code | https://github.com/ckan/ckan |
| Software Version | 2.2.1 |
| Software Documentation | http://docs.ckan.org/en/latest/ |
| Technology | • Python, JavaScript<br>• PostgreSQL<br>• SQLAlchemy Python SQL toolkit and Object Relational Mapper<br>• Apache Solr search platform<br>• Pylons web framework |
| Extension and Plugins | • FileStore<br>• Shibbolet authentication<br>• Text preview<br>• PDF preview |

## 1.3 Archival description toolkit and archival directory

| | |
|---|---|
| Description | The Archival Directory is a large database of archival descriptions and collections and is part of the CENDARI Virtual Research Environment (VRE).<br><br>It has a strong transnational focus and one of its aims is to include many archives and institutions which are little known or rarely used by researchers. The Archival Directory allows historians to view sources in a rarely seen transnational and comparative view. It is focused on archives and libraries containing resources on the Medieval era and World War One.<br><br>All the content of Archival Directory is created manually by users.<br>It is synchronized with the CENDARI repository on daily basis. |
| URL | https://archives.cendari.dariah.eu/ |
| Software | Based on AtoM open source software<br>https://www.accesstomemory.org/ |
| Components | MySql database, Elastic search engine |
| Core technology | Php |
| Extensions | • ATOM2CKAN<br>• ATOM Theme Plugin for CENDARI<br>• Shibboleth plugin for ATOM |
| Documentation | Introductory:<br>https://docs.cendari.dariah.eu/user/atom.html<br>Introductory documentation is available via Archival directory pages as well.<br><br>Full tool documentation:<br>https://www.accesstomemory.org/en/docs/2.2/ |

## 1.4 Litef

| | |
|---|---|
| Description | Litef provides a document storage and dispatch service with a user-facing REST API for creating and defining resources, user groups, dataspaces, and setting user permissions for them. The document dispatch subsystem handles incoming documents, processes them and sends the results to interested 3rd party services. It is plugin-based to allow easier extensibility and integration with other CENDARI components. |
| Software | Developed in-house. Published as Free Software under the GNU Affero General Public License. |
| Technology | Uses CKAN repository and PostgreSQL for data storage. Implemented in Scala programming language with Spray, Akka, Slick and Javelin libraries. |
| URL | https://github.com/CENDARI/litef-conductor |
| Plugins | • NERD plugin - Plugin to convert the data produced by the NERD service into RDF<br>• Elastic plugin - Plugin to feed the processed documents into the Elastic service;<br>• Virtuoso plugin - Plugin to feed the extracted semantic information into the Virtuoso quad store;<br>• Data extraction plugins - A set of plugins to extract the semantic information from various standard file formats. |
| Documentation | https://docs.cendari.dariah.eu/admin/install/litef.html |

## 1.5 Ontology uploader

| | |
|---|---|
| Description | Cendari ontology uploader is a web based application mainly designed for uploading ontology files to Cendari CKAN server. While uploading ontology files, it also generates a metadata file describing the relationship between the uploaded files and the metadata file will be stored with other files. In addition, it also allows user to browser and download ontology and metadata files from the |

| | |
|---|---|
| | CKAN server. And while browsing the files, it also allows user to upload more files to the server and the metadata file stored on the server will also be updated automatically. |
| URL | https://int1.cendari.dariah.eu/ontologyuploader/ |
| Software | Based on CKAN open source solution, Tomcat |
| Technology | Java Servlet, JSP |

### 1.6 Pineapple

| | |
|---|---|
| Description | PINEAPPLE is a web-based interface to the CENDARI semantic repository, enabling search and browse functionality for three types of CENDARI resources: semantically enriched harvested archival material, subject-based ontologies, and medieval manuscripts. PINEAPPLE also provides a JSON API via HTTP content negotiation. |
| URL | https://resources.cendari.dariah.eu/ |
| Software | Web-based |
| Technology | PHP, Slim Framework, SPARQL, JSON |
| Components | Virtuoso Open Source semantic repository |
| Software Home Page | https://github.com/CENDARI/PINEAPPLE |

### 1.7 NERD Service for English language

| | |
|---|---|
| Description | The NERD identifies and disambiguates entities mentioned in text. The process includes a Named Entity Recogniser for identifying open entities in text (such as person, places, dates, etc.) based on CRF and a |

| | |
|---|---|
| | disambiguation of entities against Wikipedia and FreeBases entries, based on statistical measures and a Random Tree Forest regression. |
| URL | http://traces1.saclay.inria.fr/nerd |
| Software | https://github.com/kermitt2/grobid-ner |
| Technology | Java and C++ |
| Components | grobid-ner, nerd |
| Documentation | http://traces1.saclay.inria.fr/nerd/doc/nerd-service-manual.pdf |
| Speed | between 200 and 500 words per second on a Xeon server |
| Scalability | support for multithreading |
| Required memory | around 4-5GB |

## 1.8 Multilingual NERD

| | |
|---|---|
| Description | *mner* is a multi-lingual tool for Named Entity Recognition and Disambiguation/Resolution (NERD). It features three types of entities: persons, places and organizations. Entities of these types are recognized in plain text and returned with their corresponding UTF-8 codepoint offsets. Disambiguation takes place against the Wikipedia corpus of the particular language. If no suitable disambiguation candidate is found, an entity is returned without disambiguation. *mner* is tuned towards near-real-time performance, making it suitable for interactive environments. |
| URL (preliminary Web pages) | http://136.243.145.239/nerd/ |
| REST URL (preliminary) | http://136.243.145.239/nerd/processNERDText, |
| Software | own development, no pre-existing software or libraries used |
| Technology | C (software), Perl (web interface) |

| | |
|---|---|
| Documentation | The JSON interface is similar to the English NERD service described above<br><br>http://cloud.science-miner.com/nerd/doc/nerd-service-manual.pdf.<br><br>Specific documentation will appear soon at<br><br>http://amor.cms.hu-berlin.de/~meyerale/. |
| Code | will appear soon at<br><br>http://amor.cms.hu-berlin.de/~meyerale/ |

## 1.9 Web Scraping Service

| | |
|---|---|
| Description | Scrapy is a web service which reads the contents of web pages and produces structured XML format out of it by applying specific transformation rules. The service has been developed with aim to support harvesting of data from providers who do not offer their data in other structured formats through a data API or as a database export. Data extraction templates have been developed tested with several medieval data providers such as MIRABILE, JONAS, SCRIPTORIUM and MEDIUM.<br><br>Service has been used to harvest data from web sites identified via TRAME service. |
| URL (code) | https://github.com/CENDARI/spider-farm,<br>https://github.com/CENDARI/scrapyd,<br>https://github.com/CENDARI/scrapy |
| Software | Based on Scrapy framework |
| Technology | Python 2.7 (https://www.python.org), Scrapy Framework (http://scrapy.org), Twisted Framework (https://twistedmatrix.com/trac/). |
| Documentation | All documentation available with relevant open source tools |

### 1.10 TRAME

| | |
|---|---|
| Description | TRAME (Texts and Manuscript Transmission of the Middle Ages in Europe) is a web-based application intended to provide a layer of interoperability among different digital resources in the Medieval Culture domain. It implements a metasearch engine for searching data from Medieval data providers. |
| URL | http://git-trame.fefonlus.it/index.php |
| Software | PHP 5.3 |
| Technology | PHP OOP (http://www.php.net) and Apache Httpd 2.2 (http://httpd.apache.org) |
| Code | Will be soon available on GitHub |

## 2 Data integration processing

The main access point to the CENDARI Data integration platform is the CENDARI Data API. It is a REST based API and it serves as a unified layer, which against authorization, allows retrieving, updating or querying of CENDARI data.

When new content is acquired by e.g. CENDARI Harvester it is stored in its originally acquired form within the appropriate dataspace in the Repository. Litef listens to the new acquisitions and starts the data extraction and transformation processes, by invoking respective data extraction/transformation plug-ins. Litef invokes its indexers, who on the other hand decide whether to process the content or not. Litef decides where to send the output of the processed content e.g. Virtuoso triple store, Elastic search engine (accessible via the CENDARI Notes VRE developed by WP9).

Data extraction and transformation plugins base their transformation on the CENDARI ontology. For some “known” data formats such as TEI, EAG, DC or EAD Litef will write the transformations in the knowledge base. For text-based content which adheres to arbitrary other metadata formats (or only fulltexts e.g. Publication PDFs without accompanying metadata). Litef will invoke NER services (Pineapple, CENDARI NER) and the results will be either written as NER Recommendations, or directly matched against existing entities.

Minimal information which Litef will write in the triple store for each acquired content in the Repository is a named graph containing at least system provenance for the harvested data.

The responses from the data extraction/transformation plugins are then processed and written to appropriate knowledge space in the triple store, as well as serialized as a file and written back to the appropriate dataspace in the Repository. This approach ensures:

- the knowledge produced from each data extraction/transformation plugin – is linked to the sources in the Repository at any time
- the triple store can always be re-assembled from the data serialized within the Repository (including links to the original sources)
- provenance information for each data transformation is tracked

Pineapple implements querying and searching in the triple store. The triple store organization and the results of the queries and searching retrieved back via the Data API will ensure that the answers contain direct links to the sources from where the answer and the knowledge has been inferred.

CENDARI Data API authorization component reuses the authorization component of CKAN. Rather than originally implemented local user authentication in CKAN, CENDARI team extended the CKAN implementation to connect to the DARIAH authentication service via Shibboleth, in collaboration with WP8. Thus all queries or updates will undergo same authorization procedure and the privileges and access writes will have a central point of management.

# 3 Summary

The CENDARI Data integration infrastructure and repository deliver inclusive and open data integration platform. New services can be added (or existing services extended) with smaller software modifications. These services may range from transformation, more intensive data processing, up to validation and data dissemination.

The CENDARI data integration platform implemented part of these services during the project lifetime. Strong focus has been put on development of open architecture, thus the usage of existing open source components has been seen as an enabler for further developments of additional services and domain specific applications.