EGI-Engage

Final version of Multi-Source Distributed Real-Time Search and Information Retrieval Application

D6.5

Date	09 Feb 2016
Activity	SA2
Lead Partner	GWDG
Document Status	FINAL
Document Link	https://documents.egi.eu/document/2665

Abstract

This deliverable describes the Multi-Source Distributed Real-Time Search and Information Retrieval Application pilot of the DARIAH Competence Centre of the EGI-Engage project. DARIAH is a European infrastructure for arts and humanities scholars working with computational methods. The Multi-Source Distributed Real-Time Search and Information Retrieval Application aims at bringing a next-generation search and data retrieval platform sourced by different systems and heterogeneous data to the users from the arts and humanities community. The document introduces the underlying use case, the base technology, the architecture and current implementation of the Multi-Source Distributed Real-Time Search and Information Retrieval Application. This implementation is still under development within the DARIAH Competence Centre and it is planned to be released within Q2/2016.

The EGI-Engage project is co-funded by the European Union (EU) Horizon 2020 program under Grant number 654142 http://go.egi.eu/eng

COPYRIGHT NOTICE

The EGI-Engage project is co-funded by the European Union Horizon 2020 programme under grant number 654142.

DELIVERY SLIP

	Name	***Partner/Activity***	***Date***
From:	Philipp Wieder	GWDG / SA2	15.02.2016
Moderated by:	Małgorzata Krakowian	EGI.eu/NA1	18.02.2016
Reviewed by	Gergely Sipos	EGI.eu-SZTAKI/SA2	29.02.2016
Approved by:	AMB and PMB		3.03.2016

DOCUMENT LOG

Issue	***Date***	***Comment***	***Author/Partner***
v.1	12.02.2016	Feedback incorporated from	Philipp Wieder / GWDG Tibor Kalman Oliver Wannenwetsch
v.2	15.02.2016	Full draft for external review	Philipp Wieder / GWDGs
FINAL	29.02.2016	Final version	Philipp Wieder / GWDGs

TERMINOLOGY

A complete project glossary is provided at the following page: http://www.egi.eu/about/glossary/

Contents

Executive summary

DARIAH[1] is a European infrastructure for arts and humanities scholars working with computational methods. It supports digitally enabled research as well as the teaching of digital research methods. The DARIAH Competence Centre[2] (DARIAH-CC) within the EGI-Engage project aims at broadening the service and support landscape of the DARIAH community through complementing existing offers with solutions from EGI-Engage. The purpose of the Multi-Source Distributed Real-Time Search and Information Retrieval Application (SIR) is to extend an existing tool used within DARIAH with capabilities to process documents produced by optical character recognition (OCR) and to offer the application within the EGI Federated Cloud (EGI FedCloud) infrastructure.

OCR processing within an information system is a difficult task. As such documents are essential input to various research questions in arts and humanities, the DARIAH Competence Centre aims at developing an information system pilot for the processing of such documents. According to the work plan of the DARIAH-CC, it was the objective to deliver both the software and the description of SIR by the end of February 2016. Unfortunately, it was not possible to finalise SIR completely by this deadline mainly due to technological changes related to the user interface technology. It is therefore planned to release the final version of SIR with a few months delay, within Q2/2016.

This document describes the first twelve months of work on the pilot covering the following content:

- A description of the OCR use case.
- The technical foundation of SIR including its current application within the DARIAH community.
- The overall architecture of SIR.
- A description of the current state of the application and an outline of the work to be done in the next months to finalise it.

[1] DARIAH – Digital Research Infrastructures for the Arts and Humanities. Online: http://dariah.eu/ Accessed: 08.02.2016.
[2] Competence centre DARIAH. Online: https://wiki.egi.eu/wiki/CC-DARIAH Accessed: 27.02.2016.

1 Introduction

The amount of data is growing rapidly in all research disciplines. In many areas, the number of relations between data and resources become an essential factor for successful research. Therefore it is necessary to develop strategies for science and research to safeguard and reuse unique data, to conserve relationships between data objects, and to verify research results. To realize these strategies, research data has to be loaded into an information system, which can cover the entire data lifecycle. Additionally, data availability and persistent identifiers need to be provided for long periods, during which hardware and software infrastructures are subject to change. Thus appropriate abstraction layers and sustainable service interfaces that follow open industry standards are a necessary choice.

For the arts and humanities community, which is represented in the Europe prominently by DARIAH, the above statement holds true in a particular sense. In contrast to other scientific communities, like e.g. high-energy physics, there is a younger tradition regarding the digital transformation in general and the use of digital information systems in particular. Therefore, DARIAH supports researches in various dimensions to manage the digital transformation, execute their projects, and use the services required for their daily work. We refer interested readers for more information to the DARIAH web site.

In the context of the DARIAH Competence Centre, which is embedded into the EGI-Engage project, an OCR-related use case has been chosen (see Section 2.1.1 for more details) to implement an information system pilot for the arts and humanities community and to prepare it for cloud-based deployment within the EGI Federated Cloud infrastructure.

The document at hand describes the service architecture including the use case, the technological foundation for the information system, and the high-level service architecture. This is the main contribution of this deliverable. The document also provides initial information on the software release and lays out future plans.

2 Service architecture

This section describes the service architecture of SIR. It introduces the OCR-related use case in Sub-section 2.1.1, provides technical insight regarding Common Data Storage Architecture (CDSTAR)[3], the system that provides the foundation to implement SIR, in Sub-section 2.1.2, and outlines the overall, high-level architecture of the SIR framework in Sub-section 2.1.3. The section concludes with a description of the dependencies of SIR with respect to other tools form the EGI landscape.

[3] Oliver Schmitt, Andreas Siemon, Ulrich Schwardmann, and Marcel Hellkamp. GWDG Object Storage and Search Solution for Research – Common Data Storage Architecture (CDSTAR). GWDG-Bericht Nr. 78, 2014, 65 p.

2.1.1 Use Case

The digitization of scanned and photographed books and newspapers is a very compute-intensive application, but is a necessary prerequisite for the usability of the information contained. One proven approach to deal with this use case is MapReduce[4] which can access information from large amounts of text and unstructured data in a parallelized and efficient manner. Furthermore, new computer-intensive processes in the area of OCR provide better recognition results. Improved algorithms for obtaining bitonal images and post-OCR processing with sophisticated statistical correlations achieve excellent results provided that sufficient computing power is available and accessible. The development of cloud technology provides a new foundation for the scalable operation of MapReduce. Improved tools for automatic provisioning of services and servers allow providing infrastructure and computing power in cloud systems quickly and dynamically. This creates the basis for the operation and exploration of systems for data mining, real-time analysis, and mass data processing in the digital humanities.

Big Data technologies such as MapReduce allow the accelerated execution of optical character recognition, processing, and quality assessment and they enable new levels of quality. Mapped to a MapReduce workflow, this is done in three steps: (i) *OCR-Pre-Processing*, (ii) *OCR-Processing*, and (iii) *OCR-Post-Processing* (see Figure 1). In the pre-processing step, the present in colour or grey-scale images are converted into binary images required by the OCR engine (called binarization). By modelling the pre-processing as the first *Map* step in the MapReduce workflow, different algorithms for binarization can be calculated in parallel for each image. The raster graphics generated in the first step then serve as input for the actual OCR-Processing. This is performed as a second Map step using other OCR engines. The result of the second step is then fed into the OCR-Post-Processing in a final *Reduce* step, in which an automatic review of the recognized characters is performed (lexical correction). In this review process, the recognized text is compared with encyclopaedias of the respective language used in the text and similarities of words are calculated to, for example, evaluate the flection of verbs. The process uses a two-staged approach with distinct matches found in word databases for words that have a distinct match and a probabilistic approach that uses a tree-based look-up on syllable structure that assigns recognition probability values for word fragments. The lookup of syllable trees is compute intensive and requires large amounts of memory to perform well.

The whole OCR workflow follows, as shown in Figure 1, the MapReduce paradigm and requires a modular process chain in a way that its components can be dynamically replaced. For the implementation of this use case it is therefore necessary to first define an intermediate format that bridges the different input and output parameters of the different OCR modules, allowing the combination of the modules.

[4] Jeffrey Dean and Sanjay Ghemawat. 2008. MapReduce: simplified data processing on large clusters. Commun. ACM 51, 1 (January 2008), 107-113. doi:/10.1145/1327452.1327492

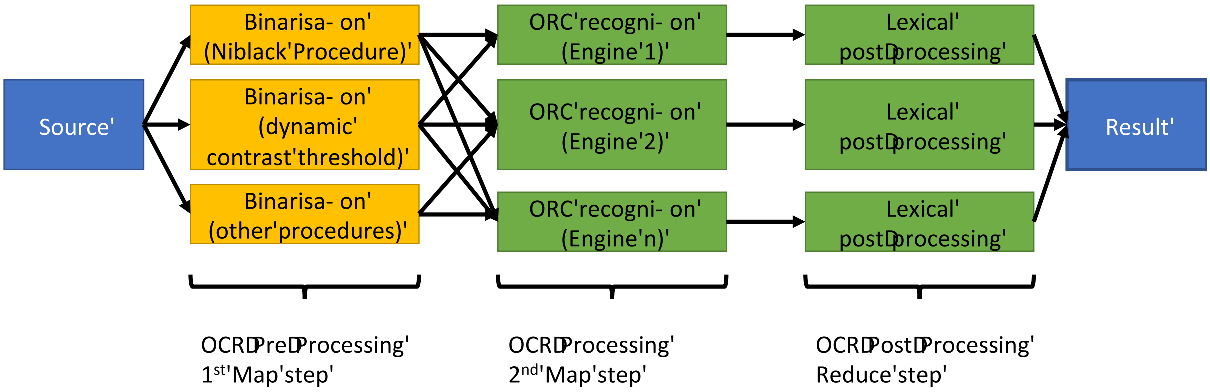
Source'
Binarisa- on' (Niblack'Procedure)'
Binarisa- on' (dynamic' contrast'threshold)'
Binarisa- on' (other'procedures)'
ORC'recogni- on' (Engine'1)'
ORC'recogni- on' (Engine'2)'
ORC'recogni- on' (Engine'n)'
Lexical' post□processing'
Lexical' post□processing'
Lexical' post□processing'
Result'
OCR□Pre□Processing' 1st'Map'step'
OCR□Processing' 2nd'Map'step'
OCR□Post□Processing' Reduce'step'

Figure 1 - Workflow showing Exploration and Quality Improvement of OCR-Scans

The realisation of the OCR use case provides a system that can be dynamically adapted to the respective research question by deploying the respective components for the different Map and Reduce steps. Through this, the researchers obtain customised and improved texts from the OCR process that serve as quality assessed sources for their research. Furthermore, through the integration of the system with cloud infrastructures, the whole framework will help the Arts and Humanities community to benefit from state-of-the-art technology.

2.1.2 Analysis of the base technology CDSTAR

The implementation of the SIR framework is based on CDSTAR, a tool which has been designed and developed by the EGI-Engage consortium member GWDG[5] (before EGI-Engage).

CDSTAR has a built-in custom object storage solution for science and research. This solution addresses the specific requirements of research data management according to the good scientific practice. The system integrates the ability of storing metadata along the research data in a flexible metadata schema that can be tailored for the specific use in different scientific disciplines. Additionally, the data objects that are stored in GWDG CDSTAR can be registered automatically at the EPIC Persistent Identifier (PID) service[6]. The EPIC service gives data sets a unique, globally resolvable identifier as an additional abstraction layer that allows citing data sets in scientific publications. A role-based security concept is also integrated into CDSTAR, which allows the protection of data sets with an individual set of permissions and rights for each user. Additionally, CDSTAR is capable to use SAML infrastructures such as the one provided by DARIAH in order to verify data access. This allows to use the data permissions for the infrastructure at a central data point provided by DARIAH and to use the DARIAH access credentials.

The CDSTAR object storage system allows the usage of different storage scenarios starting from small data sets to large data sets. In contrast to generic, industry offered object storage solutions, such as Amazon S3, CDSTAR allows a tight integration of metadata, permissions, and payload data into a single object tagged by a persistent identifier. This meets the requirements of research

[5] Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG). Online: http://www.gwdg.de. Accessed: 26.02.2016.
[6] EPIC - European Persistent Identifier Consortium. Online: http://pidconsortium.eu . Accessed: 06.02.2016.

better, where the description and long-lasting access to data are specific important requirements that remain unmet with commercial solutions yet. Additionally, it includes the choice of any storage backend that let customers select different venues for data storage, back scenarios and replication to remote data centres. Although CDSTAR follows the cloud paradigm, the researchers have control over their data and can use this storage solution to store and search their data.

2.1.2.1 Features and design principles

CDSTAR offers the following features:

- Integration of user-defined metadata along with the research data using flexible metadata schemata that can be tailored for the specific requirements of different scientific disciplines.
- Implementation of different storage back-ends.
- Provision of a stable RESTful Interface that transfers data over HTTP.
- Integrates of an enterprise-grade search engine based on Elasticsearch that operates on metadata as well as full text and indexes a wide range of file formats.
- A role-based security concept that allows the protection of data sets with an individual set of permissions and rights for each user.
- Use of a central permission system like the DARIAH Policy Decision Point (PDP).
- Support for SAML-usage in decentralized data access scenarios.

The design of CDSTAR is based on the following design principles:

- Provision of a customisable object storage solution for science and research.
- Application of a modular approach with appropriate abstraction layers and sustainable service interfaces that follow open industry standards.
- Application of industry-proven cloud technologies.
- Separation of front-end and back-end infrastructure to anticipate disruptive technology changes.

2.1.2.2 CRUD operations

CDSTAR provides storage for entities, such as binary files and text files. The five resources supported by the CDSTAR REST API are *objects, bitstreams, metadata, search, and access control* modification. Files (in the following called bitstreams) and metadata must belong to an object. CDSTAR offers two type objects plain objects for storing data and metadata as well as collection-objects for linking different objects. Objects support metadata that can be attached to objects as JSON-file. Using the REST-API plain objects can store up two million bitstreams per object and over four billion objects[7]. CDSTAR objects currently do not support partial updates of files, metadata, or access rights meaning that every update request has to upload an entire binary stream. Every object is marked by default with a persistent identifier and can be addressed through an URI.

[7] Dependent on the storage backend and its configuration.

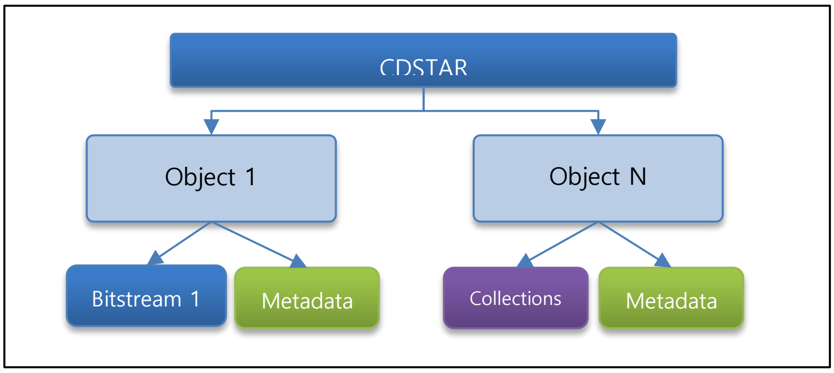

Figure 2 – Access to Bitstreams, Metadata and Collections

Actions on objects are performed by using HTTP methods that provide operations on the resources. For this, CDSTAR uses CRUD operations to describe all necessary actions that are applicable on REST-resources such as objects, bitstreams, collections, and object permissions (partly shown in Figure 2). The acronym CRUD consists of CREATE (HTTP-Post), READ (HTTP-Get), UPDATE (HTTP-PUT), and DELETE (HTTP-Delete).

2.1.2.3 Usage of CDSTAR in DARIAH

CDSTAR is already used in the DARIAH context[8] (see Figure 3), a fact that is also of interest for the DARIAH-CC project. We therefore describe briefly the implementation of the integration of the DARIAH Storage API.

The core of the support solution for the DARIAH community is, as depicted in Figure 3, the CDSTAR framework as described in Section 2.1.2.1. Following the design principle of a customised solution, we based the system on an unmodified CDSTAR implementation that has been extended with an implementation of the standardised DARIAH Storage API[9]. Through the addition of this specific service route, backwards compatibility is granted and the solution works with all DARIAH applications that follow the Storage API recommendation. This includes Liferay-based portal solutions or the DARIAH Geo-Browser.

[8] Tibor Kálmán, Dana Tonne, and Oliver Schmitt, Sustainable Preservation for the Arts and Humanities, New Review of Information Networking 20, 1-2 (2015), pp. 123–136. doi:10.1080/13614576.2015.1114831

[9] DARIAH Storage API – A Basic Storage Service API on Bit Preservation Level. Online: https://wiki.de.dariah.eu/download/attachments/10618851/DARIAH-Storage-API-v1.0_final.pdf Accessed: 08.02.2016.

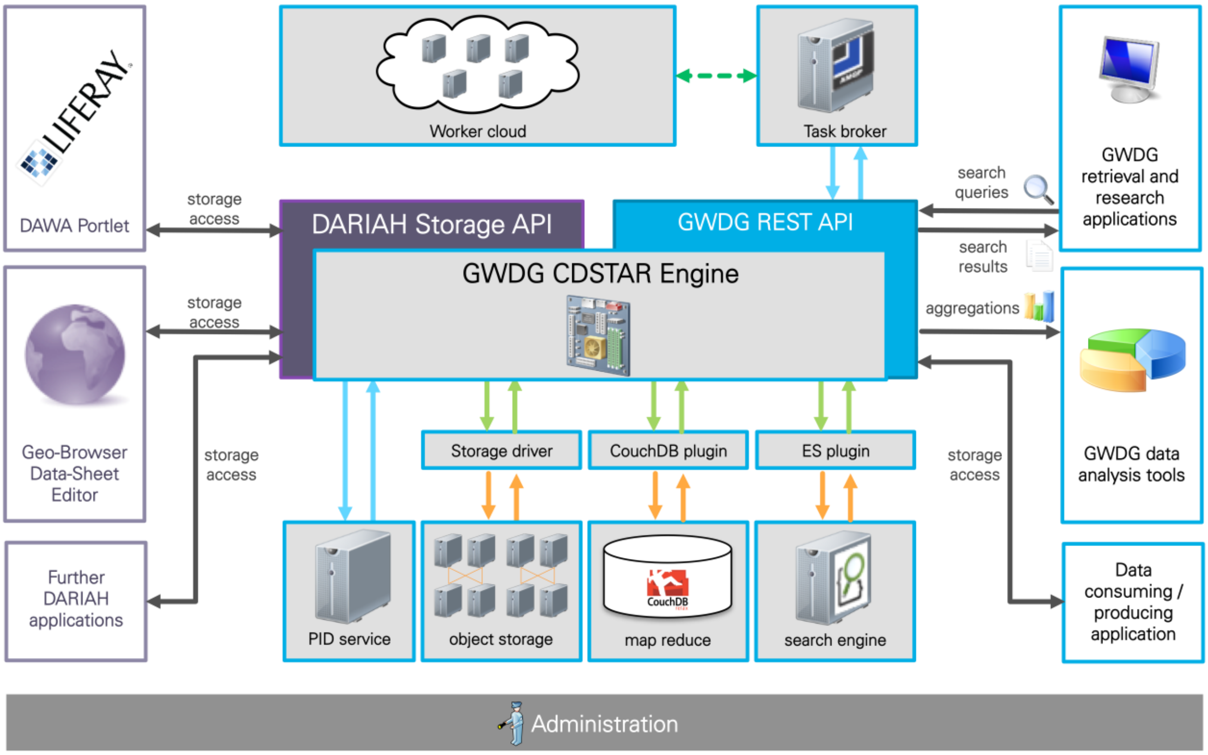

Figure 3 - CDSTAR: Support for the DARIAH Community

As the DARIAH Storage API, which offers particular REST calls following the DARIAH recommendation, has been added in addition to the native RESTful API, which has been introduced in Section 2.1.2.2 and is called *GWDG REST API* in Figure 3), all already available functions of CDSTAR can be used by DARIAH users. This includes among others full text search.

As every file upload can be processed by CDSTAR's search-engine plugin, search capabilities over all files can be easily added to existing applications. The search interface is reachable through the native API that runs along the DARIAH Storage API. The DARIAH API also features access verification through the DARIAH PDP and SAML-based access authorization for user data access provided with the DARIAH infrastructure.

2.1.3 High-Level Service Architecture

The high-level architecture of the SIR framework is depicted in Figure 4. The foundation of this framework is, as in the case of the DARIAH solution, CDSTAR. It is integrated into SIR through its native RESTful API and uses a storage instance to cache the use case-related data. Additional components that are necessary to realise the use case described in Section 2.1.1 and that are developed or customised in the DARIAH-CC are the following (each marked with an ID in the figure):

1. ETL Ingester
2. Analytics Plugin
3. Elasticsearch Plugin

4. Analytics Platform and Search Engine Cluster
5. Recovery Tool (optional)
6. Advanced Query Hub
7. Presentation Layer

The components are described in the following sub-sections.

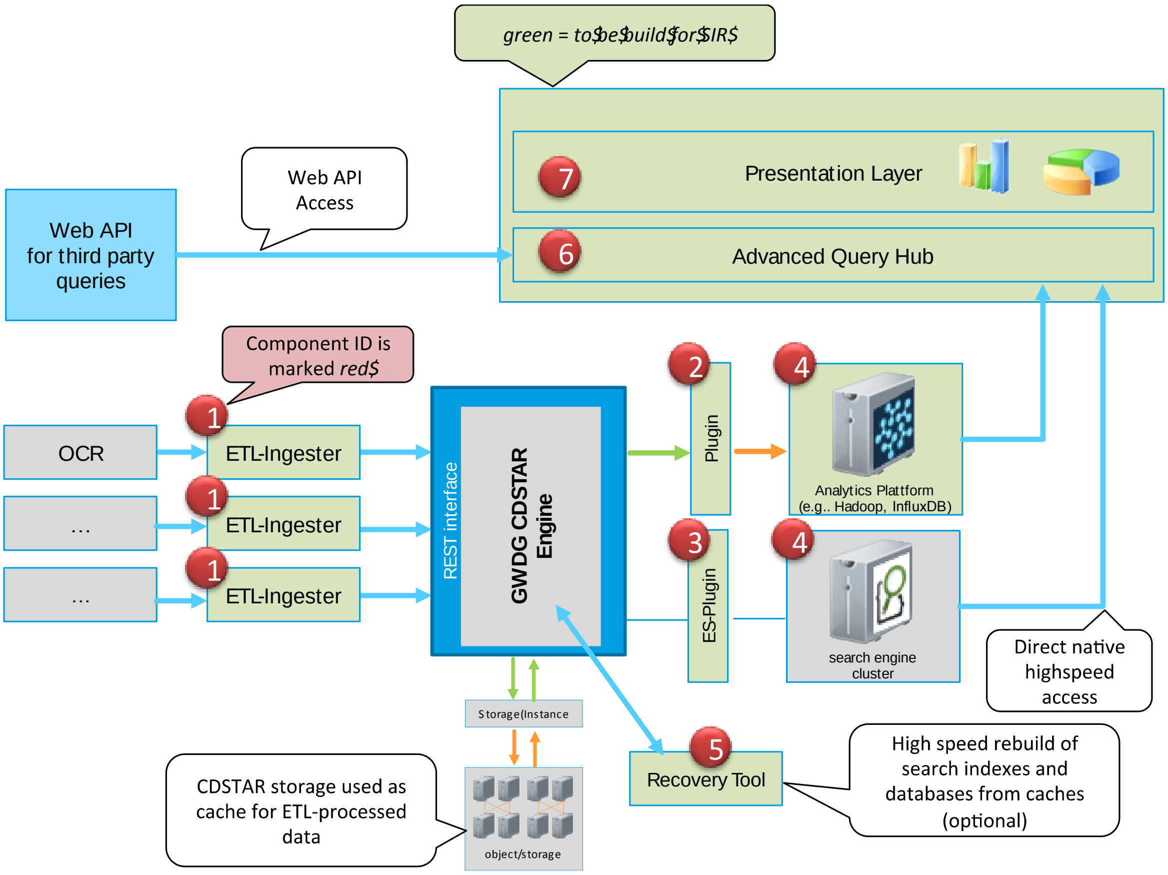
green = to$be$build$for$SIR$
Presentation Layer
7
Advanced Query Hub
6
Web API Access
Web API for third party queries
Component ID is marked red$
OCR
…
…
1
1
1
ETL-Ingester
ETL-Ingester
ETL-Ingester
REST interface
GWDG CDSTAR Engine
2
Plugin
4
Analytics Plattform (e.g.. Hadoop, InfluxDB)
3
ES-Plugin
4
search engine cluster
Direct native highspeed access
Storage(Instance
object/storage
CDSTAR storage used as cache for ETL-processed data
5
Recovery Tool
High speed rebuild of search indexes and databases from caches (optional)

Figure 4 – The High-level Architecture of the SIR Framework

2.1.3.1 ETL Ingester

The ETL Ingester realises the Extract, Transform, and Load process as shown in Figure 1 and provides, in particular, the framework to adapt the process dynamically to different research requirements and questions. OCR data is processed by the ETL Ingester to "normalise" the data and to improve the data quality. After this has been done, the data is fed into CDSTAR via the native RESTful API and cached in a storage instance. This is a core component of the whole framework as it has significant implications on the data quality and thus on the quality of the research results. It is designed in a way that allows the dynamic application of different algorithms without changing the overall process. Through this, the ETL process can be adapted to the input OCR material and the whole system can benefit from future research and development results.

2.1.3.2 Analytics Plugin

One of the design goals of the CDSTAR development is the “Application of a modular approach with appropriate abstraction layers and sustainable service interfaces that follow open industry standards” (see Section 2.1.2.1). Mapped to the application of analytics platforms this implies the necessity to implement a plugin that enables CDSTAR to feed data into different analytics platforms. This function is realised through the Analytics Plugin.

2.1.3.3 Elasticsearch Plugin

The same design goal as the one referred to above motivates the usage of an elasticsearch plugin. As CDSTAR offers a native API that should be as stable as possible it is essential to shield the CDSTAR implementation from changes within the Elasticsearch software[10]. This is realised through the Elasticsearch Plugin. In case of SIR, this component has to be customised to serve the particular use case.

2.1.3.4 Analytics Platform and Search Engine Cluster

Within SIR, Apache Hadoop[11] serves as an analytics platform and elasticsearch as the enterprise search solution.

Apache Hadoop is a framework that is designed to process large data sets on COTS hardware in a parallel fashion. It includes a variety of different components, which fulfil different tasks and which can be customised depending on the requirements of the particular use case. MapReduce has already been mentioned (see Section 2.1.1) as a component for writing applications that process large amounts of structured and unstructured data. Within Hadoop, this data is stored on the Hadoop Distributed File System (HDFS)[12]. On top of it, the YARN[13] operating system provides resource management and a central platform for data processing engines like MapReduce. Other engines can be used and customised depending on the use case and the research question behind it.

Elasticsearch is a distributed search and analysis platform for large data sets, which meets the requirements of the Arts and Humanities community as a search technology. It has several extensions and interfaces for connecting databases, web services, and analytics methodologies. The realization of complex searches with faceting is possible as well as the full text search in common file formats such as XML, JSON, CSV, PDF, or Microsoft Office documents. elasticsearch follows the well-known NoSQL approach with JSON formatted documents for indexing of data and metadata. In order to distribute load and speed up search, elasticsearch can be scaled flexibly depending on the use case.

[10] Elasticsearch – Search & Analyze Data in Real Time. Online: https://www.elastic.co/products/elasticsearch Accessed: 06.02.2016.
[11] Hadoop – Welcome to Hadoop. Online: https://hadoop.apache.org Accessed: 06.02.2016.
[12] HDFS Architecture Guide. Online: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html Accessed: 08.02.2016.
[13] Apache Hadoop Yarn. Online: https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html Accessed: 08.02.2016.

2.1.3.5 Recovery Tool

The recovery tool is used to rebuild the search index in case the search index stored in the search engine has irrecoverable been damaged. This case is very unlikely, as elasticsearch is based on self-healing cluster architecture, but a viable option for recovery could to be integrated to lower the risk of data loss. In case of recovery, all objects of CDSTAR are queued into the recovery tool and the re-indexing process is started on multiple threads to reach better speed. As multiple cluster nodes of elasticsearch can participate in the re-indexing, the time for a full index rebuild can be reduced significantly.

The Recovery Tool is an optional component that has not been fully implemented at the time of writing this document. It is planned to implement it during the life-time of the DARIAH-CC and to also offer it to the Arts and Humanities community.

2.1.3.6 Advanced Query Hub

The Advanced Query Hub integrates the different sources of information and processes them in a way that allows their presentation. Furthermore, it allows the integration of other, dynamic data sources that can be accessed via a web API. Twitter is a good example for such a source that can be accessed through a REST API[14]. It is used for various research projects as a data source and can add viable information.

2.1.3.7 Presentation Layer

The Presentation Layer is the single frontend to the researcher. Although it is not the purpose of the DARIAH-CC to develop production ready software (in the sense of TRL 7 to TRL 9), there is the necessity to have more than a mock up of a user interface to SIR.

2.1.4 Integration and dependencies

The SIR framework is released as a deployable software package that can be hosted on any cloud provider. To run it, a Java JDK (version >= 1.8), Maven (version >= 2), and Git are required.

Cloud operators from any of the NGIs could setup local instances of the system to serve specific arts and humanities communities. These communities can feed local data into their deployment. Instance operators would benefit from the centralised maintenance of the software by GWDG.

It is therefore possible to operate the SIR application in a ‘Software as a Service’ fashion, so that communities within DARIAH or EGI could gain access to it without the need of installation. An operation within the EGI Federate Cloud is planned.

[14] REST APIs | Twitter Developers. Online: https://dev.twitter.com/rest/public Accessed 08.02.2016.

3 Release notes

3.1 Requirements covered in the release

The following functions fulfil the requirements derived from the use case describe in Section 2.1.1:

- Development of ETL-Ingester
- Development of Analytics Plugin
- Customisation of CDSTAR Elasticsearch Plugin
- Development of Advanced Query Hub

The following function has not yet been fully implemented:

- Presentation Layer

The following function has not yet implemented:

- Recovery Tool (which was an optional component according to the development plan)

3.2 Roadmap towards the final version of SIR

The final version of the SIR pilot should be delivered end of February 2016 according to the EGI-Engage description of work. Unfortunately, the work was not finalised at the envisaged point in time mainly due to a change of the user interface technology used for the core CDSTAR development towards Drupal 8. This implied that initial work on the presentation layer was obsolete and the plans had to be updated. The final version of SIR will be developed according to the following roadmap:

- Design of the user interface based on Drupal 8 (January 2016 to March 2016)
- Implementation of the user interface (March 2016 to May 2016)
- Final testing of SIR (May 2016)
- Release of SIR (June 2016)
- Deployment of SIR within the EGI Federated Cloud infrastructure (Q3/2016)

4 Feedback on satisfaction

The SIR framework has not yet been fully tested by the user community due to the aforementioned delays. The DARIAH-CC will promote and demonstrate the system during PY2 alongside with the other demonstrator services that are prepared by the CC. Key milestones for this are:

- The CC started the writing of a short document that will provide ‘value proposition’ for Digital Humanities research about all of the services/demonstrators that are established by the CC in the EGI context.

- The CC submitted a workshop proposal to the 'Digital Humanities'[15] conference (to be held in July 2016) to demonstrate the services/demonstrators and to engage with users for these.
- The CC is planning to apply for funding[16] from DARIAH to organise an e-infrastructure event relating to the theme 'Public Humanities'. If the proposal is accepted, then this event could be another opportunity to engage with researchers from the field.

5 Future plans

This deliverable describes the technical details of SIR and the current state of the implementation. It is planned to finalise the SIR pilot according to the architecture design (excluding the optional Recovery Tool as this is not a mandatory functionality for the application) following the roadmap outlined in Section 3.2.

The future work furthermore includes additional analysis of the presentation layer in particular the evaluation of the integration with other gateways or gateway technologies used within DARIAH CC (Liferay, WS-PGRADE, gLibrary, and Semantic Search Engine).

[15] Digital Humanities: http://dh2016.adho.org

[16] https://dariah.eu/news/articles/details.html?tx_news_pi1%5Bnews%5D=291&tx_news_pi1%5Bcontroller%5D=News&tx_news_pi1%5Baction%5D=detail&cHash=3d11854220992cd8ac48490423e4f0aa