

DELIVERABLE

Project Acronym: DCH-RP

Grant Agreement number: 312274

Project Title: Digital Cultural Heritage Roadmap to Preservation

D5.4 Report on second Proof of Concept

Revision: Final

Authors:

Michel Drescher, EGI.eu

Contributors:

Andres Uueni, EVKM
 Rosette Vandebrouke, BELSPO
 Claus-Peter Klas, FTK e.V.
 Felix Engel, FTK e.V.
 Maciej Brzeźniak, PSNC
 Sara di Giorgio, ICCU
 Roberto Barbera, INFN

Reviewers:

Borje Justrell, RA
 Raivo Ruusalepp, EVKM

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
P	Public	X
C	Confidential, only for members of the consortium and the Commission Services	

Revision History

Rev.	Date	Author	Affiliation	Description
Skel.	16.6.2014	M. Drescher	EGL.eu	Skeleton document to collect contributions
0.1	23.6.2014	M. Drescher	EGL.eu	Added contribution for Exp. 1
0.2	25.6.2014	M. Drescher	EGL.eu	Added contrib for Exp.5 & Exp. 3
0.3	27.6.2014	M. Drescher	EGL.eu	Added contrib for Ex. 2, exec sum.
0.4	29.6.2014	M. Drescher	EGL.eu	Added contrib. for Exp. 4, conclusions
0.5	30.6.2014	M. Drescher	EGL.eu	Worked in reviewer comments
1.0	01.07.2014	C. Prandoni	Promoter	Final version for submission

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	4
1 INTRODUCTION	6
1.1 OBJECTIVES OF THE DELIVERABLE.....	7
1.2 STRUCTURE OF THE DOCUMENT	8
2 EXPERIMENT 1: EXPLORE THE SCAPE PROJECT'S MATCHBOX TOOL.....	9
2.1 RATIONALE	9
2.2 SETUP & DESCRIPTION OF TOOLS	9
2.3 EXPERIMENTATION.....	12
2.4 RESULTS & NEXT STEPS	14
3 EXPERIMENT 2: INVESTIGATE THE SCIDIP-ES PROJECT'S HAPPI PLATFORM.....	15
3.1 RATIONALE	15
4 EXPERIMENT 3: EVALUATE EUDAT STORAGE SERVICES.....	16
4.1 RATIONALE	16
4.2 SETUP & DESCRIPTION OF TOOLS	17
4.3 EXPERIMENTATION.....	19
4.4 RESULTS & NEXT STEPS	19
5 EXPERIMENT 4: RE-EVALUATE THE ECSG AND REMOTE GRID/ CLOUD STORAGE SERVICES FROM POC1	20
5.1 RATIONALE	20
5.2 SETUP & DESCRIPTION OF TOOLS	20
5.3 EXPERIMENTATION.....	20
5.4 RESULTS & NEXT STEPS	21
6 EXPERIMENT 5: A LONG-TERM DATA PRESERVATION PLATFORM	25
6.1 RATIONALE	25
6.2 SETUP & DESCRIPTION OF TOOLS	25
6.3 EXPERIMENTATION.....	27
6.4 RESULTS & NEXT STEPS	27
7 CONCLUSION	28
8 REFERENCES	31
ANNEX 1: INSTALLING AND TESTING MATCHBOX.....	32

EXECUTIVE SUMMARY

The second Proofs of Concept phase in the DCH-RP project takes into account both the results of the first Proofs of Concept phase as reported in D5.3, and the DCH-RP Roadmap to Preservation as iteratively developed through its first study in D3.1, and the intermediate version provided in D3.4. In turn, the results of the second Proofs of Concept phase will inform and contribute to the final roadmap document D3.5 due in September 2014.

Preparations for the experiments started in mid December 2013 with a first conference call between prospective experiment leaders and participants, and the final list of five experiments was agreed and approved at the fourth project plenary meeting in Catania, Italy, in January 2014. At the same time, MoUs were agreed and signed with key FP7 projects to underpin and secure support for the experiments

Focussing more on integrated solutions and services, it is even more important to assess the software for the two most paramount requirements regarding the targeted users:

1. Ease of use of the tool or service for the *end user*
2. Ease of installation/provisioning for small IT departments or IT-experienced individuals.

The experiments cover a wide variety of solutions that have the potential to implement parts of the DCH roadmap to a satisfactory level, or with reasonable integration effort.

Experiment 1 explores a tool (“Matchbox”) developed by the SCAPE project that allows automating the task of finding duplicate images in a set of files. “Data hygiene” activity is a necessary filter for diligently preparing a dataset for archiving, and for regular quality assurance and repository certification for preservation.

Experiment 2 will look at the HAPPI (Handling Authenticity Provenance and Persistent Identifiers) service developed by the SCIDIP-ES project: Cultural data is often included time and again over a long period of time in various projects, which raises a number of needs and requirements as follows:

- Digital asset authenticity – establishing and maintaining the originality of the asset
- Data provenance – Keeping a trail of data usage events for audits and data usage indication
- Data reference persistence and validity – Idempotent data reference/identifier resolution over time and space to the correct storage location

Experiment 3 assesses a combination of services provided by the EUDAT project (B2SHARE and B2SAFE) in combination with a service (Platon) provided by PSNC to its national digital libraries and archives. The aim is to evaluate EUDAT’s services for curating and publishing a research community’s digital assets, in DCH-RP’s case the preservation of digitised and born-digital cultural heritage.

Experiment 4 is investigating some of the results of the experiments in the first Proofs of Concept phase provided in D5.3. More specifically, this experiment revisited the use case of uploading digital assets to a remote Grid/Cloud infrastructure in conjunction with the e-Cultural Science Gateway (eCSG) developed by INFN-Catania. Including federated identity management and AAI into this experiment, this experiment is addressing two of the main outcomes of the previous experiment in the first PoC phase.

Experiment 5 concludes the second PoC phase with the aim of assembling a general-purpose digital preservation platform implementing a Service oriented Architecture (SOA). The focus of this experiment lies on reducing the total cost of ownership (TCO) of such a preservation platform through integrating as many generic services as possible, implementing as many preservation-specific standards as necessary, and addressing the needs of as many user communities as is feasible. In collaboration with the APA (through the APARSEN project) this experiment will also explore how an external, independent service provider might offer services around such a platform to the target market while integrating underpinning services delivered by, for example, EGI or EUDAT, or other suitable infrastructure providers.

1 INTRODUCTION

Over the course of the DCH-RP project, a number of activities have contributed to the inception and subsequent updates of the DCH sector's roadmap to digital preservation. Beginning with a study on how such a roadmap might look like and what its goals should be [R 1], the first phase of Proofs of Concepts [R 5] provided Work Package 3 with feedback on practical experimentation of a number of tools and services that are already used in daily activities similar to data preservation, or look promising to include in the Roadmap for Preservation. In December 2013 an intermediate version of the roadmap was published [R 2], which further developed the ideas and concepts described in D3.1 into a more concise roadmap for preservation for the DCH sector in Europe.

Based on the proceedings of the intermediate roadmap a number of strategic MoUs were signed with projects working in the same field or producing tools and services that could be used for preservation of digital cultural artefacts (e.g. with EUDAT, APARSEN, OpenAire, SCIDIP-ES), in order to secure general collaboration and support in conducting experiments in DCH-RP's second Proof of Concept phase in Work Package 5. These activities were closely coordinated with the planning for the second Proof of Concept phase in the project. During a phone conference on 17 December 2013¹ an initial set of 10 experiments were proposed to the consortium. Eventually, at the fourth project plenary meeting on 20-21 January in Catania Italy, the consortium took a strategic decision [R 6] to focus on five experiments, addressing actions identified in the then current intermediate roadmap [R 2] (primarily chapter 5.2), and specifically recommendation 1 given in Deliverable 4.1 [R 4]: "1) Adapt the use case scenario described in Chapter 5 to be tested and evaluated as a proof of concept in WP5".

The following vice experiments were chosen since they address at large important aspects and responsibilities of the DCH roadmap: Experiments one, two, three and four primarily focus on specific tools and services that implement key preservation capabilities (duplication detection, metadata management & data discovery, and data storing, sharing & replication) in either singular tools such as Matchbox, or tightly integrated services such as HAPPI. These experiments focus on functional capabilities of existing services and how they map into the roadmap, such as automatic metadata extraction and capture, authenticity and integrity of data, backup and restore (all D5.3 page 27.ff) Experiment 5 however, looks at organisational and political issues while providing representational preservation services integrated in a platform: The conceptualised platform addresses issues such as virtualisation, distributed systems, cross-sector integration and allows studying governance and business models for a platform that is provided as a services from a different, independent stakeholder to the DCH community at large.

1. Experiment 1: Explore the SCAPE project's Matchbox tool for detecting duplicate images
2. Experiment 2: Investigate the SCIDIP-ES project's HAPPI platform for data provenance, authenticity and identification
3. Experiment 3: Evaluate EUDAT storage services
4. Experiment 4: Re-evaluate the eCSG and remote Grid and Cloud storage services from PoC1
5. Experiment 5: A Long-term Data Preservation Platform

¹ <https://indico.eji.eu/indico/conferenceDisplay.py?confId=1980>

The overall experimentation criteria for the second phase of Proofs of Concepts are even more focussing on requirements and needs of the expected end user: The expected users of the tools and services are *digital librarians/archivists and digital preservation specialists*. These are by and large not IT savvy beyond their daily use of computers – and they neither have to be, nor are expected to have such a skill set. Their required key skills are more of scholarly nature, relying on small institute IT departments or individuals experienced in administering computers to provision the IT infrastructure they need: It is not necessary for digital archivists to know how software works; instead they need to know how to use it well.

Hence the two focal objectives of experimentation are, in order of importance:

1. Ease of use of the tool or service for the *end user*
2. Ease of installation/provisioning for small IT departments or IT-experienced individuals.

Participation in the five experiments was based on partner availability as well as the backing memory institutes interest. The following table indicates participation of partners and external institutes in these experiments. Membership of partner projects with MoUs in force is indicated where applicable.

Partner (affiliated FP7 Project)	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5
KANUT	X				
BELSPO	X		X		
Engineering Italia (SCIDIP-ES)		X			
PSNC			X		
RA			X		X
ICCU				X	
INFN-Catania				X	
GARR				X	
FTK e.V. (APARSEN)					X
Collections Trust					X
Editeur					X

1.1 OBJECTIVES OF THE DELIVERABLE

The initial planning of Work Package 5 and the timing of its deliverables envisioned a full and complete account of conducted experiments in the second Proof of Concept phase to be published in this document. However, as often is the case reality required a change of plans and the project consortium decided to initially focus on securing external project collaboration and support to conduct the experiments since it realised that available funding within DCH-RP

would not nearly cover the desired and necessary diligence in executing the experiments on its own.

As a result, this deliverable in its current state will only provide a snapshot of the experimentation activities that took place until the time of writing. All project partners are fully aware of that an update of this deliverable is required towards the end of the project, when the experiments being carried out are finished. This understanding and agreement is documented in the minutes of the fifth project plenary meeting [R 7] on 24-25 April 2014 in Tallinn, Estonia.

1.2 STRUCTURE OF THE DOCUMENT

Following this introduction, the following five sections (2 – 6) describe the current experiments, each chapter following as closely as possible the same structuring:

- **Rationale:** Providing reasons why the experiment was conducted, the gaps and requirements it is tackling, including a brief numeration of the specific objects of the respective experiment.
- **Setup & Description of tools:** Specific references to tools and how they were set up allow for better reproduction of the experiment if the need arises. Though inspired by it this is not a scientific experiment where a meticulous description of the experiment methodology and setup is necessary.
- **Experimentation:** Provides a description of the actual (perhaps formalised) tests and trials that are planned for this experiment and how these would be conducted.
- **Results & Next steps:** Since this is a snapshot account of the experiments, some may not report any results yet. Future plans will accordingly accommodate for conducting the experimentation, or in cases where results exist (even if preliminary), the next steps will describe the future course of action as planned or altered in accordance with the results.

The document concludes with a brief analysis of the results achieved so far in section 7.

2 EXPERIMENT 1: EXPLORE THE SCAPE PROJECT'S MATCHBOX TOOL

2.1 RATIONALE

One recurring issue in archiving digital assets is unwanted duplication of content. This applies to any type of digital objects, be it audio files (e.g. recordings of fables that are passed orally from generation to generation), quantitative research data (e.g. SPSS data sources, Excel sheets, etc.), video interviews (e.g. for sociological studies) or images (e.g. a digitised copy of the Rosetta stone).

As a representative for digital object duplication detection tools, the SCAPE² project's "Matchbox" tool allows detecting duplicates in digital images.

Matchbox is tried out and tested in this first experiment. KIK-IRPA (Belgium) and KANUT (Estonia) participated in this experiment.

The single overarching objective was to test:

- Ease of installation of the tool,
- Ease of use for digital librarians and archivists,
- Tool accuracy in detecting duplicate images.

2.2 SETUP & DESCRIPTION OF TOOLS

The idea of the Matchbox is that there are numerous situations in which you may need to identify duplicate images in collections, for example:

- Ensure that a page or book has not been digitised twice
- Discover whether a master and service set of digitised images represent the same set of originals
- Confirm that all scans have gone through post-scan image processing.

Checking to identify duplicates manually is a very time-consuming and error-prone process. Matchbox aims to automate this process.

Matchbox is an open source tool which:

- Provides decision-making support for duplicate image detection in or across collections
- Identifies duplicate content, even where files are different (in format, size, rotation, cropping, colour-enhancement etc.), and if they have been scanned from different original copies of the same publication
- Applies state-of-the art image processing works where OCR will not, for example images of handwriting or music scores
- Is useful in assembling collections from multiple sources, and identifying missing files.

Matchbox provides the following benefits:

- Automated quality assurance
- Reduced manual effort and error rate
- Saved time
- Lower costs, e.g. storage, effort

² <http://www.scape-project.eu/>

- Open source, standalone tool. Also as Taverna component for easy invocation
- Invariant to format, rotation, scale, translation, illumination, resolution, cropping, warping and distortions
- May be applied to wide range of image collections, not just print images.

Documentation for Matchbox indicates that a preconfigured VM is available, but this was not suitable for this project. Also, pre-compiled binary packages were available, but only for AMD64 compatible 64bit architecture. Since the available test infrastructure supports only 32 bit, Matchbox and all its software dependencies had to be compiled and installed from scratch.

Annex 1 provides detailed instructions on how this was accomplished; this section however provides a summary of this.

2.2.1 Compiling and installing Matchbox

Matchbox is a command-line tool that uses OpenCV³ for the heavy lifting of the tasks ahead – it might be considered as a wrapper around OpenCV. OpenCV is the most popular and advanced code library for Computer Vision related applications today, spanning from many very basic tasks (capture and pre-processing of image data) to high-level algorithms (feature extraction, motion tracking, machine learning). It is free software and provides a rich API in C, C++, Java and Python. Other wrappers are available. The library itself is platform-independent and often used for real-time image processing and computer vision. OpenCV has already lot of interesting developments like face detection, similar object finder and etc. , see also screenshots below.

Naturally, the installation of Matchbox comprises of the following four phases:

1. Installing a build environment
2. Installing Python
3. Compiling and installing OpenCV
4. Compiling and installing Matchbox

While this appears to be easy enough, the actual details of these four phases require regular and frequent ICT knowledge and expertise in building software and satisfying its dependencies – skills that are certainly not present nor required for the typical digital archivist or librarian working in memory institutes. The following table provides an indication of the complexity of dependencies for each of these steps.

1. Installing a build environment	
GCC	The GNU Compiler Collection includes front ends for C, C++, Objective-C, Fortran, Java, Ada, and Go, as well as libraries for these languages (libstdc++, libgccj,...). GCC was originally written as the compiler for the GNU operating system. The GNU system was developed to be 100% free software, free in the sense that it respects the user's freedom.
Build-essential	This package contains an informational list of packages which are considered essential for building Debian packages. This package also depends on the packages on that list, to make it easy to have the

³ An open source tool for computer visualisation

	build-essential packages installed.
G++	Released by the Free Software Foundation, g++ is a *nix-based C++ compiler usually operated via the command line. It often comes distributed with a *nix installation, so if you are running Unix or a Linux variant you likely have it on your system.
CMAKE	<p>CMake is the cross-platform, open-source build system. CMake is a family of tools designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files. CMake generates native makefiles and workspaces that can be used in the compiler environment of your choice.</p> <p>Important dependencies: libarchive 3.1.2, curl 7.36.0, libboost (any version)</p> <p>Optional: CMake GUI</p>
2. Install Python	
Python 2.7	<p>Matchbox has an explicit dependency on Python 2.7 while common contemporary Linux distributions provide much more recent versions of Python. For example, Ubuntu 14.04 LTS includes Python 3.4.0.</p> <p>This forces the user to install Python 2.7 manually, which introduces further complications.</p> <p>Important dependencies: libsqlite3-dev, sqlite3, bzip2 libbz2-dev</p>
3. Install OpenCV	
Build OpenCV	<p>OpenCV is a library that provides a wide variety of filters and detection algorithms, for which it makes extensive use of 3rd party libraries.</p> <p>Mandatory dependencies: build-essential, libgtk2.0-dev, libjpeg-dev, libtiff4-dev, libjasper-dev, libopenexr-dev, cmake, python-dev, python-numpy, python-tk, libtbb-dev, libeigen2-dev, yasm, libfaac-dev, libopencore-amrnb-dev, libopencore-amrwb-dev, libtheora-dev, libvorbis-dev, libxvidcore-dev, libx264-dev, libqt4-dev, libqt4-opengl-dev, sphinx-common, texlive-latex-extra, libv4l-dev, libdc1394-22-dev, libavcodec-dev, libavformat-dev libswscale-dev</p>
Configuring OpenCV	Before the actual build process can start, a number of build variables need to be initialised according to the local environment – Annex 1 provides more details on this step.
4. Install Matchbox	
CMake configuration	Building Matchbox is done using CMake (installed earlier). Annex 1 provides the important configuration options and values

Installing Matchbox	<p>Once properly configured and built, Matchbox can be installed by issuing the final command:</p> <p style="text-align: center;"><i>sudo make install</i></p>
----------------------------	--

Table 1: Installation process overview for Matchbox

2.3 EXPERIMENTATION

The developers of Matchbox suggest a standard workflow for duplication detection as follows:

1. Extract SIFTComparison features of all images
2. Train a visual vocabulary on the extracted features
3. Extract BoWHistograms using the vocabulary and all extracted SIFTComparison features
4. Create a similarity matrix for the collection using compare on all BoWHistogram features
5. Take the top-most similar images for each image and compare their SIFTComparison features
6. Set a threshold based on the retrieved data
7. Images with an SSIM exceeding the threshold are considered to be duplicates

The Matchbox command line offers following features:

```
$ python2.7 ./FindDuplicates.py
usage: FindDuplicates.py [-h] [--threads THREADS] [--featdir FEATDIR]
                        [--precluster PRECLUSTER] [--config CONFIG]
                        [--bowski BOWSIZE] [--sdk SDK] [--clahe CLAHE]
                        [--downsample DOWNSAMPLE] [--update] [--binary]
                        [--binaryonly] [-v]
                        dir {all,extract,compare,train,bowhist,clean}
```

With all necessary information at hand, Matchbox allows mass-scanning entire folders and object lists for duplicates as illustrated below:

```
$ python2.7 ./FindDuplicates.py /home/anz/Downloads/matchbox_data all
=== extracting features from directory /home/anz/Downloads/matchbox_data ===
... extracting features of dir /home/anz/Downloads/matchbox_data
... 213 files to process
[1 of 213] PMF_1603_0072_inv.jpg done
[2 of 213] PMF_1657_0665_inv.jpg done
[3 of 213] PMF_1603_0287_inv.jpg done
[4 of 213] PMF_1603_0349_inv.jpg done
[5 of 213] PMF_1657_0195_inv.jpg done
...
[127 of 213] PMF_1603_0079_inv.jpg done
[128 of 213] PMF_0928_0027_inv.jpg done
[129 of 213] JPEG error: Corrupt Image
[130 of 213] PMF_1603_0331_inv.jpg done
...
$
```

Some screenshots illustrate the look and feel of using OpenCV and Matchbox:

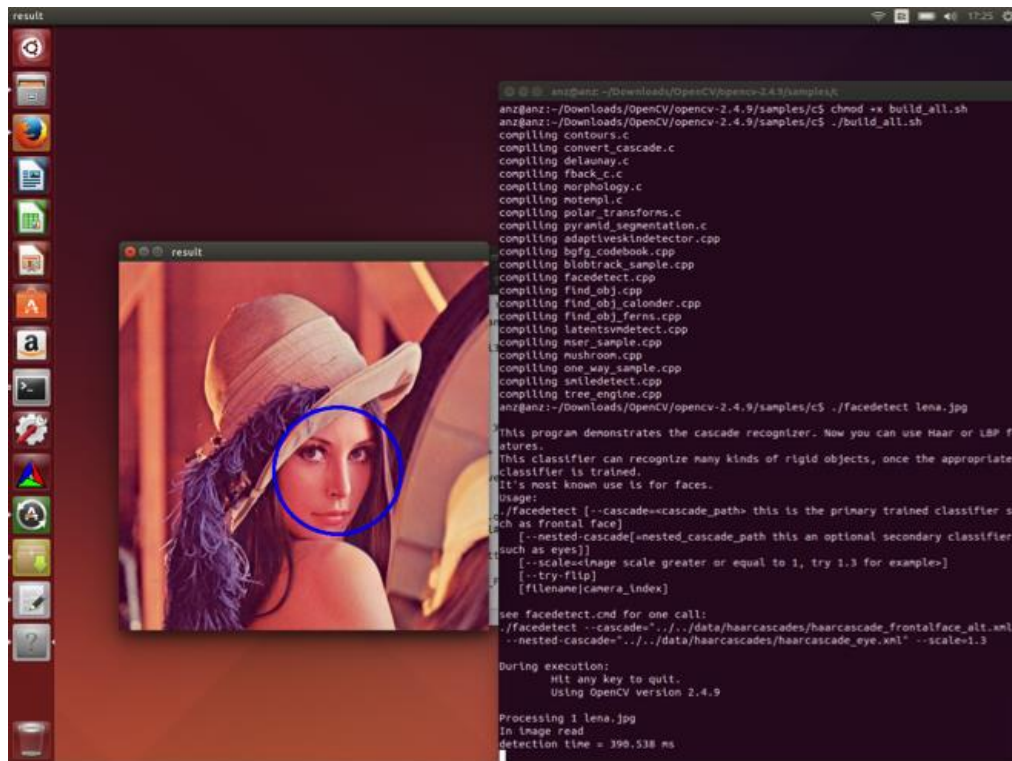


Figure 1: The OpenCV face detection feature

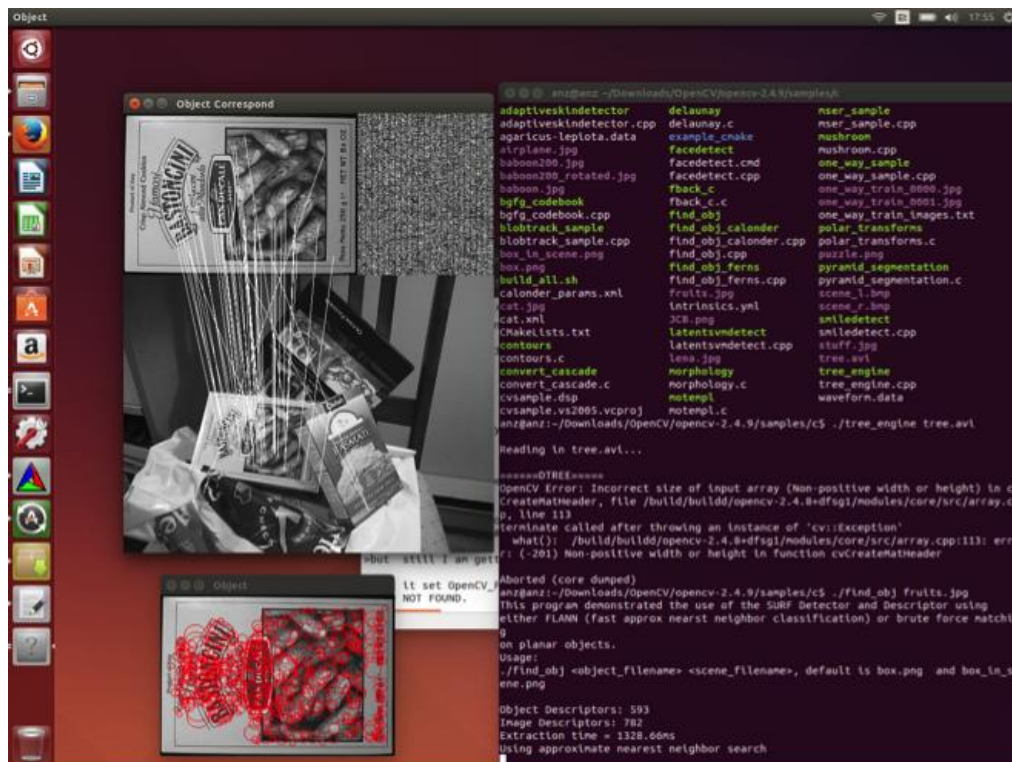


Figure 2: The OpenCV feature "detect similar objects" at work

2.4 RESULTS & NEXT STEPS

Due to the unexpected difficulties in providing Matchbox in a usable and testable environment, no results are yet available. However, in a basic test run of comparing 213 JPEG image files (or unknown file size, colour depth or image complexity) took approximately 130 minutes to finish. In a first approximation, this would translate to comparing one picture to 212 others in 36 seconds or, extrapolated, approximately 170 milliseconds to compare 2 images with each other.

Matchbox Tool works on command line, and doesn't has visual output, if necessary there are tools to parse and analyse xml files like `PMF_1603_0186_inv.jpg.BOWHistogram.feats.xml.gz` and `PMF_1603_0186_iinv.jpg.SIFTComparison.feats.xml.gz`.

At the present time Matchbox tool is available as standalone, free of charge open source tool (Apache License Version 2.0), which needs several developer tools and tests before final implementation.

At the time of writing this deliverable the Matchbox tool was only available a few days due to the complexity of building the tool from the raw programming into an executable version. Matchbox will be tested in the coming weeks by KIK-IRPA on one of their digital collections that was already chosen for carrying out the proofs of concept in DCH-RP.

3 EXPERIMENT 2: INVESTIGATE THE SCIDIP-ES PROJECT'S HAPPI PLATFORM

3.1 RATIONALE

For Digital preservation activities, it is important to also capture the digital object's so-called "evidence history", which forms a crucial part of the Preservation Descriptive Information according to the OAIS reference model.

The SCIDIP-ES project has developed the HAPPI toolkit that supports the digital archivist in collecting this part of the PDI, the evidence history of the digital artefact that needs archiving.

Since the fourth plenary meeting in Catania, project partners faced a number of constraints that hindered uptake for this experiment and testing HAPPI in a DCH-RP context.

The collaboration between the DCH-RP and the SCIDIP-ES projects is now underpinned by a mutually signed MoU that includes conducting this experiment; this activity will resume in July 2014.

4 EXPERIMENT 3: EVALUATE EUDAT STORAGE SERVICES

4.1 RATIONALE

The DCH-RP project aims to identify suitable models and tools for the governance, maintenance and sustainability of DCH data that can be effectively used by cultural institutions across Europe.

Several projects and infrastructures aim to address long-term preservation of scientific and cultural data.

At the European level, EUDAT⁴ provides a sustainable infrastructure based on the layer of common technologies, tools and services driven by user needs. It also backs the community- and domain-specific services. EUDAT currently serves several scientific communities including CLARIN, diXa, DRIHM, ENES, EPOS, INCF, LifeWatch, VPH and further expands to new communities.

On the national level NREs, data centres, computing centres and universities provide storage and data preservation, publication and sharing services to several communities including science and cultural sector. For instance Archiving Services⁵ of the PLATON project in Poland offer safely replicated storage space to academic and cultural institutions.

One of the use cases identified in DCH sector is the publication and sharing of the digital assets. Large organisations may have their own approaches, solutions and infrastructure for this purpose. However small organisations and so-called “citizen scientists” or “citizen curators” often struggle to get a reliable, adequate and affordable facilities for storing, publishing and sharing the data and metadata, as well as ensuring their long-term preservation.

4.1.1 Objectives

The aim of this experiment is to verify if and how services developed by EUDAT and nationally may address the needs of DCH community.

EUDAT’s B2SHARE⁶ service is used as a solution for data publication and sharing. It enables users to upload their data sets, enrich them with meta-data and assign persistent identifiers. It also supports keyword-based searching, digital assets preview as well as meta-data presentation and browsing.

The scope of the experiments also includes analysis of possible orchestration of EUDAT services with other services and infrastructures such as European or nationally provided facilities for long-term data storage and preservation.

At the European level, EUDAT’s B2SAFE⁷ service is considered. It ensures robust and reliable data replication and guards against data loss. It also improves data availability and locality across the continent. The service is offered by academic data centres. PSNC that represents EUDAT in DCH-RP project is one of such centres.

⁴ <http://www.eudat.eu/>

⁵ <http://www.platon.pionier.net.pl/online/archiwizacja.php?lang=en>

⁶ <http://www.eudat.eu/b2share>

⁷ <http://www.eudat.eu/b2safe>

At the national level, Archiving Services⁸ of the PLATON project in Poland offer 12,5PB of tape storage and 2PB of disk storage distributed in 10 locations across the country for the purposes of reliable replicated long-term storage. The service is offered to academic institutions and public sector including DCH institutions. The project is coordinated by PSNC.

The detailed objectives of this experiment are to:

1. Verify usability of EUDAT B2SHARE service for DCH communities in the terms of the following requirements:
 - simple data upload and access
 - easy and effective data sharing
 - assuring data refferability of the data for long term
2. examine the usefulness of European-wide and national solutions for long-term data and meta-data preservation. Usefulness is considered in following aspects:
 - a. reliability of the long-term storage process
 - b. transparency of the data protection mechanisms from the point of view of the service directly interfaced by end-users (such as e.g. data publication and sharing service)

4.2 SETUP & DESCRIPTION OF TOOLS

For the purposes of evaluation following tools and services setup is prepared. It consists of two layers (see **Errore. L'origine riferimento non è stata trovata.**) relevant for aforementioned aspects of the experiment.

First, the upper layer provides easy interface for storing the data with simple metadata and data sharing, based on EUDAT B2SHARE service instance. Technically B2SHARE is a customised version of Invenio⁹ designed to offer a simple mechanism for uploading and sharing scientific data with associated metadata.

This layer is directly interfaced by the users and exposes to them several functions including data upload, meta-data editing, repository lookup, browsing, digital assets preview, meta-data presentation, search etc. This layer implements the clue of the functionality desired by the users in the considered use case.

The lower layer, which ideally should be transparent for the end users, provides an additional assurance for data sustainability as well as long-term data availability and safety.

It might be implemented using EUDAT's B2SAFE service or PSNC-coordinated Archival Services of the PLATON project. In our experiments we decided to use the Polish national service for implementing reliable long term storage and preservation. In this way we demonstrate and evaluate national facilities that complement the Europe-wide services such as EUDAT. Such approach also creates opportunity to verify the modularity and openness of the EUDAT's solutions by trying to orchestrate them with other services transparently to end-users.

PLATON's Archival Service is a data backup/archival service with geographical replication. The service ensures long-term data availability and safety thanks to automatic data and meta-data replication as well as additional techniques such as integrity control of incoming data and

⁸ <http://www.platon.pionier.net.pl/online/archiwizacja.php?lang=en>

⁹ <http://invenio-software.org/>

periodic data scrubbing including local and remote replica integrity checks. Importantly, these mechanisms are implemented transparently to end-users. PLATON's Archival Service can be interfaced by SFTP, WebDAV and GridFTP protocols.

If enriched with NDS2-project¹⁰ provided tools the service can be accessed through convenient virtual file system interfaces from Windows and Linux clients and using portable file browser-like GUI.

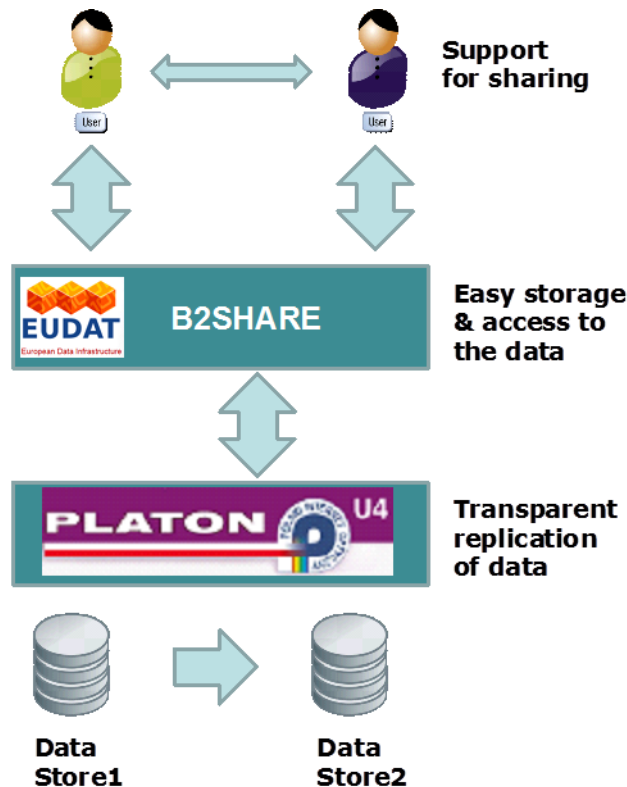


Figure 3: Setup of the EUDAT service experiment

¹⁰ <http://nds.psnc.pl>

4.3 EXPERIMENTATION

Table below summarizes actions that accomplish our experiment and provides overview of their status. Following sections discussed details of our activities and future plans.

	Action	Result / Status	Responsible
1.	Evaluations of existing B2SHARE instances	Tests performed by Swedish and Polish partners. TBD by Belgian partners. Short test report to be prepared.	Riksarkivet, PSNC, Belspo
2.	Setup dedicated B2SHARE service.	Running service.	PSNC
3.	Setup backup B2SHARE to PLATON Archival Services	In progress. Data exchange among services/layers TBD (automated backups and archival copies TBD)	PSNC
4.	Evaluation of the orchestrated services.	Tests TBD. Short report to be prepared.	Riksarkivet, PSNC, Belspo

Until now the EUDAT tests have not yet been realised by KIK-IRPA.

4.3.1 Work done so far

As far our work focused on the evaluation of the EUDAT B2SHARE service using existing publicly available production and demonstration instances of the service. Several cultural institutions from Sweden and Poland were involved in these evaluations.

In addition two-level services setup is being built by PSNC involving a dedicated B2SHARE service instance integrated with PLATON's Archival Services as described in the previous section. This configuration is going to be used in the following stages of our experiment.

4.4 RESULTS & NEXT STEPS

4.4.1 Preliminary results

Preliminary results of the evaluation show that B2SHARE service provides data sharing and publication solution suitable for the needs of small cultural institutions and "citizen" "publishers" or "curators".

However mass scale uploads and sharing may require more domain-optimised and specialised approach. Ensuring long-term data availability is not part of B2SHARE's service scope and intent; therefore B2SHARE should be orchestrated with additional layers such as EUDAT B2SAFE and PLATON's Archival Services.

4.4.2 Next steps

Activities planned for the following period include evaluation of the EUDAT's service for data sharing and publication orchestrated with long-term storage and preservations solution. At this stage transparency and reliability of the data preservation and safety mechanisms will be evaluated. B2share will be tested during July by KIK-IRPA.

5 EXPERIMENT 4: RE-EVALUATE THE ECSG AND REMOTE GRID/ CLOUD STORAGE SERVICES FROM POC1

5.1 RATIONALE

One of the limitations of the e-Culture Science Gateway (eCSG) identified in the first round of PoCs was that “[...] usability is limited to manually copy files to an external storage (grid, cloud, ...) and to fill out the metadata manually” (from D5.3, page 14). That limitation was indeed unavoidable because a general uploader, as the one that was developed in the first year of the project, cannot be used to seamlessly cope with a large variety of specific metadata formats and schemas.

We therefore decided in this experiment to demonstrate that specific uploaders could have made the use of the eCSG simpler and easier as well as the procedure of automatic upload of data into Grid/Cloud storage and insertion of metadata in the gLibrary-enabled repository build “underneath” or “behind” the eCSG.

Also, backed by DCH-RP’s promoting the importance of Identity Federations and Identity Providers (IdPs), it was decided to create an IdP at ICCU for this experiment in order to allow curators working at ICCU to upload digital assets (i.e., data and metadata) through the eCSG using their institutional credentials. Authentication and authorization are then decoupled: the former is done by the user’s organisation (ICCU in this specific case), while the latter is done by the Service Provider (the eCSG in this specific case).

5.2 SETUP & DESCRIPTION OF TOOLS

The tools which have been used are the eCSG, which has been already described in several other deliverables of DCH-RP, and the “IdP in the cloud” (<http://goo.gl/A3BNx6>) service of GARR that is a sub-contractor of ICCU.

For the second round of PoCs two sets of digital assets have been chosen:

- 1) A set of WARC archives of websites belonging to the .it domain and provided by the National Library of Florence;
- 2) A set of multi-format data belonging to SITAR (<http://sitar.archeoroma.beniculturali.it/>): the archaeological information system of the city of Rome provided by the Special Superintendence of Rome (SSBAR), also involved in the ARIADNE project.

5.3 EXPERIMENTATION

During the experiment, the following activities have been carried out as follows:

Using “IdP in the cloud”, an Identity Provider has been created for ICCU and populated with their staff. The ICCU Identity has also been registered in the Italian Identity Federation IDEM (www.idem.garr.it), managed and operated by GARR. Through IDEM this ICCU IdP has also been registered in the eduGAIN (www.edugain.org) inter-identity provider federation.

Two uploader portlets specific for the WARC and the SITAR archives have been developed and integrated in the eCSG.

5.4 RESULTS & NEXT STEPS

Figures 4 and 5 show, respectively, the selection of the ICCU IdP in the Where Are You From (WAYF) service of the IDEM federation and its login page users are re-directed to when the sign in on the eCSG.

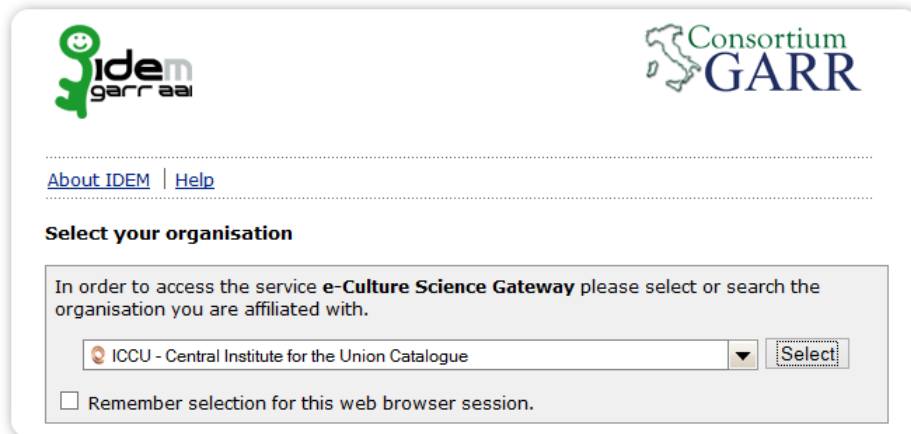


Figure 4: The ICCU IdP in the WAYF service of the IDEM Federation

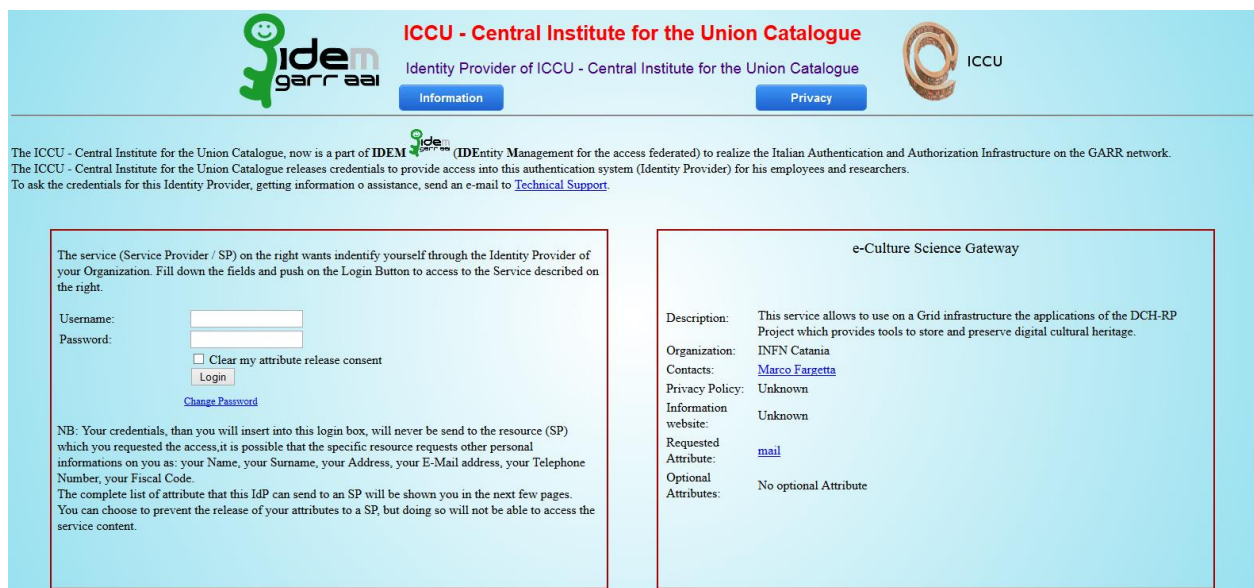


Figure 5: Login page of the ICCU IdP re-directed from the eCSG

Figure 6 shows instead the uploader portlet that has been developed for the WARC repository.

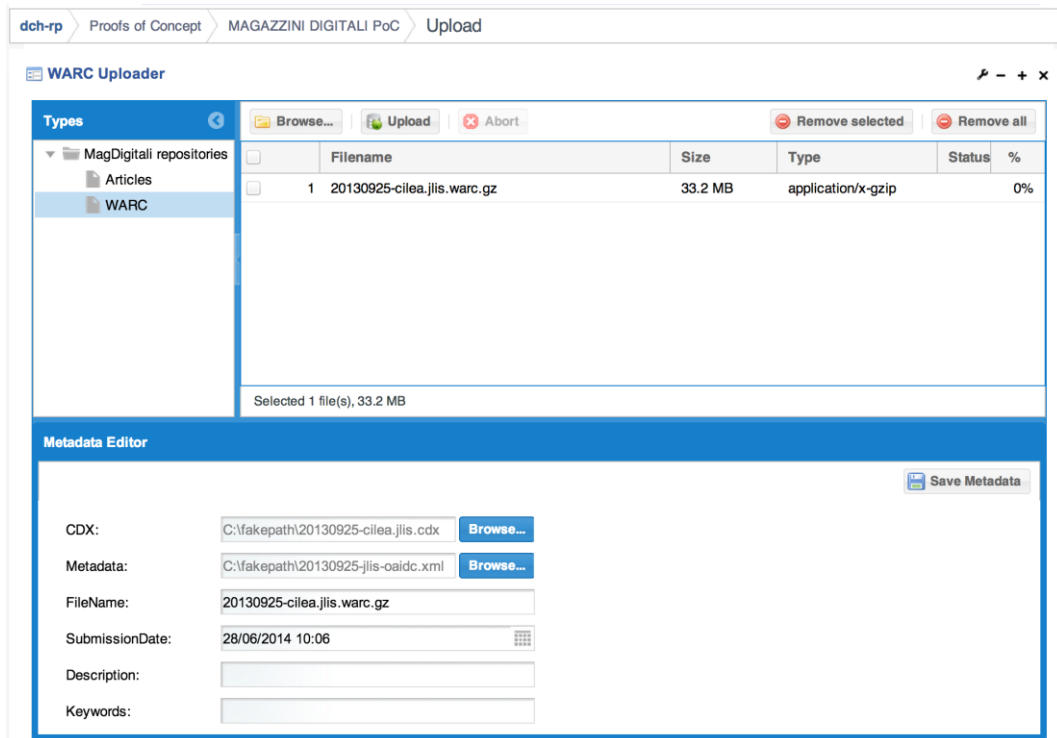


Figure 6: Uploader portlet for the WARC repository

Unlike the generic uploader portlet, with the customised uploader metadata is loaded automatically from a description file that is provided by the data owners. To demonstrate fine-grained authorisation special and separate “repository uploader” roles were defined for the WARC and the SITAR archives and have been enforced in the portal. This forbids uploaders of one repository to upload contents on the others and viceversa. The browser portlet of the WARC repository is shown in Figure 7.

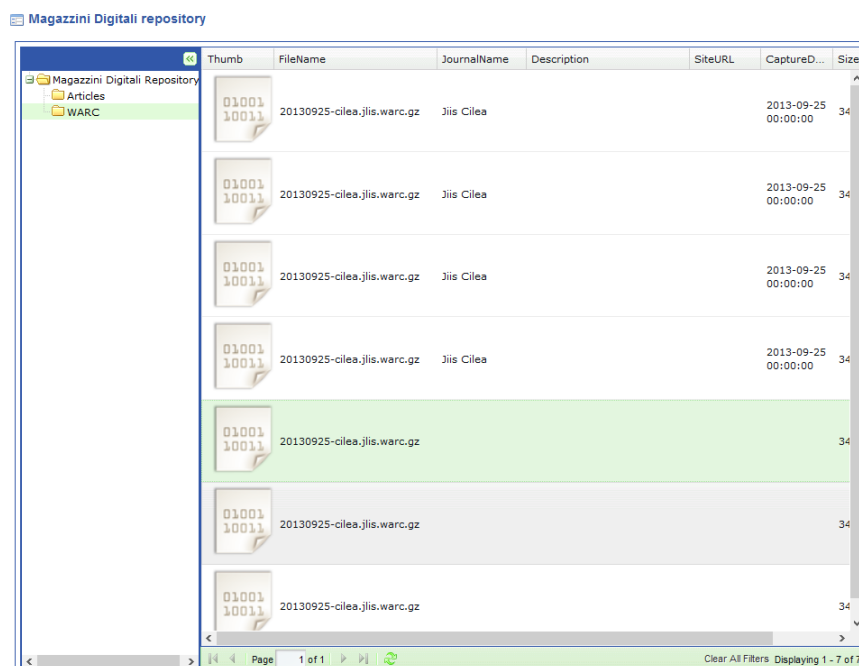


Figure 7: Browser portlet of the WARC repository

The uploader and browser portlet of the SITAR repository are shown in the Figures 8 and 9 below.

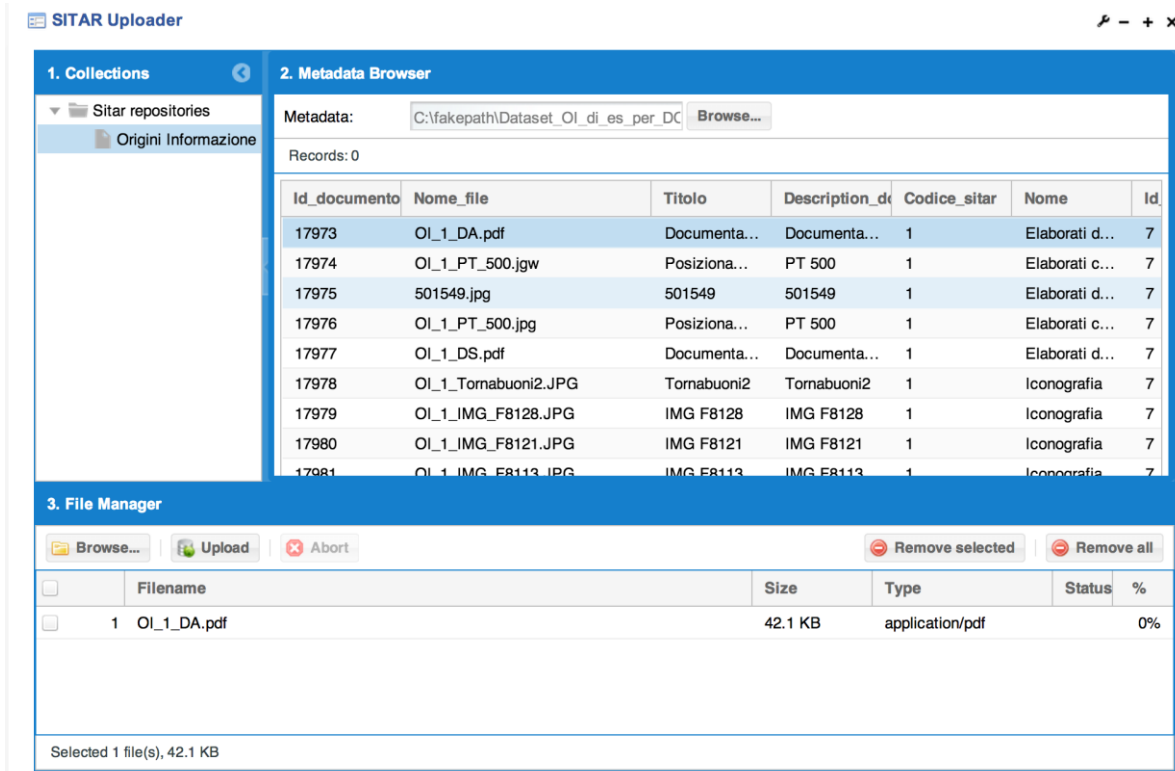


Figure 8: Uploader portlet for the SITAR repository

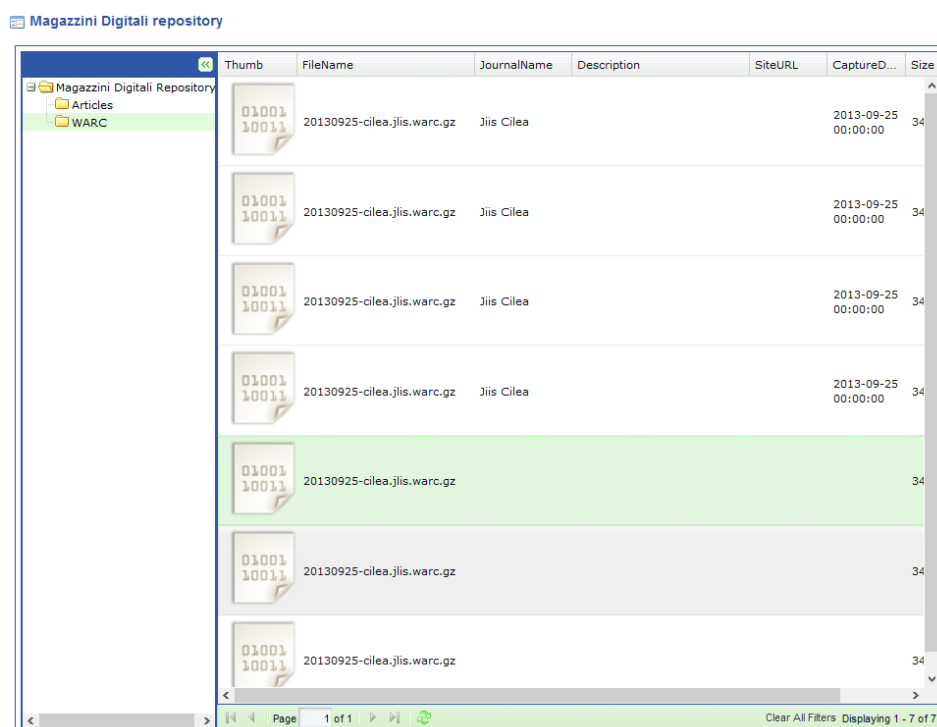


Figure 9: Browser portlet of the SITAR repository

As it emerges from the figures reported above, this second PoC has successfully demonstrated that customised uploaders can allow DCH institutions to make use of eCSG for the storing of their digital assets in automatic way. Moreover, ICCU has now a concrete example of the benefits of using federated credentials to access Service Providers belonging to the IDEM federation.

While doing this, at INFN Catania we have learned how to build an uploader portlet that can be customised in an easy and quick way for different metadata schemas and formats and this will allow further adaptations to other kind of repositories straightforward.

6 EXPERIMENT 5: A LONG-TERM DATA PRESERVATION PLATFORM

6.1 RATIONALE

The experiment on long-term data preservation explores the capabilities and service levels of data preservation with respect to standards, services and methods in the Cloud. It aims to deploy a platform that is able to create and access community specific and OAIS compliant Information Packages (IPs).

We want to gather provided data collections through the OAI-PMH protocol and build community specific IPs for further preservation.

The deployed system is based on open technologies, standards and recommendations like Tomcat, ODE, SOLR and RDF (OAI-ORE). The process of packaging is configurable through a BPEL orchestration of services. The setup and description follows in the next chapter.

For the first use case we collaborate with data from OpenAire.eu, which provides us via OAI-PMH with metadata and PDF documents to setup a running and testable system. We first focus on the metadata package generation, including community metadata.

6.2 SETUP & DESCRIPTION OF TOOLS

Our preservation system consists of a multi-layered architecture, depicted as follows:

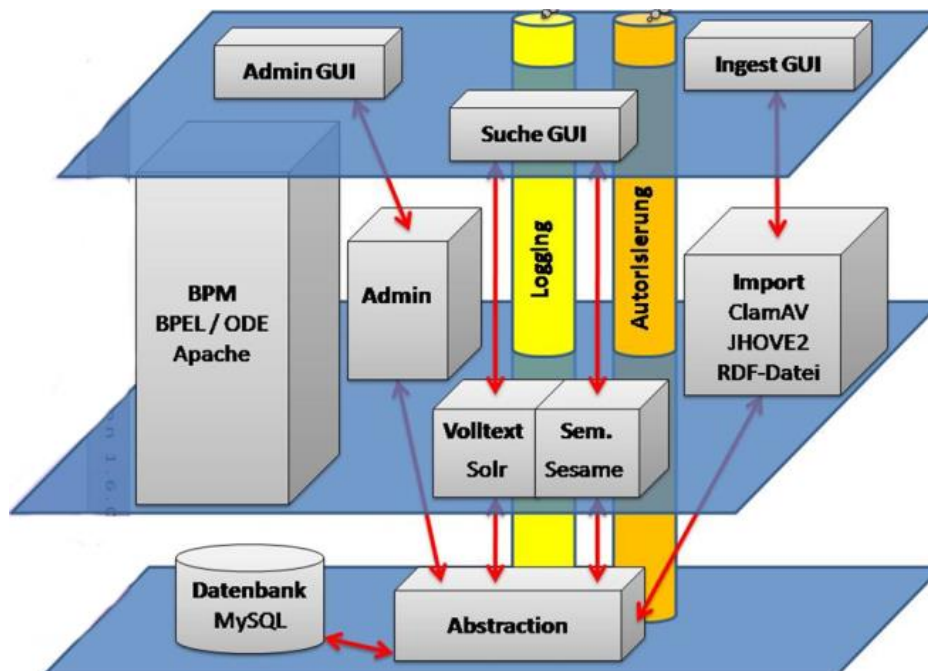


Figure 10: Architecture of the preservation system

It runs within a standard Linux operating system like Ubuntu or Debian and uses Apache as webserver and Tomcat as application server. Further main components are Apache ODE providing the BPEL-Engine to configure and run various services. MySQL is used as database system, to provide created IPs for access (see Figure 10). SOLR run the full-text search to find the packaged information. The platform also includes an authorization mechanism and logging of provenance information.

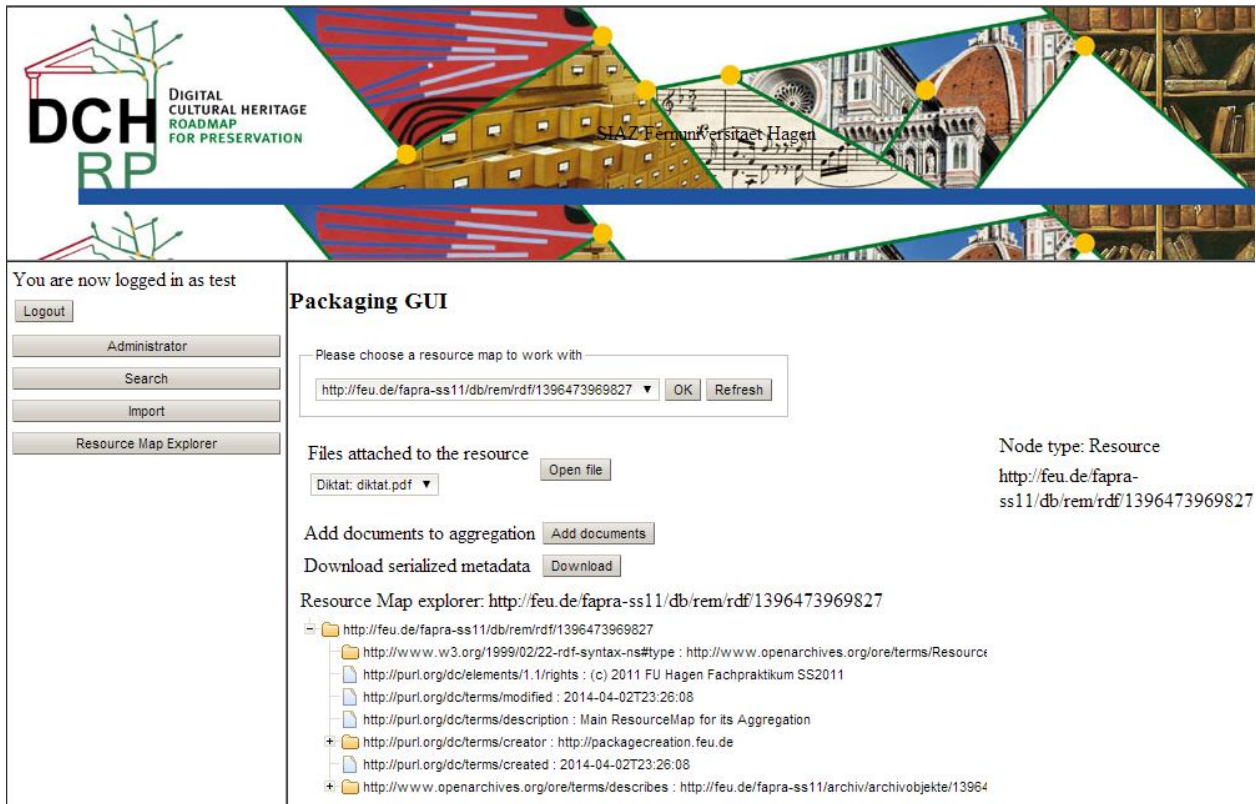


Figure 11: The envisaged GUI for packaging datasets for archival

6.2.1 Prerequisites

To install the current system you need to run a virtual machine based on Linux, preferably Ubuntu LTS.

The following software is required to build and run the services:

Software	Description	Installation
Subversion	Check out source code	apt-get install subversion
Maven	Build binaries	apt-get install maven
Zip	Create zip files (BPEL)	apt-get install zip
Tomcat	Application server that run the service	apt-get install tomcat6
MySQL	Database that holds AIPs	apt-get install mysql
Perl	Execute installation scripts	apt-get install perl

6.2.2 Installation

The following installation steps are necessary:

- Checkout the code:

- svn checkout `svn://svn.lgmmia.fernuni-hagen.de/fapra/siaz-ss2012/Install/trunk/Install`
- Change to working directory:
 - `cd Install;`
- Run the command, which uses maven to deploy the system:
 - `./fapra preinstall build install restart all`
- Further adaptations and configurations to host name, URLs etc. are currently necessary.

6.2.3 Modules

Further modules need to be installed to provide certain functionalities:

- Virus checking: ClamAV (ClamAV.net)
- File & Format checking: JHove (jhove.sourceforge.net/)

6.3 EXPERIMENTATION

As first step we investigated and installed our packaging and access system, as it needed to be adapted due to software and hardware updates. We documented the installation process and conducted a functional testing in a virtual machine. The system is up and running at: <http://kokum.fernuni-hagen.de:8080/JSFGui/start.jsf>

6.4 RESULTS & NEXT STEPS

The next step will be to run the first use case with our use case partner OpenAire. OpenAire is a metadata repository service and provides search and access to a variety of resources. The following steps will be investigated:

- Harvest a collection of data objects including meta-data and supplementary data consisting of PDF documents via OAI-PMH.
- Focus on OAIS compliant metadata packaging.

Following this experiment we will investigate with other use case partners more complex supplementary data objects like 3D visualisations, which need a different ingest processing as PDF.

7 CONCLUSION

Comparing the first Proofs of Concept phase with the second phase, which is still on-going, one cannot but identify significant progress in the DCH community on several levels:

Firstly, the first experiments in the project were very much focused around low-level tools that were already known *within* the DCH-RP consortium, mainly the memory institutes and partners in direct contact with these (e.g. ICCU, RA, BELSPO). This probably might be attributed to the project being in its early phase, and a roadmap to align the activities not being in place, though planned this way (c.f. the project's DoW). Yet, the experiments of the first Proofs of Concept phase yielded sufficient results to contribute to the subsequent intermediate Roadmap [R 2] evolving from the study provided early on in the project.

The second Proofs of Concept experiments in contrast focus more on usable solutions and services that have the potential of being integrated into existing solutions due to promising functionality – all the way towards experimenting with assembling a preservation platform fit for purpose. Although the experiments have not yet completed, the results that were obtained so far already indicate a much easier and valuable mapping of requirements and use cases described in the roadmap into functional and non-functional capabilities and potential solutions for them in a future version of the roadmap: The intermediate roadmap provided for the first time a specific list of functional capabilities that must be satisfied in a conceptual preservation platform, such as: OAIS compliance, automatic metadata capture and extraction, authenticity and integrity of data, distributed storage systems; the second Proofs of Concept phase responded with experimenting with specific solutions and services addressing some of these capabilities. In turn, the expected results allow the next iteration of the roadmap (to be published as D3.5 [R 3]) to further specify and concretise specific short-term and medium-term plans for the DCH community.

Second, the project partners significantly improved the yield of “funding-to-experimentation” ratio through developing and signing strategic MoUs with those projects that developed tools and services, and are the curators and stewards of data used or planned to be used in the second round of experiments. This is most visible in the experiments 1, 2, 4 and 5 where partners or data owned by these partners were involved through the projects SCAPE, SCIDIP-ES, ARIADNE and APARSEN. Compared with the mostly internally conducted experiments in the first PoC phase one cannot but note a significantly improved traction in the experiments themselves.

Third, it is well worth revisiting past experiments that failed when compared to the expected results, and improve on the tools and experimentation design through open and genuine analysis of the issues that were at hand – experiment 4 demonstrates this principle very well.

As to date not many experimental results are currently available since the experiments are allowed to run until September 2014. However, the following conclusions may already be drawn based on the progress made so far:

The “Matchbox” tool developed in the SCAPE project is clearly not designed for end users. However, preliminary results and existing integration with other platforms and services may well indicate that integrating a parallelised Matchbox tool into a higher-level preservation platform at reasonable cost has the potential of providing a scalable service for duplication filtering across large archives of digital data.

Experimenting with B2SHARE and eCSG both so far demonstrated that current implementations are only not entirely suitable for mass-upload of data into a preservation archive. This perhaps indicates that initial ingestion of data into an archive might best be designed as a separate activity in the overall preservation lifecycle even though specialised uploader portlets for the eCSG used in experiment 4 were key to the progress of that experiment.

This points to a more general issue concerning metadata. The current landscape of metadata standards does not support any indication of convergence towards a common standard across domains. Tool developers need to take this into account when designing software – unless they specifically target one market segment. This project’s experience with the eCSG points into that direction that pluggable metadata parsers and handlers seem the better way over general-purpose elements.

Even though the experiments have not concluded yet, the following recommendations for the DCH roadmap may be concluded as follows:

Recommendation 1: Tools designed for installation on end user IT equipment, and intended for installation by end users, should be *as easy as possible to install* – ideally by a single action. It should be as easy as copying a number of files into one directory, followed by double-clicking an icon. Exemplar applications are the Eclipse Foundation’s IDE “drops”, or Firefox releases that literally require little more than copying a number of files into a directory of choice, or on a platform level, the Mac OS X application installation process comprising of one simple dragging the application icon to drop it over the system’s Applications folder.

Recommendation 2: Tools integrating with typical Linux package management systems such as apt-get for Debian based Linux distributions or yum for Red-Hat based systems must provide an appropriate package for all supported hardware architectures (32bit and 64bit), including a well-defined and well-managed dependency manifest, so that, after downloading the package, a single command to install that package automatically installs any missing dependency without further unnecessary interaction.

Recommendation 3: Ideally, tools identified as suitable for inclusion in the DP roadmap should have active maintainers for the used/desired target platforms who ensure that recommendations 1 and 2 are adequately met, so that installing an application, tool, or service requires little more than issuing a command similar to “`sudo apt-get install scape-matchbox`”

Recommendation 4: If some software does not entirely match DCH requirements, investigate whether it has a modular design, preferably including well-documented extension interfaces (c.f. “plug-in” and “connector” design), for which DCH-specific extensions might be developed at greatly reduced cost. Aim to find partners and communities in the same market segment that might join in the maintenance effort for either the entire tool, or specific plugins.

Recommendation 5: Aim to avoid vendor lock-in by developing a service-oriented architecture for the DCH digital preservation landscape (or a desired “Preservation-as-a-Service” platform) including strategically placed and mandated publicly defined standards governing the interfaces between the services within the platform. Aim to avoid or reduce to an absolute minimum second-level dependencies such as one service directly depending on one or more specific instances of other services – operational maintenance and reliable rollout is next to impossible in an entangled network of dependencies. Ideally, an SOA with the right abstraction level and service scoping allows upgrading one service entirely independently from any other service.

Recommendation 6: Before defining the technical architecture of the preservation services and platforms, define and agree on the business process(es) you wish to implement in the technical architecture. Good business process modelling results not only in a business process architecture satisfying the requirements, but allows changes in its orchestration and sequences without redefining or altering the defined activities.

Recommendation 7: In the process of further developing the roadmap, describe each service that is required, and which capabilities it is expected to implement. For example, describing a storage service the roadmap might attach the following capabilities to it:

- Bit-level preservation of each digital object stored in and managed through this service;
- Data access and modification policies: Read-only, copy-on-write, transactional, or version-controlled;
- Self-service configuration of object replicas
- Self-service configuration of geospatial distribution of replicas
- Central or distributed data access points
- Transparent storage medium obsolescence management
- ...

These recommendations are arguably very technical in nature. However, partners expect these to change or expand once all experiments have finished later this year.

8 REFERENCES

R 1	DCH-RP D3.1 “Study on a Roadmap for preservation”, R. Ruusalepp (EVKM), M. Dobrev (EVKM), March 2013. http://www.dch-rp.eu/getFile.php?id=114
R 2	DCH-RP D3.4 “Intermediate version of the Roadmap”, B. Justrell (RA), L. Balint (NIIFI), E. Toller (RA), R. Ruusalepp (EVKM), January 2014. http://www.dch-rp.eu/getFile.php?id=221
R 3	DCH-RP D3.5 “Final version of the Roadmap”, to be published,
R 4	DCH-RP D4.1 “Trust building report”, R. Ruusalepp (EVKM), B. Justrell (RA), L. Florio (TERENA), April 2014. http://www.dch-rp.eu/getFile.php?id=274
R 5	DCH-RP D5.3 “Report on the first proof of concept”, M. Drescher (EGI.eu), E. Toller (RA), R. Vandenbroucke (BELSPO), September 2013, http://www.dch-rp.eu/getFile.php?id=198
R 6	Fourth DCH-RP plenary meeting, Catania, Italy, 20 – 21 January 2014; http://www.dch-rp.eu/getFile.php?id=340 (reserved space, may require login)
R 7	Fifth DCH-RP plenary meeting, Tallinn, Estonia, 24 – 25 April 2014; http://www.dch-rp.eu/getFile.php?id=339 (reserved space, may require login)

ANNEX 1: INSTALLING AND TESTING MATCHBOX

During the PoC2 test was target to install and test some tools created during the Scalable Preservation Environments project - SCAPE project (<http://www.scape-project.eu/>). Although there are available amd64 compiled package it is not possible to change the bit signature of a binary. It has to be compiled for a certain architecture where the system will be in use. Because of the technical issues the only tested tool was Matchbox - Duplicate image detection tool.

There are several important tools/modules to install:

1. Install GCC, the GNU Compiler Collection

The GNU Compiler Collection includes front ends for C, C++, Objective-C, Fortran, Java, Ada, and Go, as well as libraries for these languages (libstdc++, libgccj,...). GCC was originally written as the compiler for the GNU operating system. The GNU system was developed to be 100% free software, free in the sense that it respects the user's freedom.

2. Install “build-essentials”

This package contains an informational list of packages, which are considered essential for building Debian packages. This package also depends on the packages on that list, to make it easy to have the build-essential packages installed.

3. Install g++

Released by the Free Software Foundation, g++ is a *nix-based C++ compiler usually operated via the command line. It often comes distributed with a *nix installation, so if you are running Unix or a Linux variant you likely have it on your system.

4. Install CMake

CMake is the cross-platform, open-source build system. CMake is a family of tools designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files. CMake generates native makefiles and workspaces that can be used in the compiler environment of your choice.

CMake has the following dependencies that need to be satisfied:

- libarchive-3.1.2 (<http://www.linuxfromscratch.org/blfs/view/svn/general/libarchive.html>)
- curl-7.36.0 (<http://www.linuxfromscratch.org/blfs/view/svn/basicnet/curl.html>)
- Lib Boost (<http://ubuntuforums.org/showthread.php?t=1725216>)

and optionally

- Cmake GUI (<http://www.linuxfromscratch.org/blfs/view/svn/x/qt4.html>)
[and https://secure.mash-project.eu/wiki/index.php/CMake:Quick_Start_Guide](https://secure.mash-project.eu/wiki/index.php/CMake:Quick_Start_Guide)

More information can be found at:

- <http://www.cmake.org/cmake/resources/software.html>
- <http://www.linuxfromscratch.org/blfs/view/svn/general/cmake.html>

5. Install Python

Python is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C. Ubuntu 14.04 LTS has available Python 3.4.0, but for the Matchtool is necessary Python 2.7.

Python's important dependencies are: `libsqlite3-dev`, `sqlite3`, `bzip2` and `libbz2-dev`.

More information is available at <https://www.python.org/>

6. Install OpenCV

OpenCV is the most popular and advanced code library for Computer Vision related applications today, spanning from many very basic tasks (capture and pre-processing of image data) to high-level algorithms (feature extraction, motion tracking, machine learning). It is free software and provides a rich API in C, C++, Java and Python. Other wrappers are available. The library itself is platform-independent and often used for real-time image processing and computer vision. OpenCV has already lot of interesting developments like face detection, similar object finder and etc. , see also screenshots below.

Creating and compiling the OpenCV is one of the most important step.

OpenCV has quite a long list of dependencies, mostly supporting the various image filtering and detection algorithms: *build-essential*, *libgtk2.0-dev*, *libjpeg-dev*, *libtiff4-dev*, *libjasper-dev*, *libopenexr-dev*, *cmake*, *python-dev*, *python-numpy*, *python-tk*, *libtbb-dev*, *libeigen2-dev*, *yasm*, *libfaac-dev*, *libopencore-amrnb-dev*, *libopencore-amrwb-dev*, *libtheora-dev*, *libvorbis-dev*, *libxvidcore-dev*, *libx264-dev*, *libqt4-dev*, *libqt4-opengl-dev*, *sphinx-common*, *texlive-latex-extra*, *libv4l-dev*, *libdc1394-22-dev*, *libavcodec-dev*, *libavformat-dev*, *libswscale-dev*.

Final configuration of the OpenCV needs the following settings:

```
-- Video I/O
--   DC1394 1.x:                NO
--   DC1394 2.x:                YES (ver 2.2.1)
--   FFMPEG:                    YES
--   codec:                     YES (ver 54.35.0)
--   format:                    YES (ver 54.20.4)
--   util:                      YES (ver 52.3.0)
--   swscale:                   YES (ver 2.1.1)
--   gentoo-style:              YES
--   GStreamer:                 NO
--   OpenNI:                    NO
--   OpenNI PrimeSensor Modules: NO
--   PvAPI:                     NO
--   GigEvisionSDK:            NO
```

```

--      UniCap:                               NO
--      UniCap ucil:                           NO
--      V4L/V4L2:                               Using libv4l (ver 1.0.1)
--      XIMEA:                                  NO
--      Xine:                                   NO
--
--      Other third-party libraries:
--      Use IPP:                                NO
--      Use Eigen:                              YES (ver 2.0.17)
--      Use TBB:                                YES (ver 4.2 interface 7000)
--      Use OpenMP:                             NO
--      Use GCD                                 NO
--      Use Concurrency                          NO
--      Use C=:                                  NO
--      Use Cuda:                                NO
--      Use OpenCL:                             YES
--
--      OpenCL:
--      Version:                                dynamic
--      Include path:                           /home/OpenCV/opencv-2.4.9/3rdparty/include/opencvcl/1.2
--      Use AMD FFT:                             NO
--      Use AMD BLAS:                            NO
--
--      Python:
--      Interpreter:                             /usr/bin/python2 (ver 2.7.6)
--      Libraries:                               /usr/lib/i386-linux-gnu/libpython2.7.so (ver 2.7.6)
--      numpy:                                   /usr/lib/python2.7/dist-packages/numpy/core/include (ver
1.8.1)
--      packages path:                           lib/python2.7/dist-packages
--
--      Java:
--      ant:                                     NO
--      JNI:                                     NO
--      Java tests:                             NO
--
--      Documentation:
--      Build Documentation:                     YES
--      Sphinx:                                 /usr/bin/sphinx-build (ver 1.2.2)
--      PdfLaTeX compiler:                       /usr/bin/pdflatex
--
--      Tests and samples:
--      Tests:                                   YES
--      Performance tests:                       YES
--      C/C++ Examples:                          YES
--
--      Install path:                            /usr/local
--
--      cvconfig.h is in:                       /home/OpenCV/opencv-2.4.9/build

```

```
-- -----  
--  
-- Configuring done  
-- Generating done  
-- Build files have been written to: /home/anz/Downloads/OpenCV/opencv-2.4.9/build
```

More information:

<https://help.ubuntu.com/community/OpenCV>

<https://github.com/jayrambhia/Install-OpenCV/tree/master/Ubuntu>

<http://miloq.blogspot.com/2012/12/install-opencv-ubuntu-linux.html>

7. Install Matchbox

The idea of the tool is that there are numerous situations in which you may need to identify duplicate images in collections, for example:

- to ensure that a page or book has not been digitised twice
- to discover whether a master and service set of digitised images represent the same set of originals
- to confirm that all scans have gone through post-scan image processing.

Checking to identify duplicates manually is a very time-consuming and error-prone process.

In the Readme file provided on the Github repository there are information about how to compile and set up the Matchbox toolset. There is an image of a virtual machine, that we use for training within the SCAPE project but during the test there were no necessary tools to use it.

However the preparation and starting the application from scratch is requiring several ICT skills and development knowledge, especially when there is a need to use any Linux desktop version without any development tools and specific libraries, modules and programs.

Below are written down necessary steps including information about to make Matchbox to work.

Building CMAKE:

```
$ ./configure  
The C compiler identification is GNU 4.9.0  
The CXX compiler identification is GNU 4.9.0  
Check for working C compiler: /usr/bin/cc  
Check for working C compiler: /usr/bin/cc -- works  
Detecting C compiler ABI info  
Detecting C compiler ABI info - done  
Check for working CXX compiler: /usr/bin/c++  
Check for working CXX compiler: /usr/bin/c++ -- works  
Detecting CXX compiler ABI info  
Detecting CXX compiler ABI info - done  
COMPARE: Opencv found.
```

```
Boost version: 1.54.0
Found the following Boost libraries:
serialization
Configuring done
$
```

Compiling Matchbox:

```
$ make
...
[ 95%] Building CXX object DPQA_Compare/CMakeFiles/mb_compare.dir/DPQA_Compare.cpp.o
Linking CXX executable mb_compare
[ 95%] Built target mb_compare
Scanning dependencies of target mb_extractfeatures
[ 97%] Building CXX object
DPQA_ExtractFeatures/CMakeFiles/mb_extractfeatures.dir/DPQA_ExtractFeatures.cpp.o
Linking CXX executable mb_extractfeatures
[ 97%] Built target mb_extractfeatures
Scanning dependencies of target mb_train
[100%] Building CXX object DPQA_Train/CMakeFiles/mb_train.dir/DPQA_Train.cpp.o
Linking CXX executable mb_train
[100%] Built target mb_train

$ ls
CMakeCache.txt  cmake_install.cmake  CPackSourceConfig.cmake  DPQA_ExtractFeatures  DPQA_Train
CMakeFiles      CPackConfig.cmake    DPQA_Compare  DPQALib  Makefile

$ sudo make install
[ 93%] Built target DPQALib
[ 95%] Built target mb_compare
[ 97%] Built target mb_extractfeatures
[100%] Built target mb_train
Install the project...
-- Install configuration: ""
-- Installing: /usr/bin/FindDuplicates.py
-- Installing: /usr/bin/MatchboxLib.py
-- Installing: /usr/lib/libDPQALib.so
-- Removed runtime path from "/usr/lib/libDPQALib.so"
-- Installing: /usr/bin/mb_compare
-- Removed runtime path from "/usr/bin/mb_compare"
-- Installing: /usr/bin/mb_extractfeatures
-- Removed runtime path from "/usr/bin/mb_extractfeatures"
-- Installing: /usr/bin/mb_train
-- Removed runtime path from "/usr/bin/mb_train"
```

More information is available at <https://github.com/openplanets/matchbox>.