Project Acronym: Presto4U
Grant Agreement no: 600845
Project Title: European Technology for Digital Audiovisual Media Preservation

D3.2: Research Outputs Assessments v1

Project funded by the European Community in the 7th Framework Programme

Table of contents

Scope

The long-term preservation of digital audio-visual media presents a range of complex technological, organisational, economic and rights-related issues, which have been the subject of intensive research over the past fifteen years at national, European and international levels. Although good solutions are emerging, and there is a large body of expertise at a few specialist centres, it is very difficult for the great majority of media owners to gain access to advanced audio-visual preservation technologies. This deliverable 'Research Outputs Assessment v1' will describe the research outputs identified in year 1 of Presto4U, and which have the potential to address CoP needs and requirements. This document will also describe in detail, results of the assessment exercise carried out on identified ROs. The methodology for assessing these tools has been established as part of WP3 task T3.1 'Research Outcomes Assessment Methodology' documented in deliverable D3.1 'Specification of Assessment Criteria, Metrics, Processes, Datasets and Facilities'.

The deliverable is a direct outcome of Task 3.2 'Preservation Research Technology and Assessment'. The purpose of which is to identify and assess research outputs to establish their readiness for take-up. Part of the task also involves creating an initial dataset for Presto4U which initially can be used by the ROs being assessed, and it is envisaged to be released as a public dataset at the end of the project in year 2. The details of the year 1 dataset are provided in this deliverable. Further, we report on technology tracking and mapping exercise conducted during year 1 and the tools which have been developed to enable this task. We would like to note that this kind of activity has not been performed before and for this reason why the deliverable is released in two iterations (year 1 and end of year 2). Any shortcomings of the work done in year 1 will be addressed in the next iteration. The assessment methodology will also need to be changed as more needs are gathered from the Communities of Practice during the course of year 2. This deliverable is a first attempt at formalising a methodology for assessment of research outputs in the AV preservation domains and map them to the needs from the CoPs.

Executive summary

The long-term preservation of AV media presents several challenges in terms of research and development required, rights related issues and the methodology required to assess the tools based on new and existing standards. The issue related to assessment of AV preservation tools is particularly challenging because within Presto4U the assessment methodology must also take into consideration the needs expressed by the CoPs. As the goal here (goal of the tools being produced by solution providers, software vendors) is for long-term preservation and given the fact that technology cycles are relatively short, there is a need for tracking and mapping of candidate solutions to keep up with on-going information technology developments. Further, mapping between CoP needs to candidate technology solutions or research outputs is in itself a challenging task. This is due to ambiguity in the way the needs are expressed in different CoPs, the effort involved in semantic acquisition of these needs, and the ability to map them to research output tool features in a semi-automated fashion.

Before CoP needs can be mapped to candidate research outputs, we first need to methodologically assess the candidate solutions in order to determine their readiness for take-up and their positioning within the AV preservation space to create opportunities for take-up. The results of such assessment will objectively determine if a given research output is able to suffice the needs for a particular preservation task as expressed by the CoPs. The work carried out in WP3 Task 3.1 ‘Research Outcomes Assessment Methodology’ as reported in deliverable D3.1 [1] described a methodology for the assessment of preservation tools for digital collections. It merged together the commonly adapted best practices in software product quality measurement with the preservation workflow of digital audio visual media. In this deliverable, we have implemented the methodology defined in D3.1. This is a first iteration of the deliverable, which reports on the identified research outputs in year 1, lists the criteria for the assessment of these ROs, and reports on the results of the assessment. As far as we know, there have been no other efforts in related work that have carried out such assessment. Any shortcomings and updates (including the assessment of more ROs) will be reported further in the year 2 iteration of this deliverable. The rest of the document is organised as follows:

Chapter 1 reports on the identified research outputs for year 1. These ROs were chosen based on the needs expressed during various CoP interactions during the course of the project, the commercial tech watch carried out as part of Task 3.2, and preservation needs expressed by project partners. We firstly describe a tool which has been developed within the scope of Presto4U and allows semantically encoded knowledge acquisition of tool features and also allows users to give an initial score based on a TRL (Technology Readiness Level) scale. Next, we go on to describe the tools themselves. These tools have been classified into 4 broad categories: metadata mapping and validation, storage, quality assessment, preservation platforms and systems. There were a total of 7 categories to AV preservation tools mentioned in D3.1. We aim to assess tools from each category by year 2. We have also carried out a commercial research tracking and mapping exercise. Some of the results of this will be reported separately as Tech Watch reports and made publicly available as mentioned in the project DoW [2].

Chapter 2 describes the assessment criteria for each broad category of tools based on ISO/IEC 25023 standards for assessing software quality of tools which has been improved upon in D3.1. Assessment of each tool is carried out by defining functional tests to determine if the sub-characteristics mentioned against each tool are being satisfied. Here we also provide measurement functions for each sub-characteristic identified. This allows each tool assessed to have a numerical score at the end of the evaluation. Finally, a list of functional tests for each category of tool is presented along with their importance (mandatory, required, optional etc).

Chapter 3 presents the results of assessment each of the candidate ROs. We have assessed a total of seven tools in year 1. The evaluation was carried by Presto4U partners (JRS, IT Innovation, EURIX, RAI) using their own hardware installations. For each tool, we present the results of the functional tests defined for that category. For some functional tests it is not possible to derive a numerical score (e.g. evaluating user interface aspects).

Chapter 4 describes the Presto4U dataset. The tools assessed in year 1 have made use of the dataset to carry out tests. The goal here is to create a publicly available representative dataset for the assessment of AV preservation tools. The dataset sourced in year 1 consists of contributions from the communities of practice members. This dataset is available for internal tool testing purposes within Presto4U. We are currently in the process of creating a common licence for this dataset such that it is available for public consumption at the end of year 2.

In Chapter 5, we conclude the deliverable highlighting the main observations from the assessment exercise in year 1 and provide an insight into how the task will carry on in year 2 based on the lessons learnt so far.

1 Research Outputs Identification

In the context of Presto4U a **Research Output** (RO) is a software/hardware/methodology which is a direct result of research in AV and other digital preservation projects and which has the potential for commercial up-take in the future. We are specifically looking at EC FP6 and FP7 projects (results of PRESTOPRIME and previous Presto family projects are being monitored). As part of the technology and market watch sub task of WP3 task T3.2, we will also look at commercial ROs during the course of the project.

As part of task 3.2, the first stage is in identifying the ROs which can potentially address the CoP needs and present an opportunity for take-up. For the second stage, in order to objectively quantify the suitability of an RO, we need to assess the tool using a formally defined methodology and a measurement method. A measurement method is a logical sequence of operations used to quantify properties with respect to a specified scale. The result is a quality measure element. Therefore, in order to measure software quality we need for each specified characteristic to define:

- measure elements, e.g. identify which set of system properties cover a quality characteristic
- measurement method or test which measures each system property. The combination of those measures will derive the quality measure of that characteristic.

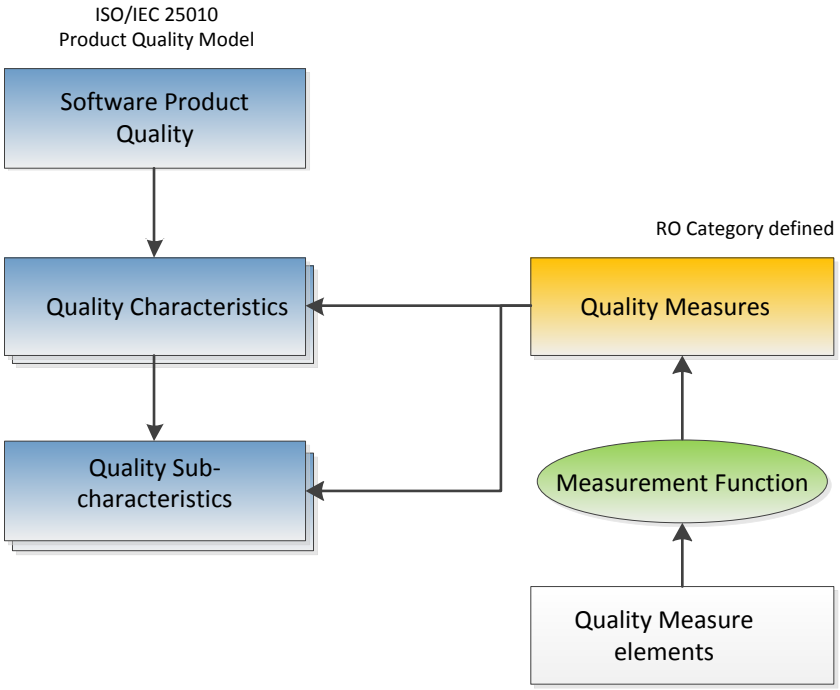

Figure 1 Relations between quality model and measures

Ideally, there should be a clear mapping between user requirements regarding the assessment and the quality characteristics provided by the standard. We call these quality requirements. Such quality requirements can be used to define measure elements as well as the measurement methods. The measurement methods will be applied on the RO during assessment. In order to carry out the assessment exercise, we have firstly specified a general assessment methodology and further specialised the quality measures for each

category of tool which is being evaluated. This allows us to define generic quality measure based on tool categories and also enables the comparison of tools within the same category.

This deliverable will present the detailed results of the assessment carried out in year 1 of the project using the methodology defined in D3.1 [1]. In the next sub-section, we will introduce the knowledge acquisition approach used for gathering information about ROs in year 1.

1.1 RO Knowledge Acquisition Approach

1.1.1 Sources of Information

During the first phase of Research Outputs Analysis we have primarily focussed upon research outcomes that have matured within recent FP7 projects. The process of identifying and cataloguing research outcomes began in early 2013 and will continue throughout the PRESTO4U project, the information gathered during this time will be captured in a tool as part of the project (Research Outputs Database). The particular research outcomes that are assessed below have been selected and prioritised based upon direct feedback on needs from the Communities of Practise formed by the project.

During the course of year 1 we had 3 main sources for RO acquisition:

1. PrestoCentre[1] contains a list of existing tools developed for digital preservation organised according as per a loosely defined taxonomy. This was the starting point for our knowledge acquisition exercise. The schema we developed (see Section 1.1.2) for semantically storing RO tool features borrows many of the classification terms expressed in the PrestoCentre website. Most of the tools collected from PrestoCentre belong to FP7 projects.
2. Input from project partners and CoPs: In D3.1 we listed 16 tools identified as candidate research technologies. Most of these tools are the result of recent research activities (some tools have been developed by Presto4U partners themselves), some are in prototype stage, while some have already reached sufficient maturity to be used in a production environment. Some tools e.g. Archivematica were chosen as candidates as a result of direct input from CoPs (TATE: Arts & museum objects, artists and their representatives) expressing the need for assessing the tool.
3. Commercial events: As part of the technology watch exercise we attended IBC 2013[2] and conducted a series of interviews with commercial vendors to gain a perspective on the current commercial and research trends in the digital preservation arena, barriers preventing adoption of new research outputs and opportunities for take-up. We also collected a list of early stage commercial research outputs as part of this exercise. The result of this technology watch will be published separately as publicly available tech watch reports. This deliverable (D3.2) will only focus on the specific tools chosen for assessment in year 1 and their assessment results.

[1] https://www.prestocentre.org/library/tools
[2] http://www.ibc.org/

In the next section, we will present a web application PrestoKAT, which allows users to easily input and semantically record RO features using intuitive, web form based interfaces. This RO knowledgebase will later aid in mapping the CoP requirements to tool features.

1.1.2 PrestoKAT Knowledge Acquisition Tool

The PrestoKAT web application was developed as part of Task 3.2 in order to enable tracking and mapping of research outputs in the AV preservation space. The tool uses easy to use and intuitive web based user interfaces at the front end. Different stakeholders (researcher, tool providers, end users etc) can register for an account with PrestoKAT and start adding information about their research outputs. All data entered is stored in a back end MySQL database. The schema for this database was developed after observing various tools in the AV preservation domain, existing schemas and standards (e.g. OAIS functional specification model [3]), and the knowledge schema developed as part of Presto4U [4]. The aim is to semantically record metadata about the features of research outputs. This will enable mapping between the requirements (captures according to the knowledge schema) and the research outputs that match the requirements in a semi-automated manner. Figure 2 shows the database schema for PrestoKAT. We capture aspects of the tools such as the functional features of the tools and additional non-functional details e.g. licence types. This evolving schema will later (in year 2) be mapped to the CoP knowledge schema in order to form brokerage mechanisms between CoP requirements and solution providers.

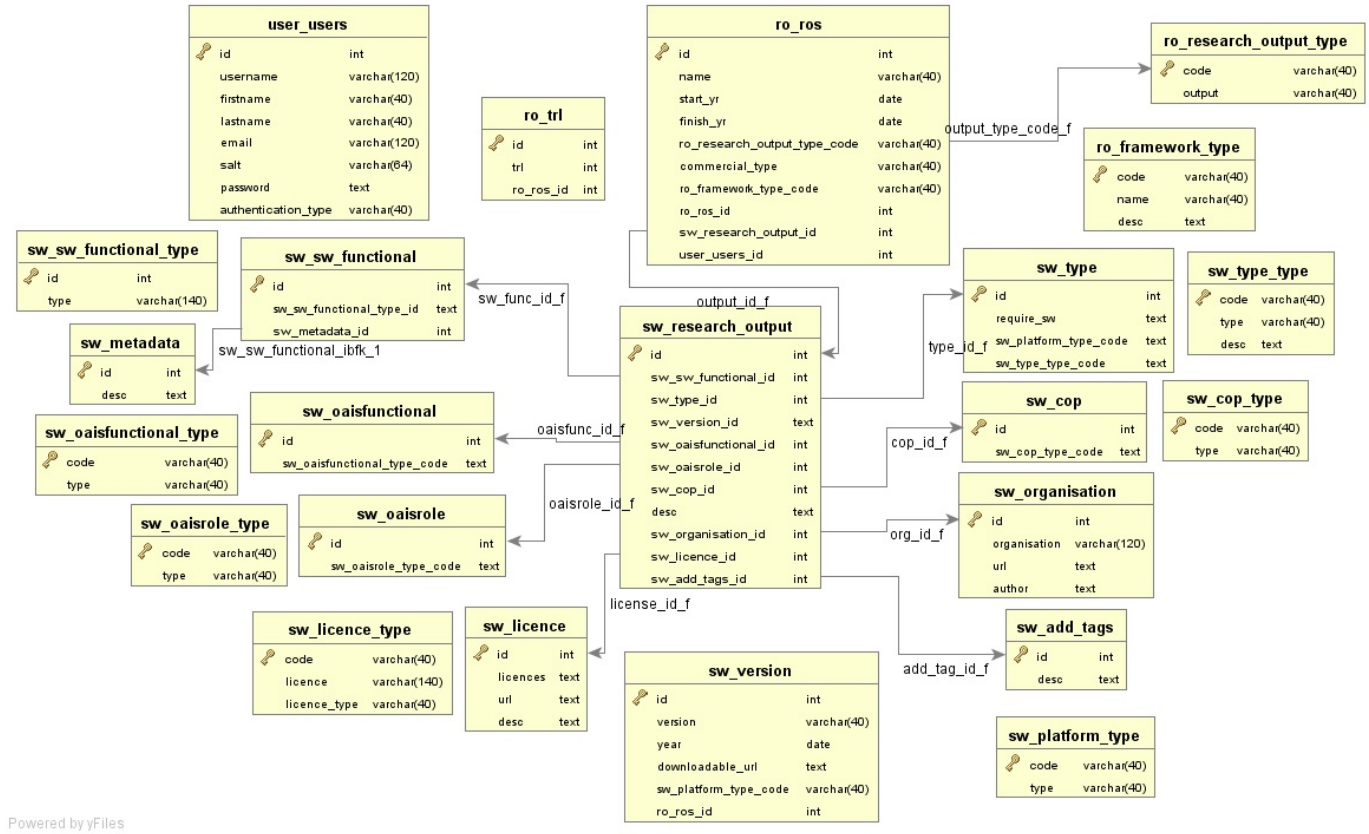

Figure 2 PrestoKAT database schema

A description of some of the main fields in the database is give in the table below:

Field	Description
user_users	Contains the information of the user. Security is maintained by applying hash encryption algorithms on the password. The tool can easily be extended to use open ID for login instead. It is currently local. Firstname, lastname, email address are optional as it can be empty when open ID is used as the login.
ro_trl	The user can provide can provide an initial estimate of the TRL level of the software. 1. Some proving has been performed suggesting that the tool is practical (algorithm or other analysis) but no implementation 2. Software can be demonstrated that shows feasibility of tool, but no optimisation or other practical considerations 3. Software has been created and run in a laboratory environment demonstrating the tool architecture, scalability, fitness for purpose, reliability of algorithms 4. Key components of the tool have been tested in the laboratory with real input data and the performance has been successfully demonstrated 5. All components run together in an integrated system in a laboratory environment using real test data and performance, including scalability and robustness has been successfully demonstrated 6. A prototype version of the system has been tested by a representative user in a production environment using real input data (alpha test) 7. Sales-ready system in the final form has been released for beta test to users for use in real production environments 8. Commercial system delivered and in use by users in their day to day operations.
ro_research_output_type	The type of research outputs. The options currently supported are:

	1. Hardware 2. Research Project 3. General Research 4. Software 5. Others
ro_framework_type	If the output type is a Research Project, the project can then be further associated to a framework program. FP7, TSB etc.
sw_research_output	The category under which the research output belongs: 1. Metadata mapping and validation 2. Storage technologies for AV master quality files 3. Automatic information extraction 4. Quality assessment technologies 5. Manual content annotation 6. Rights management technologies and formats 7. Preservation platforms/systems 8. Research technologies relevant characteristics
sw_oaisfunctional_type	List of OAIS functional types. The user can select more than one type. 1. Access 2. Administration 3. Archival Storage 4. Data Management 5. Ingest 6. Preservation Planning
sw_licence	Contains the licence details. If the licence is selected by the user, the user must provide the url for that licence.
sw_version	Current version of the software. It is also possible to add metadata about multiple versions of software.
sw_platform_type	Platforms on which the software or tool operates.
sw_organisation	The organisation that has created the tool.

Figure 3 below show the PrestoKAT user interface. It is possible to add additional metadata tags for a research output. The interface provides suggestions wherever possible using AJAX based drop-downs as the user starts typing the first few characters.

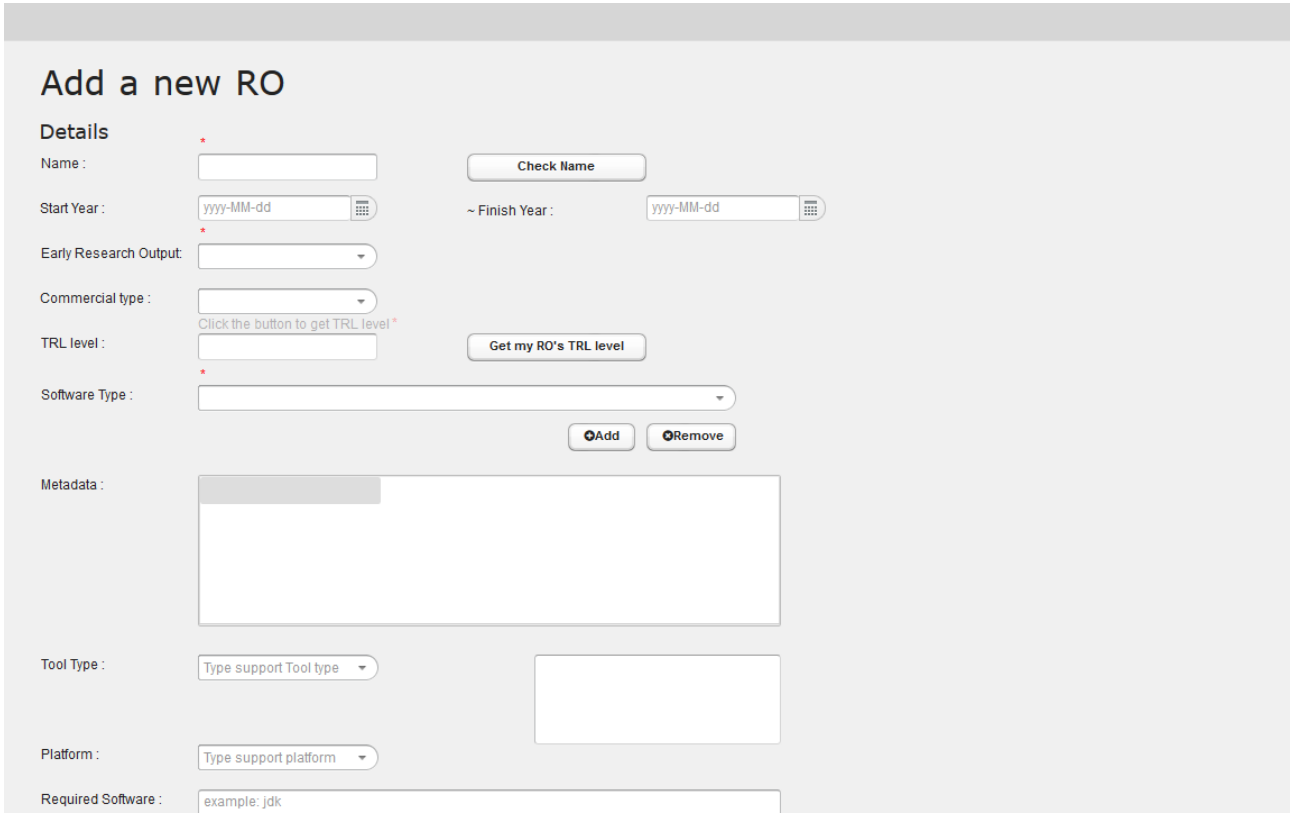

Figure 3 PrestoKAT user interface

Figure 4 below shows the TRL [5] level chooser that has been implemented in PrestoKAT. The idea here is that when a user encounters a new tool (e.g. whilst conducting interviews or browsing through stands at a AV preservation media event or seminar) they can immediately provide an initial estimate of the TRL level based on the data they have gathered during the event. This TRL level may later be adjusted based on the results of the detailed assessment exercise. It also gives users an idea of whether a particular RO is suitable for assessment straight away.

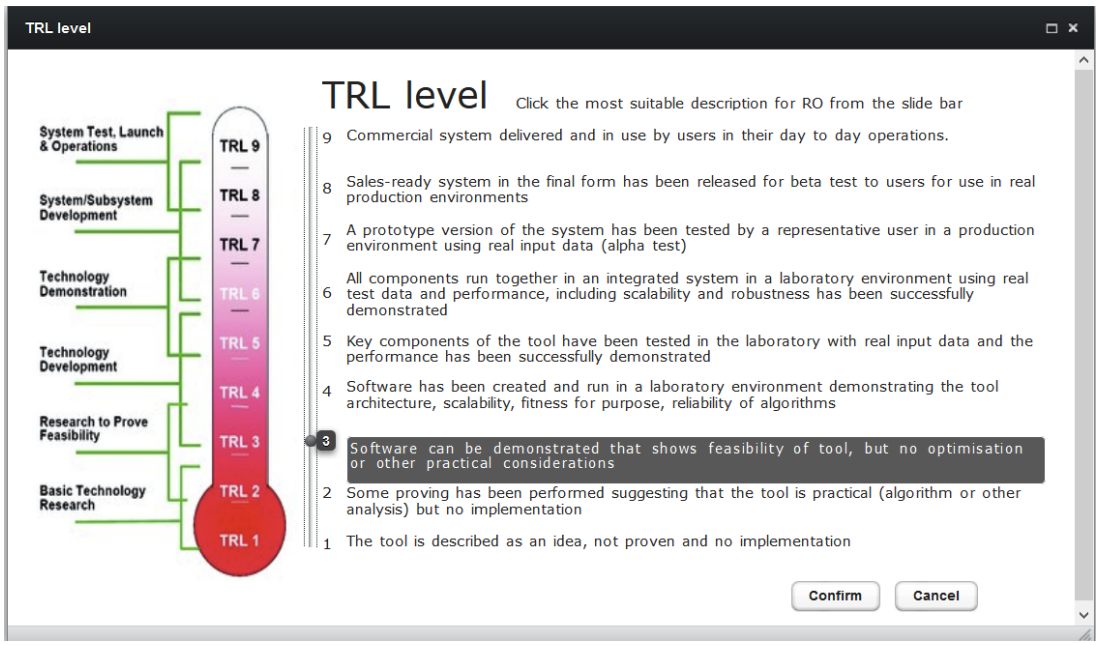

Figure 4 TRL level in PrestoKAT user interface

We have added a total of 35 research outputs during year 1 into the PrestoKAT database. The tool has undergone rigorous testing internally and our plan is to release the tool publicly in year 2 as a web service.

1.2 Research Outputs Chosen for Assessment

In deliverable D3.1 [1] we have classified ROs into seven broad categories:

- Metadata Mapping and Validation
- Storage
- Information Extraction – not assessed in this deliverable
- Quality Assessment
- Manual Content Annotation - not assessed in this deliverable
- Rights Management - not assessed in this deliverable
- Preservation Platforms/ Systems

In year 1 of Presto4U we have chosen tools belonging to 4 of these categories namely: metadata mapping and validation, storage, quality assessment, and preservation platforms and systems. In year 1 we plan to cover all remaining categories of tools. In the next section we provide description of the tools (classified by category) chosen for assessment from within the identified candidate research outputs.

1.2.1 Metadata mapping

1.2.1.1 PrestoPRIME Metadata Mapping Tool

The PrestoPRIME[3] project has also developed an approach for metadata mapping. In contrast to MINT, the approach is based on defining mappings between metadata models/schemas rather than starting from instances. The approach uses a high-level intermediate representation of metadata elements serving as a hub for mapping between formats. Therefore, hand-crafted one-to-one mappings between each pair of metadata formats are avoided, as the mappings for a specific pair of source and target format can be derived automatically. Metadata elements from a specific metadata format are described in relation to these generic elements. In addition, mapping templates on data type level are used. From these sources the code for a mapping problem between a pair of formats can be derived. The core of this approach is the *meon* ontology [6], which describes generic metadata elements and the relations between them. *meon* was originally developed to model metadata elements used throughout the audiovisual media production workflow in a format independent way in order to support content exchange and its automation. The *meon* ontology has been extended to express mapping relations between metadata formats. In addition to the ontology of generic metadata concepts, specific ontologies are created for each format taken into account. Then it is possible to infer how concepts from different metadata formats are related by observing the relations among generic concepts and to the format specific concepts. Since mapping instructions are derived from the ontology description, these mapping instructions are easier to maintain than hard coded mapping instructions. Adding a new metadata format can be done without side effects to existing definitions.

[3] http://www.prestoprime.org/

In addition, the mapping approach handles different data types and supports mappings between them. For handling hierarchical metadata structures, metadata elements can be assigned to a context (e.g., broadcast, shot, frame), and weights can are automatically assigned to mappings, in order to give precedence to direct mappings over indirect ones. Details on the mapping approach are described in PrestoPRIME D4.0.4a [7].
The mapping service is deployed as a rest web service. In order to configure the mappings, a graphical user interface (see Figure 5) has been developed. Prototypes are available at http://prestoprime.joanneum.at. The development of the mapping tool chain is continued in the EEXCESS project[4].

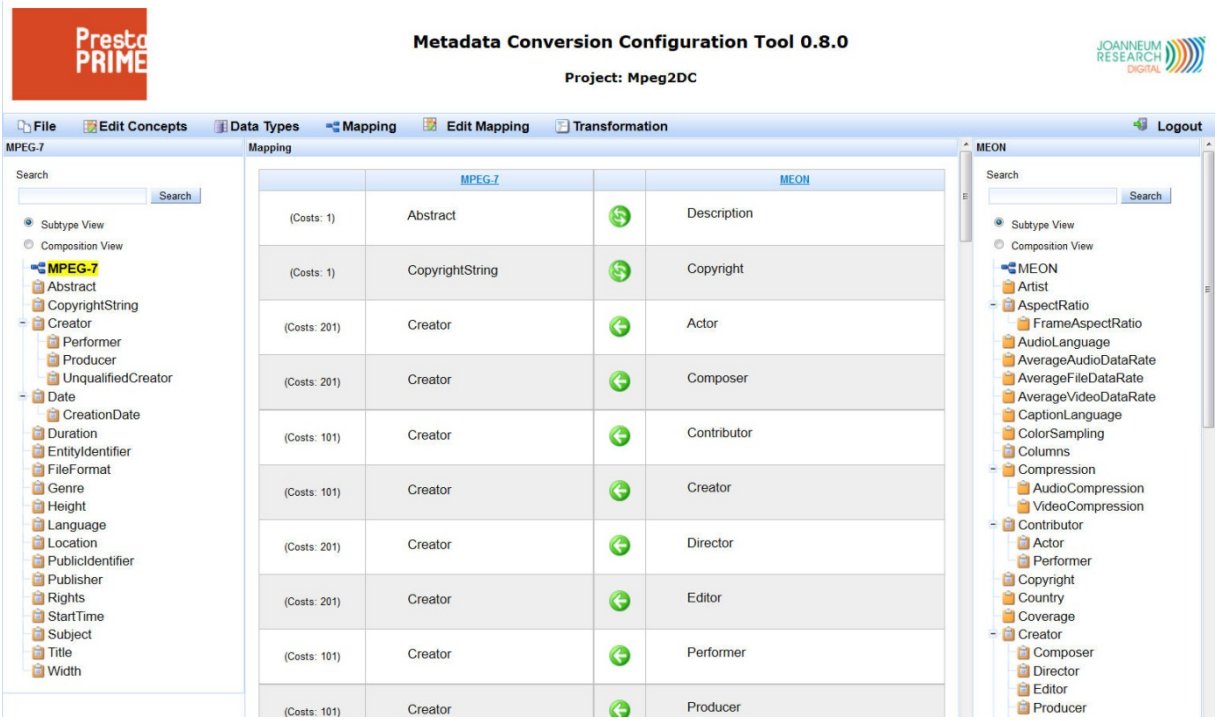

Figure 5: Metadata mapping configuration interface.

As this tool uses an intermediate ontology, the completeness of defined mappings for a certain format can be tested by mapping from a format back to the source format via the intermediate ontology. In this case, no information loss should occur.

1.2.1.2 MINT Mapping Tool

MINT offers a web based platform to facilitate the aggregation initiatives for cultural heritage content and metadata in Europe. It offers functionality for the ingestion, mapping and aggregation of metadata records, and proceeds to implement a variety of remediation approaches for the resulting repository. This framework has been used in the Athena, Carare[5] and EUscreen[6] projects (all completed) as well as in ongoing projects such as LoCloud[7], AthenaPlus[8], and EUscreenXL[9] to prepare different types of cultural heritage metadata for ingest into Europeana[10]. Mappings to a reference data model are created

[4] http://eexcess.eu/
[5] http://www.carare.eu/
[6] http://www.euscreen.eu/
[7] http://www.locloud.eu/
[8] http://www.athenaplus.eu/
[9] http://blog.euscreen.eu/euscreenxl
[10] http://www.europeana.eu/

using a visual mapping editor that displays the input and target XML file. Mapping starts from one or more metadata instances and needs as complete as possible instances. In addition, string manipulation operations, conditional mappings, and constant or controlled value assignments can be defined. All mapping instructions are represented using XSLT. The EBU[11] will soon open an installation to its members for mapping their metadata to EBUCore [8].

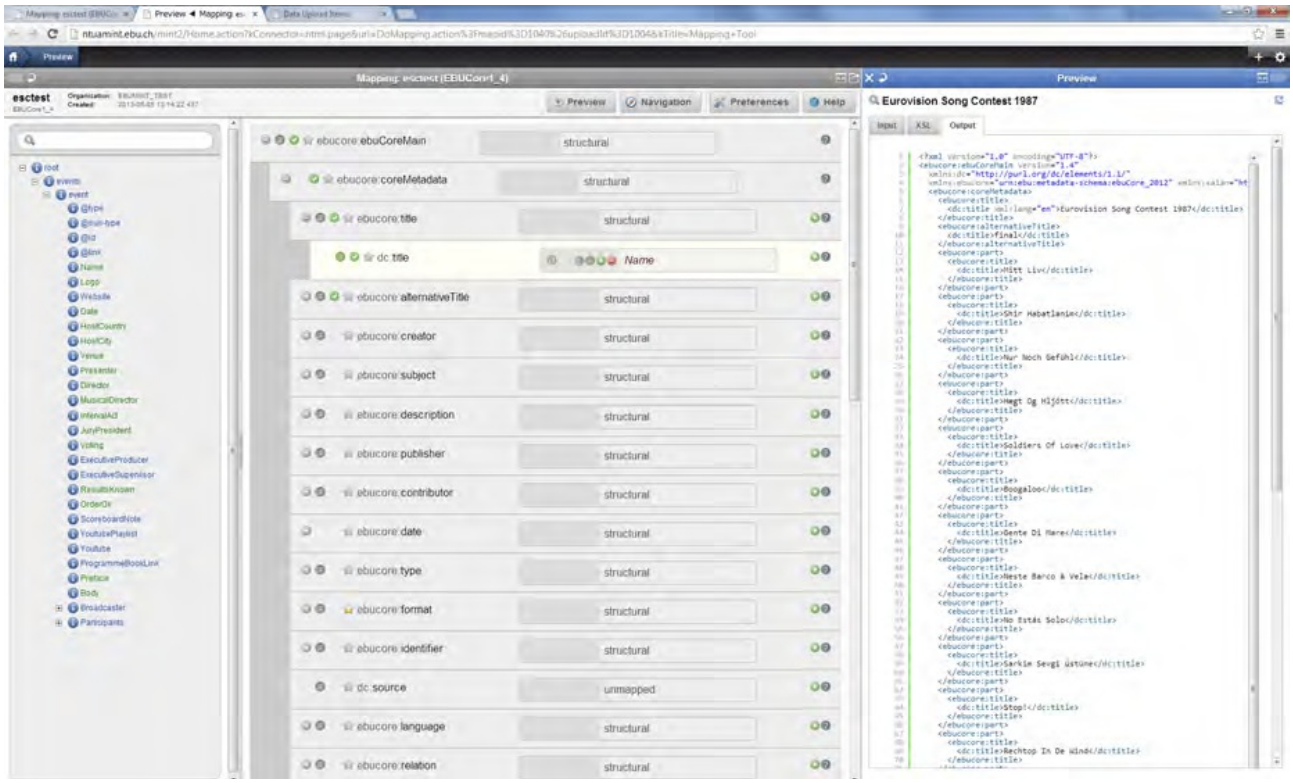

Figure 6 Metadata mapping GUI

1.2.2 Storage

1.2.2.1 MServe

MServe[12] is a RESTFul Web Framework for Service Providers for storing and processing data. MServe has been developed by IT Innovation in the PrestoPRIME FP7 ICT project and the POSTMARK[13] project. It has also been used successfully as a viable storage system. MServe's Technology Readiness Level score is above 7, which makes MServe an ideal candidate for a storage assessment tool for this phase. The following sections present a brief introduction to MServe.

MServe's framework is configurable and can provide human and machine usable interfaces to control the ingest, access, processing and manipulation of content using compute resources. Data storage is achieved through file-system mounts and there are built in processes for integrity checking and file replication and repair. Data processing is performed using the Celery distributed task queue along with RabbitMQ[14] to distribute jobs on a cluster and third-party tools.

[11] http://www3.ebu.ch/cms/en/home
[12] https://prestocentre.org/library/tools/mserve
[13] http://www.thepostmarkproject.co.uk/
[14] http://www.rabbitmq.com/

There are 3 main interfaces in MServe :

- HTML (web browser) interface for human manipulation of the content;
- HTTP/REST interface for machine workflows and automated systems;
- WebDAV10 interface to provide file system access to the content.

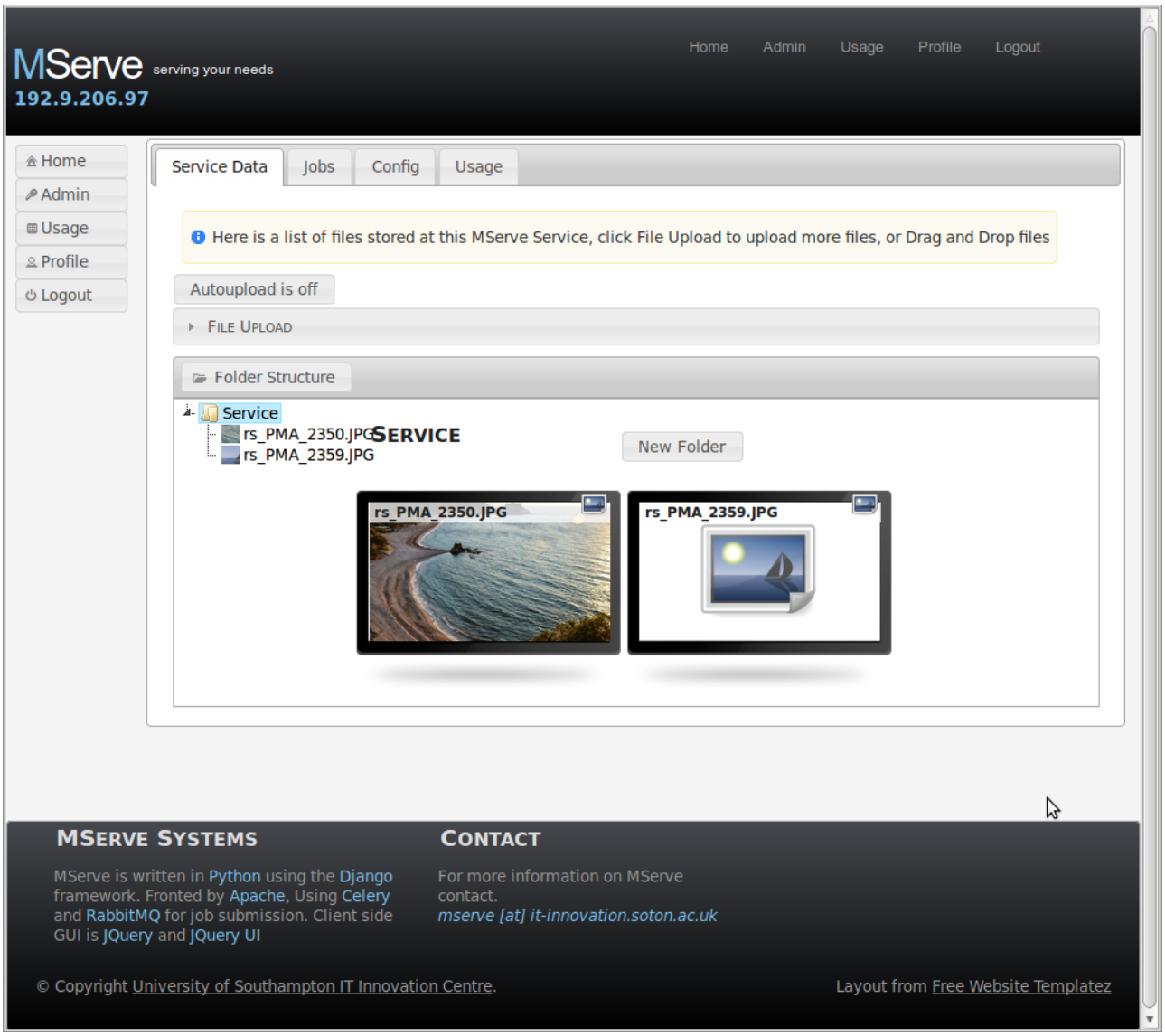

Figure 7 MServe service UI screenshot

1.2.2.1.1 MServe Architecture

MServe is a Django[15] application. Django defines the model for resources, content and their usage. Django handles requests for access ingest and manipulation of the content. It uses Apache2 as a front-end web server to provide access to static files, and bandwidth control to content. For batch processes Django uses a distributed task queue called Celery which controls queues of tasks to be processed, farming the tasks out to worker nodes. The queuing and brokering is handled by an efficient messaging service

[15] https://www.djangoproject.com/

RabbitMQ. This architecture is illustrated in Figure 8.

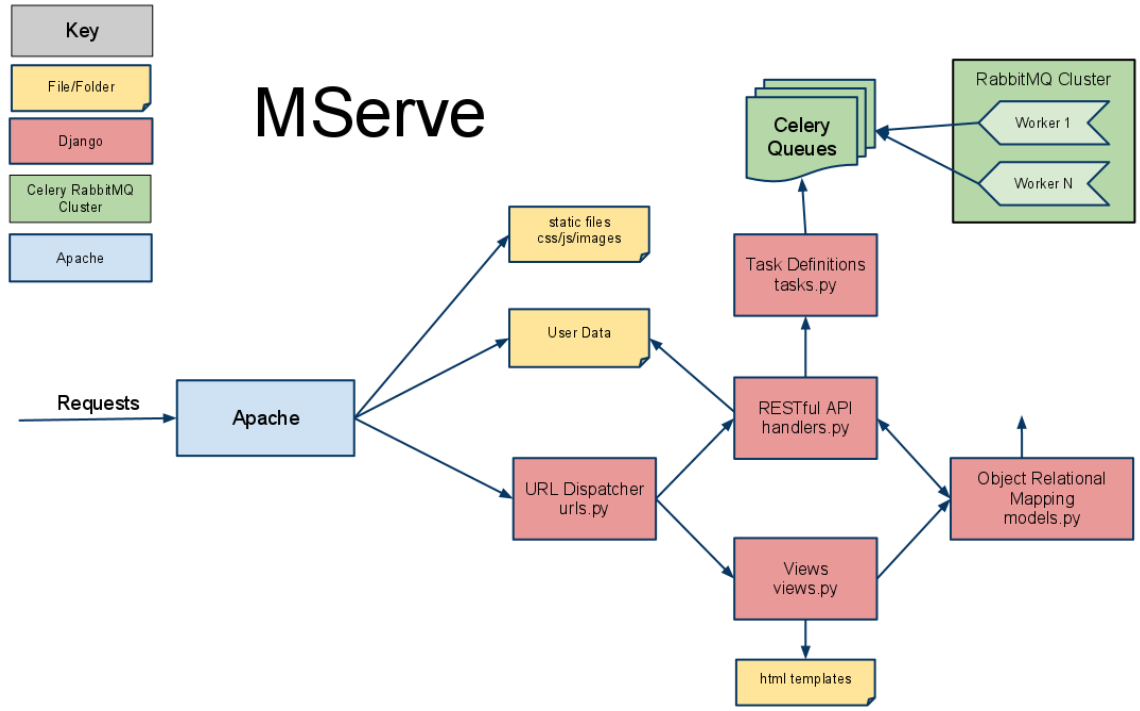

Figure 8 The MServe architecture

The Service Provider can define sets of tasks to run as part of the following stages or times:

- Ingest - When content is uploaded into MServe
- Access - When content is requested from MServe
- Update - When content is changed in MServe
- Periodic - At specified times
- Manual - At the request of the consumer or workflow engine

Tasks that can be run include (but are not limited to)

- Checksum computation and verification
- Archive and backup
- MIME type detection
- Thumbnail generation
- Video proxy generation
- Video meta-data extraction, for instance using the MXF meta-data extractor
- Archive retrieval
- Other batch processes or web services

1.2.2.1.2 Mserve Jobs, Tasks and Task Sets

MServe provides a way to define workflows for “Jobs”. In MServe the concept of a Job is a set of ordered Task Sets, Task Sets are made up of Tasks. When a Job is started (triggered either by upload, access, periodic or explicitly by the user) then the first Task

Set in the order is executed and all Tasks in the Set are executed in parallel. Once all these tasks have completed successfully, then the next Task Set in the ordering is run and so on until the workflow is complete. Each Task can be executed conditionally based on some simple conditions such as the MIME type of the file.

Figure 9 The relationship between Jobs, Task Sets and Tasks in MServe

1.2.2.1.3 Metrics

In MServe a metric is a reference to an entity that can be monitored. The process of monitoring metrics produces usage reports. MServe records a lot of data about all the processes it enacts and files that are stored. By default MServe provides the following general metrics (See Table 1) in usage reports.

Metric Name	Unit	Description
http://mserve/container	Containers	Number of Containers
http://mserve/service	Services	Number of Services
http://mserve/file	Files	Number of Files
http://mserve/backupfile	Backup Files	Number of Backup Files
http://mserve/disc	bytes	The amount of data stored
http://mserve/disc_space	bytes	How much data is currently being stored on disc
http://mserve/responsetime	seconds	Response time to serve a file
http://mserve/ingest	bytes	Bytes ingested into the system
http://mserve/access	bytes	Bytes ingested into the system
http://mserve/corruption	bytes	Bytes corrupted on the system
http://mserve/dataloss	bytes	Bytes lost in the system
http://mserve/job	Jobs	Number of Jobs
http://mserve/task	Tasks	Number of Tasks
http://mserve/jobruntime	seconds	Job Runtime

Table 1: MServe Metrics

Users can also define their own metrics and the system can produce more specific usage reports accordingly. Usage data can be viewed either from its API or in the MServe web pages (See Figure 10).

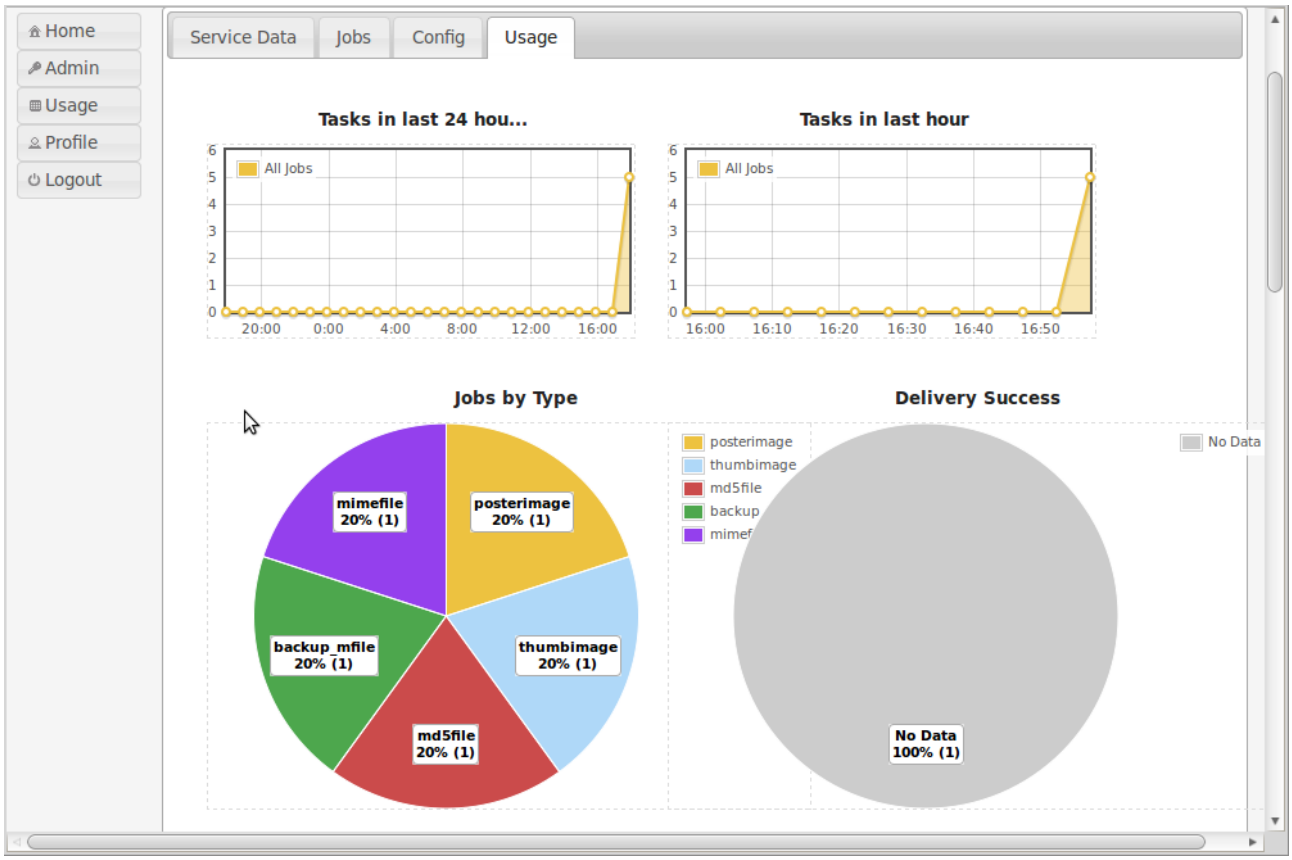

Figure 10 MServe usage reports

MServe has been developed under the EU-funded project PrestoPRIME and POSTMARK and is available under the LGPL v2.1[16] license.

1.2.2.2 LTFS Archiver

LTFSArchiver is a software which exposes a set of services for archiving media files over LTO/LTFS storage technology [9]. Until end of 2012 it was developed by RAI[17] within the European project PrestoPRIME, as documented in [10]. RAI has been continuing the development of newer versions along 2013 and is expected to carry on in 2014. The software is released under Affero General Public License v. 3.0[18].

Data tape technologies, such as LTO[19] (Linear Tape Open), are attractive for long term preservation of large audiovisual files at master quality level, because of low cost of tape versus disc based storage. However it's only with LTFS (Linear Tape File System), available since LTO v.5 and providing almost normal file access modalities with lower access latency, that this attractiveness became undeniable for having removed most of the limitations due to the tape linear nature, allowing the access to the data tape as a removable storage device with a file system. Moreover, the facts that LTFS is an open format and that open-source software drivers are freely available, make this option particularly interesting for long term preservation because it overcomes the typical issues

[16] http://www.gnu.org/licenses/lgpl-2.1.html
[17] http://www.rai.it/
[18] http://www.gnu.org/licenses/agpl-3.0.txt
[19] http://www.lto.org/

of proprietary solutions. Nevertheless LTO/LTFS is still a niche domain and requires awareness regarding the modalities of use.
LTFSArchiver provides the means for the fruition of LTO/LTFS storage technology in digital audiovisual preservation contexts. It has been conceived for use in wider frameworks, such as the PrestoPRIME Preservation Platform (P4), or by other client applications, through the defined interfaces. However it also provides its own basic GUI (Figure 11) for use by a web-browser.

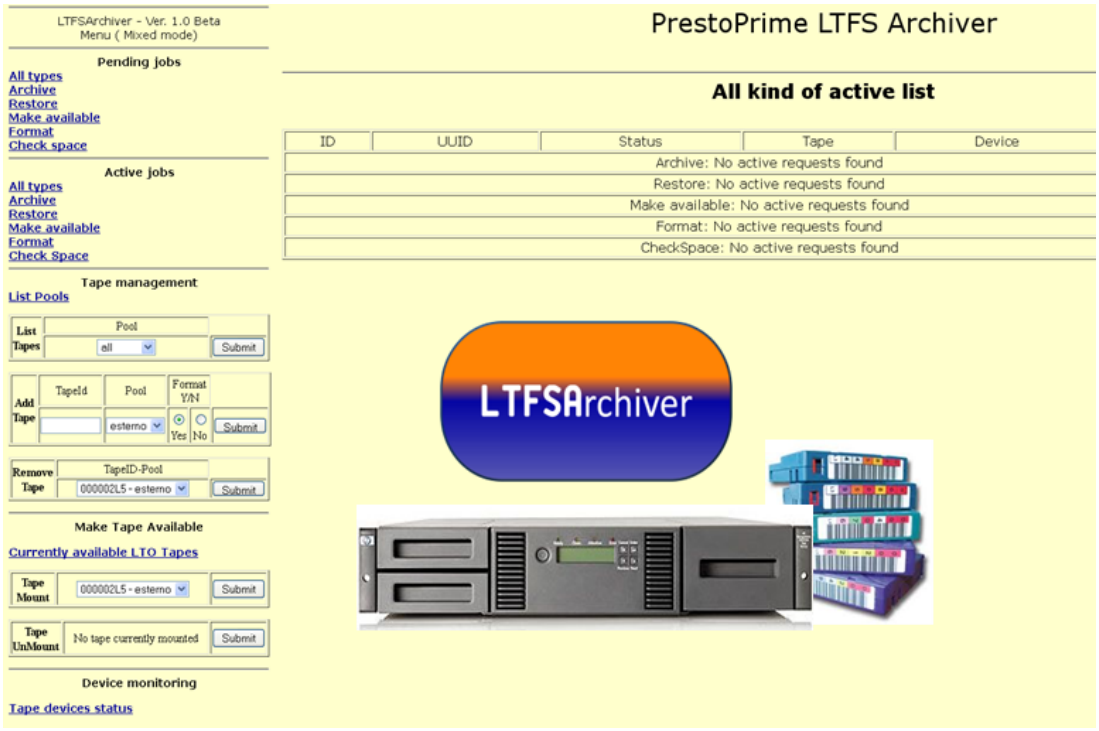

Figure 11: LTFS Archiver user interface.

LTFSArchiver offers the following major functionalities:

- Tape management – it handles a registry of LTO tapes in use, grouped in user defined “pools”, allowing allocation of data tapes on basis of the user needs, such as customer subdivision and copy redundancy, and implementing the management of the available storage capacity.
- Writing to LTO/LTFS – possible for both single files and whole folder hierarchies; the client application has to indicate the target tape pool and desired archiving options, such as creation of fixity data.
- Reading from LTO/LTFS – possible for both folder hierarchies, same or subset of the archived ones, and single files, even if archived within a folder hierarchy.
- Direct access to LTFS for processing by other components – the data tape file system is mounted and made available, in read-only, to other components that complement the LTFSArchiver functionalities, for example implementing “partial restore”.
- Direct support to creation, storage, and verification of fixity data (checksums) – Checksum computation (supported MD5 and SHA1) can be requested at archiving time or later, and the results are saved on the same tape holding the data. Subsequent requests can be done for integrity verification purposes.

Client applications can integrate LTFSArchiver by means of the documented web APIs, over HTTP or HTTPS protocol. All the service responses and outputs are returned either in XML, according to the defined XML Schema, or in the equivalent JSON format.

A few characteristics and known limitations have to be taken into account:

- LTFSArchiver is for Linux operating systems only.
- The services requiring asynchronous execution, return to the client a *TaskID*, which must be used by the client application for checking the status of the task and for retrieving its results on completion. This mechanism is compliant to that defined in [11] for non-blocking service calls.
- LTFSArchiver is not a complete preservation system and, in the current version, does not provide persistency of the content catalogue information.
- The storage areas used as source for writing to LTO and as target for reading from LTO have to be storage resources available to the server hosting the LTFSArchiver service as file systems (i.e. local disks or shares available through the file protocol).
- The current version does not provide security features for accessing the service, it must then conveniently be deployed within a safe and trusted intranet environment. It is announced that this limitation will be overcome in 2014.

1.2.3 Quality assessment

1.2.3.1 VidiCert

Ensuring quality is an essential part of moving image and video production, post-production, delivery and archive operations as well as archive migration. Quality assessment of audio-visual content is an extremely time- and therefore cost intensive part of the work. The VidiCert software enables significant automation of visual quality assessment in a two-step approach. In the first step video or movie file content is analysed, within the VidiCert Analyser application, fully automatically in regard to visual impairments, e.g. image noise, severe analogue video distortions, digital tape dropouts, monochrome frame sections or blurriness and some audio defects, e.g. for silence or unwanted changes of the audio encoding format.

Verification of the gathered information is done by an efficient, streamlined and customizable user interface within the VidiCert Summary application. The result is a human verified quality report described in the standardised MPEG-7 format[20].

[20] http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm

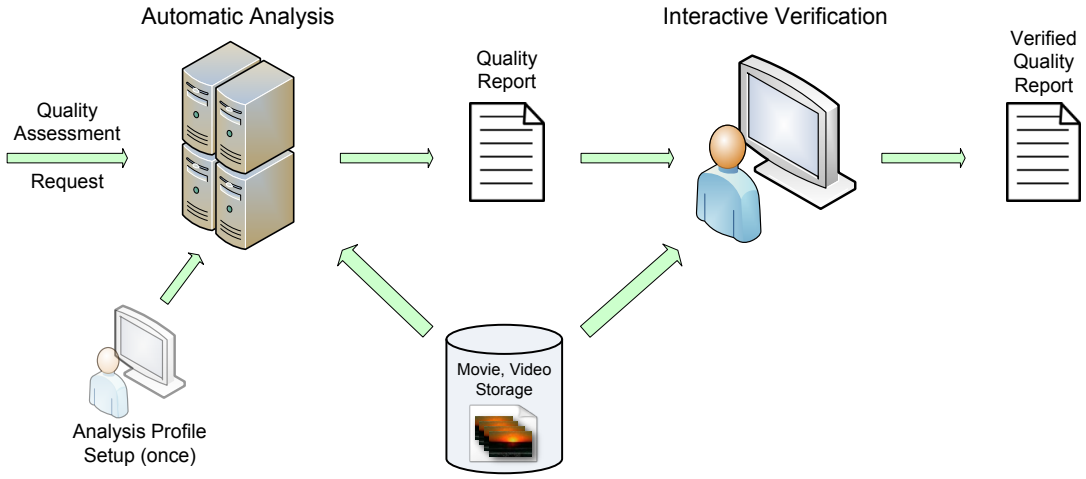

Figure 12 VidiCert system consisting of VidiCert Analyser for automatic video quality analysis and VidiCert Summary for interactive video quality verification.

VidiCert essence quality assessment capabilities are intended to speedup following workflows:

1. Efficient quality assurance for archive digitisation and migration to file based environments
2. Quality assurance of video and movie production
3. Incoming/outgoing quality assessment in video and movie post-production
4. Incoming/outgoing quality assessment in video and movie post-production
5. Estimation of the restoration effort of archived digital content
6. Quality assurance before broadcast / distribution
7. Selection of and search for content with specific quality properties in digital archives

VidiCert focuses especially on following AV archive / preservation related QC tasks:

Content Ingest/Migration

- to monitor if the video player shows problems (head clog, drop-out, video breakup, off-lock, etc.)
- to monitor the film scanning process (instability, out of focus, white/black point,etc.)
- to ingest only high quality content (no up-scaled one, etc.)
- to check the encoding/transcoding (blocking, sharpness, etc.)

Content Selection/Access/Usage

- to select my ‘best quality copy’
- to search for a video with minimum quality for a certain usage
    - noise reduction necessary?
    - sharpness high enough?

Restoration Planning

- to estimate costs / to select restoration tools & systems

VidiCert focuses on the assessment of the audio-visual quality of the video content by analysing and verifying the video essence/baseband information. It assumes that the video files to be analysed are compliant with container (e.g. MXF) and encoding (e.g. h.264) standards, so that a video file can be decoded and played properly. For this purpose a container/stream checking tool might be useful to be applied within a QC workflow before applying VidiCert for efficient essence quality checking.

VidiCert contains of two components: VidiCert Analyser is an automatic video analysis tool integrating automatic detection algorithms for defects like video breakuo, blurriness and noise. Efficient visualization and verification of defect analysis results supports an operator to get a quick overview of the condition of the material and to allow for manual corrections and final quality judgment by the operator.

In the following we describe the user interface shown in Figure 13, which is composed of these four main parts: Global timeline views (1) show the occurrence of defect events for the full temporal range of the video. A global timeline view also shows the shot structure and the temporal zoom period for the timeline views in (3). For efficient verification, a defect list component (2) shows defect events and their properties. Timeline views showing a zoomed temporal resolution providing a level of temporal detail that can be freely adjusted are shown in (3) of the user interface. A video player (4) with frame accurate positioning support and audio playback is also provided.
The video player is the central component of the user interface. All other components synchronize with it. The video player can be positioned on an extra monitor for full resolution playback. The other components like the event list component and the timeline views can also be displayed on a second monitor.

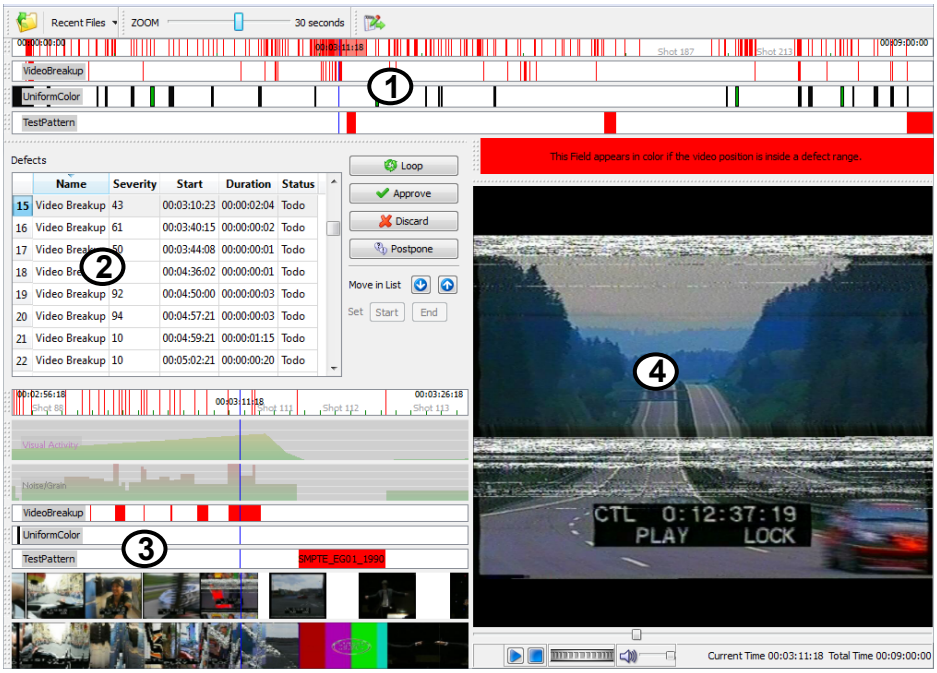

Figure 13 User Interface for efficient interactive verification of automatic detections

All components provide additional navigation functionalities. The key frame and stripe image timeline views shown in the bottom of (3) provide a quick visual overview of the video content. Key frames and stripe images are aligned on the timeline according to their respective time points. Navigation is possible by clicking on the timeline, or by moving the scroll wheel for frame accurate positioning.
Timeline views showing impairment detection results either visualize continuous or segment-based quality measures. Continues quality measures are visualized in form of line or bar charts like the visual activity and the noise/grain level within specific time ranges. Detections having an event-like character are also visualized on timeline views by indicating the temporal segment of the detection. These are for example video breakups, uniform colour, and digital tape dropout and test pattern segments. The different views appear both over the full video range in (1) and for the selected zoom period in (3). For

uniform colour detections the respective segments are additionally filled with the color detected.
The time an operator can devote to verify automatic analysis results is typically limited and it may be the case that not all defect detections can be manually verified. So the time the operator has available should be optimally used. For this it is very useful to be able to handle the most relevant detections first. To support this, the detections listed in the defect list view can be sorted by all columns. When sorting by severity an operator can efficiently verify the most relevant detections first. A detection can either be approved, discarded, or postponed for later verification by the operator. After such a manual verification the next detection in the list not yet verified will be selected. This verification process is supported by a special mode where the video will play in a loop around the currently selected detection including a configurable pre roll and post roll time.
The user interface allows also for manually adding defect events on the timeline, where defects are defined by a classification scheme. Additionally an operator can rate the overall quality of the video by selecting predefined rating classes. All the information visualised in Figure 13 and all the decisions made by an operator during his work are stored in an MPEG-7 AVDP compliant XML document.

1.2.4 Preservation platforms and systems

In the following Sections we quickly describe two digital preservation platforms which have been evaluated during the first year. Both of them have been designed and implemented in order to be compliant with the OAIS model. Several solutions are available, originating from other research projects and initiatives or from commercial vendors. In the open source community other platforms that is worth mentioning and which will taken into account for the second year evaluation include also Dspace[21] and Fedora[22] (and their derivatives, such as RODA) from Duraspace. Almost all solutions adopt METS[23] as the main wrapper for OAIS information packages, while different approaches are used for descriptive, technical and rights metadata. Concerning data curation, the typical approach makes use of a workflow engine to execute preservation tasks on stored content, while for storage the favourite solutions include disks via NFS or CIFS, LTO tapes, shared or federated storage systems and cloud storage services. Other differences are related to the possibility to customize existing workflows and integrate with existing external systems for digital content management.

1.2.4.1 Archivematica

Archivematica[24] is a free and open-source digital preservation system that is designed to maintain standards-based, long-term access to collections of digital objects. It follows the OAIS functional model and implements a micro-services design pattern to integrate software tools that allow users to process digital objects from ingest to access. Users can monitor and control the micro-services via a web-based dashboard. For content annotation

[21] http://www.dspace.org/
[22] http://fedoraproject.org/
[23] http://easydigitalpreservation.wordpress.com/2010/06/30/mets-for-transferable-metadata/
[24] https://www.archivematica.org/wiki/Main_Page

Archivematica also makes use of existing metadata standards like METIS, PREMIS[25] and Dublin Core[26] and other best practice metadata standards.
Archivematica implements format policies based on an analysis of the significant characteristics of file formats. The archive software suite supports a variety of file formats and normalisation routines based on format policies. Archivematica is maintained by Artefactual Systems[27], in collaboration with UNESCO and other institutions.

The format policies are based on the significant characteristics of the file formats. The significant properties or characteristics are defined as the "*characteristics of digital objects that must be preserved over time in order to ensure the continued accessibility, usability, and meaning of the objects, and their capacity to be accepted as evidence of what they purport to record*." Table 2 below reports the list of content types and formats supported by Archivematica. For each format the associated preservation and access formats and the corresponding normalisation tool used for transformation during ingestion is shown.

Media type	File formats	Preservation format(s)	Access format(s)	Normalization tool
Audio	AC3, AIFF, MP3, WAV, WMA	WAVE (LPCM)	MP3	FFmpeg
Email	PST	MBOX	MBOX	readpst
Email	Maildir**	Original format	MBOX	md2mb.py
Office Open XML	DOCX, PPTX, XLSX	Original format	PDF for PPTX	OpenOffice
Plain text	TXT	Original format	Original format	None
Portable Document Format	PDF	PDF/A	Original format	Ghostscript
Presentation files	PPT	Original format	PDF	OpenOffice
Raster images	BMP, GIF, JPG, JP2*, PCT, PNG*, PSD, TIFF, TGA	Uncompressed TIFF	JPEG	ImageMagick

[25] http://www.loc.gov/standards/premis/
[26] http://dublincore.org/
[27] http://www.artefactual.com/

Raw camera files/Digital Negative format**	3FR, ARW, CR2, CRW, DCR, DNG, ERF, KDC, MRW, NEF, ORF, PEF, RAF, RAW, X3F	Original format	JPEG	ImageMagick/UFRaw
Spreadsheets	XLS	Original format	Original format	None
Vector images	AI, EPS, SVG	SVG	PDF	Inkscape
Video	AVI, FLV, MOV, MPEG-1, MPEG-2,MPEG-4, SWF, WMV	FFV1/LPCM in MKV	MP4	FFmpeg
Word processing files	DOC, WPD, RTF	• ODF (WPD and RTF) • Original format (DOC)	PDF	OpenOffice

Table 2: Content types and formats supported by Archivematica with corresponding normalization tools.

The software has been released as open source and is available for free download (including source code) under the AGPL3 licence. Source code is available on GitHub[28], the documentation can be found on Archivematica wiki. The community developing and maintaining Archivematica is supported by Artefactual Systems. The current version 1.0, released on September 2013.New software code created for integration with Archivematica can be distributed under the open source license. Commercial licenses and commercial use of the Archivematica name and logo trademarks may be negotiated with Artefactual Systems on a case-by-case basis.

The user dashboard (see Figure 14) provides interface mapped onto OAIS functional entities. The web dashboard allow users to process, monitor and control the Archivematica workflow processes. The Dashboard provides a multi-user interface that will report on the status of system events and make it simpler to control and trigger specific micro-services. This interface allows users to easily add or edit metadata, coordinate AIP and DIP storage and provide preservation planning information.

[28] https://github.com/artefactual/archivematica

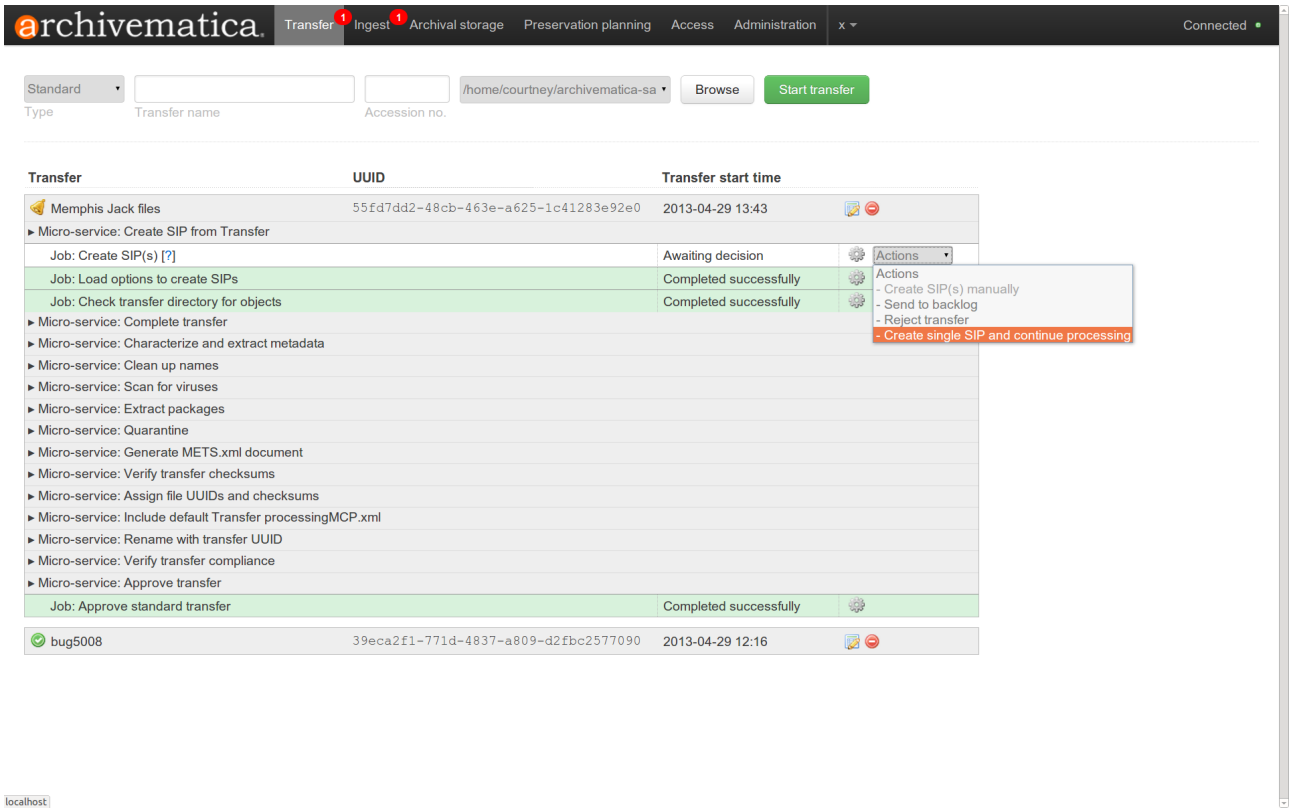

Figure 14 Archivematica Dashboard, with notification from micro-services during ingesting workflow.

Concerning the OAIS information packages, SIP is based on METS, while Library of Congress BagIt format [29](zip) is used for AIP. Archivematica supports not only DIP upload to AtoM, but also to CONTENTdm services and export to Dspace data model. DIP upload can be achieved through a guided procedure on the GUI. Concerning metadata formats, as already mentioned METS is supported for both ingest and access, while PREMIS and DC are the reference standards for preservation and descriptive metadata.

1.2.4.2 PrestoPRIME Preservation Platform (P4)

P4 is the preservation platform developed by the PrestoPRIME project, integrating tools and services developed by all partners.

P4 implements the main functional entities of the OAIS model (ingest, access, administration, data management, storage and preservation planning) for an archive managing AV content, is compliant to OAIS and is made up of three main components:

1. *core libraries*, implementing OAIS components for storage, metadata management, ingest, access, administration and preservation actions;
2. *web server*, providing REST interfaces for interacting with the archive;
3. *web user interface*, providing ingest, access and administrative functionalities according to the user profile.

The web server provides interfaces for ingest, access and administration. The user can ingest SIP files into the platform, get information about the status of the submitted jobs and

[29] http://www.dcc.ac.uk/resources/external/bagit-library

of the whole system, search for AIP available in the archive, and get access to the DIP, through the web interface. The user interface manages local users and can connect to multiple P4 instances with different user identifiers, each associated to a specific role (consumer, producer, administrator) for that platform. The external tools and services can be integrated using a plug-in framework, the motivations for this being twofold: on one hand it provides a flexible way to integrate new components (e.g. to execute some specific steps during ingestion), on the other hand the platform and the core components are decoupled from specific tools or scenarios and P4 users have access to an open framework which can be used out-of-the-box, by configuring a minimum set of parameters. P4 includes a workflow engine, a lightweight execution environment to configure custom tasks based on external tools and services, exploiting the APIs of core modules. The external tools used to implement a specific workflow can be deployed within a P4 plug-in.

The data model makes use of METS as the main wrapper format for descriptive and technical metadata, as well as for mapping AV resources within the AIP. Other metadata standards are supported, such as MPEG-7 for technical metadata, PREMIS for preservation events, MPEG-21 for rights representation, DublinCore for descriptive metadata and others. P4 also supports DNX, a metadata format built on top of PREMIS vocabulary, used in Rosetta. Using P4 plugins, virtually any metadata standard can be used in the AIP. Access interface supports also OAI-PMH protocol.

Tools developed within the project and integrated in P4 cover metadata extraction (e.g. MXF tools), quality assessment, storage (disks via NFS or CIFS, LTO tapes, shared or federated storage systems such as iRODS[30] and MServe), emulation (Multivalent), SLA and monitoring, rights (Rightsdraw2), search and indexing (Solr), AV material segmentation and access, format migration, fixity checks. The preferred integration mechanism is making use of REST interfaces over HTTP, to get loose coupling and reduce dependencies.

Concerning the storage configuration, different workflows have been tested in PrestoPRIME. In particular the configuration with two copies of the master quality file has been implemented either with LTO tapes (two copies on two different tapes) or with iRODS as policy-driven storage (the automatic replica rule, with periodic fixity checks is defined).

The current version of the platform is 2.2.0 (December 2012), the source code is available under the GNU GPL v3 on Github[31], while the documentation (deliverables, user and developer guides) can be accessed from the PrestoCentre[32].

[30] https://www.irods.org
[31] https://github.com/prestoprime/p4
[32] https://prestocentre.org/library/tools/p4

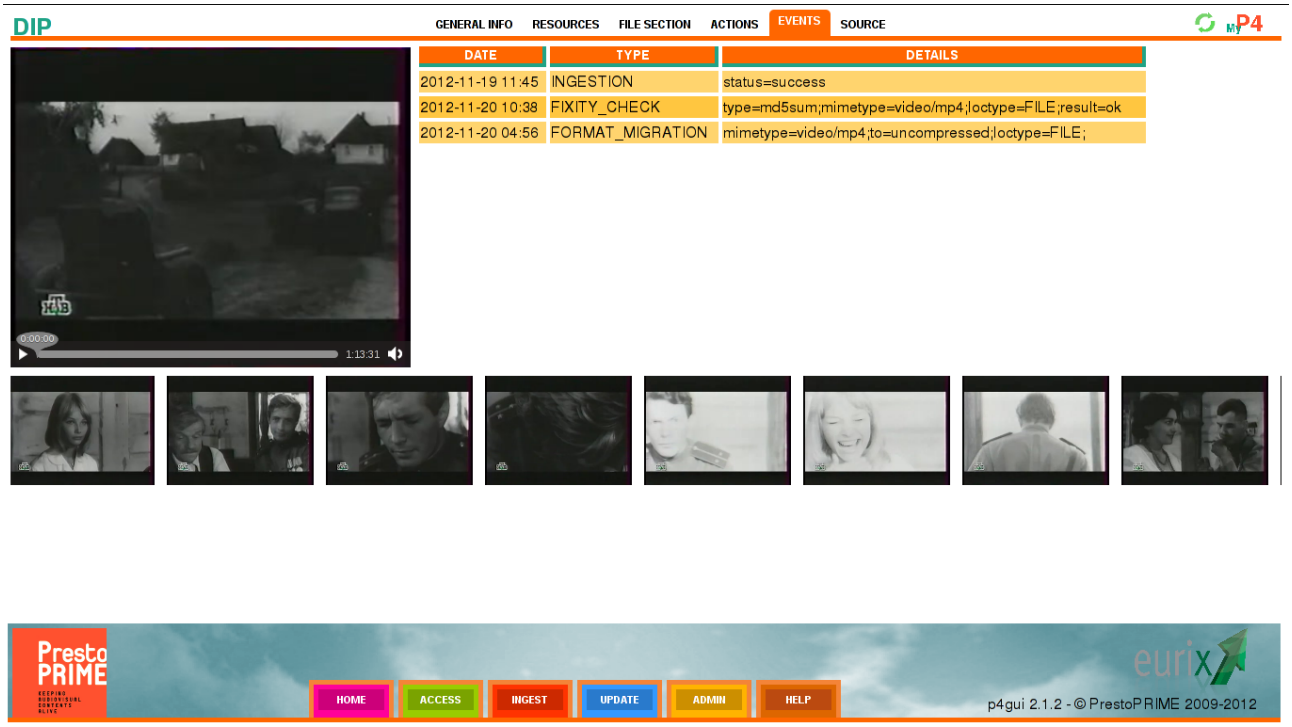

Figure 15 P4 access interface and DIP preview: for each DIP the descriptive and technical metadata are reported, as well as the information about AV files and the preservation events.

The data model makes use of METS as the main wrapper format for descriptive and technical metadata, as well as for mapping AV resources within the AIP. Other metadata standards are supported, such as MPEG-7 for technical metadata, PREMIS for preservation events, MPEG-21 for rights representation, DublinCore for descriptive metadata and others. P4 also supports DNX, a metadata format built on top of PREMIS vocabulary, used in Rosetta (a commercial solution for digital preservation developed by ExLibris). Using P4 plug-ins, virtually any metadata standard can be used in the AIP.

Access interface supports also OAI-PMH protocol. The data model is tailored to broadcast environment (editorial entities, master and browsing qualities, B2B contracts). No compressed formats such as zip, BagIt or tarball used for AIP, METS contains references to metadata and AV files. P4 develoment focus on videos, but other content types can be supported defining new workflows.

Ingest and access are provided by a web UI or REST APIs, using METS as unique format for all OAIS information packages, common to other platforms. An advanced search engine based on Solr allows indexing of different descriptive and technical metadata. Several solutions are available for Archival Storage, supporting local and distributed storage. Data Management and Administration are provided by the P4 web UI, including monitoring of jobs and workflows (see Figure 16).

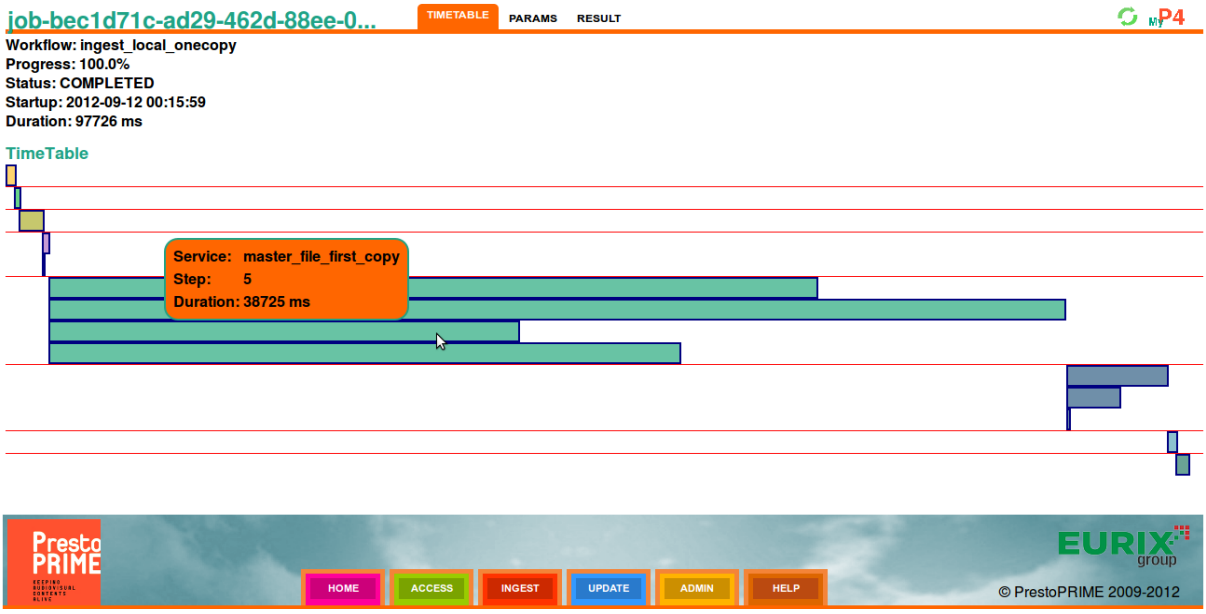

Figure 16 P4 administrative interface: monitoring of jobs and workflows

2 Assessment Criteria

This section describes in more detail the assessment criteria for the selected RO categories, including measurement plans. The characteristics and sub-characteristics considered for the assessment of the RO, according to the standards ISO/IEC 25023, are listed in Table 3. D3.1 [1] provides full details and definition of each of these characteristics. In this section we will not repeat the definitions again but specify an implementation criteria on how they can be measured and quantified. Firstly we present an overview of the assessment criteria in general and in subsequent sections we will specialise the assessment criteria for each category of tools (see Section 1.2) identified.

CHARACTERISTICS	SUB-CHARACTERISTICS
Functional suitability	Functional completeness
	Functional correctness
	Functional appropriateness
Performance efficiency	Time behaviour
	Resource utilization
	Capacity
Compatibility	Co-existence
	Interoperability
Usability	Appropriateness recognisability
	Learnability
	Operability
	User error protection
	User interface aesthetics
	Accessibility
Reliability	Maturity
	Availability
	Fault tolerance
	Recoverability
Security	Confidentiality
	Integrity
	Non-repudiation
	Accountability
	Authenticity
Maintainability	Modularity
	Reusability
	Analysability
	Modifiability
	Testability
Portability	Adaptability
	Installability
	Replaceability

Table 3: Characteristics and Sub-characteristics

Table 4 shows the in more detail each of RO categories and their matching characteristics, sub-characteristics as identified in D3.1.

CHARACTERISTICS	SUB CHARACTERISTICS	Metadata Mapping and Validation	Storage	Information Extraction	Quality Assessment	Manual Content Annotation	Rights Management	Preservation Platforms/ Systems
Functional suitability	Functional completeness	X	X	X	X	X	X	X
	Functional correctness	X		X	X	X	X	
	Functional appropriateness	X	X			X	X	X
Performance efficiency	Time behaviour	X	X	X	X			
	Resource utilization	X	X	X	X			
	Capacity	X	X	X	X			X
Compatibility	Co-existence		X					X
	Interoperability	X	X	X	X	X	X	X
Usability	Appropriateness recognisability	X			X	X		
	Learnability	X			X	X		
	Operability	X	X		X	X	X	X
	User error protection	X	X		X	X	X	X
	User interface aesthetics	X			X	X		
	Accessibility	X			X	X		
Reliability	Maturity		X					X
	Availability		X					X
	Fault tolerance	X	X	X	X			X
	Recoverability	X	X		X			X
Security	Confidentiality		X				X	X
	Integrity		X				X	X
	Non-repudiation		X				X	X
	Accountability		X		X	X	X	X
	Authenticity		X				X	X
Maintainability	Modularity	X	X	X	X		X	X
	Reusability	X		X	X		X	
	Analysability		X					X
	Modifiability		X					X
	Testability		X		X		X	X
Portability	Adaptability		X				X	X
	Installability		X		X		X	X
	Replaceability		X					X

Table 4: Mapping between the characteristics ISO-IEC 25010 and the tools and platforms aggregation tasks

In the following sections we define the measurement functions for each main characteristic and its sub-characteristics. During the assessment of ROs, we have tried to define tests to measure each of these characteristics. However, in some instances it is not possible to measure each and every characteristic as defined in Table 4.

2.1 General Assessment Criteria

2.1.1 Functional suitability

Degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions

1. Measurement function: FS=(X+Y+Z) / 3(where X,Y,Z are the scores computed as in the following
2. Interpretation of test results: FS value closer to 1 is better

Functional Completeness:

degree to which the set of functions covers all the specified tasks and user objectives

1. Possible Measure: Functional implementation coverage
2. Description of the measure: How complete is the implementation according to requirement specifications?
3. Measurement function:

   X=(X1+X2*0.5+X3*0.25)/1.75 with

   X1=1-(A/B) where

   A= Number of missing or unsatisfying mandatory functions

   B= Number of mandatory functions assessed in the evaluation

   X2= 1-C/D

   C= Number of missing or unsatisfying desirable functions assessed in the evaluation

   D= Number of recommended functions

   X3= 1-E/F

   E= Number of missing or unsatisfying desirable functions assessed in the evaluation

   F= Number of desirable functions

   NOTE: that the evaluation will be calculated by presence/absence of the function in the RO. In particular 0 value is associated to the presence of the function and 1 to its the absence.
4. Interpretation of test results: X value closer to 1 is better

Functional Correctness:
Degree to which a product or system provides the correct results with the needed degree of precision.

1. Possible Measure: Correctness of data items
2. Description of the measure: How much the required specific accuracy of data items are complied with?
3. Measurement function:

   According to the list provided for the Functional Completeness, for each of the implemented functions in the RO, provide a score of correctness among (0, 0.5, 1) where 0= not correct; 0.5 = partially correct; 1= completely correct.

   Y= A/B where
   A= sum of the scores of the implemented functions;
   B= total amount of implemented functions
5. Interpretation of test results: Y value closer to 1 is better

Functional Appropriateness:
Degree to which the functions facilitate the accomplishment of specified tasks and objectives.

1. Possible Measure: functional appropriateness
2. Description of the measure: How many functions with no problem are implemented for the appropriate functions for pursuing a specific task.
3. Measurement function:

   According to the list provided for the Functional Completeness, for each of the implemented functions in the RO, provide a score of appropriateness among (0, 0.5, 1) where 0= not appropriate; 0.5 = partially appropriate; 1= completely appropriate.

   Z= A/B where
   A= sum of the scores of the implemented functions;
   B= total amount of implemented functions
6. Interpretation of test results: Z value closer to 1 is better

2.1.2 Performance efficiency

Performance relative to the amount of resources used under stated conditions.

1. Measurement function: PE= (X+Y+Z) / 3 where X,Y,Z are the scores computed as in the following
2. Interpretation of test results: PE smaller is better

Time behaviour:

Degree to which the response and processing times and throughput rates of a product or system, when performing its functions, meet requirements.

The possible measures for *time behaviour* are: response time, turnaround time, throughput or others similar. In case of the utilization of different measures the value for the *time behaviour* is the mean value of the calculated single scores.

Here below the Response time is presented as an example.

1. Possible Measure: (Mean) Response time
2. Description of the measure: Duration from giving a command to start a batch of tasks till receiving the first response
3. Measurement function:

   X = (B – A) /C
   A = time of entering a command
   B = time of receiving the first response
   C = time criteria specifying maximum allowable waiting duration from entering request to receiving response
4. Interpretation of test results: X (mean value of X) varies from 0 to infinite. Usually, smaller is better.

Resource utilization:

Degree to which the amounts and types of resources used by a product or system when performing its functions meet requirements

The possible measures for *resource utilization* are: CPU utilization, memory utilization, I/O device utilization or others similar. In case of the utilization of different measures the value for the *resource utilization* is the mean value of the calculated single scores.

Here below the CPU utilization is presented as an example.

1. Possible Measure: (Mean) CPU utilization
2. Description of the measure: How much CPU time is used to perform a given task
3. Measurement function:

   Y = B/A where
   A = system operation time
   B = the amount of CPU time actually used to perform a task
4. Interpretation of test results: Y smaller is better.

Capacity:

Degree to which the maximum limits of a product or system parameter meet requirements

The possible measures for *capacity* are: no. of online requests, no. of simultaneous accesses, bandwith of transmission system or others similar. In case of the utilization of different measures the value for the *capacity* is the mean value of the calculated single scores.

Here below the no. of online requests is presented as an example.

1. Possible Measure: (Max) no. of online requests

Description of the measure: How many online requests can be processed per unit of time

2. Measurement function:

   Z = A/B where
   A = operation time
   B = the total no. of online requests processed
3. Interpretation of test results: Z smaller is better.

2.1.3 Compatibility

Degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment.

1. Measurement function: Co= (X+Y) / 2 where X,Y are the scores computed as in the following:
2. Interpretation of test results: Co value larger is better

Co-existence:
Degree to which a product can perform its required functions efficiently while sharing a common environment and resources with other products, without detrimental impact on any other product

1. Possible Measure: Available co-existence

2. Description of the measure: How flexible is the product in sharing its environment with other products without adverse impacts on other products
3. Measurement function:

   X = A / B where

   A= Number of entities with which product can co-exist in operation

   B= Established number of entities in that require co-existence
4. Interpretation of test results: X varies from 0 to infinite. Usually, larger is better.

Interoperability:
Degree to which two or more systems, products or components can exchange information and use the information that has been exchanged

1. Possible Measure: Data exchangeability
2. Description of the measure: How accurately is implementation of data exchange format determined between linking systems.
3. Measurement function:

   Y = A / B where
   A = number of data formats regarded as being smoothly exchanged with other software or systems
   B = total number of data formats to be exchanged

4. Interpretation of test results: X varies from 0 to infinite. Usually, larger is better.

2.1.4 Usability

Degree to which a product or system can be used by specified users to achieve specified goals with eff ectiveness, efficiency and satisfaction in a specified context of use.

1. Measurement function: Us = (H+J+K+L+M+N) / 6 where H, J, K, L, M, N are the scores computed as in the following
2. Interpretation of test results: Us value closer to 1 is better

Appropriateness Recognizability:
Degree to which users can recognize whether a product or system is appropriate for their needs.

1. Possible Measure: Description completeness
2. Description of the measure: What proportion of functions (or types of function) are described as understandable in the product description?
3. Measurement function:

   H=A/B
   A= Number of functions (or types of functions) described as understandable in the product description
   B= Total number of functions (or types of functions)
4. Interpretation of test results: X varies from 0 to 1. Usually, closer to 1 is better.

Learnability:
Degree to which a product or system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use.

1. Possible Measure: Completeness of user documentation and /or help facility
2. Description of the measure: What proportion of functions are correctly described in the user documentation and/or help facility?
3. Measurement function:

   J = A / B
   A= Number of functions described correctly in user documentation
   B= Total of number of functions implemented
4. Interpretation of test results: J varies from 0 to 1. Usually, closer to 1 is better.

Operability:
Degree to which a product or system has attributes that make it easy to operate and control**.**

The possible measures for *operability* are: operational consistency, message clarity or others similar. In case of the utilization of different measures the value for the *operability* is the mean value of the calculated single scores.

Here below the operational consistency is presented as an example.

1. Possible Measure: Operational consistency
2. Description of the measure: How consistently can similar operations be carried out?
3. Measurement function:

   K = 1 – A / B
   A = number of operations that behave inconsistently
   B= total number of operations that behave similarly.
4. Interpretation of test results: K varies from 0 to 1. Usually, closer to 1 is better.

User error protection:
Degree to which a system protects users against making errors.

1. Possible Measure: Avoidance of incorrect operation
2. Description of the measure: How many functions have incorrect operation avoidance capability?
3. Measurement function:

   L = A / B
   A = number of functions implemented to avoid critical or serious malfunctions being caused by incorrect operation
   B = total number of incorrect operation patterns
4. Interpretation of test results: L varies from 0 to 1. Usually, closer to 1 is better.

User interface aesthetics:
Degree to which a user interface enables pleasing and satisfying interaction for the user

1. Possible Measure: Appearance customizability of user interface
2. Description of the measure: What proportion of user interface elements can be customised in appearance?
3. Measurement function:

   M = A / B
   A=Number of types of interface elements that can be customized.
   B=Total number of types of interface elements
4. Interpretation of test results: M varies from 0 to 1. Usually, closer to 1 is better.

Accessibility:
Degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use

1. Possible Measure: Physical accessibility
2. Description of the measure: What proportion of functions can a user with a physical handicap access?
3. Measurement function:

   N = A / B where
   A = number of functions accessible by the disabled person.
   B = total number of functions implemented
4. Interpretation of test results: N varies from 0 to 1. Usually, closer to 1 is better.

2.1.5 Reliability

Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.

1. Measurement function: Re = (H+J+K+L) / 4 where H, J, K, L are the scores computed as in the following
2. Interpretation of test results: Re value closer to 1 is better

Maturity*:*
Degree to which a system meets needs for reliability under normal operation.

The possible measures for *maturity* are: fault removal, Mean time between failures (MTBF) or others similar. In case of the utilization of different measures the value for the *maturity* is the mean value of the calculated single scores.

Here below the fault removal is presented as an example.

1. Possible Measure: Fault removal
2. Description of the measure: What proportion of detected faults have been corrected?
3. Measurement function:

   H= A / B
   A=Number of corrected faults in design/coding/testing phase
   B= Number of faults detected in review or testing
4. Interpretation of test results: H varies from 0 to 1. Usually, closer to 1 is better.

Availability:

Degree to which a system, product or component is operational and accessible when required for use.

The possible measures for *availability* are: service time ratio, mean down time or others similar. In case of the utilization of different measures the value for the *availability* is the mean value of the calculated single scores.

Here below the service time ratio is presented as an example.

1. Possible Measure: Service time ratio
2. Description of the measure: What proportion of system service time is actually provided?
3. Measurement function:

   J = A/B where
   A = system service time actually provided
   B = system service time regulated in the operational schedule
4. Interpretation of test results: J varies from 0 to 1. Usually, closer to 1 is better.

Fault tolerance:

Degree to which a system, product or component operates as intended despite the presence of hardware or software faults.

The possible measures for *fault tolerance* are: failure avoidance, redundancy (components) or others similar. In case of the utilization of different measures the value for the *fault tolerance* is the mean value of the calculated single scores.

Here below the failure avoidance is presented as an example.

1. Possible Measure: Failure avoidance
2. Description of the measure: How many fault patterns were brought under control to avoid critical and serious failures?
3. Measurement function:

   K = A / B
   A= Number of avoided critical and serious failure occurrences
   B= Number of executed test cases of fault pattern (almost causing failure) during testing
4. Interpretation of test results: K varies from 0 to 1. Usually, closer to 1 is better.

Recoverability:

Degree to which, in the event of an interruption or a failure, a product or system can recover the data directly affected and re-establish the desired state of the system

1. Possible Measure: (Mean) recovery time
2. Description of the measure: What is(the average) time the system takes to complete recovery from a failure?
3. Measurement function:

   L = 1 / A where
   A= Time to recover the downed software /system

4. Interpretation of test results: L varies from 0 to 1. Usually, closer to 1 is better.

2.1.6 Security

Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization

1. Measurement function: Se = (H+J+K+L+M) /5 where H, J, K, L , M are the scores computed as in the following
2. Interpretation of test results: Se value closer to 1 is better

Confidentiality:

Degree to which a product or system ensures that data are accessible only to those authorized to have access

The possible measures for *confidentiality* are: access controllability, data encryption or others similar. In case of the utilization of different measures the value for the *confidentiality* is the mean value of the calculated single scores.

Here below the access controllability is presented as an example.

1. Possible Measure: access controllability
2. Description of the measure: How controllable is the accesses to the system?
3. Measurement function:

   H = A/B where
   A= Number of detected different types of illegal operations
   B= Number of types of illegal operations in the specification
4. Interpretation of test results: H varies from 0 to 1. Usually, closer to 1 is better.

Integrity:

Degree to which a system, product or component prevents unauthorized access to, or modification of computer programs or data.

1. Possible Measure: Data corruption prevention
2. Description of the measure: To what extent can the data corruption be prevented?
3. Measurement function:

   J = 1 – A/B
   A = number of data corruption instances actually occurring
   B = number of accesses where data damage or breakage is expected to occur.
4. Interpretation of test results: J varies from 0 to 1. Usually, closer to 1 is better.

Non-repudiation:

Degree to which actions or events can be proven to have taken place, so that the events or actions cannot be repudiated later

1. Possible Measure: utilization of digital signature
2. Description of the measure: What proportion of events requiring non- repudiation are processed using digital signature?

3. Measurement function:

   K = A/B

   A = number of events processed using digital signature
   B = number of events requiring non- repudiation property.

4. Interpretation of test results: K varies from 0 to 1. Usually, closer to 1 is better.

Accountability:

Degree to which the actions of an entity can be traced uniquely to the entity

1. Possible Measure: Access auditability
2. Description of the measure: How complete is the audit trail concerning the user access to the system and data?
3. Measurement function:

   L = A/B

   A = number of accesses to system and data recorded in the system log
   B = number of accesses actually occurred.

4. Interpretation of test results: L varies from 0 to 1. Usually, closer to 1 is better.

Authenticity:

Degree to which the identity of a subject or resource can be proved to be the one claimed

1. Possible Measure: Authentication methods
2. Description of the measure: How well does the system authenticate the identity of a subject or resource?
3. Measurement function:

   M = A/B

   A= number of provided authentication methods (e.g., ID/password or IC card)
   B= total number of authentication methods specified in the requirements (e.g., ID/password or IC card)

4. Interpretation of test results: M varies from 0 to 1. Usually, closer to 1 is better.

2.1.7 Maintainability

Degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers.

1. Measurement function: Ma = (H+J+K+L+M) / 5 where H, J, K, L , M are the scores computed as in the following
2. Interpretation of test results: Ma value closer to 1 is better

Modularity:

Degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components

1. Possible Measure: Condensability
2. Description of the measure: How strong is the relation between the components in a system or computer program?
3. Measurement function:

H = A / B

A = number of components which are not impacted from changes of other components directly and severely.
B = total number of discrete components.

4. Interpretation of test results: H varies from 0 to 1. Usually, closer to 1 is better.

Reusability:

Degree to which an asset can be used in more than one system, or in building other assets

1. Possible Measure: Execution of reusability
2. Description of the measure: How many assets can be reusable?
3. Measurement function:

J = A / B

A = number of assets actually reused in implementing the other system
B = total number of reusable assets in a system

4. Interpretation of test results: J varies from 0 to 1. Usually, closer to 1 is better.

Analysability:

Degree of effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures or to identify parts to be modified.

The possible measures for *analysability* are: audit trail capability, diagnosis function sufficiency or others similar. In case of the utilization of different measures the value for the *analysability* is the mean value of the calculated single scores.

Here below the audit trail capability is presented as an example.

1. Possible Measure: audit trail capability
2. Description of the measure: Can users easily identify specific operation which caused failure?
3. Measurement function:

K = A/B

A= Number of operational data actually recorded during operation
B= Number of data planned to be recorded enough to monitor status of system/software during operation

4. Interpretation of test results: K varies from 0 to 1. Usually, closer to 1 is better.

Modifiability:

Degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality

The possible measures for *modifiability* are: modification complexity, modification success rate or others similar. In case of the utilization of different measures the value for the modifi*ability* is the mean value of the calculated single scores.

Here below the modification complexity is presented as an example.

1. Possible Measure: Modification complexity

2. Description of the measure: Can the maintainer easily modify the software to meet some modification requirements?
3. Measurement function:

   L = 1 / A

   A= Work time spent to modify
4. Interpretation of test results: L varies from 0 to 1. Usually, closer to 1 is better.

Testability:

Degree of effectiveness and efficiency with which test criteria can be established for a system, product or component and tests can be performed to determine whether those criteria have been met

1. Possible Measure: Functional completeness of embedded test functions
2. Description of the measure: How completely are test functions and facilities implemented?
3. Measurement function:

   M = A/B

   A = number of test functions implemented as specification
   B = number of required test functions.
4. Interpretation of test results: M varies from 0 to 1. Usually, closer to 1 is better.

2.1.8 Portability

Degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another.

1. Measurement function: Po = (X+Y+Z) / 3 where X, Y, Z are the scores computed as in the following
3. Interpretation of test results: Po value closer to 1 is better

Adaptability:

Degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments.

The possible measures for *adaptability* are: hardware environmental adaptability, system software environmental adaptability or others similar. In case of the utilization of different measures the value for the *adaptability* is the mean value of the calculated single scores.

Here below the Hardware environmental adaptability is presented as an example.

1. Possible Measure: Hardware environmental adaptability
2. Description of the measure: is software system capable enough to adapt itself to different hardware environment?
3. Measurement function:

   X = 1 - A/B

   A= Number of operational functions of which tasks were not completed or not enough resulted to meet adequate levels during testing

B= Total number of functions which were tested in different hardware environment

4. Interpretation of test results: X varies from 0 to 1. Usually, closer to 1 is better.

Installability:

Degree of effectiveness and efficiency with which a product or system can be successfully installed and/or uninstalled in a specified environment

The possible measures for *installability* are: installation time efficiency, ease of installation or others similar. In case of the utilization of different measures the value for the *installability* is the mean value of the calculated single scores.

Here below the Installation time efficiency is presented as an example.

1. Possible Measure: installation time efficiency
2. Description of the measure: How much time and trouble is required to make an install?
3. Measurement function:

   Y = 1/A

   A = total elapsed time to install a system successfully.
4. Interpretation of test results: Y varies from 0 to 1. Usually, closer to 1 is better.

Replaceability:

Degree to which a product can be replaced by another specified software product for the same purpose in the same environment.

1. Possible Measure: User support function consistency
2. Description of the measure: How consistent is the new component with the existing user interface?
3. Measurement function:

   Z = 1 – A/B

   A = number of new functions that are considered not to be consistent with user's expectations

   B = number of new functions
4. Interpretation of test results: Z varies from 0 to 1. Usually, closer to 1 is better.

2.2 Metadata mapping and validation

In this section a measurement plan for metadata mapping and validation RO category is refined as a more complete example of assessment. Firstly, we start by defining the functions required to be tested followed by a measurement plan on specific functions which need to be specialised (specialisation from the generalised criteria mentioned in the section above) for this particular category. The levels of need are classified as follows:

- Mandatory - Must have
- Recommended - Could deal also without, but it would be better to have
- Desirable - May be appreciated in some cases, but in most cases it doesn't make the difference

2.2.1 Definition of functions

Functions	**Levels of Need**	**Description**
Metadata input formats		Support for a metadata format as source format of the mapping process
Dublin Core	Mandatory	
ESE	Desirable	
EDM	Desirable	
EBU Core	Mandatory (for broadcasting related CoPs) / Recommended	
MPEG-7	Recommended (for CoPs using automatic content analysis)	
LIDO	Mandatory (for museum/gallery related) / Recommended	
EAD	Mandatory (for non-a/v archive related) / Recommended	
Metadata output formats		Support for a metadata format as target format of the mapping process
Dublin Core	Mandatory	
ESE	Recommended	
EDM	Recommended	
EBU Core	Mandatory (for broadcasting related CoPs) / Recommended	
MPEG-7	Recommended (for CoPs using automatic content analysis)	
LIDO	Mandatory (for museum/gallery related) / Recommended	

EAD	Mandatory (for non-a/v archive related) / Recommended	
option to add custom formats	Recommended	Support for adding new metadata formats
XML representation support	Mandatory	Support for metadata documents in XML format
RDF representation support	Recommended	Support for metadata documents in RDF format
Metadata model constructs		
single -> multiple elements	Recommended	Support mapping a single element into a set of elements (e.g., string into structured)
multiple -> single elements	Mandatory	Support mapping a set of elements into a single elements (e.g., structured into string)
structure using context elements	Mandatory	Define mapping of content structure constructs using contextual elements
conditional mapping based on element/attribute values	Mandatory	Define mapping rules that are conditioned on values of elements or attributes
map collections	Recommended	Support for mapping of collections of metadata records rather than single metadata records only
merge string values	Mandatory	Support merging values of separate string values into one string value
split string values	Recommended	Support splitting a string value into a set of separate string values
number of levels in data structure	Mandatory: 2 / Recommended: 2+	Support for number of hierarchy levels in the document structure
start from example(s)	Recommended	Support initiating mapping from example documents
start from schema	Recommended	Support initiating mapping from a schema instance
configuration user interface	Mandatory	Provision of a configuration user interface (instead of/in addition to configuration files)
user interface		
drag & drop mappings	Recommended	Support of drag&drop for configuring the mappings
preview	Mandatory	Provide preview of configured mappings
map constructs not found in available examples	Recommended	Support the definition of mappings for constructs not found in one of the examples

Table 5: List of functions for metadata mapping and validation

2.2.2 Measurement plan

Here below, for metadata mapping and validation RO, the measurement plan proposed in Section 3.1.1 has been customized in some characteristics and sub characteristics by their measures. To this aim, peculiarities and specific features have been taken in consideration. In particular some metadata/vocabulary mapping tools are automatic. However, they have a user interface for configuration of the mapping, thus the UI criteria can be applied to it. Vocabulary mapping tools follow similar workflows, and only functional criteria differ significantly.

For general information on the measurements see Section 2 .

2.2.2.1 Functional suitability

Degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions.

Measurement function: FS=(X+Y+Z) / 3(where X,Y,Z are the scores computed as in the following

FS = Functional Suitability

X = Functional Completeness

Y = Functional Correctness

Z = Functional Appropriateness

In particular the ability to reach a defined goal using the tool can be considered

Interpretation of test results: FS value closer to 1 is better

Functional Completeness:

Measure: functional metadata formats coverage

Description of measure: Evaluation of the metadata formats supported and the constructs of data model supported. Measured by comparing against metadata formats/models used by CoPs.

Note: Currently we have very little information about the metadata formats/models/vocabularies used by some CoPs.

Functional Correctness:

Measure: functional correctness of mapping

Description of the measure: How much the required specific accuracy of mapping are complied with?

Note: The evaluation of the mapping can be performed and validated by an expert.

Functional Appropriateness:

Measure: functional appropriateness of the mapping tools

Note: the evaluation can take in consideration the set of mapping tools in specific steps of a workflow (ingest, B2B exchange, export to European) and can be validated by validated by an expert.

2.2.2.2 Performance efficiency

Performance relative to the amount of resources used under stated conditions.

Measurement function: PE= (X+Y+Z) / 3 where

PE = Performance Efficiency
X = Time behavior
Y = Resource utilization
Z = Capacity
For calculating the X, Y and Z scores, see section 2.1.2

Interpretation of test results: PE smaller is better

Time behaviour:

Possible measures: the processing time of mapping, or the response time of configuration application

Description of the measure: Measures are the processing time of mapping, and response time of configuration application, expressed as time wrt. number of elements/number of mapping rules in the input document and absolute response time

1. Measurement function (example for response time):

   X = (B – A) /C

   A = time of entering a command/invoking processing
   B = time of receiving the first response/completing of processing
   C = time criteria specifying maximum allowable waiting duration from entering request to receiving response

2. Interpretation of test results: X (mean value of X) varies from 0 to infinite. Usually, smaller is better.

Resource utilization:

Degree to which the amounts and types of resources used by a product or system when performing its functions meet requirements

Here below the CPU utilization is presented as an example.

1. Measure: (Mean) CPU/RAM utilization
2. Description of the measure: How much CPU time/RAM is used to perform a given task
3. Measurement function:

   Y = fraction of CPU/RAM actually used to perform a task on a reference system

4. Interpretation of test results: Y smaller is better.

Capacity:

Degree to which the maximum limits of a product or system parameter meet requirements

Measure: Maximum throughput using a specific reference configuration

1. Measurement function:

   Z = A/B where

   A = operation time
   B = the total no. of processed documents

2. Interpretation of test results: Z smaller is better.

2.2.2.3 Compatibility

Degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment.

Measurement function: Co = Y where

Co = Compatibility
Y = Interoperability
For calculating the Y score, see section 2.1.3

Interpretation of test results: Co value larger is better

Interoperability:
Degree to which two or more systems, products or components can exchange information and use the information that has been exchanged

1. Measure: Supported service interfaces for information exchange

   Note: For some CoPs there are no tools/systems at hand to actually test interoperability of service interfaces and output formats.

2. Description of the measure: the service interfaces are smoothly exchanged with other software or systems

3. Measurement function:

   Y = A / B where

   A = number of interfaces for information exchange
B = total number of interfaces to be supported

4. Interpretation of test results: Y varies from 0 to infinite. Usually, larger is better.

2.2.2.4 Usability

Degree to which a product or system can be used by specified users to achieve specified goals with eff ectiveness, efficiency and satisfaction in a specified context of use.

Measurement function: Us = (H+J+K+L+M+N) / 6 where

Us = Usability
H = Appropriateness Recognizability
J = Learnability
In particular the time to being able to use the user interface can be considered
K = Operability
L = User error protection
M = User interface aesthetics
N = Accessibility
For calculating the H, J, K, L, M and N scores, see section 2.1.4

Interpretation of test results: Us value closer to 1 is better

Note: It will not be possible to do actual user tests in this initial assessment.

Accessibility:

Note In particular the adaptability of text sizes, alternative image texts, alternative use of input devices can be considered

2.2.2.5 Reliability

Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.

Measurement function: Re = (K+L) / 2 where

Re = Reliability
K = Fault Tolerance
L = Recoverability
For calculating the K and L scores, see section 2.1.5

Interpretation of test results: Re value closer to 1 is better

Recoverability:

Note: In particular the possibility of using the tool after incorrect input files, incomplete mapping projects, user errors should be considered

2.2.2.6 Maintainability

Degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers.

Measurement function: Ma = (H+J) / 2 where

Ma = Maintainability
H = Modularity
J = Reusability
For calculating the H and J scores, see section 2.1.7

Interpretation of test results: Ma value closer to 1 is better

Modularity:

1. Possible Measure: Easiness of adding new formats/vocabularies
2. Measurement function:

   H = assessment of effort for adding a new mapping
3. Interpretation of test results: H varies from 0 to 1. Usually, closer to 1 is better.

Reusability:

Degree to which an asset can be used in more than one system, or in building other assets

1. Possible Measure: Service interfaces
2. Description of the measure: Supported service interfaces?
3. Measurement function:

J = A / B

A = number of service interfaces supported

B = total number of service interfaces considered

Interpretation of test results: J varies from 0 to 1. Usually, closer to 1 is better.

2.3 Storage

In this section a measurement plan for Storage RO category is refined as a more complete example of assessment. In particular, implementations of *Time behaviour* (sub characteristic of *Performance Efficiency*) relative to “list file and folder” function, and *Integrity* (sub characteristic of *Security*) relative to "store file" and "compute fixity checksums and verify checksum matching" functions, are provided.

2.3.1 Definition of functions

The functions typically expected to be provided by storage tools, services, or systems are listed in Table 6, associated to a level of need defined as follows:

- Mandatory - Must have
- Recommended - Could deal also without, but it would be better to have
- Desirable - May be appreciated in some cases, but in most cases it doesn’t make the difference

Function	**Level of need**	**Description**
M1- store File	mandatory	the function of receiving the file in input for writing on storage and hold it persistently (in a non volatile way). The stored File must be exactly the same (at bit level) than the one in input. The function must return to the client/producer the information which will be required for requesting a subsequent restore.
M2 - restore File	mandatory	the function of copying the File from its storage location and make a perfect copy of it on an external location suitable to client/consumer. The restored File must be exactly the same (at bit level) than the one on Storage.
R1 - store Folder	recommended	the function of executing “store File” function recursively for a Folder, including the whole Folder hierarchy which must be reflected on the Storage
R2 - restore Folder	recommended	the function of executing “restore File” function recursively for a stored Folder including the whole Folder hierarchy. In order to be complete the function must provide: - possibility to restore a sub-Folder of the originally stored Folder - possibility to restore a single File from the originally stored Folder hierarchy.
R3 - list Files and Folders	recommended	the function of returning the list of stored Files and Folders, with all the available details, if requested.
R4 - provide File Access	recommended	the function of allowing a third component to access File for reading, including seek to any point of file, without need to restore it in advance
R5 - compute fixity checksums and verify checksum matching	recommended	the function of processing File with checksum algorithms and verifying the checksum matching against expected values.
R6 - spread copies of File/Folder	recommended	the function of spreading automatically (according to configuration) multiple copies of the same file (or folder) in locations not sharing the same risks of lost or corruption.
R7 - recover copies of	recommended	the function of re-establishing automatically the desired number of

File/Folder		independent file copies when a corrupted copy is detected.
R8 - Configurable ingest	recommended	possibility for client/producer to select options to store File / Folder operation
D1 - record stream	desirable	the function of storing a media-File as the result of recording from a continuous real-time stream (instead of from a source File). This function acts with a best effort paradigm and might be adopted only when necessary (e.g. the stream is the only available source) In order to be complete the function must implement start and stop at input signals.
D2 - Output Streaming	desirable	the function of playing a media-File and generate an output stream. This functions acts with a best effort paradigm and should not be used for creating exact copies at bit level. In order to be complete the function must implement start, stop, pause, resume at input signals. Note: how to consider other sub-functions, such as: (1) playing a client-defined time-fragment, (2) supporting adaptive use of available bandwidth, and other ..?
D3- File partial restore	desirable / recommended	the function of restoring a part of a media File, resulting in a new media File with content extracted, with no change, from the original File and identified in term of media fragment, e.g. a time interval on the media playback timeline.
D4 - File/Folder tagging	desirable	Capability to associate to a file or a folder some tags i.e. peace of metadata useful to characterize and group resources. These tags will be useful for searching and getting additional information on a file/folder. Need more thinking
D5 - File search	desirable	Ability to search for a file not only by its ID but also using any tags or metadata previously associated. Related to line above
D6 - Format migration / Transwrapping / Transcoding	desirable	Ability to transwrap/transcode essence file (e.g. from MOV to MXF)
D7 - Produce and manage automatic workflow	desirable	Ability to configure and manage automatic workflow for example periodic sanity check, massive format migration etc.
D8 - Metadata extraction	desirable	?what about thumbnails, or exif information? or MIME type?
D9 - Preview web based access	desirable	Check if it deals with usability! Def: to provide a kind of preview (browsing copy/ icon / excerpt) quickly available to web based user interface. May imply a two (at least) levels storage.

Table 6: List of functions for Storage tools

2.3.2 Measurement plan

2.3.2.1 Functional Suitability

Degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions.

Measurement function: FS = (X+Z) / 2 where

FS = **Functional Suitability**
X = *Functional Completeness*
Z = *Functional Appropriateness*

The list of storage functions considered during the assessment is presented in section 3.3.1.

For calculating the X and Z scores, see section 2.1.1

Interpretation of test results: FS value closer to 1 is better

2.3.2.2 Performance efficiency

Considering the function “list Files and Folders” and its definition “the function of returning the list of stored Files and Folders, with all the available details, if requested” provided in table 1 in section 3.3.1 in the following, an example of refinement of X sub characteristic measurement (**Time Behaviour**) is provided.
In particular, time behaviour is meant as the degree to which the response and processing times and throughput rates of a product or system, when performing its functions, meet requirements. In the following, *response time* is adopted as possible measure for **time behaviour** and an example of application is provided.
If *response time* is meant as *duration from giving a command to start a batch of tasks till receiving the first response*, then the Measurement function of X is:

X = (B – A) /C

A = time of entering a command to list stored files and folders
B = time of receiving the first response
C = time criteria specifying maximum allowable waiting duration from entering request to receiving response according to the software requirements

X value smaller is better.

Performance relative to the amount of resources used under stated conditions.

Measurement function: PE= (X+Y+Z) / 3 where

PE = **Performance Efficiency**

X = *Time behaviour*

Y = *Resource utilization*

Z = *Capacity*

For calculating the X, Y and Z scores, see section 2.1.2

Interpretation of test results: PE smaller is better

Considering the function “list Files and Folders” and its definition “the function of returning the list of stored Files and Folders, with all the available details, if requested” provided in table 1 in section 3.3.1 in the following, an example of refinement of X sub characteristic measurement (**Time Behaviour**) is provided.

In particular, time behaviour is meant as the degree to which the response and processing times and throughput rates of a product or system, when performing its functions, meet

requirements. In the following, *response time* is adopted as possible measure for **time behaviour** and an example of application is provided.

If *response time* is meant as *duration from giving a command to start a batch of tasks till receiving the first response*, then the Measurement function of X is:

X = (B – A) /C

A = time of entering a command to list stored files and folders

B = time of receiving the first response

C = time criteria specifying maximum allowable waiting duration from entering request to receiving response according to the software requirements

X value smaller is better.

2.3.2.3 Compatibility

Degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment.

Measurement function: Co= (X+Y) / 2 where
Co = **Compatibility**
X = *Co-existence*
Y = *Interoperability*

Interpretation of test results: Co value larger is better

For calculating the X and Y scores, the general criteria of section are modified for the Storage RO category as in the following:

Co-existence

Degree to which the storage RO can share its working environment, i.e. the system(s) on which it is deployed, with other products.

The measurement is based on the identification of a number of products or tools or services which are relevant for deployment in the same environment of the RO under assessment.

Having defined sets A and B as:

B={b|b product or tool expected to be co-deployed with RO}

A={a|a∈B and can be co-deployed successfully}

we apply the following formula:

X = #(A) / #(B) , bounded to 1, higher is better, where:

#(A) and #(B) are the number of elements of sets A and B respectively

According to the given definition it is worth to mention that A⊆B. Note also that the elements of B can vary according to specific RO installation environment (e.g. the operative system).

The reference list of elements of set B, for the generic storage RO is given below. For a particular RO, there might be reasons for removing one or more elements.

b1	preservation platform (eg. P4 or archivematica)
b2	HTTP/HTTPS service for different content
b3	SSH service
b4	file system sharing services (e.g. CIFS)
b5	database service with other databases
b6	other user of hardware devices normally used by the RO (e.g. LTO Library or drive)
b7	other processing services running (e.g. transcoding) on host
b8	other user accounts on host

Table 7: Co-existence table for storage RO

Interoperability

Degree to which the clients, either human user or application, of the storage RO can exchange and use information to/from the storage RO.

The measurement is based on the identification of a number of protocols and formats which are relevant to information exchange to/from the storage RO, either as exchange of storage payload (files, carriers) or as exchange of ancillary information (messages, logs, operation results).

Having defined sets A and B as:

B={b|b is an exchange format expected to be supported by the RO}

A={a|a∈B and is provided by the RO}

we apply the following formula:

X = #(A) / #(B) , bounded to 1, higher is better, where:

#(A) and #(B) are the number of elements of sets A and B respectively.

According to the given definition it is worth to mention that A⊆B.

The reference list of elements of set B, for the generic storage RO is given below. For a particular RO, there might be reasons for removing one or more elements.

b1	modality for making requests to storage RO by human operator	interoperable if web-based GUI working on most commonly used web browsers
b2	modality for making requests to storage RO by workflow or automated	interoperable if based on defined and documented APIs for services , to be used

	system	over HTTP/HTTPS protocols (e.g. WDSL, REST, CGI)
b3	format of responses from storage RO	interoperable if returned responses are compliant to a defined model (e.g. XML Schema) and according to a standard encoding (e.g. XML or JSON)
b4	format of removable storage item for exchange AS IS, i.e. not as a result of export/restore operation (e.g. exchange LTO tape)	interoperable if exchanged storage item is open format (e.g. LTFS or tar in the case of LTO tapes)

Table 8: Interoperability table for storage RO

2.3.2.4 Usability

Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

Measurement function: Us = (K+L) / 2 where

Us = **Usability**
K = *Operability*
L = *User error protection*

For calculating the K and L scores, see section 2.1.4

Interpretation of test results: Us value closer to 1 is better

2.3.2.5 Reliability

Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.

Measurement function: Re = (H+J+K+L) / 4 where

Re = **Reliability**
H = *Maturity*
J = *Availability*
K = *Fault Tolerance*
L = *Recoverability*

For calculating the H, J, K and L scores, see section 2.1.5 .

Interpretation of test results: Re value closer to 1 is better

2.3.2.6 Security

Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization

Measurement function: Se = (H+J+K+L+M) /5 where

Se = **Security**
H = *Confidentiality*
J = *Integrity*
K = *Non-repudiation*
L = *Accountability*
M = *Authenticity*

For calculating the H, J, K, L and M scores, see section 2.1.6 .

Interpretation of test results: Se value closer to 1 is better

Considering the definition of “store files” function provided in table 1 in section 3.3.1 according to which “The stored File must be exactly the same (at bit level) than the one in input”, and the definition of “compute fixity checksums and verify checksum matching” function, according to which files are processed with checksum algorithms and the checksum verified against expected values, in the following, an example of refinement of J sub characteristic measurement (**integrity**) is provided.

In particular, integrity check is meant as *ability to check that the files have not been corrupted; this task is carried out typically using checksum comparisons*.

It can be evaluated in three different situations:

J.1 Integrity check: *upload*
Calculate checksum when uploading and compare against a value given by the user.
J.2 Integrity check: *download*
Calculate checksum when downloading and compare against a pre-calculated value.
J.3 Integrity check: *periodic background*
Periodically (e.g. one per week) calculate checksum and compare against pre-calculated values.

If all the three situations are contemporary measured, the computation of J is:

J= (J.1+J.2+J.3)/3 where

J.1=1 - (A/B) where
A= Number of data corruption instances actually occurring during upload verified by means checksum values
B= Total number of uploads in a unit of time

J.2= 1 - (C/D)
C= Number of data corruption instances actually occurring during download verified by means checksum values
D= Total number of download in a unit of time

J.3= 1 - (E/F)
E= Number of data corruption instances actually occurring during periodic background verified by means checksum values
F= Total number of periodic background

If the three situations are not contemporary measured, the computation of J is different according to the observed situation.

2.3.2.7 Maintainability

Degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers.

Measurement function: Ma = (H+K+L+M) / 4 where

Ma = **Maintainability**
H = *Modularity*
K = Analysability
L = Modifiability
M = *Testability*

For calculating the H, K, L and M scores, see section 2.1.7

Interpretation of test results: Ma value closer to 1 is better

2.3.2.8 Portability

Degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another.

Measurement function: Po = (X+Y+Z) / 3 where

Po = **Portability**
X = *Adaptability*
Y = *Installability*
Z = *Replaceability*

For calculating the X, Y and Z scores, see section 2.1.8

Interpretation of test results: Po value closer to 1 is better

2.4 Quality Assessment

In this section a measurement plan for quality assessment RO category is refined as a more complete example of assessment. Firstly, we start by defining the functions required to be tested followed by a measurement plan on specific functions which need to be specialised (specialisation from the generalised criteria mentioned in the section above) for this particular category. The levels of need are classified as follows:

- Mandatory - Must have
- Recommended - Could deal also without, but it would be better to have
- Desirable - May be appreciated in some cases, but in most cases it doesn't make the difference

2.4.1 Definition of functions

Functionality	Level of need	Description
Automatic Defect Analysis Functions (categorisation based on EBU QC checks[33])		
Analogue Synchronisation Errors Aliases: lost lock, time-base corrector (TBC) hit, video breakup, lost video sync, horizontal distortion	Mandatory	System shall check for analogue synchronisation problems that have caused severe visual line/field/frame distortions. Analogue synchronisation problems can lead to visible artefacts. These can be created during the analogue tape read process (e.g. during tape digitisation), as part of the analogue video transmission process or as a side-effect of improper analogue video editing. It manifests itself in visual line/field/frame/multi-frame distortions of varying degree with typically a horizontal/line oriented appearance and a temporal extent of one or more fields/frames. Problems with vertical synchronization usually result in rolling (up or down) frames. As a severity measure the relation between the distorted area and the area of the entire frame is used.
Coloured Frames Aliases: Black Frames, Monochrome Frames, Uniform Color Frames	Mandatory	System shall detect frames which have no active video and point out full-sized single coloured frames. Coloured frames may be produced by video tape players (during migration) or by software errors in the production cycle.

[33] EBU Strategic Programme on QC (EBU QC) http://tech.ebu.ch/groups/qc, First draft release of QC test definitions available at http://tech.ebu.ch/docs/tech/tech3363.zip

Digital Tape Dropouts Aliases: digital video tape dropout, digital hits, digital tape hits	Mandatory	System shall detect visible artefacts caused by digital tape errors. The result may include tape/error type and severity together with spatial locations. This is about visible artefacts which occur within the digital tape read process and manifests itself when head problems or tape overuse cause the error correction of the VTR to create short term failures of parts of frames. The visual effect is the appearance of impairments, such as alternating lines in a block, duplicated block areas, arrays of similar pixels within a block area, and random portions of blocks with changed luminance or chrominance within one or multiple consecutive frames. The appearance of those blocks as well as the spatiotemporal pattern of those blocks strongly depends on the kind of tape, such as DigiBeta, IMX, DV.... Most relevant in the context of archive migration applications are early digital tape formats, e.g. DigiBeta.
Video Noise Aliases: image noise, noise	Desirable	System shall detect video segments whose essence shows a noise level that is above a user-defined threshold. The visual noise level might be estimated by a signal to noise ratio (SNR). Noise constitutes an unwanted signal that inevitably adds to the useful part, it may originate from different sources, e.g. electronic sensor noise, quantisation noise, film grain noise…. For archive applications the knowledge on the noise level is relevant to estimate restoration costs for content re-used (e.g. in a program, DVD, BD...)
Blurriness Aliases: out of focus, blur detection, sharpness	Recommended	System shall detect video segments whose image content would be perceived as blurry by the viewer. For archive applications the knowledge on content blurriness is relevant to decide if it can be re-used for a certain purpose (e.g. is SD content sharp enough to be re-used for an HD program, BD,....)
Video Test Pattern Aliases: test card, colour bars	Mandatory	System shall detect video segments containing specific test pattern content. A test pattern is a sequence of (often still) images with showing particular characteristics. For video experts, test patterns allow to quickly detect problems in a generic video chain and facilitate calibration, alignment, and matching of video devices. In typical broadcaster workflows, test patterns often have to be cut off or checked for a specific position and duration(e.g. at the beginning and end of a programme). For archive applications test pattern segments shall be detected after the migration of content /programs, especially on multi-program tapes. Usually no test pattern segments shall be present in a program file.
Video Field Order Aliases: field order, field dominance	Recommended	System shall detect video segments containing a field order differing from an desired one.

Audio Silence Aliases: mute test, minimum level	Mandatory	System shall check if the audio level is lower than a user defined threshold value. In archive migration applications the actual audio channel usage can be assessed by audio silence detection. The actual audio channel usage in the video needs to correspond with the audio channel usage described in a content/asset management system.
Audio Encoding Format Change	Recommended	System shall check if the audio encoding is changing within a channel of a program, e.g. from PCM to Dolby-E. The actual audio encoding used in the video for the individual channels needs to correspond with the audio encoding format described for these channels within a content/asset management system.
General Analysis Properties		
Analysis profiles	Mandatory	Capability to adapt quality analysis functions (detectors) and its parameters to the desired QC task/use case
No reference video required	Mandatory	For content within archives stored or to be migrated very often only one copy do exist. The capability to assess the audiovisual quality without any other copy required is therefore crucial
Detection of multi-generation defects	Recommended or Mandatory	Defects within content of AV archives may have been copied/migrated from one earlier format to the next. These defects (e.g. analogue synchronisation errors or analogue tape dropouts) are visible within the current copy on a certain media format (e.g. DigiBeta) but are not originated from the current format or encoding. An AV quality assessment system for AV archive assessment or migration shall be able to work independently from the current media format and encoding
Multi-Resolution support	Mandatory	Capability to process content with different nominal resolution, e.g. SD, HD, 2k. Practically any archive holds content with different nominal resolution
Interactive Validation/Verification Functions		
Check file efficiently for right content	Mandatory	Check efficiently that file contains correct content (potentially described in an content/asset management system) and that correct tape segment has been digitised, e.g. containing no test patterns pre-recorded on multi-program tapes.

Human validation of automatic analysis functions	Mandatoy	The system shall support the human verification of detections from an automatic analysis step. This way verified reports can be generated.
Interactive defect annotation support	Mandatory	Defects can be manually created and modified (time and duration) by a human operator. Detections missed by the automatic analysis can be annotated this way.
Overall quality rating support	Mandatory	The system shall support to give an overall quality rating for the entire content/program (e.g. OK, Error...).
Defect severity based operation/validation	Recommended	The system should support efficient verification by prioritizing the most relevant annotations
Video output devices	Desirable	The system should support the output of videos on the following devices: - Desktop within a single screen - Desktop on a second full screen - SDI
Interlaced video output	Recommended	The system supports to output interlaced video on capable devices

Table 9: List of functions for quality assessment ROs

2.4.2 Measurement plan

Here below, for Quality Assessment RO, the measurement plan proposed in Section has been customized in some characteristics and sub characteristics by their measures. To this aim, peculiarities and specific features have been taken in consideration.
In particular, we consider automatic and semi-automatic tools, i.e. some of the tools have a user interface, while others are automatic services. Thus, some criteria (most notably the usability related ones), apply on to tools/components with a UI.
For general information on the measurements see Section 2.1 .

2.4.2.1 Functional suitability

Degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions

1. Measurement function: FS=(X+Y+Z) / 2(where X,Y are the scores computed as in the following

   FS = Functional Suitability
   X = Functional Completeness
   Y = Functional Correctness

2. Interpretation of test results: FS value closer to 1 is better

2.4.2.2 Performance efficiency

Performance relative to the amount of resources used under stated conditions.

1. Measurement function: PE= (X+Y+Z) / 3 where X,Y,Z are the scores computed as in the following

   PE = Performance Efficiency
   X = *Time behaviour*
   Y = *Resource utilization*
   Z = *Capacity*

2. Interpretation of test results: PE smaller is better

Time behaviour:

Degree to which the response and processing times and throughput rates of a product or system, when performing its functions, meet requirements.

1. Possible Measure: *Frame Accurate Player* or *Responsive User Interface* or *Throughput of automatic tools*

2. Description of the measures: Respectively, for Frame Accurate Player, the system shall be able to navigate to any time code within a video in a frame accurate way. Also single frame steps forward/backward shall be supported. For Responsive User Interface, the response time of entire validation application shall be immediately responsive to user actions, for throughput of automatic tools, the system shall run in approximately realtime for SD material. State also the runtime for HD material

Resource utilization:

Degree to which the amounts and types of resources used by a product or system when performing its functions meet requirements

1. Possible Measures: *Efficient RAM usage* or *Efficient GPU usage* or *Efficient CPU usage* or *Efficient Network usage*.

2. Description of the measures: Respectively, for *Efficient RAM usage*, RAM utilization for processing SD and HD material should not exceed RAM available in state-of-the-art machines (8 - 12 GB). For *Efficient GPU usage,* the GPU (if used) shall be used for algorithms where appropriate. For *Efficient CPU usage,* the system shall be able to utilize all CPU cores for analysing SD and HD material. For *Efficient Network usage,* the system shall not transfer content or parts of the content multiple times over network during analysis

Capacity:

Degree to which the maximum limits of a product or system parameter meet requirements

1. Possible Measure: Capability to scale throughput

2. Description of the measure: The system shall be able to scale throughput by the following methods:

- Activate/Deactivate detectors based on the customers needs
- Configure and optimize parameters on customers contents
- Scale with additional hardware resources like more CPU cores
- Scale with the ability to distribute analysis on multiple machines on a file basis

2.4.2.3 Compatibility

Degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment.

1. Measurement function: Co=Y is computed as in the following

   Co = Compatibility
   Y = *Interoperability*

2. Interpretation of test results: Co value larger is better

Interoperability:
Degree to which two or more systems, products or components can exchange information and use the information that has been exchanged

Note: possible measures are presented below

Supported video input formats and metadata output format, supported service interfaces	Probably no tools/systems at hand to test interoperability of service interfaces and standard compliance of input video data.
Supported video container formats	The following container formats should be supported: - MPEG TS and PS - MXF - MP4 - MOV - AVI to be extended by CoP input
Supported commonly used video encoding formats	The following video formats should be supported: - MPEG-2 (incl. IMX50 and XDCAM HD) (e.g. P4U RAI in MXF) - MPEG-4 AVC (H.264) (e.g. P4U UIBK) - JPEG2000 (SAMMA format) - DV and DVCPro to be extended by CoP input

Supported additional video encoding formats	The following video formats could be supported: - Uncompressed 10bit in MOV (e.g. P4U Tate) - Uncompressed 10bit in MXF (e.g. BBC) - ProRes - DCP, MAP - WMV to be extended by CoP input
Supported audio encoding formats	The following audio formats should be supported: - PCM - MPEG-1 Audio, MPEG-1 Layer 3, MPEG-2 Audio - AAC - AC3 - WMA to be extended by CoP input
Standardized metadata output format	The output metadata format shall conform to an international metadata standard
Workflow integration support (Web service, drop folder)	The system shall support following interfaces for workflow integration: - Drop folder - Web services (REST or SOAP)

Table 10: Possible measures

Here below the supported service interfaces for information exchange is presented as an example.

1. Measure: Supported service interfaces for information exchange

2. Description of the measure: Probably no tools/systems at hand to test interoperability of service interfaces and standard compliance of input video data.
3. Measurement function:

   Y = A / B where

   A = number of interfaces for information exchange
   B = total number of interfaces to be supported

4. Interpretation of test results: Y varies from 0 to infinite. Usually, larger is better.

2.4.2.4 Usability

Degree to which a product or system can be used by specified users to achieve specified goals with eff ectiveness, efficiency and satisfaction in a specified context of use.
Note: It will not be possible to do actual user tests/study in this initial assessment (lacking users and controlled test environment).

1. Measurement function: Us = (H+J+K+L+M+N) / 6 where H, J, K, L, M, N are the scores computed as in the following

   Us = Usability
   H = *Appropriateness Recognizability*
   J = *Learnability*
   K = *Operability*
   L = *User error protection*
   M = *User interface aesthetics*
   N = *Accessibility*

2. Interpretation of test results: Us value closer to 1 is better

Appropriateness Recognizability:
Degree to which users can recognize whether a product or system is appropriate for their needs.

Note: When using the AV quality assessment tool the first time; how fast a user is able to understand which functions are available in the interface and what they are intended for; just by looking to the interface

Learnability:
Degree to which a product or system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use.

Note: How fast a user is able to learn/understand, how the user interface functions of the AV quality assessment tool work in detail (to learn to use them). Potential functions to be learned are listed in the section Functional completeness (Interactive validation/verification functions)

Operability:
Degree to which a product or system has attributes that make it easy to operate and control**.**

Note: Potential goals to be reached are listed in the section Functional Completeness (Interactive validation/verification functions)

Accessibility:
Degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use

Note: This measure is not generally applicable, a user performing audio/visual quality verification cannot have vision/hearing impairments. Certain aspects can be evaluated, e.g. is it possible to use key board alternatively to the mouse, are specific interaction functions common in the media domain supported (e.g. use space bar to start and pause the video player), can the viewing size be adapted

2.4.2.5 Reliability

Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.

1. Measurement function: Re = (K+L) / 2 where K, L are the scores computed as in the following

Re = Reliability
K = *Fault Tolerance*
L = *Recoverability*

2. Interpretation of test results: Re value closer to 1 is better

Fault tolerance:

Degree to which a system, product or component operates as intended despite the presence of hardware or software faults.

1. Possible measures: *nu. of Incorrect file input* or *nu. of User operation errors* or *Tolerance against analysis failures*
2. Description of the measures: Respectively, for *Incorrect file input*, the system shall be tolerant against unsupported file formats or missing files. For *User operation errors*, the system shall be tolerant against the following user operation errors: in an interactive verification application, the user should be able to undo human operations, in an interactive verification application, the user should not be able to close the application without being reminded of unsaved changes. Finally, for *tolerance against analysis failures*, the system interfaces shall be tolerant against analysis failures. If for example an analysis error occurs (e.g. due to network interruption), the analysis web service and user interface should still be operable, and capable of processing further video files.

Recoverability:

Degree to which, in the event of an interruption or a failure, a product or system can recover the data directly affected and re-establish the desired state of the system

Note:

- Recoverability of Jobs: If the system was unexpectedly terminated by external circumstances it shall be able to recover file analysis jobs and shall be able to reanalyse them.
- Documentation of user decisions: User decisions shall be logged for tracing back the decision making processes. If supported, operator information shall also be stored in the metadata document.

2.4.2.6 Security

Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization

Measurement function: Se = L where

Se = **Security**
L = *Accountability*

Interpretation of test results: Se value closer to 1 is better

2.4.2.7 Maintainability

Degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers.

1. Measurement function: Ma = (H+J+M) / 3 where H, J, K, M are the scores computed as in the following

   Ma = Maintainability
   H = *Modularity*
   J = *Reusability*
   M = *Testability*

2. Interpretation of test results: Ma value closer to 1 is better

Modularity:

Degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components

Note: Analysis functional extension: Capability to extend Analysis Functions in a modular way. This is useful when use case or customer specific QC analysis functions needs to be integrated in and fast and easy way.

Modular I/O components: I/O components shall be modular. This brings the benefit that only those required by the customer need to be licensed and that new formats required by a customer can be easily extended

Capability to extend visualisation and interaction functions in a modular way: The user interface shall be modular, so that visualization can be customized and extended to the customers' needs

Reusability:

Degree to which an asset can be used in more than one system, or in building other assets

Note: Applicability in other application domains. Specify, in what application domains (additional to AV preservation/migration) the tool is applicable:

- QC in AV content production/ post-production

- QC in AV content delivery

- QC for web-video portals

Testability:

Degree of effectiveness and efficiency with which test criteria can be established for a system, product or component and tests can be performed to determine whether those criteria have been met

Note: Ground truth for regression tests. Is ground truth data available for the detectors, so that the results can be benchmarked against this ground truth

2.4.2.8 Portability

Degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another.

1. Measurement function: Po = Y where Y computed as in the following

Po = Portability

Y = *Installability*

2. Interpretation of test results: Po value closer to 1 is better

Installability:

Degree of effectiveness and efficiency with which a product or system can be successfully installed and/or uninstalled in a specified environment

Note: Installation tools available: Automatic installation tools (wizards) should be available. Uninstalling the tools should be possible

Documentation available: Interface (UI, services) documentation should be available.

Preservation platforms and systems

2.5 Preservation platforms and systems

In this Section we provide a measurement plan for the RO category “Preservation platforms and systems”.

In the following we refine the measurement plan reported at the beginning and taken from D3.1. The identified functions and the measurement plan will be improved for the second year assessment.

2.5.1 Definition of functions

We assume that the platform is compliant to OAIS model and that its architecture is driven by OAIS functional entities. For each function we provide a short description and also the associated OAIS functional entities. Each function could affect more than one OAIS entity, we simply highlighted the parts of the OAIS model which are more strongly related.
For each function we provide a level (mandatory, recommended or desirable). The list below is not complete because evaluating a platform is a complex activity and several criteria can be defined. We decided, for the first year evaluation, to stress high level aspects which provide a clear understanding of the level or maturity (the TRL) of the solution, disregarding those which require more extensive tests using the Presto4U dataset, such as the performance.

Function	OAIS functional entity	Level of requirement	Description
M1 - GUI ingestion	Ingest	Mandatory	Ingestion using guided procedure offered by the GUI
M2 - Preservation of original content properties	Ingest	Mandatory	The original file received by the producer is stored in the archive
M3 - Support for AV formats	Ingest, Data Management	Mandatory	Support for AV formats selected for the Presto4U dataset
M4 - Preservation Workflows Management	Ingest, Preservation Planning	Mandatory	The platform implements workflows including tasks for content curation
M5 - Export of DIP	Access	Mandatory	Allow creation of Dissemination Packages for access
M6 - Periodic integrity checks of the material and storing information in the AIP	Preservation Planning	Mandatory	Periodic checks for file corruption (related also to availability of multiple copies for restore)
M7 - Format migration	Data Management	Mandatory	when format is at risk of obsolescence (a few tools working on it)
M8 - Ability to deal with large files	Archival Storage	Mandatory	Integrate storage technologies suitable even for huge files, for example larger than 10 GB (=20 min of MXF/D10)
M9 - Content quality control	Data Management	Mandatory	Integrate tools for QC
M10 - Virus check	Ingest, Data Management	Mandatory	Integrate tools for virus check of ingested content
R1 - Batch ingestion	Ingest	Recommended	Capability to ingest list of SIP files from CLI, managing ingestion queue
R2 - Support for METS	Ingest, Data Management, Access	Recommended	METS is used as a wrapper for SIP, AIP, DIP
R3- Support for PREMIS	Data Management	Recommended	PREMIS is used for preservation metadata and for logging preservation events
R4 - definition of requirements for restitution/playback	Access	Recommended	support reconstruction of the desired characteristics of the playback environment
R5 - Extension with Add-ons and plugins	All	Recommended	Integration of tools and services for specific purposes
R6 - Usage Documentation	All	Recommended	For archive administrators

R5 - Dashboard for job monitoring	All	Recommended	Provide real time information about active jobs (e.g. ingestion queue, periodic preservation tasks, …), including used resources and status
R6- Automatic extraction of technical metadata	Data Management	Recommended	Extraction of technical metadata during ingestion
R7 - User profiles and ACL	Administration	Recommended	Manage user authentication and authorization, enable functionalities in the GUI according to permissions, etc
R8 - Creation of proxy copies (browsing quality)	Access	Recommended	Creation of low quality copy
R9- Multiple copies for redundancy	Ingest, Preservation Planning	Recommended	Ability to create device independent AIPs to ensure future access
D1 - Customize existing workflows	Ingest, Preservation Planning	Desirable	Allow configuration and customization of existing preservation tasks
D2 – Export of DIP to different formats	Access	Desirable	transcoding to format on Consumer's request
D3 - Export of AV content fragments	Access	Desirable	Partial restore
D4 – Ability to integrate with alternative collection management systems	Archival Storage	Desirable	Possibility to integrate with an alternative system providing functions different from preservation (e.g. cataloguing and searching)
D5 – Populate and draw data and statistics from collection management systems	Administration, Access	Desirable	Provide information about use of resources, number of accessed contents, etc

Table 11: List of functions for preservation platforms

2.5.2 Measurement plan

2.5.2.1 Functional Suitability

Degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions.

Measurement function: FS = (X+Z) / 2 where

FS = **Functional Suitability**
X = *Functional Completeness*
Z = *Functional Appropriateness*

For calculating the X and Z scores, see section 2.1.1

Interpretation of test results: FS value closer to 1 is better
.

The list of functions for the preservation platform considered during the assessment is presented in Section 2.5.1 .

2.5.2.2 Performance Efficiency

Measurement function: PE = Z where

PE = **Performance Efficiency**
Z = *Capacity*

Interpretation of test results: PE smaller is better

Note: Useful element for the evaluation of the capacity can be: the number of requests or simultaneous access per unit of time; the number of simultaneous jobs accepted in the ingestion queue;r the number of tasks executed in parallel during a preservation workflow.

2.5.2.3 Compatibility

Measurement function: Co= (X+Y) / 2 where

Co = **Compatibility**
X = *Co-existence*
Y = *Interoperability*

Interpretation of test results: Co value larger is better

For calculating the X and Y scores, see section 2.1.3 .

Note This measurement is quite important for preservation systems because demonstrates also the level of integration for different technologies and systems used by the platform to implement the OAIS model. Possible measure can include the use of external systems for storage only or for complete collection management, taking into account the complexity of the integration, the interfacing mechanism and any known limitation for example in terms of supported protocols or technologies.

2.5.2.4 Usability

Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

Measurement function: Us = (K+L) / 2 where

Us = **Usability**
K = *Operability*
L = *User error protection*

Interpretation of test results: Us value closer to 1 is better

For calculating the K and L scores, see section 2.1.4

Note: for preservation systems the operability can be mainly associated to the user interface, because it should provide user all required information to interact with the

platform, although with different levels (a basic user should be able to perform a limited number of operations with respect to an administrator, which should be ready to perform complex operations to solve problems).
The user error protection can also be associated to the user interface, but should mainly reflect the capability of the system to prevent wrong operations which can have disrupting consequences (e.g. deletion of content or execution of wrong resource consuming tasks).

2.5.2.5 Reliability

Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.

Measurement function: Re = (H+J+K+L) / 4 where

Re = **Reliability**
H = *Maturity*
J = *Availability*
K = *Fault Tolerance*
L = *Recoverability*

Interpretation of test results: Re value closer to 1 is better

For calculating the H, J, K and L scores, see section 2.1 .

2.5.2.6 Security

Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization

Measurement function: Se = (H+J+K+L+M) /5 where

Se = **Security**
H = *Confidentiality*
J = *Integrity*
K = *Non-repudiation*
L = *Accountability*
M = *Authenticity*

Note: Confidentiality and integrity are often based on user authorization and authentication, with the definition of appropriate ACLs and mechanisms for protecting data from unauthorized access.

For calculating the H, J, K, L and M scores, see section 2.1.6 .

Interpretation of test results: Se value closer to 1 is better

2.5.2.7 Maintainability

Degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers.

Measurement function: Ma = (H+K+L+M) / 4 where

Ma = **Maintainability**
H = *Modularity*
K = Analysability
L = *Modifiability*
M = *Testability*

For calculating the H, K, L and M scores, see section 2.1.7

Interpretation of test results: Ma value closer to 1 is better

2.5.2.8 Portability

Degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another.

Measurement function: Po = (X+Y+Z) / 3 where

Po = **Portability**
X = *Adaptability*
Y = *Installability*
Z = *Replaceability*

Note: During the evaluation of these sub characteristics the following consideration can be evaluated: which dependencies should be taken into account during migration from one environment to the other? Which requirements should be satisfied before installing the platform? Can the solution be replaced by a similar one with additional features without changing the whole environment?

For calculating the X, Y and Z scores, see section 2.1.8

Interpretation of test results: Po value closer to 1 is better

3 Results of Research Outputs Assessment - Year 1

This chapter presents the results of the tests carried out against each research output as per the definition of functions and the measurement properties specified for the category in which the tool belongs in Chapter 2

3.1 Metadata mapping and validation

Note: Some characteristics have been assessed only in a qualitative way in this initial assessment. Thus scores have not been summed per group of characteristics. As discussed in Section 3, performing a user interface with a reasonably large group of users was not feasible, thus these characteristics were not rated.

3.1.1 PrestoPRIME Metadata Mapping Tool

3.1.1.1 Assessment results

Characteristics	Sub-characteristics	Level of Need	Fulfilment
Functional suitability			
Functional completeness			
Metadata input formats			
	Dublin Core	Mandatory	1
	ESE	Desirable	0
	EDM	Desirable	0
	EBU Core	Mandatory (for broadcasting related CoPs) / Recommended	0,5
	MPEG-7	Recommended (for CoPs using automatic content analysis)	1 (profile dependent)
	LIDO	Mandatory (for museum/gallery related) / Recommended	0
	EAD	Mandatory (for non-a/v archive related) / Recommended	0,5
Metadata output formats			
	Dublin Core	Mandatory	1
	ESE	Recommended	0
	EDM	Recommended	0,5
	EBU Core	Mandatory (for broadcasting related CoPs) / Recommended	0
	MPEG-7	Recommended (for CoPs using automatic content analysis)	0,5
	LIDO	Mandatory (for	0

Characteristics	Sub-characteristics	Level of Need	Fulfilment
		museum/gallery related) / Recommended	
	EAD	Mandatory (for non-a/v archive related) / Recommended	0,5
option to add custom formats		Recommended	0
XML representation support		Mandatory	1
RDF representation support		Recommended	0,5
Metadata model constructs			
	single -> multiple elements	Recommended	1
	multiple -> single elements	Mandatory	1
	structure using context elements	Mandatory	1
	conditional mapping based on element/attribute values	Mandatory	0,5
	map collections	Recommended	0,5
	merge string values	Mandatory	1
	split string values	Recommended	1
	number of levels in data structure	Mandatory: 2 / Recommended: 2+	1 / 1
start from example(s)		Recommended	0
start from schema		Recommended	1
configuration user interface		Mandatory	1
user interface			
	drag & drop mappings	Recommended	1
	preview	Mandatory	1
map constructs not found in available examples		Recommended	0
Functional correctness			
Tested mappings			
	Dublin Core -> EDM		1
Functional appropriateness			
ingest of metadata (e.g. SIP creation)			1
B2B exchange			1
provision to portals (e.g. Europeana)			0,7
Performance efficiency			
Time behaviour			
processing time			
	simple document (one bibliographic record)		not tested
	complex document (detailed metadata)		not tested
	document with collections		not tested
response time of user interface			not tested
time for generating preview			not tested
Resource utilization			
RAM requirement			4 GB RAM

Characteristics	Sub-characteristics	Level of Need	Fulfilment
CPU requirements			dual core
GPU requirement			not required
Capacity			
Number of metadata elements per core/minute			>300
Compatibility			
Interoperability			
mapping			
	standalone application		0
	REST		1
	SOAP		0
configuration			
	desktop UI		0
	web-based UI		1
Reliability			
Fault tolerance			
incorrect input file format			passed
load incomplete mapping definition			Passed
user creates contradicting mapping			Passed
wrong drag & drop operation			passed
example file in wrong format			depends on format
Recoverability			
incorrect input file format			Passed
load incomplete mapping definition			Passed
user creates contradicting mapping			Passed
wrong drag & drop operation			Passed
example file in wrong format			passed
Maintainability			
Modularity			
add new metadata models			Not possible
add new data types			Not possible
change mapping rules			Possible
load and modify existing mapping			Possible
Reusability			
documented service interfaces			Yes
documented exchange format between configuration and mapping components			Yes

3.1.1.2 Summarization of results

The PrestoPRIME mapping tool has been designed with a large and flexible range of functionalities, providing a tool for defining mapping between any schema-specified format. However, not of all them are mature, and not all of these functionality can be accessed via user interfaces. The development of the tool continues in the EEXCESS project, with the aim to fix this shortcomings.

3.1.2 MINT Mapping Tool

3.1.2.1 Assessment results

Note: The EBU deployment has been taken offline in autumn 2013 due to security issues. Thus not all features could be tested with a current version. This will be updated in 2014.

Characteristics	Sub-characteristics	Level of Need	Fulfilment
Functional suitability			
Functional completeness			
Metadata input formats			
	Dublin Core	Mandatory	1
	ESE	Desirable	1
	EDM	Desirable	0
	EBU Core	Mandatory (for broadcasting related CoPs) / Recommended	1
	MPEG-7	Recommended (for CoPs using automatic content analysis)	1
	LIDO	Mandatory (for museum/gallery related) / Recommended	1
	EAD	Mandatory (for non-a/v archive related) / Recommended	1
Metadata output formats			
	Dublin Core	Mandatory	0
	ESE	Recommended	0
	EDM	Recommended	0
	EBU Core	Mandatory (for broadcasting related CoPs) / Recommended	1
	MPEG-7	Recommended (for CoPs using automatic content analysis)	0
	LIDO	Mandatory (for museum/gallery related) / Recommended	0

Characteristics	Sub-characteristics	Level of Need	Fulfilment
	EAD	Mandatory (for non-a/v archive related) / Recommended	0
option to add custom formats		Recommended	1
XML representation support		Mandatory	1
RDF representation support		Recommended	0
Metadata model constructs			
	single -> multiple elements	Recommended	0,5
	multiple -> single elements	Mandatory	1
	structure using context elements	Mandatory	1
	conditional mapping based on element/attribute values	Mandatory	0,5
	map collections	Recommended	1
	merge string values	Mandatory	1
	split string values	Recommended	1
	number of levels in data structure	Mandatory: 2 / Recommended: 2+	1 / 0
start from example(s)		Recommended	1
start from schema		Recommended	0
configuration user interface		Mandatory	1
user interface			
	drag & drop mappings	Recommended	1
	preview	Mandatory	1
map constructs not found in available examples		Recommended	0
Functional correctness			
Tested mappings			
	MPEG-7 -> EBU Core		1
	DC -> EBU Core		1
Functional appropriateness			
ingest of metadata (e.g. SIP creation)			0,3
B2B exchange			0,5
provision to portals (e.g. Europeana)			1
Performance efficiency			
Time behaviour			
processing time			
	simple document (one bibliographic record)		not tested
	complex document (detailed metadata)		not tested
	document with collections		not tested
response time of user interface			not tested
time for generating preview			not tested
Resource utilization			

Characteristics	Sub-characteristics	Level of Need	Fulfilment
RAM requirement			no data
CPU requirement			no data
GPU requirement			not required
Capacity			
Number of metadata elements per core/minute			> 300
Compatibility			
Interoperability			
mapping			
	standalone application		0
	REST		0
	SOAP		0
configuration			
	desktop UI		0
	web-based UI		1
Reliability			
Fault tolerance			
incorrect input file format			passed
load incomplete mapping definition			passed
user creates contradicting mapping			passed
wrong drag & drop operation			passed
example file in wrong format			not tested
Recoverability			
incorrect input file format			Passed
load incomplete mapping definition			Passed
user creates contradicting mapping			Passed
wrong drag & drop operation			passed
example file in wrong format			not tested
Maintainability			
Modularity			
add new metadata models			possible
add new data types			no data
change mapping rules			Possible
load and modify existing mapping			possible
Reusability			
documented service interfaces			n/a
documented exchange format between configuration and mapping components			yes

3.1.2.2 Summarization of results

The MINT mapping tool has been developed for a specific case of metadata mapping, i.e. mapping from in-house formats to EBUCore. The use case is provision of metadata on publication on portals. This results in some functional limitations, however, the implemented functionality is quite stable and mature.

3.2 Storage

3.2.1 MServe

In order to access the storage capabilities of MServe the service was installed on a local dedicated VM server. The following sections describe what are the quality product characteristics and sub-characteristics and what measure elements along with measurement methods can be defined for Mserve as a storage tool.

3.2.1.1 Functional suitability characteristics

Functional suitability is evaluated considering the list of functions provisionally identified in Table 6. Most of the tests in order to verify functional suitability are done over the MServe web interface.

3.2.1.1.1 Functional Completeness

The following table shows the implemented tests and their results according to Table 6**Fout! Verwijzingsbron niet gevonden.**.

Function	Level of need	Available Yes/No
M1- store File	mandatory	Yes
M2 - restore File	mandatory	Yes
R1 - store Folder	recommended	Yes
R2 - restore Folder	recommended	No
R3 - list Files and Folders	recommended	Yes
R4 - provide File Access	recommended	Yes
R5 - compute fixity checksums and verify checksum matching	recommended	Yes
R6 - spread copies of File/Folder	recommended	Yes
R7 - recover copies of File/Folder	recommended	No
R8 - configurable ingest	recommended	Yes
D1 - record stream	desirable	No
D2 - output Streaming	desirable	No
D3- file partial restore	desirable	No
D4 - file/Folder tagging	desirable	No
D5 - file search	desirable	No
D6 - format migration / Transwrapping / Transcoding	desirable	No

D7 - produce and manage automatic workflow	desirable	Yes
D8 - metadata extraction	desirable	Yes
D9 - preview web based access	desirable	Yes

Table 12- Storage functions availability in MServe

From the above table and according to Table 6 the MServe functional completeness is calculated as

$$X=(X_1+X_2*0.5+X_3*0.25)/1.75$$

Where X_1, X_2, X_3 for MServe are:

$X_1 = 1 – 0/2$

$X_2 = 1 – 2/8$

$X_3 = 1 – 6/9$

Therefore, the final Mserve completeness score is 0.83.

3.2.1.1.2 Appropriateness

In a similar way the functional appropriateness of MServe is summarised in the following table:

Function	Level of need	Score	Remarks
M1- store File	mandatory	1	
M2 - restore File	mandatory	1	
R1 - store Folder	recommended	0.5	
R3 - list Files and Folders	recommended	0.5	
R4 - provide File Access	recommended	1	
R5 - compute fixity checksums and verify checksum matching	recommended	1	
R6 - spread copies of File/Folder	recommended	0.5	
R8 - configurable ingest	recommended	1	
D7 - produce and manage automatic workflow	desirable	1	
D8 - metadata extraction	desirable	0.5	
D9 - preview web based access	desirable	1	

According to the above table and to ***Table 6*** the MServe functional appropriateness is calculated as:

$$Z = 9 / 11$$

MServe functional appropriateness score: 0.82

The overall score of Functional suitability is therefore:

$$(0.83 + 0.82) / 2 = 0.825$$

3.2.1.2 Performance efficiency

Currently there are no performance tests run on MServe.

3.2.1.2.1 Time behaviour

Currently there are no time behaviour tests for MServe.

3.2.1.2.2 Resource utilization

In a typical MServe service with a single backup copy storage requirements are storage for the file ifself, its metadata, and its backup file.

3.2.1.2.3 Capacity

MServe storage capacity is limited from the available storage capacity of its host system.

3.2.1.3 Compatibility

MServe provides a high degree of compatibility, it has been successfully used as a core component in a number of projects such as PRESTOPrime, P4, POSTMARK.

3.2.1.3.1 Co-existence

MServe does not require exclusive use of h/w or s/w resources, it can be installed and used on a moderate h/w platform.

3.2.1.3.2 Interoperability

There are 3 main interfaces in MServe which provide a high degree of interoperability with other systems.

- HTML interface for human manipulation of the content
- HTTP/RESTful interface for machine workflows and automated systems
- WEBDAV interface to provide file system access to the content

3.2.1.4 Usability

3.2.1.4.1 Operability

Mserve's web interface is designed to run on modern browsers, it also includes "drag'n drop" features for file related operations.

3.2.1.4.2 User error protection

Mserve provides confirmation dialogue boxes to ensure user's aproval for critical operations such as deleting files. The following screenshot shows a warning message.

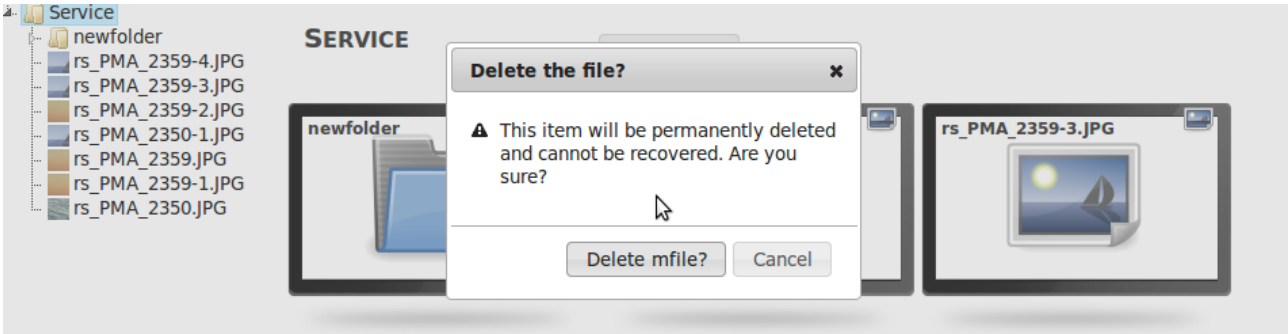

Figura 1 Confirmation dialogue deleting a file

3.2.1.5 Reliability

3.2.1.5.1 Maturity

MServe has been used already in several research projects such as PrestoPrime, POSTMARK, its TRL level score is 7 or higher. The source code of MServe is now released under LGPL and it is available on GitHub repositories.

3.2.1.5.2 Availability

There are no explict requirements for Mserve services not to be available while Mserve is running.

3.2.1.5.3 Fault tolerance

At user level Mserve requires users confirmation for critical operations, e.g. deleting file.

3.2.1.5.4 Recoverability

Periodic tasks on stored files can be defined in order to identify and restore corrupted files,

3.2.1.6 Security

Mserve services operate over HTTP or HTTPS. Using HTTPS it ensures that data exchange between a client and the MServe service will be encrypted.

3.2.1.6.1 Confidentiality

There are two ways in which you can be given access to an MServe Service.

- Directly given an URL by a service provider - this allows the use of a service without a login using a capability ID.
- If you have a login to MServe you can request access to a service from the home page. The service will appear on your home page if approved by the service provider.

3.2.1.6.2 Integrity

Currently there is no way to manipulate the access control of a service. In the future you should be able to delegate access to 3rd parties via the creation of capability URL's which will allow access to a subset of features of the service.

3.2.1.6.3 Non-repudiation

MServe logs and system logs in /var/logs can be used.

3.2.1.6.4 Accountability

MServe provides a great detail of accountability:

- MServe log files
- MServe main component log files, e.g. MySQL, Apache2 log files.

User level accountability, MServe records usage of compute resources.

3.2.1.6.5 Authenticity

Users can log-in via

- Open-ID
- Using a standard user name password.

3.2.1.7 Maintainability

3.2.1.7.1 Modularity

The MServe framework is based on several well established components:

- Web framework Django
- Web server Apache2 or Lightty
- Database MySQL,
- Job scheduler Celery
- Message broker RabbitMQ

Several of those components can be easily replaced by other ones with a similar functionality.

3.2.1.7.2 Analysability

MServe provides a great degree of analysability:

- MServe log files
- MServe main component log files, e.g. MySQL, Apache2 log files.
- User level accountability, MServe records usage of compute resources.

USAGE SUMMARY

Metric Name	Min	Max	Count	Average	Sum	Std. Dev
http://mserve/service	0	0	1	0.00	0.00	0.00
http://mserve/file	0	2 M	8	618.78 K	4.83 M	697.00 K
http://mserve/disc	0	208 G	8	61.19 G	489.51 G	62.74 G
http://mserve/disc_space	90 K	201 K	8	117.78 K	942.25 K	48.04 K
http://mserve/ingest	90 K	201 K	8	117.78 K	942.25 K	48.04 K
http://mserve/job	0	2 M	16	618.78 K	9.67 M	697.00 K
http://mserve/access	90 K	201 K	2	145.52 K	291.03 K	55.47 K
http://mserve/responsetime	0	0	2	0.39	0.79	0.01
http://mserve/jobruntime	-	-	0	-	-	-
http://mserve/jobruntime/success/	-	-	0	-	-	-
http://mserve/jobruntime/failed/	-	-	0	-	-	-

Show/Hide Variance

Figura 2MServe reported service usage summary

3.2.1.7.3 Modifiability

Mserve Container and Service tasks are configurable at user level. The default list of tasks include the following operations:

- Checksum Verification
- Archive and Backup
- Mime type detection
- Thumbnail generation
- Video Proxy Generation
- Video Meta data extraction
- Archive Retrieval
- Other batch process or web service

Additional tasks can be created and added to MServe.

3.2.1.7.4 Testability

MServe does not provide any explicit self-testing functionality.

3.2.1.8 Portability

3.2.1.8.1 Adaptability

Mserve is based on open standards and therefore is adaptable to a wide range of hardware and software platforms. The current release of Mserve has been successfully installed on Ubuntu 10.04 LTS and 12.04 LTS and other DEBIAN based Linux platforms.

3.2.1.8.2 Installability

MServe has its own setup script that can perform one of the following operations:

- installation
- uninstall
- update

Table 13 shows the various options available during the installation of MServe.

```
sysman@p4umserve:~/mserve$ ./scripts/setup-mserve.sh -h
usage: ./scripts/setup-mserve.sh [-m mserve home] [-d mserve data] [-s schema] [-a
mserve tarball]
       OPTIONS:
       -c <install|update|uninstall|dependencies> # script operation, default: install
       -m <MSERVE home directory> # default: /var/opt/mserve
       -d <MSERVE data directory> # default: /var/opt/mserve-data
       -l <MSERVE log directory> # defautl: /var/log/mserve
       -t <MSERVE HTTP server> # [apache|lighttpd] default: apache
       -u <MSERVE admin user name> # administrtor user name
       -p <MSERVE admin password> # admin password
       -e <MSERVE admin email> # administrator email
       -U <Database admin user> # Database admin user
       -P <Database admin password> # Database admin password
       -s <schema> # Schema (http/https)
       -g install from git repository
       -v verbose mode

       example: ./scripts/setup-mserve.sh -s https
```

Table 13: MServe installation options

The installation procedure is configurable and comprehensive. It can handle required dependences on the fly and does not require any other information during the installation procedure.

3.2.1.8.3 Replaceability

MServe is based on open standards, most of its storage functionality can be easily replaced by other storage tools.

3.2.1.9 Summarization of results

Although the assessment process for MServe is not complete yet, the tool provides a high degree of functionality and interoperability. Performance measurements need to be done in order to complete the assessment.

3.2.2 LTFS Archiver

The tests have been done quite extensively and are deemed sufficient to provide a complete evaluation. Nevertheless it will be decided if necessary to further assess the tool in the second year in relation to the possibility that:

- New functional requirements will be identified basing on CoPs interviews and questionnaires
- New development of the tool will be carried on
- New interesting material will be collected for tests

The audiovisual material that has been used for the tests roughly includes all the files provided till the date of writing within the project. For the nature of the tests it is not that important the file formats but the size of files and their statistical distribution in size (e.g. big for master quality and smaller for matadata and possible proxy representations).
To date, the actual Presto4U collection is already enough varied in that sense but could be better in the second year.

3.2.2.1 Functional suitability characteristics

Functional suitability is evaluated considering the list of functions provisionally identified in Table 6.

3.2.2.1.1 Completeness

According to the generic guidelines given in chapter 2 , the completeness is calculated following the measurement function:

$$X=(X_1+X_2*0.5+X_3*0.25)/1.75$$

where X_1, X_2, X_3 must be calculated considering respectively the percentage of functions available or not within the "*level of need":* mandatory, recommended and desirable.

Table 14 shows the storage functions with an additional column specifying whether they are provided or not by LTFSArchiver.

Function	Level of need	Available Yes/No
M1- store File	mandatory	Yes
M2 - restore File	mandatory	Yes
R1 - store Folder	recommended	Yes
R2 - restore Folder	recommended	Yes
R3 - list Files and Folders	recommended	Yes
R4 - provide File Access	recommended	Yes
R5 - compute fixity checksums and verify checksum matching	recommended	Yes
R6 - spread copies of File/Folder	recommended	No
R7 - recover copies of File/Folder	recommended	No
R8 - configurable ingest	recommended	Yes
D1 - record stream	desirable	No
D2 - output Streaming	desirable	No
D3- file partial restore	desirable	No

D4 - file/Folder tagging	desirable	No
D5 - file search	desirable	No
D6 - format migration / Transwrapping / Transcoding	desirable	No
D7 - produce and manage automatic workflow	desirable	No
D8 - metadata extraction	desirable	No
D9 - preview web based access	desirable	No

Table 14- Storage functions availability in LTFSArchiver

Here follows the calculation of the score for the functional completeness:

$X_1 = 1 – (0/2) = 1$
$X_2 = 1 – (2/8) = 0.75$
$X_3 = 1 – (9/9) = 0$

$X = (1 + 0.75 * 0.5 + 0 * 0.25) / 1.75 =$ **0.786**

3.2.2.1.2 Appropriateness

According to the generic guidelines given in chapter 2 , the appropriateness is calculated following the measurement function Z= A/B where A is sum of the scores of the implemented functions and B is the total amount of implemented functions.
In this evaluation is important to understand how well the considered functions are implemented and to which extend they are satisfying the specific needs.

Table 15 gives for each function a score among 0 (not appropriate), 0.5 (partially appropriate) and 1 (completely appropriate). The column *Remarks* justifies the choice.

Function	Level of need	Score	Remarks
M1- store File	mandatory	1	Basic but effective feature of the tool.
M2 - restore File	mandatory	1	Basic but effective feature of the tool.
R1 - store Folder	recommended	1	Recursively creates on the target data tape the same folder structure of the source.
R2 - restore Folder	recommended	1	Recursively creates on the target storage area, the same folder structure of the source data tape.
R3 - list Files and Folders	recommended	0.5	Available through the make-available feature. An explicit service for the purpose is foreseen as a future improvement.
R4 - provide File Access	recommended	1	Available through the make-available feature.
R5 - compute fixity checksums and verify checksum matching	recommended	1	Completely available through several modalities (during upload, download, on demand) and algorithms (md5,sha1).
R8 - configurable ingest	recommended	0.5	More configuration options would be desirable.

Table 15 - Storage functions appropriateness in LTFSArchiver

The final score for this sub-characteristic is Z = 7 / 10 = **0.7**

The overall score of Functional suitability according to 2.3.2.1 is therefore:

(0.786 + 0.7) / 2 = 0.743

3.2.2.2 Performance efficiency

This characteristic concerns the performances in terms of elapsed time, throughput and resource utilization for accomplishing specific tasks. This kind of tests make sense only if a reference system is fixed and considered. For LTFSArchiver the hardware/software system has been configured as follows:

- Host HP Workstation Z800
  - 8 cores 2.5 Ghz,
  - 8 GB RAM,
  - Linux release “Ubuntu 12.04 LTS”
  - LTFSArchiver 1.2 installed
  - AH401A PCIe 8Gb 2-Port Fibre Channel (QLogic) HBA for the connection to the HP LTO5 library and its drive
  - HP Smart Array P212 256M for the external SAS connection to the desktop LTO5 drives
  - Gigabit Ethernet connection to remote disk storage areas where the test files are stored
- HP MSL2024 2 LTO-5 , LTO library with 24 slots and drive LTO5

3.2.2.2.1 Time Behaviour

The performed tests have been run as follows: the whole Presto4U data set, as available at November 2013, was used, split into 4 folders, one for each provider, namely: MediaEval-Blip (888MB), RAI (15GB), TATE (11GB), and UIBK (25GB). The following operations were requested on that folder basis.

- Ingest operations for copy A, requiring computation of MD5 checksums for all files both at source (on disc) and on destination (LTO5 tape of PoolA) and first integrity check.
- Ingest operations for copy B, requiring computation of MD5 checksums for all files only at destination (LTO5 tape, of PoolB) and integrity checks against results of the previous operation.
- Restore operations from copy A.
- The whole set of 12 requests was repeated twice to verify the consistency of the recorded timing.

The results of this measurement is summarised in the box below in which the requests are listed in reverse execution order[34].

operation	checksum	callingtime	starttime	endtime	waitingtime	exectime	sourcesize
R		15:36:47	15:57:23	16:03:31	00:20:36	00:06:08	25168
R		15:36:44	15:53:44	15:57:23	00:17:00	00:03:39	10918
R		15:36:40	15:44:52	15:53:44	00:08:12	00:08:52	15162
R		15:36:37	15:38:25	15:44:52	00:01:48	00:06:27	888
W	FILE	14:56:33	15:12:34	15:27:49	00:16:01	00:15:15	25168
W	FILE	14:56:30	15:06:46	15:12:31	00:10:16	00:05:45	10918
W	FILE	14:56:26	14:59:23	15:06:45	00:02:57	00:07:22	15162
W	FILE	14:56:23	14:57:54	14:59:20	00:01:31	00:01:26	888
W	MD5_both	13:49:49	14:14:10	14:27:43	00:24:21	00:13:33	25168

[34] Restore of the second run was executed from items archived in the previous run

```
 W | MD5_both | 13:49:46 | 14:07:56 | 14:14:09 | 00:18:10 | 00:06:13 | 10918
 W | MD5_both | 13:49:43 | 13:53:37 | 14:07:55 | 00:03:54 | 00:14:18 | 15162
 W | MD5_both | 13:49:40 | 13:51:13 | 13:53:34 | 00:01:33 | 00:02:21 | 888

-------------------------------------------------------------------------------------------------
 operation | checksum | callingtime | starttime | endtime | waitingtime | exectime | sourcesize
-----------+----------+-------------+-----------+----------+-------------+----------+------------
 W | FILE | 10:19:03 | 11:36:22 | 11:47:22 | 01:17:19 | 00:11:00 | 25168
 W | FILE | 10:18:59 | 11:31:14 | 11:36:18 | 01:12:15 | 00:05:04 | 10918
 W | FILE | 10:18:56 | 11:23:43 | 11:31:10 | 01:04:47 | 00:07:27 | 15162
 W | FILE | 10:18:53 | 11:20:36 | 11:23:40 | 01:01:43 | 00:03:04 | 888
 W | MD5_both | 10:18:20 | 11:01:43 | 11:16:39 | 00:43:23 | 00:14:56 | 25168
 W | MD5_both | 10:18:16 | 10:53:55 | 11:01:42 | 00:35:39 | 00:07:47 | 10918
 W | MD5_both | 10:18:13 | 10:45:32 | 10:53:54 | 00:27:19 | 00:08:22 | 15162
 W | MD5_both | 10:18:10 | 10:43:58 | 10:45:29 | 00:25:48 | 00:01:31 | 888
 R | | 10:14:03 | 10:34:41 | 10:40:50 | 00:20:38 | 00:06:09 | 25168
 R | | 10:14:00 | 10:31:02 | 10:34:40 | 00:17:02 | 00:03:38 | 10918
 R | | 10:13:57 | 10:22:02 | 10:31:01 | 00:08:05 | 00:08:59 | 15162
 R | | 10:13:54 | 10:15:34 | 10:22:02 | 00:01:40 | 00:06:28 | 888
```

Table 16: Time behaviour in LTFS Archiver

The requests have been submitted as batches, thus the LTFSArchiver queued and executed them in sequence. The overall restore took about 25 minutes, archiving with double checksum took about 35 minutes, and archiving with only one checksum took about 30 minutes.
We can observe that the execution time for the single request can result higher than expected considering the folder size. This is caused by the possibility that the execution of that request required the tape to be loaded into the drive and mounted, while the subsequent requests found the tape ready. Another possible delay is caused by the time needed for seeking the file position on the tape.

A time behaviour performance assessment can be computed considering the generic formula X= (B-A)/C where B-A is the duration of the ingest/restore operation and C is the maximum allowable duration of that operation. In order to provide a reasonable value to C, we consider the theoretic limit of read/write speed of LTO5 that by specifications is 140 MB/s and the size of the file/folder being handled, according to the following formula:

C = FS / S * c where:
FS is the file or folder size in MB
S is the maximum theoretical read/write speed of LTO5 i.e. 140 MB/s
C is a coefficient greater than zero taking into account that a real system is accepted to be slower that what the strict specification of pure read/write of LTO are.
Assuming c equal to 3 and making an average on the instances being run, we obtain **X=0.79** meaning that on average the expected execution time is under the maximum established time.
Table 17 reports the time behaviour scores obtained in the tests done, it is possible to note that for most of the instances run, the score is below one. The variance is quite high (in one case up to 20) because of the tape movement to and from the library and tape positioning before read/write operations. In particular bad performance is notable for lower sized folders where the aforementioned overhead is more relevant.

Read/write	Checksum	Op.Duration	Size [MB]	Num. R/W	Maximum Time [s]	TimeBehaviour Score
R		00:06:08	25168	1	539,31	0,7
R		00:03:39	10918	1	233,96	0,9
R		00:08:52	15162	1	324,90	1,6
R		00:06:27	888	1	19,03	20,3

W	FILE	00:15:15	25168	2	1078,63	0,8
W	FILE	00:05:45	10918	2	467,91	0,7
W	FILE	00:07:22	15162	2	649,80	0,7
W	FILE	00:01:26	888	2	38,06	2,3
W	MD5_both	00:13:33	25168	3	1617,94	0,5
W	MD5_both	00:06:13	10918	3	701,87	0,5
W	MD5_both	00:14:18	15162	3	974,70	0,9
W	MD5_both	00:02:21	888	3	57,09	2,5
					6703,20	**0,82**
W	FILE	00:11:00	25168	2	1078,63	0,6
W	FILE	00:05:04	10918	2	467,91	0,6
W	FILE	00:07:27	15162	2	649,80	0,7
W	FILE	00:03:04	888	2	38,06	4,8
W	MD5_both	00:14:56	25168	3	1617,94	0,6
W	MD5_both	00:07:47	10918	3	701,87	0,7
W	MD5_both	00:08:22	15162	3	974,70	0,5
W	MD5_both	00:01:31	888	3	57,09	1,6
R		00:06:09	25168	1	539,31	0,7
R		00:03:38	10918	1	233,96	0,9
R		00:08:59	15162	1	324,90	1,7
R		00:06:28	888	1	19,03	20,4
					6703,20	**0,76**
Max speed [MB/s]	140					**Mean Score**
Coeff.	3					**0,79**

Table 17 - Time Behaviour scores of the test

Regarding the Response Time, it has not been considered in this case, as the requests are handled asynchronously and the response time is negligible with respect to execution time.

3.2.2.2.2 Resource utilization

When performing the tests described in the previous section, a tool for monitoring the system resources has been run in the background in order to collect data that have been plotted afterwards. For that purpose we used an open source tool named *nmon* which also provides effective ways for plotting and analysing the collected data.

Figure 17 shows the overall CPUs usage (user+system averaged on the 8 available cores) along the timeline, also in relation with the points in time where the Restore and Store operation started (see the little circles at the base of the chart).

It can be seen that the system is always in good health with CPU usage peaks always under 30%, this means that a host configured in this way can be used also for other concurrent tasks (see 3.2.2.3.1 on co-existence) or could be downsized for example to a total of 4 cores instead of 8.

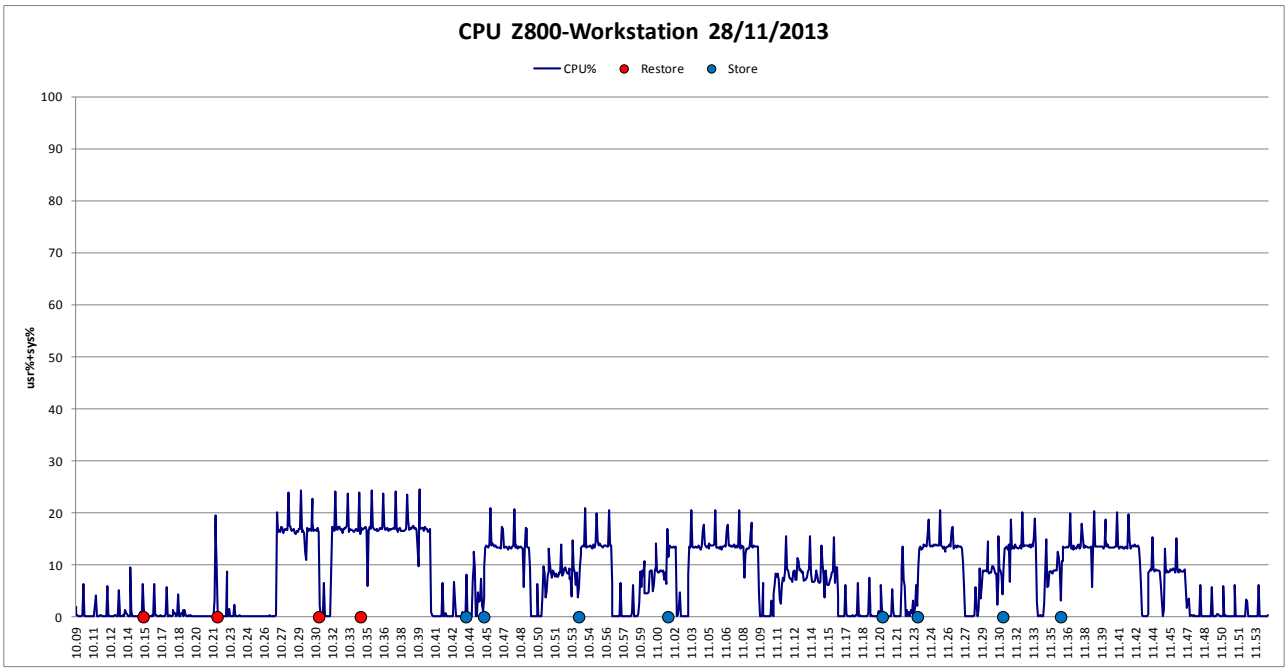

Figure 17 - Overall CPU usage

Figure 18 shows the read and write transfer rates on the disks of the host used as destination for the Restore operations and as source for the Store operations. Store and Restore operation starts are indicated on the graph as small circles at the base.

The blue coloured line represents the write speed in correspondence of the Restore operations, it is visible that the rate stands around 70 MB/s.

The orange coloured line represents the read speed in correspondence of the Store operations (read from disk and write to LTO). From time 10:40 to time 11:16 (central part of the graph) is showed the behaviour of disks for the Store operation with the option MD5_both. This means that the files have to be read twice, once for writing to the LTO and the other to calculate the MD5 checksum from the source. This justifies the course of the graph that has a smoother part when writing to LTO, which stands to a quite regular transfer rate around 60 MB/s, and a sharper part that corresponds to the reading just into memory for MD5 calculation. In this last case disks reach spikes as high as 250 MB/s because of disk caching. The final part of the graph is related to the Store with the FILE option, this means that checksums are not calculated at the source as they are user provided with an input text file. This last part is thus smooth as for the Restore operation and presents again a transfer rate around 60 MB/s.

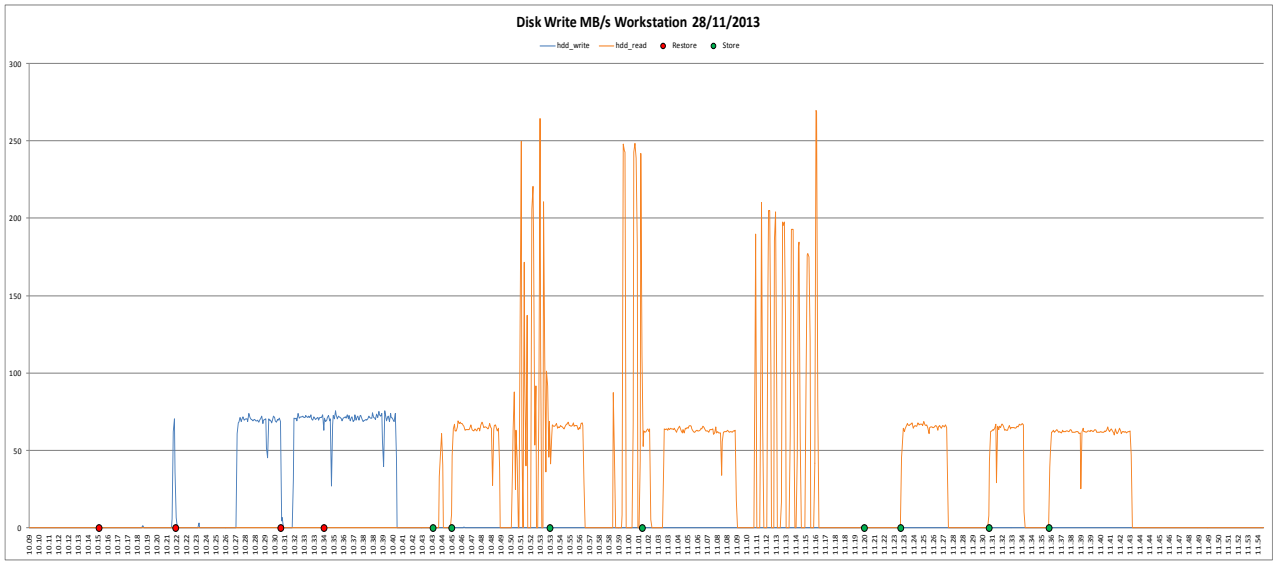

Figure 18 - Disk read/write transfer rate

Figure 19 shows average, maximum and weighted average of disk read and write along all the test duration. The most significant values are the weighted means that give an average of the speed considering only when disks are effectively working. Those values stand around 70 MB/s for writing and 100 MB/s for reading which is roughly the limit of the disks. This demonstrates that the LTO transfer rate is today comparable with common SATA disks.

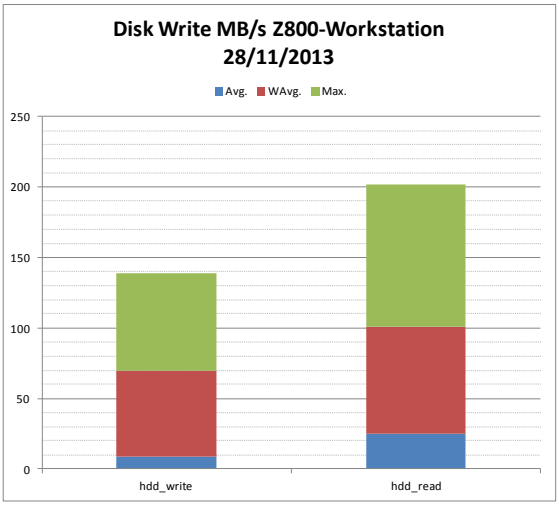

Figure 19 - Disk read/write average

Figure 20 with the green line, shows the free memory on the host, for convenience also the disk read and write have been drawn.
The overall system memory is 6 GB and we notice that when the host is idling, the free memory is around 5 GB. Then, during Restore and Store operations, the memory is nearly entirely used. This is because data is read from source, buffered as much as possible into the memory and finally written to the destination. At the end of all operations the memory is freed back to the original situation.

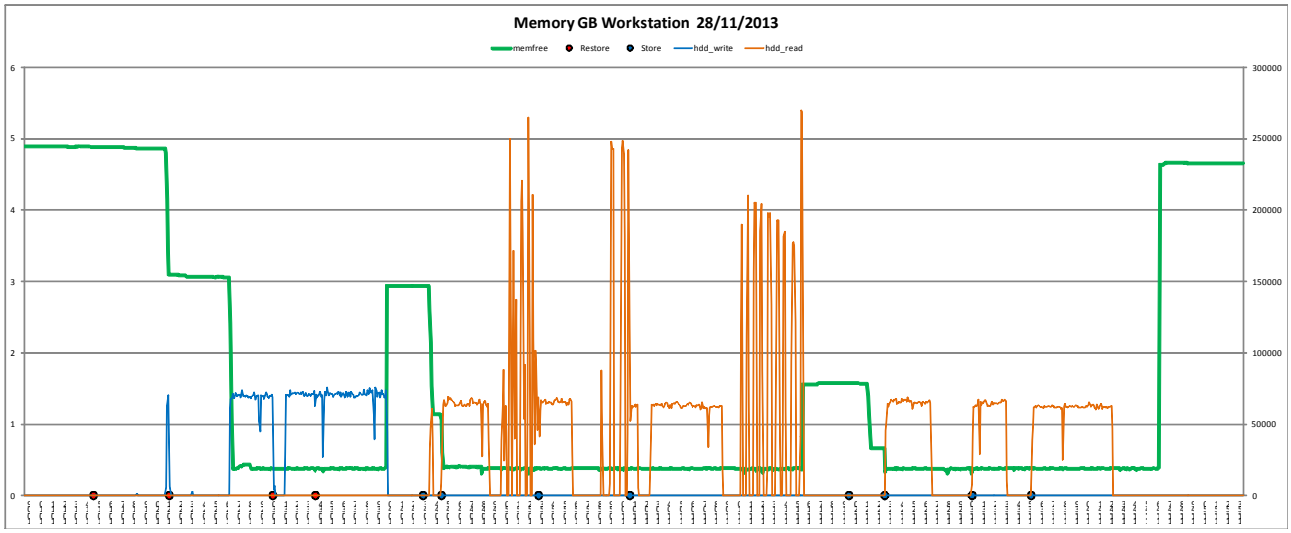

Figure 20 – Free memory

3.2.2.2.3 Capacity

Stress test on the reference system and see when the quality of service goes under acceptable limits.

The system basically works as a booking system that receives requests and later satisfies them trying to optimize the operations, for example by reducing the number of tape movements to and from the library. This kind of decoupling means that the majority of the services (e.g. Ingest and Restore) are asynchronous, they are hence queued very quickly and executed according to the optimization we said.

A stress test consists in sending a burst of operations, the system has not problems with that because it just take note of the requests by writing them on the internal database. The operations are then executed with the maximum performance allowed by the hardware configuration in use.

This way of working basically assure that the system will hardly get stuck because of requests overload while continuing to work with the maximum possible efficiency. Of course when a lot of requests are enqueued, the average wait time i.e. the time elapsed from booking and start of the operation, will grow indefinitely.

On our system we measured an average throughput of around 37 MB/s, this number can be obtained from chapter 3.2.2.2.1 as 0,79*140/3.

3.2.2.3 Compatibility

Overall compatibility score:Co=(X+Y)/2=15/16=0.9375 where X and Y were obtained as follows.

3.2.2.3.1 Co-existence

With respect to the definition of section 2.2.2.3 and Table 6 the values for LTFSArchiver are:

#(B) = 8; thus all cases are relevant.

#(A)= 7; because b6 is not met: the service can be configured for not to use a library/drive, but a used library/drive should not be shared with other services.

X=7/8=0.875

3.2.2.3.2 Interoperability

With respect to the definition of section 2.2.2.3 and Table 6 the values for LTFSArchiver are:

#(B) = 4; thus all cases are relevant. In particular the LTO tape with LTFS can be the object of exchange, if needed.

#(A)= 4; all cases are met; b2 is interoperable through CGI over HTTP/HTTP with defined and documented APIs; b3 is interoperable by means of XML, and equivalent JSON, format according to defined XML Schema; b4 is met as LTFS is open standard.

Y= 4/4 = 1

3.2.2.4 Usability

The software has a complete documentation available both as pdfs and web pages callable also from the operational GUI.

3.2.2.4.1 Operability

The actual API is the result of a quite recent rationalization that has made the most from previous experiments and usage on the field. The GUI is easy to comprehend and use even if aesthetic improvements are possible.

3.2.2.4.2 User error protection

We considered the following dangerous operations with respect to user error:

Danger	User error protection
Overwrite content on tape	Yes, overwrite is impossible as new storage instances are created for each writing request.
Overwrite content in restore	Yes, overwrite on destination is never allowed. A bad request is immediately returned.
Using the wrong tape	Yes, impossible in the library where barcode labels are required. For tapes on shelves a security mechanism is provided as the label is also written on tape (error possible at first use).
Formatting a used tape	Yes, by default the service doesn't format an already formatted tape, which is possible only by specific request. However if the client insist by mistake the old content is lost.

Table 18: User error protection measures in LTFS Archiver

Resulting **X = 4/4 = 1.0**

The result of this measurement is upper bounded to 1, where higher=better.

3.2.2.5 Reliability

3.2.2.5.1 Maturity

LTFSArchiver is higher than TRL4 (probably TRL6) as it has been developed within the PrestoPrime project as a prototype and has proved to work fine during the testbeds and with other internal testings.

3.2.2.5.2 Availability

Not tested

3.2.2.5.3 Fault tolerance

Not tested

3.2.2.5.4 Recoverability

Not tested

3.2.2.6 Security

3.2.2.6.1 Confidentiality

No authentication and authorization is provided to date even it is foreseen to provide this.

3.2.2.6.2 Integrity

No authentication and authorization is provided to date even it is foreseen to provide this.

3.2.2.6.3 Non-repudiation

Not tested

3.2.2.6.4 Accountability

Not tested

3.2.2.6.5 Authenticity

Not tested

3.2.2.7 Maintainability

3.2.2.7.1 Modularity

LTFSArchiver is a module itself thought to be integrated easily in a wider system like for example a Media Asset Management System. API are driven by http protocol, are easy and well documented thus allowing effective integration in a distributed environment.

3.2.2.7.2 Analysability

Not tested

3.2.2.7.3 Modifiability

The source code is written with Linux shell scripting, the overall logic of the software is explained in the main documentation while the most important technical details are revealed and commented within the code. This should allow quite easy modifications of

the software provided that a sufficient knowledge of the operative system, of the devices and of the language are gained. Beside, LTFSArchiver gives the possibility to customize several aspects of the system by simply modifying using its configuration files.

3.2.2.7.4 Testability

Not tested

3.2.2.8 Portability

3.2.2.8.1 Adaptability

To date, only HP and IBM are supported with specific models and the compatibility with different drives/libraries should be evaluated and provided. The software is straightforward and does not have many strong dependencies with other software. Migration to newer Linux releases should not be problematic.

3.2.2.8.2 Installability

LTFSArchiver comes with its own installer that includes some manual or assisted steps like the configuration of drives and libraries. Further improvements could be done especially with this automatic recognition of attached devices.

3.2.2.8.3 Replaceability

LTFSArchiver can be used in place of a software doing more or less the same thing with comparable performances. The API is not standard but straightforward hence the replacement should not be critical.

3.2.2.9 Summarization of results

LTFSArchiver has reached good scoring in nearly all the characteristics considered. Its principal point of strength are the simplicity of use and maintainability, the good interoperability given by the LTFS and the low consumption of resources. Still some improvements have to be done especially in the field of Security and Configurations.

3.3 Quality assessment

3.3.1 VidiCert

Note: Some characteristics have been assessed only in a qualitative way in this initial assessment. Thus scores have not been summed per group of characteristics. As discussed in Section 3, performing a user interface with a reasonably large group of users was not feasible, thus these characteristics were not rated.

3.3.1.1 Assessment results

Characteristics	Sub-characteristics	Level of Need	VidiCert fulfillment of functions/features/capabilities	Completeness for AV QC in Preservation/Migration	Appropriateness for AV QC in Preservation/Migration
Functional characteristics					
Automatic Defect Analysis Functions					
	Analogue Synchronisation Errors Aliases: lost lock, time-base corrector (TBC) hit, video breakup, lost video sync, horizontal distorsion	Mandatory	The VideoBreakup detection module of VidiCert allows for the detection of different kinds of analogue synchronisation errors in a general way. Severe, horizontally line oriented defects, which are apparent for more than one frame, are detected. A severity measure based on the distorted area is provided. Furthermore the visualisation of detected analogue synchronisation error segments over video time allows for efficient interactive verification.	1	1
	Coloured Frames Aliases: Black Frames, Monochrome Frames, Uniform Color Frames	Mandatory	The Monochrome Frame module of VidiCert allows for the detection of an arbitrary number of user definable colours in one go. The visualisation of detected coloured frame segments over video time allows for efficient interactive verification.	1	1
	Digital Tape Dropouts Aliases: digital video tape dropout, digital hits, digital tape hits	Mandatory	The DigiBeta Dropout module of VidiCert allows for the detection of different classes (luminance dropouts, chrominance dropouts) of DigiBeta tape dropouts. The visualisation of detected DigiBeta dropout segments over video time allows for efficient interactive verification.	0,5	1
	Video Noise	Desireable	The Noise module of VidiCert	1	1

Characteristics	Sub-characteristics	Level of Need	VidiCert fulfillment of functions/features/capabilities	Completeness for AV QC in Preservation/Migration	Appropriateness for AV QC in Preservation/Migration
	Aliases: image noise, noise		allows for the temporally dense estimation of the amount of video noise. This information provided to the user as a line graph over the video time allows for efficient interactive verification of the video condition in respect to noise.		
	Blurriness Aliases: out of focus, blur detection, sharpness	Recommended	The Blurriness module of VidiCert allows for the temporally dense estimation of the amount of video blurriness. This information provided to the user as a line graph over the video time allows for efficient interactive verification of the video condition in respect to blur.	1	1
	Video Test Pattern Aliases: test card, colour bars	Mandatory	The Test Pattern module of VidiCert allows for the detection of an arbitrary number of different test pattern. The user defines a test pattern by providing a still image with the desired test pattern content. Furthermore the visualisation of detected test pattern segments over video time allows for efficient interactive verification.	1	1
	Video Field Order Aliases: field order, field dominance	Recommended	Not supported	0	0
	Audio Silence Aliases: mute test, minimum level	Mandatory	The Silence module of VidiCert allows for the detection of video segments where the audio level is lower than a user defined threshold. The detection is carried out individually for each audio channel. Furthermore the visualisation of detected audio silence segments of all audio channels over video time allows for efficient interactive verification of actual audio channel usage.	1	1
	Audio Encoding Format Change	Recommended	The Dolby-E module of VidiCert allows for the detection of segments where audio is encoded in Dolby-E. The	1	1

Characteristics	Sub-characteristics	Level of Need	VidiCert fulfillment of functions/features/capabilities	Completeness for AV QC in Preservation/Migration	Appropriateness for AV QC in Preservation/Migration
			detection is carried out individually for each audio channel. Furthermore the visualisation of detected Dolby-E segments of all audio channels over video time allows for efficient interactive verification of the actual audio encoding formats used within the channels.		
General Analysis Properties					
	Analysis profiles	Mandatory	VidiCert allows the user to define any number of different analysis profiles. In each analysis profile the detectors desired as well as its parameters can be configured by the user	1	1
	No reference video required	Mandatory	All AV quality assessment functionality offered in Vidicert is capable of working with the only one copy and don't requires any other reference video	1	1
	Detection of multi-generation defects	Recommended or Mandatory	All assessment fuctionality of VidiCert is based on decoded, raw audiovisual data, therefore there is no limitation to detect multi-generation defects, as long they are visible or audible within the audiovisual essence.	1	1
	Multi-Resolution support	Mandatory	VidiCert is designed to support the detector analysis functions as well as the interactive verification functions in an resolution independent way	1	1
Interactive Validation/Verification Functions					
	Check file efficiently for right content	Mandatory	The interactive Summary application visualises content of the entire video in a temporally dense way. Compared with traditional checking of content in only a few temporal locations (spot checking) this allows a very efficient quick check of the entire video content in a single view.	1	1

Characteristics	Sub-characteristics	Level of Need	VidiCert fulfillment of functions/features/capabilities	Completeness for AV QC in Preservation/Migration	Appropriateness for AV QC in Preservation/Migration
			Direct navigation to any timepoint in the video allows to check whether the video contains the right content / program, e.g. by checking the title and/or the trailer information. Automatic checking for test patterns allows the direct detection when wrong tape segments have been migrated		
	Human validation of automatic analysis functions	Mandatoy	Any automatic detection can be approved or discarded by the user. This annotation status is stored in the metadata document.	1	1
	Interactive defect annotation support	Mandatory	Annotations can be created based on a comprehensive classification scheme. A list of favorite defects can be configured that are accessible on the UI. The time and duration of automatic as well as manually created defects can be modified.	1	1
	Overall quality rating support	Mandatory	VidiCert supports ratings. The desired ratings can be configured via a classification scheme. If different QC tasks are needed (e.g. migration technical QC and content QC), then rating is possible individually for each QC task.	1	1
	Defect severity based operation/validation	Recommended	All automatic detectors provide a relevance measure for a detection. The UI supports to sort the detection list by this relevance measure.	1	1
	Video output devices	Desirable	VidiCert supports to detach the video playback on a second screen.	0,7	1
	Interlaced video output	Recommended	VidiCert does not support interlaced video output	0	0
Performance efficiency					
Time behaviour					
	Frame Accurate Player	Mandatory	The decoders used by VidiCert support frame accurate positioning. VidiCert supports	1	1

Characteristics	Sub-characteristics	Level of Need	VidiCert fulfillment of functions/features/capabilities	Completeness for AV QC in Preservation/Migration	Appropriateness for AV QC in Preservation/Migration
			navigation in single frame steps.		
	Responsive User Interface	Mandatory	The VidiCert user interface is implemented in a way so that it is immediately responsive to all user interactions.	1	1
	Throughput of automatic tools	Recommended	The runtime varies greatly by the processed material and detector configuration. The results below are given for the standard detector settings on a state-of-the-art machine (3,4 GHz HexaCore CPU and an NVIDIA GTX 780 Card) - SD: ~0,7 - 1.5 x video duration - HD: ~4 - 8 x video duration The most efficient/minimal configuration for interactive QC only requires a factor of 0.2 (SD) and 0.6 (HD) of the video duration for processing.	0,8	1
Resource utilization					
	Efficient RAM usage	Desirable	VidiCert utilizes the following amount of RAM SD: 300 MB HD: 600MB	1	1
	Efficient GPU usage	Recommended	VidiCert uses the GPU for selected algorithms and provides a significant performance increase for those parts.	1	1
	Efficient CPU usage	Recommended	All Processor Cores can be utilized for SD and HD. On defective parts of a video more processing is done on the GPU and CPU utilization decreases.	0,9	1
	Efficient Network usage	Recommended	The content is read/accessed only once	1	1
Capacity					
	Capability to scale throughput	Recommended	Modules can be activated/deactivated and parameters be optimized via UI and web service interface. Additional CPU Cores can be utilized due to multithreading capabilities. Analysis can be distributed on machines on a file basis (e.g. by using drop folders or the web service interface)	1	1

Characteristics	Sub-characteristics	Level of Need	VidiCert fulfillment of functions/features/capabilities	Completeness for AV QC in Preservation/Migration	Appropriateness for AV QC in Preservation/Migration
Compatibility					
Interoperability					
	Supported video container formats	Recommended	VidiCert: - MPEG TS and PS supported - MXF supported - MP4 supported - MOV supported - AVI supported	1	
	Supported commonly used video encoding formats	Recommended	VidiCert: - MPEG-2 (incl. IMX50 and XDCAM HD) supported in MXF and MPEG-2 program stream - MPEG-4 AVC (H.264) supported in AVI, MP4 and MOV - JPEG2000 supported in MXF (SAMMA format) - DV and DVCPro supported	0,9	1
	Supported additional video encoding formats	Desirable	VidiCert: - Uncompressed 10bit in MOV (e.g. P4U Tate) not supported - Uncompressed 10bit in MXF supported - ProRes not supported - DCP, MAP not supported - WMV supported	0,4	1
	Supported audio encoding formats	Recommended	VidiCert: - PCM supported in analysis for MXF and playback - MPEG-1 Audio, MPEG-1 Layer 3, MPEG-2 Audio supported in playback - AAC supported in playback - WMA supported in playback	0,3	1
	Standardized metadata output format	Recommended	VidiCert uses the ISO/IEC standard 15938-9:2005/AMD1:2012 (MPEG-7 AVDP) with standard compliant quality extensions.	1	1
	Workflow integration support (Webservice, drop folder)	Recommended	VidiCert supports drop folders and provides a REST interface. Multiple machines running VidiCert Analyser can access a shared drop folder.	1	1
Reliability					
Fault tolerance					
	Incorrect file input	Mandatory	VidiCert reports if a file cannot be	1	1

Characteristics	Sub-characteristics	Level of Need	VidiCert fulfillment of functions/features/capabilities	Completeness for AV QC in Preservation/Migration	Appropriateness for AV QC in Preservation/Migration
			processed and continues to be fully operable.		
	User operation errors	Mandatory	VidiCert does not support undo functionality. When closing the application it prompts for unsaved changes.	0,2	1
	Tolerance against analysis failures	Mandatory	The analysis is detached in separate processes where errors do not affect the UI or service interfaces. Vidicert additionally monitors the health of a single file analysis process and stops it if it is stalled. The analysis of further video files is not affected.	1	1
Recoverability					
	Recoverability of Jobs	Mandatory	VidiCert persists file analysis job information and can reanalyse jobs after restart.	1	1
Accountability					
	Documentation of user decisions	Recommended	VidiCert supports to log user actions. For human defect annotation and for verification of automatic detections, the operating user is documented in the MPEG-7 AVDP metadata document.	1	1
Modularity					
	Analysis functional extension	Recommended	VidiCert offers a Plug-In architecture for use case or customer specific QC functions within the automatic Analyser application as well as within the interactive Summary verification application	1	1
	Modular I/O components	Recommended	Decoders can be shipped in a modular way. Additional format support can be implemented via plug-ins	1	1
	Capability to extend visualisation and interaction Functions in a modular way	Recommended	All visualization components are implemented as plug-ins and can be replaced by other visualization components. The visualization can also be extended by new plug-ins	1	1
Reusability					
	Applicability in other	Desirable	VidiCert can be applied in all of	0,9	

Characteristics	Sub-characteristics	Level of Need	VidiCert fulfillment of functions/features/capabilities	Completeness for AV QC in Preservation/Migration	Appropriateness for AV QC in Preservation/Migration
	application domains		the application domains due to its customization capabilities and wide range of detectors		
Testability					
	Ground truth for regression tests	Recommended	Specific development ground truth data is available and results are benchmarked on updates	1	1
Portability					
Installability					
	Installation tools available	Mandatory	VidiCert provides a setup wizard with some basic installation options. Uninstall is available via the Windows control panel.	1	1
	Documentation available	Mandatory	Both interfaces are documented - User Interface Manual for operators and service interfaces for integration developers.	1	1

3.3.1.2 Summarization of results

VidiCert is designed for video quality analysis in a professional workflow. The tool and the related user interfaces are quite mature. More discussions with CoPs are needed to assess how well the functionalities fit the needs of other CoPs than broadcast archives and video production.

3.4 Preservation platforms and systems

In the following we report the results of the assessment of two preservation platforms (Archivematica and P4), which have been selected for the first year assessment. We used the measurement plan described in Section 3.5.

For the first year assessment we consider just two examples: one is Archivematica and the other is P4. The reasons for this choice are the following: (1) Archivematica is currently under evaluation in several CoPs as one of the candidate solutions for the preservation archive (2) P4 was the resulting platform developed and tested in the PrestoPRIME project, and already integrates several tools which are under evaluation during the first year.

Both platforms are available under open source license and the source code is hosted on public repositories (GitHub). Both have been designed to be compliant with OAIS model and their user interfaces provide explicitly functionalities based on the main OAIS entities.
The main difference between the two solutions is the level of maturity, as detailed in the following. For further information about the two platforms, refer to the descriptions in Section 2, which contain also references to the project pages and documentation.

Note: some of the characteristics have been assessed only in a qualitative way in this initial assessment, as anticipated in Section 3.5. Therefore we didn't calculate the total score resulting from each group of characteristics.

3.4.1 Archivematica

3.4.1.1 Platform installation

In order to assess the characteristics of Archivematica, we first downloaded and installed the provided virtual machine with the last stable release. The VM contains a Ubuntu operating system with an instance of the platform which is started at boot. In this way we could test the platform in a properly configured environment with all available features. Then we checked out from code repository the source code and built it from scratch following the documentation. Both approaches produced a running instance of the platform, although some features were not properly working (this should be further investigated in the future).

3.4.1.2 Technical facts about Archivematica

Concerning the adopted technologies, Python is used for implementing micro-services (requires Django MVC framework). For testing a Virtual Appliance is provided for different virtualization environments (VirtualBox, VMWare, KVM). Archivematica exposes REST APIs, the default access system for DIP is AtoM. Additionally, Archivematica allows export of DIP to DSpace format: with this configuration, Dspace can act as a dark archive for Archivematica, providing back-end preservation functionality while DSpace remains the user deposit and access system. Programmatic access to indexed AIP is available through Elasticsearch. Concerning the other OAIS information packages, SIP is based on METS,

while Library of Congress BagIt format (zip) is used for AIP. Archivematica supports not only DIP upload to AtoM, but also to CONTENTdm services (see Archivematica wiki). DIP upload can be achieved through a guided procedure on the GUI. Concerning metadata formats, as already mentioned METS is supported for both ingest and access, while PREMIS and DC are the reference standards for preservation and descriptive metadata.

For what concerns compliance to OAIS model, it has already been described that Archivematica implements a micro-services approach to digital preservation. The Archivematica micro-services are granular system tasks which operate on a conceptual entity that is equivalent to an OAIS information package. The physical structure of an information package will include files, checksums, logs, submission documentation, XML metadata, and others. These information packages are processed using a series of micro-services. Micro-services are provided by a combination of Archivematica Python scripts and one or more of the free, Open Source software tools bundled in the Archivematica system. Each micro-service results in a success or error state and the information package is processed accordingly by the next micro-service. There are a variety of mechanisms used to connect the various micro-services together into complex, custom workflows.

Preservation plans available for different media types, based on analysis of the significant characteristics of the files. The user dashboard provides interface mapped onto OAIS functional entities. The web dashboard allow users to process, monitor and control the Archivematica workflow processes. It is developed using Python-based Django MVC framework. The Dashboard provides a multi-user interface that will report on the status of system events and make it simpler to control and trigger specific micro-services. This interface allows users to easily add or edit metadata, coordinate AIP and DIP storage and provide preservation planning information.

Archivematica maintains the original format of all ingested files to support migration and emulation strategies. However, the primary preservation strategy is to normalize files to preservation and access formats upon ingest. Normalizing is the process of converting ingested digital objects to preservation and/or access formats. In Archivematica the original objects are always kept along with their normalized versions. Archivematica groups file formats into format policies (e.g. text, audio, video, raster image, vector image, etc.). Archivematica’s preservation formats must all be open standards. Additionally, the choice of formats is based on community best practices, availability of free and Open Source normalization tools, and an analysis of the significant characteristics for each media type. The choice of access formats is based largely on the ubiquity of web-based viewers for the file format. Not all files can be normalized on ingest because for example there are no available Linux-based Open Source tools to handle the conversions and/or no agreed upon preservation formats. In addition, some filetypes are not necessarily in the best preservation format but are still so ubiquitous and well- supported that they need not be normalized at the present time. In these cases, the files are kept in their original formats. A Format Policy Registry is available to implement rules of Preservation Planning.

Archivematica provides a full-fledged preservation platform which can be installed and used out-of-the-box. Extending Archivematica for integration with external components requires modification of the source code. Default storage mechanism is local file system.

3.4.1.3 Current and past assessment of the platform

Archivematica is currently maintained by a large community of users and developers, and is under investigation also in the project CoPs. The software is maintained and support is provided by Artefactual, the company which developed Archivematica. The project wiki contains links to the documentation and to the community forum, with discussions about issues and improvements. Information about release notes and a workplan is also provided. Further analysis of this resources is required during the second year to improve the evaluation of the solution. Moreover direct involvement of individuals from the community to get feedbacks about this assessment activity could be considered.

3.4.1.4 Assessment results

3.4.1.4.1 Functional suitability characteristics

Functional suitability is evaluated according to the list of functions reported in Section 3.5.

Completeness

According to the generic guidelines given in chapter 2 , the completeness is calculated following the measurement function:

$$X=(X_1+X_2*0.5+X_3*0.25)/1.75$$

where X_1, X_2, X_3 must be calculated considering respectively the percentage of functions available or not within the "*level of requirement":* mandatory, recommended and desirable.

Table below reports the identified functions with a column specifying whether the function is available or not.

Function	Level of requirement	Available
M1 - GUI ingestion	Mandatory	YES
M2 - Preservation of original content properties	Mandatory	YES
M3 - Support for AV formats	Mandatory	YES
M4 - Preservation Workflows Management	Mandatory	YES
M5 - Export of DIP	Mandatory	YES
M6 - Periodic integrity checks of the material and storing information in the AIP	Mandatory	YES
M7 - Format migration	Mandatory	YES
M8 - Ability to deal with large files	Mandatory	YES
M9 - Content quality control	Mandatory	YES

M10 - Virus check	Mandatory	YES
R1 - Batch ingestion	Recommended	NO (not easy mechanism found to submit bunch of prepared SIP files from command line, requires further investigation)
R2 - Support for METS	Recommended	YES
R3- Support for PREMIS	Recommended	YES
R4 - definition of requirements for restitution/playback	Recommended	NO
R5 - Extension with Add-ons and plugins	Recommended	YES
R6 - Usage Documentation	Recommended	YES
R5 - Dashboard for job monitoring	Recommended	YES
R6- Automatic extraction of technical metadata	Recommended	YES
R7 - User profiles and ACL	Recommended	NO (to be further investigated)
R8 - Creation of proxy copies (browsing quality)	Recommended	YES
R9- Multiple copies for redundancy	Recommended	NO (requires additional configuration)
D1 - Customize existing workflows	Desirable	NO (requires development and changes to the source code)
D2 – Export of DIP to different formats	Desirable	YES
D3 - Export of AV content fragments	Desirable	NO
D4 – Ability to integrate with alternative collection management systems	Desirable	YES
D5 – Populate and draw data and statistics from collection management systems	Desirable	NO

Table 19: Availability of functions for Archivematica

Computing the scores for functional completeness, we get:

$X_{1 =}$ 1 – (0/10) = 1
$X_{2 =}$ 1 – (4/9) = 0.556
$X_{3 =}$ 1 – (3/5) = 0.4

X = (1 + 0.556 * 0.5 + 0.4 * 0.25) / 1.75 = **0.787**

Appropriateness

In this assessment we assign a score for each function representing how well the functions are implemented and to what extent they satisfy the specified requirements.

For each function we provide a column with the associated score: 0 (not appropriate), 0.5 (partially appropriate) and 1 (completely appropriate).

Function	Level of requirement	Score
M1 - GUI ingestion	Mandatory	1
M2 - Preservation of original content properties	Mandatory	1
M3 - Support for AV formats	Mandatory	0.5 (requires more detailed tests)
M4 - Preservation Workflows Management	Mandatory	1
M5 - Export of DIP	Mandatory	1
M6 - Periodic integrity checks of the material and storing information in the AIP	Mandatory	1
M7 - Format migration	Mandatory	1
M8 - Ability to deal with large files	Mandatory	0.5 (requires further tests)
M9 - Content quality control	Mandatory	1
M10 - Virus check	Mandatory	1
R1 - Batch ingestion	Recommended	0 (not available, requires further tests)
R2 - Support for METS	Recommended	1
R3- Support for PREMIS	Recommended	1
R4 - definition of requirements for restitution/playback	Recommended	0
R5 - Extension with Add-ons and plugins	Recommended	1
R6 - Usage Documentation	Recommended	1
R5 - Dashboard for job monitoring	Recommended	1
R6- Automatic extraction of technical metadata	Recommended	1
R7 - User profiles and ACL	Recommended	0 (to be further investigated)
R8 - Creation of proxy copies (browsing quality)	Recommended	1

R9- Multiple copies for redundancy	Recommended	0 (requires additional configuration)
D1 - Customize existing workflows	Desirable	0 (requires development and changes to the source code)
D2 – Export of DIP to different formats	Desirable	1
D3 - Export of AV content fragments	Desirable	0
D4 – Ability to integrate with alternative collection management systems	Desirable	1
D5 – Populate and draw data and statistics from collection management systems	Desirable	0

Table 20: Appropriateness of functions for Archivematica

The final score of this sub-characteristic, using the function defined in section 3.5 is:

Z = 18/24 = 0.750

The resulting score of Functional suitability is:

FS = (0.787 + 0.750) / 2 = 0.768

3.4.1.4.2 Performance efficiency

As already mentioned in section 3, it was not possible to perform extensive tests of capacity during this initial assessment. A better definition of the scenarios as well as a dedicated testbed with users performing concurrent operations on the platform would be necessary.
At this point we can only refer to the documentation and reports for Archivematica, which provide evidence of usage with hundreds of jobs ingested and some users performing queries and access operations. This assessment is strongly dependent on the hardware resources used for running the platform.

3.4.1.4.3 Compatibility

No complete evaluation of this characteristic has been performed during this initial test. We listed some examples related to compatibility at a high-level and based on basic usage of the platform.

Co-existence

For what concerns Archivematica, the co-existence of other products sharing the same software and hardware resources could be affected by incompatibility related to

dependencies (different versions of a tool to be maintaned on the same system, e.g. ffmpeg), integrated services (e.g. the same database with possible issues for performances) or hardware resources (e.g. concurrent execution of resource consuming tasks, e.g transcoding of big files, in an environment with limited performances). On the hand the application container (Django, python-based) is built to support such co-existence, for example to manage different web applications with different contexts.
We can say that the co-existence with other software should be carefully evaluated: if the preservation of digital contents is a mission critical task within an institution, allocating dedicated resources for the preservation system could be wise. The preservation system could be configured in a specific way to reach other systems or information which should not be accessible to other applications and the creation of a protected environment could be necessary, preventing unauthorized access to other applications running on the same server.

Considering the 24 functions reported above, those which could be affected by the co-existence with other products are:

- M2, M6, M7, M8, M9, M10: these functions could involve resource consuming tasks, so their performance could be affected by other running applications
- R5: if Add-ons and plug-ins require dependencies which are in conflict with other running applications
- D4: communication with external systems could require network policies or other configurations in conflict with existing applications

Assuming that what stated above is true, we get the following score for co-existence:
X = 15/24 = 0.625

Interoperability

The level of interoperability is associated to the adoption of standard technologies and protocols, which allow the communication and exchange with other systems. Archivematica clearly demonstrates a huge effort from developers to adopt standards such as METS for content representation, while REST APIs are the main interfacing mechanism. Supported formats are selected among widely used ones and also the normalization tools are based on the best open source solutions which are adopted by a large majority of people in the field.

Therefore concerning interoperability, even if a quantitative measurement could be performed in the future, Archivematica should get good score.

3.4.1.4.4 Usability

Archivematica provides user and developer guides, with documented source code, as well as instructions for getting started and installation.

Operability

Concerning Archivematica, the user interface is stable and clearly demonstrates that the platform was conceived having the OAIS model in mind. The ingestion process is guided

and the user can follow all steps in the workflows, getting real time notifications about the status of jobs and accessing additional information. The python web GUI is fast and nice-looking, although some advanced operations are not clearly shown and it is necessary to read documentation for better usage. The user has access to detailed information about process results. No easy mechanism to upload SIP files edited using external tools is available. The platform provides also a table with the list of supported format and the normalization tools. Access is provided by external publication services based on Atom and ContentDM.

User error protection

Archivematica implements a micro-services pattern with different workflows available. The tasks are executed in the appropriate way to prevent errors and in case of failure they can be executed again. Using the GUI and the provided options seems to be safe in terms of user error protection, although further tests are required.

3.4.1.4.5 Reliability

No detailed evaluation of reliability has been performed in this initial assessment. Although we can consider Archivematica a mature solution with a high TRL, no actual tests have been performed so far. Such information could be retrieved by the community maintaining the platform, but a set of specific tests relevant for the project should be performed.

Concerning availability, the limitations depend on the underlying systems hosting the platform, the network and storage quality. No issues related to the platform implementation can be identified, they are rather related to the infrastructure used to deliver the platform.

Fault tolerance and reliability functionalities are available in Archivematica, although a better evaluation would require additional testing.

The overall evaluation for Archivematica in terms of TRL can be roughly estimated as **7 - 8**

Further operational tests are required to refine the TRL evaluation. At the moment the provided ranges should reflect the overall evaluation.

3.4.1.4.6 Security

Archivematica provides basic functionalities for what concerns user authentication and authorization. Further tests are required to assess the sub-characteristics associated to Security.

3.4.1.4.7 Maintainability

According to the definition, maintainability is the degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers. For software platforms, maintainability is one of the key factors behind the choice of adopting one

solution rather than an other. Any software solution needs to be maintained over time, to be kept up and running.

Modularity

Archivematica has been implemented using best practices for software development: the architecture includes separate components which are integrated in a coherent framework and are coordinated by a workflow manager. The documentation provides UML diagrams with the design of the platform, explaining the relationship between the Archivematica components and the OAIS entities.

The overall score for Archivematica in terms of modularity should be close to 1, but further analysis of the source code is required for better evaluation.

Analysability

Archivematica looks better in terms of notifications for error diagnosis, although no specific functionality seems to be available. Further analysis is required. We could assign 0.5, but further tests and future releases could improve the score.

Modifiability

The modular structure of Archivematica allows modifications (changes or updates) for the inner components. Archivematica code is written in Python and therefore allows hot deployment of the changes. It is worth noticing that since Archivematica depends on specific tools (e.g. ffmpeg) for executing tasks such as transcoding on AV contents, changing or upgrading these tools could affect the correct behaviour of the platform. The score for Archivematica should anyway be close to 1.

Testability

Archivematica provides documentation for executing tests and validation of the platform, although no test environment seems to be available. Therefore we can assign 0.5.

The overall score for Maintainability for Archivematica is:

Ma = (1+0.5+1+0.5) / 4 = **0.75**

Portability

Further tests are required for better assessment. Archivematica is built on top of open source technologies. The virtual machine with a default installation of the platform is available for VirtualBox and includes Ubuntu Desktop. The dependencies from low level binaries affect the possibility to install it on other operating systems.

In general, porting Archivematica to other Linux distributions should be feasible, although not supported. Migrating to Windows or Mac OS is not possible at the moment. Obviously using virtualization can always be a solution which can be supported by all operating systems. Archivematica provides virtual machines for VirtualBox, but migration to other virtualization environments such as VMWare or KVM should be straightforward.

Concerning hardware, no major constraint have been identified. Further tests are required to provide a quantitative measure of portability.

3.4.2 P4 – PrestoPRIME Preservation Platform

3.4.2.1 Platform installation

In order to assess the characteristics of P4, we installed the platform using the provided installer. No ready-to-use VM was available. The installer requires Ubuntu (64-bit server edition) and automatically retrieves from repositories the required dependencies. Also the configuration of the environment (folders, services, etc) is managed by the installer. As an alternative approach we checked out the code from the repository, built the binaries and replaced them in the installer, then repeated the installation. Both approaches produced a runnning instance of the platform. The two installations resulted in exactly the same software (although minor differences have been added in the last version with respect to the release contained in the installer.

3.4.2.2 Technical facts about P4

P4 is implemented in Java, a Servlet Container (such as Apache Tomcat) is required for execution. Data are stored in a native XML DB, so no RDMS is used. The data model makes use of METS as the main wrapper format for descriptive and technical metadata, as well as for mapping AV resources within the AIP. Other metadata standards are supported, such as MPEG-7 for technical metadata, PREMIS for preservation events, MPEG-21 for rights representation, DublinCore for descriptive metadata and others. P4 also supports DNX, a metadata format built on top of PREMIS vocabulary, used in Rosetta. Using P4 plug-ins, virtually any metadata standard can be used in the AIP.

Access interface supports also OAI-PMH protocol. The data model is tailored to broadcast environment (editorial entities, master and browsing qualities, B2B contracts). No compressed formats such as zip, BagIt or tarball used for AIP, METS contains references to metadata and AV files.

P4 develoment focused on videos, but other content types can be supported defining new workflows. Ingest and access are provided by a web UI or REST APIs, using METS as unique format for all OAIS information packages, common to other platforms. An advanced search engine based on Solr allows indexing of different descriptive and technical metadata. Several solutions are available for Archival Storage, supporting local and distributed storage. Preservation Planning is provided by integrated tools for fixity checks or format migration, no scheduler is implemented in the platform, makes use of external systems (e.g. iRODS). The index is stored in a fast native-XML DB and periodic triggers are executed for backup and integrity checks of the AIP XML files. Additional preservation operation are provided by storage solutions (e.g. the LTO component). Data Management and Administration are provided by the P4 web UI, including monitoring of jobs and workflows. The favorite integration mechanism is making use of REST interfaces over HTTP, to get loose coupling and reduce dependencies. P4 provides a plug-in mechanism to integrate external components or services in the workflow. In order to integrate cloud

services, a new storage plugin should be implemented and added to the storage layer (if we use REST APIs this should be straightforward).

3.4.2.3 Current and past assessment of the platform

P4 was developed during the PrestoPRIME project by EURIX and has been improved collecting feedbacks during several testbed events. P4 integrates several tools which will be tested in this project. The support to the platform development cannot be compared to Archivematica, although some components of the platform (e.g. the workflow engine) have already been used in other projects after PrestoPRIME and the improvements have been committed to the original code. The platform page contains information about the code, its usage and the integrated tools. EURIX still maintains the source code, upgrading the software to new releases of libraries and operating system.

3.4.2.4 Assessment results

3.4.2.4.1 Functional suitability characteristics

Functional suitability is evaluated according to the list of functions reported in Section 3.5.

Completeness

According to the generic guidelines given in chapter 2 , the completeness is calculated following the measurement function:

$$X=(X_1+X_2*0.5+X_3*0.25)/1.75$$

where X_1, X_2, X_3 must be calculated considering respectively the percentage of functions available or not within the "*level of requirement":* mandatory, recommended and desirable.

Table below reports the identified functions with a column specifying whether the function is available or not.

Function	Level of requirement	Available
M1 - GUI ingestion	Mandatory	YES
M2 - Preservation of original content properties	Mandatory	YES
M3 - Support for AV formats	Mandatory	YES
M4 - Preservation Workflows Management	Mandatory	YES
M5 - Export of DIP	Mandatory	YES

M6 - Periodic integrity checks of the material and storing information in the AIP	Mandatory	NO (currently the operation can be performed only manually by the platform admin, although the information is properly registered in the AIP using PREMIS events)
M7 - Format migration	Mandatory	NO (currently only migration to uncompressed MXF or AVI is supported, no format migration strategy implemented)
M8 - Ability to deal with large files	Mandatory	YES
M9 - Content quality control	Mandatory	YES
M10 - Virus check	Mandatory	NO (currently no virus check tool is provided, although it could be easily integrated in the ingestion workflow)
R1 - Batch ingestion	Recommended	YES
R2 - Support for METS	Recommended	YES
R3- Support for PREMIS	Recommended	YES
R4 - definition of requirements for restitution/playback	Recommended	NO
R5 - Extension with Add-ons and plugins	Recommended	YES
R6 - Usage Documentation	Recommended	YES
R5 - Dashboard for job monitoring	Recommended	YES
R6- Automatic extraction of technical metadata	Recommended	YES
R7 - User profiles and ACL	Recommended	YES
R8 - Creation of proxy copies (browsing quality)	Recommended	YES
R9- Multiple copies for redundancy	Recommended	YES (all available ingest workflows in the vanilla installation include double copies)
D1 - Customize existing workflows	Desirable	YES
D2 – Export of DIP to different formats	Desirable	YES
D3 - Export of AV	Desirable	YES

content fragments		
D4 – Ability to integrate with alternative collection management systems	Desirable	YES
D5 – Populate and draw data and statistics from collection management systems	Desirable	NO

Table 21: Availability of functions for P4

Computing the scores for functional completeness, for P4 we get:

$X_{1 =}$ 1 – (3/10) = 0.7
$X_{2 =}$ 1 – (1/9) = 0.889
$X_{3 =}$ 1 – (1/5) = 0.8

X = (0.7 + 0.889 * 0.5 + 0.8 * 0.25) / 1.75 = **0.768**

3.4.2.4.2 Appropriateness

In this assessment we assign a score for each function representing how well the functions are implemented and to what extent they satisfy the specified requirements.

For each function we provide a column with the associated score: 0 (not appropriate), 0.5 (partially appropriate) and 1 (completely appropriate).

Function	**Level of requirement**	**Score**
M1 - GUI ingestion	Mandatory	0.5 (limited feature)
M2 - Preservation of original content properties	Mandatory	1
M3 - Support for AV formats	Mandatory	0.5 (requires more detailed tests)
M4 - Preservation Workflows Management	Mandatory	0.5 (requires further development)
M5 - Export of DIP	Mandatory	1
M6 - Periodic integrity checks of the material and storing information	Mandatory	0 (only manual)

in the AIP		
M7 - Format migration	Mandatory	0 (limited features, requires further development)
M8 - Ability to deal with large files	Mandatory	1 (extensively tested in PrestoPRIME with huge master quality files)
M9 - Content quality control	Mandatory	1
M10 - Virus check	Mandatory	0 (not available)
R1 - Batch ingestion	Recommended	1
R2 - Support for METS	Recommended	1
R3- Support for PREMIS	Recommended	0.5 (limited to events)
R4 - definition of requirements for restitution/playback	Recommended	0
R5 - Extension with Add-ons and plugins	Recommended	1
R6 - Usage Documentation	Recommended	1
R5 - Dashboard for job monitoring	Recommended	0.5 (only provides monitoring of ingestion/migration/access tasks, no information about resource usage)
R6- Automatic extraction of technical metadata	Recommended	1
R7 - User profiles and ACL	Recommended	1
R8 - Creation of proxy copies (browsing quality)	Recommended	0.5 (not supported for all formats)
R9- Multiple copies for redundancy	Recommended	1
D1 - Customize existing workflows	Desirable	1
D2 – Export of DIP to different formats	Desirable	0.5 (limited feature)
D3 - Export of AV content fragments	Desirable	0.5 (only available through external tools)
D4 – Ability to integrate with	Desirable	1

alternative collection management systems		
D5 – Populate and draw data and statistics from collection management systems	Desirable	0

Table 22: Appropriateness of functions for P4

The final score of this sub-characteristic, using the function defined in section 3.5 is:

Z = 17/24 = 0.708

The resulting score of Functional suitability is:

FS = (0.768 + 0.708) / 2 = 0.738

3.4.2.4.3 Performance efficiency

As already mentioned in section 3, it was not possible to perform extensive tests of capacity during this initial assessment. A better definition of the scenarios as well as a dedicated testbed with users performing concurrent operations on the platforms would be necessary.

At this point we can only refer to the documentation and reports for P4, which provide evidence of usage with hundreds of jobs ingested and some users performing queries and access operations. This assessment is strongly dependent on the hardware resources used for running the platform.

3.4.2.4.4 Compatibility

No complete evaluation of this characteristic has been performed during this initial test. We listed some examples related to compatibility at a high-level and based on basic usage of the platform.

Co-existence

For what concerns P4, the co-existence of other products sharing the same software and hardware resources could be affected by incompatibility related to dependencies, integrated services or lack of hardware resources in an environment with limited performances. The application container (Apache Tomcat) is built to enable such co-existence, for example to manage different web applications with different contexts.

Considering the 24 functions reported above, those which could be affected by the co-existence with other products are:

- M2, M6, M7, M8, M9, M10: these functions could involve resource consuming tasks, so their performance could be affected by other running applications
- R5: if Add-ons and plug-ins require dependencies which are in conflict with other running applications
- D4: communication with external systems could require network policies or other configurations in conflict with existing applications

Assuming that what stated above is valid for P4, we get the following score for co-existence: **X = 15/24 = 0.625**

Interoperability

The level of interoperability of P4 (and in general of any preservation platform) is associated to the adoption of standard technologies and protocols, which allow the communication and exchange with external systems providing specific services (e.g. federated storage, collection management systems, etc.). P4 leverages standards such as METS for content representation, while REST APIs are the main interfacing mechanism.

Supported formats are selected among widely used ones and also the normalization tools are based on the best open source solutions which are adopted by a large majority of people in the field. A huge effort was performed in PrestoPRIME to identify the best formats and standards for digital preservation and P4 integrates the achievements of the project.

Therefore concerning interoperability, even if a quantitative measurement could be performed in the future, P4 should get a good score.

3.4.2.4.5 Usability

P4 provides user and developer guides, with documented source code, instructions for getting started and an installer which manages the whole configuration and setup starting from a clean installation of Ubuntu server.

Operability

Concerning P4, the web GUI makes use of Java technology and is quite stable. The same GUI can be used to manage different instances of the platform, using the same user with different identifiers. Different strategies for the ingestion are available and the workflows can be fully customized and published in the GUI dashboard. Real time information about tasks is provided, although the details about task execution should be improved. The GUI is logically divided in different sections mapped onto OAIS entities and different views are provided according to the user profile. The access interface provides a player to consume the access copy, a detailed description of the DIP based on information in the METS and the possibility to export the whole file or just a fragment selected from the video shots.

User error protection

P4 provides great extensibility and the possibility to define and upload custom workflows. The definition of new tasks is straightforward and is documented in the developer guide. P4 uses a native-XML DB with a web interface and uploading of new configurations is an easy task. The only drawback is that these workflows could contain errors or inconsistencies, which should be verified by the user before deploying them. At the moment a simple check on input and output parameters for each task is available, although this should be improved. There is no functionality for execution of limited parts in a workflow for partially failed tasks.

3.4.2.4.6 Reliability

No detailed evaluation of reliability has been performed in this initial assessment. We can say that compared to other solutions P4 is a far less mature solution and this is also due to the shorter lifetime and the limited community of developers maintaining it.

Concerning availability, the limitations are mainy due to the underlying systems hosting the platform, to the network and storage performances. No issues related to the platform implementation itself can be identified, they are rather related to the infrastructure used to deliver the platform. The application container is Apache Tomcat, which is the reference implementation for servlet container, it is widely used in a huge amount of projects and commercial solutions and is supported by a large community of developers.

Fault tolerance and reliability are not guaranteed in P4, due to the prototype status of the solutions.

The overall evaluation for P4 in terms of TRL can be roughly estimated as **5 – 6.**

Further operational tests are required to refine the TRL evaluation. At the moment the provided ranges should reflect the overall evaluation.

3.4.2.4.7 Security

P4 provides basic functionalities for what concerns user authentication and authorization. In addition, P4 associates to each user a unique identifier which is used to retrieve information about user role and permissions. The same instance of P4 GUI can be used to access and manage different instances of P4 server, with different roles and permissions simultaneously, using the mechanism of unique ID mentioned above. Further details can be found in the P4 documentation.

Further tests are required to assess the sub-characteristics associated to Security.

3.4.2.4.8 Maintainability

Also for P4 maintainability is one of the key factors for evaluation, to guarantee software maintenance and continuous support. P4 provides detailed technical specifications and the

source code is fully available, this enables the improvement of the platform by developers and early adopters.

Modularity

P4 architecture are based on separate components (user interface, REST web server and core components implementing OAIS) which are integrated in a coherent framework and are coordinated by a lightweight workflow manager.

The overall score should be close to 1, but further analysis of the source code is required for better evaluation.

Analysability

P4 does not provide any functionality related to error diagnosis is limited to parsing manually log files. Further analysis is required. We could assign 0, but further tests are required and mainly the evaluation could change dramatically if new releases will be available.

Modifiability

P4 is built with a modular architecture, it provides a bunch of Java archive files for each subcomponent and replacing or updating one of them simply requires building and deploying a new version. Obviously some constraints must be respected, such as keeping interfaces and input/output format for exchanged data. It is worth noticing that since P4 depends on specific tools (e.g. ffmpeg) for executing specific tasks on contents, changing or upgrading these tools could affect the correct behaviour of the platform. P4 includes the sources for all used tools, which are built during installation, rather than retrieving the binaries from repositories: this allows better control of dependencies especially due to the prototype status of the platform. The score should be close to 1.

Testability

P4 does not provide such functionality yet. Since further tests are required, we can assign 0 and change the score in the future after further tests.

The overall score for Maintainability is:

Ma = (1+0+1+0) / 4 = **0.5**

3.4.2.4.9 Portability

Further tests are required for better assessment. P4 makes use of open source technologies but the installer is only available for Ubuntu Server OS.

Even if P4 is implemented in Java, so portability should be guaranteed, the dependencies from low level binaries affect the possibility to install it on other operating systems.

In general, porting P4 to other Linux distributions should be feasible, although not supported. Migrating to Windows or Mac OS is not possible at the moment and virtualization is the only way to be supported by all operating systems.

P4 provides pre-built virtual machines for VirtualBox, but migration to other virtualization environments such as VMWare or KVM should be straightforward. Concerning hardware, no major constraint have been identified. Further tests are required to provide a quantitative measure of portability.

4 Presto4U Dataset

The test dataset in Presto4U was created after considering the requirements from different CoPs collected as part of the WP2 requirements gathering task [4]. The formalism for such a dataset collection exercise is specified in the Presto4U Knowledge Schema.

The sourced dataset descriptions are as follows:

Art & museum objects, artists and their representatives (TATE)	
Dataset description	A collection of compressed and uncompressed video files in 4 different standards. The files are encoded using a Quicktime codec and the variations are: 1. 1x h264,1080p 25fps mov file 2. 1 uncompressed 8bit NTSC 30fps mov file 3. 1x uncompressed 10 bit PAL mov file 4. 1x ProRes422 1080 25fps mov Each file is 3 minutes long, with an initial 5 second slate stating the details of the source file, 55 seconds of colour bars and tone and then 2 minutes of a bicycle wheel turning and then quick movements of the camera.
Dataset Author	TATE -- http://www.tate.org.uk/
Creation process	The source video was shot using a Cannon EOS 550D DSLR camera, which natively records H264 files, at 1080p, 25fps and 16:9 aspect ratio. The video is recorded as progressive onto an SD card. The H264 file was then cloned to a mac computer for editing.
Copyright and licence	Cleared for research purposes within Presto4U. Creative Commons Licence.

Learning and Teaching Repositories (UIBK) University of Innsbruck	
Dataset description	A collection of MP4 files with the following characteristics: Resolution: 720x576, preset average video Bitrate: 5600 kbps (mpg4v), preset average audio bitrate: 192 kbps (AAC); for effective bitrates please see the technical metadata contained in the files themselves (e.g. with ffprobe or mediainfo, or codec properties in VLC player).
Dataset Author	University of Innsbruck -- http://www.uibk.ac.at/index.html.en
Creation process	MP4 is the master format chosen in this case. This format was chosen because the visual quality was sufficient and subjectively indistinguishable from the original. Also they did not need to use an uncompressed/lossless format because of the costs that this would have entailed (storage, difficulty with editing, etc.). So, preservation for their academic team is often a matter of doing it with reasonable,

	usable quality instead of not doing it at all (which could mean a complete loss of the material).
Copyright and licence	Cleared for research purposes within Presto4U.

Broadcasting (RAI)	
Dataset description	A collection of MXF files both in SD and HD quality: 1. SDTV MXF/D10 samples 8 files, total duration around 13 min, total size around 6 GB 2. HDTV MXF/HD422@25p samples 12 files, total duration around 14 min, total size around 6 GB and their H264 proxy version in CIF resolution, at 2Mbps, about 300 MB total size. 3. Media Contracts for rights format/technologies: to be completed in 2014
Access	Radiotelevisione italiana S.p.A. -- http://www.rai.it/
Creation process	SD Resulting from digitisation of Betacam tapes,hold by RAI Archive, by means of IMX player with eVTR option, subsequently verified and cut with software tools. HD Resulting from downscaling from 4K captured material by RAI
Copyright and licence	Copyright RAI. Provisionally made available for research purposes to Presto4U partners, until end of 2014. Waiting for definition of a common license for Presto4U data set to be issued.

Footage Sales Libraries (Cinecittà)	
Dataset description	The test dataset provided by Cinecittà Luce is made of 5 files representing the same footage clip in different formats and resolutions (720p and 1080p). A storage format DPX representation of this clip has been also provided, which is made of a series of files (in DPX every frame is stored in a single file). For Luce sales dept. (and for most of the footage sales archives) the formats used to deliver content, are the formats requested by the client. Anyway, Luce provided the two formats that statistically cover the 85% of the requests it receives: 1. AppleProRes 4:2:2 (720p/1080p) 2. H.264 4:2:2 (720p/1080p) For archiving films (35 and 16mm) Luce scans them in 2K and store them in a still frames storage format in its digital tape library. The format used is the above mentioned ANSI/SMPTE standard: Digital Picture Exchange - DPX (2K) For the audio part of the film Luce uses the uncompressed audio format: LPCM in a WAV container.
Access	Cinecittà Luce -- http://www.cinecitta.com/
Creation process	Luce's master digital format is the DPX format, which is a storage format; when Luce have to send footage to clients it usually convert this format into one of the above mentioned "sales formats" (AppleProRes or H.264).

	For screening purposes only, Luce projects film material in a Digital Cinema environment, packaging the content in DCP (Digital Cinema Package) format, which means basically wrapping in MXF files the video essences in JPEG 2000 @2K and audio essences in LPCM. But this conversion process happens rarely at the moment, so DCP should not be consider a commonly used format.
Copyright and licence	Copyrighted by Istituto Luce Cinecittà and its use is allowed only for non-commercial purposes and only in the Presto4u project framework.

Research and Scientific Collections (CNR)	
Dataset description	20 videos coming from the Blip dataset used by the MediaEval Benchmarking Initiative for Multimedia Evaluation (member of the Research and Scientific Collections CoP). Videos are stored and encoded in Ogg Vorbis video format. MediaEval is a benchmarking initiative dedicated to evaluating new algorithms for multimedia access and retrieval. It emphasizes the 'multi' in multimedia and focuses on human and social aspects of multimedia tasks. MediaEval attracts participants who are interested in multimodal approaches to multimedia involving, e.g., speech recognition, multimedia content analysis, music and audio analysis, user-contributed information (tags, tweets), viewer affective response, social networks, temporal and geo-coordinates. Uploaded videos for Presto4U have been randomly selected from the test subset of the Blip10000: A social Video Dataset containing SPUG Content for Tagging and Retrieval dataset containing comprehensive semi-professional user-generated (SPUG) content, including audiovisual content, user-contributed metadata, automatic speech recognition transcripts, automatic shot boundary files, and social information for multiple 'social levels'. Videos have been taken from blip.tv and have a creative common license (unfortunately, I cannot be more precise). The dataset has been used for scientific experiments and there preservation is very important for comparison of future research with nowadays state of the art. Moreover, for the scope of benchmarking, lot of metadata and information have been collected for the videos in the datasets. More information about the dataset can be found in the following paper: http://dl.acm.org/citation.cfm?doid=2483977.2483988 The MediaEval initiative website instead is: http://www.multimediaeval.org/about/
Access	MediaEval -- http://www.multimediaeval.org
Creation process	Videos have been selected videos for research purpose by the MediaEval initiative between the Blip.
Copyright and licence	Copyright free. Cleared for research purposes within Presto4U.

The datasets collected in year 1 was approximately 50 GB in size. We will continue to work on sourcing the dataset in year 2 with the goal of publishing a publicly available dataset which can be used as a test set for the assessment of AV preservation tools.

5 Conclusion and Future Work

In this report we have presented the work done in year 1 with regards to the research output assessment exercise carried out as defined in WP3 Task 3.2 ‘Preservation Research Technology Watch and Assessment’. Both quantitative and qualitative analysis of the tools was performed against a series of criteria and metrics which measured characteristics such as functionality, east of installation, robustness, performance, and scalability. During the course of the first year, we have also sourced the first version of the Presto4U dataset from representative CoP organisations. The purpose of this dataset is twofold. Firstly to act as a test dataset for the assessment of research outputs carried out internally within the project, and secondly the aim make a publicly available dataset at the end of the project which can be used for testing tools outside the project in the AV preservation domain.

In the first part of the deliverable, we have focussed on the methodology used during the acquisition of information regarding new and upcoming research outputs. We have developed a web framework (PrestoKAT) which allows the collation of all such information in a machine understandable format. This knowledge base of research outputs will be used for brokering between the requirements from the CoPs and solution providers offering tools. Next, we have taken the assessment methodology defined in D3.1 [1] and have made a first attempt at implementing this methodology with the tools chosen for assessment in year 1. It is worth mentioning again that both the definition of a standardised methodology for assessment of AV preservation tools and the actual implementation and analysis carried out on the ROs has not been attempted before in literature. In chapter 3 we have presented the full results of the assessment exercise. Although we have defined measurement functions and a way to quantify the results for each specific category of tools, it may be necessary to further refine these functions in year 2 (and also re-evaluate some of the tools). The results from year 1 have shown that it is possible to carry out RO assessment based on the methodology proposed in Task 3.1 of WP3. Although, the tools chosen were based on direct interactions with the CoPs, in year 2 we expect to have much more concrete results in terms of assessment. This is because the needs of the CoPs are now being formally recorded based on the knowledge schema using questionnaires. RO analysis will be much more streamlined when we have recorded these requirements and are able to extract semantic meaning out of them. Finally, we have provided details of the year 1 dataset in Chapter 4

In terms of future work: We have started the analysis of the quantitative results outputted from first year and plan to test more tools during year 2 to cover all the broad tool categories as defined in Chapter 1 Based on the analysis and the lessons learnt from the first assessment exercise, we will further refine the assessment methodology. Also, we are currently in the process of building a storage mechanism for the assessment results. This will be linked to the RO tool database which has been created for PrestoKAT. Once each RO has associated assessment results readily available, the output of this task will act as input towards mapping and brokering CoP requirements to tool features being undertaken in WP4 (particularly task 4.3 ‘Brokering Technologies to Communities of Practice and Suppliers).

Glossary

Term	Definition
CoP	Community of Practice
RO	Research Output
LTO	Linear Tape Open
LTFS	Linear Tape File System
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
JSON	JavaScript Object Notation
XML	Extensible Markup Language
MD5	Message Digest algorithm 5. An hash function used to check data integrity.
SHA1	Secure hash algorithm 1. An hash function used to check data integrity

References

[1] Presto4U Consortium, “D3.1: Specification of Assessment Criteria, Metrics, Processes, Datasets and Facilities,” 2013.

[2] Presto4U Consortium, “Description of Work: "European Technology for Digital Media Preservation",” Grant agreement no: 600845, 2011.

[3] The Consultative Committe for Space Data Systems, “Reference Model for an Open Archival Information System (OAIS),” Magneta Book, 2012.

[4] Presto4U Consortium, “D2.1: CoP Knowledge Schema,” 2013.

[5] European Space Agency, “Technology Readiness Levels Handbook for Space Applications,” 2008.

[6] W. B. M. Höffernig, “Formal Metadata Semantics for Interoperability in the Audiovisual Media Production Process,” in *Workshop on Semantic Multimedia Database Technologies (SeMuDaTe)*, Graz, AT, 2009.

[7] M. H. K. M. G. N. W. W. W. Bailer, “Final version of metadata conversion and deployment services, PrestoPRIME Deliverable ID4.0.4a,” 2012.

[8] EBUCore, “EBU Core Metadata Set v1.4,” EBU Tech 3293, 2012.

[9] PrestoPRIME Consortium, “LTFS Archiver,” 2012. [Online]. Available: http://www.crit.rai.it/EN/attivita/opensource/LTFSArchiver_en.pdf. [Accessed 18 12 2013].

[10] M. H.-M. L. B. R. B. A. H. S.C. Phillips, “Second version of prototype for distributing and storing audiovisual content using federated storage and computer services,” PrestoPRIME deliverable ID3.4.4, 2012.

[11] W. Bailer, “D4.0.2 Annex A: Non-blocking service calls,” [Online]. Available: https://prestoprimews.ina.fr/public/deliverables/PP_WP4_ID4.0.2e_Annex_A_v1.01.pdf [Accessed 18 12 2013].

Document information

Delivery Type	Report
Deliverable Number	3.2
Deliverable Title	Research Outputs Assessments v1
Due Date	31 December 2013
Submission Date	18 December 2013
Work Package	WP3 - Identification and Analysis of Research Outcomes
Partners	IT Innovation, Beeld en Geluid, INA, RAI, MM, CNR, EURIX, JRS
Author(s)	Bailer Werner, Biscoglio Isabella, Boch Laurent, Borgotallo Roberto, Chakravarthy Ajay, Foster Simon, Fabrizio Falchi, Gallo Francesco, Ligios Linda, Laurenson Pip, Marchetti Eda, Melas Panos, Rendina Marco, Walland Paul
Keywords	Assessment, digital preservation, research output
Document Identifier	Deliverable_D3.2_presto4u_21_12_2013_V1(R).pdf
Dissemination level	PU=Public or RE=Restricted to a group specified by the consortium (including the Commission services)
Document Status	Released

Project Acronym	Presto4U
Project Full Title	European Technology for Digital Audiovisual Media Preservation
Grant Agreement	600845
Project Coordinator	Beeld en Geluid
Contact Details	Sumatralaan 45, 1217GP Hilversum, The Netherlands. msnyders@beeldengeluid.nl

Document Status Sheet

Version	Delivery Date	Comment	Author
0.1	Oct 29 2013	First outline created	Ajay Chakravarthy
0.2	Nov 19 2013	Updated TOC based on WP3 discussions	Ajay Chakravarthy
0.3	Nov 21 2013	Updated dataset details	CoP representatives
0.4	Nov 21 2013	Defined implementation plans based on assessment criteria	Eda Marchetti, Isabella Biscoglio and Roberto Borgotallo.
0.6	Nov 26 2013	Tool descriptions	WP3 partners

		for RO identification section	
0.7	Nov 27 2013	Updates to measurement plans for specific category of tools	Eda Marchetti, Isabella Biscoglio
0.8	Nov 29 2013	MServe and LTFS Archiver evaluation results	Panos Melas and Laurent Boch
0.8a	Dec 10 2013	Updates to measurement plan for metadata mapping tools and preservation tools and platforms	WP3 partners
0.9	Dec 15th 2013	Updated assessment results for all ROs	WP3 partners coordinated by Panos
1.0	Dec 18th 2013	Added scope, exec summary and performed a full review of the document and made several formatting and content updates	Ajay Chakravarthy
1.0R	Dec 21th 2013	Final version for release	Beeld en Geluid