




Gap analysis on action services tools and SCAPE platform and testbeds requirements

Authors

Miguel Ferreira, Hélder Silva, Rui Castro (KEEP Solutions), Per Møldrup-Dalum (The State & University Library Aarhus), Zeynep Pehlivan (Université Pierre et Marie Curie), Carl Wilson (Open Planets Foundation), Sven Schlarb (Austrian National Library)

January 2013

This work was partially supported by the SCAPE Project. The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).

This work is licensed under a CC-BY-SA International License 



Executive Summary

This document presents a gap analysis on the current state of development of preservation components in SCAPE against the SCAPE Platform and the Testbed scenarios. In this analysis, the functionality of existing preservation components is compared to SCAPE Testbed scenario requirements and published as a Gap Analysis, which will serve as the basis for tool development in the Preservation Components Sub-project during the next phase of the project.

The approach to the gap analysis consisted of the following steps:

- 1) Identify requirements;
- 2) Define levels of compliance;
- 3) Define levels of importance;
- 4) Define where we want to be;
- 5) Determine where we are (level of compliance);
- 6) Propose actions to improve the current state.

After applying the gap analysis methodology we have concluded that none of the tools have yet met the desired level of compliance specified in this analysis. There are 11 scenario requirements that have not been addressed yet, meaning that no tools have been identified as possible solutions to solving those particular requirements. In such cases brand new tools will have to be developed.

Apart from the scenarios for which there are no tools available yet, there are 13 additional scenario requirements for which a considerable amount of effort is required in order to produce a tool capable of tackling it in an appropriate manner.

The final result of this analysis is a set of prioritised actions for each work package within the Sub-project that define the development roadmap for the coming year.

Table of Contents

1	Introduction.....	1
1.1	Approach.....	2
2	Gap analysis.....	3
2.1	Requirements for the gap analysis.....	3
2.2	LSDRT Scenarios.....	4
2.2.1	Assessing preservation risks in large media files (LSDRT 1).....	4
2.2.2	Validating files migrated from TIFF to JPEG2000 (LSDRT 2).....	5
2.2.3	Validating migrated images “visually” (LSDRT 3).....	6
2.2.4	Out-of-sync sound and video in WMV to Video Format-X migration (LSDRT 4).....	7
2.2.5	Detecting audio files with very bad sound quality (LSDRT 5).....	7
2.2.6	Large scale migration from mp3 to wav (LSDRT 6).....	7
2.2.7	Characterise very large video files (LSDRT 7).....	8
2.2.8	Characterisation of large amounts of wav audio (LSDRT 9).....	8
2.2.9	Capturing Representation Information from original image files (LSDRT 10).....	9
2.2.10	Duplicate image detection within one book (LSDRT 11).....	9
2.2.11	Quality assurance in redownload workflows of digitised books (LSDRT 12).....	10
2.2.12	Potential bit rot in image files that were stored on CD (LSDRT 13).....	11
2.3	Research Dataset Scenarios.....	12
2.3.1	General scientific data handling scenario (RDST 1).....	12
2.4	Web Content Scenarios.....	13
2.4.1	Comparison of Web Archive pages (WCT 1).....	13
2.4.2	ARC to WARC migration (WCT 2).....	14
2.4.3	Characterise web content in ARC and WARC containers (WCT 3).....	14
2.4.4	(W)ARC to HBase migration (WCT 6).....	15
2.5	Levels of compliance.....	16
2.6	Levels of importance.....	16
2.7	Compliance threshold.....	17
2.8	Gap analysis.....	17
3	Analysis.....	25



3.1	Complete list of scenario requirements and development priorities	28
3.2	Scenario requirements with no tools.....	29
3.3	Scenario requirements with need of attention	29
3.4	Per work package analysis	30
4	Actions to implement	32
5	Conclusions.....	34
6	References.....	34

1 Introduction

The Preservation Components Sub-project addresses three known limitations of components in digital preservation, namely: **scalability**, **functional coverage** and **quality**. The Sub-project aims to improve existing digital preservation tools, develop new ones where necessary, and apply proven approaches to the problem of ensuring quality in digital preservation.

The Sub-project is divided into three work packages, each of which is dealing with one type of preservation component:

1. **Characterisation components** are able to detect and extract a limited number of aspects from digital objects that describe the technological properties of those objects. Typically, these components operate locally (on a single machine) and at small scale. The SCAPE project aims at enabling scalable characterisation on large-sized collections of items.
2. **Action components** are responsible for ensuring the long-term access to digital objects and their content. Current actions are not all able to cope with large-size collections. SCAPE is focusing on the applicability of such components to large collections of complex digital objects, by focusing on analysing and improving the interfaces and internal functionality, extending and creating new preservation functionality and enabling tools to deal with complex objects.
3. **Quality assurance tools** provide automated methods for assessing the outcome of preservation actions. Until now, quality assurance has mostly relied on a combination of human intervention and sampling methods. However, in order to meet the challenges posed by very large and heterogeneous collections, one must develop automatic quality assurance approaches based on complex aggregate functions that capture many of the properties of an object and assess them by means of comparison metrics.

One of the limitations of existing preservation components is that they have not been specifically designed to operate in large-scale scenarios. This can be seen both in the way tools are built (e.g. no parallelisation, not multithreaded, etc.), as well as in a poor interoperability in the sense that it is difficult to embed them in automated preservation workflows (service integration, normalised outputs etc.).

A second limitation of current tools is functional and includes, for instance, the ability to capture relevant provenance, contextual information and metadata, as well as their ability to provide usable outputs for automated policy-driven preservation. Semantics are also an important factor, especially when numeric data sets are migrated. This limitation is a real obstacle to full automation and, in consequence, might hinder the scalability of digital preservation processes.

In what concerns quality assurance, SCAPE will investigate methods to automatically detect quality faults, based on the conversion of objects to comparable derivatives and then apply techniques in order to detect differences introduced by preservation actions. The approach is to develop workflows for quality assurance, with interfaces between the various tools used.

The Preservation Components Sub-project maintains close interaction with the Planning and Watch Sub-project, with characterisation components feeding into the planning system. It also maintains a



close relationship with the Platform Sub-project where preservation action components are expected to be deployed in order to execute preservation plans on a large scale. A close collaboration with the Testbeds Sub-project is also established in order to define the focus and the scope of the components as well as for testing their outcome and performance against real data.

From a supplier-consumer perspective, one might say that Preservation Components is a supplier of both the Testbeds and 'Planning and Watch' Sub-projects. Moreover, it should be able to comply with the requirements laid out by these Sub-projects while, at the same time, being compatible with the functional requirements of the SCAPE execution platform.

This report aims at identifying current gaps between preservation components that have been collected or developed during the SCAPE project and the real requirements of the Testbed scenarios in terms of file format coverage, functionalities and performance. The gap analysis will also report on how the tools meet the requirements of the SCAPE Platform in terms of operating system, deployment mechanism, end-user license agreement, etc. The goal of the gap analysis is to determine "where we are" in terms of fulfilling these requirements and "where we want to be". This serves the basis of understanding what we are missing and what we need to do to "bridge the gap".

Originally, as depicted in the project's description of work, this document was intended to report solely on the gaps that exist between preservation action components and the Testbed scenarios requirements. However, during the course of the project, it was felt that the same methodology and analysis should also be applied to the rest of components in the Preservation Components Sub-project, namely: characterisation and quality assurance components. The results of this analysis will serve as the basis for tool development during the following stages of the SCAPE project.

1.1 Approach

In our understanding, gap analysis means to take a critical look at what is the current status of a given setting and comparing this with a desired setting. In order to make any improvements to the current state of the art, one must understand what the current setting conditions are and which conditions are we willing to meet. Gap analysis can also be used as a means for classification of how well a setting complies with a predefined set of requirements.

The first step in performing a gap analysis is therefore to define where we want to be, followed by an analysis of where we are (Bowen & McDonough, 2010). After lining out the "where we want to be" and comparing it against the "where we are", we can start to have a closer look at the different steps that one needs to take in order to accomplish the desired effects. By doing a gap analysis one can see areas where our current setting is not particularly strong, and what actions need to be implemented in order to move our setting to the desired state (Bowen & McDonough, 2010).

The approach to this gap analysis consists of the following steps:

1. **Identify requirements** – identify the set of requirements against which we want to evaluate the current status of our setting; Requirements were drawn by the "consumers" of Preservation Components, i.e. Testbeds;



2. **Define levels of compliance** – for each requirement identified previously we must assess our maturity/ability to fulfil that requirement. This step identifies the SCAPE Preservation Component specific scale used in this classification, and is presented in section 2.5;
3. **Define levels of importance** – each requirement should be weighted in terms of its importance to the overall success of the project. This classification enables the prioritisation of actions needed to bring the current setting to the desired state (section 2.6);
4. **Define where we want to be** – this step consists of the definition of the compliance threshold we wanted to meet. If our current setting is below the defined threshold, actions should be defined to improve the setting to the desired level of compliance;
5. **Determine where we are** – this step consists of determining the level of compliance of our setting against each of the given criteria. The current state was determined according to the levels of compliance previously devised in step 2;
6. **Propose actions to improve the current state** – whenever our setting was below the threshold of compliance, actions were defined to move the current status to a level above the defined threshold. Actions are prioritised according to the levels of importance previously defined on step 3.

2 Gap analysis

2.1 Requirements for the gap analysis

The requirements that constitute this gap analysis were extracted from the current snapshot of the of the Testbeds scenario descriptions. The Testbeds scenarios describe specific preservation issues that will drive the development and evaluation of a number of key outputs from the SCAPE Project. During the first half of the SCAPE project, the focus of these pages is primarily on the development of Preservation Components. Over time, scenarios will be expanded to describe developments in Preservation Planning and Watch and the underlying infrastructure provided by the SCAPE Platform.

SCAPE Testbed scenarios are defined as triples of the following concepts: a dataset, a preservation issue and a possible solution. Scenarios have been organized according to 3 types (or Testbeds):

- 1) Large Scale Digital Repositories Testbed (LSDRT)
- 2) Research datasets Testbed (RDST)
- 3) Web content Scenarios Testbed (WCT)

The following sections provide a brief description of the Testbeds scenarios. This aims at crystallizing the current state of the scenarios as they are in constant revision and evolution during the course of the project. From these descriptions, we have extracted the requirements as the basis for our gap analysis.

Among the existing SCAPE scenarios, there were some scenarios that were not included in this gap analysis. The reason for this was three-fold:

- 1) The scenario did not appear to have a dependency on a preservation component (PC);

- 2) The scenario description was too vague and key parts of it were still under development, or
- 3) The scenario depicted an issue already portrayed, but applied to a different dataset of the same type, meaning that no additional components were necessary to solve the issue depicted in the scenario.

A list of the scenarios that were not considered in this gap analysis is presented in Table 1.

Table 1 - List of scenarios not included in the gap analysis.

Ref.	Scenario name	Reason for not being included
RDST 2	Format Migration of (raw) Scientific Datasets	The scenario RDST 2 is based on the same dataset and includes the same issues of scenario RDST 1.
RDST 3	Maintaining understandability and usability of raw data through external resources	Does not depend on Preservation Components.
RDST 4	Preserving the value of raw data and verifiability of processed datasets forming part of a scientific workflow	Does not depend on Preservation Components.
WCT 4	Web Archive Mime-Type detection	The preservation issue associated to scenario WCT 4 is the same as WCT 3. The dataset is a different, but of the same type (i.e. web content). It is not expected that a different set of tools would be necessary to solve the requirements of this scenario.
WCT 7	Format obsolescence detection	The scenario is still under development. At the time of writing, there were no concrete issues identified.
WCT 8	Huge text file analysis using Hadoop	Does not depend on Preservation Components

2.2 LSDRT Scenarios

The scenarios published on the Open Planets Foundation Wiki are under permanent improvement. The following sections represent a “snapshot” of the scenarios published at <http://wiki.opf-labs.org/display/SP/Scenarios> on 2012-11-02. Slight text enhancements have been introduced to the original text to make it more readable in the context of this report.

2.2.1 Assessing preservation risks in large media files (LSDRT 1)

At the Statsbiblioteket (SB), data from broadcasters contain huge media files like MPEG2 transport streams (MPEG2-TS). There is an end user agreement that only allows streaming this data, but not distribution of copies of the archived content. SB captures broadcast television as complex MPEG2-TS. The video content is accompanied by metadata, typically used to support the production of TV guides. SB preserves the MPEG2-TS as the preservation masters. Chunks of this data that relate to specific programmes are extracted, migrated and served to users as streaming Flash video. The master MPEG2-TS files are so large that characterisation is a significant challenge.

The difficulty lies in pulling out metadata for these huge media files in a large scale. Deep characterisation, in this context, means that for container formats the contained streams (typically mpeg-2 or mpeg-4 (h.264) video and AAC audio) are also identified and characterised.

It is difficult to apply typical validation tools to such large files. A detailed characterisation of the MPEG2-TS is needed in order to identify technical dependencies for extracting from or rendering the embedded content in the MPEG2-TS. This would enable preservation risks related to current access services to be monitored and action taken as necessary to ensure continued access and preservation.

2.2.1.1 In this scenario, specific component requirements include:

- Tools capable of doing deep characterization of large video files in MPEG2-TS format.
- Tools are expected to be able to identify the format and extract relevant properties from the data streams transported inside the container formats.

2.2.1.2 In this scenario, success criteria include:

- Being able to extract provided metadata:
 - The technical metadata, which is used by the player machines to decode the stream
 - The program metadata that is used to display program and channel information
 - The subtitles, which to some extent is a full text dump of the program content (
 - Teletext information
- Being able to process streams faster than their defined bitrate.

2.2.2 Validating files migrated from TIFF to JPEG2000 (LSDRT 2)

An important part of digital preservation is the willingness and financial commitment of a memory institution to preserve the data for the long term. Given the time scales in question any cost saving is to be welcomed. At the British Library (BL), as in many other institutions, the cost of storing uncompressed TIFFs currently outweighs the risk of replacing these images with a (perhaps) compressed format¹.

Migration to JPEG2000 can be problematic - both because of the interpretation of the standard and also because migration tools may fail mid-process. We need a post-migration quality assurance tool that can validate a JPEG2000 against the original TIFFs.

2.2.2.1 In this scenario, specific component requirements include:

- A tool that can migrate a TIFF to JPEG2000 in a consistent and preservation-safe fashion, maintaining (or normalizing) the embedded ICC profile, resolution headers and other significant metadata.
- A tool capable of verifying in an automated fashion that the migrated files are complete (i.e. have not been arbitrarily truncated) and that the files are valid and/or will render in one or more common viewing applications without error.

¹ As a side benefit, replacing the TIFF images with alternative representations will facilitate access to the materials - smaller files to manipulate and download and native tool support in browsers and standard OSs.

- A tool capable of identifying what metadata properties have not been correctly converted to the new format.

2.2.2.2 In this scenario, success criteria include:

- Large-scale digitisation projects need to check in content and verify its compliance to a profile quickly and efficiently despite the high volume of data. For example, JPEG2000s digitised for a current BL project will be received at between 0.25 and 0.5TB per day. Checking must be performed at a sufficient rate to prevent a build-up of material and allow timely rejection of content that does not match the profile (problem pages can be re-digitised if issues are identified in a timely manner: i.e. within days rather than weeks)

2.2.3 Validating migrated images “visually” (LSDRT 3)

Some forms of content arrive at the preserving institution and will be preserved "as is" regardless of how the files have been constructed (e.g. web archived content). Other content can be acquired under a specific agreement with the creator or publisher, and the preserving institution typically expects the content in a particular form. This may go further than describing formats used, and will actually describe specific technical constraints on the construction of the files. For example, the British Library's Technical Guidelines for Digitisation states that digitised TIFFs should be TIFF version 6, LZW compressed and each TIFF should contain only one image. These technical constraints are typically described as a "format profile".

If content received from the creator or publisher does not conform to the agreed profile, the preserving institution can reject the content and request new/revised/re-scanned content. However, the preserving institution must have the capability to verify a digital object's compliance with a profile, and if it is not compliant, identify how it fails. It is necessary to perform this check in an automated manner.

Image files may be constructed imperfectly or may be damaged during storage or transfer. It would therefore be useful to be able to verify in an automated fashion that the files are complete (i.e. have not been arbitrarily truncated) and that the files are valid and/or will render in one or more common viewing applications without error (e.g. there are examples of truncated JPEG2000s in the JISC1 dataset are typically reported as valid and well formed by JHOVE).

2.2.3.1 In this scenario, specific component requirements include:

- A tool capable of verifying that a TIFF image complies with a predefined profile.
- A tool capable of verifying that a JPEG 2000 image complies with a predefined profile
- A tool capable of verifying a file is complete and renderable.

2.2.3.2 In this scenario, success criteria include:

- Check that each image conforms to an image profile
- Check that each image for completeness, validity and renderability.

2.2.4 Out-of-sync sound and video in WMV to Video Format-X migration (LSDRT 4)

SB holds a collection of about 4 TB of Windows Media Video files (WMV) with Danish television broadcasts. However, SB does not want to keep files in this file format for preservation. Earlier attempts at migration of these objects using the ffmpeg² tool failed on some files. Some of the migrated files had sound and video out of sync. A new migration with an updated or different conversion tool is necessary. Additionally, a QA tool which can detect these out-of-sync errors is required in order to validate the success of migration.

2.2.4.1 In this scenario, specific component requirements include:

- A tool capable of migrating WMV video/audio files to MPEG 2.
- A tool capable of detecting failures in WMV video/audio migrations.

2.2.4.2 In this scenario, success criteria include:

- The tools should be able to process 20 files per hour per node (average file size is 273 Mb).
- A tool to detect poor quality AV (audio and video) when compared to the original (e.g. when a high compression rate is used).
- 99% of the files should be similar when compared to the original.

2.2.5 Detecting audio files with very bad sound quality (LSDRT 5)

In a collection of MP3 files (360.000 files adding up to 20 TB) one has discovered files with very bad sound quality. Before ingesting everything into our DOMS repository we would like to be able to discover the bad files and potentially get those re-digitized from the original analogue media.

2.2.5.1 In this scenario, specific component requirements include:

- A tool capable of comparing analogue audio with digital surrogates (in MP3 format)

2.2.5.2 In this scenario, success criteria include:

- The solution should be executable in a reasonable time.
- Reliability and precision are relevant since we need to detect files with very bad sound - files that will potentially be removed from the repository and re-digitised.

2.2.6 Large scale migration from mp3 to wav (LSDRT 6)

SB currently owns a small collection of real audio files (digitised CDs). They are part of the Danish publications that SB preserves. The rest of the Danish CD collection is in WAV. This format has been chosen as the preservation format as this is a RAW format, that needs fewer layers of interpretation to be understood by humans and it is also a robust format.

² <http://ffmpeg.org>



The Danish Radio Broadcast MP3 files are also to be migrated to WAV according to the existing policy. The actual migration will be done by one of the SCAPE Action components recommended tools. It is necessary to insure that the migrated files are similar to the original MP3 versions.

2.2.6.1 In this scenario, specific component requirements include:

- A tool capable migrating MP3 audio files to WAV format.
- A tool or workflow capable of doing QA on MP3 to WAV migrations.

2.2.6.2 In this scenario, success criteria include:

- The solution should be able to handle 20 files per hour per node.
- Validation that the migrated file is in the correct format is needed.
- Validation that the header information properties of the migrated files are similar to the original versions.
- Audio waves should be compared with the original ones.

2.2.7 Characterise very large video files (LSDRT 7)

Collections of very large video files (50GB+ each) are hard to handle when it comes to characterisation and validation. Known characterisation tools do not necessarily like very large files. Not all needed formats are well supported (if supported at all) by known tools (JHove, JHove2, FITS, XC*L).

Characterization tools need to be able to work on very large files (50GB+) and in a distributed environment in order to scale well (SB holds more than 400Tbytes mpeg-1/2).

2.2.7.1 In this scenario, specific component requirements include:

- A tool capable characterising very large video files

2.2.7.2 In this scenario, success criteria include:

- The solution should be able to handle file sizes in the order of 50GB-75GB.
- The characterization tool should be compatible with MPEG1 and 2.
- Should be able to process 2TB of content per day.
- The output should be understandable by curators.

2.2.8 Characterisation of large amounts of wav audio (LSDRT 9)

SB holds large amounts of WAV audio files (200TB+) in different resolutions (ranging from 22 KHz 16 bit to 96 KHz 24 bit). Different resolutions have been chosen over the years for different reasons (equipment, budgets for storage space, quality of original media in digitisation). Before we ingest all these older collections into our DOMS repository we need to do simple characterisation on the files to generate correct technical metadata (in PREMIS format) for those files. We know that certain collections that claim to hold only e.g. 48 KHz 16 bit files have files in other resolutions - most likely as a result of faulty operation of the digitisation equipment.

2.2.8.1 In this scenario, specific component requirements include:

- A tool capable of characterising audio files

2.2.8.2 In this scenario, success criteria include:

- The tool should support WAV and BWF files up to 10 GB of file size.
- The solution should process 2 TB of content in less than 24 hours.
- The solution should be accurate (0% error tolerance in characterisation).

2.2.9 Capturing Representation Information from original image files (LSDRT 10)

Camera Raw images are captured to record the colour balance setup as part of mass digitisation projects, e.g. in the Canon Raw format. They contain critical representation information that must be effectively preserved. However, camera RAW formats are typically complex and proprietary, making preservation a challenge.

The primary scalability challenge in this scenario is associated with complexity of proprietary formats. It is also concerned with the migration of the RAW files to Adobe DNG with appropriate quality assurance and extraction of metadata.

2.2.9.1 In this scenario, specific component requirements include:

- A tool capable of converting Canon Camera RAW image files to Adobe DNG format
- A tool capable of doing QA on Canon Camera RAW files against Adobe DNG image files
- A tool capable of extracting metadata from Canon Camera RAW files.

2.2.9.2 In this scenario, success criteria include:

- Being able to handle various Canon raw file formats (each camera model has its own raw format)

2.2.10 Duplicate image detection within one book (LSDRT 11)

Cultural heritage institutions such as libraries, museums and archives have been carrying out large-scale digitisation projects over the last decade. Due to specific processes in a digital book production process (e.g. different scanning sources, various book page image versions, etc.), it can occur that book image duplicates are introduced into the compiled version of a digital book.

The issue presented here is the need to identify books within a large digital book collection that contain duplicated book pages, and to know which book page images are actually duplicate book page image pairs.

Currently there are about 50.000 books (at least 320 pages each), 16 million pages (one image and one OCR output file each in hOCR³ format). For example, a simple compression process that takes 2 seconds for each hOCR file would last 185 days on one single processing node.

2.2.10.1 In this scenario, specific component requirements include:

- A tool capable of detecting duplicate pages within a collection of digitised book pages.

2.2.10.2 In this scenario, success criteria include:

- *Scalability* in terms of throughput (books/time) related to defined quality assurance workflows with increasing sample size (50, 500, 5000 books) in various steps up to a very large data set (50000 books).
- *Reliability* in terms of error-free processing of defined quality assurance workflows
- *Preciseness* in terms of the number of pages and books correctly identified.

2.2.11 Quality assurance in re-download workflows of digitised books (LSDRT 12)

The production of digitised versions of books or newspapers took place either in-house or it was outsourced to an external partner. However, even if commercial partners were involved in the production of digital masters, usually, the results and any attached property rights had mostly been transferred entirely to the originator's institution.

These circumstances have changed in some public-private partnerships, where the digitisation is carried out by the commercial partner, which keeps the original master copy and then produces surrogates, which are provided to the cultural heritage institution. As a consequence, from the point of view of the cultural heritage institution, the preservation challenges relate to the surrogates rather than the original master copies (considering the very unlikely event that the commercial company disappears together with the digital master copies).

This changes an important parameter regarding the use of long-term preservation repositories. Instead of producing a master copy once, which is stored "forever" in the repository and not supposed to change in future, new surrogates of master copies are continuously being made available. The surrogates can be downloaded and ingested into the repository as a new version, which is either added or replaces the original derivate.

In the concrete context of this issue, there are mainly three objects available:

- A METS container for each book item.
- A series of digital images (JPEG2000) for each page of the book.
- A hOCR file containing text and layout information from the OCR.

All the three object types provide information that can be used in a quality assurance process that helps to determine if a new derivate is better in terms of quality compared to previous versions.

³ hOCR is an open standard which defines a data format for representation of OCR output. The standard aims to embed layout, recognition confidence, style and other information into the recognized text itself.

First, images can be used for image analysis and comparison, and context information from the METS file can be taken to compare images from one book (possibly duplicated pages) or from different versions against each other. Second, hOCR files can be used for doing text content and layout analysis. Finally, a hybrid approach, using image comparison and text/layout analysis can be used.

2.2.11.1 In this scenario, specific component requirements include:

- A tool capable of detecting and assessing differences between digitised books

2.2.11.2 In this scenario, success criteria include:

- *Scalability* in terms of throughput (books/time) related to defined quality assurance workflows with increasing sample size (50, 500, 5000 books) in various steps up to a very large data set (50000 books).
- *Reliability* in terms of error-free processing of defined quality assurance workflows.
- *Preciseness* in terms of the number of pages and books correctly identified.

2.2.12 Potential bit rot in image files that were stored on CD (LSDRT 13)

Digitised master image files (TIFFs) from a legacy digitisation project were stored for a number of years on CD. Corresponding service/access images (JPEGs, at a lower resolution, cropped, scale added, and colour balanced) were stored on a web server during this period. Consequently there is a higher confidence in the bit integrity of the service copies. Without checksums, the only method of checking the master images for bit rot is to open each one and visually inspect it.

This scenario also aims at supporting digital preservation quality assurance. It handles the image based document comparison challenges like detection of differences in file format, colour information, scale, rotation, resolution, cropping, and slight differences in content. If the master and service images are the same, or similar, a simple comparison between them would enable bit rot to be detected. However, the high degree of processing applied to the service images means that they are quite different in appearance to the service images. Fuzzy matching between the images may enable parts of the images to be matched, but image focused approaches may be extremely challenging. OCR based comparison may be possible, although OCR engines may struggle with hand written Chinese characters.

In this scenario, specific component requirements include:

- A tool capable of comparing TIFF and JPEG images despite differences caused by processing tasks (e.g. rotation, cropping, etc.)

In this scenario, success criteria include:

- The similarity tool is expected to match a large number of images to its original counterparts (close to 100% of the images)

2.3 Research Dataset Scenarios

2.3.1 General scientific data handling scenario (RDST 1)

“Scientific data” are data files that contain measurements collected from instrument detectors. There is no typical size or number of detectors that an instrument has. For example, for STFC ISIS facility, the number of detectors ranges from several thousands to a quarter of a million. The typical formats of these files are Raw or NeXus. The latter is an international standard for neutron and synchrotron communities. The former is facility specific - many historic data files are in this format. Increasingly, NeXus format is being adopted as the standard format for instrument data.

In order to ensure that the data to be preserved is of adequate quality, there is a need for structural/syntactical verification and characterisation upon ingesting data into repository. These are individual data files produced by the experiments. These files are readings of individual experimental runs. They, themselves, do not have enough information to allow anybody to process them because, basically, they are neutron counts in the STFC ISIS facility case. They are raw data because it contains errors and noises that are needed to be removed before it can be analysed. Therefore, first of all, they have to be preserved alongside with the contextual information describing where it was produced (e.g. which instrument), when it was produced (which ISIS cycle), and what experiment it was produced for. All these information allow establishing the linkages between these raw files and relevant files generated at the same time while the files are being produced during an experiment.

Other types of contextual information needed to be preserved include the software needed to process the files and the samples that are used to produce the files.

Each data model is specified in a NeXus Definition Language (NXDL) file and contains assertions that define the expected content of a NeXus file. For example, a data model could define a metadata element (key-value pair) called “Integral” to represent the total integral monitor counts for grazing incidence small angle diffractometer GISAS for either x-ray or neutrons. In this scenario, the data type of the metadata element “Integral” would be an integer. For a NeXus data file conforming to this data model, it would be necessary to validate the value(s) assigned to “Integral” to ensure it is of appropriate data type.

In this scenario, specific component requirements include:

- A tool to do structural/syntactical verification and characterisation of the NeXus files.
- A tool to capture fixity information (e.g. checksum) to ensure continuous integrity of data files to be preserved.
- A tool to validate the contents of NeXus data files for their correctness against a given data model (e.g. semantic validation).
- A tool capable of doing a basic migration of Raw files to NeXus format
- A tool capable of doing an advanced migration of Raw files to NeXus format

In this scenario, success criteria include:

- Being able to ensure that the data to be preserved is of adequate quality, i.e. syntactically and semantically valid.

- The challenge is to scale the solution to the throughput and file size handling capabilities. The traditional NeXus format validation tool is not designed for large data files (tens to hundreds of GBs per file) in the sense that many such tools take a long time to validate a file. In its peak time, ISIS generates data files concurrently across 40+ instruments, which amounts to hundreds of MB/s.

2.4 Web Content Scenarios

2.4.1 Comparison of Web Archive pages (WCT 1)

The best practice in preserving websites is by crawling them using a web crawler like Heritrix⁴. However, crawling is a process that is highly susceptible to errors. Often, essential data is missed by the crawler and thus not captured and preserved. So if the aim is to create a high quality web archive, doing quality assurance is essential.

Currently, quality assurance requires manual effort and is expensive. Since crawls often contain thousands of pages, manual quality assurance will be neither very efficient nor effective. It might make sense for “topic” crawls but remains time consuming and costly. Especially for large-scale crawls, automation of the quality control processes is a necessary requirement.

Some efforts are undertaken to use a setup with sets of standard web browsers running “headless” and a Wayback Machine in proxy mode. Using this method, links that are missed by the crawler but are detected by the browser can be recorded using the Wayback logs might be thus added to the harvest.

The headless browser approach will help to detect and solve a certain set of problems, specifically links that were missed due to the highly interactive nature of the pages involved or due to robot evasive measures. The approach doesn't help answering QA questions like: is the harvested site still active, how much has the content changed (or should we lower or raise the harvesting frequency), has part of the content moved to a different domain, etc.

The approach suggested to tackle these potential quality issues is to create reference images for each crawled site. A reference image is a snapshot that has undergone the usual (still labour intensive) manual quality assurance process or a screenshot of the live site taken while crawling on a number of different browsers. Using this reference image each new crawl can be compared by automated means. This can be done using various metrics as indicators, like the change in the size of the crawl, the number of changed pages, but also more advanced methods like automated visual comparison of percentage and location of changes in the rendered pages.

Now with the various metrics as indicators of the changes in the new crawl compared to the reference crawl, various actions can be undertaken:

1. Changes are very small - the quality of this crawl is good, a lower crawl frequency can be set;
2. Changes are relatively small - the quality of this crawl is good;

⁴ <https://webarchive.jira.com/wiki/display/Heritrix/Heritrix>

3. Changes are relatively big - manual inspection is required.

When this crawl is approved it will become the new reference crawl. We could also envisage that such automated comparison could trigger automated actions that would improve crawls completeness/quality and allow monitoring of the Web archives from a qualitative and long-term preservation perspective. Overall this approach is likely to improve the effectiveness, efficiency and scalability of quality assurance for specific crawls and Web Archives in general.

2.4.1.1 In this scenario, specific component requirements include:

- A tool capable of comparing two versions of the same web page

2.4.1.2 In this scenario, success criteria include:

- The tool is expected to provide a similarity measure of two web pages

2.4.2 ARC to WARC migration (WCT 2)

ARC and WARC are both container formats and at the moment SB has around 300 TB of web content in ARC format. At some point within the next 2 years they plan to migrate the content from ARC to WARC. A crucial step in this migration is automatic QA to ensure that the migrated container has exactly the same content as the original. This is very important since they do not have the budget to keep both the original ARC files and the new WARC versions.

2.4.2.1 In this scenario, specific component requirements include:

- A tool capable of migrating ARC to WARC
- A tool capable of checking that the content of the migrated WARC is the same as the original ARC

2.4.2.2 In this scenario, success criteria include:

- The challenge is in the size of the collection and the capacity of the tool to produce correct results, so *performance* and *correctness* are the success criteria.

2.4.3 Characterise web content in ARC and WARC containers (WCT 3)

The issue with web content is mainly the fact that web archive data is very heterogeneous. Depending on the policy of the institution, data contains text documents in all kinds of text encoding, HTML content loosely following different HTML specifications, audio and video files that were encoded with a variety of codecs, etc. But in order to take any decisions in preservation, it is indispensable to have detailed information about the content in the web archive, especially those pieces of information that preservation tools depend on.

It is not possible to perform a data migration without knowing exactly what kind of digital object is encountered in the collection and what the logical and technical dependencies of the object are. And it is not only necessary to identify the single objects contained in an ARC/WARC file, but also identify



container formats, like packaged files or any other container formats. Video files, for example, are often available as so called wrapper formats, like AVI, where each, the audio and video stream, can be encoded using different codecs. Down to this level the content stream must be identified if the institutional policy would foresee to preserve all video and audio content contained in a web archive.

Furthermore, the issue has two different aspects. One is the challenge to identify content that is already known. In this sense, the main goal of identification is to identify the content correctly. The second aspect is unknown content in the web archive, which is measured by the coverage of identification tools, where coverage indicates the part of the content that can be identified. Coverage depends on reliability in the sense that a bad reliability can hide a bad coverage in case that many objects are incorrectly identified, but are actually unknown. The challenge regarding this second aspect is to reach a precise set of the unknown objects in order to be able to derive a plan dealing exactly with these objects.

In this scenario, specific component requirements include:

- A tool or set of tools capable of doing deep characterization of files included in container formats such as ARC or WARC
- A tool or set of tools capable of doing deep characterization of video wrapper formats (e.g. AVI)

In this scenario, success criteria include:

- The challenge is in assuring correctness and coverage of the characterisation tool as well as in guaranteeing the scalability of the solutions.

2.4.4 (W)ARC to HBase migration (WCT 6)

IM is migrating its web content, currently stored into (W)ARC files to a new infrastructure based on HBase. The archive contains around 200 TB of data and is growing rapidly. Most of the content crawled will need to be migrated sometime this year.

Once the new infrastructure is ready, services provided to cultural institutions by IM will have to rely on this new infrastructure. The Foundation is currently providing a high-level quality archive and related services such as redirection from live missing content to the archive or resolution of access issues through its access tool. Looking at the investment in terms of manual quality assurance, crawl preparation and development, it is not acceptable to get a lower quality after content is migrated to this new infrastructure.

2.4.4.1 In this scenario, specific component requirements include:

- A tool capable of unwrapping and copying the contents of ARC/WARC files into HBASE.
- A tool capable of validating that the files migrated to HBASE are equal to the original files in the ARC/WARC files

2.4.4.2 In this scenario, success criteria include:

- The copied files should be equal to the original files
- Tools should be able to handle terabytes of data without interfering with the running of the existing system.

2.5 Levels of compliance

Table 2 depicts the various compliance levels of a given component that may be associated with each requirement from a Testbed scenario, as well as a measure of the SCAPE Platform and a SCAPE tool's maturity. These levels represent the maturity of the tool in terms of the way it is capable of solving part (or all) of the scenario and the level of adaptation that the tool has achieved in making it more compatible with the SCAPE Platform and future take up.

Table 2 - Levels of tool compliance against the Platform.

Level of compliance	
0	• No tool has been identified that supports the requirement
1	• A tool has been identified that supports the requirement
2	• The tool is compatible with the SCAPE platform and has an open-source license • All the items in level 1
3	• The tool has been properly wrapped⁵ and packaged for easy distribution • All the items in levels 1 and 2
4	• The tool is available as a Taverna Component for easy invocation and testing • All the items in levels 1, 2 and 3
5	• The tool complies to all SCAPE functional review criteria⁶ • All the items in levels 1, 2, 3 and 4

2.6 Levels of importance

Table 3 depicts the importance given to the fulfilment of a given Testbed scenario requirement. The level of importance will be used to prioritise the actions necessary to transform a non-compliant component into a compliant one.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119⁷.

⁵ Wrapping tools is a procedure developed in SCAPE that unifies the way tools are invoked in the command-line. It also normalises error handling and adds extra functionality to support streams as input and output on tools that do not support it natively. For more information on this procedure read the SCAPE deliverable D5.1 available at http://www.scape-project.eu/deliverable/d5-1_guidelines_for_deploying_preservation_tools_and_environments-2.

⁶ The criteria included in the SCAPE functional review include quality and maturity aspects of software developed in SCAPE related to 1) development process, 2) code quality, 3) documentation, 4) functional evaluation, 5) installation and deployment. Detailed information is available at <http://wiki.opf-labs.org/display/SP/The+SCAPE+Functional+Review+Process>.

⁷ <http://www.ietf.org/rfc/rfc2119.txt>

Table 3 - Levels of importance that a given requirement is fulfilled.

Level of importance		
1	MAY	The word "MAY", or the adjective "OPTIONAL", mean that an item is truly optional. One may choose to include it if it enhances the outcome. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality.
2	SHOULD	The words "SHOULD", or the adjective "RECOMMENDED", mean that there may be valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
3	MUST	The word "MUST", or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement.

2.7 Compliance threshold

The compliance threshold is used to define “where we want to be” in terms of the fulfilment of a given requirement. If our current setting is below the specified threshold, actions should be defined in order to improve the setting to the desired level of compliance.

At this stage of the project, the compliance level will be set to 4 (see Table 2). Any component with a level of compliance below this level should be improved to meet this desired level of conformity.

2.8 Gap analysis

This section depicts “where we are” in terms of being able to comply with the requirements laid out by the Testbed scenarios for Preservation Components.

The gap analysis depicted in Table 4 includes the following information:

- **Ref.** – the reference code of the scenario for referral purposes. The reference codes are the same as the ones found on the Open Planets Foundation Wiki where the primary scenario texts are published;
- **Requirement** - a quick description of each requirement inferred from the scenario description;
- **Level of importance** – The level of importance assigned to a particular requirement according to the scale depicted on Table 3. The levels of importance were assigned by the Testbeds Sub-project lead;
- **Tool description** – The name or brief description of a tool adopted or being developed in the project that fulfils a scenario requirement;
- **Level of compliance** - The level of compliance or maturity of the selected tool according to Table 2;
- **Comments** – A brief note or set of actions to be developed to improve the level of compliance of the tool.

Table 4 - Gap analysis.

Ref.	Requirement	Level of importance	Tool description	Level of compliance	Comments
LSDRT 1 – Assessing preservation risks in large media files					
LSDRT 1.1	<p>There should be a tool capable of doing deep characterization of large video files in MPEG2-TS format</p> <p>The tool should be able to extract all the provided metadata and be able to process streams faster than their defined bitrate.</p>	SHOULD	No tool has been identified yet	0	Using some support tools like dd, ffprobe may be able to fulfil this requirement. Further investigation is needed.
LSDRT 2 – Validating files migrated from TIFF to JPEG 2000					
LSDRT 2.1	<p>There must be a tool capable of migrating TIFF to JPEG 2000</p> <p>The tool should be able to migrate images from the format TIFF to JPG 2000. The tool should operate reliably at scale (80TB, 2 million JPEG 2000 files). The tool should also operate in a consistent and preservation safe fashion, maintaining (or normalizing) the embedded ICC profile, resolution headers and any other metadata that may emerge as being significant.</p>	MUST	<p>ImageMagick is a tool capable of migrating TIFF to JPEG 2000</p>	3	Must enhance to Toolwrapper to be able to produce automatically a Taverna component
			<p>Ffmpeg is a tool capable of migrating TIFF to JPEG 2000</p>	3	Must enhance to Toolwrapper to be able to produce automatically a Taverna component
			<p>GStreamer is a tool capable of migrating TIFF to JPEG 2000</p>	3	Must enhance to Toolwrapper to be able to produce automatically a Taverna component
			<p>Kakadu is a tool capable of migrating TIFF to JPEG 2000</p>	1	Reassess software license to see if it is compatible with SCAPE requirements.
			<p>OpenJPEG is a tool capable of migrating TIFF to JPEG 2000</p>	3	Must enhance to Toolwrapper to be able to produce automatically a Taverna component
			<p>Lurawave is a tool capable of migrating TIFF to JPEG 2000</p>	1	License is not compatible with SCAPE. The tool should not be used in the project, as there are viable alternatives.
			<p>Aware AccuRad is a tool capable of migrating TIFF to JPEG 2000</p>	1	License is not compatible with SCAPE. The tool should not be used in the project, as there are viable alternatives.
			<p>GraphicsMagick is a tool capable of migrating TIFF to JPEG 2000</p>	3	Must enhance to Toolwrapper to be able to produce automatically a Taverna component
LSDRT 2.2	<p>There must be a tool capable verifying that TIFF to JPG2000 migrated files are complete</p> <p>An automated QA process is required to validate the success of the migration. The tool should be capable of verifying, in an automated fashion, that the migrated files are complete (i.e. have not been arbitrarily truncated) and that the files are valid and/or will render in one or more common viewing applications without error.</p>	MUST	<p>Jpylyzer is a tool capable of solving this requirement.</p>	3	<p>Jpylyzer is both a validator and feature extractor for JP2 images. The tool is already packaged in distributed with Debian.</p> <p>Future actions are focused on making use of the Toolwrapper to be able to produce a Taverna component.</p>

LSDRT 2.3	<p>There must be a tool capable of identifying what metadata properties have not been correctly converted to the new format</p> <p>An automated QA process is required to validate the success of the migration in what concerns the embedded image metadata.</p>	MUST	<p><i>“extractFeatures” and “compare”</i> are tools capable of solving this requirement.</p>	3	<p><i>“ExtractFeatures”</i> extracts the features from an input image file. The tool <i>“compare”</i> is responsible for comparing the results of the tool <i>“extractFeatures”</i>. Both tools are command-line tools.</p>
LSDRT 3 – Validating migrated images “visually”					
LSDRT 3.1	<p>There should be a tool capable of verifying that a TIFF image complies with a predefined profile</p> <p>If content received from the creator or publisher does not conform to a predefined agreed profile, the preserving institution can reject the content and request new/revised/re-scanned content. The preserving organisation must have the capability to verify a digital object's compliance with a profile, and if it is not compliant, identify how it fails. It is necessary to perform this check in an automated manner. A validation tool that can check if TIFF files comply with a profile is needed. Profile includes checking for dpi, colour depth, compression type, format, etc.</p>	SHOULD	No tool has been identified yet	0	<p>This tool will be developed under PC.WP.1 Task 5 that is only going to start on 2013.</p>
LSDRT 3.2	<p>There must be a tool capable of verifying that a JPEG 2000 image complies with a predefined profile</p> <p>A preserving organisation must have the capability to verify a digital object's compliance with a profile, and if it is not compliant, identify how it fails. It is necessary to perform this check in an automated manner. A validation tool that can check if JPEG 2000 files comply with a profile is needed. Profile includes checking for the following JPEG 2000 properties: compression type, number of components, component transform, tile size, wavelet filter, number of levels, number of layers, progression order, Codestream markers, Precincts, Codeblock size, Coder Bypass.</p>	MUST	<p>Jpylyzer is a tool capable of partially solving this requirement</p>	3	<p>PC.CC should adopt the Jpylyzer tool and enhance it in order to make it capable of checking if a JP2 matches a predefined profile.</p> <p>At the moment Jpylyzer only extracts technical metadata from image files.</p>
LSDRT 3.3	<p>There should be a tool capable of verifying that a JPEG 2000 image file is complete and renderable</p> <p>Image files may be constructed imperfectly or may be damaged during storage or transfer. It would therefore be useful to be able to verify in an automated fashion that the files are complete (i.e. have not been arbitrarily truncated) and that the files are valid and/or will render in one or more common viewing applications without error.</p>	SHOULD	No tool has been identified yet	0	<p>A tool should be adopted or developed to make sure that an image file is complete and renderable.</p>
LSDRT 4 - Out-of-sync sound and video in WMV to Video Format-X migration					
LSDRT 4.1	<p>There must be a tool capable of migrating WMV video/audio files to MPEG-2 format⁸</p> <p>The tool should be able to process 20 files per</p>	MUST	<p>FFmpeg is a tool capable of converting WMV 1, 2 and 3 to MPEG-2</p>	3	<p>Must enhance to Toolwrapper to be able to produce automatically a Taverna component</p>

⁸ In this context, “preservable” means that the file will be easier to render or will be able to be rendered on more computers than before. This approach is a mere suggestion. A preservation planning approach might recommend to keep the WMV because open source renderer for this format will be available in the future and it will be better to keep the current format rather than taking the risk to lose information by migrating it to a new format.

	hour per node (average file size is 273 Mb). 99% of the files should be accurate when compared to the original.		GStreamer is a tool capable of converting WMV 1, 2 and 3 to MPEG-2	3	Must enhance to Toolwrapper to be able to produce automatically a Taverna component
			Avidemux is a tool capable of converting WMV 2 and 3 to MPEG-2	3	Must enhance to Toolwrapper to be able to produce automatically a Taverna component
LSDRT 4.2	There should be a tool capable of detecting failures in audio/video synchronisation of WMV migrations The tool should be able to detect out-of-sync video and audio. The tool should also detect poor quality video and audio when compared to the original (e.g. when a high compression rate is used).	SHOULD	xCorrSound is a tool capable of solving this requirement.	1	The xCorrSound tool will be able to detect if the sound has shifted, this requires a workflow extracting the sound tracks for comparison. The development of this tool is currently paused. Will restart possibly in 2013.
LSDRT 5 - Detecting audio files with very bad sound quality					
LSDRT 5.1	There should be a tool capable of comparing analogue audio with digital surrogates (in mp3 format) The solution should be executable in a reasonable time (20 TB - 360.000 files to be processed). Reliability and preciseness are also relevant since we need to detect files with very bad sound - files that will potentially be removed from the repository and re-digitised.	SHOULD	Dobbin Audio Analyser ⁹	1	It is an automatic audio processing and rendering solution. The work on this tool will start at 2013.
			PrismSound dScope Series III ¹⁰	1	It is a comprehensive and powerful measurement system for analogue and digital audio generation and analysis. The work on this tool will start at 2013.
LSDRT 6 - Large scale migration from mp3 to wav					
LSDRT 6.1	There should be a tool capable of migrating mp3 audio files to wav format The solution should be able to handle 20 files per hour per node.	SHOULD	Ffmpeg is a tool capable of migrating MP3 to WAV.	3	Must enhance to Toolwrapper to be able to produce automatically a Taverna component
			SOX is a tool capable of migrating MP3 to WAV.	3	Must enhance to Toolwrapper to be able to produce automatically a Taverna component
			GStreamer is a tool capable of migrating MP3 to WAV.	3	Must enhance to Toolwrapper to be able to produce automatically a Taverna component
LSDRT 6.2	There should be a tool capable doing QA on MP3 to WAV migrations The solution should be able to handle 20 files per hour per node.	SHOULD	MigrationQA is a tool aimed at comparing two WAV files. MP3 must be "rendered" to WAV before using the tool.	2	It basically compares two waveforms using fast fourier transformations in order to decide if the actual content of the two files is the same or close to the same. It is important to note that the conversion within the QA tool must be done by a different encoder that the one used for the actual migration. Wrapping and packaging of the tool is in progress.
LSDRT 7 – Characterise very large video files					

⁹ This tool is mentioned at <http://wiki.opf-labs.org/display/SP/LSDRT5+Detecting+audio+files+with+very+bad+sound+quality>. No product link could be found.

¹⁰ http://www.prismosound.com/test_measure/products_subs/dscope/dscope_home.php?src=GGL_ADWRD_RD

LSDRT 7.1	<p>There must be a tool capable characterising very large video files</p> <p>The solution should be able to handle file sizes in the order of 50GB-75GB and process 2TB of content per day. The output should be understandable by curators.</p>	MUST	ffprobe with some additional developments will be able to fulfil this requirement	1	Using some support tools like dd, ffprobe will be able to fulfill this requirement.
LSDRT 9 – Characterisation of large amounts of wav audio					
LSDRT 9.1	<p>There may be a tool capable of characterising WAV and BWF files up to 10Gb of file size</p> <p>The tool should be able to support for both WAV and BWF formats, files up to 10Gb, process 2 TB of sample content in less than 24 hours, and be accurate (0% errors in characterisation).</p>	MAY	No tool has been identified yet	0	The WP hasn't worked on this scenario yet.
LSDRT 10 - Capturing Representation Information from original image files					
LSDRT 10.1	<p>There may be a tool capable of converting Canon Camera Raw image files to Adobe DNG format</p> <p>The dataset includes only 500 files, so the primary scalability challenge is associated with complexity of the proprietary formats. The tool should be able to handle various Canon raw file formats (each camera model has its own raw format).</p>	MAY	Photoshop is a tool capable of converting Raw to DNG	1	License and the execution requirements are not compatible with the SCAPE platform. The tool should not be used in the project, as there are viable alternatives.
			dng4ps-2 is a tool capable of converting Raw to DNG	1	Include it in D10.1r2, create a toolspec and generate a debian package with a component workflow
LSDRT 10.2	<p>There may be a tool capable of doing QA on Canon Camera Raw and Adobe DNG format</p> <p>The dataset includes only 500 files, so the primary scalability challenge is associated with complexity of the proprietary formats.</p>	MAY	There is an experimental tool available, but it does not have name yet	2	For now, it can be called more "experiments" than a tool, looking at which kind of components exist, how they can be composed and how QA task can be automated for preservation planning for raw photographs.
LSDRT 10.3	<p>There may be a tool capable of extracting metadata from Canon Camera Raw</p> <p>The dataset includes only 500 files, so the primary scalability challenge is associated with complexity of the proprietary formats.</p>	MAY	ImageMagick is a tool capable of solving this requirement (however it depends on the library ufraw).	3	Create a toolspec and generate a debian package with a component workflow
			ACDSee is a tool capable of solving this requirement.	1	License and the execution requirements are not compatible with the SCAPE platform. The tool should not be used in the project, as there are viable alternatives.
			GraphicConverter is a tool capable of solving this requirement.	1	License and the execution requirements are not compatible with the SCAPE platform. The tool should not be used in the project, as there are viable alternatives.
			Photoshop is a tool capable of solving this requirement.	1	License and the execution requirements are not compatible with the SCAPE platform. The tool should not be used in the project, as there are viable alternatives.
LSDRT 11 - Duplicate image detection within one book					

LSDRT 11.1	<p>There should be a tool capable of detecting duplicate pages within a collection of digitised book pages</p> <p>The tool should be scalable in terms of throughput (books/time) related to defined quality assurance workflows with increasing sample size (50, 500, 5000 books) in various steps up to a very large data set (50000 books). The tool should be reliable in terms of error free processing of defined quality assurance workflows and also precise in terms of the number of pages and books correctly identified.</p>	SHOULD	<p>FindDuplicates in MatchBox is a tool capable of solving this requirement.</p>	3	<p>FindDuplicates tool is based on SIFT features and SSIM methods for comparing images and find duplicates.</p>
LSDRT 12 - Quality assurance in redownload workflows of digitised books					
LSDRT 12.1	<p>There must be a tool capable of detecting differences between digitised books</p> <p>The tool should be scalable scalability in terms of throughput (books/time) related to defined quality assurance workflows with increasing sample size (50, 500, 5000 books) in various steps up to a very large data set (50000 books). The tool should be reliable in terms of error free processing of defined quality assurance workflows and also precise in terms of the number of pages and books correctly identified.</p>	MUST	<p>MatchBox is a tool capable of solving this requirement</p>	3	<p>Wrapping and packaging of the tool is in progress.</p>
LSDR 13 - Potential bit rot in image files that were stored on CD					
LSDRT 13.1	<p>There may be a tool capable of detecting fuzzy differences between images in TIF and JPEG formats</p> <p>The JPEG files have been generated from the original TIFs, however, image processing algorithms have been applied during the migration process to make the images more usable (JPEGs are service images, i.e. DIPs). The similarity tool should be able to bypass any filters that may have been applied to the service images.</p>	MAY	<p>MatchBox is a tool capable of solving this requirement</p>	3	<p>Wrapping and packaging of the tool is in progress.</p>
RDST 1 - General scientific data handling scenario					
RDST 1.1	<p>There must be a tool capable of validating a large XML file against the NeXus schema</p> <p>The challenge is to scale the solution to the throughput and file size handling capabilities. The traditional NeXus format validation tool is not designed for large data files (10s to 100s GBs per file) in the sense that many such tools take a long time to validate a file. Additionally, some tools will fail in the presence of large files. In its peak time, ISIS generates data files concurrently across 40+ instruments, which amounts to 100s MB/s. In the coming years, due to the upgrading of instruments and the introduction of new instruments, the volume is likely to increase to 1GB/s. Although the main data file format is standardised in ISIS, which is mainly the nexus format, each instrument generates other types of data files, which are also essential for downstream processing. The complication is that these other types of data files vary between instruments.</p>	MUST	<p>nxalyser is a tool that will be developed that will be able to fulfill this requirement</p>	0	<p>The tool is still to be developed, however, it already has a name</p>
RDST 1.2	<p>There may be a tool capable of generating checksums for large-sized files</p> <p>The challenge is in handling extremely large files in large quantities in a timely fashion.</p>	MAY	<p>No tool has been identified yet</p>	0	<p>A tool will be developed. No name has been defined yet.</p>

RDST 1.3	<p>There may be a tool capable of doing semantic validation of NeXus files</p> <p>In order to ensure that the data to be preserved is of adequate quality, the contents of NeXus data files would need to be validated for their correctness against a given data model. Each data model is specified in a NeXus Definition Language (NXDL) file and contains assertions that define the expected content of a NeXus file. For example, a data model could define a metadata element (key-value pair) called "Integral" to represent the total integral monitor counts for grazing incidence small angle diffractometer GISAS for either x-ray or neutrons. In this scenario, the data type of the metadata element "Integral" would be an integer. For a NeXus data file conforming to this data model, it would be necessary to validate the value(s) assigned to "Integral" to ensure it is of appropriate data type.</p>	MAY	nxvalid is a tool that is able to fulfill this requirement	1	The tool is still under development.
RDST 1.4	<p>There must be a tool capable of doing a basic migration of Raw files to NeXus format</p> <p>The Nexus format has become an international standard for the exchange of scientific data. Being able to migrate all raw files to Nexus is important for both dissemination and preservation purposes.</p>	MUST	raw2nx is a tool that is able to fulfill this requirement	2	The tool has been developed however it still needs to be packaged for easy distribution.
RDST 1.5	<p>There should be a tool capable of doing an advanced migration of Raw files to NeXus format</p> <p>There is a desire to enhance the value of the dataset with additional information about an experiment that is not present in the basic data file, so as to enrich the dataset with representation information. Apart from the file size and volume of content, the raw to NeXus format migration tool can be customised to take into account of various other types of experiment data files in the process of the migration. However, the scalability challenge here is that for different instrument (specific to each facility), the other types of experiment data files vary significantly. This makes it difficult to efficiently migrate large quantity of complex raw data files systematically.</p>	SHOULD	raw2nxplus is a tool that is able to fulfill this requirement	0	The tool is still to be developed, however, it already has a name
WCT 1 - Comparison of Web Archive pages					
WCT 1.1	<p>There must be a tool capable of comparing two versions of the same web page</p> <p>The solution could be a combination of structural and visual comparison methods embedded in a statistical discriminative model, a visual similarity measure designed for Web pages that improves change detection, and a supervised feature selection method adapted to Web archiving.</p>	MUST	Pagelyzer is a tool capable of solving this requirement.	2	Pagelyzer compares web pages based on structural and visual approaches. Tool is in a testing phase. Wrapping and packaging of the tool will start after the tests.
WCT 2 - ARC to WARC migration					
WCT 2.1	<p>There must be a tool capable of migrating ARC to WARC</p> <p>The challenge is in the size of the collection and the capacity of the tool to produce correct results.</p>	MUST	No tool has been identified yet	0	warc-tools and heritrix are tools capable of reading ARC and WARC files. A new tool must be developed for copying contents to HBASE. Then create a toolspec and generate a debian package with a component workflow

WCT 2.2	<p>There must be a tool capable of checking that the content of the migrated WARC is the same as the original ARC</p> <p>The challenge is in the size of the collection and the capacity of the tool to produce correct results.</p>	MUST	No tool has been identified yet	0	<p>Discussions in the first year of the project lead to the decision that there is no need to check the content of the migrated WARC is the same as the original ARC.</p> <p>No development planned.</p> <p>For more information, contact to Matthieu Cord or/and Stéphane Gañçarski.</p>
WCT 3 - Characterise web content in ARC and WARC containers					
WCT 3.1	<p>There may be a tool capable of doing deep characterization of ARC and WARC files</p> <p>The challenge is in assuring correctness and coverage of the characterisation tool as well as in guaranteeing the scalability of the solutions.</p>	MAY	Web Archive Mime-Type detection workflow	1	<p>Web Archive Mime-Type detection workflow based on Droid and Apache Tika. More information available at¹¹. The workflow might need to be encapsulated to make it more user-friendly and behave more like a component.</p> <p>We are working on integrating Tika, DROID, and FITS with Hadoop and ARC and WARC files.</p>
WCT 3.2	<p>There may be a tool capable of doing deep characterization of video wrapper formats (e.g. AVI)</p> <p>The challenge is in assuring correctness and coverage of the characterisation tool as well as in guaranteeing the scalability of the solutions.</p>	MAY	ffprobe with some additional developments will be able to fulfil this requirement	1	<p>Using some support tools like dd, ffprobe will be able to fulfill this requirement.</p> <p>Apart from ffprobe we have not yet started working with these formats.</p>
WCT 6 - (W)ARC to HBase migration					
WCT 6.1	<p>There should be a tool capable of unwrapping and copying the contents of ARC/WARC files into HBASE</p> <p>The tool should be able to handle large volumes of data without interfering with the running of the existing system.</p>	SHOULD	No tool has been identified yet	0	<p>warc-tools and heritrix are tools capable of reading ARC and WARC files. A new tool must be developed for copying contents to HBASE. Then create a toolspec and generate a debian package with a component workflow</p>
WCT 6.2	<p>There should be a tool capable validating that the files migrated to HBASE are according to the original ARC/WARC files</p> <p>The tool should be able to handle large volumes of data without interfering with the running of the existing system.</p>	SHOULD	No tool has been identified yet	0	<p>No development planned.</p> <p>WCT 6 description will be updated by IM soon. Contact information: Leila Medjkoune from IM.</p>

¹¹ <http://wiki.opf-labs.org/display/SP/WCT3+Characterise+web+content+in+ARC+and+WARC+containers+at+State+and+University+Library+Denmark>

3 Analysis

In this section we analyse the results of the gap analysis presented in section 2.8. Particular attention is given to scenario requirements that do not have an associated tool or a roadmap for its short-term development.

For each scenario requirement, we have calculated a “level of priority”, i.e. the priority one should give to solving a particular scenario requirement. The level of priority is calculated as a function that combines the “level of importance” assigned to a given requirement and the effort necessary to bring a component to the desired “level of compliance”. The level of priority is calculated according to the following formula:

$$\text{level of priority} = \text{level of importance} \times (5 - \text{level of compliance})$$

Table 5 depicts the level of priority that should be placed in the development of a given component, or in other words, the level of effort that should be placed into the fulfilment of a particular testbed scenario requirement. Higher levels of priority mean that a greater effort is necessary to fulfil the corresponding requirement, either because the requirement is considered to be very important (e.g. a requirement that MUST be fulfilled) or because the level of maturity of a tool is, at the time of analysis, very low.

For some scenario requirements there are several competing tools that are capable of solving them. In some cases, tools exist that have a low level of compliance that are competing against tools that have a level of compliance already at level 3. In these situations, little effort will be put in bringing those tools up to level 3 or greater as there are other tools that already fulfil this requirement. These tools will therefore be given little priority in terms of development roadmap and will be ignored in the following analysis.

Table 5 – Component development priority as a function of level of importance to fulfil a requirement and effort necessary for its development.

Ref.	Requirement	Level of importance	Tool name	Level of compliance	Level of priority
LSDRT 1 – Assessing preservation risks in large media files					
LSDRT 1.1	There should be a tool capable of doing deep characterization of large video files in MPEG2-TS format	2	N/A	0	10
LSDRT 2 – Validating files migrated from TIFF to JPEG 2000					
LSDRT 2.1	There must be a tool capable of migrating TIFF to JPEG 2000	3	ImageMagick	3	6
			FFmpeg	3	6
			GStreamer	3	6
			Kakadu	1	Ignored
			OpenJPEG	3	6
			Lurawave	1	Ignored
			Aware AccuRad	1	Ignored
			GraphicsMagick	3	6
			ACDSee	1	Ignored
LSDRT 2.2	There must be a tool capable verifying that the migrated files are complete	3	Jpylyzer	3	6
LSDRT 2.3	There must be a tool capable of identifying what metadata properties have not been correctly converted to the new format	3	“extractFeatures” and “compare”	3	6

LSDRT 3 - Characterisation of very large image collections					
LSDRT 3.1	There should be a tool capable of verifying that a TIFF image complies with a predefined profile	2	N/A	0	10
LSDRT 3.2	There must be a tool capable of verifying that a JPEG 2000 image complies with a predefined profile	3	Jpylyzer	3	6
LSDRT 3.3	There should be a tool capable of verifying that a JPEG 2000 image file is complete and renderable	2	Jpylyzer	0	10
LSDRT 4 - Out-of-sync sound and video in WMV to Video Format-X migration					
LSDRT 4.1	There must be a tool capable of migrating WMV video/audio files to a more preservable format (e.g. mpeg 2)	3	FFmpeg	3	6
			GStreamer	3	6
			Avidemux	3	6
LSDRT 4.2	There should be a tool capable of detecting failures in WMV video/audio migrations	2	xCorrSound	1	8
LSDRT 5 - Detecting audio files with very bad sound quality					
LSDRT 5.1	There should be a tool capable of comparing analogue audio with digital surrogates (in mp3 format)	2	Dobbin Audio Analyser	1	8
			PrismSound dScope Series III	1	8
LSDRT 6 - Large scale migration from mp3 to wav					
LSDRT 6.1	There should be a tool capable of migrating mp3 audio files to wav format	2	FFmpeg	3	4
			SOX	3	4
			GStreamer	3	4
LSDRT 6.2	There should be a tool capable doing QA on MP3 to WAV migrations	2	MigrationQA	2	6
LSDRT 7 - Characterise very large video files					
LSDRT 7.1	There must be a tool capable characterising very large video files	3	ffprobe with some additional developments	1	12
LSDRT 9 - Characterisation of large amounts of wav audio					
LSDRT 9.1	There may be a tool capable of characterising WAV and BWF files up to 10Gb of file size	1	N/A	0	5
LSDRT 10 - Capturing Representation Information from original image files					
LSDRT 10.1	There may be a tool capable of converting Canon Camera Raw image files to Adobe DNG format	1	Photoshop	1	4
			dng4ps-2	1	4
LSDRT 10.2	There may be a tool capable of doing QA on Canon Camera Raw and Adobe DNG format	1	There is a tool, but it does not have name yet	2	3
LSDRT 10.3	There may be a tool capable of extracting metadata from Canon Camera Raw	1	ImageMagick	3	2
			ACDSee	1	Ignored
			GraphicConverter	1	Ignored
			Photoshop	1	Ignored
LSDRT 11 - Duplicate image detection within one book					
LSDRT 11.1	There should be a tool capable of detecting duplicate pages within a collection of digitised book pages	2	FindDuplicates in MatchBox	3	4
LSDRT 12 - Quality assurance in redownload workflows of digitised books					
LSDRT 12.1	There must be a tool capable of detecting differences between books	3	MatchBox	3	6
LSDR 13 - Potential bit rot in image files that were stored on CD					
LSDRT 13.1	There may be a tool capable of detecting fuzzy differences between images in TIF and JPEG formats	1	MatchBox	3	2
RDST 1 - General scientific data handling scenario					
RDST 1.1	There must be a tool capable of validating a large XML file against the NeXus schema	3	N/A but it will be called nxalyser	0	15
RDST 1.2	There may be a tool capable of generating checksums for large-sized files	1	N/A	0	5
RDST 1.3	There may be a tool capable of doing semantic validation of NeXus files	1	nxvalid	1	4
RDST 1.4	There must be a tool capable of doing a basic migration of Raw files to NeXus format	3	raw2nx	2	9
RDST 1.5	There should be a tool capable of doing an advanced migration of Raw files to NeXus format	2	N/A but it will be called raw2nxplus	0	10
WCT 1 - Comparison of Web Archive pages					
WCT 1.1	There must be a tool capable of comparing two versions of the same web page	3	Pagelyzer	2	9

WCT 2 - ARC to WARC migration					
WCT 2.1	There must be a tool capable of migrating ARC to WARC	3	N/A	0	15
WCT 2.2	There must be a tool capable of checking that the content of the migrated WARC is the same as the original ARC	3	No tool has been identified yet	0	15
WCT 3 - Characterise web content in ARC and WARC containers					
WCT 3.1	There may be a tool capable of doing deep characterization of ARC and WARC files	1	Web Archive Mime-Type detection workflow	1	4
WCT 3.2	There may be a tool capable of doing deep characterization of video wrapper formats (e.g. AVI)	1	ffprobe with some additional developments	1	4
WCT 6 - (W)ARC to HBase migration					
WCT 6.1	There should be a tool capable unwrapping and copying the contents of ARC/WARC files into HBASE	2	No tool has been identified yet	0	10
WCT 6.2	There should be a tool capable validating that the files migrated to HBASE are according to the original ARC/WARC files	2	No tool has been identified yet	0	10

In section 2.7, we have set the acceptable minimum for tool maturity to be level 4. This means that a tool should be compatible with the SCAPE platform and have an open-source license. Additionally, a tool should be packaged for easy distribution and deployment on a cluster of servers and should also be available as a Taverna Component for easy invocation in testing environments (e.g. to ease the integration with Preservation Planning). We should point out that none of the tools analysed have met the minimum level of compliance previously specified.

At the time of writing, the specification of a Taverna Component is still under development; therefore, it's difficult for a tool to be compliant with something that is still ill defined. Nonetheless, it is important to point out that the generation of Taverna Components will be automatic if developers make use of the "tool wrapper" application¹².

The "tool wrapper" is a command-line application that reads a tool specification file (called *toolspec*) describing a particular digital preservation tool, i.e. what it does, who developed it, how to install it, how to invoke it, what are its dependencies and other technical details. The *toolspec* is written in XML and follows a well-defined schema. Writing a *toolspec* file enables the following outputs to be automatically generated:

- A command-line script that uniforms the name of the tool, parameter passing, adds support for input and output streams, normalizes output errors, etc.;
- A web service for invoking the tool over the Web;
- A single-step Taverna workflow that enables anyone to easily use the tool in larger, more complex, Taverna workflows;
- A software package for easy installation of these artefacts and all its dependencies in Debian Linux machines.

A new version of the "tool wrapper" (to be released in the first trimester of 2013) will also output a Taverna component that will simplify the invocation of existing tools and normalize its output ports. This means that tools that currently are at a compliance level 3 will easily be raised to a level 4 as soon as the new version of the tool wrapper is released and re-run. However, one may note that

¹² Updated information on the tool wrapper is available at <https://github.com/openplanets/scAPE/tree/master/pc-as/toolwrapper>

existing toolspecs might require small enhancements in order to generate compliant Taverna components.

3.1 Complete list of scenario requirements and development priorities

Table 6 depicts the list of scenarios requirements, the identified tools that fulfilled them, and the level of priority/effort necessary to bring those tools to level 4 of compliance. Tools that have been previously ignored have been removed from this analysis.

Table 6 - List of requirements and tools sorted by level of development priority.

Ref.	Requirement	Tool name	Level of compliance	Level of priority
RDST 1.1	There must be a tool capable of validating a large XML file against the NeXus schema	N/A but it will be called nxylyser	0	15
WCT 2.1	There must be a tool capable of migrating ARC to WARC	N/A	0	15
WCT 2.2	There must be a tool capable of checking that the content of the migrated WARC is the same as the original ARC	No tool has been identified yet	0	15
LSDRT 7.1	There must be a tool capable characterising very large video files	ffprobe with some additional developments	1	12
LSDRT 1.1	There should be a tool capable of doing deep characterization of large video files in MPEG2-TS format	N/A	0	10
LSDRT 3.1	There should be a tool capable of verifying that a TIFF image complies with a predefined profile	N/A	0	10
LSDRT 3.3	There should be a tool capable of verifying that a JPEG 2000 image file is complete and renderable	N/A	0	10
RDST 1.5	There should be a tool capable of doing an advanced migration of Raw files to NeXus format	N/A but it will be called raw2nxplus	0	10
WCT 6.1	There should be a tool capable unwrapping and copying the contents of ARC/WARC files into HBASE	No tool has been identified yet	0	10
WCT 6.2	There should be a tool capable validating that the files migrated to HBASE are according to the original ARC/WARC files	No tool has been identified yet	0	10
RDST 1.4	There must be a tool capable of doing a basic migration of Raw files to NeXus format	raw2nx	2	9
WCT 1.1	There must be a tool capable of comparing two versions of the same web page	Pagealyzer	2	9
LSDRT 4.2	There should be a tool capable of detecting failures in WMV video/audio migrations	xCorrSound	1	8
LSDRT 5.1	There should be a tool capable of comparing analogue audio with digital surrogates (in mp3 format)	Dobbin Audio Analyser	1	8
LSDRT 5.1	There should be a tool capable of comparing analogue audio with digital surrogates (in mp3 format)	PrismSound dScope Series III	1	8
LSDRT 3.2	There must be a tool capable of verifying that a JPEG 2000 image complies with a predefined profile	Jpylyzer	3	6
LSDRT 2.1	There must be a tool capable of migrating TIFF to JPEG 2000	ImageMagick	3	6
LSDRT 2.1	There must be a tool capable of migrating TIFF to JPEG 2000	FFmpeg	3	6
LSDRT 2.1	There must be a tool capable of migrating TIFF to JPEG 2000	GStreamer	3	6
LSDRT 2.1	There must be a tool capable of migrating TIFF to JPEG 2000	OpenJPEG	3	6
LSDRT 2.1	There must be a tool capable of migrating TIFF to JPEG 2000	GraphicsMagick	3	6
LSDRT 2.2	There must be a tool capable verifying that the migrated files are complete	Jpylyzer	3	6
LSDRT 2.3	There must be a tool capable of identifying what metadata properties have not been correctly converted to the new format	"extractFeatures" and "compare"	3	6
LSDRT 4.1	There must be a tool capable of migrating WMV video/audio files to a more preservable format (e.g. mpeg 2)	FFmpeg	3	6
LSDRT 4.1	There must be a tool capable of migrating WMV video/audio files to a more preservable format (e.g. mpeg 2)	GStreamer	3	6
LSDRT 4.1	There must be a tool capable of migrating WMV video/audio files to a more preservable format (e.g. mpeg 2)	Avidemux	3	6
LSDRT 6.2	There should be a tool capable doing QA on MP3 to WAV migrations	MigrationQA	2	6
LSDRT 12.1	There must be a tool capable of detecting differences between books	MatchBox	3	6
LSDRT 9.1	There may be a tool capable of characterising WAV and BWF files up to 10Gb of file size	N/A	0	5
RDST 1.2	There may be a tool capable of generating checksums for large-sized files	N/A	0	5
LSDRT 6.1	There should be a tool capable of migrating mp3 audio files to wav format	FFmpeg	3	4

LSDRT 6.1	There should be a tool capable of migrating mp3 audio files to wav format	SOX	3	4
LSDRT 6.1	There should be a tool capable of migrating mp3 audio files to wav format	GStreamer	3	4
LSDRT 10.1	There may be a tool capable of converting Canon Camera Raw image files to Adobe DNG format	Photoshop	1	4
LSDRT 10.1	There may be a tool capable of converting Canon Camera Raw image files to Adobe DNG format	dng4ps-2	1	4
LSDRT 11.1	There should be a tool capable of detecting duplicate pages within a collection of digitised book pages	FindDuplicates in MatchBox	3	4
RDST 1.3	There may be a tool capable of doing semantic validation of NeXus files	nxvalid	1	4
WCT 3.1	There may be a tool capable of doing deep characterization of ARC and WARC files	Web Archive Mime-Type detection workflow	1	4
WCT 3.2	There may be a tool capable of doing deep characterization of video wrapper formats (e.g. AVI)	ffprobe with some additional developments	1	4
LSDRT 10.2	There may be a tool capable of doing QA on Canon Camera Raw and Adobe DNG format	There is a tool, but it does not have name yet	2	3
LSDRT 10.3	There may be a tool capable of extracting metadata from Canon Camera Raw	ImageMagick	3	2
LSDRT 13.1	There may be a tool capable of detecting fuzzy differences between images in TIF and JPEG formats	MatchBox	3	2

3.2 Scenario requirements with no tools

There are 10 scenario requirements that have not been addressed yet, meaning that there are no tools identified as possible solutions to that particular requirement. In such cases a brand new tool will have to be developed.

Table 7 summarises the scenarios requirements that lack the existence of a tool.

Table 7 - Scenario requirements with no tools.

Ref.	WP	Requirement	Tool name	Level of compliance	Level of priority
RDST 1.1	CC	There must be a tool capable of validating a large XML file against the NeXus schema	N/A but it will be called nxalyser	0	15
WCT 2.1	AS	There must be a tool capable of migrating ARC to WARC	N/A	0	15
WCT 2.2	QA	There must be a tool capable of checking that the content of the migrated WARC is the same as the original ARC	No tool has been identified yet	0	15
LSDRT 1.1	CC	There should be a tool capable of doing deep characterization of large video files in MPEG2-TS format	N/A	0	10
LSDRT 3.1	CC	There should be a tool capable of verifying that a TIFF image complies with a predefined profile	N/A	0	10
LSDRT 3.3	CC	There should be a tool capable of verifying that a JPEG 2000 image file is complete and renderable	N/A	0	10
RDST 1.5	AS	There should be a tool capable of doing an advanced migration of Raw files to NeXus format	N/A but it will be called raw2nxplus	0	10
WCT 6.1	AS	There should be a tool capable unwrapping and copying the contents of ARC/WARC files into HBASE	No tool has been identified yet	0	10
WCT 6.2	QA	There should be a tool capable validating that the files migrated to HBASE are according to the original ARC/WARC files	No tool has been identified yet	0	10
RDST 1.2	AS	There may be a tool capable of generating checksums for large-sized files	N/A	0	5
LSDRT 9.1	CC	There may be a tool capable of characterising WAV and BWF files up to 10Gb of file size	N/A	0	5

3.3 Scenario requirements with need of attention

Apart from the scenarios requirements for which there are no tools available, there are 13 requirements for which a considerable amount of effort is necessary to produce a tool capable of tackling the requirement in an appropriate manner.

Table 8 summarises the list of tools that need further development in order to meet the minimum level of compliance expected. Tools that are already at a level 3 of compliance have been removed from this analysis as they will be automatically raised to level 4 as soon as the new version of the “tool wrapper” is released.

Table 8 - List of scenario requirements that need developer’s attention.

Ref.	WP	Requirement	Tool name	Level of compliance	Level of priority
LSDRT 7.1	CC	There must be a tool capable characterising very large video files	ffprobe with some additional developments	1	12
RDST 1.4	AS	There must be a tool capable of doing a basic migration of Raw files to NeXus format	raw2nx	2	9
WCT 1.1	QA	There must be a tool capable of comparing two versions of the same web page	Pagelyzer	2	9
LSDRT 4.2	QA	There should be a tool capable of detecting failures in WMV video/audio migrations	xCorrSound	1	8
LSDRT 5.1	QA	There should be a tool capable of comparing analogue audio with digital surrogates (in mp3 format)	Dobbin Audio Analyser	1	8
LSDRT 5.1	QA	There should be a tool capable of comparing analogue audio with digital surrogates (in mp3 format)	PrismSound dScope Series III	1	8
LSDRT 6.2	QA	There should be a tool capable doing QA on MP3 to WAV migrations	MigrationQA	2	6
LSDRT 10.1	AS	There may be a tool capable of converting Canon Camera Raw image files to Adobe DNG format	Photoshop	1	4
LSDRT 10.1	AS	There may be a tool capable of converting Canon Camera Raw image files to Adobe DNG format	dng4ps-2	1	4
RDST 1.3	CC	There may be a tool capable of doing semantic validation of NeXus files	nxvalid	1	4
WCT 3.1	CC	There may be a tool capable of doing deep characterization of ARC and WARC files	Web Archive Mime-Type detection workflow	1	4
WCT 3.2	CC	There may be a tool capable of doing deep characterization of video wrapper formats (e.g. AVI)	ffprobe with some additional developments	1	4
LSDRT 10.2	QA	There may be a tool capable of doing QA on Canon Camera Raw and Adobe DNG format	There is a tool, but it does not have name yet	2	3

3.4 Per work package analysis

In this section we provide an analysis of the tools that need to be enhanced or developed in order to solve the requirements outlined by testbed scenarios. In this case, the results are presented on a per work package basis.

Table 9 presents a summary of the per work package analysis. In this table it is possible to understand the total number of tools that need to be enhanced or developed and how much effort is required to take them to level 4 of compliance.

Table 9 - Summary of per work package analysis.

Work package	Total No. of tools	No. tools at level of compliance 0	No. tools at level of compliance 1	No. tools at level of compliance 2	Average compliance	Average priority
CC - Characterization components (WP 9)	9	5	4	0	0,44	8,22
AS - Action Services (WP 10)	7	4	2	1	0,57	8,14
QA - Quality Assurance (WP 11)	8	2	3	3	1,33	8,11
Total	24	11	9	4	0,78	8,16

Table 10 summarises the development roadmap for the Characterization Components Work Package (WP 9). In this work package, 7 tools have been identified that require development effort.

Table 10 – Characterization tools with need of attention

Ref.	WP	Requirement	Tool name	Level of compliance	Level of priority
RDST 1.1	CC	There must be a tool capable of validating a large XML file against the NeXus schema	N/A but it will be called nxalyser	0	15
LSDRT 7.1	CC	There must be a tool capable characterising very large video files	ffprobe with some additional developments	1	12
LSDRT 1.1	CC	There should be a tool capable of doing deep characterization of large video files in MPEG2-TS format	N/A	0	10
LSDRT 3.1	CC	There should be a tool capable of verifying that a TIFF image complies with a predefined profile	N/A	0	10
LSDRT 3.3	CC	There should be a tool capable of verifying that a JPEG 2000 image file is complete and renderable	N/A	0	10
LSDRT 9.1	CC	There may be a tool capable of characterising WAV and BWF files up to 10Gb of file size	N/A	0	5
RDST 1.3	CC	There may be a tool capable of doing semantic validation of NeXus files	nxvalid	1	4
WCT 3.1	CC	There may be a tool capable of doing deep characterization of ARC and WARC files	Web Archive Mime-Type detection workflow	1	4
WCT 3.2	CC	There may be a tool capable of doing deep characterization of video wrapper formats (e.g. AVI)	ffprobe with some additional developments	1	4

Table 11 summarises the development roadmap for the Action Services Components Work Package (WP 10). In this work package, 7 tools have been identified that require development effort.

Table 11 – Action services with need of attention

Ref.	WP	Requirement	Tool name	Level of compliance	Level of priority
WCT 2.1	AS	There must be a tool capable of migrating ARC to WARC	N/A	0	15
RDST 1.5	AS	There should be a tool capable of doing an advanced migration of Raw files to NeXus format	N/A but it will be called raw2nxplus	0	10
WCT 6.1	AS	There should be a tool capable unwrapping and copying the contents of ARC/WARC files into HBASE	No tool has been identified yet	0	10
RDST 1.4	AS	There must be a tool capable of doing a basic migration of Raw files to NeXus format	raw2nx	2	9
RDST 1.2	AS	There may be a tool capable of generating checksums for large-sized files	N/A	0	5
LSDRT 10.1	AS	There may be a tool capable of converting Canon Camera Raw image files to Adobe DNG format	Photoshop	1	4
LSDRT 10.1	AS	There may be a tool capable of converting Canon Camera Raw image files to Adobe DNG format	dng4ps-2	1	4

Table 12 summarises the development roadmap for the Quality Assurance Work Package (WP 11). In this work package, 9 tools have been identified that require development effort.

Table 12 – Quality assurance tools with need of attention

Ref.	WP	Requirement	Tool name	Level of compliance	Level of priority
WCT 2.2	QA	There must be a tool capable of checking that the content of the migrated WARC is the same as the original ARC	No tool has been identified yet	0	15
WCT 6.2	QA	There should be a tool capable validating that the files migrated to HBASE are according to the original ARC/WARC files	No tool has been identified yet	0	10
WCT 1.1	QA	There must be a tool capable of comparing two versions of the same web page	Pagelyzer	2	9
LSDRT 4.2	QA	There should be a tool capable of detecting failures in WMV video/audio migrations	xCorrSound	1	8
LSDRT 5.1	QA	There should be a tool capable of comparing analogue audio with digital surrogates (in mp3 format)	Dobbin Audio Analyser	1	8
LSDRT 5.1	QA	There should be a tool capable of comparing analogue audio with digital surrogates (in mp3 format)	PrismSound dScope Series III	1	8

LSDRT 6.2	QA	There should be a tool capable doing QA on MP3 to WAV migrations	MigrationQA	2	6
LSDRT 10.2	QA	There may be a tool capable of doing QA on Canon Camera Raw and Adobe DNG format	There is a tool, but it does not have name yet	2	3

4 Actions to implement

This section we summarise the set of actions that need to be taken in each work package in order to meet the minimum level of tool compliance/maturity.

Table 13 lists the actions required close the gaps found on the Characterization Components work package (WP 9).

Table 13 – Characterization tools development roadmap

Ref.	WP	Requirement	Tool name	Level of compliance	Level of priority	Actions to implement
RDST 1.1	CC	There must be a tool capable of validating a large XML file against the NeXus schema	N/A but it will be called nxalyser	0	15	An entirely new tool must be developed and wrapped.
LSDRT 7.1	CC	There must be a tool capable characterising very large video files	ffprobe with some additional developments	1	12	A new tool based on ffprobe must be developed and wrapped.
LSDRT 1.1	CC	There should be a tool capable of doing deep characterization of large video files in MPEG2-TS format	N/A	0	10	An entirely new tool must be developed and wrapped.
LSDRT 3.1	CC	There should be a tool capable of verifying that a TIFF image complies with a predefined profile	N/A	0	10	An entirely new tool must be developed and wrapped.
LSDRT 3.3	CC	There should be a tool capable of verifying that a JPEG 2000 image file is complete and renderable	N/A	0	10	An entirely new tool must be developed and wrapped.
LSDRT 9.1	CC	There may be a tool capable of characterising WAV and BWF files up to 10Gb of file size	N/A	0	5	An entirely new tool must be developed and wrapped.
RDST 1.3	CC	There may be a tool capable of doing semantic validation of NeXus files	nxvalid	1	4	The nxvalid tool is under development. Once that development is finished, the tool should be tested and wrapped.
WCT 3.1	CC	There may be a tool capable of doing deep characterization of ARC and WARC files	Web Archive Mime-Type detection workflow	1	4	There is a workflow that aims at doing mime-type detection. This workflow must be enhanced to support ARC and WARC files and be encapsulated to behave more like a component. The result should then be wrapped for distribution and easy deployment.
WCT 3.2	CC	There may be a tool capable of doing deep characterization of video wrapper formats (e.g. AVI)	ffprobe with some additional developments	1	4	A new tool based on ffprobe must be developed and wrapped.

Table 14 lists the actions required close the gaps found on the Action Services Components work package (WP 10).

Table 14 – Action services development roadmap

Ref.	WP	Requirement	Tool name	Level of compliance	Level of priority	Actions to implement
WCT 2.1	AS	There must be a tool capable of migrating ARC to WARC	N/A	0	15	A new tool based on heritrix and warc-tools must be developed and wrapped.
RDST 1.5	AS	There should be a tool capable of doing an advanced migration of Raw files to NeXus format	N/A but it will be called raw2nxplus	0	10	An entirely new tool must be developed and wrapped.
WCT 6.1	AS	There should be a tool capable unwrapping and copying the contents of ARC/WARC files into HBASE	No tool has been identified yet	0	10	An entirely new tool must be developed and wrapped.
RDST 1.4	AS	There must be a tool capable of doing a basic migration of Raw files to NeXus format	raw2nx	2	9	The tool needs to be wrapped.
RDST 1.2	AS	There may be a tool capable of generating checksums for large-sized files	N/A	0	5	An entirely new tool must be developed and wrapped.
LSDRT 10.1	AS	There may be a tool capable of converting Canon Camera Raw image files to Adobe DNG format	Photoshop	1	4	Photoshop is not compatible with the SCAPE platform so it will not be selected for further improvement. No action required.
LSDRT 10.1	AS	There may be a tool capable of converting Canon Camera Raw image files to Adobe DNG format	dng4ps-2	1	4	The tool needs to tested and then wrapped.

Table 15 lists the actions required close the gaps found on the Quality Assurance work package (WP 11).

Table 15 – Quality assurance development roadmap

Ref.	WP	Requirement	Tool name	Level of compliance	Level of priority	Actions to implement
WCT 2.2	QA	There must be a tool capable of checking that the content of the migrated WARC is the same as the original ARC	No tool has been identified yet	0	15	An entirely new tool must be developed and wrapped.
WCT 6.2	QA	There should be a tool capable validating that the files migrated to HBASE are according to the original ARC/WARC files	No tool has been identified yet	0	10	An entirely new tool must be developed and wrapped.
WCT 1.1	QA	There must be a tool capable of comparing two versions of the same web page	Pagelyzer	2	9	The tool needs to be tested and then wrapped.
LSDRT 4.2	QA	There should be a tool capable of detecting failures in WMV video/audio migrations	xCorrSound	1	8	The tool needs to be tested and then wrapped.
LSDRT 5.1	QA	There should be a tool capable of comparing analogue audio with digital surrogates (in mp3 format)	Dobbin Audio Analyser	1	8	This is a hardware solution. No action is required.
LSDRT 5.1	QA	There should be a tool capable of comparing analogue audio with digital surrogates (in mp3 format)	PrismSound dScope Series III	1	8	This is a hardware solution. No action is required.
LSDRT 6.2	QA	There should be a tool capable doing QA on MP3 to WAV migrations	MigrationQA	2	6	The tool should be named and then wrapped.
LSDRT 10.2	QA	There may be a tool capable of doing QA on Canon Camera Raw and Adobe DNG format	There is a tool, but it does not have name yet	2	3	The tool should be named and then wrapped.



5 Conclusions

This deliverable constitutes a gap analysis on the current state of preservation components against the SCAPE Testbed scenarios. In this analysis, existing preservation components were evaluated against SCAPE Testbed scenario requirements in order to detect functionality gaps that should be addressed during the next development stages of the project. A total of 24 requirements were identified that are in need of attention by SCAPE developers. 11 of these need new tools to be developed.

As previously stated, this report is based on a snapshot of the Testbed scenarios text published at the Open Planets Foundation wiki at <http://wiki.opf-labs.org/display/SP/Scenarios>. The Testbed scenarios are “live” in the sense that they are under continuous improvement from a wide range of people from distinct content holding institutions. This led to the fact that some of the scenarios lack conformity in what concerns the existence of testable and measurable requirements. This has made the gap analysis work more difficult to determine when the scenarios conditions have been met by a particular tool. Work is underway to improve these conditions, i.e. the wiki is being restructured and scenarios are being refactored so that they are clearer and easier to work with.

6 References

Bowen, R., & McDonough, M. (2010). Different Ways to Approach a Gap Analysis. *Bright Hub Project Management*.