National Library of Estonia- Digital Archive Technical Specification Report

Rahvusraamatukogu

26.04.2019

Euroopa Liit
Euroopa Sotsiaalfond

Eesti
tuleviku heaks

Building a better working world

Table of Content

1. Introduction

1.1 Purpose of the project

The purpose of the project is to renew a digital archive of the National Library of Estonia (hereinafter NLE) which would help to preserve Estonian cultural heritage and cover the needs of the users and employees of NLE.

The digital archive of the NLE is currently based on the Fedora Commons 3.6.1 storage platform, which has several different user interfaces (DIGAR, DEA). The current platform is outdated and does not offer the required long-term preservation options, which is why a project for modernizing the digital archive solution in NLE has been launched.

The goal of the project is to improve the long-term preservation capability of all types of content needed for NLE, to make the reception, storage and recovery of digital objects more efficient, and to provide storage services to external parties. It also aims to improve the functionality of technical metadata management and to guarantee the auditability of the digital archive in line with CoreTrust requirements.

The digital archive must ensure its stored digital repositories and associated metadata to function indefinitely long time. The digital archive must be open, allow to switch to new standards and components easily after the initial deployment and offer opportunities for migrating objects to other storage platforms. The digital archive solution must also take into account the continuous growth of data volumes, thus be well scalable and support the efficient launch and comprehensive documentation of large-scale processes.

The purpose of the procurement procedure is to find a partner who would offer to NLE the most suitable solution for digital archive. This document is giving the requirements for the new digital archive, it’s implementation and requirements.

1.2 Abbreviations and Definitions

The most important definitions are brought out in the following table.

Definition	Explanation
ABBYY FineReader	Software that enables text recognition (OCR) in image files and PDF files to make their content easier to search and use. https://www.abbyy.com/en-eu/finereader/
AIP	The Archival Information Package is an archive package that consists of a digital object and the information needed to store it. Created on the Submission Information Package (SIP). Ref: https://public.ccsds.org/pubs/650x0m2.pdf
Archive object	Publisher ingested file which is converted into suitable long term preservation format. The formats of the files (submitted and converted) may or may not match. The archive object is formed according the Digital Preservation Policy.
Authority record	An authority record is created for the standardized form of a name or term. For example, a description of the person, body, family or work written in the standard form. https://www.loc.gov/marc/uma/pt1-7.html In the context of NLE, authority records are created, stored and managed in the library system ESTER. Data is also integrated with VIAF (The Virtual International Authority File). Authority data is compliant with the following international standards and rules: AACR2 (Anglo-American Cataloguing Rules, ed.2) and MARC21. Soon there will be a shift towards RDA and FRAD/ IFLA LRM. Estonian-language guidelines for creating authority records are available here: http://www.elnet.ee/en/e-kataloog-ester/juhendmaterjalid See also: FRAD:https://www.ifla.org/files/assets/cataloguing/frad/frad_2013.pdf VIAF: https://viaf.org/ LRM: http://www.isko.org/cyclo/lrm RDA: http://rda-rsc.org/content/about-rda
Bibliographic record	A bibliographic record is an entry in a bibliographic index (or a library catalogue) which represents and describes a specific resource. A bibliographic record contains the data elements necessary to help users identify and retrieve that resource, as well as additional supporting information, presented in a formalized bibliographic format. Additional information may support particular database functions such as search, or browse (e.g., by keywords), or may provide fuller presentation of the content item (e.g., the article's abstract). In the context of NLE: The bibliographic records are created, stored and managed in the library system ESTER/Sierra. Bibliographic data is compliant with the following international standards and rules: ISBD (International Standard Bibliographic Description), AACR2 (Anglo-American Cataloguing Rules, ed.2) and MARC21. Soon there will be a shift towards RDA and FRBR/IFLA LRM. Estonian-language guidelines for creating bibliographic records are available here: http://www.elnet.ee/en/e-kataloog-ester/juhendmaterjalid FRBR: https://www.oclc.org/research/activities/frbr.html

	LRM: http://www.isko.org/cyclo/lrm
CIP record	Cataloging-in-publication (CIP) record. Basic cataloging data for a work, prepared before publication.
Collection	Documents collected and / or categorized by the employee according to different selection principles for display to the reader. Ref: digital collage policy_2018_aug.docx
CRM	Customer-relationship management system
Dark Archive	A part of digital archive which does not reveal presence and public access of preserved data. Mostly, this feature is used by external clients who wish to preserve their data in NLE digital archive but do not want to share it with the public.
DDL	Data definition language - standard for commands that define the different structures in a database.
DEA	A unite web portal for all digitally born or digitised newspapers, journals, and other serials that are published in Estonia or published abroad in Estonian dea.digar.ee
DIGAR	NLE information system, which holds digital materials and digitized books, periodicals, image material, sheet music, music, etc., relevant to the Estonian cultural heritage. DIGAR consists of an Operator interface (archiving), Viewer, an administration interface and a Fedora archive. For the public user, DIGAR is a Viewer. See also http://www.digar.ee/arhiiv/en
DIGAR converter	Part of DIGAR, which prepares binary files for viewing and stores them in local DIGAR storage.
DIP	Dissemination Information Package, Package for the viewing files. An information package that is created on the basis of one or more AIPs and transmitted to the user for viewing.
DML	Data manipulation language – standard for inserting, modifying, deleting, and retrieving data.
docWorks	docWorks is a software program used to digitize (transform printed documents to digital images—i.e., by scanning pages and receiving JPGs, TIFFs, or PDFs) and convert these images into intelligent units using OCR (text recognition) and zoning (the identification of headlines, text blocks and images on a page). In NLE, docWorks is used in the workflow of processing periodicals. https://content-conversion.com/
DROID	Digital Record Object Identification - software tool for the PRONOM technical registry https://digital-preservation.github.io/droid/
Dublin Core	An international metadata standard that includes 15 generic descriptive elements to describe any content and guarantee the findability of the content. http://dublincore.org/

Dump link	The URL inside the Fedora object, which refers to the location pointer of the file. In order to find out the actual location, there is a Dump script that converts the dump link into a usable location path. The original URL is not present in Fedora because otherwise in the case of an update of the link it should be also changed in Fedora objects. Dump script creates an intermediate layer with translation logic that can be easily changed.
EDM	Europeana Data Model. EDM is the format for transferring metadata about NLE objects to the Europeana portal. https://pro.europeana.eu/resources/standardization-tools/edm-documentation
Pre-object	An archiveable object created in the DIGAR Operator, to which metadata and files will be added or already automatically generated. Fedora ID is reserved for each pre-object.
EMS - Estonian Subject Thesaurus	The Estonian Subject Thesaurus is a universal controlled vocabulary in Estonian for indexing and searching various library material. EMS is the main thesaurus used in ESTER. http://ems.elnet.ee/
EPUB	A standard e-book format published by the International Digital Publishing Forum (IDPF). EPUB format is an archive-file, which consists of HTML files, images and other supporting files. EPUB is the most widely supported independent XML-based e-book format. The EPUB format has the potential to become an open and supported e-document standard, but current versions are not backwards compatible.
ESTER (Sierra)	ESTER is Estonian libraries electronic catalogue, based on integrated library system Sierra developed by Innovative Interfaces Inc. ESTER is a joint catalog of 18 largest libraries in Estonia. https://www.ester.ee/ https://www.iii.com/products/sierra-ils/
ExifTool	Open source software for reading, writing and editing metadata of image, audio, video and PDF files. https://en.wikipedia.org/wiki/ExifTool
Fedora	Open source digital repository information system, which stores descriptions, objects, and collection objects of archived files. Current digital archive is using Fedora Commons version 3.6.1. https://wiki.duraspace.org/display/FEDORA36/
FITS	FITS (File Information ToolSet) is a software which contains / packages several different file checking and technical metadata extraction (Tika, Jhove, MediaInfo, etc.) components. https://projects.iq.harvard.edu/fits/home
FoXML	Fedora Object XML format of information package implemented in Fedora system (i.e., format for packing data and metadata, including descriptions in Dublin Core format).
GOOBI	Goobi is an open source software application for digitisation projects. It allows to model, manage and control digitisation processes. It contributes to the creation of standardized metadata (METS, MODS, etc.) and creates a possibility to display digitized content on the web. NLE uses GOOBI application which is customised by Intranda. https://www.intranda.com/en/digiverso/goobi/goobi-overview/

GUI	Graphical user interface
Heritrix	Free web site saving / crawler bot. NLE uses version 3. For web archiving, NLE uses several robots at the same time. According to the configuration, the software will automatically save web pages (downloading the contents of web pages in WARC format). Crawl.log forms part of the WARC format from which the metadata is being extracted. See also https://github.com/internetarchive/heritrix3/wiki
IP	Information package (For further information, please see paragraph 3.3)
ISBN	(International Standard Book Number) - A unique identifier for each book. The ISBN is a unique worldwide, i.e., there is no two books with the same number.
ISMN	(International Standard Music Number) - A unique standard number for sheet music.
ISN	International Standard Number (ISBN, ISMN, ISSN) - Common acronym for international standard numbers.
ISSN	An ISSN (International Standard Serial Number) identifies all continuing resources, irrespective of their medium (print or electronic). https://www.issn.org
JHOVE	Open source software for identifying, validating, and separating technical metadata of file format (s). http://jhove.openpreservation.org/
JPEG 2000	JPEG 2000 allows files to be compressed without loss, while maintaining quality and reducing file size. It is also an acceptable image archive format since in some cases it is more suitable than TIFF.
Krool	MySQL-based web archiving workflow management software that is connected via API to Hetitrix. The Krool script writes metadata into one metadata WARC file. Krool is used to manage web harvesting tasks done by Heritrix bot.
MARC	Machine Readable Cataloging - A technical standard set of bibliographic records created by the US Library of Congress. In ESTER the libraries are currently using the MARC21 standard. https://www.loc.gov/marc/
METS	METS (Metadata Encoding and Transmission Standard) is one of the most widely used XML standards for creating information packets (i.e. for linking and / or encapsulating metadata and data). http://www.loc.gov/standards/mets/
METS/ALTO	ALTO (Analyzed Layout and Text Object) is a XML Schema that details technical metadata for describing the layout and content of physical text resources, such as pages of a book or a newspaper. In the context of NLE ALTO is used in combination with METS. The METS.xml lists which files the METS/ALTO group is composed of and where each file is located. The ALTO xml contains text recognition (OCR) coordinates, i.e. it describes where each of the detected words is located on the page.

	Note that the METS/ALTO files are located in the storage area of the Veridian software and are NOT stored or registered in the Fedora file repository. Ref: digital collage policy_2018_aug.docx. See also: https://www.veridiansoftware.com/knowledge-base/metsalto/ http://www.loc.gov/standards/alto/
Minio	Minio is an open source object storage server with Amazon S3 compatible API. https://www.minio.io/
OAI-PMH	Protocol for metadata harvesting https://www.openarchives.org/pmh/
OAIS	Open Archive Information System Model. Internationally recognized standard model for the principles of digital archive functionality and archive information package architecture. https://public.ccsds.org/pubs/650x0m2.pdf
OCR	Optical character recognition - the conversion of typed, printed or handwritten text into machine-readable form.
ONIX	Online Information exchange - publishing protocol https://en.wikipedia.org/wiki/ONIX_(publishing_protocol)
Operaator	A software module used in DIGAR for creating and archiving pre-objects and managing archive objects. AIP is formed in the Operaator interface.
PDF	Widespread file format especially for text and image content. In the context of NLE, PDF is one of the main archive and transfer formats (i.e., downloadable print files from the publisher portal must in most cases be in PDF format).
PDF/A	PDF / A is a file format developed for archiving and for long-term storage. PDF / A differs from regular PDFs by prohibiting features which are unsuitable for long-term storage, such as linking and encrypting fonts. PDF / A assure that electronic documents can be reproduced in the same format as different software. Portable Document Format; ISO 19005-1:2005 Document management – Electronic Document file format for long-term preservation – Part 1: Use of PDF 1.4, Level B Conformance (PDF/A-1b)
PhantomJS	Software that saves your website as a screenshot. PhantomJS will only take a screenshot of the home page when run. Purpose for future use: Display users next a website entry and to perform automatic quality control - comparison of images. Phantom JS supports PNG, JPEG, GIF and PDF formats. See also http://phantomjs.org/screen-capture.html
PLOP	(Software) Tool for optimizing, linearizing, analyzing, repairing, etc. PDF files. Developed by PDFlib GmbH. Ref: https://www.pdflib.com/fileadmin/pdflib/pdf/datasheets/PLOP-datasheet.pdf
PREMIS	Preservation Metadata Maintenance Activity - standard in the area of digital preservation https://www.loc.gov/standards/premis/

PubCode	PubCode is a part of metadata which is used to identify publication in processing and accessing periodical publications in docWorks / DEA. PubCode is generated of the title and type of the publication. Pubcode does not contain diacritics or numbers. Types: AK - journal, JV - serials, without a prefix newspaper. For example: *AKakadeemia - Akadeemia, JVriigiktoim- serial Riigikogu toimetised, ohtuleht- newspaper Õhtuleht*
Publisher portal	The publisher portal is a custom developed web-based solution where publishers can apply for international standard numbers (ISBN, ISSN, ISMN), add descriptive metadata, determine copyright and access restrictions, upload files of the publication and any accompanying digital content for preservation.
S3	Amazon Simple Storage Service object storage solution.
SIP	Submission Information Package (SIP), Input Information Package. An information packet from the depositary which is formatted to archiving form, to form AIPs (archive packages).
Squidwarc	Open source crawler for capturing web pages. The functioning of Squidwarc is similar to Heritrix. The main difference is that Squidwarc "presents" itself as a Chrome Web browser to the archievable web page.
SRU/SRW	Protocol for metadata harvesting http://www.loc.gov/standards/sru/
StructMap	The structural map is the heart of the METS document - defines the primary hierarchical layout of the source document. The hierarchy is coded as a tree of elements. Any given can refer to another METS document through the mptr element. can also refer to a single file, file groups through fptr or additional elements.
Template	The form used to create a pre-object in the Operaator.
TIFF	File format for uncompressed or lossless compression of raster images. The format is widely used for storing and transmitting raster images in consumer graphics, publishing and polygraphy enterprises, as it supports the CMYK color model. It is also suitable for archiving documents as they will remain in their original form for future use and processing. Ref: file formats_long_address_2017.pdf
TIKA	Apache Tika, open source software to detect and extract metadata from different file formats. https://tika.apache.org/
Veridian	DEA presentation layer, web interface. REF: digikogu_silitusp_chool_2018_aug.docx Licensed product (https://www.veridiansoftware.com/).
VIAF	Virtual International Authority File - an international authority file https://viaf.org/
WARC (Web ARChive)	File format developed for archiving webpages. Allows you to take the entire content of the web page (text, images, links, audio, video, etc.) and "wrap" it into one big WARC file.

	WARC is also an ISO standard (ISO 28500: 2017). See also https://iipc.github.io/warc-specifications/specifications/warc-format/warc-1.1/
WAV	Waveform Audio File Format is a Microsoft and IBM audio file format standard for storing audio bitstream. WAV has been selected by the National Library as an acceptable format for archiving audio materials. WAV is a very widespread format that allows audio to be delivered in lossless manner. Ref: failivormingud_pikaajaliseks_sailitamiseks_2017.pdf
WayBack Machine	Software to view archived Web pages in WARC format. Created by the Internet Archive, a non-profit organization.
Web Archive	Preserved web content, published in Estonia, under ee domain or relevant to Estonian culture. The content is stored in WARC format.
WebCrawler	Web search and storage robot. Allows the administrator to configure which webpages (or domains) to search for, and the frequency and depth, etc. from the servers. According to the settings, the software will automatically search and run web pages that meet the preset conditions and save their contents in the WARC format to a 4TB buffer disk. NLE uses two crawlers - Heritrix and Squidwarc.
XML	XML is a standard general-purpose markup language developed and recommended by W3C for the purpose of structured information sharing between information systems. XML is a human and machine-readable language. Ref: failivormingud_pikaajaliseks_sailitamiseks_2017.pdf
Z39.50	Protocol for metadata harvesting http://www.loc.gov/z3950/

2. Overview of digital archive

2.1 Processes of the National Library digital archive

This paragraph outlines the details about specific flows and components of NLE digital archive. Please note that process descriptions outlined below also include parts that are not in scope regarding the procurement of the digital archive.

2.1.1 Pre-Ingest and Ingest

The current NLE architecture differentiates clearly between four main content types and according pre-ingest and ingest flows:

- **Born-digital monographic publications**: Publishers submit legal deposit copies of their publications through the *Publisher Portal*. The portal is a custom developed web-based solution where publishers can apply for international standard numbers (ISBN, ISSN, ISMN), add descriptive metadata, determine copyright and access restrictions, upload files of the publication and any accompanying digital content for preservation. The portal enables the manual review of added content by the employees of NLE. Once NLE has accepted the publication, its metadata and data files are manually loaded into the *Operaator* tool. *Operaator* allows employees of NLE to assemble the Information Packages (SIP and AIP), execute further technical controls, create additional technical and administrative metadata (with Jhove and TIKA), format metadata into the FOXML, store binary files within the local storage layer and metadata within *Fedora Commons*. Please note that descriptive metadata is not transferred directly from the *Publisher Portal* to *Operaator*. Instead at first, a MARC21 bibliographic record is created, exported and then imported into the integrated library system Sierra. Part of this record is transferred from Sierra to Operaator (converted from MARC21 to Dublin Core).
- **Born-digital periodical publications:** In the case of periodical publications the manual creation of descriptive metadata, copyright and access restrictions is only done once, and afterwards reused in the form of XML metadata templates. Born-digital files of periodical publications undergo additional processing (segmenting, creating METS/ALTO representations) using the *docWorks* software component. Instead of the *Operaator*, FOXML creation and data storage is handled by custom scripts created in-house.
- **Digitisation:** A separate workflow is a digitisation where its main software component *Goobi* is used to manage digitisation queues, overview initial quality etc. Digitised content mostly undergoes OCR using the *ABBYY FineReader* software. The script searches for the match of

Operaator pre-object ID from the network of the Digitization Centre. When finding the correct ID, the files (tif-> ARHIIV1, processed tiff → ARHIIV2 and pdf → ARHIIV3) are forwarded to the *Operaator* pre-object. In some cases, ARHIIV4 (sound) and ARHIIV5 (photo) are added manually. Finally, the *Operaator* component is used to assemble Information Packages, add information about copyright and access restrictions, create technical and administrative metadata, and store data and metadata. However, some digitised periodicals (newspapers, journals and serials) are processed with *docWorks* software (in the same way as for born-digital periodicals).

- **Web archiving:** The NLE Web archiving process is independent from other flows and managed by the locally developed *Krool* component. *Krool* covers most OAIS functions regarding the NLE web archive - in pre-ingest and ingest it allows NLE employees to record websites to be archived and their crawl settings, gather descriptive, technical and administrative metadata, and execute the actual crawl job with the help of external components (*Heritrix, squidwarc, PhantomJS* and *Puppeteer*). *Krool* also handles the creation of Information Packages (in WARC format) and their storage.

2.1.2 Data management

Storage items' metadata is managed mainly in three systems:

- The primary system for creating, storing and managing descriptive metadata is the shared online catalogue *ESTER* (https://www.ester.ee). *ESTER* is based on the integrated library system *Sierra* and it is used both for managing the library's catalogue and the national bibliography. Additionally, *ESTER* contains the registry of international standard numbers (ISBN, ISSN, ISMN) issued in Estonia.
- For the digital archive (digitised and born-digital publications), most metadata is managed within *Operaator*. As described above, *Operaator* handles the creation and management of technical, administrative and preservation metadata, and handles the information about copyright[1] and access restrictions for digital content. NLE imports also a small part of descriptive metadata from *ESTER/Sierra* into *Operaator* during the creation of Information Packages. Please note that descriptive metadata added into *Operaator* as part of Information Package metadata, is not automatically changed or updated in case of updates in the e-catalogue *ESTER*. While all metadata held in *Operaator* is also stored within *Fedora Commons*

[1] Please note that information about copyrights and access restrictions is not synchronised in Publisher Portal.

as a FOXML file, most changes and updates to technical, preservation, administrative and access metadata occur in *Operaator* (i.e. *Operaator* acts as a Data Management layer and GUI for *Fedora Commons*).

- The Data Management function for the web archive is managed by *Krool*. The *Krool* database records and manages descriptions of websites (not standardized yet, but implementation of Dublin Core metadata framework is planned in 2019) and individual crawl jobs, technical metadata about WARC files, administrative and preservation metadata, crawl logs and content storage locations. Note that none of the information is synchronized into other NLE systems, information about archived websites is also not available in ESTER / Sierra.

2.1.3 Storage

Historically RAID disk storage has been used for digital content at NLE. In August 2018, the setup and data transfer to local S3 Minio storage took place and by early 2019, most content has been transferred to S3. The transition included all the files that were in the digital archive file repository, although due to technical errors, about 4000 files were not transferred. Some content remains on the so-called online archive disks for AIPs (most notably for the web archive). DIPs generated during the pre-ingest / ingest processes also remain stored on online disk storage. In addition to online storage a local tape backup is created for all digital content and an off-site tape backup is created.

2.1.4 Access

Three different software solutions are used for access to digital archive:

- For digital monographs and journals/serials published before 2017, NLE has developed its own access portal *DIGAR* (https://www.digar.ee/arhiiv/en). *DIGAR* receives its data and metadata from *Fedora* during the ingest process (i.e. AIP and DIP creation are handled almost in parallel). Once an Information Package has been created within *Operaator* and sent to *Fedora*, data destined for online access is sent to processing within the *DIGAR converter* component (which prepares binary files for viewing and stores them in local *DIGAR* storage) and metadata is populated into the *DIGAR* database. *DIGAR* does not regularly interact with other Storage and Data Management components but acts independently. However, when changes to descriptive metadata, copyright and access details occur within *Operaator,* these are also updated in *DIGAR*.
- For periodicals NLE has developed an access portal *DEA* (https://dea.digar.ee/cgi-bin/dea?l=en). The specific viewer used within *DEA* is *Veridian* software

(https://www.veridiansoftware.com/discovery-delivery/). In the portal the *Veridian* software is used which allows to give access to the digital content in a more comprehensive digital format (METS/ALTO, and searchable PDF). In *DEA,* files are segmented by their contents (f.ex, PDF is splitted by articles, it contains). Similarly to *DIGAR*, all content is populated into *DEA* during ingest[2] and there is no regular interaction between *DEA* and other Data Management and Storage components.

- The content of the *web archive* is published using a locally managed and installed instance of the *WayBack Machine* (http://veebiarhiiv.digar.ee/). Internally for quality assurance of captured websites is used pywb (https://pypi.org/project/pywb/) which is expected to be installed also to the public interface in the future.

The figure below (Figure 1) shows the scope of the new digital archive solution from processes' perspective:

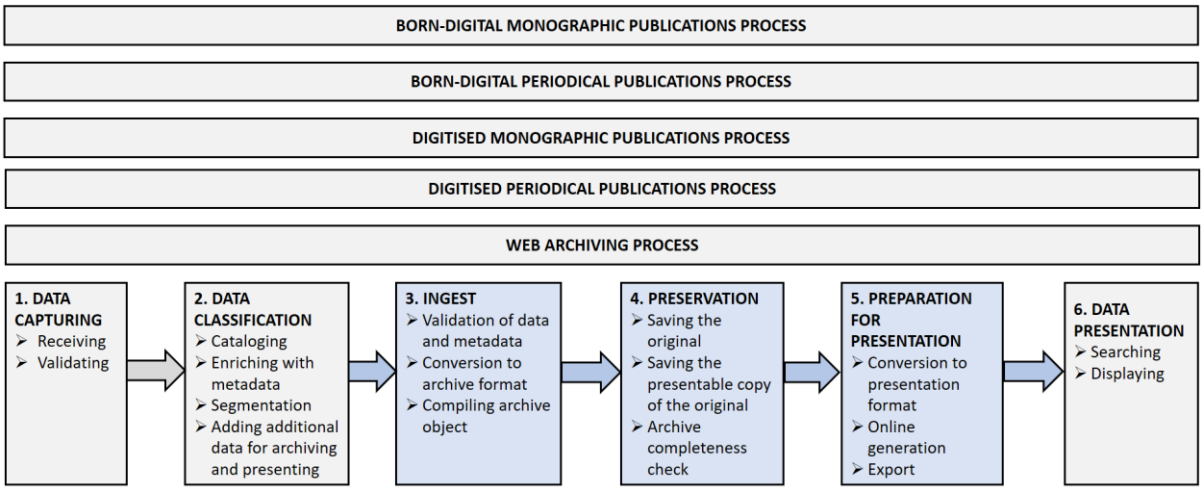

Figure 1. Digital archiving processes in NLE

2.2 Current architecture of the digital archive

NLE started developing its digital archiving ecosystem more than a decade ago. The current NLE digital archiving ecosystem comprises of more than 20 individual software components, and presents a mix of local custom-developed components, open-source and commercially available packages. Figure 2 presents a high-level overview of the current architecture.

[2] Certain metadata management is done in the corresponding XML files.

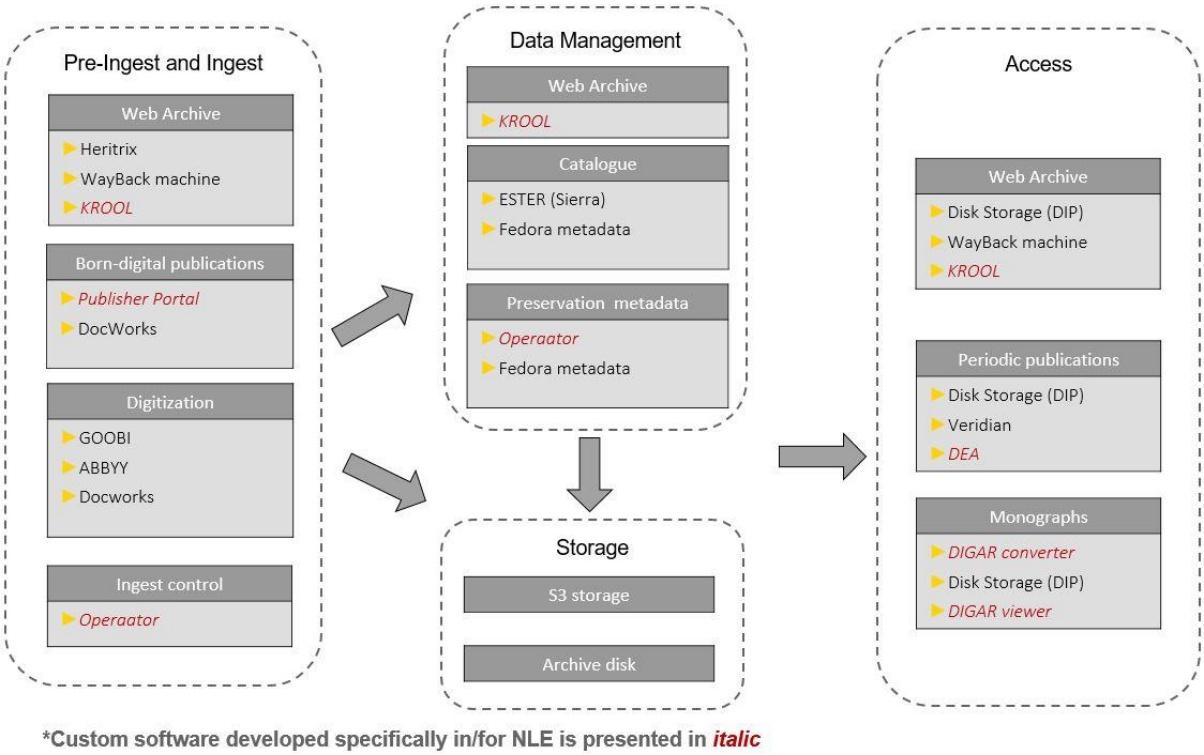

Figure 2. High-level architecture of NLE current digital archiving components

2.3 Business goals for the future digital archive

The business goal for this project is to preserve and make available digital cultural heritage that NLE is preserving in digital format. It is already known that Estonia is planning vast cultural heritage digitization projects which will be preserved in NLE. In the next 5-10 years, NLE is planned to be one of the state's digital preservation competence centres. Details about the requirements and expectations of NLE regarding its future digital archive are outlined in section 3.

NLE looks to replace its core digital repository components (Fedora Commons, Operaator, the preservation layer of the web archive) with modern, state-of-the-art component(s). However, please note that it is also within the scope of this project to connect with other systems in the areas of pre-ingest, data management and access.

The current system components planned to replace with the new digital preservation system are highlighted on the following schema (Figure 3).

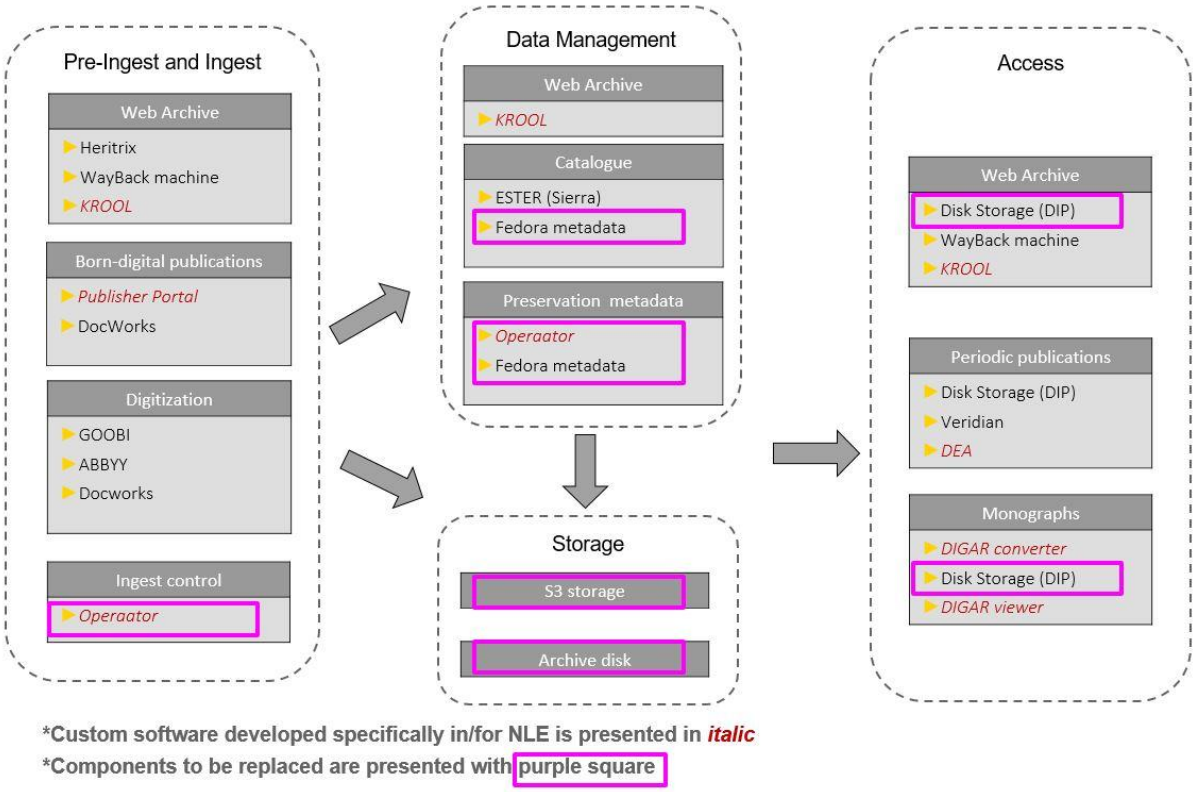

Figure 3. High-level architecture of NLE current digital archiving components with highlighted components which will be replaced

The new digital archive should meet at least the following business requirements:

- NLE has technical control over the main components of the digital archive[3]:
  - NLE has full control over its SIP and DIP formats and associated rules;
  - NLE must be able to develop and implement its own ingest, preservation, storage and access workflows;
  - NLE can set up (or change existing) variety of quality control rules on Information Packages (such as allowed file formats, metadata rules, virus checks, AIP and storage integrity controls etc).
  - NLE can manage all logical components of its digital archive (Publisher portal, KROOL, GOOBI) in one preservation solution.
- The new digital archive provides state-of-the-art long-term preservation functionality:

[3] Please note that under current legislation NLE is unable to make use of commercial cloud-based digital preservation services outside of EU.

- It provides ingest, storage, characterization, active preservation options and access to a variety of file formats and data types (most importantly digitised and born-digital publications including audio-visual components, content of the NLE web archive);
- It applies clearly and precisely defined input(s) with maximum of 4-5 accepted SIP structures (all other systems are customized according to these structures);
- It enables detailed mapping of ingested content (*know your data*) and supports identification, validation, and processing of major library file formats (Please see Table 10. Digital Archive integrations with other systems);
- It provides methods for managing the storage, including random checks on the integrity and (continuous) presence of files;
- It supports receiving descriptions in (at least) Dublin Core format, including syncing descriptions with Sierra/ESTER and KROOL;
- It must be possible to search for data / information packages in the digital archive based on Dublin Core descriptions;
- It allows the pre-generation of the digital content to the application layer, including creating DIP for current Digar / DEA / web archive presentation layers / viewers;
- It supports mass import and mass export of data in open format.
- It keeps an overview of the archive versions, copies and storage sites (this also applies if the volume of data reaches to PBs).
- It provides a multitude of means for monitoring, analysing and reporting on the content stored within the repository;
- It allows to carry out a large-scale active preservation actions in an efficient and secure manner;
- It allows NLE to use/implement current and future digital archiving best-practices and digital preservation community effort (such as shared active preservation workflows, file format registries, environment descriptions, digital preservation tools etc), including offering the ability to manually intervene with workflows (by NLE admin).
- It provides intgrations and/or built-in components to support text and data mining, as well as linked open data applications and data exchange;
- It provides an administrator's user interface for the NLE’s digital archive employees.

- The new digital archive implements a modular architecture, making it possible to replace individual modules or components as needed:
    - It is, to the largest possible extent, built on following recognized standards in the area of digital preservation (such as OAIS, PREMIS) and digital content management;

- It implements standardized and documented API(s) to allow easy integration with external components (such as pre-ingest, access, (meta)data management and content manipulation tools);
- It's API(s) allow(s) to address at least the information package as a whole, possibly also the information package component itself (e.g., one file);
- It supports, by default, a variety of metadata standards (such as Dublin Core, MODS, MADS, MARC21, EAD, EDM, RDF (derived) from BIBFRAME, common protocols for metadata harvesting (such as OAI-PMH, Z39.50, SRU/SRW), multiple ingest and access structures (such as METS, CSV, ONIX, XML) and integrates a variety of tools for content identification and active preservation (such as Jhove, FITS, DROID, veraPDF, epubcheck);
- It enables to replace, update and/or add components easily;
- There are clear principles for exchanging and renewing individual components.

- The new digital archive is scalable, secure, stable and sustainable:
  - It is capable of handling a digital archive in size of multiple PBs (i.e. capable of preserving all current and foreseeable future NLE digital content in one environment[4]);
  - It supports scaling ingestion, for example parallel launch of multiple ingest workflows (with appropriate hardware) and managing the ingest queues;
  - It is possible to be successfully certified as a Trusted Digital Repository;
  - It meets the requirements of the Estonian IT Baseline Security System ISKE level M (https://www.ria.ee/en/cyber-security/it-baseline-security-system-iske.html);
  - It provides means for tracking/logging all preservation actions undertaken with NLE digital content within the preservation platform;
  - The core digital preservation platform is well-established, tried out and tested, and used in several institutions comparable to NLE and its data volume;
  - The platform has a clear development roadmap which takes into account emerging best-practices and innovative approaches in digital preservation and ICT in general;
  - The vendor provides possibilities for its user base to interact with each other and to influence the development roadmap of the software components;
  - The vendor offers technical support to resolve problems as they occure (quickly);
  - The vendor has at least one strong external development partner from who NLE can order larger developments if needed;

[4] Please note that future data volume forecast can be found in table 5.

- It is possible for NLE digital archive to offer storage and active preservation services to external clients (for example other Estonian memory institutions).

2.3.1 The future process scope of digital archive

To achieve described goals, the new digital archive solution must be able to support the following OAIS framework components:

1. **INGEST**

   (please also refer to functional requirements section 2 in paragraph 3.1.1)

   1.1. Receive SIP

   1.1.1. Receive SIP from Publishers portal

   1.1.2. Receive SIP from docWorks

   1.1.3. Receive SIP from GOOBI

   1.1.4. Receive SIP from Heritrix/KROOL

   1.1.5. Receive SIP from API (mass ingest projects)

   1.2. Validate received SIP structure, file formats and metadata

   1.3. Generate technical and administrative metadata

   1.4. Generate AIP for long-term preservation

   1.5. Generate metadata

   1.6. Send package description to archive database

   1.7. Send AIP to storage

2. **DATA MANAGEMENT**

   (please also refer to functional requirements section 4 in paragraph 3.1.1and non-functional requirements section 11and 12 in paragraph 3.1.2)

   2.1. Register package metadata

   2.2. Manage metadata updates (e.g., from ESTER/Sierra, re-ingest[5], recharacterization, active preservation, etc.)

   2.3. Delete package description / metadata

   2.4. Perform queries based on Access restrictions

   2.5. Provide package description / metadata for requests from access process

   2.6. Generate reports (e.g., statistical reports)

   2.7. Administer database based on established standards and policies (e.g., preservation policy)

[5] Re-ingestion means ingesting (meta)data from a source file, which is already preserved, but have updated metadata or contents, which adds value to the source file. Re-ingestions are managed by versions- both, updated and old versions are available in digital archive.

3. **ARCHIVAL STORAGE**

(please also refer to functional requirements sections 3 and 5 in paragraph 3.1.1 and non-functional requirements section 6 in paragraph 3.1.2)

3.1. Receive / store AIP

3.2. Update / store AIP

3.3. Delete AIP

3.4. Provide AIP

3.5. Invoke storage policy

3.6. Manage storage locations and hierarchy (must be able to define various storage locations of information packages)

3.7. Check storage integrity and auto-recovery

3.8. Replace / refresh media (data) drives

3.9. Backup and disaster recovery

4. **ACCESS**

(please also refer to functional requirements section 6 in paragraph 3.1.1)

4.1. Receive and process DIP requests (through API)

4.2. Generate DIP (generic)

4.3. Create dissemination package (DIP) for Digar

4.4. Create dissemination package (DIP) for Veridian and DEA

4.5. Create dissemination package (DIP) for public user interface of Estonian Web Archive

5. **ADMINISTRATION**

(please also refer to non-functional requirements section 3 in paragraph 3.1.2)

5.1. Establish Standards and policies (which is not in scope of the IT system)

5.2. Manage system configuration

5.2.1. Get ingest, preservation and access statistics

5.2.2. Get Data Management reports

5.2.3. Manage ingest, preservation and access configurations

5.3. Manage digital archive access rights

5.4. Perform physical Access control (which is not in scope of the IT system)

5.5. Audit Information Packages (SIP, AIP, DIP)[6] (which is not in scope of the IT system)

5.6. Invoke AIP updates

[6] Includs periodical review of Information Package structure and manual check if the data in the database is saved correctly based on the predefined structure of Information Packages.

5.7. Manage dissemination policies

6. **ACTIVE PRESERVATION PLANNING**

(please also refer to information packages requirements in paragraph 3.3)

6.1. Assess preservation risks

6.2. Assess migration risks

6.3. Develop SIP, AIP, DIP designs and migration plans

6.4. Develop preservation strategies and standards

6.5. Monitor designated community (based on surveys)

6.6. Monitor technology (technology alerts, external data standards, reports etc)

2.4 Current volumes and future forecast of digital archive usage

In this paragraph we have outlined statistics about current and future volumes regarding the usage of NLE digital archive.

Table 1. Number and volumes of archived objects by object type

Archived objects	In 2017		In 2018		Total by the end of 2018	
Object type	Number of objects	Total volume (GB)	Number of objects	Total volume (GB)	Number of objects	Total volume (GB)
Digital archive DIGAR (Fedora)						
Journals	3 880	1 493	11 324	9 798	45 178	28 631
Newspapers	10 469	409	10 196	874	129 358	8 235
Serials	639	980	662	816	5 115	3 193
Sound recordings	346	207	425	195	2 738	1 391
Maps	775	198	949	427	2 638	925
Manuscripts	1	11	6	40	92	629
Other	12	7	12	231	248	324
Sheet music	73	90	150	176	951	1 233
Ephemera	681	205	816	99	2 865	638

Posters	2 668	562	2 824	501	9 182	1 848
Postcards	2	0,4	11	0,98	7 257	304
Books	5 260	13 274	6 205	17 402	26 982	52 446
Standards	178	0,49	188	0,32	2 827	8
DIGAR Estonian articles, shortly DEA						
Periodicals	61 206	5 800	145 995	6 100	368 201	20 000
Estonian web archive						
Web pages	73 348	11 647	91 668	14 441	228 945	29 284
Screenshots of websites' frontpages	434 553	430	714 772	690	1 149 764	1 121
TOTAL	**594 091**	**35 313**	**986 203**	**51 791**	**1 982 341**	**150 210**

Table 2. Number and volumes of archived objects by main file types based on percentage of total archived volume

Archived files	**In 2017**		**In 2018**		**Total by the end of 2018**		
File type	**Number of files**	**Total volume (GB)**	**Number of files**	**Total volume (GB)**	**Number of files**	**Total volume (GB)**	**Percentage of total archived volume**
Digital archive DIGAR (Fedora)							
TIFF	581 225	16 173	1 051 935	28 628	3 813 524	90 165	59,6%
PDF	121 476	1 089	164 374	1 726	2 179 697	8 238	5,5%
All others	20 266	174	24 648	206	89 278	1 402	0,9%
DIGAR Estonian articles, shortly DEA							

All (image files, PDF-s, xml-s etc)	552 571	5 800	1 040 813	6 100	3 424 455	20 000	13,2%
Estonian web archive							
WARC	150 852	11 553	209 575	14 335	489 669	28 083	18,6%
PNG/JPG (screenshots)	434 553	430	714 772	690	1 149 764	1121	0,7%
MP4, WEBM (videos)	0	0	28 814	2 240	28 814	2 240	1,5%
TOTAL	**1 860 943**	**35 219**	**3 234 931**	**53 925**	**11 175 201**	**151 249**	**100%**

Table 3. Archived object sizes

	In 2017	**In 2018**	**Total by the end of 2018**
Digital archive DIGAR (Fedora)			
Digitised			
Number of Titles	6 235	7 701	38 275
Capacity (TB)	16	29	89
Born-digital			
Number of Titles	3 959	4 106	20 102
Capacity (TB)	1	1	11
DIGAR Estonian articles, shortly DEA			
Number of Titles	509	957	1 925
Numbers	61 206	145 995	368 201
Pages	552 571	1 040 813	3 253 517
Articles	523 408	599 426	5 702 533

Capacity (TB)	6	6	20
Estonian web archive			
Number of Web sites	6 140	23 125	95 303
Capacity (TB)	11	14	29

Table 4. DIGAR, DEA and Estonian Web Archive presentation layer users and page views (based on Google Analytics data for DIGAR.ee, DEA.digar.ee and veebiarhiiv.digar.ee)

	In 2017		**In 2018**		**Total by the end of 2018**
	All per year	**Max per day**	**All per year**	**Max per day**	**All**
www.digar.ee*					
Users	180 904	1 230	221 859	1 535	726 954
Unique page views	1 199 893	9 479	1 340 287	8 460	5 064 889
dea.digar.ee**					
Users	586 626	4 099	727 470	5 479	2 193 147
Page views	1 920 250	10 397	2 569 924	16 440	7 663 458
veebiarhiiv.digar.ee					
Users	20 332	603	29 952	590	91 796
Page views	143 406	4 207	175 583	4 409	681 874

* total amount is given for the period 2013-2018

** total amount is given for the period 2014-2018

Table 5. Estimated Growth in Data

Year	Total data volume (TB)
2019	507

2020	827
2021	1 212
2023	2 402
2025	4 812
2030	13 347

2.5 Scope of implementing Digital Archive

The scope of the current tender and the project is procurement and implementation of the new Digital Archive solution in Estonian National Library.

The specific tasks to be accomplished during the project (scope of the project) is determined as follows:

- Delivering the digital archive software licenses (if relevant);
- installing digital archive software on the infrastructure of NLE
- configuring and parameterizing the digital archive software according to requirements defined in this technical specification and additional instructions given by NLE;design and/or configuring of digital archiving workflows;
- configuring preservation logic (S3 and tape recording eg);
- integration of digital archive software with external systems (Publisher Portal, DocWorks, Goobi, library system (currently ESTER), EMS, KROOL, DEA, Digar);
- data migration from current archiving solutions Fedora, DEA (METS/ALTO and image files) and Krool (web archive);
- training employees of NLE regarding the provided digital archive software;
- technical support and License support for 5 years.

Digital archive must be integrated to the existing NLE environment. The integrations needed to be made with other systems are outlined in paragraph 3.2 alongside with information needed for data migration. Ingest, preservation and access workflows are described in paragraph 2.1.

3. Technical description of the procurable digital archive solution

3.1 Functional and non-functional requirements for the Digital Archive

Functional and non-functional requirements for the Digital Archive are presented in the following tables. The information serves as one of the assessment categories for the proposal.

3.1.1 Functional requirements

Functional requirements for the Digital Archive are presented in the following table.

For each requirement the expected vendor response is as follows:

- Is the requirement fulfilled by the system “by default” (yes/no)
- If not, is it possible to fulfil the requirement during the implementation of the product (f.ex. through additional development within or external to the product, customisation of default components etc)
- If yes, what is the cost of the additional development or customisation
- Respondents are welcome to provide additional clarifications

Example:

Table 6. Functional requirements example table

Req no	REQUIREMENT	Is available by default	Is possible to add	Effort needed to add	Comment
1.1	The digital archive must provide means for checking the integrity of information packages	Yes. By default the system calculates hashes for all content			
1.2	The digital archive must be able to validate EPUB files	No.	Yes. It is possible to integrate epub-check into the system	20 hours	

Table 7. Functional requirements table

1	WORKFLOW MANAGEMENT	Is available by default (Yes/No)	Is possible to add (Yes/No)	Effort needed to add	Comment
1.1	The system must be based on general workflow logic – activities undertaken in digital archiving can be modelled as workflows.				
1.2	The system must provide a GUI for setting up and customising the details of workflows:				
	a. Input / output parameters of workflow steps;				
	b. Specify internal or external components being called within a step;				
	c. (Error) reporting and workflow halt/continuation criteria;				
	d. File conversion based on files technical attributes (f.ex. to not convert black and white TIFF to TIFF 6.0).				
1.3	The system must provide a GUI for monitoring the activity and progress of currently running workflow instances				
1.4	The system must provide a GUI which allows to manually address errors in workflows, cancel or restart them.				
1.5	Detailed documentation must be available about the creation of new workflows and tasks				

1.6	The system must identify, validate, normalize, migrate, extract metadata				
1.7	The system must validate SIP structure, file format and metadata				
1.8	The system must provide output for queries in the following file formats: PDF, EPUB, JPEG2000, MP3, TIFF, PNG, WARC, WAV, METS/ALTO, MXF, MOV, MP4, AVI, M4A, WebM, Daisy, TXT, XML and provide extendibility to add formats in the future.				
1.9	The system must allow to implement new (versions) of third-party components and address these as individual workflow tasks.				
1.10	The system must be scalable and capable of running multiple workflows in parallel.				
1.11	The system must provide customizable standard workflow describing which files will be converted to long-term preservation formats.				
1.12	NLE employees must be able to define file format conversion rules. For example, long term preservation formats for AIP:				
	a. All PDFs → PDF-A/1b				

	b. All TIFFs → TIFF 6				
	c. ARC, WARC → WARC				
	d. PNG, JPEG → JPEG2000 (NLE employee can define rules which files to convert into JPEG2000)				
	e. MP4				
	f. WebM				
	g. WAV				
	h. iBooks				
	i. EPub				
	j. MP3				
2	**INGEST**	**Is available by default (Yes/No)**	**Is possible to add (Yes/No)**	**Effort needed to add**	
2.1	The system must provide workflows suitable for the ingest of core data flows and file formats of the NLE. For further details please consult chapter 3.3.4.				

2.2	The system must provide at least the following range of tools for receiving data files and their metadata:				
	a. Web-based GUI for preparing and uploading SIPs manually				
	b. Monitored drop folders				
	c. Network transfer (sFTP)				
	d. Open and well documented API including a "GetSIP" type service				
	e. Harvest with the OAI-PMH protocol				
2.3	The system must be able to integrate with and receive SIP from the following systems NLE uses: Publisher Portal, DocWorks, Krool, Goobi.				
2.4	The system must support the ingest of SIPs formatted according to the current versions of METS and BagIt specifications. In case the system supports a specific implementation of METS, it is registered as an official METS Application Profile				
2.5	The system must provide at least the following quality assurance checks within the ingest workflow:				

	a. SIP integrity check (validate packages and their components against submitted checksums),				
	b. SIP validation (validate packages against the expected SIP format),				
	c. virus check and (optional) quarantine,				
	d. metadata format validation,				
	e. file format identification,				
	f. Checking the SIP against a list of archival file formats,				
	g. file format validation.				
2.6	The system must provide metadata format validation for at least the following formats:				
	a. Dublin Core,				
	b. MODS,				

	c. MADS,				
	d. MARC21,				
	e. PREMIS				
	f. EAD,				
	g. EDM,				
	h. RDF from BIBFRAME				
2.7	The system must provide common protocols for metadata harvesting such as:				
	a. OAI-PMH,				

	b. Z39.50,				
	c. SRU/SRW				
2.8	The system must provide multiple ingest and access structures such as:				
	a. METS,				
	b. CSV,				
	c. ONIX,				
	d. XML				
	e. JSON				

2.9	The system must provide configurable quality assurance checks for ingested files.				
2.10	Ingest workflows must be able to include automated conversion or normalisation steps for binary files. For example, it must be possible to develop a workflow which automatically converts files into appropriate archival file formats.				
2.11	The system must provide means for manually reviewing and updating the SIP if needed.				
2.12	The ingest workflow must be capable of handling Information Packages including multiple file groups / representations of the Information Object (see chapter 3.3.2 or further details about file groups)				
2.13	The ingest workflow must be capable of creating an AIP which includes multiple file groups / representations (which may be already submitted within the SIP or created during the ingest process)				
2.14	The ingest workflow must automatically extract and create relevant technical, administrative and preservation metadata regarding all files included into the AIP and all actions done during the ingest process				
2.15	The system must allow to pull additional metadata from external knowledge bases (i.e. EMS, VIAF)				

2.16	The system must be capable of storing, validating and reporting the success of storing information packages regardless of the means of storage (i.e., for both database and physical disk storage).				
3	**ACTIVE PRESERVATION**	**Is available by default (Yes/No)**	**Is possible to add (Yes/No)**	**Effort needed to add**	
3.1	The system must provide means for assessing preservation risks / formats at risk.				
3.2	The system must be, at any point in time, capable of showing the volume of content stored in the system affected by a specific preservation risk.				
3.3	The system must be capable of recharacterising [7] stored content and recalculating preservation risks.				
3.4	The system must provide means to carry out the Preservation Planning function and help the digital archivist to evaluate and select suitable active preservation methods and plans.				
3.5	The system must provide means for executing migration operations across the whole digital archive or based on criteria specified by the digital archivist				

[7] In time T1, object is characterised with f. ex Jhove ver 1.4 and collected metadata MD1. In time T2, Jhove version 1.7 is installed and Digital Archive recharacterises object to get improved metadata MD2

3.6	The system must be capable of storing migrated content as a new file group / representation of the information object				
3.7	The system must provide means for automatically validating the success of migration actions regarding the core content types of NLE (monographic publications, periodical publications, digitised content, web archive).				
3.8	The system must allow scaling out for migrations affecting millions of files and information packages.				
3.9	The system must support the digital archivist in recording metadata of suitable emulation environment (or reference to an external record).				
3.10	The system must provide native support for the PRONOM file format registry.				
4	**DATA MANAGEMENT AND ADMINISTRATION[8]**	**Is available by default (Yes/No)**	**Is possible to add (Yes/No)**	**Effort needed to add**	
4.1	The system must be able to register and store metadata (administrative, preservation, technical, descriptive) in both the system's database and within physical information packages.				
4.2	The Archival Information Package (AIP) implemented by default within the system must be open source and well documented,				

[8] Please note that additional requirements for NLE information packages are available in chapter 3.3

	NLE has full access to the documentation				
4.3	The AIP specification must clearly define the possibilities for configuration and extension (f.ex. for adding any NLE custom metadata within the information package).				
4.4	Metadata stored within the database must be possible to be updated by both automated workflows and manually by authorised NLE staff.				
4.5	Metadata editing must be configurable in the system, f.ex assigning specific edit and view permissions to specific information packages and types of metadata.				
4.6	The system must allow to define rules for when it is possible to overwrite metadata, and when it is only possible to create a new version of the metadata record				
4.7	The system must allow to define rules for when it is possible to define new preservation file formats, and when it is only possible to create a new version of the file format.				
4.8	The system must allow to store files in their original state without migrating into supported long-term preservation formats. There must be extracted as much metadata as possible from the files and shortcomings must be reported (preservation risks, detected errors, etc).				
4.9	The system must provide automatically persistent and unique identifiers to information packages and its components (information objects, file groups / representations, files,				

	metadata records)				
4.10	The system must provide customisable means for automatic and manual classification and reclassification of information packages For example, NLE might choose to classify legal deposit objects according to their primary publisher. It must be possible to develop a workflow which classifies the object correctly according to provided SIP metadata, and/or to add or change this classification manually during Ingest or at a later point in time.				
4.11	The system must be able to perform queries and provide package descriptions based on access request.				
4.12	The system must provide a report building functionalities (e.g. possibility to build statistical reports).				
4.13	The system must be able to conceal preservation data from external parties in order to create dark archives.				
4.14	The system must be able to provide collections with configurable access restrictions.				
4.15	The system must be able to provide GUI for external parties, through which dark archive files can be ingested and extracted.				
5	**STORAGE**	**Is available by default (Yes/No)**	**Is possible to add (Yes/No)**	**Effort needed to add**	

5.1	The system must support a variety of means for passive storage, incl at least:				
	a. Simple file storage				
	b. Tape storage				
	c. Cloud storage				
5.2	The system must be capable of managing passive storage in the scale of multiple petabytes (PBs)				
5.3	The system must provide effective and scalable means for setting up integrity checking activities across all online storage components.				
5.4	The system must notify NLE staff about any problems found during integrity checking and offers possibilities for automated recovery.				
5.5	The system must provide means for setting up storage policies (i.e. how many copies of information packages have to be stored and on which storage components. Different policies can be set up for different types of content / ingest workflows.				
5.6	The system must provide effective and scalable means for migrating all content from a storage component to another (i.e. to execute the OAIS Replace Media function).				

5.7	The system must provide possibility to store multiple versions of AIP and provide access to them upon request.				
6	**ACCESS**	**Is available by default (Yes/No)**	**Is possible to add (Yes/No)**	**Effort needed to add**	
6.1	The system must provide at least the following means for access:				
	a. A simple web based access environment for browsing and viewing all stored content (which NLE has authorized access) within the digital archive.				
	b. A well documented API for serving DIPs to external access portals[9] (Digar, DEA, Web Archive viewer)				
6.2	The system must provide built-in viewers for at least the following file formats: PDF, EPUB, JPEG2000, MP3, TIFF, PNG, WARC, WAV, METS/ALTO, MXF, MOV, MP4, AVI, M4A, WebM.				
6.3	The system must provide file format extendability to built-in viewer.				

[9] Please note that current viewers Digar, DEA and Web Archive viewer will be replaced in the future with one common viewer. If you offer a viewer that is integratable with Digital Archive and is aslo capable of displaying Web Archive file formats then please specify this as well in your answer to this question.

6.4	The system must be able of storing DIPs (or dissemination representations) within the system. The DIPs (or dissemination representations) must be clearly separateable from AIPs – it must be possible to understand for both human users and automated workflows which IPs are intended for preservation and which for dissemination.				
6.5	The system must allow authorised NLE staff to set up conditional retention policies for DIPs.				
6.6	The system must allow to set up at least the following access workflows / scenarios:				
	a. Creating presentation copies during ingest and sending these to external access portals (Digar, DEA, Web Archive viewer).				
	b. Creating individual presentation copies automatically on-demand (i.e. based on access requests forwarded from external access portals and APIs).				
	c. Manual creation of presentation copies (potentially including tasks like metadata editing/selection, anonymization, etc)				

	d. Bulk creation or replacement of presentation copies				
6.7	The system must support the creation and management of dynamic collections[10] for access purposes:				
	a. It must be possible to add one object into several collections				
	b. It must be possible to link various objects in collections.				
6.8	The system must support following collections:				
	a. Dynamic collections formed in search results;				
	b. Static collections formed in metadata - references to another collections and / or objects exist in metadata;				
	c. Hierarhical collections - collections of collections;				
	d. Subcollections				
6.9	The system must support changing order of files / objects in collections				

[10] Dynamic collections are created of pointers to each component. For example, dynamic collections formed in search results.

6.10	The system must provide full text versions of content and metadata files				
6.11	The system must have an ability to perform full text search (of metadata and content files).				
6.12	The system must support OAI-PMH protocol and adding Internet Protocol restrictions to limit the access.				
6.13	The system must provide permanent links on file, collection and object level.				
6.14	Permanent links must be usable in integrated systems (f.ex link must be migrateable to search engine and direct back to the preserved file)				
6.15	The system must provide means for mass data extraction (f. ex data mining)				
6.16	The system must provide access restrictions in file level				
6.17	The system must support 3 scenarios of DIP creation:				
	a. Generating DIP in ingestion phase				
	b. Pre-generating DIP from AIP and storing DIP before use request				

	c. Generating DIP by request for viewing “on the fly”				
6.18	System must provide possibility to collect data from CRM and support SSO access control.				
6.19	The system must support machine-to-machine communication.				

3.1.2 Non-functional requirements

Non-functional requirements for the Digital Archive are presented in the following table. The information serves as one of the assessment categories for the proposal. Please note that some of the requirements might not be relevant for your product. Respondents are expected to highlight which non-functional requirements they fulfil and how.

For each requirement the expected vendor response is as follows:

- Is the requirement fulfilled by the system “by default” (yes/no)
- Comment(s) detailing the exact way how compliance with the requirement is achieved

Example:

Table 8. Non-functional requirements example table

Req no	REQUIREMENT	Is supported by default (Yes/No)	Comments
1.1	Application, database and third-party components platform(s)/version(s) end-of-life (EOL) cannot be less than 2 years.	Yes.	
1.2	The application must delete all temporary files from the server immediately if they are no longer used.	No.	All temporary files are deleted from the server only after they haven’t been used for 30days.

Table 9. Non-functional requirements table

1. GENERAL		Is supported by default (Yes/No)	Comments
1.1	Application, database and third-party components		

	platform(s)/version(s) end-of-life (EOL) cannot be less than 2 years.		
1.2	All transferable versions of the application must be tested before handing over to the customer. The test plan and test results must be forwarded to the customer together with the transfer of the application.		
1.3	The application must delete all temporary files from the server immediately if they are no longer used.		
1.4	Supported technology: Integration platforms: WSO2 Enterprise Service Bus Application servers: Apache Tomcat, Oracle WebLogic Operation systems: SUSE Linux Enterprise Server 64-bit, CentOS 64-bit Load divider: Apache Database platforms: Oracle; PostgreSQL (SLES) Database propagator: Liquibase Database-based web applications: Oracle Application Express (ApEx) Code repository: GIT Java application building tools: Gradle; Maven Java application logging: SLF4j + logback Tools for performance testing: Jmeter; Selenium; Gatling Excel format generation: Apache POI-XSSF (http://poi.apache.org/spreadsheet) For low-critical systems and upon agreement: Apache, MariaDB, PHP Content Management Systems (CMS): Drupal PHP Framework: Laravel Image Framework: IIIF Storage: Local S3 Minio		
1.5	Application resource requirements beyond which there is a need for separate agreement: Typical storage capacity per application per node: Max storage capacity 2GB per node (max 2 node count) Database connections max. number: max 25 database connections per node Session max size per user: 2MB Maximum file size when uploading: 10M (larger files require packing and splitting).		

1.6	Web browser support: On standard screen devices (desktop computers, laptops, etc.): Firefox, Safari, Chrome, Internet Explorer, Edge and other web browsers supported by the manufacturer at the time of the implementation. For small-screen mobile devices (phones, tablets, etc.), Chrome, Safari and other web browsers supported by the manufacturer at the time of the implementation.		
1.7	URLs with reserved or global restrictions: / info * | / BrainLogin * | / health * | / test *Browser support minimal		
2. STANDARDS & METHODS		**Is supported by default (Yes/No)**	**Comments**
2.1	The database and application must use UTF-8 encoding.		
2.2	Communication between applications must be realized through services. Appropriate interface formats are agreed upon during the project (REST, SOAP, JMS).		
2.3	Cryptographic algorithms and methods should be prepared based on the current life cycle study of cryptographic algorithms. The current version of the study is available on the RIA website in Estonian (https://www.ria.ee/sites/default/files/content-editors/publikatsioonid/kruptograafiliste_algoritmide_elutsukli_uuring_2017.pdf).		
2.4	PHP development should be based on the standards and recommendations offered by PHP Framework Interop Group. Standards and recommendations are available on the PHP website at http://www.php-fig.org/psr/		
2.5	Drupal developments should be based on the coding standard offered by Drupal. Standards are available on Drupal's website https://www.drupal.org/docs/develop/standards		
2.6	PHP source code style must conform to the common PSR-2 standard. PSR-2: Coding Style Guide is located at https://www.php-fig.org/psr/psr-2/. Analyzing the code must not cause errors at the Error or Warning level.		
2.7	To enable the user interface automatic testing, identifiers (IDs) must be used in HTML tags (tag).		

3. AUDIT AND SECURITY		Is supported by default (Yes/No)	Comments
3.1	The products software components must comply with ISKE security class K1T2S1, security level M – medium (https://www.ria.ee/en/cyber-security/it-baseline-security-system-iske.html).		
3.2	The product's software components must be possible to be audited against IT security standards (f.ex. ISO27000 series).		
3.3	The product's software components must be possible to be certified as trusted digital repositories.		
3.4	If the product does include user authorization mechanisms, it must allow: limiting the number of unsuccessful logins per minute/hour/day from one IP address; according to the agreement with the customer the usage of captcha, account lock or delay time; limiting access to admin interfaces only from specific IP addresses.		
3.5	The location of the file in the system must not be passed on to the client.		
3.6	In the case of authenticated user sessions between the client and the server, there must be a session setting up an encrypted HTTPS protocol. The default is HTTPS. HTTP protocol can be used between the load divider and the application server.		
3.7	All database entries / tables containing information classified with ISKE integrity = 2 must be versioned.		
3.8	All data changes must be maintained at the base. When the user edits data then the data is not deleted, instead a new entry is made with the new data and old is marked as invalid. Each new entry must contain the following information: reference to the entry which it has invalidated (if any); the user who created the item; time of item creation; session ID (if available).		

	Each invalidated item must have the following information: a user who has revoked an item; date of record invalidation.		
3.9	The application must not use database activities that require DDL rights (Dynamic-link library).		
3.10	The application and data must only be accessed through documented and customized paths and documented authentication procedures. There must be no other way of accessing applications or databases.		
3.11	All passwords must be saved in the application only in encrypted form. Passwords can only be in an unencrypted form temporarily on the remote RAM. Unencrypted passwords may not be temporarily stored on any disk. The encryption must be at least equivalent to the AES256 algorithm.		
3.12	If the ISKE security class is 2 or higher, the application must show the last successful login time after the successful login. If unsuccessful login attempts have occurred then the system must also show when they occurred and how many unsuccessful login attempts there was. The user must be able to make sure that someone has not logged in under his or her name. The system does not need to show failed attempts when ID card, mobile ID or digital ID was used for logging in.		
3.13	Ending sessions must be done on the server side and all applications must have a configurable user session expiration time. Time must be configurable with other configuration parameters. If the customer has not received any queries within the specified time then the session must be terminated on the server's own initiative.		
3.14	Data entered or otherwise transmitted by users to the application must be cleaned with an XSS filter or HTML tags removed. Preferably applied before saving to the database but definitely before displaying the data.		
3.15	Forms sent by web-based applications must have a hidden hash that is checked when the form is received in order to prevent CSRF attacks.		
3.16	For encryption and/or hash calculation strong algorithms should be used. Respectively AES-256, RSA-2048, SHA-2 or stronger.		
3.17	Authenticated session ID should not be solved with a simple cookie. It must not be possible to take over a session by copying a session ID from one computer to another.		

3.18	When using Active Directory (AD) authentication, the application must also use restrictions tied to the AD account. For example: password expired, account locked, account expired, etc.		
3.19	The system is not allowed to store user information on application login forms.		
3.20	If users are also managed in the application and authenticated over AD then the user must first check users' access rights in the application and only then turn to AD. The aim for this is to reduce AD load.		
3.21	To ensure the security of the application (for example, XSS, SQLInjection, etc.), OWASP best practices must be followed. (https://www.owasp.org/index.php/Cheat_Sheets)		
3.21	The application may not allow multiple simultaneous sessions for the same user.		
3.22	The application may only accept session keys that it has issued itself. Upon logging in, the user must receive the new session key and the former key must be revoked.		
3.23	When the files are uploaded to the application then the file type must be checked and verified and the files must pass through the antivirus check. The antivirus solution is provided by the RMIT Maintenance Department. File storage is used to upload files.		
3.24	When files are uploaded by the user to the application then the file name must include a random component when the file is saved, so that the path of the file is not easy to be guessed.		
3.25	When files are downloaded by the user to the application, these files must be validated. For example, an uploaded file should not refer directly to other resources in the file system.		
3.26	The web application may not leave temporary files containing non-public data on the user's workstation when it is closed.		
3.27	The data entered by the user must be filtered before displaying. If possible, whitelisting should be used instead of blacklisting. Special characters must be filtered according to the environment in which these data are viewed (html; console, etc.).		
3.28	The application does not store data on the client's computer (including cookies). The exception is session cookie and the choice of language for		

	a multilingual system.		
3.29	Iframe is not allowed and "X-Frame-Options: DENY" header must be added to HTTP requests.		
3.30	The application cannot respond to HTTP queries that the application does not actually use in its work. For example, if an application uses a GET query for url /hello, the application cannot respond to the same POST query if it is not used in the application's work.		
4. AUTHENTICATION		**Is supported by default (Yes/No)**	**Comments**
4.1	Users must be authenticated by using native system (f.ex via application, which is already installed to library). If the functional requirements determine the creation of a new authentication solution, a new one can be used.		
4.2	In addition to the mobile number, it is mandatory to also ask personal identification code (ID code) for the Mobile ID.		
4.3	If the users of the application are outside of the domain resources managed by the RMIT then the application must allow authentication also by an Estonian ID card and a Mobile ID, also Smart ID. If such a proposed application does not support authentication with an ID card or a Mobile ID then the offer must specify an alternative solution with the equivalent price costs incurred by the vendor for implementation, deployment, administration, etc. This supports the overall objective that the proposed solution should be adapted to the existing environment so that it works fully in that environment.		
5. PERFORMANCE		**Is supported by default (Yes/No)**	**Comments**
5.1	The product needs to be able to process and store the current and foreseeable future data volumes (please see Table 1).		
5.2	The products performance needs to comply with agreed topology (incl. storage capacity) and outlined performance indicators (please see Table 2).		
5.3	Application launch (site restart, configuration change, etc.) should take place within a reasonable time (max 30 sec). The application must reach the running status of the server during this time. Longer activities should be done after this if necessary.		

5.4	The application must include a script for conducting performance tests that allow to identify the stress test / load test that is tolerated by the application and to explain how the application behaves with a load test. When updating an existing application, the corresponding script must also be updated. The exact description of the performance tests and the tools to be used must be agreed upon in detail analysis. The developer must supply the script with the application and the necessary software tools for conducting the agreed performance tests. Conducting performance tests may not require the customer to develop software, write scripts or purchase licenses.		
5.5	The product must be able to set up and manage archival bulk operations in a stable, efficient and timely manner, f.ex. processing at least 10 TB of data per day - from ingest to storage, including: a. Ingest of hundreds of GBs of content; b. Storage integrity checking for a PB-size archive; c. Mass-migrations for several hundreds of GB of content.		
6. MONITORING & LOGGING		**Is supported by default (Yes/No)**	**Comments**
6.1	The application must have at least two monitoring sections. One must show application status and version to the monitoring system, and the other must show application dependencies. Monitoring shows statistics for both in size and server load. Accurate monitoring services and format are agreed upon with the customer.		
6.2	It must be possible to see the application version number (for example on a monitoring page) from the application interface.		
6.3	The application must have an administrative interface through which the application administrator can perform normal administrative operations (user management, menu management, etc).		
6.4	If an application uses non-administrative services (such as ID authentication, use of a bank link, etc.), it must be able to keep records of the respective service volume usage. Usage information must be stored at a minimum of log level for all external services. In the case of essential services, the relevant information must be available to the average user independently of the application. In the case of such services, this part must also be written out in functional requirements.		
6.5	The application must log the start and end of the session, the user Internet Protocol address, the authentication method (ID card, mobile ID, etc.). In the case of a successful authentication, the user's personal		

	identification code and, in the case of a mobile ID, a telephone number should be logged. It must be possible to direct the security log to a separate file.		
6.6	The application must allow logging of all outbound and incoming HTTPS queries. It must be possible to switch the logging on/off separately, ie independently from the same level logging. This instruction must be included in the application's installation guide. The goal is to simplify software debugging and solve production environment problems.		
6.7	The application must log any technical errors that occur in the application either in the file or in the database. The log must contain in minimum (in the order shown): a. the time of the error, b. the error code, c. the error description (stack trace, traceback, etc.), d. user data (name, ID, Internet Protocol address and URL), e. HTTP, GET and POST parameters and their values.		
6.8	Logging database (including encrypted logs) must be stored out of the operational base so that the application remains operational when removing old records. For example, it is important that the log structure allows deleting old records and does not implement a business logic that prevents the deletion of old records.		
6.9	For user accounts, a separation of functions must be ensured. Data updates and changes made to the information system from a specific user ID must be uniquely associated with that user ID.		
6.10	The following log level descriptions should be used: a. Fatal – Can only be used when an error occurs that does not allow the application to work (eg no N. base conf). b. Error – Technical errors (eg no connection, base errors, etc.). There are no business logical errors. c. Warning – Problems of a technical nature that the system can recover itself (eg try again). d. Info – Calling out important services and objects in terms of business logic. Information and error messages displayed to the user (data stored, input incorrect etc.). Notifications of data changes with the primary key of the object.		

	e. Debug – Information for the developer about the system status. f. Trace – All SQL queries with parameters and other detailed information if necessary.		
6.11	The application logs must not contain special characters that may distort the log format or appearance. They need to be replaced in advance (escape). Eg line change signs; ANSI escape codes.		
6.12	The solution provides means for tracking/logging all preservation actions undertaken with NLE digital content.		
7. CONFIGURATION		**Is supported by default (Yes/No)**	**Comments**
7.1	The application server must be able to operate on a separate server which is independent from the database server.		
7.2	All application components must be capable of operating at high speed and work in cluster with high availability. Components of the supplied application (user interface, services, etc.) and functionality can be used in the active cluster. Database clustering is generally not required.		
7.3	The user session of the application must not be based on the cluster node. The user must be able to proceed even if his session for some reason goes to another node.		
7.4	The business-critical interfaces of the application must be fail-safe. In the case of malfunctions of external interfaced systems, the system must not freeze but issue an error message within a reasonable time. Whenever possible, asynchronous interfaces must be used to increase reliability. The reasonable time must be configurable from the system configuration.		
7.5	The installation unit must be installed unchanged in the application servers of all environments. The configuration of the file path application must be specified inside the installation unit.		
7.6	If the configuration contains sensitive parameters then they must be compiled into separate files, divided by environment.		
7.7	Files must be cataloged on the basis of agreed features. Features must be selected so that no more than 1,000 files are created in one folder.		

7.8	The names of configuration parameters must be substantive. If this is not possible, there must be an explanation next to it.		
7.9	Configuration files must not be visible to the end user.		
7.10	Different parameters with the same meaning should not exist in the configuration		
7.11	All parameters should only be described once in the configuration, not with the same parameters being repeated in several files or sections of the file.		
7.12	Communication between the client and the database must be through the application server. The client application must not connect directly to the database.		
7.13	All environmental parameters must be configurable from the configuration (numeric security parameters, e-mail server parameters, LDAP (AD) configuration parameters, database connection parameters, etc.).		
7.14	The application may not use methods that exclude shared hosting usage.		
7.15	The application must have separable services that are accessed by users from those services that other services/servers access (can be installed on a separate server).		
7.16	The application is forbidden to write to the standard outputs of the application server. The configuration of RMIT environments must direct the entire log of the application being delivered to the rotating files.		
7.17	The technical documentation of the application must focus on the central service layer of the application, where the public methods of all service interface interfaces with the JavaDoc documentation must be covered. Additional description of the central service layer provides a quick overview of the basic functionality of the application on the basis of API documentation.		
7.18	The configuration files that depend on the applications' work environment must be located in separate directories in the source repository.		
7.19	The product must be able to support multi-tenancy solutions. For example, multiple external clients must be able set up their own specific		

	access rights and security configurations.		
8. WEB		**Is supported by default (Yes/No)**	**Comments**
8.1	The product's user interface needs to comply with the WCAG 2.0 level AA. http://www.w3.org/TR/WCAG20/		
8.2	The product's user interface needs to fully comply with current HTML5 and CSS standards. CSS content must be valid (http://jigsaw.w3.org/css-validator/)		
8.3	The application must be able to be moved without re-programming between different domains and domain sites. Must not use absolute URLs.		
8.4	Uniform resource identifier (URI) length must not exceed the maximum allowed value of any supported browser.		
8.5	The app may only use relative URLs/paths. If a full-length path is required, it should be described as a configuration parameter in one place and the information required should be used from there.		
8.6	The total size of the components of a single website may not exceed 500 kB (including HTML, scripts, style sheets, pictures, WARC files, screenshots etc).		
8.7	If the application requires JavaScript to be launched in the user's web browser, but JavaScript is not supported in the user's browser then the system must display an understandable error message to the user.		
9. E-MAIL		**Is supported by default (Yes/No)**	**Comments**
9.1	The product must use an external mail server to send e-mails. When sending a message, the application must make sure that the server received the e-mail. Server data must be configured with configuration parameters without rebuilding the application		
9.2	E-mails to invalid e-mail addresses should not be sent. The e-mail address must comply with RFC5322 and/or RFC6854. Before sending the e-mail the system needs to check the format even if it is already being validated during data entry.		

10. USER INTERFACE		Is supported by default (Yes/No)	Comments
10.1	When entering data, the application must always check whether the text the user enters matches the field type and/or the preset values. It must not be possible to insert inappropriate text. This control must also be performed on the server side. The user can only enter numbers in the number field, date in the date field, e-mail address in the e-mail address field, only allowed values for enumerated input forms and so on. If the input control fails then the system must stop using the input and inform the user.		
10.2	All user interface design decisions must be agreed upon with the customer. *(Applies to additional developments)*		
10.3	The user interface must always ask for confirmation of data deletion and mass changes.		
10.4	The content of system error messages may not be displayed to the end user. Along with the error the system must display a unique error identifier that can be found in the logs. The user should be given the most accurate message possible about which action failed from the user's view. (For example, not "database connection error" but "saving failed"). The error identifier allows a specific user to associate a problem with technical error logs.		
10.6	The texts/translations of multilingual user interfaces must be separate from the code and design. Adding a new language must be possible from the configuration file or from the administration interface.		
10.7	The application user interface must inform the user about the expiration of the session. The notification time must be configurable without rebuilding the application.		
10.8	If the form consists of many small data fields (eg an application), the form is divided into stages and stored at the end of the respective stage.		
10.9	When entering data to the input forms then the user must be able to move between the fields by using the tabulator TAB according to the business logic.		
10.10	For interactive forms the action (such as uploading a file) must not be repeated by refreshing the page (uploading the file, sending data,		

	submitting the application, etc).		
10.11	For Internet Protocol queries with a duration of more than 3 seconds, the user must see that the request is being processed. This is mandatory to avoid unnecessary repetition of users' activity.		
10.12	The browser navigation buttons must work in the application but must not repeat the data alteration activities. It must be possible to move forward/backwards between at least one non-altering view (for example search results pages).		
10.13	The application may not open new browser windows. The application itself does not use pop-up windows. As an exception, links from the application may open in a new window (using target = "_ blank").		
10.14	The system must provide feedback on user activities. For example, "Successfully saved", "Action not allowed", etc.		
10.15	For queries that burden the server, the application must limit unnecessary repetition of the same activity. If> 10sec (load test plan) is enabled at the time of the request, you must limit the pressing of the button until the page is reloaded, for example.		
10.16	Screenshots that require moving the display horizontally for reading must be avoided.		
10.17	When a list is displayed, pagination must be used, where the number of items agreed upon at one time is displayed. The number of records per page must be configurable.		
10.18	The application must alert the user about the older browsers that have lost their support and are not supported by the IT profile (anything besides Firefox, Safari, Chrome, Internet Explorer ja Edge).		
10.19	The application must provide the user with clear, relevant and understandable error messages.		
11. DATABASE		**Is supported by default (Yes/No)**	**Comments**
11.1	The database must use indexes and other measures to ensure that future performance requirements are met according to planned lifetime and amount of data (please refer to Table 1, Table 2, Table 4 and Table 5).		

11.2	Query variables (Parameter Binding) must be used. Inquiring SQL queries from outside the database must use query variables to avoid SQL cache fragmentation and SQL injection attacks.		
11.3	The database must support both cold and hot backup (mirroring) in another service room. Services that exclude database mirroring (eg "filestream") must not be used.		
11.4	The operational base of the application must be kept to a minimum. For large amounts of data, if the business process allows, data archiving outside the base must be used.		
11.5	Database object names may contain only Latin alphabets, numbers, and underscores. Not allowed to use dots.		
11.6	If new items are added to the database during the delivery, new items must be listed separately in the delivery description. Required for application administrators to subscribe to new tables for viewing rights.		
11.7	The application must be created and delivered so that the owner of the application data in the database and the user of the connection to the application database can be assigned separately.		
11.8	The objects are not saved into application database as a file format. Files are stored in a file repository which is outside of the database.		
11.9	Object dependencies in the database are kept as simple as possible without the need to use WITH GRANT OPTION privileges.		
11.10	Database schemes must be in one language throughout.		
11.11	Large objects are not stored in the application's operational database. Appropriate alternatives will be agreed upon during the project.		
11.12	All the creation and modification of the database tables and spreadsheet fields must be described with meaningful and up-to-date information in the database in the relevant commentary, so that it explains their purpose. The data object comment code must be delivered with DDL.		
12. VERSIONS & UPDATES (Applies to additional developments)		**Is supported by default (Yes/No)**	**Comments**

12.1	Every new version of the system must always include release notes. Release notes must reflect any changes between the previous and the new version.		
12.2	The developer must make sure that the changes are made to the current code. Before making the changes, the last code is taken from the RMIT code repository.		
12.3	The installation package that comes with the installation guide may only contain the file set required for running the minimal application. For example, compiled languages should not contain source code, specific components of the test environment and so on.		
12.4	It must be possible to move the installation package produced under the installation instructions between different machines. For example, you should not create a situation where it is absolutely necessary to compile the application on the new server where it is being run.		
12.5	The source code compilation must also be possible in the absence of an external network connection. The required dependencies must be on the local network (RMIT Maven for Java).		
12.6	The code to be installed must be located in the RMIT code repository. External dependencies whose code has not been changed are located in Maven.		
12.7	Only source code, design elements (images, icons, templates) and database scripts are uploaded to the repository. Loading large files (libraries, database dumps, etc.) to the repository is prohibited.		
12.8	If the base service is not installed with Liquibase then the database updates are transmitted as one master script, which triggers sub-scripts if necessary. The sub-scripts must be referenced to the relative path exactly as they are in the version management.		
12.9	It must be possible to install base updates on database administrator (DBA) privileges. All basic objects in the scripts contain a schema name.		
12.10	Installing base updates must generate a log. In case of Oracle's case a spool file must be generated.		
12.11	It must be possible to install the new delivery/version repeatedly without breaking the data. Commit to use as little as possible and as much as necessary to make a defective delivery rollback. Restarting DML must prevent duplicate entries from being constrained.		

12.12	All database scripts must be in UTF-8 encoding (without BOM) format.		
12.13	Delivery may not include dependencies on the SNAPSHOT versions of java libraries.		
12.14	Database changes must be executable from the command line and not be packaged within the application. It must be possible to export the changes as a script (also afterwards). This allows to install and control basic changes separately. In addition, this eliminates potential parallelism issues within the cluster. If the solution used does not meet the requirements, then base changes must be forwarded as scripts.		

3.2 Requirements of Digital Archive integrations

The Digital Archive is connected with many systems to provide data preservation. The connections with external systems, as Publishers Portal, DocWorks, Goobi, ESTER, EMS, KROOL, DEA, DIGAR etc. must be established. In addition to the ingestion, these integrations are needed for usability and viewability of the preserved data.

The list of systems which need to be integrated to the Digital Archive is as follows:

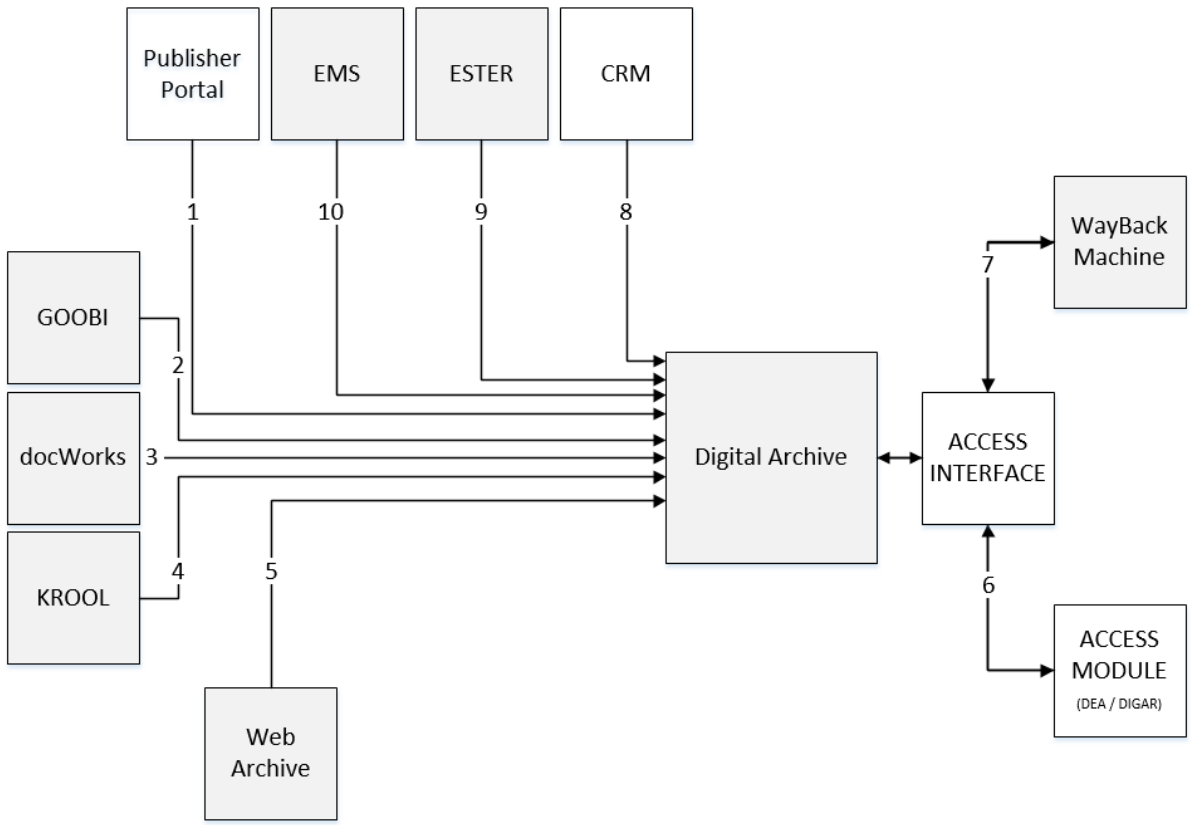

Figure 4. Systems need to be integrated with digital archive

Systems which are still in development phase, are presented with a white box. Each system integration is marked with a number. Following legend maps the data formats used to integrate with Data Archive:

1. Copyrights and access restrictions; original files (PDF; PDF/A-2b; EPUB; MP3; TIFF; PNG; WAV; MXF; MOV; AVI; M4A; 3D objects; JPEG2000; Rare and unknown formats)
2. Raw files (TIFF; PNG; EPUB; PDF; PDF/A-2b)
3. Pub Code; processed files (Text; PDF; PDF/A-2b; EPUB; TIFF; JPG; JPEG2000; METS/ALTO; PNG; …)
4. Web archived files (WARC; MP4; PNG; WebM; JPEG)
5. Raw files, crawl log (WARC; LOG)
6./7. Metadata, Archived files (PDF; PDF/A-2b; EPUB; MP3; TIFF; PNG; WARC; WAV; MXF; MOV; MP4; AVI; M4A; WebM; EPUB; METS/ALTO; JPEG2000; 3D objects …) **Web archived files; view files** (WARC; PNG; JPEG; MP4, WebM)
8. Login info
9. Bibliographic record (MARC21; XML)
10. Subject terms(MARC21; MARCXML; XML)

In addition to the previous list of systems that need to be integrated with Digital Archive, the following table outlines all the systems which the Digital Archive uses in its operations (including the ones needed to be integrated with). For each system the list of files and file formats regarding data exchange with Digital Archive is outlined in the Table 10.

Table 10. Digital Archive integrations with other systems

System	File	Format	Needs to be integrated with Digital Archive?
Publisher Portal	Rights and restrictions metadata	.JSON	Yes
	CIP record	MARC21	
	Original files	.PDF .PDF/A-2b .EPUB .MP3 .TIFF .PNG .WAV .MXF .MOV .MP4 .AVI .M4A .iBooks MOBI 3D objects JPEG	

System	File	Format	Needs to be integrated with Digital Archive?
		JPEG2000	
EMS	Subject terms	MARC21 MARC XML	Yes
ESTER	Bibliographic record	MARC21	Yes
	Pubcode (with no diacritic marks)	Text	
	• Original files	.PDF .PDF/A-2b .EPUB .TIFF .JPG JPEG2000 METS/ALTO .XML	
DocWorks	Display files	METS/ALTO Mets.XML ALTO.XML .PDF .JPEG .ePub .TXT JPEG2000 .TIFF .PNG .JPG	Yes
	Group record: • Acronyms.xml • Access.xml Publication.xml	 .XML .XML .XML	
	Original files	.TIFF .JPEG .PNG .JPG	
	Original file metadata	METS.XML ALTO.XML	
GOOBI	Display files	.JPG .TIFF .PNG .EPUB	Yes
	Object metadata	.XML METS.XML ALTO.XML	
DEA / Veridian	Display files	METS.XML ALTO.XML .JPEG JPEG2000 .TXT .PDF .PDF/A-2b .EPUB	Yes

System	File	Format	Needs to be integrated with Digital Archive?
		.IBOOK .MOBI .MP3 .TIFF .PNG .WAV .MXF .MOV .MP4 .AVI .M4A .WebM .EPUB 3D objects IIIF support	
DIGAR	Display files	.PDF .PDF/A-2b .EPUB .MP3 .TIFF .PNG .WAV .MXF .MOV .MP4 .AVI .M4A .WebM .EPUB METS.XML ALTO.XML JPEG2000 3D objects .MOBI .TXT	
	Crawled original files	WARC .XML (DublinCore) .WebM .MP4	Yes
	Object metadata files	WARC .XML (DublinCore) .JSON	
Web Archive KROOL	Screenshots	.PNG JPEG2000	Yes
	WayBack display files	WARC	
	Authentication information	.XML	

System	File	Format	Needs to be integrated with Digital Archive?
WayBack			Yes
(New) Access Module[11]	Object metadata	Dublin Core	
	Display files	.WebM .MP4 .PNG JPEG2000	
CRM[12]			Yes

3.3 Requirements for information packages

This paragraph includes:

- Requirements for NLE information packages (SIP and AIP);
- Conceptual model of NLE information packages;
- Overview of current FOXML packages (for migration purposes) and
- Requirements for information packages for specific content types (digitised content, periodical and monographic publications, web harvest snapshots).

The information below serves as one of the assessment categories for the proposal. Respondents are expected to:

- highlight which information package requirements they fulfil and how;
- provide a general description on how their physical SIP and AIP models map to the proposed conceptual model;
- provide a clear description of the foreseen issues and cost of migration;

[11] New Access Module is planned to be developed which should replace DEA / DIGAR in the future.

[12] NLE CRM is planned to be developed.

- provide a clear description of the implementation of information packages for the NLE content types.

3.3.1 Requirements for NLE information packages

Requirements for NLE information packages are listed in a table below:

Table 11. NLE information packages requirements table

Requirement		Vendor response
1	***The information package does not limit the types of data or metadata which can be used within it.*** While the information package might include specific mandatory components (f.ex. require administrative metadata to be in a specific format), it must not limit the use of any file format or any additional metadata.	
2	***The information package does not propose a specific relation to information objects.*** In most cases the current NLE information package equals one information object (i.e. one publication, one issue of a newspaper, one website harvest, …). However, the information package format shall also allow creation of information packages which include multiple information objects (f.ex., series of monographic publications) or parts of information objects (f.ex., one chapter of a digitised book).	
3	***The information package allows inclusion of multiple file groups or representations*** NLE must have the possibility to include multiple file groups or representations into an information package (f.ex., draft files of a publication, print file submitted to NLE, file normalised within NLE, migrated file after an active preservation event, dissemination copy in low resolution etc). In addition, the platform must support setting up preservation, management and reporting actions according to these file groups (f.ex. allow for finding “formats at risk” only within file groups destined for active preservation, allow for creating file format statistics per file groups, setting access permissions per file groups etc).	

Requirement		Vendor response
4	***The information package supports metadata versioning*** In case of updating metadata for/within an Information Package it must be possible to include it as a new version / representation of metadata (as opposed to overwriting previous metadata entries).	
5	***The information package provides persistent and unique identifiers for all its components*** The information package provides, by default, persistent and unique identification for the package itself, all its file groups / representations, individual files and bitstreams, individual metadata sections (descriptive, technical, administrative, structural) and their versions. It is possible, by default, to address these components for repository management and dissemination purposes (f.ex. in order to provide only one file group of a package to the end-user viewer).	
6	***The information package provides built-in means for integrity checking and alteration detection*** In addition to the functionality of the platform, the information package itself provides metadata or other means which can be used to check the integrity and/or detect the alteration of the package and its components.	
7	***The information package is capable of holding extremely large files or information objects*** NLE is already facing scenarios where information packages must hold individual extremely large files (f.ex. a 80 GB video file) or information objects (the largest web archive package is currently 533 GB, consisting of ca 300 files). The information package must be capable of including the abovementioned current largest objects of NLE and have the potential to include even larger files / objects in future.	
8	***Ingest of METS SIPs including Dublin Core descriptive metadata is supported by default*** Submissions which have formatted according to the METS specification are by default supported by the platform. The platform provides also a clear description about the preferred	

Requirement		Vendor response
	ways of using METS and/or Dublin Core as the SIP. These descriptions are preferably in the form of an officially registered METS Application Profile.	
9	***AIP exists as a physical package*** While the preservation platform might offer other means for storing an AIP (f.ex. as a set of database entries), it shall also offer (by default) means for storing the AIP as a physical package.	
10	***The AIP follows METS, PREMIS and Dublin Core specifications*** The AIP is either: a) directly built on METS, PREMIS and Dublin Core specifications or b) possible to be converted to METS, PREMIS and Dublin Core in a straightforward and lossless way. As such the AIP supports the: • description of METS components (package identification, administrative and descriptive metadata, identification of file groups and files, creation of structural maps); • description of PREMIS entities (objects, agents, events, rights) • relating these descriptions to intellectual objects described both within the information package according to the Dublin Core specification and also in the external library catalogue. Preferably the specific details of using METS, PREMIS and Dublin Core are clearly documented in the form of an officially registered METS Application Profile.	
11	***The physical AIP is human-readable*** The physical AIP is implemented in a way that the package is accessible to a digital preservation expert with standard text editors / readers and sufficiently understandable for the purpose of restoring or restructuring the data / metadata within the information package. It is also possible to address the information package with text-based scripts (f.ex. in order to restore the administrative metadata database based on the	

Requirement		Vendor response
	content of information packages).	
12	***The AIP format does not limit the use of any active preservation method*** The AIP format is sufficiently open to support, for example: • the description of migration actions and the inclusion of multiple representations of the object, • support the inclusion of (PREMIS) descriptions for suitable emulation environments set up internally; • allow for the description and referencing to external emulation environments (i.e Emulation as a Service type solutions).	
13	***The information package allows inclusion of user generated data or metadata*** The structure of the information package is sufficiently open to allow for the inclusion of data or metadata created by the users of NLE (f.ex. transcriptions or tagging done within crowdsourcing projects). It is possible to clearly highlight and separate the "official" portions of the package provided by NLE and the "user-content".	
14	***The information package allows inclusion or referencing to derivative information products*** The information package allows for the inclusion of derivative information products, like OCR results or data mining outcomes within the package; or the inclusion of references to and descriptions of the derivative products.	
15	***The information package allows inclusion of RDF representations of data and/or metadata*** It is possible to include (Linked) Open Data representations of data and/or metadata into the package in RDF format.	

3.3.2 Conceptual model of NLE information packages

NLE recognises that specific Information Package implementations within software platforms differ. It is not the intention of NLE to force the respondent to change the built-in implementation. However,

NLE proposes in this section a conceptual model which summarises key requirements outlined above and describes the core data and metadata components required by the NLE at this point in time.

NLE expects respondents to provide a clear mapping of their physical implementation to this conceptual model and highlight any areas or components which might be problematic and require an alternative (conceptual) approach.

The conceptual model is based on a layered approach towards information packages where the layers are:

- information package itself;
- information object (i.e., a publication, website crawl, etc);
- file group (one named representation of the information object consisting of one or many files);
- file (computer file stored separately on storage media);
- component file (computer files packaged within the file);
- bitstream (individual identifiable content with format, embedded within the file or component file, f.ex., images embedded into a PDF document).

The relations between these layers are as follows:

- (standard scenario) 1 information package includes 1 information object;
- (exception) n information packages include 1 information object or 1 information package includes n information objects;
- 1 information object includes 1..n file groups
- 1 file group includes 1..n files
- 1 file includes 0..n component files
- 1 file or component file includes 0..n bitstreams.

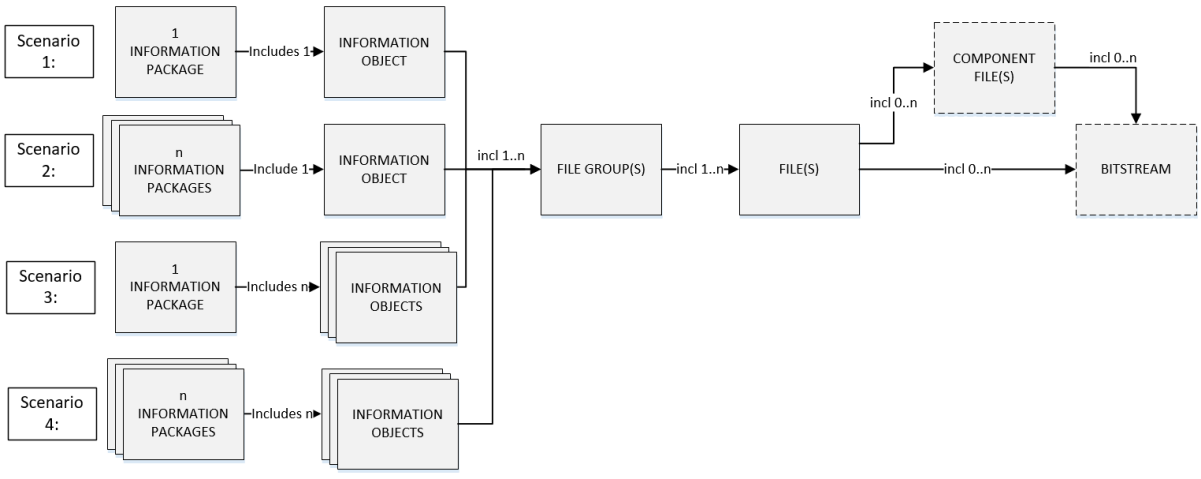

Figure 5. NLE information package and its layers

The minimal metadata portions which must be possible to be recorded on these layers are as follows:

Component / layer	***Metadata type***	***Cardinality[13]***
Information Package	Persistent unique identifier	Mandatory
	Name of the Information Package	Optional
	IP creation details (when, who)	Mandatory
	IP modification details (when, who, reference to previous versions, if appropriate)	Mandatory if applicable
	Workflow / content context (i.e. web harvest, monographic publication, periodical publication, digitised work)	Mandatory
Information Object	Persistent unique identifier	Mandatory
	Primary descriptive metadata (Dublin Core)	Mandatory
	Secondary descriptive metadata (any other format next to Dublin Core, f.ex MARC21), user or artificial intelligence created metadata etc	Optional
	Collection identifier(s) which the information object belongs to	Mandatory if applicable
	Rights metadata (generic statement for the information object: open to all, open in reading rooms only, closed)	Mandatory
	Metadata modification details / log (when, who has updated metadata about the information object within the package)	Mandatory if applicable
File group	Persistent unique identifier	Mandatory
	Name of the file group	Optional
	Purpose of the file group (based on a built-in or preset classifier, f.ex. “submission”, “preservation 1”, “preservation 2”, “dissemination 1” etc)	Mandatory
	METS structMap (or comparable) describing the order of files within the group	Mandatory
	Rights and licence metadata specific to the file group (available to end-users, within NLE)	Mandatory
	File group creation and modification details / log (when, who)	Mandatory
File, component file	Persistent unique identifier	Mandatory
	Technical metadata (mime type, checksum, size, creation date, modification date, etc)	Mandatory
	PREMIS (detailed) technical metadata	Mandatory
	PREMIS event and provenance metadata	Mandatory if applicable
	PREMIS technical environment metadata	Optional

[13] Note that the values “optional” and “mandatory if applicable” describe the intended use of appropriate metadata groups (i.e. metadata groups which are not expected to be available for all objects). All the metadata groups outlined in the table MUST be supported within the proposed solution.

Component / layer	*Metadata type*	*Cardinality*[13]
Bitstream	PREMIS (detailed) technical metadata	Optional

3.3.3 Overview of current FOXML packages

As of now most NLE digital content is stored within Fedora in its native Fedora Object XML format (FOXML, https://wiki.duraspace.org/pages/viewpage.action?pageId=66585857). The exception is the content of the NLE Web Archive which is stored as separate WARC files, managed by the custom KROOL application and database.

Below you can find a brief description of the core elements within the NLE implementation of the FOXML package. For further details please consult the full FOXML example files attached to the document.

- At root level each package includes a PID which has been formatted according to the scheme nlib-digar:*nnnn* (example: “nlib-digar:16”)
- objectProperties includes generic metadata about the package and the object (name, created and modified dates);
- the FOXML package includes multiple “datastreams”:
    - **AUDIT:** generic descriptions about events / actions done within Fedora and Operaator.
    - **DC and NLIB_DC:** Parallel description of the object in standard and NLE specific Dublin Core. Please note that multiple versions of both (DC and NLIB_DC) might exist. In case of changes to the description the previous version is maintained within the FOXML file and a new version is created. All metadata versions carry an identifier (f.ex DC.1, DC.2, DC.3 etc) and a version date.
    - **ARHIIV:** datastream for “archival” computer files. This datastream includes limited technical metadata (size, hash) and an identifier which can be used to locate the actual file (note, that the packages do not include the file itself or its path). Using numeric extensions / sequal identifiers the computer files are implicitly grouped into one or many object representations.

      Example: FOXML package includes two representations each consisting of three computer files. Datastream ID’s are as follows: ARHIIV_1_1, ARHIIV_1_2, ARHIIV_1_3; and ARHIIV_2_1, ARHIIV_2_2, ARHIIV_2_3.
    - **ORIGINAAL:** none, one or many “original” or submission computer file. This datastream includes limited technical metadata (size, hash) and an identifier which

can be used to locate the actual file (note, that the packages do not include the file itself or its path). The use of numeric extensions / sequal identifiers is similar as for the ARHIIV datastreams. Note that for digitised content no ORIGINAAL files exist within the package!

- **TECH_ARHIIV and TECH_ORIGINAAL:** full technical metadata about the "archival" and "original" files. The technical metadata portions follow the same identifier convention as for the file datastreams (i.e. TECH_ARHIIV_1_1, TECH_ARHIIV_1_2 etc).
- **RELS-EXT:** references to the collection(s) the object belongs to. Note that collections in Fedora are also FOXML objects which carry a similar identifier (i.e. nlib-digar:nnnn);
- **RIGHTS:** metadata about reuse restrictions (closed, open only in reading rooms, open to all), date(s) until which the restrictions apply; and information about the owner/depositor of the object;
- **STRUCTMAP:** the structural map of the package. The structural map joins ARHIIV and ORIGINAAL files into file groups / representations (following the example above, the structural map would define two file groups each consisting of three files/datastreams). Further the structural map defines whether the scope of a file group / representation is "store" or "view", defines the mapping between file datastreams and technical metadata datastreams; and defines the order of files (i.e. for a digitised book which file presents the first, second, etc page).

3.3.4 Information package implementations for NLE core content types

As described above the five main content workflows of NLE are the following:

- Born-digital **monographic publications** accepted through the *Publisher Portal*;
- Born-digital **periodical publications** submitted by publishers (in future can also be accepted through the *Publisher Portal)* and further processed with the *docWorks* software component;
- **Digitisation of monographic publications**;
- **Digitisation of periodical publications**;
- **Websites.**

Below you will find short descriptions of the specific setup and file formats for information packages including these content types. The intention of the NLE is to maintain, for the foreseeable future, the current setup of file formats and further requirements. Any exceptions are specially highlighted below.

We expect the solution providers to use the information below in order to:

- State that it is possible to create information packages with the same scope;
- Assess the difficulty and cost of data migration.

3.3.4.1 Born-digital monographic publications

The main objects of born-digital monographic publications according to medium are listed below:

- **Books** (incl. e-book, sound recording (audiobook) can appear: in print, Braille', on a physical medium (CD, DVD, USB), online edition, pdf, e-pub, html, Mobi, Kindle
- **Videorecording** can be published on a physical medium (CD, DVD, USB), online publication
- **Sound recording** can be published on a physical medium (CD, DVD, USB), online publication
- **Multimedia** can be published on a physical medium (CD, DVD, USB), online publication
- **Software, database** can be published on a physical medium (CD, DVD, USB), online publication
- **Map, atlas** can be published in print, Braille', on a physical medium (CD, DVD, USB), online publication (pdf, html)
- **Sheet music** can be published in print, Braille', on a physical medium (CD, DVD, USB), online publication (pdf, html)
- **Songbook** can be published in print, Braille', on a physical medium (CD, DVD, USB), online publication (pdf, e-pub, html, Mobi, Kindle)
- **Standards** can be published in print, Braille', on a physical medium (CD, DVD, USB), online publication (pdf, e-pub, html, Mobi Kindle)
- **Poster** can be published in print, Braille', online publication (pdf, html)
- **Ephemera** (such as the printed materials of everyday life, generally regarded as having little or no permanent value because they are produced in large quantities or in disposable formats for a specific limited use, advertising publications, programs and methodical materials, information publications, theatre programs, crossword publications and Sudoku, textless coloring and sticker books, election materials, publishers lists, reports) can be published in print, Braille, online publication (pdf, html)
- **Art literature** (photos, graphic documents etc.) can be published: in print, Braille', on a physical medium (CD, DVD, USB), online publication (pdf, html)
- **Manuscript** can be published in print, Braille', on a physical medium (CD, DVD, USB), online publication (pdf, html)

An information package including born-digital monographic publications submitted to the digital archive contains the following components:

- **„draft" files**:

- The Publisher Portal allows publishers to submit any content the publisher regards as relevant. This might include high resolution images, maps, spreadsheets, MS Word documents and other content in their initial format;
- Technically “draft” files can come in any file format;
- The “draft” files are specially marked in the Publisher Portal and are only included into the ORIGINAAL file group;
- NLE does not include “draft” files into preservation or access representations of the object, the files are currently stored “as-is”.

- **Print files (Output-ready files of printed publications):**
  - Print files are the complete and final file(s) which the publisher uses for printing or electronically disseminating the publication;
  - Print files have to follow NLE requirements for archival file formats, typically print files are delivered as PDF files;
  - Initial print file(s) are included into the ORIGINAAL file group.
- **Archival files:**
  - The ingest workflow undertakes additional normalisation of the pdf file(s). This results in the ARHIIV file group;
  - In case of preservation actions (additional normalisation) or ingest updates (f.ex. replacing an invalid file after the initial ingest) executed within the digital archive additional ARHIIV file groups are created.
- **Descriptive metadata:**
  - A full bibliographic record (in MARC21 format and based on previously ingested record, which has identical ID) is created during the pre-ingest phase and recorded into ESTER/Sierra;
  - During ingest some basic descriptive metadata from the catalogue is converted into Dublin Core format and included into the information package;
  - After ingest the digital archive identifier of the information object is added to the linked bibliographic record in ESTER/Sierra.
- **Other metadata:**

- Metadata about access restrictions, copyright and the submitter/publisher of the publication are added within the Publisher Portal and submitted to the digital archive using a custom NLE metadata schema;
- There is no further administrative, preservation or technical metadata being created within the Publisher Portal. All such metadata is to be created within the ingest process of the digital archive;
- The structural map of the object is present in all current packages.

- **Classification / collections**
  - Some monographic publications might be part of a (permanent, static) collection. Currently these collections are created within Operaator and stored by Fedora Commons as FOXML objects, the information package of the publication includes references to the collection object.

3.3.4.2 Born-digital periodical publications

The main objects of born-digital periodical publications according to medium are listed below:

- **Journal** can be published in print, Braille, on a physical medium (CD, DVD, USB), online publication (pdf, html)
- **Newspaper** can be published in print, Braille, on a physical medium (CD, DVD, USB), online publication (pdf, html)
- **Serials** (annual report, annual yearbook, almanac, proceeding, serial publication) can be published in print, Braille, on a physical medium (CD, DVD, USB), online publication (pdf, e-pub, html, Mobi, Kindle)

An information package including born-digital periodical publications contains the following components:

- **Print files**: **(Output-ready files of printed publications)**
  - Typically the publishers of periodical publications have access to a dedicated FTP site where they can upload the print files on a daily/weekly/monthly etc basis;
  - Print files have to follow requirements for NLE archival file formats. Typically print files are delivered as PDF files;
  - Submitted print files are stored as the initial ORIGINAAL file group within the information package.

- **Archival files:**
    - Submitted print files undergo segmentation within the DocWorks application and are saved as segmented JP2000 + METS + ALTO combinations;
    - These PDF files are included into the current information package as the ARHIIV file group.
- **Access files:**
    - Within the DocWorks application the submitted print files are processed to become content in JPEG2000 and METS/ALTO formats *(a jpeg2000 file is created for each page; the jpeg2000 files undergoe an OCR process; all jpeg2000 files are joined into a single PDF file constituting the whole publication which includes an OCR text layer beneath an image layer).*
    - *PS! The JPEG2000 and METS/ALTO files are currently NOT included into the information package. The requirement of NLE is to change this (i.e. to start including the JPEG2000 and METS/ALTO files as another representation / file group of the object within the information package).*
- **Descriptive metadata:**
    - All issues have the same descriptive metadata (of the newspaper). Descriptive metadata is input only once during the submission of the first issue;
    - For all consequent issues descriptive metadata in Dublin Core format is input into the pre-ingest process as pre-prepared XML files and is as such also available to the digital archive ingest process.
    - After the ingest of the first issue of a newspaper the digital archive identifier of the newspaper collection is added to the linked bibliographic record in ESTER/Sierra (note: an identifier is NOT sent to ESTER for each individual issue but for the whole collection).
- **Other metadata:**
    - Metadata about access restrictions, copyright and the publisher of the periodical publication are input into the pre-ingest process as “metadata templates” and submitted to the digital archive using a custom NLE metadata schema;

- There is no further administrative, preservation or technical metadata being created within the pre-ingest process. All such metadata is to be created within the ingest process of the digital archive;
- Each information package includes a structural map which describes the various file groups and the order of files within the file group.

- **Classification / collections**
  - All periodical publications are structured according to a hierarchical classification: publication / year / month / date and edition number.

3.3.4.3 Digitised monographic publications

An information package including monographic publications digitised with the help of the Goobi application contains the following components:

- **Digitised master files:**
  - Monographic publications are mostly digitised as master TIFF files (each scanner view as a separate TIFF file, mostly one view includes 1 page of the original book, exept in case of tables, illustrations are larger than 1 page;
  - Strict naming conventions are used for files, where the first part of the file name is the unique ID of the publication (according to the syntax: “b12345678”) and the last part of the file name constitutes the order of the file (f.ex the 11th file including pages 20-21 of a digitised publication is named: [uniqueID]_0011.tif);
  - All master TIFF files are included into the first ARHIIV file group (ARHIIV_1);
  - *Note that there is no ORIGINAAL file group present within the information package of the digitised publication!*

- **Edited TIFF files:**
  - Master TIFF files are edited (f.ex cropped to size) by NLE staff and saved as TIFF files;
  - All edited TIFF files are included into the second ARHIIV file group;

- **PDF representation:**
  - All TIFF files are merged into a single PDF file constituting the whole publication;
  - The TIFF file undergoes an OCR process, resulting in a PDF file which includes an OCR text layer beneath an image layer;

- The PDF file (and special content if relevant) is included to become a third ARHIIV file group

- **Special content:**
  - In some cases publications include special content (f.ex. audiovisual extras, photos etc). Such special content is digitised separately and added as “loose” files into the information package as a separate ARHIIV file group(s).
  - Please note that all file groups are file format specific (i.e., it is not allowed to add mp3 files into a TIFF file group). As such, there might be a number of ARHIIV file groups with specifial content.

- **Digitised audiobooks:**
  - In the case of digitised audiobooks the numbering an naming of ARHIIV file groups is as follows: ARHIIV_1 for wav files, ARHIIV_2 for txt files, and ARHIIV_3 for tif files.

- **Descriptive metadata:**
  - A full bibliographic record (in MARC format) is already available within the ESTER/Sierra;
  - During ingest of the digitised content some basic descriptive metadata from the ESTER/Sierra is converted into Dublin Core format and included into the information package;
  - After ingest the digital archive identifier of the information object is added to the linked bibliographic record in ESTER/Sierra.

- **Other metadata:**
  - Metadata about access restrictions, copyright and the publisher of the publication are added manually ~~in XXX~~ and submitted to the digital archive using a custom NLE metadata schema;
  - There is no further administrative, preservation or technical metadata being created within Goobi or the digitisation process. All such metadata is to be created within the ingest process of the digital archive;
  - The structural map of the first ARHIIV group describes the order of the TIFF files (i.e. pages of the publication).

- **Classification / collections:**

- Some monographic publications might be part of a (static, permanent) collection. Currently these collections are created within Operaator and stored by Fedora Commons as FOXML objects, the information package of the publication includes references to the collection object.

3.3.4.4 Digitised periodical publications

An information package including digitised periodical publications contains the following components:

- **Digitised TIFF files:**
  - Periodical publications are digitised as master TIFF files (each scanner view as a separate TIFF file);
  - Strict naming conventions are used for files, where the first part of the file name is either the pubcode of the publication or the name of the publication (for serials), the year and number of the issue; and the last part of the file name constitutes the order of the file (f.ex file 11 of a digitised newspaper: [nameOfNewspaperYearAndNumberOfIssue]_0011.tif);
  - All master TIFF files are included into the first ARHIIV file group.
  - *Note that there is no ORIGINAAL file group present within the information package of the digitised publication!*
- **Edited TIFF files:**
  - Master TIFF files are edited (f.ex cropped to size) by NLE staff and saved as TIFF files;
  - All edited TIFF files are included into the second ARHIIV file group;
- **PDF representation:**
  - All TIFF files from either the first or second ARHIIV file group are merged into a single PDF file constituting the whole publication;
  - TIFF files undergo an OCR process, resulting in a PDF file, which includes an OCR text layer under an image layer;
  - The PDF file is included to become a next ARHIIV file group.
- **Access files:**

- In the case of digitised newspapers, the TIFF files are sent for processing into the DocWorks application to create a METS/ALTO representation (saved as segmented JP2000 + METS + ALTO combination);
- *PS! The image and METS/ALTO files are currently NOT included into the information package which is sent for storage into Fedora. The requirement of NLE is to change this (i.e. to start including the JPEG2000 and METS/ALTO files as another representation / file group of the object within the information package).*

- **Descriptive metadata:**
  - All issues have the same descriptive metadata (of the newspaper). Descriptive metadata is input only once for the whole newspaper;
  - Descriptive metadata in Dublin Core format is input into DocWorks processing as XML files and is also available to the digital archive ingest process.
- **Other metadata:**
  - Metadata about access restrictions, copyright and the publisher of the periodic publication are input into DocWorks processing as "metadata templates" and submitted to the digital archive using a custom NLE metadata schema;
  - There is no further administrative, preservation or technical metadata being created within the pre-ingest process. All such metadata is to be created within the ingest process of the digital archive;
  - Each information package includes a structural map which describes the various file groups and the order of files within the file group.
- **Classification / collections:**
  - All periodic publications are structured according to a hierarchical classification: publication / year / month.

3.3.4.5 NLE Web Archive

Currently, content of the NLE Web Archive is stand-alone and not included into the Fedora repository. Web archiving content consists of archived web pages and screenshots of websites front pages.

The main objects of web archive according to medium are listed below:

- **Web harvests** in WARC format;

- **Screenshots** of websites' frontpages can appear in JPG, PNG;
- **Videos** from streaming media platforms (such as Youtube) can appear in MP4, Webm.

Below is presented some core information about the **expectations** of NLE regarding inclusion of Web Archive content into the digital archive in future.

- **Files and formats included in an information package:**
    - A web archiving information package includes one web archiving job (i.e. crawl job);
    - Each job includes a metadata WARC file which gathers information about the settings of the crawl job and its log;
    - Each job includes one or many content WARC files with the actual content of the crawled site;
    - Maximum size of an individual WARC file is 1 GB. Note that this restriction was enforced in 2015;
    - Each job might include one or many screenshots of the crawled site in PNG format.
- **Descriptive metadata:**
    - Currently no separate descriptive metadata is included into the information packages of the Web Archive;
    - Dublin Core is expected to be the metadata standard for web archive (by the end of 2020)
    - It is expected that in the future the digital archive identifier of the stored information package is sent back to the KROOL database after successful ingest of the package.
- **Other metadata:**
    - The submitted information package already includes a lot of administrative metadata within the metadata WARC file. It is expected that this metadata is stored "as is";
    - It is expected that the digital archive is capable of identifying the components of the content WARC file(s) – f.ex the HTML, PNG, TIFF etc files which constitute the harvested site;
    - Further administrative and preservation metadata, such as metadata about technical environments to be used for emulation, is expected to be created by the digital archive during ingest and preservation.
- **Classification / collections:**

- The current KROOL database includes individual crawl jobs into site / domain-based collections;
- It is expected that the new digital archive implements the same classification logic when storing web content.

- **Additional notes on web archiving**
  - NLE makes use of WARC deduplication (i.e., if a file like a jpg logo is being harvested is already available within the previous harvest, it is not included into the WARC file but instead referenced through a revisit record). As such, access to an individual harvest effectively also requires access to other harvests in this collection;
  - Next to the harvests NLE takes image snapshots of the home screens of websites. These snapshots are not part of any individual harvest job but rather a separate collection. The same logic applies also for harvested YouTube videos (i.e. these are also stored outside the WARC files as separate objects, linked only through metadata within the KROOL database).

3.4 Security, users, user groups and rights

The digital archive must be accessible for various parties, for instance system administrators, internal users, data donators (external) etc. users. Each party must be able to perform activities in their specific user permissions scope.

There are different user interfaces through which the associated parties can gain access to the system:

- **External user interface** – A universal user interface for occasional users to access public information. Can be accessed:
  - anonymously (for public users);
  - authenticated users (by logging in).
- **Internal user interface** – Interface inside the application with full functionality available to experienced accredited (Digital Archive internal) users.
- **API** – A series of APIs to allow third party systems to consume Digital Archive data

Following parties are associated with the system:

- **Administrator** who can:
  - import, export, create, read, update, publish and delete any record in the system;
  - customize application to institution specific requirements;

- manage user accounts and profiles;
- create new user roles;
- set granular permissions for roles;
- assign or unassign users from the new role.

Administrators can also create their own user groups with custom permissions, and individual user permissions can also be configured. Admins can configure API permissions.

- **Editor** who can:
  - search, browse, create, edit/update, view draft and export descriptions;
  - change the publication status of an information object;
  - access the reference and master digital object;
  - access the accessions module.
- **Contributor** who can:
  - search, browse, create, edit/update, view draft and export descriptions;
  - access the reference and master digital object.
- **Researcher (not logged in / unauthenticated user)** who can:
  - have view-only access to the application;
  - search and browse published descriptions;
  - view and download digital objects, depending on the permissions attached to the objects.

The following CRUD matrix is presented in order to give an overview of user groups rights. It lists different users in the system, able to either C- create, R- read, U- update D- delete content in the databases. The information package is divided into 3 sections, users can access: record- content file, technical metadata about file format, size and other technical information and content metadata, which gives an overview of the record contents.

Table 12. System user groups rights CRUD matrix

Data \ User	Admin	Researcher	Contributor	Editor	Authorised workstation	Public workstation
Record	C;R;U;D	R	C;R;U;	R;U	R	R
Technical metadata	C;R;U;D	R	C;R;U;	R;U	R	R
Content metadata	C;R;U;D	R	C;R;U;	R;U	R;U	R

Considering processes, digital archive is being used by various roles in NLE. Many of them can access digital archive via integrated systems, for example Publisher Portal, Web Archive etc. The roles in NLE are:

- Depositor (user)
- Publisher portal employee
- Digital archivist
- Digitiser
- Main user
- Archiver
- NLE employee
- Segmenter
- DA Workflow manager
- Web archive employee
- Web archive admin
- Data miner
- User
- Dark archive administrator
- Application administrator
- System administrator

Following CRUD matrix indicates, which operations can be performed by roles. Operations are divided into sections. Section 1 operations are related to SIP packet creation and ingestion to digital archive. Section 2 contains list of processes with AIP. Section 3 informs about the viewability of files by users.

Table 13. NLE roles user groups rights CRUD matrix

Operation	Depositor (user)	Publisher portal employee	Digital archivist	Digitiser	main user	Archiver NLE employee	Segmenter	DA Workflow manager	Web archive employee	Web archive admin	Data miner	User	Dark archive admin	Application admin	Sys admin
Section 1															
SIP Publisher Portal ingest	CRU	CRU	CRUD	-	CRUD	R	-	R	R	-	-	-	-	CRUD	-
SIP from Publisher Portal to Archive	CRU	RU	CRUD	-	CRUD	R	-	-	-	-	-	-	-	CRUD	-
SIP docWorks create	-	-	CRUD	-	CRUD	-	R	-	-	-	-	-	-	CRUD	-
SIP docWorks to Archive	-	-	CRUD	-	CRUD	R	CRU	-	-	-	-	-	-	CRUD	-
SIP Web archive	-	-	CRUD	-	CRUD	CRU	-	-	CRUD	CRUD	-	-	-	-	-
SIP Goobi	-	-	CRUD	CRU	CRUD	CRU	-	-	-	-	-	-	-	CRUD	-

SIP API (mass ingestion)	-	-	CRUD	-	CRUD	CRU	-	-	CRUD	CRUD	-	-	CRUD	CRUD	-
Section 2															
Create AIP	-	RU	CRUD	-	CRUD	-	-	R	CRUD	CRUD	-	-		CRUD	-
Change AIP file formats	-	-	CRUD	-	CRUD	-	-	CRU	CRU	CRUD	-	-	CRUD	CRUD	-
Bulk actions with files	-	-	CRUD	-	R	R	-	CRU	R	CRUD	-	-	CRUD	CRUD	-
Test new features	-	-	CRUD	-	R	-	-	CRU	R	CRUD	-	-	CRUD	-	-
Order original file from AIP	R	R	CRUD	-	CRUD	CRUD	-	R	R	CRUD	R?	-	CRUD	-	-
Section 3															
View files in DA	-	R	CRUD	-	R	R	R	CRU	R	CRUD	R	-	CRUD	CRUD	-
View files in UI	R	R	R	R	R	R	R	R	R	R	R	R	R	-	-
Get files											R				
Get metadata											R				

Security-analysis matrix informs which user groups can access the processes. Access restrictions are categorized into 4 different groups:

- C- create;
- R- read;
- U- update;
- D- delete;

Users can perform operations in the processes, for example import, export, publish data, also search, view, browse content and configure, set, customize different parts of the system. Empty cells indicate, the process is not visible to the listed user group. The following security-analysis matrix is presented to inform about the user groups restrictions and abilities.

Table 14. User groups security-analysis matrix

Process \ User	Administrator	Researcher	Contributor	Editor
Record management	import, export, publish, C,R,U,D			change publication status / validate
Application management	customize application, incl. web crawler configuration	access		
Account management	C,R,U,D			
User roles management	C,R,U,D (+assign, unassign)			

Process \ User	Administrator	Researcher	Contributor	Editor
User permissions settings	set granually			
Public descriptions	(search and browse)	search and browse	export	
Digital objects	set permissions	view and download	access (ref and master)	access (ref and master)
Draft management			C,R,U, search, browse	C,R,U,D, search, browse
Terms managment				edit controlled terms
Accessions module				access

3.4.1 Approximate number of users (with increment in time)

DIGAR, DEA and Estonian Web Archive users and page views (based on Google Analytics data for DIGAR.ee, DEA.digar.ee and veebiarhiiv.digar.ee) are presented in the following table.

Table 15. DIGAR, DEA and Estonian Web Archive users and page views (based on Google Analytics data for DIGAR.ee, DEA.digar.ee and veebiarhiiv.digar.ee)

	In 2017		In 2018		Total by the end of 2018
	All per year	Max per day	All per year	Max per day	All
www.digar.ee*					
Users	180 904	1 230	221 859	1 535	726 954
Unique page views	1 199 893	9 479	1 340 287	8 460	5 064 889
dea.digar.ee					
Users	586 626	4 099	727 470	5 479	2 193 147
Page views	1 920 250	10 397	2 569 924	16 440	7 663 458
veebiarhiiv.digar.ee[14]**					
Users	20 332	603	29 952	590	91 796

[14] Web archive has only very little part of its content publicly available (mainly governmental websites).

Page views	143 406	4 207	175 583	4 409	681 874

* total amount is given for the period 2013-2018

** total amount is given for the period 2014-2018

3.5 Digital archive back up and restore

3.5.1 Restoring and backing up application / database

Digital archive application and database backuping process must support RTO of 24 hours- the application and database must be restored within 24 hours of the incident. In order to avoid rollbacks, database and application must be backed up daily.

Incremental forever backuping process must be supported. First of all, initial full backup is created and after that, only incremental backups are performed.

Logs are managed incrementally:

- Last 5 days are always fully recoverable;
- Last 4 weeks are recoverable by each week end state;
- Last 12 months are recoverable by each month end state;
- Last n years are recoverable by each year end state.

3.5.2 Restoring and backuping contents of the archive

Other than that, Digital Archive storage policy requirements for content storage is based on the following principles:

- Metadata of all objects must be held in the Digital Archive database;
- All information packages must be saved on physical storage in 3 copies:
    - An online, accessible copy;
    - An offline, tape copy placed in NLE;
    - An offlline, tape copy placed off site.

The object metadata must be held in the Digital Archive database, because in long term preservation, the risk of system decay must be considered. If the metadata is held in the Digital Archive database, it is possible to recover information by individual components.

To increase the recovering possibility, 3 copies in physical storages must be used in the process of saving the information packages. To guarantee a quick restore process, one of the copies must be online and accessible.

To make sure the information is protected in long term, offline tapes as copies are needed. To avoid the destruction of both data storages, these must be kept in different physical locations. One of the tape copies must be on-site at NLE and the other one must be stored off-site.

The Digital Archive must support the storage policy and the vendor must provide description if the result is achieved.

3.6 Main risks related to the project

Integrations and data migration create risks. The following lists contain main risks which may occur during the project.

Risks related with data migration from current Digital Archive:

- There is no regular inventory of the different components of the archive conducted in the past years. Therefore, there is no overview of the regularity of the metadata (there have been different rules and standards have been excecuted over time);
- The preservation, administrative etc. metadata is currently preserved in METS XML format in DEA and FOXML format in Fedora. The data needs to be migrated directly from files.

Risks related with the Digital Archive integrations with other systems:

- There are many integrations to current Digital Archive, and some of the integrated systems are developed specifically for NLE, which might create problems in integration phase. If some of the integrations to ingest and/or access components do not function properly then the Digital Archive cannot perform its tasks;
- Ingested files capacity may be larger than the Digital Archive can take in. If the ingested file is larger than the Digital Archive can take in it will have a negative impact on Digital Archive performance and the chance of an error increase in the phase of ingesting files;
- Changes in integrated systems API’s might result in integration malfunctions;
- Integrated system is under development – changes in the Publishers portal occur while Digital Archive is implemented.

Risks related with Digital Archive project flow:

- Team member potentially leaving or being on a prolonged absence. The skills and knowledge of team members are in many cases not duplicated. Absence of a team member will also lead to a loss of knowledge of the field;
- Cooperation risk between NLE and contracting partner to provide IT services and support. NLE uses external IT organization resources which might cause risks of difficulties in project management.

Appendix 1: List of attached FOXML example files

1_FOXML_BornDigitMono2005.xml

2_FOXML_BornDigitMono2018.xml

3_FOXML_BornDigitMono2018.xml

4_FOXML_DigitisedMono2019.xml

5_FOXML_DigitisedMonoWithAudio2018.xml

6_FOXML_DigitisedMono2019.xml

7_FOXML_BornDigitPeriodic2008.xml

8_FOXML_BornDigitPeriodic2015.xml

9_FOXML_BornDigitPeriodic2017.xml

10_FOXML_DigitisedPeriodic2018.xml