DELIVERABLE

Project Acronym: **E-ARK**

Grant Agreement Number: **620998**

Project Title: **European Archival Records and Knowledge**
Preservation

DELIVERABLE DETAILS

DELIVERABLE REFERENCE NO.	**D3.3**
DELIVERABLE TITLE	E-ARK SIP Pilot Specification (revision of D3.2, main part of the D3.3)
REVISION	**1.0**

AUTHOR(S)

Name(s)	Organisation(s)
Tarvo Kärberg	National Archives of Estonia (NAE)
Karin Bredenberg	National Archives of Sweden / ES Solutions (ESS)
Björn Skog	ES Solutions (ESS)
Anders Bo Nielsen	Danish National Archives (DNA)
Kathrine Hougaard Edsen Johansen	Danish National Archives (DNA)
Hélder Silva	KEEP Solutions (KEEPS)
Gregor Zavrsnik	Slovenian National Archives (SNA)
Levente Szilágyi	National Archives of Hungary (NAH)
Phillip Mike Tømmerholt	Danish National Archives (DNA)

REVIEWER(S)	
Name(s)	Organisation(s)
Kuldar Aas Sven Schlarb David Anderson Andrew Wilson	National Archives of Estonia (NAE) Austrian Institute of Technology (AIT) University of Brighton University of Brighton

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

REVISION HISTORY AND STATEMENT OF ORIGINALITY

Submitted Revisions History

Revision No.	Date	Authors(s)	Organisation	Description
0.1	20.10.2014	Tarvo Kärberg	NAE	First draft.
0.2	13.11.2014	Tarvo Kärberg	NAE	Updating content.
0.3	02.12.2014	Tarvo Kärberg	NAE	Updating content.
0.4	17.01.2015	Tarvo Kärberg	NAE	Updating content.
0.5	21.01.2015	Karin Bredenberg	ESS	Updating content.
0.6	23.01.2015	Anders Bo Nielsen	DNA	Updating content.
0.7	23.01.2015	Kathrine Hougaard Edsen	DNA	Updating content.
0.71	26.01.2015	Björn Skog	ESS	Updating content.
0.72	27.01.2015	Hélder Silva	KEEPS	Updating content.
0.8	27.01.2015	Angela Dappert	DLM/UPHEC	Quality assurance and proof-reading.
0.9	29.01.2017	Kuldar Aas	NAE	Quality assurance and proof-reading.
0.91	30.01.2015	David Anderson	UPHEC	Quality assurance and proof-reading.
1.0	30.01.2015	Tarvo Kärberg	NAE	**Final version (D3.2).**
0.1	11.05.2015	Karin Bredenberg	ESS/NAS	Updating content.
0.2	30.06.2015	Tarvo Kärberg	NAE	Updating content.
0.3	27.07.2015	Tarvo Kärberg	NAE	Updating content.
0.4	23.10.2015	Tarvo Kärberg	NAE	Updating content, synchronising with the SMURF profile.
0.41	17.11.2015	Tarvo Kärberg	NAE	Integrating the feedback.
0.42	07.12.2015	Tarvo Kärberg	NAE	Updating content.
0.5	12.01.2016	Tarvo Kärberg	NAE	Updating content, synchronising with the Common Specification.
0.6	15.01.2016	Anders Bo Nielsen	DNA	Updating content.
0.61	15.01.2016	Gregor Zavrsnik	SNA	Updating content.
0.62	18.01.2016	Tarvo Kärberg	NAE	Updating content.
0.63	20.01.2016	Phillip Mike Tømmerholt	DNA	Updating content.
0.64	25.01.2016	Phillip Mike Tømmerholt	DNA	Updating content.
0.7	26.01.2016	Sven Schlarb	AIT	Quality assurance and proof-reading.
0.8	27.01.2016	Kuldar Aas	NAE	Quality assurance and proof-reading.
0.9	29.01.2016	Andrew Wilson and David Anderson	University of Brighton	Quality assurance and proof-reading.
1.0	29.01.2016	Tarvo Kärberg	NAE	**Final version** (general part of D3.3)

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

EXECUTIVE SUMMARY

According to the Open Archival Information System Reference Model (OAIS) every submission of information to an archive by a producer occurs as one or more discrete transmissions of submission information packages.[1] Unfortunately there is currently no central SIP format which would cover all national and business needs as identified in the E-ARK Report on Available Best Practices.[2] The E-ARK project has acknowledged this problem and started to develop a solution in the form of the E-ARK SIP format which is described in this deliverable.

The first outcome of this work was Deliverable 3.2: E-ARK SIP Draft Specification.[3] This gives an overview of the structure and main metadata elements for the E-ARK SIP and provides initial input for the technical implementations of E-ARK pre-ingest and ingest tools. The current specification (Deliverable 3.3) extends the previous one by providing a revised version of the D3.2 content, adding more details relevant for tool development and implementation, and describing specific profiles for the transfer of relational databases, electronic records management systems (ERMS) and simple file system based records (SFSB).

The target group for this deliverable are records creators, archival institutions and software providers creating or updating their SIP format specifications. The specification is also important for electronic records management systems (ERMS) providers as it presents a standardised profile for exporting records and metadata out of their systems.

This document provides an overview of:

- The general structure for Submission Information Packages.
  This chapter explains how records creators should construct/structure their SIPs in order to meet the requirements of the E-ARK SIP specification and achieve interoperability by following the common rules for all information packages (SIPs, AIPs, DIPs) as described in the documents describing EARK IP's[4].
- General SIP metadata.
  This chapter provides a detailed overview of metadata sections and the metadata elements in these sections. The tables with all metadata elements could possibly be of interest to technical stakeholders who wish to implement the E-ARK SIP.
- Content-type-specific profiles.
  This section introduces profiles for SMURF (Semantically Marked Up Records Format) and relational databases. The profiles themselves are separate documents.

[1] Reference Model for an Open Archival Information System (OAIS), 2012, public.ccsds.org/publications/archive/650x0m2.pdf

[2] Deliverable D3.1 E-ARK Report on Available Best Practices, 2014, http://eark-project.com/resources/project-deliverables/6-d31-e-ark-report-on-available-best-practices

[3] Deliverable 3.2 E-ARK SIP Draft Specification, 2015, http://eark-project.com/resources/project-deliverables/17-d32-e-ark-sip-draft-specification

[4] E-ARK Draft Common Specification for the Information Packages in the E-ARK project, version 0.13, http://eark-project.com/resources/specificationdocs/50-draftcommonspec-1

- The submission agreement.
  This chapter provides an overview of submission agreement usages and recommended metadata elements for the E-ARK project.

NB! This deliverable D3.3 is a revision of the deliverable D3.2 E-ARK SIP Draft Specification (2015, http://eark-project.com/resources/project-deliverables/17-d32-e-ark-sip-draft-specification).

- The deliverable D3.3 repeats some parts of the D3.2 for convenience. It aims at being easily readable and understandable for all interested parties without having to read the previous deliverable D3.2.

- The full deliverable D3.3 consists of multiple documents which all together form the SIP specification.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENT

The authors of this deliverable would like to thank all national archives, tool developers, the Advisory Board of the E-ARK project and other stakeholders who provided valuable knowledge about their submission information packages and feedback to our previous deliverables.

A special gratitude goes to the National Archives of Sweden whose FGS (Förvaltningsgemensam Specifikation) structure influenced the first version of the E-ARK SIP METS profile development significantly.

The authors would also like to express their gratitude to the team behind the Common Specification for Information Packages document for their enormous effort in agreeing common principles for submission, archival and dissemination packages.

1. INTRODUCTION

1.1. Scope and purpose

The purpose of this document is to revise the deliverable D3.2 (E-ARK SIP Draft Specification) and describe the pilot specification for the E-ARK SIP and, therefore, this work will be important mainly for E-ARK project partners; including partners who develop pre-ingest and ingest solutions for the E-ARK pilots. However, to gain further insights beyond those provided by the project partners, the specification will be disseminated and discussed among other organisations, mainly archival institutions and archival software providers.

The current deliverable does not define use cases or workflows within a digital archive, but rather describes the SIP package structure and minimum set of required metadata for SIP delivery to the archive.[5]

This deliverable does not provide information about archival information packages (AIPs) or dissemination packages (DIPs)[6] although it follows the same common principles agreed in the “Common specification for information packages” document.[7]

1.2. Composition of the deliverable

This document is the core / general SIP specification. Beyond the general SIP there are two specific profiles which are again separate documents (see Figure 1) and which together with this document make up deliverable 3.3:

- content type specific specifications
  - profile for relational databases[8]
  - profile for electronic records management systems (ERMS) and simple file system based records (SFSB) – SMURF (semantically marked up record format) specification)[9]
- general SIP specification (the current document)

Although the Common specification for information packages document is very important for this work (as mentioned earlier), neither the current specification nor the full deliverable D3.3 aim at fully duplicating or

[5] Therefore, this deliverable is also important input to pilots in the E-ARK project.

[6] These are provided respectively in deliverables 4.3 (released January 2016) and 5.3 (released April 2016) of the E-ARK project.

[7] E-ARK Draft Common Specification for Information Packages in the E-ARK project, version 0.13, http://eark-project.com/resources/specificationdocs/50-draftcommonspec-1

[8] The SIARD 2.0 specification for relational databases can be found at http://eark-project.com/resources/specificationdocs/32-specification-for-siard-format-v20

[9] The SMURF profile for ERMS and SFSB can be found at http://eark-project.com/resources/project-deliverables as a part of the official deliverable D.3.3 (released January 2016).

even largely repeating the information presented there – only the information that is absolutely necessary to understand the SIP specification will be mentioned here.

The following diagram illustrates the relationship between the Common specification for information packages and other documents (Figure 1).

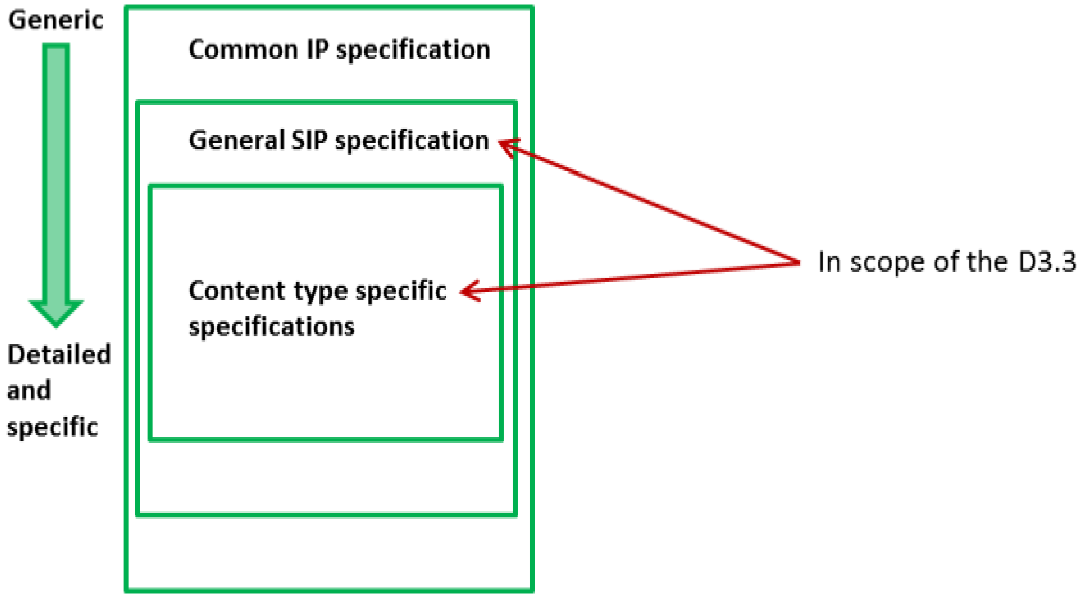

Figure 1: Relations between specifications

Additional technical documentation regarding this specification can be found in the following documents:

- E-ARK METS Profile
  http://www.ra.ee/METS/v02/METS.xml
- XML Schema for the E-ARK METS Profile
  http://www.ra.ee/METS/v02/METS.xsd
- XLink Schema
  http://www.ra.ee/METS/v02/xlink.xsd
- Extension Schema
  http://www.ra.ee/METS/v02/ExtensionMETS.xsd

1.3. Related work

This deliverable is based on or influenced by the following documents and best practices:

- **Deliverable D3.1** E-ARK Report on Available Best Practices, 2014, http://eark-project.com/resources/project-deliverables/6-d31-e-ark-report-on-available-best-practices

  D3.1 was one of the inputs to the deliverable D3.2 and as the D3.2 is one of the main inputs for the current deliverable then some of the principles introduced in D3.1 will be represented in D3.3 as well.

- **Deliverable D2.1** General pilot model and use case definition, 2014, http://eark-project.com/resources/project-deliverables/5-d21-e-ark-general-pilot-model-and-use-case-definition.

  We have developed the SIP specification to support the workflows defined in the general model.

- **FGS package structure**, 2013, https://riksarkivet.se/Media/pdf-filer/Projekt/FGS_Earkiv_Paket.pdf This specification was one of the main inputs for the first draft SIP specification. The newest version (https://riksarkivet.se/Media/pdf-filer/doi-t/FGS_Paketstruktur_RAFGS1V1.pdf) was also investigated in the SIP definition process.

- **Reference Model for an Open Archival Information System** (OAIS), 2012, public.ccsds.org/publications/archive/650x0m2.pdf

  We have used the same terminology as introduced in the OAIS model and also the same division of information package types: Submission Information Package (SIP), Archival Information Package (AIP), Dissemination Information Package (DIP).

- **Producer-Archive Interface Methodology Abstract Standard** (PAIMAS), 2004, public.ccsds.org/publications/archive/651x0m1.pdf

  We have looked at the four phases (Preliminary, Formal Definition, Transfer, Validation) of PAIMAS, their aims and expected results and decided to support the phases as far as possible with the current specification. Furthermore, the requirements for the submission agreement were influenced by the PAIMAS standard.

- **Producer-Archive Interface Specification** (PAIS) – CCSDS, 2014, public.ccsds.org/publications/archive/651x1b1.pdf

  We have investigated the structure of a SIP presented in PAIS, but as the implementation of this specification is far from comprehensive (only few prototypes exist), we decided to rely more on the best practices introduced in E-ARK reports.

- **e-SENS** (Electronic Simple European Networked Services) project, http://www.esens.eu/

  We have investigated the e-Delivery and e-Documents related work in e-SENS and made sure that our work is neither duplicating the work done there nor producing any conflicts between deliverables.

- **Deliverable D3.2** E-ARK SIP Draft Specification, 2015, http://eark-project.com/resources/project-deliverables/17-d32-e-ark-sip-draft-specification

  Deliverable D3.3 is a revision of deliverable D3.2.

2. GENERAL STRUCTURE AND DATA MODEL FOR SUBMISSION INFORMATION PACKAGES

The SIP pilot specification for E-ARK follows the general structure which is common for all submission information packages in the E-ARK project. The SIP data model describes the package submitted to the archive, which consists of representations (submitted data and metadata) and metadata as seen in Figure 2[10] and mandated/required by the E-ARK SIP, AIP and DIP formats.

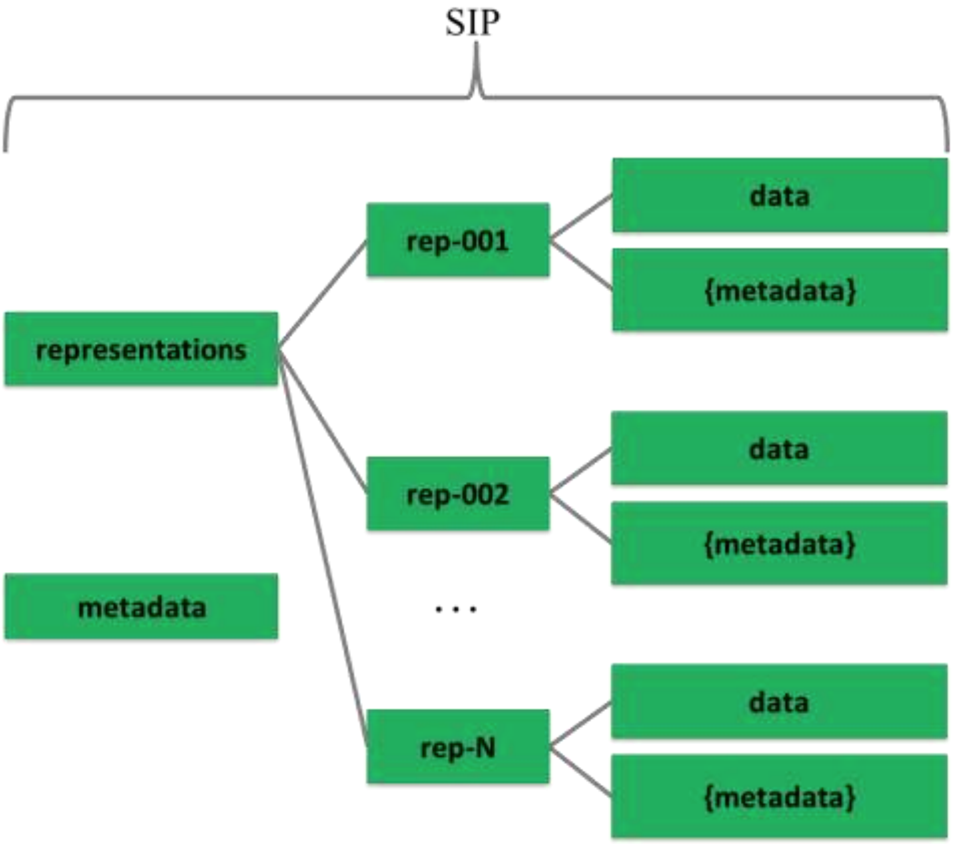

Figure 2: E-ARK SIP data model

As one SIP can contain more than one representation[11] of the same intellectual entity then it is reasonable to separate different representations (e.g. Rep-001 and Rep-002 under Representations). This requires additional metadata about the SIP. If we store all metadata (even about representations) at the IP level then we do not need to use the Metadata folder at the representation level. In this case, the Metadata directory under representations is considered optional, as are:

[10] This is a conceptual model and does not describe the actual implementation structure.

[11] Digital Object or physical object instantiating or embodying an Intellectual Entity. A Representation that is a Digital Object is the set of stored Files and Structural Metadata needed to provide a complete rendition of the Intellectual Entity. PREMIS Data Dictionary (full document), Version 3.0, 2015, http://www.loc.gov/standards/premis/v3/premis-3-0-final.pdf

- Documentation folder – for including additional documents that explain the content or its use (e.g. user manual).

- Schemas folder – for adding schemas for the XML files in the data/metadata directly into the package.

According to the E-ARK Draft Common Specification for Information Packages in the E-ARK project, the information package folder must include a mandatory core metadata file called ”METS.xml”, which includes core information needed to identify and describe the structure of the package itself and the rest of its components. [12] Therefore the minimal general SIP structure is as follows (Figure 3). The name of a representation can be freely chosen.

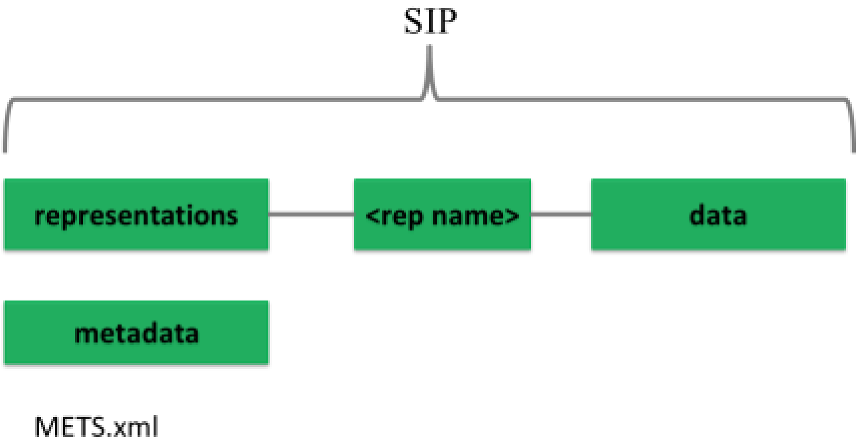

Figure 3: Minimal SIP structure

If needed, a METS.xml file can be present under representations as well to handle scalability issues. This proposed extended IP structure using divided METS files, is introduced in the E-ARK Draft Common Specification for Information Packages[13] and in deliverable D4.3 E-ARK AIP pilot specification[14] to more easily manage the splitting of large packages using a divided METS structure.

More information about the structure of an IP and its other requirements can be found in the Appendix A on page 46.

The detailed folder structure of a SIP will also be present and agreed upon in the submission agreement (page 40) by indicating the data model for the submission. All E-ARK tools will implement the exact folder structure presented in the Common Specification. The details of the internal structure of the data and metadata folders can be further specified in submission agreements.

[12] E-ARK Draft Common Specification for Information Packages in the E-ARK project, version 0.13, page 23, http://eark-project.com/resources/specificationdocs/50-draftcommonspec-1

[13] E-ARK Draft Common Specification for Information Packages in the E-ARK project, version 0.13, page 24, http://eark-project.com/resources/specificationdocs/50-draftcommonspec-1

[14] E-ARK AIP pilot specification, released January 2016, http://eark-project.com/resources/project-deliverables

The E-ARK Report on Available Best Practices[15] showed that METS (Metadata Encoding and Transmission Standard)[16], EAD (Encoded Archival Description), [17] EAC-CPF (Encoded Archival Context – Corporate bodies, Persons, and Families)[18] and PREMIS (PREservation Metadata: Implementation Strategies)[19] standards are most commonly used in SIPs so we will continue using them in the current specification. These standards have different purposes:

- PREMIS provides metadata to support the long-term preservation of digital objects and their usability.
- EAD is a non-proprietary de facto XML standard for hierarchical archival descriptions based on ISAD-G.[20]
- EAC-CPF primarily addresses the description of individuals, families and corporate bodies that create, preserve, use and are responsible for and/or associated with records in a variety of ways.
- METS standard provides a means of associating the metadata related to an object with the object, and describes its relationships with other objects.

According to deliverable D3.2 the E-ARK SIP will use:

- METS for the high-level descriptions of an information package
- EAD together with EAC-CPF for the archival description and archival creator description
- PREMIS for the administrative and descriptive metadata to support the long-term preservation.

The metadata model for the E-ARK SIP will be multi-layered by starting from general common metadata elements and finishing with optional local elements (Figure 4).

[15] E-ARK Report on Available Best Practices, 2014, URL: http://eark-project.com/resources/project-deliverables/6-d31-e-ark-report-on-available-best-practices

[16] METS, 2015, http://www.loc.gov/standards/mets/

[17] Encoded Archival Description, 2015, http://www.loc.gov/ead/

[18] Encoded Archival Context for Corporate Bodies, Persons, and Families, 2015, http://eac.staatsbibliothek-berlin.de/

[19] PREservation Metadata: Implementation Strategies , 2015, http://www.loc.gov/standards/premis/

[20] ISAD-G, http://www.ica.org/10207/standards/isadg-general-international-standard-archival-description-second-edition.html

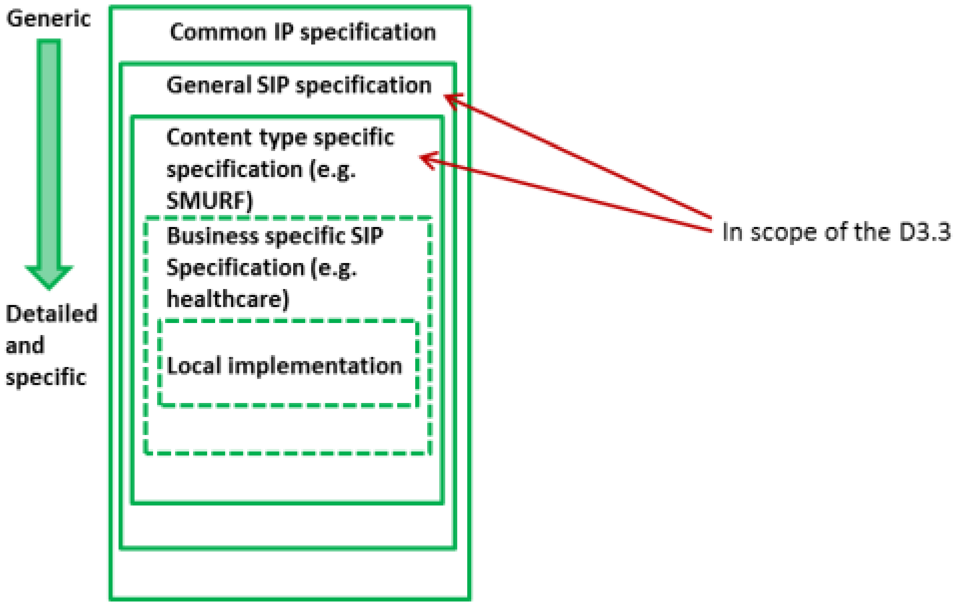
Generic
Detailed and specific
Common IP specification
General SIP specification
Content type specific specification (e.g. SMURF)
Business specific SIP Specification (e.g. healthcare)
Local implementation
In scope of the D3.3

Figure 4: The logic of specification layers

- Common IP specification – common requirements and metadata for all information packages (SIP, AIP, DIP).
- General SIP specification – common metadata for submission information packages.
- Content type specific specification – delivery-specific content descriptions. For example SMURF (profile) for data delivered from records management systems or SIARD for databases.
  One of the core requirements for the E-ARK SIP specification is that it shall be able to be extended to support any content type a digital repository needs to ingest – additional separate content type descriptions can be specified and developed for different types of submissions.
- Business specific SIP Specification – more specific and detailed profiles based on business of an agency (e.g. healthcare records). This is optional and will be not developed in E-ARK.
- Local implementation - As the specifications can be undertaken at different scales, with different types of data and locations, with their constituent technical components, more detailed or localised specifications may be needed. This is optional and will be not developed in E-ARK.

3. GENERAL SIP METADATA

The general SIP metadata is based on the METS standard and presented as a profile. METS profiles are intended to describe a class of METS documents in sufficient detail to provide both document authors and programmers with the guidance they need to create and process METS documents conforming to a particular profile.[21]

Creating a METS profile requires a good understanding of the METS Profile components. An overview of these components can be found in the METS online documentation[22] and in Appendix D on page 44 in the D3.2 specification.

There are 5 main sections in this METS profile:

- <metsHdr> - METS header (metadata about the creator, contact persons, etc. of the IP).
- <dmdSec> - descriptive metadata (references to EAD, EAC-CPF, etc.).
- <amdSec> - administrative metadata (how files were created and stored, intellectual property rights, etc.).
- <fileSec> - file section, lists all files containing content (may also contain metadata about files).
- <structMap> - structural map, describes the hierarchical structure of the digital object and the whole IP (i.e. object + metadata).

These sections will be described in more detail in sections 3.1 to 3.6. All these sections present the E-ARK SIP requirements for METS elements in table form according to the following structure:

- Element - The name of the element in plain text used in the accompanying schema for elements or attributes. For more information regarding elements and attributes in XML see WWW Consortium (http://www.w3.org/).
- Definition - Defines the functions of the element. Contains an explanation of the element and some example values.
- Cardinality – Represents the number of occurrences of an element (see below).
    - 0..1 – The element is optional and cannot be repeated.
    - 0..* – The element is optional and can be repeated.
    - 1 – The element is mandatory and can only be stated once.
    - 1..* – The element is mandatory and has one or more occurrences.
- METS - Defines the element in the METS standard used for designing the E-ARK element. The column uses XML-syntax. [ ] defines where the value is placed.

[21] METS Profiles, http://www.loc.gov/standards/mets/mets-profiles.html

[22] METS Profile Components, http://www.loc.gov/standards/mets/profile_docs/components.html

3.1. Root

The root of a METS document can contain a number of optional attributes, namespaces (xmlns:) and schema instance locations (xsi:) of the external standards referenced in the METS record and a number of other elements as seen in Table 1.

Table 1: METS root

Element	Definition	Explanation	Card.	METS
Identity	Identification of the package	A code that uniquely identifies the whole SIP and the digital object/objects being submitted. A UUID or GUID should be used to create globally unique identifiers. Example: *"UUID:550e8400-e29b-41d4-a716-446655440004"*	1	<mets: OBJID="[Identity]" >
Description	Description of the package	Short text describing the package. Example: *"Example of SIP for delivery of personnel information"*	0..1	<mets: LABEL="[Description]">
Content type	The content type being submitted with this package	Describes the content type the information being submitted in the package belongs to. It is recommended to use the pattern [OAIS_Type]:[Standard_Name]. Example: *"SIP:SIARD2"*	1	<mets: TYPE="[Content type]"

Element	Definition	Explanation	Card.	METS
Profile[23]	Profile name	Describes the METS-profile being used. The name should contain the version number and the version may be captured in the profile file path. Example: *" http://eark-project.com/resources/METS/v02/METS.xml"*	1	<mets: PROFILE="[Profile] "

Example:

```
<mets xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns="http://www.loc.gov/METS/"
PROFILE="http://www.ra.ee/METS/v02/METS.xml" TYPE="SIP:SIARD2" OBJID="5d378f86-28a1-41d8-a2b9-
264b10fbd511" LABEL="METS file describing the SIP." xsi:schemaLocation="http://www.loc.gov/METS/
schemas/METS.xsd http://www.w3.org/1999/xlink schemas/xlink.xsd">
```

3.2. Header

The METS header element <metsHdr> describes metadata about the creator, contact persons, etc. of the submission information package as seen in Figure 5.

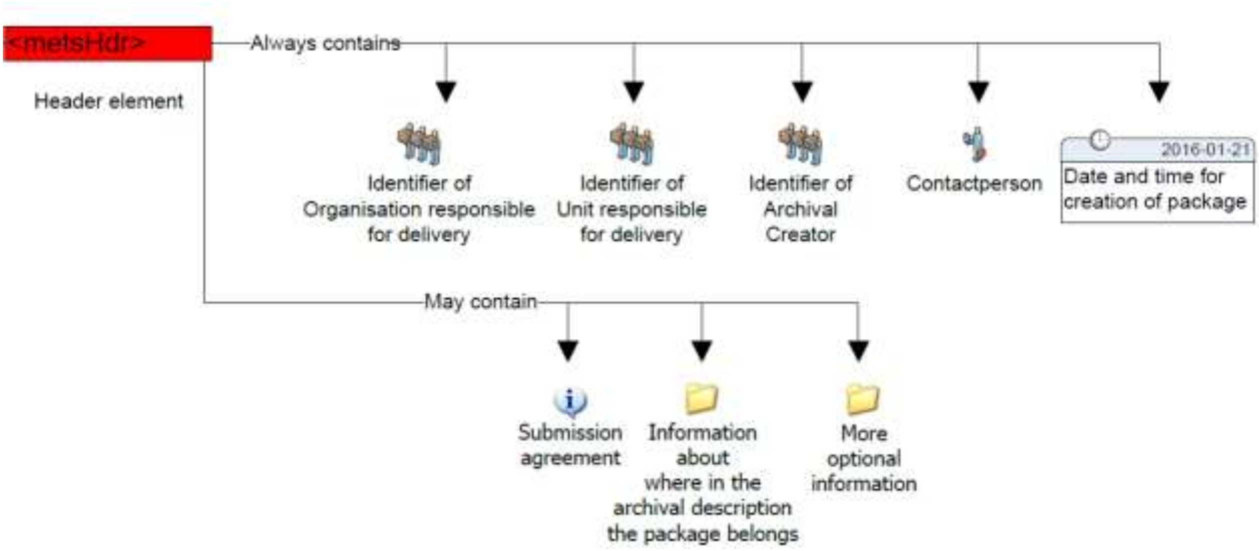
<metsHdr>
Header element
Always contains
Identifier of Organisation responsible for delivery
Identifier of Unit responsible for delivery
Identifier of Archival Creator
Contactperson
2016-01-21
Date and time for creation of package
May contain
Submission agreement
Information about where in the archival description the package belongs
More optional information

Figure 5: METS header

[23] This element is not used for representation METS XML files.

These are the elements that give information about the submission of the SIP in the METS header element.

Table 2: Metadata about the information package

Element	Definition	Explanation	Card.	METS
Date and time*[24]	Time of creation of package	Date and time for creation of the package must be described according to the XML-standard ("YYYY-MM-DDThh:mm:ssZ").[25] This timestamp states when the whole package and the package file was created.[26] Example: *"2012-04-26T12:45:00+01:00"*	1	<metsHdr: CREATEDATE="[Date and time]">
Status*	Package status	A way of indicating the status of the package for making it easier to know how to handle the package (allows package status specific processing or rendering) if for example an identical package is being submitted replacing the previous one sent. Example: *"NEW"* Example: *"TEST"* Example: *"REPLEACEMENT"*	0..1	<metsHdr: RECORDSTATUS="[Status]">

[24] Header elements marked with an asterisk * represent elements which are required only for the root METS.xml. All other elements can be recorded in the representation METS.xml files.

[25] XML Schema Part 2: Datatypes Second Edition, https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/#isoformats

[26] The package file is the METS-document which describes the content of the whole package.

Element	Definition	Explanation	Card.	METS
DocID*	METS document ID	A unique identifier for the METS document itself. This identifier may be different from the Identity given in the mets-element. The recommendation is to use the file name given to the METS-document. Example: *"SIP20150127.xml"*	0..1	<metsDocumentID>[DocID]
Submission agreement[27]	Reference to the used submission agreement	Complete reference to the submission agreement for the submission of the package. In this way it provides an alternative identifier for the package due to the requirement that a package can only belong to one submission agreement. Example: *RA 13-2011/5329; 2012-04-12* Example: *"http://submissionagreement.kb.se/dnr331-1144-2011/20120711/"*	0..1	<altrecordID: TYPE="SUBMISSIONAGREEMENT">[Submission agreement]
Previous submission agreement	The previous submission agreement(s) the information belongs to in the case the information is recorded.	Reference to the previous submission agreement(s) which the information may have belonged to is recorded if the information is available. Example: *"FM 12-2387/12726, 2007-09-19"*	0..*	<altrecordID: TYPE="PREVIOUSSUBMISSIONAGREEMENT">[Previous submission agreement]

[27] Reference Model for an Open Archival Information System (OAIS), 2012, public.ccsds.org/publications/archive/650x0m2.pdf

Element	Definition	Explanation	Card.	METS
Archival reference code	Reference code in the archival description	It is possible to give a reference code indicating where in the archival hierarchy the package shall be placed. Example: *"SE/RA/123456/24/P"*	0..1	<altrecordID: TYPE="REFERENCE CODE">[Archival reference code]
Previous reference code	An earlier used reference code in the archival description	In case where the SIP originates from other institutions maintaining a reference code structure, this element can be used to record these reference codes and therefore support the provenance of the package when a whole archival description is not submitted with the submission. Example: *"SE/FM/123/123.1/123.1.3"*	0..*	<altrecordID: TYPE="PREVIOUSREFERENCECODE">[Previous reference code]
Archival creator	Name of archival creator	Name of the original creator (organisation) of the data being transferred. Please note that this might be different from the organisation which has been charged with preparing and sending the SIP to the archives. Example: *"The Swedish health agency"*	1[28]	<agent: ROLE="ARCHIVIST" TYPE= "ORGANIZATION"> <name>[Archival creator]

[28] One SIP can contain information form one and only creator.

Element	Definition	Explanation	Card.	METS
Archival creator identification code	A unique identification code for the archival creator	A unique identification code for the archival creator. The code uses a prefix followed by a ”:” Prefix[29] according to vcTypeOfIdentificationCode. Example: *”VAT:SE201345098701”*	0..1	<agent: ROLE=”ARCHIVIST ” TYPE= ”ORGANIZATION”> <note>[Archival creator identification code]
Submitting organisation name[30]	Name of the organisation submitting the package to the archive	Name of the organisation submitting the package to the archive. For stating and extending the information, use of additional agents may be necessary. Example: *”The agency, Personnel*	1	<agent: ROLE=”CREATOR” TYPE= ”ORGANIZATION”> <name>[Submitting organisation name]
Delivering organisation identification code	A unique identification code for the delivering organisation	A unique identification code for the delivering organisation. The code uses a prefix followed by a ”:” Prefix according to vcTypeOfIdentificationCode. Example: *” VAT:SE2098109810-AF87”*	0..1	<agent: ROLE=”CREATOR” TYPE= ”ORGANIZATION”> <note>[Delivering organisation identification code]

[29] All prefixes referred in this table are described in the E-ARK METS Profile, http://www.ra.ee/METS/v02/METS.xml

[30] All similar <agent> elements (the archival creator, delivering organisation, submitting organisation and producing organisation) may not be needed in the final SIP METS Profile. The decision will be done later in the E-ARK project by taking into account the feedback from pilot projects.

Element	Definition	Explanation	Card.	METS
Contact person name	Contact person for the submission	It may be useful to record a contact person for the submission at the time of the submission. This information is only valid during a short time period. Example: *”Sven Svensson”*	0..*	<agent ROLE=”CREATOR” TYPE= ”INDIVIDUAL”> <name>[Contact person name]
Contact person contact information	Contact information for the contact person	Phone number and e-mail for the contact person. This information is only valid during a short time period. Example:[31] *”08-12 34 56”* *“sven.svensson@fm.se”*	0..*	<agent ROLE=”CREATOR” TYPE= ”INDIVIDUAL”> <note>[Contact person contact information]
Software*	The software which has been used to create the package	The metsHdr must include at least one agent describing the software which has been used to create the package.	1	<agent ROLE=”CREATER” TYPE=”OTHER” OTHERTYPE=”SOFTWARE”>
Preservation organisation name	Name of organisation preserving the package	Name of the organisation responsible for preservation of a submitted package. Example: *”National Archives of Hungary”*	1	<agent ROLE= ”PRESERVATION” TYPE= ”ORGANIZATION”> <name>[Preservation organisation name]

[31] As the SIP profile allows for 0..* fields then it is possible to have e-mail, phone, physical address details etc in separate fields. The coice is up to specific implementations.

Element	Definition	Explanation	Card.	METS
Preservation organisation identification code	Identification code of organisation preserving the package	A unique identification code for the organisation responsible for preservation. The code uses a prefix followed by ":" Prefix according to vcTypeOfIdentificationCode. Example: *"ORG:2010340987"*	0..1	<agent ROLE= "PRESERVATION" TYPE= "ORGANIZATION"> <note>[Preservation organisation identification code]

Example:

```
<metsHdr CREATEDATE="2016-01-15T10:40:09" LASTMODDATE="2016-01-15T10:40:09“>
  <agent TYPE="OTHER" ROLE="CREATOR" OTHERTYPE="SOFTWARE">
    <name>E-ARK SIP Creator</name>
    <note>VERSION=0.0.2</note>
  </agent>
</metsHdr>
```

3.3. Descriptive metadata

The METS descriptive metadata element <dmdSec> references to archival description metadata (EAD, EAC-CPF, etc.) as seen in Figure 6.

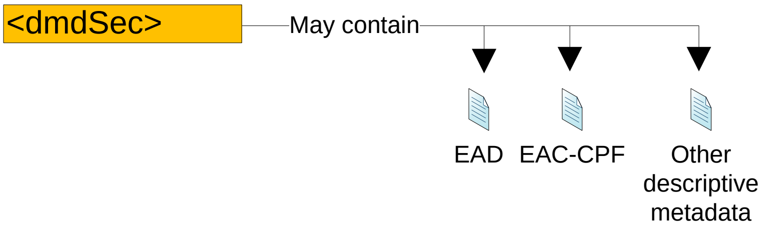

Figure 6: METS descriptive metadata

Archival information can be included in the METS package. For the core interest groups in E-ARK – archival institutions – this information is delivered to the recipient in EAD and EAC-CPF formats.
To include EAD and EAC-CPF in a METS profile the use of <dmdSec> is to be preferred according to the METS implementation guide referenced above. The complete rules for all elements and attributes in the <dmdSec> are stated in the profile, the specific elements used when referencing and embedding is shown below.

Other metadata standards for description and administrative purposes can be used and referencing or embedding[32] them must adhere to the <amdSec> and <dmdSec> rules stated in the profile.

Table 3: EAD metadata

Element	Definition	Explanation	Card.	METS
EAD metadata file	Metadata file in EAD format referenced in the METS document	Metadata file in EAD format when it is referenced in the METS document. Its needs to be stated in the submission agreement if referencing or embedding are used.	0..*	<dmdSec> <mdRef: MDTYPE=”EAD” xlink:href=”file:///[33][EAD-metadata file]” xlink:type=”simple”>
EAD metadata	Metadata in EAD format embedded in the METS document	Metadata in EAD format when it is embedded[34] in the METS document. Its needs to be stated in the submission agreement if referencing or embedding are used.	0..*	<dmdSec> <mdWrap: MDTYPE=”EAD”> <xmlData> [EAD metadata in XML-format]
EAC-CPF metadata file	Metadata file in EAC-CPF format referenced in the METS document	Metadata file in EAC-CPF format when it is referenced in the METS document. Its needs to be stated in the submission agreement if referencing or embedding are used.	0..*	<mdSec> <mdRef: MDTYPE= ”EAC-CPF” xlink:href=”file:///[EAC-CPF metadata file]” xlink:type=”simple

[32] The Common Specification recommends to avoid embedding.

[33] The coice of using ”///” or ”//” depends on implementations. For example, a Windows file path could be file://C:/somefile.txt and file:///somefile.txt would be the corresponding Unix path in case of absolute paths.

[34] Usually if EAD metadata has been embedded in the METS document once, it will be not referenced as EAD file again.

Element	Definition	Explanation	Card.	METS
				”>
EAC-CPF metadata	Metadata in EAC-CPF format embedded in the METS document	Metadata in EAC-CPF format when it is embedded[35] in the METS document. Its needs to be stated in the submission agreement if referencing or embedding are used.	0..*	<dmdSec> <mdWrap: MDTYPE= ”EAC-CPF”> <xmlData> [EAC-CPF metadata in XML-format]

Example:

```
<dmdSec ID="ID74f5dd4e-0a83-49d7-af50-21a4cc974744">
      <mdRef MIMETYPE="application/xml" xlink:href="file:metadata/descriptive/EAD.xml"
LOCTYPE="URL" CREATED="2015-11-25T14:22:52"
CHECKSUM="58b7855c94bb817af06bc969f7791b357c5ee22946981b8c18cc216384c25628" xlink:type="simple"
ID="IDa9abe6db-84eb-4af3-9d45-ca235a959312" MDTYPE="EAD" CHECKSUMTYPE="SHA-256"/>
</dmdSec>
```

3.4. Administrative metadata

The METS administrative metadata element <amdSec> references to technical and preservation metadata as seen in Figure 7.

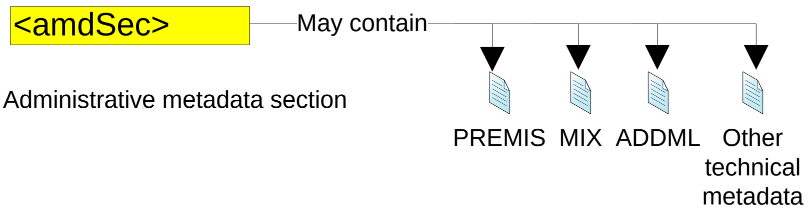

Figure 7: METS administrative metadata

Preservation metadata can be included[36] in the METS package. It is recommended that PREMIS[37] is used for preservation metadata. For further reading:

[35] Usually if EAC-CPF metadata has been embedded in the METS document once, it will be not referenced as EAC-CPF file again.

[36] The Common Specification recommends avoiding embedding preservation or any other type of metadata directly in to the METS file.

- More information about PREMIS can be found at: http://www.loc.gov/standards/premis/ .
- A guide on using PREMIS with METS may be found at: http://www.loc.gov/standards/premis/guidelines-premismets.pdf.
- Decisions made during the use of PREMIS can be recorded using this document: http://www.loc.gov/standards/premis/premis_mets_checklist.pdf

The guide on using PREMIS with METS (referenced above) recommends using the <amdSec> in order to reference PREMIS metadata. The complete rules for all elements and attributes in the <amdSec> are stated in the profile, the specific elements used when referencing or embedding are shown below. However, please note that preservation metadata varies for different content types and therefore best practice guidelines should be applied as required.

Table 4: PREMIS metadata

Element	Definition	Explanation	Card.	METS
PREMIS metadata file	Metadata file in PREMIS format referenced in the METS document	Metadata file in PREMIS format when it is referenced in the METS document. Its needs to be stated in the submission agreement if referencing or embedding are used.	0..*	<amdSec> <digiProvMD> <mdRef: MDTYPE= "PREMIS" xlink:href="file:///[PREMIS metadata file]" xlink:type="simple">
PREMIS metadata[38]	Metadata in PREMIS format embedded in the METS document	Metadata in PREMIS format when it is embedded in the METS document. It needs to be stated in the submission agreement if referencing or embedding are used.	0..*	<amdSec> <digiProvMD> <mdWrap: MDTYPE= "PREMIS"> <xmlData>

[37] We expect that PREMIS will be created automatically by the E-ARK SIP and AIP Creation Tools.

[38] We do not recommend embedding PREMIS metadata into the METS file.

Element	Definition	Explanation	Card.	METS
				[PREMIS metadata in XML-format]

Example:

```
<amdSec ID="ID1a57e479-20e2-4e99-868b-88d0f816d109">
    <digiprovMD ID="ID41d8bb3c-f7c1-4254-aa9f-825009314fb0">
      <mdRef MIMETYPE="text/xml" xlink:href="file:metadata/preservation/premis1.xml"
LOCTYPE="URL" CREATED="2015-11-18T15:50:14"
CHECKSUM="8aa278038dbad54bbf142e7d72b493e2598a94946ea1304dc82a79c6b4bac3d5" xlink:type="simple"
ID="ID58ecdae0-b6af-4ad9-abf1-f6c2971f253a" MDTYPE="OTHER" CHECKSUMTYPE="SHA-256"/>
    </digiprovMD>
</amdSec>
```

3.5. Files

The METS file section element <fileSec> lists all files containing content (may also contain metadata about files) as seen in Figure 8.

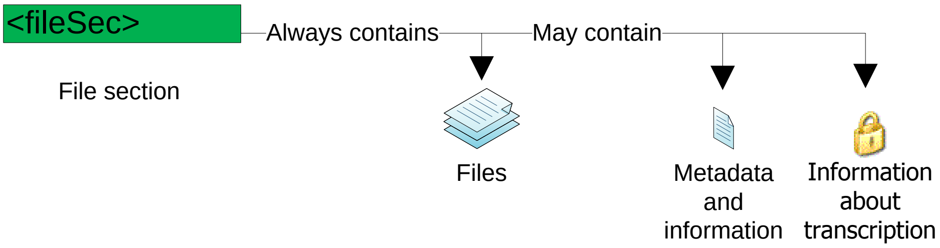

Figure 8: METS files

All files found in the submission package should be referenced once and only once in the METS-document describing the submission. The elements and attributes are the same regardless of the content type submitted.

When describing the content and documentation files in METS they are placed in the fileSec element in one or more fileGrp elements. The fileGrp element can be used for grouping files together in different ways. In this profile we do not group files in different groups, we only use one mandatory fileGrp. Use of more fileGrp’s must be decided in every implementation and described in a METS profile.

Table 5: Files metadata

Element	Definition	Explanation	Card.	METS
Identification of the file	Identification of the file object	A code that uniquely identifies the file inside the METS-document for referencing in the structMap element. Suggested use is a prefix "ID" directly followed by an UUID or GUID or own local identification code. ID follows the rules of the xml attribute XML:ID. Example: *"ID550e8400-e29b-41d4-a716-4466554400bg"*	1	<file ID="[Identification of the file]">
File location	Name of the file	Name of the file and the path to locate it in the package. The file name must use the prefix file:///. The attribute LOCTYPE is mandatory to describe how to find the file and uses a value list present in METS. Example: *"file:///personnelexport.xml"*	1	<file <flocat: LOCTYPE="URL" xlink:href="file:///[File name]" xlink:type="simple">
Date and time	Timestamp for the file[39]	The timestamp seen on the file and used for validating the file. In most cases this is the last modification date. Described using xml type datetime rules. Example: *"2012-04-20T13:30:00+01:00"*	1	<file CREATED="[Date and time]">

[39] It is the timestamp recorded inside the file (i.e. information we can read with JHove and similar tools).

Element	Definition	Explanation	Card.	METS
MIME type[40]	Simplest way of describing a file type	The simplest way of describing a file type. Example: *"text/xml"*	1	<file MIMETYPE="[MIME type]">
File format name[41]	Name of file format	Sometimes a more detailed name needs to be given to the file format when use of PREMIS has not been agreed upon in the submission agreement. Example: *"Extensible Markup Language"* Example: *"PDF/A"* Example: *"ISO/IEC 26300:2006"*	0..1	<file ext:FILEFORMATNAME="[File format name]"
File format version	Version of file format	The version of the file format and use of PREMIS has not been agreed upon in the submission agreement. Example: *"1.0"*	0..1	<file ext:FILEFORMATVERSION="[File format version]"

[40] Media Types, 2015, https://www.iana.org/assignments/media-types/media-types.xhtml

[41] This and all following file elements can be recorded in the SIP by using extension schemas (as shown here), but also by using PREMIS or some other metadata standard.

Element	Definition	Explanation	Card.	METS
Format registry name	Name of the format registry identifying the file format	Name of the format registry identifying the file format when use of PREMIS has not been agreed upon in the submission agreement. According to vocabulary[42] vcTypeOfFormatregistry Example: *" PRONOM”"*	0..1	<file ext:FORMATREGISTRY=”[Format registry name]”
Format key	Key of the file format in the registry	Key of the file format in the registry when use of PREMIS has not been agreed upon in the submission agreement. Example: *“fmt/101"*	0..1 1 If Format registry name is used	<file ext:FORMATREGISTRYKEY=”[Format key]”
File size	Size of the file in bytes	Size of the file in bytes. Example: *”8765324”*	1	<file SIZE=”[File size]”>
Function	Identifies the function of the file	Sometimes it is useful to record information on the function of the file. A vocabulary for stating the “Function” is recommended in every implementation and the vocabulary should be stated in the METS profile. Example: *”Submission file”*	0..1	<file: USE=”[Function]”>

[42] All vocabularies referred in this table are described in the E-ARK METS Profile, http://www.ra.ee/METS/v02/METS.xml

Element	Definition	Explanation	Card.	METS
Checksum type	Used algorithm for creating the checksum	Algorithm used for creating the checksum. Values are predefined in METS. The algorithm to use is to be stated in the submission agreement. Example: *”SHA-256”*	1	<file: CHECKSUMTYPE=” [Checksum type]”>
Checksum value	Calculated checksum for the file	Check sum for the file. Example: *”574b69cf71ceb5534c8a2547f5547d”*	1	<file: CHECKSUM=”[Checksum]”>
Transformation algorithm	Transformation algorithm used for transformation	Transformation algorithm used for any file transformation (decryption/encryption). The algorithm to use is to be stated in the submission agreement. The attribute “TRANSFORMATIONTYPE” is to be used to state the transformation type according to predefined set of values. Example: ”DES”	0..1	<file <transformFile TRANSFORM-TYPE=”decryption” TRANSFORM-ALGORITHM= ”[Transformation algorithm]”>
Transformation key	Transformation key for a transformed file	Transformation key for a transformed file. The attribute “TRANSFORMATIONTYPE” states the transformation type according to a predefined set of values. Example: *”574b69cf71ceb5534c8a2547f5547d”*	0..1 1 If Transformation algorithm is used	<file <transformFile TRANSFORMTYPE= ”decryption” TRANSFORMKEY= ”[Transformation key]”>

Example of the <fileSec> element (root METS file):

```
<fileSec>
   <fileGrp USE="E-ARK files root" ID="IDae911aa8-24f0-4bd8-a684-32044b89d687">
      <fileGrp USE="schemas" ID="IDae911aa8-24f0-4bd8-a684-32056b89d789">
```

```
         <file MIMETYPE="application/xsd" USE="Schema" CHECKSUMTYPE="SHA-256" CREATED="2015-12-
04T09:59:45" CHECKSUM="41d38f0a204e7dbda2838d93ad8eb5cf6bed92acd9c2f06f497faf47722e990d"
ID="ID04918b96-cf9f-41fa-ab13-3d550aaf94f5" SIZE="6814">
            <FLocat xlink:href="file://schemas/METS.xsd" xlink:type="simple" LOCTYPE="URL"/>
         </file>
      </fileGrp>
      <fileGrp USE="representations" ID="IDae055ba8-24f0-4bd8-a684-32056b89d882">
         <fileGrp USE="representation123" ID="IDbc911aa8-24f0-4bd8-a684-32056b89d789">
            <file MIMETYPE="application/xml" USE="Representation METS" CHECKSUMTYPE="SHA-256"
CREATED="2015-12-04T09:59:45"
CHECKSUM="41d38f0a204e7dbda2838d93ad8eb5cf6bed92acd9c2f06f497faf47722e990d" ID="ID04918b96-cf9f-
41fa-ab13-3d550aaf94f5" SIZE="6814">
               <FLocat xlink:href="file://representations/representation123/METS.xsd"
xlink:type="simple" LOCTYPE="URL"/>
            </file>
         </fileGrp>
      </fileGrp>
      <fileGrp USE="documentation" ID="ID7d136e4c-26fe-40da-85a2-67a42efd6b27">
      ...
      </fileGrp>
   </fileGrp>
</fileSec>
```

Example of the <fileSec> element (representation METS file):

```
<fileSec>
   <fileGrp USE="E-ARK files representation representation123" ID="IDae911aa8-24f0-4bd8-a684-
32044b89d687">
      <fileGrp USE="data" ID="IDae911aa8-24f0-4bd8-a684-321556389d687">
         <fileGrp USE="user-defined-data-subfolder" ID="IDae911aa8-24f0-4bd8-a684-32044b89d789">
            <file MIMETYPE="application/pdf" USE="data" CHECKSUMTYPE="SHA-256" CREATED="2015-12-
04T09:59:45" CHECKSUM="41d38f0a204e7dbda2838d93ad8eb5cf6bed92acd9c2f06f497faf47722e990d"
ID="ID04918b96-cf9f-41fa-ab13-3d550aaf94f5" SIZE="6814">
               <FLocat xlink:href="file://data/contentfile.pdf" xlink:type="simple"
LOCTYPE="URL"/>
            </file>
         </fileGrp>
         ...
      </fileGrp>
      <fileGrp USE="documentation" ID="ID7d136e4c-26fe-40da-85a2-67a42efd6b27">
      ...
      </fileGrp>
   </fileGrp>
</fileSec>
```

3.6. Structure

The mandatory METS structural map element <structMap> describes the hierarchical structure for the digital object as seen in Figure 9.

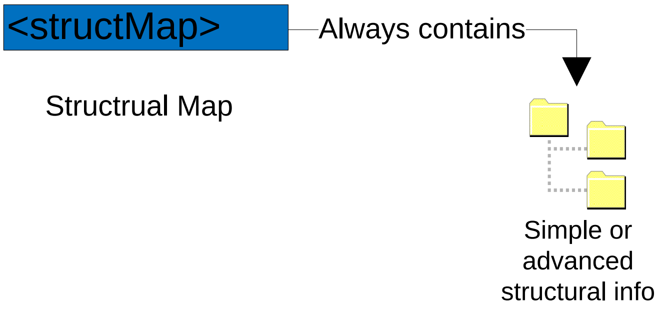

Figure 9: METS structural section

Example:

```
<structMap TYPE="physical" LABEL="E-ARK structural map">
   <div LABEL="9da99df7-2237-48d6-90ef-01d99447c16f">
      <div LABEL="metadata">
         <div LABEL="descriptive">
            <fptr FILEID="IDc04f8f55-802e-4646-b5f9-78b8e864e530"/>
            <fptr FILEID="IDa2da0aa8-bf9c-4a79-a83d-2944cb2031ab"/>
         </div>
         <div LABEL="preservation">
            <fptr FILEID="IDc2ccef19-802e-4646-b5f9-78b8e864e532"/>
            <fptr FILEID="IDa2da11a8-bf9c-4a79-a83d-2944cbfee654"/>
         </div>
      </div>
      <div LABEL="schemas">
         <fptr FILEID="ID845a7a5b-0cfe-43ff-acd9-14f5f0463e28"/>
      </div>
      <div LABEL="representations"/>
         <div LABEL="representations/aip-docs_mig-1">
            <mptr xlink:href="file://representations/aip-docs_mig-1/METS.xml" xlink:type="simple"
LOCTYPE="URL"/>
         </div>
      <div LABEL="representations/aip-imgs_mig-1">
         <mptr xlink:href="file://representations/aip-imgs_mig-1/METS.xml" xlink:type="simple"
LOCTYPE="URL""/>
      </div>
   </div>
</structMap>
```

4. CONTENT-TYPE-SPECIFIC PROFILES

As discussed above (Section 2), an SIP can include content-type specific data and metadata. Types of data files and their structural relationships, and metadata elements vary for different content-types. Metadata is submitted to an archive so that it can support functions in the archive. The metadata created by business systems can be in different structures / formats. The amount and type of available metadata depends very much on the type and owner/developer of the system. As such there are also differences in how much metadata can a specific system or type of system export and in which formats. To deal with these differences there’s the possibility of content type profiles which define detailed metadata requirements beyond the Common Specification and general SIP.

The METS standard used in the E-ARK SIP specification does not offer one single structure in which the content-type specific metadata could be stored as a whole. In order to efficiently use the metadata to support archival functions the SIP defines separate SIP METS sections as containers for the various metadata functions, such as the METS header for package management function, the <dmdSec> for EAD metadata standard (i.e. using ”<dmdSec> for package discovery) and other descriptive metadata standards, the <amdSec> for preservation (PREMIS), technical and other functions and standards. In order to use the submitted metadata it has to be mapped to and referenced from the SIP METS sections.[43] To do this the content-type specific metadata elements need to be mapped to those containers and implemented in the agreed standards. Therefore, complementary E-ARK SIP metadata profiles are needed for the key E-ARK content types. Deliverable D3.3 presents 2 profiles which define how the submitted content-specific metadata should be mapped to the E-ARK SIP structure:

- The SMURF (semantically marked up record format) profile will contain mappings for ERMS (electronic records management systems) based on MoReq2010 and SFSB (simple file-system based) records as described in 4.1 and 4.2.

  Further information about the ERMS content type can be found in the E-ARK SMURF (semantically marked up record format) Profile document which is a part of the deliverable D3.3.[44]

- The SIARD 2.0 profile for relational databases as described in a section 4.3.

Previously listed content types are the content types which E-ARK develops. Other institutions are welcome to add further content types.

All SIPs will need to be transformed into AIPs in the archival ingest process. The SIP to AIP conversion is described in the E-ARK AIP specification.[45]

4.1. Electronic records management systems (ERMS)

The first case represents ERMS records encapsulated in the E-ARK SIP.[46] This profile aims to standardise the export of records management systems into a single easy to use model. The basic workflow is described on Figure 10.

[43] In the case of descriptive metadata it is even additionally integrated with external systems, such as the catalogue in order to support external access to the archive. One might also want to do that for other metadata (e.g. technical or preservation) in order to ease management of the archive.

[44] The SMURF profile for ERMS and SFSB can be found at http://eark-project.com/resources/project-deliverables

[45] E-ARK AIP pilot specification, released January 2016, http://eark-project.com/resources/project-deliverables

[46] The scope of this chapter is to give short introductions; more details are available in a separate document E-ARK SMURF (semantically marked up record format) Profile.

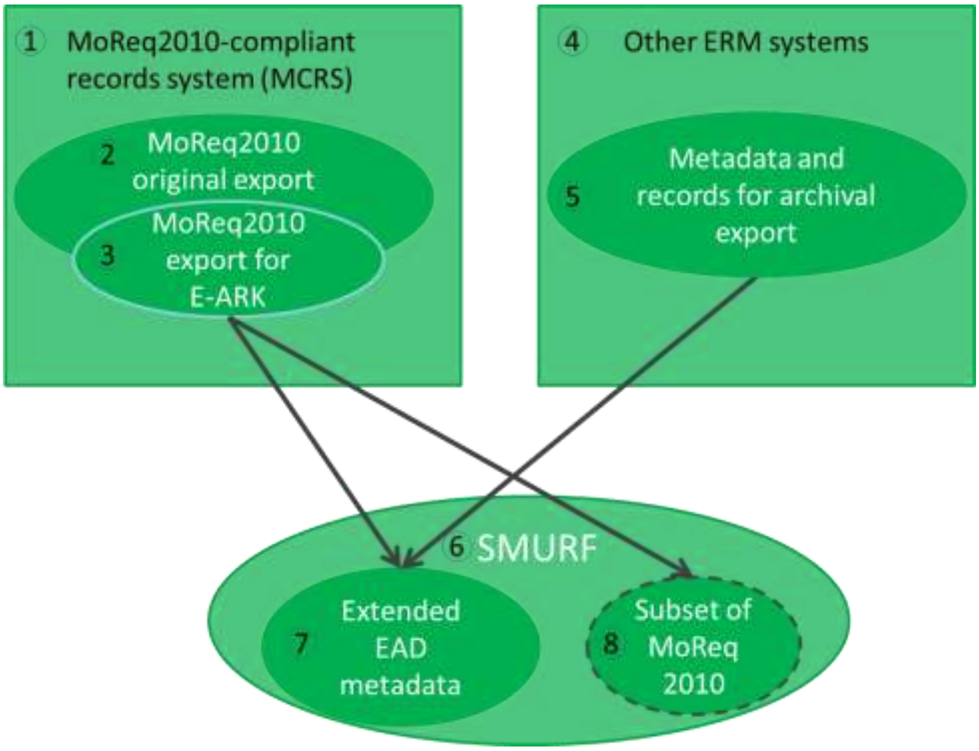
1 MoReq2010-compliant records system (MCRS)
2 MoReq2010 original export
3 MoReq2010 export for E-ARK
4 Other ERM systems
5 Metadata and records for archival export
6 SMURF
7 Extended EAD metadata
8 Subset of MoReq 2010

Figure 10: Extraction at pre-ingest

In case of ERMS we distinguish two scenarios – MCRS and non-MCRS (1, 4). The latter is assumed to be able to export metadata and records in a native export format (5),[47] the first supports in addition the specific MoReq2010 export format (2). Further, the E-ARK project has defined some additional elements (3) needed for archival purposes.

The SMURF profile (6) defines a set of Extended EAD metadata (7)[48] which are created during the pre-ingest phase. In some cases it may be not possible to map all relevant original elements to a set of Extended EAD metadata, therefore some MoReq 2010 elements (8) are allowed[49] for guaranteeing that all required elements are included in the SIP.

The SMURF extraction **should** be complemented with more general information about the information package and **could** be complemented with PREMIS, EAC-CPF metadata as well (Figure 11).

[47] The metadata extracted from a non-MCRS system should be mapped and transformed into the SMURF format by using external mechanisms (i.e. XSL transformation) or by updating the export format to support the SMURF profile.
[48] The EAD extraction will be created automatically by a MCRS.
[49] The E-ARK project does not recommend using MoReq2010 elements in the SMURF profile and therefore only the mapping from MoReq2010 elements to EAD will be provided.

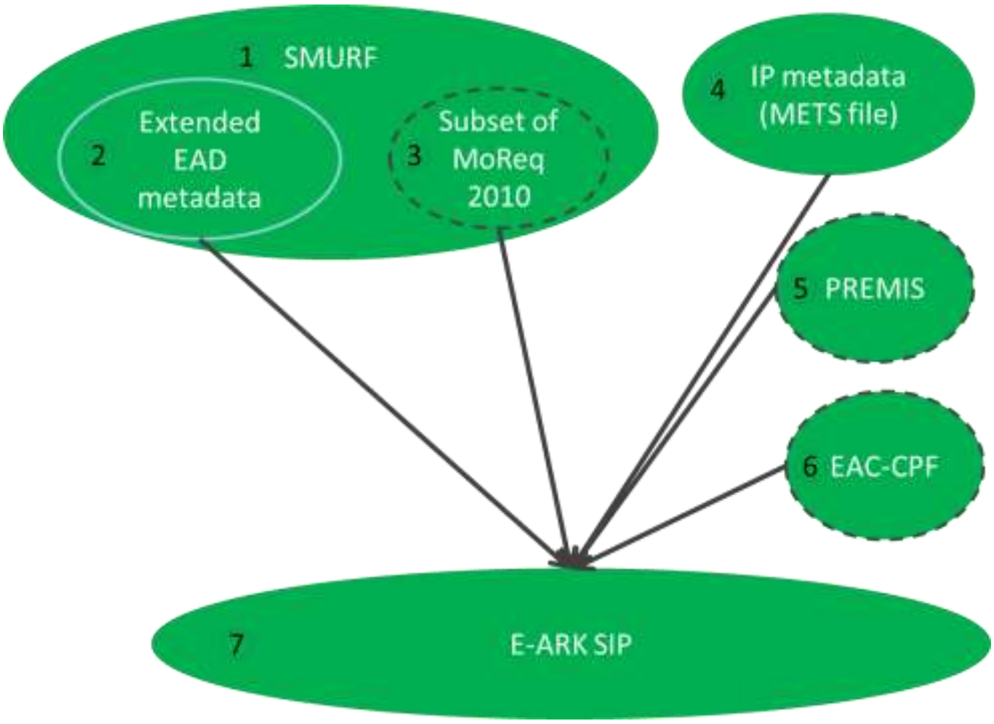

Figure 11: Creation at Pre-Ingest

The SMURF profile (1) includes MoReq2010 metadata that has been mapped to EAD (2) and some additional elements from the E-ARK project.[50] The structural metadata for the submission information package (represented as a METS file) will be added (4) during the SIP preparation process. If possible the EAC-CPF metadata (6) should be created and SIP creation events logged as PREMIS metadata (5). The full E-ARK SIP will consist of items 1, 4 and optionally (5), (6).

4.2. Simple file system based records (SFSB)

The second case represents an encapsulation of computer files into the E-ARK SIP. It is based on an assumption that the files can be described in an extended EAD format[51] (Figure 12).

[50] As referred earlier, the E-ARK does not recommend to use the original MoReq 2010 elements (3) in the SIP.
[51] It is assumed that already available EAD creator tools (e.g. EAD editor at https://github.com/ewg118/eaditor) can be used.

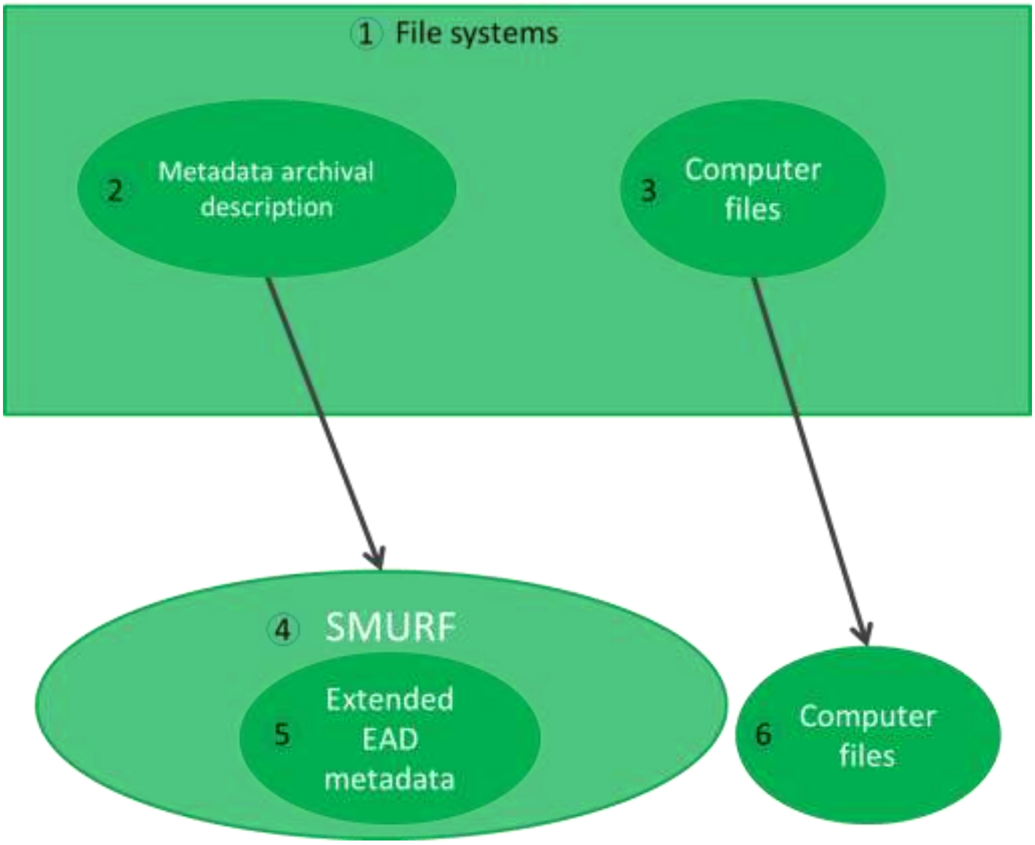

Figure 12: SFSB metadata and computer files

The blocks in the diagram refer to the following. Computer files reside in some file system (e.g. shared drives, 3). The metadata (2) about the files needed for the long time preservation may or may not exist. If the metadata exists then it has to be transformed into the EAD metadata (5). If the metadata does not exist then it has to be created and included in the SIP.

The SMURF metadata **should** be complemented with more general information about the information package and **could** be complemented with PREMIS, EAC-CPF metadata as well to build a full SIP (Figure 13).

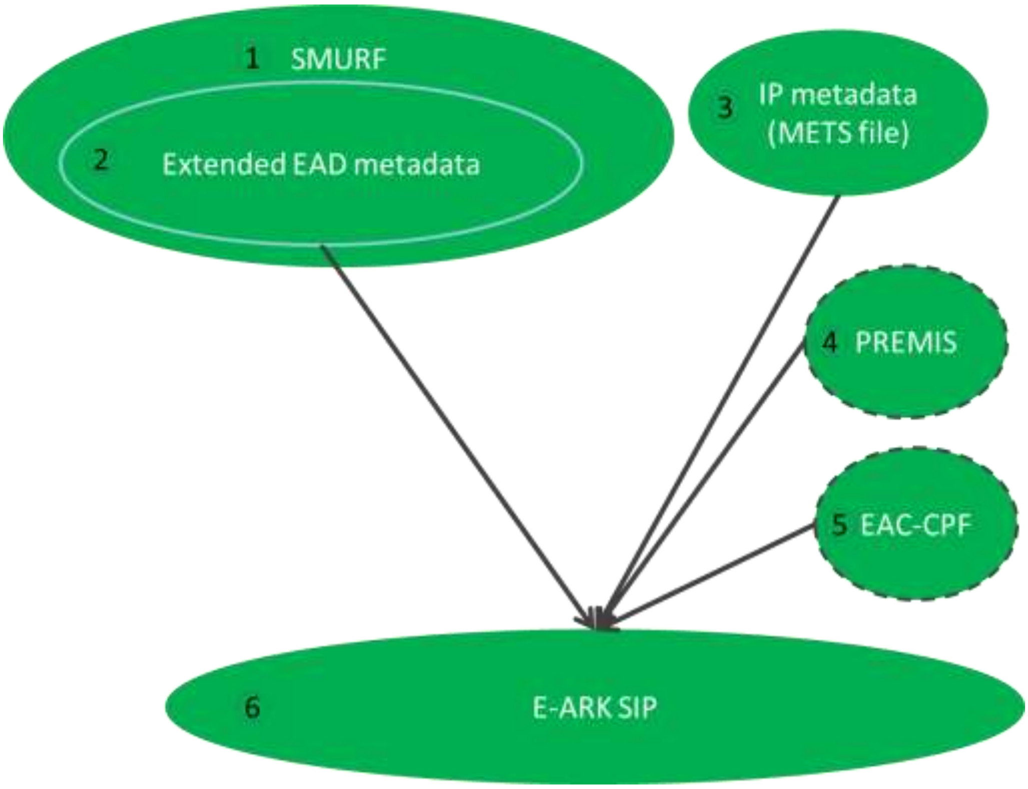

Figure 13: SFSB SIP

The blocks in the diagram refer to the following:

1. The SMURF profile for SFSB records.
2. Archival descriptions following the EAD extended schema for SFSB records.
3. Structural metadata for the submission information package (represented as METS file).
4. If possible then SIP creation events should be logged as PREMIS metadata.
5. If possible then EAC-CPF metadata should be created during the SIP creation process.
6. The E-ARK SIP consists of items 1, 3 and optionally (4), (5).

4.3. Relational databases

The third case represents a relational database encapsulated in the E-ARK SIP. This case structure presumes that the database is previously exported in the SIARD 2.0 format (a harmonised format for database archiving based on SIARD, Figure 14).[52]

[52] The SIARD 2.0 specification (http://www.eark-project.com/resources/specificationdocs/32-specification-for-siard-format-v20/file) represents the SIP profile for the relational databases content type.

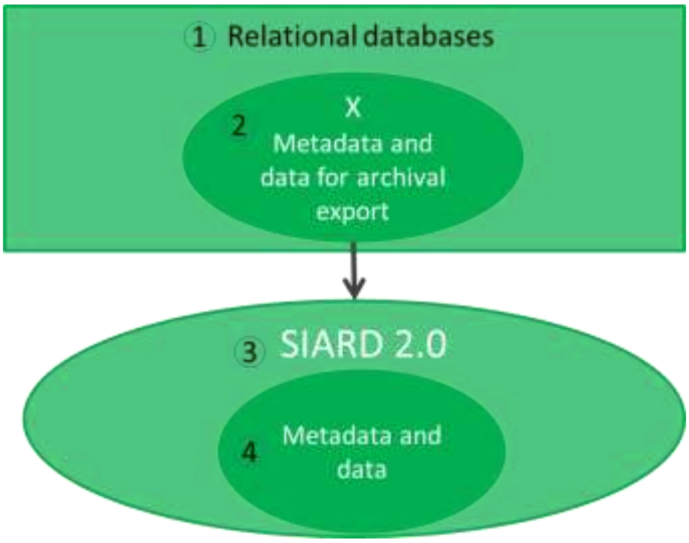

Figure 14: Export to SIARD 2.0

Various relational databases (e.g. Oracle, PostgreSQL, etc.) exist (1). These databases contain the metadata and records in its native format (2) which can be extracted into a standardised format (4) by following SIARD 2.0 (3). The SIARD extraction **should** be complemented with more general information about the information package and **could** be complemented with PREMIS, EAC-CPF, EAD metadata as well (Figure 15).

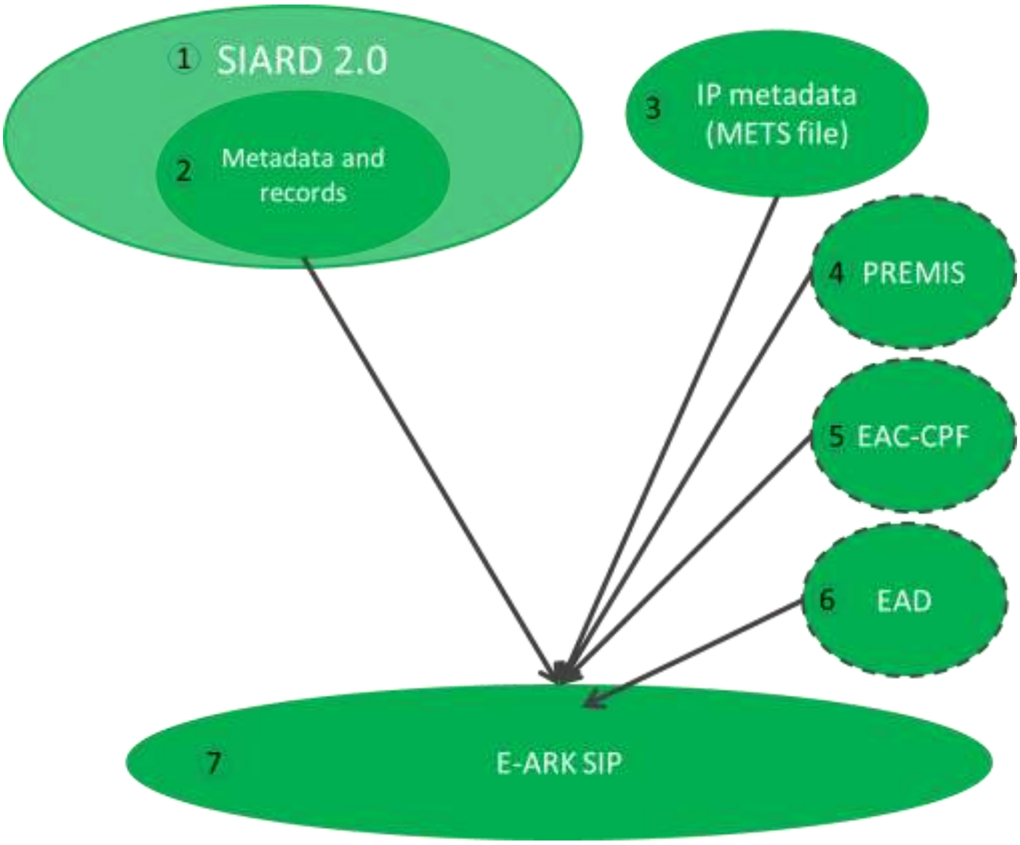

Figure 15: SIARD 2.0 to E-ARK SIP

BLOBs and CLOBs in relational databases

The Figure 14 and Figure 15 show the most common profile for relational databases with metadata and records. However, in some cases there can be binary data in a relational database which will be exported as external files in SIP creation. This might cause a situation where it is necessary to consult with *"Recommendations for external file structure of binary data for the SIARD 2.0 format",* which is a specific and technical recommendation that is not included in the SIARD 2.0 specification. The detailed recommendation can be found in Appendix D, but in order to see if this recommendation is needed a short introduction to where it applies is presented in the following.

Binary data in regard to relational databases is defined as information which is stored in the database as a bit stream following a specific file format. The potentially huge size of binary data within a database can lead to problems in the handling and archival processing of the database. Binary data is mostly referred to as binary large object (BLOB). Similarly large amounts of character data are named CLOB. CLOBs pose a problem due to size more than lack of a proper data type. For the rest of this section CLOBs will be treated as BLOBs.

An example of a relational database with BLOBs could be a database where images are stored.

Databases and the handling of binary data has always been a challenge, regardless of whether the handling was based on:

1. Internal BLOBs - where data is contained in the records.
2. External direct references (path and filename) - where BLOBs are stored as files.
3. External indirect reference (file ID)- where BLOBs are stored as files.
4. Other methods which may exist.

The first method using internal BLOBs is supported in the SIARD 2.0 format, but if a table contains data with BLOBs that are more than 2000 bytes or 2000 characters in size, BLOBs will be produced as separate files and a reference to the location of the individual files stored in the cell content. The SIARD 2.0 format therefore also supports external reference to BLOBs stored as files inside the SIARD table folder structure (i.e. inside the SIARD ZIP package file).

The above scenario will therefore have no consequences regarding the Figure 14 and Figure 15 presented above.

The SIARD 2.0 format, however, also supports methods using external files outside the SIARD table folder structure (i.e. outside the SIARD ZIP package file) but it does not describe in detail how to handle BLOBs if this is the case. It is in this particular scenario that it is advisable to consult the detailed recommendations in Appendix D.

When a SIP creation includes BLOBs stored as external files outside the table folder structure this will have influence on the SIP package since in this case there is not only one SIARD-file containing data from the database, but a SIARD-file and one or several other folders containing the external BLOB files.
A diagram for external files outside the SIARD table folder structure is presented in Figure 16:

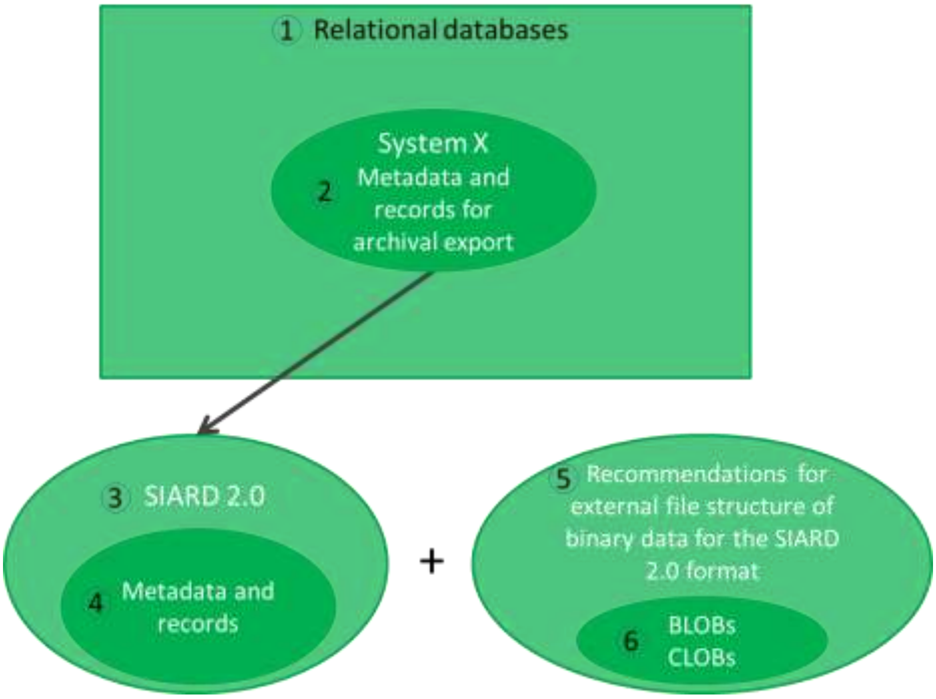
1 Relational databases
2 System X Metadata and records for archival export
3 SIARD 2.0
4 Metadata and records
+
5 Recommendations for external file structure of binary data for the SIARD 2.0 format
6 BLOBs CLOBs

Figure 16: Relational databases with BLOBs/CLOBs stored as external files

1. Various relational databases (e.g. Oracle, PostgreSQL, etc.).
2. The metadata and records in a relational database.
3. The SIARD 2.0 specification.
4. The metadata and records in the SIARD 2.0 format.
5. Recommendations for external file structure of binary data for the SIARD 2.0 format.
6. BLOBs and/or CLOBs stored as external files outside the table structure.

External BLOBs influence on METS file

If there are several data files and folders in the SIP package, this consequently has influence on the IP metadata (METS file). Therefore, *"Recommendations for external file structure of binary data for the SIARD 2.0 format"* also describes how to represent the files in METS. Further information can be found in the SIARD2.0 Profile document.[53]

[53] The SIARD 2.0 specification (http://www.eark-project.com/resources/specificationdocs/32-specification-for-siard-format-v20/file).

5. SUBMISSION AGREEMENT

Interaction between the Archive and Producers is often formalized and guided by a Submission Agreement, which establishes specific details of the interaction such as the type of information submitted, the metadata the Producer is expected to provide, the logistics of the actual transfer of custody from the Producer to the archive, and any access restrictions attached to the submitted material.[54] According to the OAIS model the submission agreement is an agreement reached between an Archive and the Producer that specifies a data model, and any other arrangements needed, for the Data Submission Session. This data model identifies format/contents and the logical constructs used by the Producer and how they are represented on each media delivery or in a telecommunication session.[55]

The E-ARK project acknowledges the importance of submission agreements and requires submission agreements to be referenced in a METS.xml header (<altrecordID:TYPE="SUBMISSIONAGREEMENT">) regardless of its form.[56] The E-ARK SIP does describe a recommended format for a Submission Agreement (Appendix B: Submission Agreement), but of course does not forbid the use of any other Submission Agreement format.

According to the PAIMAS standard the submission agreement should include a complete and precise definition of:[57]

- information to be transferred (e.g., SIP contents, SIP packaging, data models, Designated Community, legal and contractual aspects);
- transfer definition (e.g. specification of the Data Submission Sessions);
- validation definition;
- change management (e.g. conditions for modification of the agreement, for breaking the agreement);
- schedule (submission timetable).

The E-ARK submission agreement is inspired by the PAIMAS requirements and the submission agreement template provided by the National Oceanic and Atmospheric Administration (NOAA). The E-ARK project will propose a list of elements which are recommended to be recorded in the submission agreements created by the project partners (8.2).

[54] Lavoie B, The Open Archival Information System (OAIS) Reference Model: Introductory Guide (2nd Edition), 2014, www.dpconline.org/component/docman/doc_download/1359-dpctw14-02

[55] Reference Model for an Open Archival Information, 2012, public.ccsds.org/publications/archive/650x0m2.pdf

[56] A submission agreement can be delivered in a digital (e.g. PDF or XML file) or an analogue way (e.g. paper document).

[57] PAIMAS, 2004, http://public.ccsds.org/publications/archive/651x0m1.pdf

6. SUMMARY

This document described the proposed general structure for submission information packages by explaining how the E-ARK SIP can be generally constructed by following the common rules developed for all (including archival, dissemination) information packages.

The central section described the metadata sections and elements in these sections.

It also introduced the profiles for the E-ARK SIP and shows how SIPs can manage various content and metadata. The profiles itself can be found in separate documents.

The work introduced in this deliverable is a part of the E-ARK SIP pilot specification (D3.3) focusing on the introductions of the general SIP and content-specific profiles for pilot projects in E-ARK.

7. REFERENCES

1. A Checklist for Documenting PREMIS-METS Decisions in a METS Profile, 2010, URL: http://www.loc.gov/standards/premis/premis_mets_checklist.pdf
2. E-ARK Report on Available Best Practices, 2014, URL: http://eark-project.com/resources/project-deliverables/6-d31-e-ark-report-on-available-best-practices
3. e-SENS (Electronic Simple European Networked Services) project, 2015, URL: http://www.esens.eu/
4. Encoded Archival Context for Corporate Bodies, Persons, and Families, 2015, URL: http://eac.staatsbibliothek-berlin.de/
5. FGS packet structure, 2013, URL:https://riksarkivet.se/Media/pdf-filer/Projekt/FGS_Earkiv_Paket.pdf
6. Guidelines for using PREMIS with METS for exchange, Revised September 17, 2008 URL: http://www.loc.gov/standards/premis/guidelines-premismets.pdf.
7. Media Types, 2015, URL: https://www.iana.org/assignments/media-types/media-types.xhtml
8. METS, 2015, URL: http://www.loc.gov/standards/mets/
9. METS Profile Components, 2011, URL: http://www.loc.gov/standards/mets/profile_docs/components.html
10. METS Profiles, 2012, URL: http://www.loc.gov/standards/mets/mets-profiles.html
11. Producer, Submission Agreements: Glossary of Terms, 2015, URL: http://sites.tufts.edu/dca/about-us/research-initiatives/taper-tufts-accessioning-program-for-electronic-records/project-documentation/submission-agreements-glossary-of-terms/
12. Producer-Archive Interface Methodology Abstract Standard (PAIMAS), 2004, URL: public.ccsds.org/publications/archive/651x0m1.pdf
13. Producer-Archive Interface Specification (PAIS) – CCSDS, 2014, URL: public.ccsds.org/publications/archive/651x1b1.pdf
14. Records Creator, Submission Agreements: Glossary of Terms, 2015, URL: http://sites.tufts.edu/dca/about-us/research-initiatives/taper-tufts-accessioning-program-for-electronic-records/project-documentation/submission-agreements-glossary-of-terms/
15. Reference Model for an Open Archival Information System (OAIS), 2012, URL: public.ccsds.org/publications/archive/650x0m2.pdf

8. APPENDICIES

8.1. Appendix A: Quality requirements for a submission information package

Every SIP in the E-ARK project should follow the requirements set out in the common specification for information packages.

8.1.1. General requirements

Requirement 1.1: *The SIP Specification MUST allow for the inclusion of any data or metadata, regardless of its type or format, in the Information Package.*

Requirement 1.2: *The SIP Specification MUST support the transfer of the Information Package by any means, methods or tools.*

Requirement 1.3: *The SIP Specification MUST not restrict the cardinality of transformations between SIPs, AIPs and DIPs.*

8.1.2. Identification of the Information Package

Requirement 2.1: *Any Information Package MUST allow for the identification of it as a SIP, AIP or DIP.*

Requirement 2.2: *Any Information Package MUST allow for the identification of the content type of its data.*

Requirement 2.3: *Not relevant for the SIP.*

Requirement 2.4: *It SHOULD be possible to identify any Information Package globally uniquely.*

Requirement 2.5: *Any Information Package MUST, within the package, allow for the unique identification of all its components.*

Requirement 2.6: *Not relevant for the SIP.*

8.1.3. Structure of the Information Package

Requirement 3.1: *The structure of the Information Package MUST allow for the separation of data and metadata*

Requirement 3.2: *The structure of the Information Package SHOULD allow for the separation of different types of metadata.*

Requirement 3.3: *The structure of the Information Package SHOULD allow for the separation of multiple representations of data.*

Requirement 3.4: *The structure of the Information Package MUST be extensible to meet additional local or business-specific needs.*

Requirement 3.5: *Any Information Package MUST follow a common logical structure for its data, metadata and all other components.*

Requirement 3.6: *Any Information Package SHOULD follow a common physical structure for its data, metadata, and all other components.*

8.1.4. Information Package Metadata

Requirement 4.1: *Any Information Package MUST, to the largest possible extent, use internationally recognised and standardised metadata schemas for administrative, preservation, structural and technical metadata.*

Requirement 4.2: *Information Package metadata MUST allow validating the structure and content of any Information Package in terms of integrity, fixity and syntax.*

Requirement 4.3: *Any Information Package MUST allow for including any additional metadata.*

8.1.5. Implementation of the Common Specification

Requirement 5.1: *Any implementation of the SIP Specification SHOULD support Information Packages regardless of their size.*

Requirement 5.2: *Any implementation of the SIP Specification MUST be machine-readable.*

Requirement 5.3: *Any implementation of the SIP Specification SHOULD be human-readable.*

Requirement 5.4: *Any implementation of the SIP Specification MUST not limit the use of preservation methods.*

8.2. Appendix B: Submission Agreement

Table 6: Submission agreement

Elements			**Explanations**
PROJECT INFORMATION			
Project			Elements of a transfer project.
	Project Name		Name of the transfer project (e.g. Transfer I, 2016).
	Project ID		Identification code of the transfer project (e.g. 201601122044).
CHANGE MANAGEMENT			
VERSION / REVISION			Elements for tracking the changes in versions of the submission agreement.
		RELEASE DATE	The date of the version.
		CHANGE AUTHORITY	The information about the person who changed the submission agreement (e.g. John Smith (The National Archives of Estonia)).
		CHANGE DESCRIPTION	A short textual description of the change.
		SECTION(S) AFFECTED	This element is meant for recording more detailed information about changes.
PRODUCER, ARCHIVE, DESIGNATED COMMUNITY			
Producer Organization			Elements describing the Producer.
	Organization name	Organization name	The official name of the organization
	Main Contact		Elements describing the main contact of the transfer project.
		Address	The address of the main contact.

Elements			**Explanations**
		Telephone	The telephone number of the main contact.
		E-mail	The e-mail of the main contact.
		Additional Information	Meant for recording any additional information needed to describe the contact.
	Individual Contacts		Elements describing other individual contacts of the organization.
		Name	The full name of the contact person.
		Role	The role of the contact person.
		Telephone	The telephone number of the contact person.
		E-mail	The e-mail of the contact person.
		Additional Information	Meant for recording any additional information needed to describe the contact.
Archive Organization			Elements describing the Archive.
	Organization name		The official name of the organization
	Main Contact		Elements describing the main contact of the transfer project.
		Address	The address of the main contact.
		Telephone	The telephone number of the main contact.
		E-mail	The e-mail of the main contact.
		Additional Information	Meant for recording any additional information needed to describe the contact.
	Individual Contacts		Elements describing other individual contacts of the organization.
		Name	The full name of the contact person.
		Role	The role of the contact person.
		Telephone	The telephone number of the contact person.
		E-mail	The e-mail of the contact person.

Elements			Explanations
		Additional Information	Element for recording any additional information needed to describe the contact.
Designated Community			Elements describing the Designated Community.
	Description		The textual description of the skills and knowledge base of the designated community.
	Individual Contacts		Elements describing the individual contacts of the designated community.
		Name	The full name of the contact person.
		Role	The role of the contact person.
		Telephone	The telephone number of the contact person.
		E-mail	The e-mail of the contact person.
		Additional Information	Meant for recording any additional information needed to describe the contact.
SUBMISSION INFORMATION PACKAGE			
Content and metadata			Elements describing the content and metadata of the submission information package.
	Description		A description of data origination, content and coverage.
	Platform Information		A short description of the source system.
	Representation Information		A description of the means to represent the data content (e.g. referencing to data dictionaries, decoding software etc.).
	Preservation Descriptive Information		A description for keeping data independently understandable (e.g. processing history, platform documentation, algorithm information, technical reports, user manuals, etc.).
	Supplemental Information		Meant for recording any additional information needed to describe the content or metadata of the SIP.

Elements		Explanations
	Access Constraints	A description of access restrictions and other legal or contractual access aspects.
Data Model		Elements describing the agreements for the SIP data model.
	Content Type	A short description of the content type (e.g. ERMS content).
	Reference	A reference to the full agreed data model description.
	Additional Information	A description of any other additional information (e.g. description of the physical folder structure of the SIP) related to the data model.
SUBMISSION SESSION INFORMATION		
Submission Session		Elements describing the agreements for the submission session.
	Submission Method	The description of the submission method (e.g. through a digital interface, a physical transfer).
	Delivery Schedule	A description of a delivery schedule (a submission session may have a routine or a complex schedule).
	Data Submission Inventory	A description of the complete inventory of data objects (and other items) in the submission session.
INGEST		
Submission Reception		Elements describing the agreements for the ingest.
	Validation	A description of procedures for the quality assurance.
	Error Handling	A description of procedures for the error handling.
	Receipt Confirmation	A description of the receipt confirmation.
SUBMISSION RISKS		

Elements			Explanations
Risks			Elements describing the risks and mitigation options of the submission.
	Risk Factor		Meant for listing all risk factors (e.g. the designated community is not properly defined) of the submission.
	Mitigation Option		Meant for listing all mitigation options (e.g. define the designated community together with producers) for the risks.

8.3. Appendix C: Terminology

Archival creator[58]	An organization unit or individual that creates records and/or manages those records during their active use.
Archive*	An Organisation that intends to preserve information for Access and use by a Designated Community.
Content type	In the e-government context of E-ARK we use "content type" to specify the functional nature of the original business system which was used to manage the data. Examples of content types would therefore include accounting system, personnel database, ERMS, a more general relational database, or even unstructured content. For E-ARK project purposes we will develop and support the following specific "content types": ERMS, relational databases, and unstructured files.
Delivering organisation	The organisation delivering the package to the archive. For stating and extending the information use of the "Producer organisation name" and "Submitting organisation name" elements is recommended.
ERMS	A type of content management system known as an electronic records management system.
Information Package*	A logical container composed of optional Content Information and optional associated Preservation Description Information. Associated with this Information Package is Packaging Information used to delimit and identify the Content Information and Package Description information used to facilitate searches for the Content Information.
Ingest Functional Entity*	The OAIS functional entity that contains the services and functions that accept Submission Information Packages from Producers, prepares Archival Information Packages for storage, and ensures that Archival Information Packages and their supporting Descriptive Information become established within the OAIS.
OAIS*	The Open Archival Information System is an archive (and a standard: ISO 14721:2003), consisting of an organisation of people and systems that has accepted the responsibility to preserve information and make it available for a Designated Community.
Producing organisation[59]	The organizational unit or individual that has the authority to transfer records to an archive. Usually the producer is also the records creator, the organizational unit or

[58] Records Creator, Submission Agreements: Glossary of Terms, 2015, http://sites.tufts.edu/dca/about-us/research-initiatives/taper-tufts-accessioning-program-for-electronic-records/project-documentation/submission-agreements-glossary-of-terms/

[59] Producer, Submission Agreements: Glossary of Terms, 2015, http://sites.tufts.edu/dca/about-us/research-initiatives/taper-tufts-accessioning-program-for-electronic-records/project-documentation/submission-agreements-glossary-of-terms/

	individual that created and managed the records during their active use. This is not always the case, sometimes the producer is different from the records creator. For example: An author dies and her literary executor gains the authority to transfer her papers to an archive. The author is the records creator and the literary executor is the producer. For example: Department X gets reorganized out of existence and Department Y, which takes over the functional responsibilities of Department X, gains the authority to transfer the records of Department X to the archive. Department X is the records creator and Department Y is the producer. Counter example: The Department of Widget Science transfers some of its own records to the archive. The Department of Widget Science is the records creator and the producer.
Submission Information Package (SIP)*	An Information Package that is delivered by the Producer to the OAIS for use in the construction or update of one or more AIPs and/or the associated Descriptive Information.
Submitting organisation	Name of the organisation submitting the package to the archive. Extends the delivery information since it may be the case that the content of a creator is held by another part of the organisation.

***** Reference Model for an Open Archival Information System (OAIS), 2012, http://public.ccsds.org/publications/archive/650x0m2.p

8.4. Appendix D: Recommendations for external file structure of binary data for the SIARD 2.0 format

This appendix is a recommendation for the external file structure of binary data for the SIARD 2.0 format. The SIARD 2.0 format has its own specification, which deliberately does not specify the exact external file structure of binary data. This is avoided in order to get a high degree of durability and longevity for the specification, which requires flexibility. Instead, such details are left to recommendations like this one.

This recommendation will be implemented in the E-ARK open source tool Database Preservation Toolkit (DPT) for the export of relational databases to the SIARD 2.0 format.

Methods for binary data handling in relational databases

Binary data in regard to relational databases is defined as data for which no simple datatype (such as integer or date) exists. In addition the size of binary data is also important due to efficient handling in databases. Binary data is mostly referred to as binary large object (BLOB). Similarly large amounts of character data are named CLOB; they pose a problem due to size more than lack of a proper data type. For the rest of this recommendation CLOBs will be treated as BLOBs.

Databases and handling of binary data has always been a challenge, regardless of the handling was based on: a: internal BLOBs, b: external direct references (path and filename) or c: external indirect reference (file ID). Other methods may exist.

Binary data handling in SQL Standards

The first method using internal BLOBs has been available for many versions of the SQL standard. It is supported by all current relational database management systems.

The other method using external files has been available since SQL:2003 and is named Management of External Data (SQL/MED). It is still poorly supported by the current relational database management systems, and maybe due to lack of detailed specification in the SQL standard those RDBMS that support it do it differently.

Binary data handling in the SIARD 2.0 format

The SIARD 2.0 format is based on among others standards on the SQL:2008 standard.

Support for internal BLOBS (ISO/IEC 9075-2:2008 - BLOBS) in the SIARD 2.0 *format*

The SIARD 2.0 format specification supports the method in SQL:2008 using internal BLOBS (ISO/IEC 9075-2:2008) as did SIARD 1.0 (SQL:1999).
The SIARD 2.0 format supports BLOBS stored as files inside the SIARD table folder structure and describes this in details in the SIARD 2.0 format specification (similar to SIARD 1.0).
The SIARD 2.0 format supports BLOBS stored as files outside the SIARD table folder structure (new feature in SIARD 2.0), but does not describe this in details, it is left for recommendations like this one.

Support for external files (ISO/IEC 9075-9:2008 – SQL/MED) in the SIARD 2.0 format

The SIARD 2.0 format does not support the method in SQL:2008 using external files (ISO/IEC 9075-9:2008 – SQL/MED) due to lack of specification in the SQL standard and especially due to lack of support in the RDBMS. The SIARD 2.0 format does not forbid the use of methods using external files, but leaves it to recommendations like this one to add such support.

Recommendation for the folder structure for BLOBS stored outside the SIARD table folder structure

This recommendation specifies the folder structure for files stored externally to the SIARD folder structure, i.e. outside of the SIARD ZIP package file.
Due to file system and media limitations users might be required to limit: :

- the amount of files in one folder (such as 10,000 files)
- the size of files in one folder hierarchy (such as 1 TB)

In order to get effective and easy file and folder handling (such as copying, packing, hashing, mounting) new folders are created when reaching a folder amount or folder size limitation, whatever comes first. These folders are placed at the same top level to get effective and easy file and folder handling.

Below is a "generic" example for a single column *k* in a single table *j*, in which the file amount limit is reached before the file size limit. [xxx] to be replaced including the [ ]. (very simple BNF)

```
[databaseName].siard <!- packaged as a ZIP file ->
   content/
   header/
      metadata.xml
      metadata.xsd


[databaseName]_lobseg_[h]/..
   content/schema[i]/table[j]/lob[k]/record[1].bin
   ..
   content/schema[i]/table[j]/lob[k]/record[FILEAMOUNTLIMIT].bin
   <!-- file amount limit per folder reached -->
[databaseName]_lobseg_[h+1]/
   content/schema[i]/table[j]/lob[k]/record[FILEAMOUNTLIMIT+1].bin
   ..
   content/schema[i]/table[j]/lob[k]/record[FILEAMOUNTLIMIT+q].bin
   <!-- total file size limit per folder reached -->
[databaseName]_lobseg_[h+2]/
   content/schema[i]/table[j]/lob[k]/record[FILEAMOUNTLIMIT+q+1].bin
```

Note that the base value for records and tables is 0, whereas it is 1 for columns and row.

Below is an example of the tables in example database Northwind with external BLOBs.

```
Orders (table0),
Products (table1),
Categories (table2), - this table contains BLOBS which exceed 2000 bytes or 2000 characters.
Shippers (table3),
Employees (table4), - this table contains BLOBS which exceed 2000 bytes or 2000 characters.
Territories (table5),
CustomerDemographics (table6),
CustomerCostumerDemo (table7),
Suppliers (table8),
EmployeeTerritories (table9),
Customers (table10),
Sysdiagrams (table11),
Region (table12),
```

Only the table "Categories" is shown:

Table 7: Categories

CategoryID	CategoryName	Description	Picture
1	Beverages	Soft drinks, coffees, teas, beers, and ales	BLOB (Size: 10151)
2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	BLOB (Size: 12107)
3	Confections	Desserts, candies, and sweet breads	BLOB (Size: 12007)
4	Dairy Products	Cheeses	BLOB (Size: 9756)
5	Grains/Cereals	Breads, crackers, pasta, and cereal	BLOB (Size: 12131)
6	Meat/Poultry	Prepared meats	BLOB (Size: 11280)
7	Produce	Dried fruit and bean curd	BLOB (Size: 12338)
8	Seafood	Seaweed and fish	BLOB (Size: 12069)

The file amount limit per folder is in this case set to 4 and the total file size limit per folder is set to 45,000 bytes (unrealistic but useful for an example).

Row 1, 2, 3 and 4 will have its LOBs from column 4 ('Picture') (record0.bin, record1.bin, record2.bin, record3.bin) stored together in a folder named **Northwind_lobseg_0**.
Hereby the *file amount limit* of 4 is reached, no more files can be stored in this folder, and therefore a new folder is created named **Northwind_lobseg_1**.
Row 5, 6, and 7 will have its LOBs from column 4 ('Picture') (record4.bin, record5.bin, record6.bin) stored together in a folder named **Northwind_lobseg_1**.
Row 8 will *not* have its LOB from column 4 ('Picture') (record7.bin) stored together with the ones from row 5, 6 and 7 in the folder named **Northwind_lobseg_1**. Not because the *file amount limit* of 4 is reached, but because the *accumulated file size per folder limit* of 45,000 is reached.
The LOBs from row 5, 6 and 7 have respectively a size of 12,131; 11,280 and 12,338 accumulated to 35,749. Adding the LOB from row 8 with a size of 12,069 to the 35,749 of row 5, 6 and 7 would accumulate to 47,818 and break the *accumulated file size per folder limit* of 45,000. Therefore a new folder is created named **Northwind_lobseg_2**, and Row 8 will have its LOB from column 4 ('Picture') (record7.bin) stored in it.

Below is an illustration of the example:

```
Northwind.siard <!- packaged as a ZIP file ->
   content/
   header/
      metadata.xml
      metadata.xsd
```

```
Northwind_lobseg_0/
   content/schema0/table2/lob4/record0.bin <!- row 1, col 4 has record0.bin ->
   content/schema0/table2/lob4/record1.bin
   content/schema0/table2/lob4/record2.bin
   content/schema0/table2/lob4/record3.bin <!-- amount limit reached -->
```

```
Northwind_lobseg_1/
   content/schema0/table2/lob4/record4.bin
   content/schema0/table2/lob4/record5.bin
   content/schema0/table2/lob4/record6.bin <!-- size limit reached -->
```

```
Northwind_lobseg_2/
   content/schema0/table2/lob4/record7.bin
```

Below is an extract of metadata.xml using the above folder structure:

```
<dbname>Northwind</dbname>
<dataOwner>...</dataOwner>
<dataOriginTimespan>2015</dataOriginTimespan>
<lobFolder>file:///S:/Archives/Northwind/</lobFolder>
```

Note the lobFolder content – it assumes that the **Northwind_lobseg_[n]**/ folders reside in a folder named **Northwind**.

Below is an extract of table2.xml using the above folder structure:

```
<row><c1>1</c1><c2>Beverages</c2><c3>Soft drinks, coffees, teas, beers, and ales</c3>
   <c4>file=”file:///Northwind_lobseg_0/content/schema0/table2/lob4/record0.bin”
   length="10151" messageDigest="md574f24080fc9d234d3ac221b8e743c763 ”</c4></row>

<row><c1>2</c1><c2>Condiments</c2><c3>Sweet and savory sauces, relishes, spreads, and seasonings</c3>
   <c4>file=”file:///Northwind_lobseg_0/content/schema0/table2/lob4/record1.bin”
   length="12107" messageDigest="md522a0cbe8960b78ce48b07a285ce69e3c ”</c4></row>

<row><c1>3</c1><c2>Confections</c2><c3>Desserts, candies, and sweet breads</c3>
   <c4>file=”file:///Northwind_lobseg_0/content/schema0/table2/lob4/record2.bin”
   length="12007" messageDigest="md53e2f2028a9147c29bdcd36ed4e5f25b3 ”</c4></row>

<row><c1>4</c1><c2>Dairy Products</c2><c3>Cheeses</c3>
   <c4>file=”file:///Northwind_lobseg_0/content/schema0/table2/lob4/record3.bin”
   length="9756" messageDigest="md512f588040e11cc2021ea37d46aa10c51 ”</c4></row>

<row><c1>5</c1><c2>Grains/Cereals</c2><c3>Breads, crackers, pasta, and cereal</c3>
   <c4>file=”file:///Northwind_lobseg_1/content/schema0/table2/lob4/record4.bin”
   length="12131" messageDigest="md5e2d8ef03e1b24edd946820dbbf44fdfd ”</c4></row>

<row><c1>6</c1><c2>Meat/Poultry</c2><c3>Prepared meats</c3>
   <c4>file=”file:///Northwind_lobseg_1/content/schema0/table2/lob4/record5.bin”
   length="11280" messageDigest="md5814a3eb95253c08137f70bcfc279e00f ”</c4></row>

<row><c1>7</c1><c2>Produce</c2><c3>Dried fruit and bean curd</c3>
   <c4>file=”file:///Northwind_lobseg_1/content/schema0/table2/lob4/record6.bin”
   length="12338" messageDigest="md5ee114cd7700f566b1f7c7e8e0f68ca0f ”</c4></row>

<row><c1>8</c1><c2>Seafood</c2><c3>Seaweed and fish</c3>
   <c4>file=”file:///Northwind_lobseg_2/content/schema0/table2/lob4/record7.bin”
   length="12069" messageDigest="md52de1ac4c4e8ebb853e17db01af3fb7c3 ”</c4></row>
```

Note that the messageDigest indicates hexadecimal values and it is therefore not of strict importance whether they are set in upper or lower case. However, lower cases are mostly used and enforced, see e.g. RFC 2831 https://www.ietf.org/rfc/rfc2831.txt.

Recommendation for using METS to index folders containing BLOBS

In the previous section the recommended folder structure for BLOBs stored outside the SIARD table was specified. This structure allows for the segmentation of large amounts of BLOBs from a database into folders at the same hierarchical level (top level) to get effective and easy file and folder handling. These folders can easily be distributed over many different media to overcome storage and I/O limitations, but will then need an index in order to find these folders.

This section is a recommendation for using METS to index such folders.

Several methods have been described and discussed, and in order to limit the deviation within the E-ARK project we have decided to use the same method which is used for indexing and segmenting IPs in the common specification, which in turn is based on the divided structure of AIPs.

Using METS to refer between child IP(s) and parent IP

The child IP(s) refer to the parent IP using the value of the xlink:href attribute in the <mptr> element. This value must equal the value of the OBJID attribute for the <mets> element in the parent IP mets file. Therefore the same type of ID must be used to these to attributes.

Note that the ID attribute for the <mets> file is for identifying that specific element in the mets file whereas the OBJID attribute in the <mets> element is for identifying the whole mets file (seen as an object.) (according to the mets standard).

Here follows a partial example, where the value of the xlink:href attribute in the <mptr> element (inside the <div> element inside the <structMap> element) is “ID.AVID.RA.18005”.

Extract of <mprt> element from child IP mets file:

```
<mets:structMap TYPE="logical" LABEL="parent IP">
   <mets:div LABEL="IP parent identifier">
      <mets:mptr xlink:href="ID.AVID.RA.18005" xlink:title="Referencing the parent IP of this child IP"
LOCTYPE="OTHER" OTHERLOCTYPE="Some ID scheme" ID="ID755d4d5f-5c5d-4751-9652-fcf839c7c6f2"/>
```

The LOCTYPE attribute has the value “OTHER” and the OTHERLOCTYPE has the value “Some ID scheme”. Instead of “OTHER” for LOCTYPE the value URN, URL, PURL etc. could have been used.

When OTHER is used for LOCTYPE, OTHERLOCTYPE should be used to define the type (and preferably more meaningful than just using “Some ID scheme” in this example).

The value “ID.AVID.RA.18005” must now match the value of the OBJID attribute for the <mets> element in the parent IP mets file.

Extract of mets element from parent IP mets file:

```
mets:mets xmlns:mets="http://www.loc.gov/METS/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.loc.gov/METS/ schemas/mets.xsd"
OBJID="ID.AVID.RA.18005" ID="ID.AVID.RA.18005.mets.element" TYPE="SIARD2.0 INDEX"
PROFILE="http://SIARD2.0.xml" LABEL="SIARD 2.0 parent">
```

The parent IP’s mets file has the value “ID.AVID.RA.18005” for the attribute OBJID in the <mets> element which matches the value of the xlink:href attribute in the <mptr> element in the child ID’s mets file, cf above.

Using METS to refer between parent IP and child IP(s)

Apart from referring from child IP to parent IP the mets file is also used to refer from parent to child(ren). Thereby not only does the child know its parent, but the parent also knows its children.

The mechanism used to refer from parent to child is by using the value of the xlink:href attribute in the <mptr> element in the mets file of the parent IP.

This value must match the value of the ID attribute of the <file> element in the mets file of the parent IP. The <file> element has a sub element named <FLocat> which has a xlink:href attribute which has the value which is the location of the file for the child IP.

This matching of ID and location is necessary to find the location of the file.

Note that the value of the xlink:href attribute in the <mptr> element for a certain child IP in the mets file of the parent IP is set equal to the value of the OBJID attribute of the <mets> element in the mets file of the child IP. This is sufficient for linking the IDs of the parent and the child, but without the matching value of the ID attribute of the <file> element in the mets file of the parent IP the location of the child IP is not known.

Extract of mptr element from parent IP mets file:

```
<mets:div LABEL="child IPs">
   <mets:div LABEL="child IP">
      <mets:mptr xlink:href="ID_Northwind_lobseg_0_mets" xlink:title="Reference to a child IP"
LOCTYPE="OTHER" OTHERLOCTYPE="someID" ID="IDd98e416f-55a7-4237-8d45-59c22d221669"/>
```

The xlink:href value “ID_Northwind_lobseg_0_mets” is equal to the <mets:file> attribute ID value “ID_Northwind_lobseg_0_mets” in the following:

Extract of file element from parent IP mets file:

```
<mets:fileGrp USE="child IP" ID="IDf98e416f-55a7-4237-8d45-59c22d221663">
          <mets:file ID="ID_Northwind_lobseg_0_mets" MIMETYPE="xml/mets"
                    OWNERID="ID.AVID.RA.18005.NORTHWIND" CHECKSUMTYPE="MD5"
                    CHECKSUM="c84642ff066720a5b8f4193a7f213782">
             <mets:FLocat LOCTYPE="URL" xlink:type="simple"
                    xlink:href="file:///Northwind_lobseg_0/mets.xml"/>
```

The value of the attribute xlink:href is “file:///Northwind_lobseg_0/mets.xml"/>”.
Extract of <mets> element from the child IP whose mets file has the location Northwind_lobseg_0/mets.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<mets:mets xmlns:mets="http://www.loc.gov/METS/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.loc.gov/METS/
schemas/mets.xsd" OBJID="ID_Northwind_lobseg_0_mets" ID="ID.AVID.RA.18005.seg.0.mets.element"
TYPE="SIARD2.0 INDEX" PROFILE="http://SIARD2.0.xml" LABEL="SIARD 2.0 child IP">
```

The OBJID has the value “ID_Northwind_lobseg_0_mets”

Below is an example for the Northwind database

Parent IP mets file

```
<?xml version="1.0" encoding="UTF-8"?>
<mets:mets xmlns:mets="http://www.loc.gov/METS/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.loc.gov/METS/ schemas/mets.xsd"
OBJID="ID.AVID.RA.18005" ID="ID.AVID.RA.18005.mets.element" TYPE="SIARD2.0 INDEX"
PROFILE="http://SIARD2.0.xml" LABEL="SIARD 2.0 parent">
  <!-- note that the ID and OBJID do not need to be the same - we need to decide what ID should be explicitly
understandable -->
  <mets:metsHdr CREATEDATE="2015-06-13T14:40:00+01:00" RECORDSTATUS="NEW">
    <mets:agent ROLE="CREATOR" TYPE="OTHER" OTHERTYPE="SOFTWARE">
      <mets:name>SIARD 2.0 index</mets:name>
      <mets:note>Created by XX Software following E-ARK recommendation for external LOB handling in SIARD
2.0</mets:note>
    </mets:agent>
    <mets:metsDocumentID>IP parent SIARD content</mets:metsDocumentID>
  </mets:metsHdr>
  <mets:fileSec>
    <mets:fileGrp ID="ID_SIARD_FILES">
      <!-- the fileGrp for all SIARD files -->
      <mets:fileGrp ID="ID_SIARD_DDL">
        <!-- the fileGrp for the central SIARD file named [databasename].siard -->
        <mets:file ID="ID_NORTHWIND.SIARD" MIMETYPE="OTHER/SIARD"
OWNERID="ID.AVID.RA.18005.NORTHWIND" CHECKSUMTYPE="MD5"
CHECKSUM="A1FB456A233542293459034589034534">
          <mets:FLocat LOCTYPE="URL" xlink:type="simple" xlink:href="file:///northwind.siard"/>
          <mets:transformFile TRANSFORMALGORITHM="ZIP64" TRANSFORMORDER="1"
TRANSFORMTYPE="decompression"/>
          <!--ZIP32 is also allowed for SIARD-->
        </mets:file>
      </mets:fileGrp>
      <mets:fileGrp USE="child IP" ID="IDf98e416f-55a7-4237-8d45-59c22d221663">
        <!-- the fileGrp for the optional external LOBS stored in segments (folders or folders packaged as files in a
package format) at the same parallel level as the SIARD file named .siard -->
        <!-- these files are all refered to in other mets files using the <mtpr> element, therefore only the mets files
ments are here - we need to store the check sum of the mets files in a file as well -->
        <mets:file ID="ID_Northwind_lobseg_0_mets" MIMETYPE="xml/mets"
OWNERID="ID.AVID.RA.18005.NORTHWIND" CHECKSUMTYPE="MD5"
CHECKSUM="c84642ff066720a5b8f4193a7f213782">
          <mets:FLocat LOCTYPE="URL" xlink:type="simple"
xlink:href="file:///Northwind_lobseg_0/mets.xml"/>
        </mets:file>
        <mets:file ID="ID_Northwind_lobseg_1_mets" MIMETYPE="xml/mets"
OWNERID="ID.AVID.RA.18005.NORTHWIND" CHECKSUMTYPE="MD5"
CHECKSUM="132d3d9f7e6a199c893f49b315708c3e">
          <mets:FLocat LOCTYPE="URL" xlink:type="simple"
xlink:href="file:///Northwind_lobseg_1/mets.xml"/>
        </mets:file>
        <mets:file ID="ID_Northwind_lobseg_2_mets" MIMETYPE="xml/mets"
OWNERID="ID.AVID.RA.18005.NORTHWIND" CHECKSUMTYPE="MD5"
CHECKSUM="03b7eaebbdab6584b9c23dd1895bde21">
          <mets:FLocat LOCTYPE="URL" xlink:type="simple"
xlink:href="file:///Northwind_lobseg_2/mets.xml"/>
        </mets:file>
      </mets:fileGrp>
    </mets:fileGrp>
  </mets:fileSec>
  <mets:structMap LABEL="SIARD structmap" TYPE="PHYSICAL">
    <!-- structmap for the SIARD files - the central SIARD file named x.siard and the possible LOBs as external files - only one <structmap> element is used for both, even though seperate is possible -->
    <mets:div LABEL="SIARD files">
      <mets:div LABEL="SIARD central file">
        <mets:fptr FILEID="ID_NORTHWIND.SIARD" CONTENTIDS="ID.AVID.RA.18005.Northwind.siard" ID="IDe98e416f-55a7-4237-8d45-59c22d221660"/>
        <!--the central SIARD file named .siard - packaged as a ZIP64 file -->
      </mets:div>
      <mets:div LABEL="child IPs">
        <mets:div LABEL="child IP">
          <mets:mptr xlink:href="ID_Northwind_lobseg_0_mets" xlink:title="Reference to a child IP" LOCTYPE="OTHER" OTHERLOCTYPE="someID" ID="IDd98e416f-55a7-4237-8d45-59c22d221669"/>
          <!-- the href value is a link to the mets file -->
        </mets:div>
        <mets:div LABEL="child IP">
          <mets:mptr xlink:href="ID_Northwind_lobseg_1_mets" xlink:title="Reference to a child IP" LOCTYPE="OTHER" OTHERLOCTYPE="someID" ID="ID70f8ec28-23f1-4364-9163-b3e99165b6e6"/>
        </mets:div>
        <mets:div LABEL="child IP">
          <mets:mptr xlink:href="ID_Northwind_lobseg_2_mets" xlink:title="Reference to a child IP" LOCTYPE="OTHER" OTHERLOCTYPE="someID" ID="ID3f0cc05c-f27d-499d-a6fd-63bdfed13cb0"/>
        </mets:div>
      </mets:div>
    </mets:div>
  </mets:structMap>
</mets:mets>
```

Child IP mets file

```
<?xml version="1.0" encoding="UTF-8"?>
<mets:mets xmlns:mets="http://www.loc.gov/METS/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.loc.gov/METS/ schemas/mets.xsd" OBJID="ID_Northwind_lobseg_0_mets" ID="ID.AVID.RA.18005.seg.0.mets.element" TYPE="SIARD2.0 INDEX" PROFILE="http://SIARD2.0.xml" LABEL="SIARD 2.0 child IP">
  <mets:metsHdr CREATEDATE="2015-06-13T14:40:00+01:00" RECORDSTATUS="NEW">
    <mets:agent ROLE="CREATOR" TYPE="OTHER" OTHERTYPE="SOFTWARE">
      <mets:name>SIARD 2.0 index</mets:name>
      <mets:note>Created by XX Software following E-ARK recommendation for external LOB handling in SIARD 2.0</mets:note>
    </mets:agent>
    <mets:metsDocumentID>IP child SIARD content</mets:metsDocumentID>
  </mets:metsHdr>
  <mets:fileSec>
    <!-- The following fileGrp is for the SIARD files. Other files in this IP must have their own fileGrp -->
    <mets:fileGrp ID="ID_SIARD_FILES">
      <!-- The following fileGrp is for a segment of the SIARD files. Only the parent IP knows the amount of segments-->
      <mets:fileGrp ID="seg_0">
        <!-- The following OWNERID=table no.row no.column. no -->
        <!-- The following ID is equal to the OWNERID but does not need to be -->
```

```
          <mets:file ID="s0.t2.c4.r1" MIMETYPE="binary" OWNERID="s0.t2.c4.r1" CHECKSUMTYPE="MD5"
CHECKSUM="74f24080fc9d234d3ac221b8e743c763" SIZE="10151">
            <mets:FLocat LOCTYPE="URL" xlink:type="simple"
xlink:href="file:///Northwind_lobseg_0/content/schema0/table2/lob4/record0.bin"/>
          </mets:file>
          <mets:file ID="s0.t2.c4.r2" MIMETYPE="binary" OWNERID="s0.t2.c4.r2" CHECKSUMTYPE="MD5"
CHECKSUM="22a0cbe8960b78ce48b07a285ce69e3c" SIZE="12107">
            <mets:FLocat LOCTYPE="URL" xlink:type="simple"
xlink:href="file:///Northwind_lobseg_0/content/schema0/table2/lob4/record1.bin"/>
          </mets:file>
          <mets:file ID="s0.t2.c4.r3" MIMETYPE="binary" OWNERID="s0.t2.c4.r3" CHECKSUMTYPE="MD5"
CHECKSUM="3e2f2028a9147c29bdcd36ed4e5f25b3" SIZE="12007">
            <mets:FLocat LOCTYPE="URL" xlink:type="simple"
xlink:href="file:///Northwind_lobseg_0/content/schema0/table2/lob4/record2.bin"/>
          </mets:file>
          <mets:file ID="s0.t2.c4.r4" MIMETYPE="binary" OWNERID="s0.t2.c4.r4" CHECKSUMTYPE="MD5"
CHECKSUM="12f588040e11cc2021ea37d46aa10c51" SIZE="9756">
            <mets:FLocat LOCTYPE="URL" xlink:type="simple"
xlink:href="file:///Northwind_lobseg_0/content/schema0/table2/lob4/record3.bin"/>
          </mets:file>
        </mets:fileGrp>
      </mets:fileGrp>
  </mets:fileSec>
  <!-- The following structMap is for the files in the segment - design proposal by Karin Bredenberg -->
  <mets:structMap LABEL="SIARD External LOB structmap" TYPE="LOBPOINTER">
    <mets:div LABEL="SIARD LOBs as external files">
      <mets:fptr CONTENTIDS="ID.AVID.RA.18005.seg_0"/>
    </mets:div>
  </mets:structMap>
  <!-- The following structMap is to inform that this child IP belongs to a certain parent ID using the value of the
xlink:href attribute which must match the value of the OBJID attribute for the <mets> element in the parent IP mets file
; design proposal by AIT -->
  <mets:structMap TYPE="logical" LABEL="parent IP">
    <mets:div LABEL="IP parent identifier">
      <mets:mptr xlink:href="ID.AVID.RA.18005" xlink:title="Referencing the parent IP of this child IP"
LOCTYPE="OTHER" OTHERLOCTYPE="Some ID scheme" ID="ID755d4d5f-5c5d-4751-9652-fcf839c7c6f2"/>
    </mets:div>
  </mets:structMap>
</mets:mets>
```