

Guideline for:

E-ARK Common Specification for Information Packages (CSIP)

E-ARK Common Specification for Submission Information Packages (SIP)

E-ARK Common Specification for Archival Information Packages (AIP)

E-ARK Common Specification for Dissemination Information Packages (DIP)

E-ARK Content Information Type Specification for Archival Information (CITS Archival Information)

E-ARK Content Information Type Specification for Preservation Metadata (CITS Preservation Metadata)

A proper front page will be created for the publication occurring after implementation of review comments.

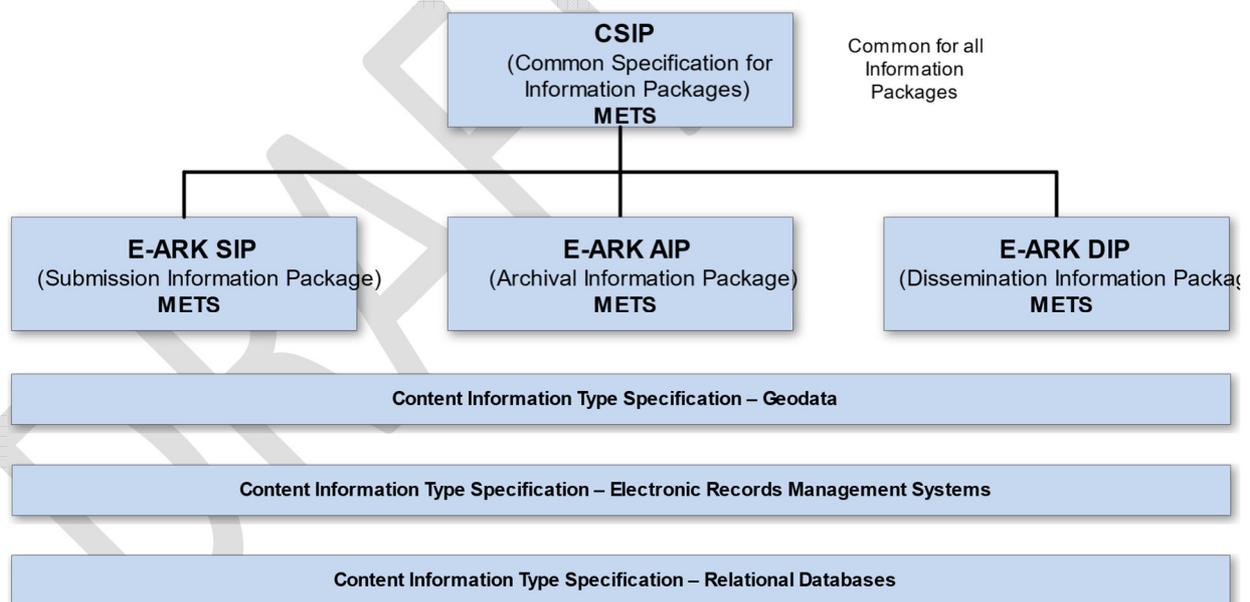
1 Preface

The correct preface will be inserted for the publication occurring after implementation of review comments.

1.1 Aim of the specification

This E-ARK specification is part of a family of specifications that provide a common set of requirements for packaging digital information. These specifications are based on common, international standards for transmitting, describing and preserving digital data. They have been produced to help data creators, software developers and digital archives tackle the challenge of short-, medium- and long-term data management and reuse in a sustainable, authentic, cost-efficient, manageable and interoperable way.

The foundation for these specifications is the Reference Model for an Open Archival Information System (OAIS) which has Information Packages at its core. Familiarity with the core functional entities of OAIS is a prerequisite for understanding the specifications. A visualisation of the current specification network can be seen here:



The E-ARK specification dependency hierarchy

Specification	Aim and Goals
Common Specification for	This document introduces the concept of a Common Specification for Information Packages (CSIP). Its three main purposes are to:

Specification	Aim and Goals
Information Packages	<ul style="list-style-type: none"> • Establish a common understanding of the requirements which need to be met in order to achieve interoperability of Information Packages. • Establish a common base for the development of more specific Information Package definitions and tools within the digital preservation community. • Propose the details of an XML-based implementation of the requirements using, to the largest possible extent, standards which are widely used in international digital preservation. <p>Ultimately the goal of the Common Specification is to reach a level of interoperability between all Information Packages so that tools implementing the Common Specification can be adopted by institutions without the need for further modifications or adaptations.</p>
E-ARK SIP	<p>The main aims of this specification are to:</p> <ul style="list-style-type: none"> • Define a general structure for a Submission Information Package format suitable for a wide variety of archival scenarios, e.g. document and image collections, databases or geographical data. • Enhance interoperability between Producers and Archives. • Recommend best practices regarding metadata, content and structure of Submission Information Packages.
E-ARK AIP	<p>The main aims of this specification are to:</p> <ul style="list-style-type: none"> • Define a generic structure of the AIP format suitable for a wide variety of data types, such as document and image collections, archival records, databases or geographical data. • Recommend a set of metadata related to the structural and the preservation aspects of the AIP as implemented by the reference implementation eArchiving ToolBox (formerly earkweb). • Ensure the format is suitable to store large quantities of data.
E-ARK DIP	<p>The main aims of this specification are to:</p> <ul style="list-style-type: none"> • Define a generic structure of the DIP format suitable for a wide variety of archival records, such as document and image collections, databases or geographical data. • Recommend a set of metadata related to the structural and access aspects of the DIP.
Content Information Type	<p>The main aim and goal of a Content Information Type Specification is to:</p>

Specification	Aim and Goals
Specifications	<ul style="list-style-type: none"> • Define, in technical terms, how data and metadata must be formatted and placed within a CSIP Information Package to achieve interoperability in exchanging specific Content Information. <p>The number of possible Content Information Type Specifications is unlimited.</p>

1.2 Organisational support

This specification is maintained by the Digital Information LifeCycle Interoperability Standards Board (DILCIS Board). The DILCIS Board (<http://dilcis.eu/>) was created to enhance and maintain the draft specifications developed in the European Archival Records and Knowledge Preservation Project (E-ARK project) which concluded in January 2017 (<http://eark-project.com/>). The Board consists of eight members, but there is no restriction on the number of participants in the work. All Board documents and specifications are stored in GitHub (<https://github.com/DILCISBoard>) while published versions are made available on the Board webpage. Since 2018 the DILCIS Board has been responsible for the core specifications in the Connecting Europe Facility eArchiving Building Block (<https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eArchiving>).

1.3 Authors

A full list of contributors to this specification, as well as the revision history can be found in Appendix 1.

TABLE OF CONTENT

1	Preface	2
1.1	Aim of the specification.....	2
1.2	Organisational support.....	4
1.3	Authors.....	4
1	Context	9
1.1	Purpose.....	9
1.2	Scope	9
2	The document setup	9
2.1	Explanation of the preface	9
3	The Digital Information LifeCycle Interoperability Standards Board (DILCIS Board)	9
4	The CEF Building Block	10
4.1	The eArchiving Building Block	10
5	Standard/Standards used	10
5.1	Open Archival Information Systems Reference Model (OAIS).....	10
5.1.1	OAIS terms	11
5.1.2	How the terms are used in the specifications	13
5.2	Extensible Markup Language (XML).....	15
5.2.1	Key concepts in XML.....	15
5.2.2	XML schema.....	16
5.2.3	Learning XML	16
5.3	Metadata Encoding and Transmission Standard (METS)	16
5.3.1	Overview of METS.....	16
5.4	Schematron	20
5.5	PREservation Metadata Implementation Strategies (PREMIS)	21
5.5.1	PREMIS data model	21
5.5.2	PREMIS fundamentals	23
5.5.3	Types of objects	23
5.5.4	Using PREMIS with METS.....	24
5.5.5	Vocabularies in PREMIS	24
5.5.6	A note on identifiers	24
5.5.7	PREMIS in the IP:s.....	24
5.6	Description standards	25
5.6.1	Implementing description standards.....	25
5.6.2	Encoded Archival Description (EAD).....	25
5.6.3	Encoded Archival Creators- Corporate Bodies, Persons and Families (EAC-CPF).....	25

5.6.4	Encoded Archival Guide (EAG).....	26
5.6.5	Records in Context (RIC)	26
5.7	[More standards needing a description can be added]	26
6	Glossary.....	26
7	Metadata.....	28
7.1	Explanation of tables and values used in the specification.....	28
7.2	Specification tables	28
7.3	Cardinality values	29
7.4	Level of requirement values.....	29
7.5	Identifiers in the IP specifications	29
8	In depth information regarding different concepts and terms	30
8.1	CS and CITS	30
8.2	In the CSIP there Explanation of redundancy and incompatible requirements	30
8.2.1	Example 1	30
8.2.2	Example 2	30
8.2.3	Example 3	31
8.2.4	Example 4	31
8.3	Explanation of principles in CSIP	32
8.3.1	Explanation of Principle 1.1.....	32
8.3.2	Explanation of Principle 1.2.....	32
8.3.3	Explanation of Principle 1.3.....	32
8.3.4	Explanation of Principle 1.4.....	32
8.3.5	Explanation of Principle 1.5.....	32
8.3.6	Explanation of Principle 1.6.....	33
8.3.7	Explanation of Principle 1.7.....	33
8.3.8	Explanation of Principle 2.1.....	33
8.3.9	Explanation of Principle 2.2.....	33
8.3.10	Explanation of Principle 2.3.....	33
8.3.11	Explanation of Principle 2.4.....	34
8.3.12	Explanation of Principle 2.5.....	34
8.3.13	Explanation of Principle 3.1.....	34
8.3.14	Explanation of Principle 3.2.....	34
8.3.15	Explanation of Principle 3.3.....	35
8.3.16	Explanation of Principle 3.4.....	35
8.3.17	Explanation of Principle 3.5.....	35
8.3.18	Explanation of Principle 3.6.....	35

8.3.19	Explanation of Principle 4.1	35
8.3.20	Explanation of Principle 4.2	36
8.3.21	Explanation of Principle 4.3	36
8.4	Folder structure requirements	36
8.4.1	Explanation of CSIPSTR1	36
8.4.2	Explanation of CSIPSTR2	36
8.4.3	Explanation of CSIPSTR3	36
8.4.4	Explanation of CSIPSTR4	36
8.4.5	Explanation of CSIPSTR5	37
8.4.6	Explanation of CSIPSTR6	37
8.4.7	Explanation of CSIPSTR7	37
8.4.8	Explanation of CSIPSTR8	37
8.4.9	Explanation of CSIPSTR9	37
8.4.10	Explanation of CSIPSTR10	37
8.4.11	Explanation of CSIPSTR11	38
8.4.12	Explanation of CSIPSTR12	38
8.4.13	Explanation of CSIPSTR13	38
8.4.14	Explanation of CSIPSTR14	38
8.4.15	Explanation of CSIPSTR15	38
8.4.16	Explanation of CSIPSTR16	38
8.5	Explanation of the concept of representations	39
8.5.1	Explanation of levels of packages and nesting of representations	39
8.6	Signatures	39
8.7	Vocabularies	39
8.8	Referencing	40
9	Validation	40
10	Own adoptions of the specifications	40
10.1	Adapting CSIP/SIP/AIP/DIP specifications	40
10.2	Adapting any CITS specifications	40
10.3	Adapting PREMIS	40
11	Example following CSIP	41
11.1	Part 1 of example, use of the mets element	41
11.2	Part 2 of example, use of the metsHdr element	41
11.3	Part n+1 of example	42
11.4	Full example	42
12	Postface	43

LIST OF TABLES

Table 1: Glossary	26
Table 2: Requirement tables headings	28
Table 3: Explanation of the parts of the requirement table	28
Table 4: Cardinality	29
Table 5: Level of requirement	29

LIST OF FIGURES

Figure 1: OAIS reference model drawn by digitalbevaring.dk	11
Figure 2: OAIS reference model as drawn in the specifications	12
Figure 3: PREMIS data model (with permission from the PREMIS Editorial Committee)	22

DRAFT V.0.2

1 Context

1.1 Purpose

The purpose of this guideline is to further explain and describe the Common Specification for Information Packages and the extending specifications for Submission Information Packages, Archival Information Packages and Dissemination Information Packages. To make the guideline cover all parts of an information package, the content information type specifications for archival information and preservation metadata is also covered.

The goal is to make the specifications as easy as possible to use with explanations and deeper descriptions being in the guideline.

The guideline is an evolving document, and more concepts and standards will be explained following the needs of the users of the different specifications. There will also be accompanying guidelines which will describe the specific content information type specifications published later.

1.2 Scope

This guideline will provide further information and insights to the information packages which is not covered in the four specifications with explanations for archival information and preservation metadata.

2 The document setup

This guideline is using textual parts to describe the content and concepts for the different specifications.

2.1 Explanation of the preface

The preface describes on a general level the different packages and the different content information types available to be used in information transfer, whether it be between systems or to an archive. At the same time, the preface is the common introduction to all the specifications and text maintained by the DILCIS Board, and thus it is repeated in all the specification documents that are published.

3 The Digital Information LifeCycle Interoperability Standards Board (DILCIS Board)

The Digital Information LifeCycle Interoperability Standards Board (DILCIS Board) <https://dilcis.eu/> is an international group of experts committed to maintain and sustain a set of interoperability specifications which allow for the transfer, long-term preservation, and reuse of digital information regardless of the origin or type of the information.

More specifically, the DILCIS Board maintains specifications initially developed within the E-ARK Project (02.2014 – 01.2017), and which are now the core of the eArchiving Building Block.

4 The CEF Building Blocks

The CEF Building Blocks

<https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/CEF+Digital+Home> offer basic capabilities that can be used in any European project to facilitate the delivery of digital public services across borders. The foundation for the Building Blocks are interoperability agreements between European Union member states, and their aim is to ensure interoperability between IT systems so that citizens, businesses and administrations can benefit from seamless digital public services wherever they may be in Europe.

To do so, the European Commission provides a Core Service Platform for each Building Block, which consists of three layers:

- At the core of each Building Block, a layer of technical specifications and standards that have to be complied with;
- To facilitate the implementation of the technical specifications and standards, a layer of sample software that complies with them and is meant for reuse (for certain Building Blocks only);
- To facilitate the adoption of the technical specifications and standards, a layer of services (e.g. conformance testing, help desks, onboarding services, etc.) meant for use (which varies depending on the Building Block).

The Building Blocks can be combined and used in projects in any domain or sector at European, national or local level.

4.1 The eArchiving Building Block

The aim of eArchiving

(<https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eArchiving>) is to provide the core specifications, software, training and knowledge to help data creators, software developers and digital archives tackle the challenge of short, medium and long-term data management and reuse in a sustainable, authentic, cost-efficient, manageable and interoperable way.

5 Standard/Standards used

The specifications for the information packages are built upon several standards which all are described in this section.

5.1 Open Archival Information Systems Reference Model (OAIS)

The basis for describing an electronic archive is the Reference Model for an Open Archival Information System (OAIS). A reference model created by the Consultative Committee for Space Data Systems (CCSDS) and in 2012 it became an ISO standard. The reference model document is available as recommendation CCSDS 650.0-B-2 of the Consultative Committee for Space Data Systems found <https://public.ccsds.org/pubs/650x0m2.pdf> and this text is identical to ISO 14721:2012 found here <https://www.iso.org/standard/57284.html> for purchase.

It was developed for use in facilitating a broad, discipline independent, consensus on the requirements for an archive or repository to provide long-term, preservation of digital information. It was also intended to support the development of additional digital preservation standards.

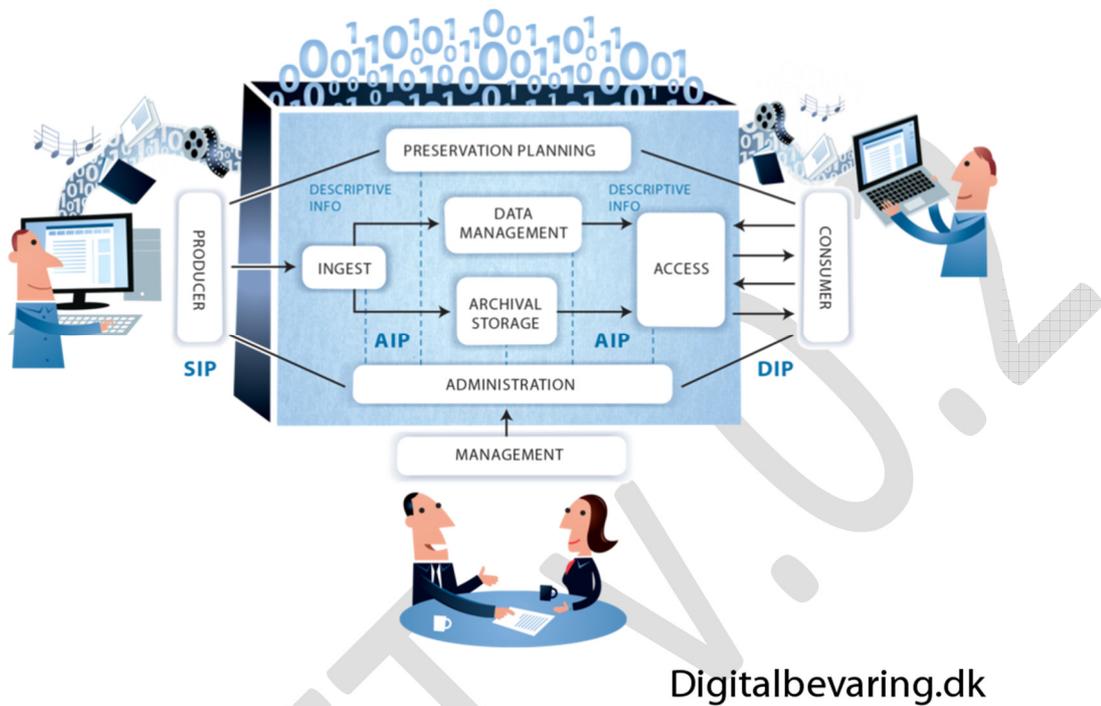


Figure 1: OAIS reference model drawn by digitalbevaring.dk

The reference model describes an OAIS where the archive is consisting of an organisation of people and systems that has accepted the responsibility to preserve information and make it available to a Designated Community. The standard defines a set of responsibilities that an OAIS archive must fulfil, and this allows an OAIS archive to be distinguished from other uses of the term archive.

5.1.1 OAIS terms

The OAIS reference model defines several terms, and some of them will be explained in the following subsections. We start with the terms found in the OAIS Reference Model and the definition given there and after that explains them in our specifications.

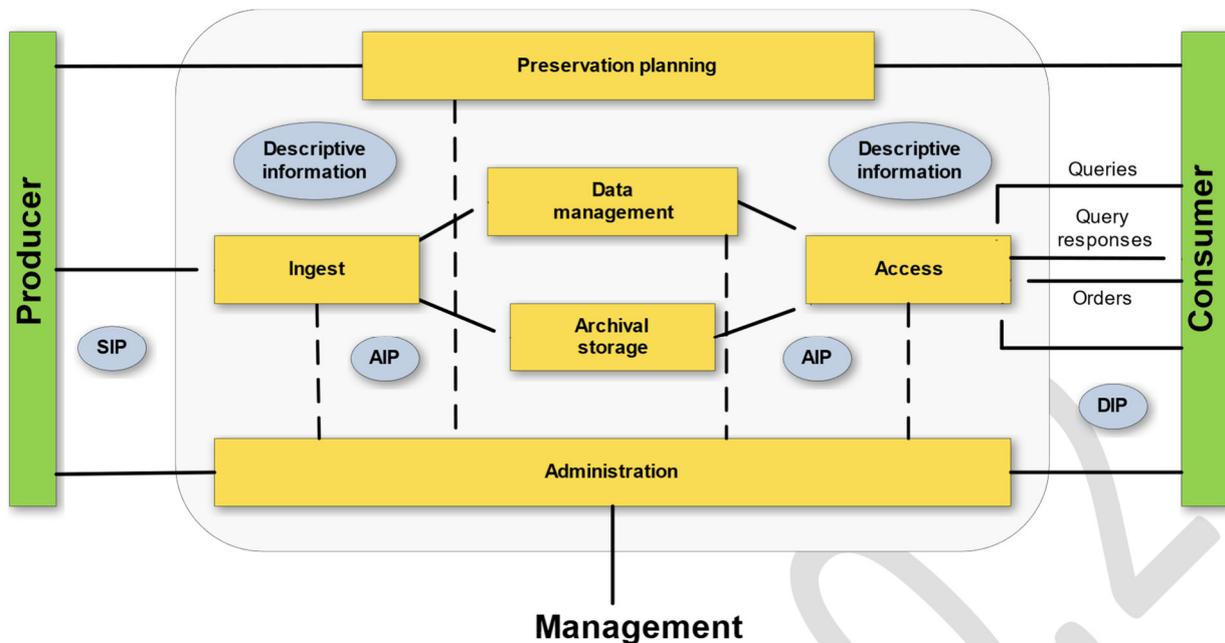


Figure 2: OAIS reference model as drawn in the specifications

5.1.1.1 Information Package

Definition from OAIS:

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.

5.1.1.2 Submission Information Package (SIP)

Definition from OAIS:

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.

5.1.1.3 Archival Information Collection (AIC)

Definition from OAIS:

An Archival Information Package whose Content Information is an aggregation of other Archival Information Packages.

5.1.1.4 Archival Information Package (AIP)

Definition from OAIS:

An Information Package, consisting of the Content Information and the associated Preservation Description Information (PDI), which is preserved within an OAIS.

5.1.1.5 Archival Information Unit (AIU)

Definition from OAIS:

An Archival Information Package where the Archive chooses not to break down the Content Information into other Archival Information Packages. An AIU can consist of multiple digital objects (e.g., multiple files).

5.1.1.6 Dissemination Information Package (DIP)

Definition from OAIS:

An Information Package, derived from one or more AIPs, and sent by Archives to the Consumer in response to a request to the OAIS

5.1.1.7 Submission Agreement

Definition from OAIS:

The agreement reached between an OAIS 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.

5.1.1.8 Representation Information

Definition from OAIS:

The information that maps a Data Object into more meaningful concepts. An example of Representation Information for a bit sequence which is a FITS file might consist of the FITS standard which defines the format plus a dictionary which defines the meaning in the file of keywords which are not part of the standard. Another example is JPEG software which is used to render a JPEG file; rendering the JPEG file as bits is not very meaningful to humans but the software, which embodies an understanding of the JPEG standard, maps the bits into pixels which can then be rendered as an image for human viewing.

5.1.2 How the terms are used in the specifications

Within the specifications, we are using several terms, especially the ones relating to the packages defined in the OAIS. We do not put another meaning to them; they are used as they are defined, but in some cases, there are extra information needed to understand the use of the terms in the specifications.

5.1.2.1 Common Specification for Information Packages (CSIP)

Within the work with the information packages, the decision was made to move all common requirements to one specification. The decision is based upon instead of repeating requirements needed in all the different specifications or trying to incorporate different needs into one specification splitting it up to make the specifications easier to use and implement. It is possible to extend the specifications with own requirements valid only in the OAIS implementation hosted by the user.

5.1.2.2 SIP specification

This section will be extended for the publication occurring after implementation of review comments.

This is the specification for the SIP in the OAIS. The SIP specification extends CSIP with the metadata needed in the transfer moment incorporation all the needed information. The SIP is the package existing in the ingest moment. Another term used is pre-ingest, and that is the package before it becomes a SIP. It is in the pre-ingest step all different information that will be placed in the package and transformations needed of the information to become the digital object is performed.

5.1.2.3 AIP specification

This section will be extended for the publication occurring after implementation of review comments.

The AIP specification doesn't define a special format for the AIP stored in the OAIS instead it gives the description of how an OAIS compliant electronic archive is working especially considering the maintenance needed to be performed on the digital object stored using CSIP as the basis and how to move the AIP's to a new storage.

5.1.2.4 DIP specification

This section will be extended for the publication occurring after implementation of review comments.

The DIP specification extends CSIP with information needed for the consumer of the information stored in the OAIS. When a request for information to be disseminated is made one part is the information the other part is to provide the information to make the information possible to view. Thus, the DIP specification focus on giving the information about software that can be used for viewing the information to be disseminated.

5.1.2.5 Submission agreement

This section will be extended for the publication occurring after implementation of review comments.

All transfers need to be following a submission agreement to make sure there are establishes specific details about how these interactions should take place, e.g. the type of information expected to be exchanged, the metadata the Producer is expected to deliver, the logistics of the actual transfer, statements regarding access restrictions on the submitted material, etc. There are already submission agreements in place in a lot of the organisations taking part in digital transfers and thus we do not define a format for the agreement instead in the SIP specification a suggestion and recommendation of what needs to be present is given in appendix A.

5.1.2.6 Representation information

In the specifications we have chosen to use the term "Documentation" for information needed to understand the digital objects. The term includes the definition of representation information from OAIS and other kind of documentation needed. As examples of other kinds of documentation that can be needed are manuals for the system from where the information becoming the digital object has been exported. The manuals can provide understanding for the digital object with showing how parts of it were used when it was in use in an organisation.

5.2 Extensible Markup Language (XML)

[Source: <https://en.wikipedia.org/wiki/XML> ,
[https://en.wikipedia.org/wiki/XML_Schema_\(W3C\)](https://en.wikipedia.org/wiki/XML_Schema_(W3C))]

Currently, the used format for the transfer and storage of metadata used in the specifications is based upon the use of XML. In the future, there will be other formats used, and the specifications will be adapted in revisions to use new formats like rdf and json or others that will be the preferred formats.

Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). The specification is maintained by the organisation named w3 who are responsible for the different XML languages as well as other formats in the same family. The specification itself is found at <https://www.w3.org/XML/Core/> .

5.2.1 Key concepts in XML

There are several parts of the XML document that needs to be understood to fully understand and read the document as a human being. As a human just opening the XML document is possible and it is possible to read as text since the document is only characters. Not all Unicode characters is possible to use but not all depending on the encoding used. The machines reading the document with the help of an XML processor that analyses the text and passes it as structured information to an application. This means that the machine can be given instructions to use the information in many ways where one way is simplification for reading by humans.

5.2.1.1 XML declaration

The XML documents usually begins with an XML declaration that describes which encoding (character set) that are used in the document. An example is `<?xml version="1.0" encoding="UTF-8"?>`.

5.2.1.2 Markup and content

The characters making up an XML document are divided into markup and content, which may be distinguished by the application of simple syntactic rules. Generally, strings that constitute markup begins with the character `<` and end with a `>`. Strings of characters that are not markup are content.

5.2.1.3 Markup

The markup is also known as a tag. That means the tag is the same as markup, a construct that begins with `<` and ends with `>`. Tags come in three flavours:

- start-tag, such as `<section>`;
- end-tag, such as `</section>`;
- empty-element tag, such as `<line-break />`.

5.2.1.4 Element

An element is a logical document component that either begins with a start-tag and ends with a matching end-tag or consists only of an empty-element tag. The characters between the start-tag and end-tag, if any, are the element's content, and may contain

markup, including other elements, which are called child elements. An example is `<greeting>Hello, world!</greeting>`. Another is `<line-break />`.

5.2.1.5 Attribute

An attribute is a markup construct consisting of a name–value pair that exists within a start-tag or empty-element tag. An example is ``, where the names of the attributes are "src" and "alt", and their values are "madonna.jpg" and "Madonna" respectively. Another example is `<step number="3">Connect A to B.</step>`, where the name of the attribute is "number" and its value is "3". An XML attribute can only have a single value, and each attribute can appear at most once on each element. In the common situation where a list of multiple values is desired, this must be done by encoding the list into a well-formed XML attribute with some format beyond what XML defines itself. Usually, this is either a comma or semi-colon delimited list or, if the individual values are known not to contain spaces, a space-delimited list can be used. `<div class="inner greeting-box">Welcome!</div>`, where the attribute "class" has both the value "inner greeting-box" and also indicates the two CSS class names "inner" and "greeting-box".

5.2.2 XML schema

For creating the rules of availability and what the XML document can contain in form of elements and attributes XML schemas are created. These XML schemas is to be the rule book for creating a specific type of XML document describing a specific type of information. The specification for XML-schema is maintained by W3C and found at <https://www.w3.org/XML/Schema> .

For the specifications, the XML schemas are created and maintained by the groups responsible for the standard which means we do not usually create new XML schemas.

5.2.3 Learning XML

There are several different sources accessible online where it is possible to learn XML.

5.3 Metadata Encoding and Transmission Standard (METS)

For describing the different packages, the specifications utilise the de-facto standard METS. The standard consists of a primer describing all elements and attributes and one XML-schema making it possible to create XML documents following METS. The standard can be found here <http://www.loc.gov/standards/mets/mets-home.html> .

For each use case of METS the request from the standard is to create a METS profile to describe the use of METS. The specifications are therefore described with the help of three METS profiles, one each for the CSIP, SIP and DIP.

In METS it is possible to reference metadata and digital objects or include them in the document. The specification strongly advises to always reference both metadata and digital objects. The decision for this advice comes from the need of the metadata and digital objects to be placed in the folder structure and thus being able to understand without the METS document.

5.3.1 Overview of METS

[Source: <http://www.loc.gov/standards/mets/METSOverview.v2.html>]

The METS document gathers all the information needed to understand the digital object being transferred as well as giving which the digital objects being transferred is. The METS document consists of seven major sections described here:

1. **METS Header** - The METS Header contains metadata describing the METS document itself, including such information as creator, editor, etc.
2. **Descriptive Metadata** - The descriptive metadata section may point to descriptive metadata external to the METS document (e.g., a MARC record in an OPAC or an EAD finding aid maintained on a WWW server), or contain internally embedded descriptive metadata, or both. Multiple instances of both external and internal descriptive metadata may be included in the descriptive metadata section.
3. **Administrative Metadata** - The administrative metadata section provides information regarding how the files were created and stored, intellectual property rights, metadata regarding the original source object from which the digital library object derives, and information regarding the provenance of the files comprising the digital library object (i.e., master/derivative file relationships, and migration/transformation information). As with descriptive metadata, administrative metadata may be either external to the METS document, or encoded internally.
4. **File Section** - The file section lists all files containing content which comprise the electronic versions of the digital object. <file> elements may be grouped within <fileGrp> elements, to provide for subdividing the files by object version.
5. **Structural Map** - The structural map is the heart of a METS document. It outlines a hierarchical structure for the digital object, and links the elements of that structure to content files and metadata that pertain to each element.
6. **Structural Links** - The Structural Links section of METS allows METS creators to record the existence of hyperlinks between nodes in the hierarchy outlined in the Structural Map. This is of particular value in using METS to archive Websites.
7. **Behavior** - A behaviour section can be used to associate executable behaviours with content in the METS object. Each behaviour within a behaviour section has an interface definition element that represents an abstract definition of the set of behaviours represented by a particular behaviour section. Each behaviour also has a mechanism element which identifies a module of executable code that implements and runs the behaviours defined abstractly by the interface definition. (The name is in American-English.)

The following sections describes the parts used in the specifications a little further.

5.3.1.1 METS Header

The METS Header element is used for creating minimal descriptive metadata about the METS object itself within the METS document. This metadata includes the date of creation for the METS document, the date of its last modification, and a status for the METS document. There is also possible to record names and information regarding one or more agents who have played some role with respect to the METS document, specify the role they have played, and add a small note regarding their activity.

5.3.1.2 Descriptive Metadata

The descriptive metadata section of a METS document consists of one or more <dmdSec> (Descriptive Metadata Section) elements. For example, it is possible to reference the finding aid for a particular digital library object. It is possible to provide the type of descriptive metadata in the attribute named MDTYPE. These types have been tested and deemed valid standards to use together with METS MARC, MODS, EAD, VRA (VRA Core), DC (Dublin Core), NISOIMG (NISO Technical Metadata for Digital Still Images), LC-AV (Library of Congress Audiovisual Metadata), TEIHDR (TEI Header), DDI (Data Documentation Initiative), FGDC (Federal Geographic Data Committee Metadata Standard [FGDC-STD-001-1998]). It is also possible to use other descriptive metadata.

Note that all <dmdSec> elements must possess an ID attribute. This attribute provides a unique, internal name for each <dmdSec> element which can be used in the structural map to link a particular division of the document hierarchy to a particular <dmdSec> element. This allows specific sections of descriptive metadata to be linked to specific parts of the digital object.

5.3.1.3 Administrative Metadata

The administrative metadata section of a METS document consists of one or more <amdSec> (Administrative Metadata Section) elements. For example, it's possible to express all the administrative metadata pertaining to the files comprising a digital library object, as well as that pertaining to the original source material used to create the object. The <amdSec> element in its turn contains four main forms of administrative metadata provided for in a METS document:

1. Technical Metadata <techMD> (information regarding files' creation, format, and use characteristics),
2. Intellectual Property Rights Metadata <rightsMD>, (copyright and license information),
3. Source Metadata <sourceMD>, (descriptive and administrative metadata regarding the analog source from which a digital library object derives), and
4. Digital Provenance Metadata <digiprovMD>, (information regarding source/destination relationships between files, including master/derivative relationships between files and information regarding migrations/transformations employed on files between original digitisation of an artifact and its current incarnation as a digital library object).

Note that <amdSec>, <techMD>, <rightsMD>, <sourceMD> and <digiprovMD> must carry an ID attribute so that other elements within the METS document (such as divisions within the structural map or <file> elements) may be linked to the <amdSec> subelements which describe them.

5.3.1.4 File Section

The file section (<fileSec>) contains one or more <fileGrp> elements used to group together related files. A <fileGrp> lists all of the files which comprise a single electronic version of the digital library object. For example, there might be separate <fileGrp> elements for the thumbnails, the master archival images, the pdf versions, the TEI encoded text versions, etc.

The <file> element describes the digital objects with for example a checksum, the mime type, and the name of the file. There is also needed to note that all the <file> elements have a unique ID attribute. This attribute provides a unique, internal name for this file which can be referenced by other portions of the document.

5.3.1.5 Structural Map

The structural map section of a METS document defines a hierarchical structure which can be presented to users of the digital library object to allow them to navigate through it. The <structMap> element encodes this hierarchy as a nested series of <div> elements. Each <div> carries attribute information specifying what kind of division it is, and may contain multiple METS pointer (<mptr>) and file pointer (<fptr>) elements to identify content corresponding with that <div>. METS pointers specify separate METS documents as containing the relevant file information for the <div> containing them. This can be useful when encoding large collections of material (e.g., an entire journal run) to keep the size of each METS file in the set relatively small. File pointers specify files (or in some cases either groups of files or specific locations within a file) within the current METS document's <fileSec> section that correspond to the portion in the hierarchy represented by the current <div>.

5.3.1.6 Possibility of own extensions in METS

In METS in several places it is possible to add the use of own attributes defined in an own XML-schema. This is a function to make it possible to add information that is not hosted in METS but is needed in the local implementation and use of METS.

The extension with own attributes is possible in the following METS elements: mets, metsHdr, note in agent, dmdSec, amdSec, techMD, rightsMD, sourceMD, digiprovMD, fileSec, fileGrp, file, structMap, fptr, structLink and behaviorSec

5.3.1.7 Code example from the METS Primer

A full example of a METS document is the following one describing 3 images which in their turn are described with a MODS document. Two of the images is used as service copies and one is the saved master image. The full document is available here, <http://memory.loc.gov/diglib/ihis/loc.afc.afc9999005.1153/mets.xml>

```
<?xml version="1.0" encoding="UTF-8"?><mets:mets
xmlns:mets="http://www.loc.gov/METS/"
xmlns:lc="http://www.loc.gov/mets/profiles"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:rights="http://www.loc.gov/rights/"
xmlns:mods="http://www.loc.gov/mods/v3"
xmlns:bib="http://www.loc.gov/mets/profiles/modsBibCard"
OBJID="loc.afc.afc9999005.1153" PROFILE="lc:modsBibCard">
<mets:metsHdr LASTMODDATE="2016-08-09T12:12:51.320141-04:00"/>
<mets:dmdSec ID="dmd1">
<mets:mdWrap MDTYPE="MODS">
<mets:xmlData>
<mods:mods ID="mods1" version="3.4">
.....
</mods:mods>
</mets:xmlData>
</mets:mdWrap>
```

```

</mets:dmdSec>
<mets:fileSec>
  <mets:fileGrp USE="MASTER">
    <mets:file MIMETYPE="image/tiff" GROUPID="G1" ID="f0178m">
      <mets:FLocat LOCTYPE="URL"
xlink:href="http://lcweb4.loc.gov/natlib/ihas/warehouse/afc9999005/AFS\_300\_A-734\_B/0178.tif"/>
    </mets:file>
  </mets:fileGrp>
  <mets:fileGrp USE="SERVICE">
    <mets:file MIMETYPE="image/jpeg" GROUPID="G1" ID="f0178s">
      <mets:FLocat LOCTYPE="URL"
xlink:href="http://lcweb4.loc.gov/natlib/ihas/service/afc9999005/AFS\_300\_A-734\_B/0178v.jpg"/>
    </mets:file>
    <mets:file MIMETYPE="image/tiff" GROUPID="G1" ID="f0178z">
      <mets:FLocat LOCTYPE="URL"
xlink:href="/media/loc.afc.afc9999005.1153/0178.tif"/>
    </mets:file>
  </mets:fileGrp>
</mets:fileSec>
<mets:structMap>
  <mets:div DMDID="dmd1" TYPE="bib:modsBibCard">
    <mets:div TYPE="bib:card">
      <mets:div TYPE="lc:image">
        <mets:fptr FILEID="f0178m"/>
        <mets:fptr FILEID="f0178s"/>
        <mets:fptr FILEID="f0178z"/>
      </mets:div>
    </mets:div>
  </mets:div>
</mets:structMap>
</mets:mets>

```

5.3.1.8 Implementing the METS profiles and the IP:s

In most cases the IP specifications is not implemented in each and every system to be able to create an SIP package but it can be made instead they are part of a tool that creates packages after an export of digital objects have been made in the originating system. The tool needs to be able to sort the files into its placement according to the IP as well as creating needed checksums and information. It is also possible to create a package by hand, but it's not recommended.

5.4 Schematron

Schematron <http://schematron.com/> is a ISO standard describing a rule-based validation language for making assertions about the presence or absence of patterns in XML trees. It is a structural schema language expressed in XML using a small number of elements and XPath.

Schematron can express constraints in ways that other XML schema languages like XML Schema and DTD cannot. For example, it can require that the content of an element be controlled by one of its siblings. Or it can request or require that the root element, regardless of what element that is, must have specific attributes. Schematron can also specify required relationships between multiple XML files.

Constraints and content rules may be associated with "plain-English" validation error messages, allowing translation of numeric Schematron error codes into meaningful user error messages.

The current ISO recommendation is Information technology, Document Schema Definition Languages (DSDL), Part 3: Rule-based validation, Schematron (ISO/IEC 19757-3:2016).

A useful introduction to Schematron has been created by Mulberry Tech and its available on-line, <https://www.mulberrytech.com/papers/schematron-Philly.pdf> .

5.5 PREservation Metadata Implementation Strategies (PREMIS)

[Source: <http://www.loc.gov/standards/premis/>]

PREMIS (Preservation Metadata: Implementation Strategies) and its PREMIS Data Dictionary <http://www.loc.gov/standards/premis/> is a comprehensive, practical resource for implementing preservation metadata in digital preservation systems. The Data Dictionary defines preservation metadata that:

- Supports the viability, renderability, understandability, authenticity and identity of digital objects in a preservation context;
- Represents the information most preservation repositories need to know to preserve digital materials over the long term;
- Emphasises “implementable metadata”: rigorously defined, supported by guidelines for creation, management, and use, and oriented toward automated workflows; and,
- Embodies technical neutrality: no assumptions are made about preservation technologies, strategies, metadata storage and management, etc.

The current version of the PREMIS data dictionary is version 3 found at <http://www.loc.gov/standards/premis/v3/index.html> .

5.5.1 PREMIS data model

The PREMIS Data Dictionary defines semantic units. Each semantic unit defined in the Data Dictionary is mapped to an entity that is organised within a simple data model. A semantic unit can, therefore, be understood as a property of an entity. The model defines four entities important regarding digital preservation activities: Objects, Events, Agents and Rights. Figure 4 provides a graphical illustration of the PREMIS Data Model.

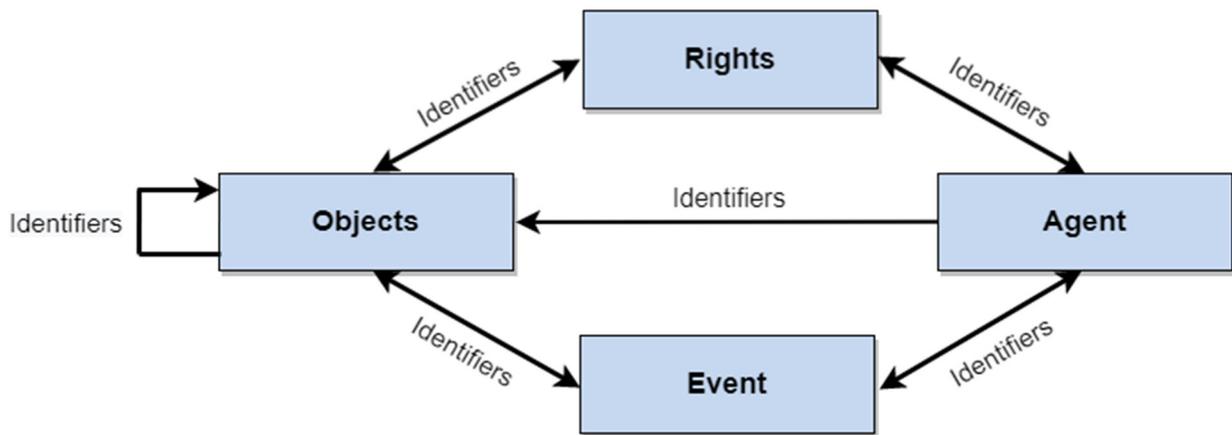


Figure 3: PREMIS data model (with permission from the PREMIS Editorial Committee)

In Figure 4, entities are represented by boxes and relationships between entities are represented by arrows. When arrows are bi-directional, then each entity type contains a semantic unit allowing it to link to the other. So, for example, the Rights entity includes a semantic unit recording information about the relationship with an Agent, and the Agent entity includes a semantic unit recording information about associated Rights.

The entities in the PREMIS data model are defined as follows:

- **Object (or Digital Object):** A discrete unit of information subject to digital preservation. Version 3 introduces the notion that this can be an environment used as part of the preservation process.
- **Environment:** Technology (software or hardware) supporting a Digital Object in some way (e.g. rendering or execution). Environments can be described as Intellectual Entities and captured and preserved in the preservation repository as Representations, Files and/or Bitstreams.
- **Event:** An action that involves or affects at least one Object or Agent associated with or known by the preservation repository.
- **Agent:** A person, organisation, or software program/system associated with Events in the life of an Object, or with Rights attached to an Object. It can also be related to an environment Object that acts as an Agent.
- **Rights Statement:** Assertion of one or more Rights or permissions pertaining to an Intellectual Object and/or Agent.

To thoroughly understand PREMIS, it is recommended that users study the data dictionary and participate in the events led by the PREMIS Editorial Committee. More information can be found on the PREMIS website.

5.5.2 PREMIS fundamentals

The standard describes all its elements in the Data Dictionary available online at <http://www.loc.gov/standards/premis/v3/index.html>.

Elements are called semantic components, and each semantic component has its own entry in the Data Dictionary. A semantic unit that has semantic components does not have any value of its own. Only semantic units at the lowest level have values. Thus:

- **Semantic Component**, which is a grouping of semantic units and,
- **Semantic Units**, which are the lowest level of description.

It is important to remember that PREMIS is implementation-independent, but the XML-schema has been chosen as the implementation form in our specifications. Thus, it is possible to implement these structures in, for example, a database and export it as the XML-document when a transfer of the information with its preservation metadata is performed.

5.5.3 Types of objects

PREMIS defines four different kinds of Objects and requires implementers to make a distinction between them. These are Bitstreams, Files, Representations, and Intellectual Entities.

5.5.3.1 Bitstreams

Bitstream Objects are subsets of files. A Bitstream Object is defined as data (bits) within a file that a) have common properties for preservation purposes, and b) cannot stand alone without adding a file header or other structure. So for example, if you had a file in AVI (audio-video interleaved) format, you might want to distinguish the audio bitstream from the video bitstream, and describe them as separate Bitstream Objects.

5.5.3.2 Files

A File Object is just what it sounds like, a computer file, like a PDF or JPEG.

5.5.3.3 Representations

A Representation Object is the set of all File Objects needed to render an Intellectual Entity. For example, say you want to preserve a Web page, perhaps your institution's home page as of some date. The chances are good that the home page you see in your browser is actually composed of many different files – one or more HTML files, a handful of GIF or JPEG images, maybe a little audio or Flash animation. It probably also uses a stylesheet to create the display you see. It takes all of these files together for a browser to render the home page for viewing, so if a repository wants to preserve a renderable home page, it has to know about all these files and how to put them together. The Representation Object allows the repository not only to identify the set of related files, but also to describe characteristics of the totality (e.g. the Web page as a whole) that may be different from any of its parts.

5.5.3.4 Intellectual Entity

An Intellectual Entity Object is defined as a set of content that is considered a single intellectual unit for purposes of management and description: for example, a particular book, map, photograph, or database. PREMIS does not generally define descriptive metadata pertaining to Intellectual Entities because there are plenty of descriptive metadata standards to choose from. From version 3 an Intellectual Entity can be described with descriptive

metadata outside of PREMIS or with preservation metadata as an Object within PREMIS. PREMIS says that an Object in a preservation system should be associated with the conceptual Intellectual Entity it represents by including an identifier of the Intellectual Entity in the metadata for the Object. So, for example, if we were preserving a copy of Buddhism: The Ebook: an Online Introduction we might use the ISBN as the link to the Intellectual Entity description in the PREMIS description of the ebook.

5.5.4 Using PREMIS with METS

When using PREMIS and METS together it is strongly advised and encouraged to use the published guidelines developed in cooperation by the PREMIS Editorial Committee and the METS Board <http://www.loc.gov/standards/premis/guidelines2017-premismets.pdf> which outlines the decisions needed to use the two standards together.

5.5.5 Vocabularies in PREMIS

The standard recommends uses of vocabularies in several semantic units. The vocabularies have been developed by the PREMIS Editorial Committee and the PREMIS users and are published by the Library of Congress. All relevant vocabularies are presented in the PREMIS Data Dictionary together with the semantic unit it is used in. All the available vocabularies can be found at this web resource: <http://id.loc.gov/vocabulary/preservation.html>.

5.5.6 A note on identifiers

In PREMIS each of the entities (objects, events, agents, rights) are identified by a generic set of identifier containers. These containers follow an identical syntax and structure consisting of an [entity]Identifier container holding two semantic units:

- [entity]IdentifierType
- [entity]IdentifierValue

The PREMIS data dictionary recognises that the use of identifier types is an implementation-specific issue and does not recommend or require vocabularies for identifier types. The Library of Congress has a Standard Identifiers Scheme (<http://id.loc.gov/vocabulary/identifiers.html>). Its use is recommended in this specification instead of implementation-specific vocabularies.

5.5.7 PREMIS in the IP:s

There is a CITS available for preservation metadata based upon PREMIS. There is one thing to be noted, and that is that the CITS for preservation metadata in no way describe a full preservation planning or archival plan for all possible software's and repositories it is a starting point but in the repository implementing the specification if it is not decided by the software used a need to do a preservation planning plan and go through PREMIS and set up for example which events and the granularity of events occurring in the archive that are stored is needed. The CITS for Preservation is to be used as the transfer format for preservation metadata which means the PREMIS implementation in the system can be as a database table or likewise and the XML-schema is used in the transfer of the preservation metadata.

5.6 Description standards

When a transfer is made it is important to connect the digital objects found in the information package with its descriptive information. The descriptive information can take many forms and have different flavours depending on if it is an archive, library or a museum needing the descriptive information. In many cases, they can use the same standards for descriptive information, and in others, there are different standards used. This section will in its first versions focus on the descriptive standards used within the archives. The specifications itself are not in any way restricted to be used only in an archival setting instead they are aimed at all different settings which means that a library or museum can use their descriptive standards in the same way that the archival standards is used.

5.6.1 Implementing description standards

The standards describe below are used in the creation of the archival information systems and in most cases one of the possible export formats is in the form of an XML-document following one of the archival XML-based formats described. The implementation will be needed to be done if the system does not have an export possibility and will consist of creating the mapping between the system and the selected format and then populate an XML-document. There might also be occasions where its needed to transform from the exported XML-format to the wanted XML-format where a mapping between the two formats is needed to be developed.

The specification for descriptions do not supply you with mapping or transformations to achieve your wanted output the goal of the specification is to give you the options in formats to use.

5.6.2 Encoded Archival Description (EAD)

EAD is the Archival Description it can also be called the Finding Aid.

This document describes the scope, structure, and other specific information about the archival material itself. The document follows a structure developed by the International Council on Archives (ICA) called the General International Standard Archival Description (ISAD-G) (<https://www.ica.org/en/isadg-general-international-standard-archival-description-second-edition>). ISAD-G does not provide a transfer format but uses the Encoded Archival Description (EAD) (<http://www.loc.gov/ead/index.html>) maintained by the Technical Subcommittee on Encoded Archival Standards (<https://www2.archivists.org/governance/handbook/section7/groups/Standards/TS-EAS>).

5.6.3 Encoded Archival Creators- Corporate Bodies, Persons and Families (EAC-CPF)

EAC-CPF is describing the Archival Creator.

This document provides information about the individual or organisation that created the records. The document follows a structure developed by the International Council on Archives (ICA) called the International Standard Archival Authority Record for Corporate Bodies, Persons and Families (ISAAR(CPF)) (<https://www.ica.org/en/isaar-cpf-international-standard-archival-authority-record-corporate-bodies-persons-and-families-2nd>). ISAAR(CPF) does not provide a transfer format, but uses the Encoded Archival Context for Corporate Bodies, Persons and Families (EAC-CPF) <https://eac.staatsbibliothek-berlin.de/> maintained by the Technical Subcommittee on Encoded Archival Standards (<https://www2.archivists.org/governance/handbook/section7/groups/Standards/TS-EAS>).

5.6.4 Encoded Archival Guide (EAG)

EAG is the description of the Archival Institution itself.

This document provides information about the entity that maintains the archival holdings. The document follows a structure developed by the International Council on Archives (ICA) called the International Standard for Describing Institutions with Archival Holdings (ISDIAH) (<https://www.ica.org/en/isdiah-international-standard-describing-institutions-archival-holdings>). ISDIAH does not provide a transfer format but uses the Encoded Archival Guide (EAG) (<http://www.archivesportaleurope.net/eag>) maintained by the Archives Portal Europe Foundation (<http://www.archivesportaleuropefoundation.eu/index.php>).

5.6.5 Records in Context (RiC)

A content model together with an ontology binding together the description with the creators and their function is Records in Contexts.

The International Council on Archives (ICA) Expert Group on Archival Description (EGAD) (<https://www.ica.org/en/about-egad>) is creating a conceptual model for archival description called Records in Contexts (RiC). This consists of a model described in a textual form (<https://www.ica.org/en/egad-ric-conceptual-model>) and an ontology (<https://www.ica.org/en/records-in-contexts-ontology>). The first version will be published in November 2020. It will be possible to include documents following RiC in an information package.

5.7 [More standards needing a description can be added]

[More text].

6 Glossary

A glossary with terms mainly from the OAIS which are described here for understandability of the specifications and the guideline.

Table 1: Glossary

Name	Description
Component	Every part of the information package (Guideline)
Digital object	An object composed of a set of bit sequences. (OAIS)
Data	A reinterpretable representation of information in a formalised manner suitable for communication, interpretation, or processing. Examples of data include a sequence of bits, a table of numbers, the characters on a page, the recording of sounds made by a person speaking, or a moon rock specimen. (OAIS)
Metadata	Data about other data (OAIS)
Manifest	A listing of all the content in a package. In the simplest form it can be a list of file names. (Guideline)

Preservation plan	In the archival facility you need to have a plan for the preservation regardless if its physical or digital objects. The preservation plan for digital objects includes for example migration planning. For gaining a fuller understanding the Digital Preservation Handbook, 2nd Edition, https://www.dpconline.org/handbook Digital Preservation Coalition © 2015 licensed under the Open Government Licence v3.0. can be used. (Guideline)
Data object	Either a physical object or a digital object. (OAIS)
Descriptive information	The set of information, consisting primarily of package descriptions, which is provided to data management to support the finding, ordering, and retrieving of OAIS information holdings by Consumers. (OAIS)

DRAFT V.0.2

7 Metadata

In this section the explanation of the different tables and key terminology in the tables are given.

7.1 Explanation of tables and values used in the specification

In the specifications there are several different tables and values used which are described in the following sections.

7.2 Specification tables

In the specifications tables describes the different requirements needed to be full filled to be following the specification.

Table 2: Requirement tables headings

ID	Name, Location and Description	Card & Level
[ID]	[Name of element] [XPath to element] [Description of the element]	[Cardinality 1..1 and so on] [Level: MUST, SHOULD, MAY]

Table 3: Explanation of the parts of the requirement table

Term	Explanation
[ID]	Identification number of the requirement. The numbering is unique and built upon acronym for specification and a running number. There are no renumbering occurring which means if a requirement gets outdated the number is obsolete and not used.
[Name of element]	Name of the element in human readable form.
[XPath to element]	The XPath describing the location of the element in the XML-document.
[Description of the element]	A longer description of the purpose of the elements and links to extending information as well as other information pertaining to the element and described in another place. For example, values of value lists.
[Cardinality]	Possible occurrence of the element. See explanation in section "5.4 Cardinality values".
[Level]	The level of requirement of the element. See explanation in section "5.5 Level of requirement values".

7.3 Cardinality values

The cardinality gives the number of possible occurrences of an element.

Table 4: Cardinality

Cardinality	In human reading	DTD	XML-schema
[0..1]	Zero or once	?	minOccurs=0 maxOccurs=1
[0..n]	Zero or one or more times	*	minOccurs=0 maxOccurs=n minOccurs=0 maxOccurs=unbounded
[1..1]	Once	-	minOccurs=1 maxOccurs=1
[1..n]	One or more times	+	minOccurs=1 maxOccurs=unbounded minOccurs=1 maxOccurs=n

7.4 Level of requirement values

The level gives the requirement of an element following RFC 2119 <http://www.ietf.org/rfc/rfc2119.txt>.

Table 5: Level of requirement

Term	Explanation
MUST	This word means that the definition is an absolute requirement.
SHOULD	This word means that in particular circumstances valid reasons may exist to ignore the requirement, but the full implications must be understood and carefully weighed before choosing a different course.
MUST NOT	This phrase means that the prohibition described in the requirement is an absolute prohibition of use of the element.
SHOULD NOT	This phrase means that in particular circumstances violating the prohibition described in the requirement is acceptable or even useful, but the full implications should be understood and the case carefully weighed before doing so. The requirement text should clarify such circumstances.
MAY	This word means that an item is not prohibited but fully optional.

7.5 Identifiers in the IP specifications

The recommendation in the specifications is to use globally unique identifiers. With that written, showing examples where UUID's are used make the text hard to read so from a user perspective the ID's in the used examples have been shortened.

8 In depth information regarding different concepts and terms

[Explain more concepts or other text needed for making understanding the specification and its use easy]

8.1 CS and CITS

This section will be extended for the publication occurring after implementation of review comments.

[An explanation of CS and CITS]

8.2 In the CSIP there Explanation of redundancy and incompatible requirements

Following the <https://github.com/DILCISBoard/E-ARK-CSIP/issues/570> further explanation on how to read some requirements in combination with each other are given.

In the CSIP there might be requirements having the same XPath but different cardinality and level. The goal for each requirement is that each requirement only contains one rule and not multiple rules. This means that there might be more than one requirement pertaining to one XPath thus making it look like there are incompatible requirements. This means that there are going to be more than one requirement at occasions where you see the same XPath but different cardinality and level. The rule when reading and understanding the specifications is that if the first requirement with the XPath is fulfilled the next one with the same XPath needs to follow also that requirement. Let's look closer at some example:

8.2.1 Example 1

Metadata division

```
mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Metadata']
```

CSIP88 The metadata referenced in the administrative and/or descriptive metadata section is described in the structural map with one sub division. **1..1 MUST**
When the transfer consist of only administrative and/or descriptive metadata this is the only sub division that occurs.

Metadata division label

```
mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Metadata']
```

CSIP90 The metadata division `<div>` element's `@LABEL` attribute value must be "Metadata". **1..1 MUST**
See also: [File group names](#)

Explanation: CSIP88 tells us that there needs to be a division in the structural map for metadata this requirement is only pertaining to the obligation of having the division for metadata. CSIP90 tells us that the mandatory division needs to have the value "Metadata" following the vocabulary named "File group names" in the attribute named LABEL.

8.2.2 Example 2

Documentation division

```
CSIP93 mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Documentation']
```

The documentation referenced in the file section file groups is described in the structural

0..1 SHOULD

map with one sub division.

Documentation division label

`mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Documentation']`

CSIP95 The documentation division `<div>` element in the package uses the value “Documentation” from the vocabulary as the value for the `@LABEL` attribute. **1..1 MUST**

See also: [File group names](#)

Explanation: CSIP93 tells us that there might be a division in the structural map for documentation this requirement is only pertaining to the occurrence of having the division for documentation. CSIP95 tells us that if we have a division for documentation there needs to be a division having the value “Documentation” following the vocabulary named “File group names” in the attribute named LABEL.

8.2.3 Example 3

Schema division

`mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Schemas']`

CSIP97 The schemas referenced in the file section file groups are described in the structural map within a single sub-division. **0..1 SHOULD**

Schema division label

`mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Schemas']`

CSIP99 The schema division `<div>` element’s `@LABEL` attribute has the value “Schemas” from the vocabulary. **1..1 MUST**

See also: [File group names](#)

Explanation: CSIP97 tells us that there might be a division in the structural map for schemas this requirement is only pertaining to the occurrence of having the division for schemas. CSIP99 tells us that if we have a division for schemas there needs to be a division having the value “Schemas” following the vocabulary named “File group names” in the attribute named LABEL.

8.2.4 Example 4

Content division

`mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Representations']`

CSIP101 When no representations are present the content referenced in the file section file group with `@USE` attribute value “Representations” is described in the structural map as a single sub division. **0..1 SHOULD**

Content division label

`mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Representations']`

CSIP103 The package’s content division `<div>` element must have the `@LABEL` attribute value “Representations”, taken from the vocabulary. **1..1 MUST**

See also: [File group names](#)

Explanation: CSIP101 tells us that there might be a division in the structural map for content this requirement is only pertaining to the occurrence of having the division for content. CSIP103 tells us that if we have a division for content there needs to be a division having the value “Representations” following the vocabulary named “File group names” in the attribute named LABEL.

8.3 Explanation of principles in CSIP

The principles have been created for setting a common ground for transferring information/data/digital objects to an archive no matter the type of archive. The principles thus work for private archives, regional archives, national archives and all the archives that can be imagined which receives digital objects to preserve for the future.

8.3.1 Explanation of Principle 1.1

*It **MUST** be possible to include any data or metadata in a Information Package regardless of its type or format.*

The principle is created and set as the most critical principle that needs to be full filled. If an information package limits what you can put inside it is not a common information package that can be used by all possible users and is not having a practical interoperability spanning all sectors and tools which is included in the wording common.

8.3.2 Explanation of Principle 1.2

*The Information Package **MUST NOT** restrict the means, methods or tools for exchanging it.*

The principle describes the need for a common information packages need to be able to exchanged between users, repositories, researchers and so on In all possible ways. It needs to be possible to exchange the package with the help of usb stick as well as through the eDelivery services provided by the eDelivery Building Block.

8.3.3 Explanation of Principle 1.3

*The package format **MUST NOT** define the scope of data and metadata which constitutes an Information Package.*

The principle describes that the information package scope needs to be decided upon. It is possible to see an export of all the content an ERMS or a single file as an information package. In the explanation given in the principle the word “intellectual” is introduced, this follows PREMIS and the Intellectual Entity where the intellectual entity can be the real life entity for example a physical on paper printed book which in its turn has been digitised and then packed in an information package and sent to the archives. Then the printed book is seen as the intellectual entity for the digitised book being stored in an information package.

The explanation is also discussed in <https://github.com/DILCISBoard/E-ARK-CSIP/issues/569>.

8.3.4 Explanation of Principle 1.4

*The Information Package **SHOULD** be scalable.*

The principle describes the need for being able to divide the information package into manageable chunks since size do matters. A thoroughly description is currently 2020 under development.

8.3.5 Explanation of Principle 1.5

*The Information Package **MUST** be machine-readable*

The principle describes the fact that it's the machines that are supposed to be able to handle the information package even if using standards with expressions in formats readable by the human eye is used. The repository tools need to be able to handle the information package and not needing human aid.

8.3.6 Explanation of Principle 1.6

*The Information Package **SHOULD** be human-readable*

The principle states the fact that even if the information package is supposed to be for the machines its needed for it to be readable and understandable by a human by using simple text editors and file viewers in case the dedicated tools doesn't work.

8.3.7 Explanation of Principle 1.7

*The Information Package **MUST NOT** prescribe the use of a specific preservation method*

The principle declares that the preservation planning and preservation methods used in the repository is decided upon by the repository. For example it's the repository who needs to create the preservation planning and decide upon how migration is handled and recorded, which events occurring to the digital objects in the repository is recorded and how rights regarding the digital objects is stored and expressed. A lot of other things is part of the preservation in the repository and only a few is mentioned here.

8.3.8 Explanation of Principle 2.1

*The Information Package OAIS type (SIP, AIP or DIP) **MUST** be clearly indicated.*

The principle explains the need of knowing where in the archiving lifecycle described in the OAIS Reference Model the information package currently is. This is needed since what happens to the information package is depending on where it is. The SIP is incoming and are thus going under actions being performed before it becomes an AIP like validation of correctness and virus check so it can be put into the repository.

8.3.9 Explanation of Principle 2.2

*Any Information Package **MUST** clearly identify the Content Information Type(s) of its data and metadata.*

The principle is describing the need for knowing the type of information that are placed in the information package. This is needed since there will be different actions preformed depending on the content and it can be automatised when the content is following a pre-set of rules from a CITS. For example, images will be handled differently than a database. It is the same time also needed to know if no specifications has been used for the content when it was placed in the information package.

8.3.10 Explanation of Principle 2.3

*Any Information Package **MUST** have an identifier which is unique and persistent within the repository.*

The principle enforces the need of being to be able to identify an information package in the repository with a unique for the repository identifier. The usual way of knowing the information packages found in the repository is to have the identifier listed in an inventory like an archival description and thus it can be

found. This identifier is for the package as an information package and do not look at the digital objects within the information package.

8.3.11 Explanation of Principle 2.4

*Any Information Package **SHOULD** have an identifier which is globally unique and persistent.*

The principle is to be seen in coherence with principle 2.3 and is a reminder of the usefulness of the identifiers to besides being unique in the repository also being unique and persistent in the wider context to facilitate cross-institutional information exchange and reuse scenarios of whole information packages. At the same time the principle in no way enforces the method for identification or type of identifications used.

8.3.12 Explanation of Principle 2.5

*All components of an Information Package **MUST** have an identifier which is unique and persistent within the repository.*

The principle explains the need for all the components and with components it means all the digital objects found in the package, in short, all files no matter what type of file it is need to have an identifier so it is possible to link them to each other as needed. This is a principle only concerning each information package and it can easily be transferred to a repository unique identifier with addition of the package identification to the component identification.

8.3.13 Explanation of Principle 3.1

*The Information Package **MUST** ensure that data and metadata are logically separated from one another.*

The principle is describing the importance of easily being able to see which of the digital objects (components) that is describing metadata concerning the data objects in the information package. This differentiation is important since in for example format migration events the data is the digital objects needing the migration not the metadata which is saved in formats usually not needing migration like XML. The logical separation is achieved using a manifest which describes all the digital objects in the information package. Currently the used standard for the manifest is METS.

Observe that some formats used for the content information type specifications in themselves contain metadata. This is not supposed to be moved out of the format and placed separately instead see the metadata in this requirement to be the overall needed metadata for understanding the package.

8.3.14 Explanation of Principle 3.2

*The Information Package **SHOULD** ensure that data and metadata are physically separated from one another.*

The principle is making the principle 3.1 easier to achieve thorough not having the manifest but also using a folder structure giving the separation of data and metadata.

8.3.15 Explanation of Principle 3.3

*The structure of the Information Package **SHOULD** allow for the separation of different types of metadata*

The principle takes principle 3.1 and 3.2 to the next level where the different kinds of metadata is divided and separated in to at least two main categories of metadata for the information package, descriptive and preservation metadata. The separation should be in both the logical description and the physical structure of the information package. Where the descriptive metadata is for example a description of the creator of the data in the form of an EAC-CPF document and preservation metadata is a PREMIS document.

8.3.16 Explanation of Principle 3.4

*The structure of the Information Package **MUST** allow for the creation of data and metadata in multiple representations.*

The principle outlines the constantly evolving digital preservation need of being able to migrate and create new data and metadata throughout the digital object's lifecycle the creation of a new representation of the digital objects. This means that the possibility to fully understand the digital objects lifecycle and events occurring in the lifecycle and preservation is easily achieved.

8.3.17 Explanation of Principle 3.5

*The structure of the Information Package **MUST** explicitly define the possibilities for adding additional components into the Information Package.*

The principle ensures the possibility of adding what is needed into the information package to get a information package fulfilling all needs of all different kind of users. This is especially important when it comes to regulations in different countries, to different sectors since all have different kind of needs that needs to be fulfilled in the creation of an information package. For example an transfer of an information package to a national archive which can be seen as the last transfer for the information might demand the XML-schemas used for creating the metadata structures to be added to ensure the possibility of understanding and validating the data in the preservation environment.

8.3.18 Explanation of Principle 3.6

*The Information Package **SHOULD** follow a common conceptual structure regardless of its technical implementation.*

The principle is the combination of principle 3.1-3.5 and explains the need for being consistent in the implementation to ensure the possibility of a collaborative way of creating tools which works in all settings. Currently the CSIP is implementing the principles with the use of a folder structure which can have folders added by the user and a manifest describing the package in a readable form using METS.

8.3.19 Explanation of Principle 4.1

*Metadata in the Information Package **MUST** conform to a standard.*

The principle enforces the use of metadata standards for describing the metadata in the package. Using a standard makes it easy to understand and share the information surrounding the digital objects. In

addition, using a standard ensures it is wide spread and used by others and that the standard is having all elements and attributes needed for the type of data it is describing.

8.3.20 Explanation of Principle 4.2

*Metadata in the Information Package **MUST** allow for unambiguous use.*

The principle is enforcing the need of writing profiles for the different metadata standards used so it is possible to make sure it is not open and needing interpretations to understand the data.

8.3.21 Explanation of Principle 4.3

*The Information Package **MUST NOT** restrict the addition of supplementary metadata.*

The principle suggests the use of other metadata which aids with discovery in the form of descriptive metadata and technical and structural metadata for the content itself.

8.4 Folder structure requirements

To ease the understanding of the information package in case the manifest gets lost a folder structure is suggested and enforced in the validation. The folders structure is described with this set of requirements.

8.4.1 Explanation of CSIPSTR1

*Any Information Package **MUST** be included within a single physical root folder (known as the “Information Package root folder”). For packages presented in an archive format, see CSIPSTR3, the archive **MUST** unpack to a single root folder.*

The requirement describes that there should always be a top folder in which all content is placed so when unpacking the information package all digital objects in the package ends up in one root folder.

8.4.2 Explanation of CSIPSTR2

*The Information Package root folder **SHOULD** be named with the ID or name of the Information Package, that is the value of the package METS.xml’s root <mets> element’s @OBJID attribute.*

The requirement suggest that the root folder is named the same thing as the identification of the package found in the METS document in the attribute OBJID.

8.4.3 Explanation of CSIPSTR3

*The Information Package root folder **MAY** be compressed (for example by using TAR or ZIP). Which specific compression format to use needs to be stated in the Submission Agreement.*

The requirement suggest that the information package is packed as one file using for example packing into TAR- or ZIP-format. Which package format to use needs to be agreed upon in the submission agreement.

8.4.4 Explanation of CSIPSTR4

*The Information Package root folder **MUST** include a file named METS.xml. This file **MUST** contain metadata that identifies the package, provides a high-level package description, and describes its structure, including pointers to constituent representations.*

The requirement gives that there needs to be a manifest inform of a METS document. This document needs to be named METS.xml. Its important to notice that due to computer operating systems the files METS.xml, Mets.xml, mets.xml and more options can be seen as different files and mean that when unpacking there will be just one saved thus its important to make sure the file is named METS.xml.

8.4.5 Explanation of CSIPSTR5

The Information Package root folder SHOULD include a folder named metadata, which SHOULD include metadata relevant to the whole package.

The requirement suggests the use of a folder named metadata on the top level for metadata documents pertaining to the whole package. Examples on metadata at this level is a full archival description over the whole package, a PREMIS document concerning all the different digital objects found in the data folder. There might be metadata stored in the different representations and therefore there is also possible to have the metadata folder in a representation.

The explanation is also discussed in <https://github.com/DILCISBoard/E-ARK-CSIP/issues/291>.

8.4.6 Explanation of CSIPSTR6

If preservation metadata are available, they SHOULD be included in sub-folder preservation.

The requirement recommends that the metadata folder is having a subfolder named preservation for storing the preservation metadata most likely in the format PREMIS.

8.4.7 Explanation of CSIPSTR7

If descriptive metadata are available, they SHOULD be included in sub-folder descriptive.

The requirement recommends that the metadata folder is having a subfolder named descriptive for storing the descriptive metadata which can be in several formats like EAD3, EAC-CPF, RiC-O, MARC and more.

8.4.8 Explanation of CSIPSTR8

If any other metadata are available, they MAY be included in separate sub-folders, for example an additional folder named other.

The requirement recommends that other metadata is stored in a sub folder named other. This is for metadata which can't be sorted as being either preservation or descriptive metadata.

8.4.9 Explanation of CSIPSTR9

The Information Package folder SHOULD include a folder named representations.

The requirement recommends the sub folder representation for placing the different representations into its own subfolders

8.4.10 Explanation of CSIPSTR10

The representations folder SHOULD include a sub-folder for each individual representation (i.e. the "representation folder"). Each representation folder should have a string name that is unique within the package scope. For example, the name of the representation and/or its creation date might be good candidates as a representation sub-folder name.

The requirement suggest that the representation folder contains sub folders for the different representations of the information package.

8.4.11 Explanation of CSIPSTR11

The representation folder SHOULD include a sub-folder named data which MAY include all data constituting the representation.

The requirement suggests that all data being the digital objects and not metadata transferred in the information package is placed in the representation sub folder named data.

8.4.12 Explanation of CSIPSTR12

The representation folder SHOULD include a metadata file named METS.xml which includes information about the identity and structure of the representation and its components. The recommended best practice is to always have a METS.xml in the representation folder.

The requirement is describing that each representation can be described with a METS document which means the METS-document in the top folder pertains to the whole information package and that each representation can be described by its own METS document. This means that the top METS document points to the lower METS documents and do not describe the digital objects in the representations more than the METS.xml document. More description of representations in other sections.

8.4.13 Explanation of CSIPSTR13

The representation folder SHOULD include a sub-folder named metadata which MAY include all metadata about the specific representation.

The requirement is describing the possibility to add a metadata folder in the different representations which then is the folder to store the metadata that pertain to the digital objects being found in the representation.

8.4.14 Explanation of CSIPSTR14

The Information Package MAY be extended with additional sub-folders.

The requirement describes the possibility to be able in the representation to add all needed sub folders.

8.4.15 Explanation of CSIPSTR15

We recommend including all XML schema documents for any structured metadata within package. These schema documents SHOULD be placed in a sub-folder called schemas within the Information Package root folder and/or the representation folder.

The requirement is stressing the need that to make the information package long term sustainable all structured metadata should have its schemas in the package in a folder named schemas. The schema folder can be found in the root folder of the package and there have all used schemas. It is also possible to have the schema folder in the representation. During the preservation planning it's also needed to figure out the extent of which schemas for structured information to have in the package or available in the preservation system. Even the structured information standard XML itself have an xml.xsd document with its rules. The xml.xsd is maintained by the W3C and is found here, <https://www.w3.org/2001/03/xml.xsd>

8.4.16 Explanation of CSIPSTR16

We recommend including any supplementary documentation for the package or a specific representation within the package. Supplementary documentation SHOULD be placed in a sub-folder called documentation within the Information Package root folder and/or the representation folder.

The requirement is closely connected to the submission agreement and what information the receiver is stating is needed to understand the digital objects when they have been transferred. The supplementary documentation includes for example manuals, screenshots of the system in use and other documentation informing about the use of the digital objects being part of the transfer.

8.5 Explanation of the concept of representations

This section will be extended for the publication occurring after implementation of review comments.

Issue <https://github.com/DILCISBoard/E-ARK-CSIP/issues/558>] shows that the concept needs to be explained in depth.

8.5.1 Explanation of METS Header element and representations

It is important to differentiate the concept of representations from the use of the METS Header element in each of the METS documents. The information in the METS Header elements pertains to the METS document with some information being specific in using the OAIS reference model. This is seen especially for in the information in CSIP9 which is pertaining to the package's status or placement in the OAIS reference model. All the other values are pertaining to the METS -document itself and the creation thus also to the package. The one mandatory agent describes the software which created the package. In local adoptions there might be other agents present, this is best seen in the E-ARK SIP which adds several agents both mandatory ones and optional ones which are important in the transferring moment.

8.5.2 Explanation of levels of packages and nesting of representations

This section will be extended for the publication occurring after implementation of review comments.

This section will explain issues: <https://github.com/DILCISBoard/E-ARK-CSIP/issues/322>
<https://github.com/DILCISBoard/E-ARK-CSIP/issues/368>

8.6 Signatures

This section will be extended for the publication occurring after implementation of review comments.

This section will explain the concept of digital signatures in an archival setting compared to the use of signatures as the method of ensuring the archival storage.

8.7 Vocabularies

Throughout the specifications several different vocabularies are used to give a fixed set of values for describing metadata and information regarding what is being described. Some vocabularies are created by

the standards and hosted in value lists in the XML-schemas or is possible to reference as linked data vocabularies. There are also other vocabularies defined by the different specifications to give a conformant use of attributes and/or elements. All vocabularies created for the specifications is published next to the specification and is fully available and free for others to use. The vocabularies have descriptions of the different terms, so they are easily to understand. The name of the file containing the vocabulary might not be the same as the name of the vocabulary since more than one vocabulary can be contained in the same file depending on how the vocabularies are maintained.

8.8 Referencing

This section will be extended for the publication occurring after implementation of review comments.

The section will explain issue: <https://github.com/DILCISBoard/E-ARK-CSIP/issues/406>

9 Validation

This section will be extended for the publication occurring after implementation of review comments.

This section will explain how to validate an information package following CSIP and give insights to issue: <https://github.com/DILCISBoard/E-ARK-CSIP/issues/398>

10 Own adoptions of the specifications

10.1 Adapting CSIP/SIP/AIP/DIP specifications

It is possible to do adaptations in the form of extending the different package specifications with an extra extending METS profile. The extending profiles is adding requirements or changing their cardinality (the correct way is to: change optional to mandatory and specify the number of occurrences). It is not allowed to remove requirements since this will make the implementation invalid. The best way of seeing how this is done is to examine the METS Profile for CSIP and after that the METS Profile for SIP or DIP.

10.2 Adapting any CITS specifications

It is possible to do adaptations in the form of extending the different content information type specifications with adding requirements or changing their cardinality (the correct way is to: change optional to mandatory and specify the number of occurrences). For some of the CITS there is a number of decisions needed to be made and all these decisions needs to be documented so the use of the CITS can be understood in its context.

10.3 Adapting PREMIS

Using the PREMIS specification and adding own requirements is possible. It's more important to look into the use of PREMIS and create a preservation plan for your repository and make sure PREMIS is used in the system you are buying or developing, and this might put more demands on the PREMIS use in the local system than what we have prescribe.

11 Example following CSIP

This section will be extended for the publication occurring after implementation of review comments.

In this section an XML document built upon the METS Profile for CSIP is described.

11.1 Part 1 of example, use of the mets element

The first part of an XML-document following the CSIP specification is the mets-element itself. It contains the information regarding which XML-schemas are used. It is here where the extension schema for the extra attributes extending METS through CSIP is referenced. There are also several attributes used which are described as requirements in the CSIP specification. Observe the use of the csip:OTHERTYPE attribute which is coming from the own extension schema for CSIP and triggered to be used by the attribute TYPE having the value "OTHER".

```
<mets:mets
  xmlns:csip="https://DILCIS.eu/XML/METS/CSIPExtensionMETS"
  xmlns:mets="http://www.loc.gov/METS/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  OBJID="uuid-4422c185-5407-4918-83b1-7abfa77de182"
  LABEL="Sample CSIP Information Package"
  TYPE="OTHER"
  csip:OTHERTYPE="Patterns"
  PROFILE="https://earkcsip.dilcis.eu/profile/E-ARK-CSIP.xml"
  xsi:schemaLocation=
    "http://www.loc.gov/METS/ http://www.loc.gov/standards/mets/mets.xsd
    http://www.w3.org/1999/xlink http://www.loc.gov/standards/mets/xlink.xsd
    https://DILCIS.eu/XML/METS/CSIPExtensionMETS
    https://earkcsip.dilcis.eu/schema/DILCISExtensionMETS.xsd">
```

The package being transferred is built and described following the METS profile belonging to the CSIP. The content in the package is not found in the value list described here [Correct referencing will be part of publication] for the attribute TYPE therefore the value is set to "OTHER" and the csip:OTHERTYPE attribute shows that the content is defined as "Patterns" following the vocabulary [Correct referencing will be part of publication]. The METS document has its OBJID attribute being set to the name of the root folder of the package in this case a UUID. The sender has given the package a LABEL so they know what they sent.

11.2 Part 2 of example, use of the metsHdr element

The next element in the mets document is the mets header element which gives information regarding the mets document itself. One of the things being described is the various forms of agents who have taken part in the creation of the mets document and the content described in the package.

```
<mets:metsHdr
  CREATEDATE="2018-04-24T14:37:49.602+01:00"
  LASTMODDATE="2018-04-24T14:37:49.602+01:00"
  RECORDSTATUS="NEW"
  csip:OAISPACKAGETYPE="SIP">
  <mets:agent ROLE="CREATOR" TYPE="OTHER" OTHERTYPE="SOFTWARE">
    <mets:name>RODA-in</mets:name>
```

```
        <mets:note csip:NOTETYPE="SOFTWARE VERSION">2.1.0-beta.7</mets:note>
    </mets:agent>
</mets:metsHdr>
```

The package was created at a specific date given in the attribute CREATEDATE which in this case are the same as the last modification date seen in the attribute LASTMODDATE. There will be packages where these two dates will differ since there might be modifications in metadata needed after the creation. The package is also a new package seen in the attribute RECORDSTATUS following the value list [Correct referencing will be part of publication]. The CSIP specification has been extended with an own attribute named OAISPOACKAGETYPE to make it easier to see where the package is in the OAIS reference model. The agent described is giving the software that have created the package , information that can prove vital when an error occurs and packages are needed to be checked for errors due to the specific version of the software used.

11.3 Part n+1 of example

This section will be extended for the publication occurring after implementation of review comments.

All parts of the example will be explained and walked through in its own small section.

[An explanation of last part of example with comments]

11.4 Full example

This section will be extended for the publication occurring after implementation of review comments.

[The full example in XML]

12 Postface

AUTHOR(S)

Name(s)	Organisation(s)
Karin Bredenber	Kommunalförbundet Sydarkivera

REVIEWER(S)

Name(s)	Organisation(s)
[Name]	[Affiliation]
[Name]	[Affiliation]
[Name]	[Affiliation]

Project co-funded by the European Commission Support Programme

within the ICT Policy

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	2020-08-10	Karin Bredenberg	Kommunalförbundet Sydarkivera	First draft version
0.2	2021-02-12	Karin Bredenberg	Kommunalförbundet Sydarkivera	Draft version in support of the new guidelines being on review during February - May 2021.

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.