

# **DRAFT E-ARK Common Specification for Information Packages**

---

**Version:** 2.0.0-DRAFT

**Published:** November 28, 2018

# Contents

---

- I. [Acknowledgements](#)
- II. [Contact & Feedback](#)
- III. [Authors](#)
- IV. [Revision History](#)
- [1 Introduction](#)
  - [1.1 The Common Specification for Information Packages and OAIS](#)
  - [1.2 The Common Specification for Information Packages and Content Information Type Specifications](#)
  - [1.3 Common Specification for Information Packages, OAIS Information Packages' specifications and Content Information Type Specifications](#)
  - [1.4. Relation to other documents](#)
    - [International standards and best-practices](#)
    - [E-ARK project \(2014 – 2017\) deliverables](#)
    - [Other E-ARK specifications](#)
  - [1.5. Structure of the document](#)
- [PART I: Common Specification for Information Packages](#)
  - [2.Need for establishing common ground](#)
  - [3. Requirements for CS IP Information Packages](#)
    - [3.1. General principles](#)
    - [3.2. Identification of the Information Package](#)
    - [3.3. Structure of the Information Package](#)
    - [3.4. Information Package Metadata](#)
- [PART II: Implementation of the CS IP](#)
  - [4.CS IP structure](#)
    - [4.1 Folder structure of the CS IP](#)
    - [4.2 Implementing the structure](#)
  - [5. Use of metadata](#)
    - [5.1. General requirements for metadata in a CS IP Information Package](#)
    - [5.2. General requirements for the use of metadata](#)
    - [5.3. Use of METS](#)
      - [5.3.1. Use of the METS root element \(element mets\)](#)
      - [5.3.2. Use of the METS header \(element metsHdr\)](#)
      - [5.3.3 Use of the METS descriptive metadata section \(element dmdSec\)](#)
      - [5.3.4. Use of the METS administrative metadata section \(element amdSec\)](#)
      - [5.3.5. Use of the METS file section \(element fileSec\)](#)
      - [5.3.6. Use of the METS structural map \(element structMap\)](#)

- [5.4. Use of PREMIS](#)
- [6. Implementation considerations](#)
  - [6.1. Content Information Type Specifications](#)
    - [6.1.1. What is a Content Information Type Specification?](#)
    - [6.1.2. Maintaining Content Information Type Specifications](#)
  - [6.2. Handling large packages](#)
    - [6.2.1. The structure for IP, their representations and their segments](#)
    - [6.2.2. Using METS to refer from parent IP to child IP](#)
    - [6.2.3. Using METS to refer from child IP to parent IP](#)
    - [6.2.4. An example for the Northwind database](#)
    - [6.2.5. Illustration of references between METS files in a segmented IP](#)
  - [6.3. Handling descriptive metadata within the Common Specification](#)
- [Appendices](#)
  - [Appendix A: E-ARK Information Package METS examples](#)
  - [Appendix B: External Schema and Vocabularies](#)

## Acknowledgements

---

The Common Specification for Information Packages was first developed within the E-ARK project in 2014 – 2017. E-ARK was an EC-funded pilot action project in the Competiveness and Innovation Programme 2007- 2013, Grant Agreement no. 620998 under the Policy Support Programme.

We would like to thank the National Archives of Sweden and Karin Bredenberg for their support and the availability of the Swedish national Common Specifications, upon which most of this document has been built.

The authors of this deliverable would like to thank all national archives, tool developers and other stakeholders who provided valuable knowledge about their requirements for information packages and feedback to this specification!

## Contact & Feedback

---

The Common Specification for Information Packages is maintained by the Digital Information LifeCycle Interoperability Standard Board (DILCIS Board). For further information about the DILCIS Board or feedback on the current document please consult the website <http://www.dilcis.eu/> or contact us at <dashboard@dilmforum.eu.>

## Authors

Name	Organisation
Karin Bredenberg	National Archives of Sweden
Björn Skog	ES Solutions
Anders Bo Nielsen	Danish National Archives
Kathrine Hougaard Edsen Johansen	Danish National Archives
Alex Thirifays	Danish National Archives // Magenta ApS
Sven Schlarb	Austrian Institute of Technology
Andrew Wilson	University of Portsmouth // University of Brighton
Tarvo Kärberg	National Archives of Estonia
Kuldar Aas	National Archives of Estonia
Luis Faria	Keep Solutions
Helder Silva	Keep Solutions
Miguel Ferreira	Keep Solutions
Carl Wilson	Open Preservation Foundation

## Revision History

Version	Date	Author	Org	Description
0.1	17.02 .2014	Björn Skog	ESS	First version.
0.2	21.02 .2014	Karin Bredenberg	ESS	Updating content.
0.3	24.02 .2014	Björn Skog	ESS	Updating content.
0.4	24.10 .2014	Tarvo Kärberg	NAE	Updating content.
0.41	05.11 .2014	Tarvo Kärberg	NAE	Adding content from Anders Bo Nielsen.
0.42	08.12 .2014	Tarvo Kärberg	NAE	Updating content.
0.43	19.12 .2014	Tarvo Kärberg	NAE	Updating content.
0.5	26.01 .2015	Kathrine Hougaard Edsen	DNA	Updating content.
0.6	11.02 .2015	Tarvo Kärberg	NAE	Rearranging content.
0.7	31.05 .2015	Kathrine Hougaard Edsen	DNA	Significant changes suggested

0.8	27.07 .2015	Tarvo Kärberg	NAE	Merging content
0.9	05.08 .2015	Andrew Wilson	UPH EC	Updating content
0.10	07.10 .2015	Kuldar Aas	NAE	Major update to include additional details
0.11	30.11 .2015	Kuldar Aas	NAE	Intermediate update to include outcomes and decisions from Common Specification meetings
0.12	08.12 .2015	Kuldar Aas	NAE	Update on the implementation, include comments from Sven, Jan (AIT) and Andrew (UPHEC)
0.13	05.01 .2016	Kuldar Aas, all	NAE, all	Update to include additional comments from E-ARK WPLs and Common Specification group members version sent for external review
0.14	04.07 .2016	Kuldar Aas	NAE	Updated structure -> basis for addressing review comments and required updates
0.15	26.08 .2016	Kuldar Aas	NAE	Adding available contributions to individual Sections
0.16	05.12 .2016	Andrew Wilson, Kuldar Aas	UoB, NAE	Major update. Added section on PREMIS. Revision of tables describing use of METS. General text revisions arising from CS meetings. Updates to requirements.

0.17	16.12 .2016	All	All	Final discussions, changes and proofreading before delivering the CS to public comment.
1.0	31.01 .2017	Kuldar Aas	NAE	Final small editorial additions
1.1	14.05 .2018	Kuldar Aas (editor), DILCIS Board	NAE	Limited proofreading and updates throughout the document. Updates in terminology. Updates in use of METS, ID and referencing section. Improved (more consistent) examples in METS section.
2.0.0	28.11 .2018	DILCIS Board		Draft for public review of updates to version 2.0.0



# 1 Introduction

---

This document introduces the concept of the Common Specification for Information Packages (CS IP). It aims to serve three main purposes:

1. Establish a common understanding of the requirements which need to be met in order to achieve interoperability of Information Packages;
2. Establish a common base for the development of more specific Information Package definitions and tools within the digital preservation community;
3. 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.

Ultimately the goal of the Common Specification for Information Packages (CSIP) is to reach a level of interoperability between all Information Packages so that tools implementing the CS IP can be taken up by institutions without needing further modifications or adaptations.

## 1.1 The Common Specification for Information Packages and OAIS

In the OAIS framework three types of Information Packages (IPs) are present in a digital preservation ecosystem: Submission Information Packages (SIPs), Archival Information Packages (AIPs) and Dissemination Information Packages (DIPs) ([Figure 1](#)). These three IP types are respectively used to submit data and metadata to digital repositories; store it in long-term preservation facilities; and deliver to consumers.

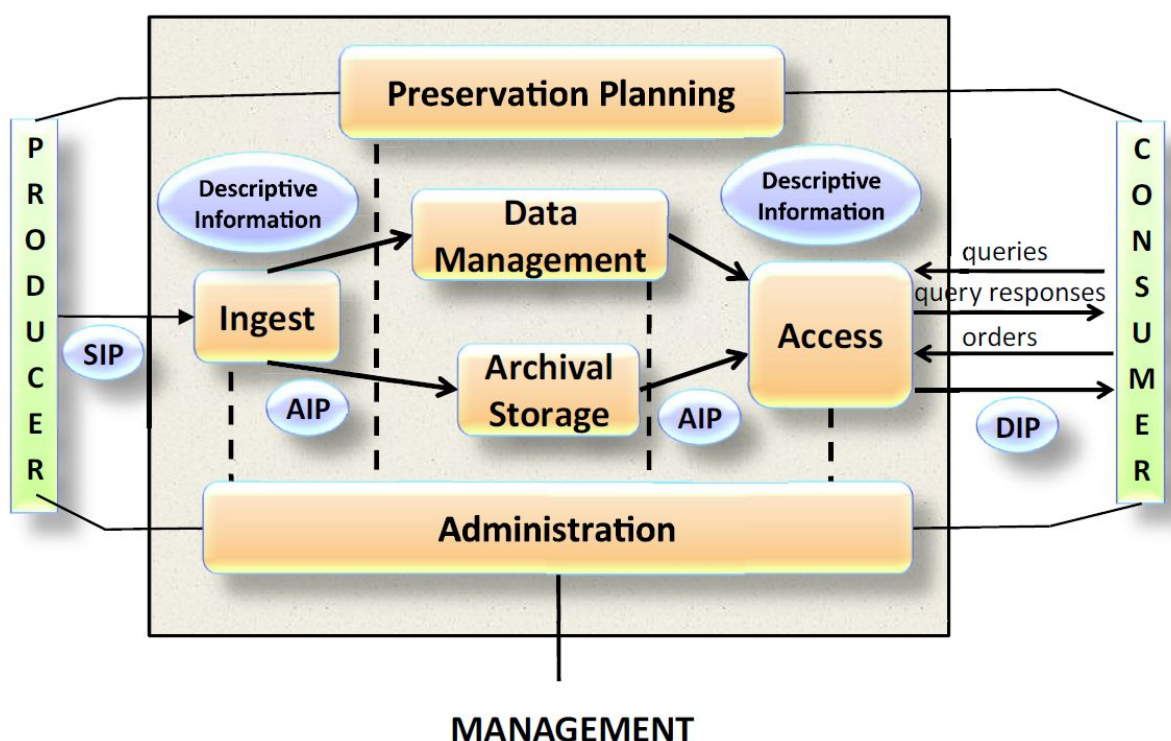


Figure 1: OAIS Functional Entities and Information Packages

The main goal in the development of this specification has been to identify and standardise the common aspects of IPs which are equally relevant and implemented by any of the functional entities of the overall digital preservation process presented in OAIS (i.e. pre-ingest, ingest, archival storage, data management and access). The practical implementation is that the specification therefore allows for the development of generic tools and code libraries which can either be applied commonly across the whole lifecycle of digital data, or be reused as the basis for developing more specific, content or process-aware tools.

To enable process level interoperability there needs to be detailed technical specifications for the OAIS information package types, e.g. SIP, AIP and DIP. For

the E-ARK specifications this Common Specification for Information Packages is accompanied by detailed E-ARK SIP, E-ARK AIP and E-ARK DIP implementation profiles.



Figure 2: The scope of Common Specification for Information Packages in regard to OAIS Information Packages.

In general, the E-ARK SIP and E-ARK DIP specifications reuse and apply fully all the requirements set in this Common Specification. However, they also extend it with aspects relevant only for the respective processes ([Figure 2](#)).

For example, the E-ARK SIP specification extends the CS IP with further requirements about recording relevant information on a submission agreement and the actors of the submission process. On the other hand, the E-ARK DIP provides possibilities for describing complex access environments needed to reuse the content of a DIP.

Regarding the E-ARK AIP format, it is important to note that it does not extend the CS IP in the same way the E-ARK SIP and E-ARK DIP formats do, i.e. in the sense of a format specification inheriting all general properties from the CS IP which is then augmented by specific AIP requirements. The reason for this is that while the SIP and the DIP are like “snapshots” in time – one capturing the state of an information package at time of submission (SIP), the other one capturing one form of delivery of the information for access (DIP) – then the AIP needs to deal with an “evolving object” which is constantly updated by preservation actions undertaken in the course of the objects life-cycle. As such, while the E-ARK AIP specification does implement all of the core metadata requirements defined in the Common Specification and extends these (for example it describes a means to record preservation actions about the IP), it does also extend the default structure of the CS IP (defined in

[Section 4](#)). Essentially the AIP introduces a more complex structure which allows at the same time to securely hold an E-ARK SIP (which itself follows in full the CS IP) and at the same time add and modify additional representations over a series of preservation actions.

## 1.2 The Common Specification for Information Packages and Content Information Type Specifications

---

As an interoperability standard, it must be possible to use the CS IP regardless of the type and format of the content users need to handle. At the same time, each individual content type and file format can have specific characteristics which need to be taken into account for purposes of validation, preservation and curation.

To allow for such in-depth control over specific content types and formats, E-ARK specifications introduce the concept of Content Information Type Specifications. A Content Information Type Specification can include detailed requirements on how content, metadata, and documentation for specific content types (for example relational databases or geospatial data) have to be handled within a CS IP (or E-ARK SIP, AIP or DIP).

As of November 2018 these Content Information Type Specifications, created by the E-ARK project and enhanced by the DILCIS Board, have been verified for usage within the Common Specification for Information Packages:

- E-ARK ERMS: Content Information Type Specification Electronic Records Management Systems. This Content Information Type Specification describes the use of the CS IP for the archiving of records exported from ERMS-type systems;
- E-ARK Geodata: This Content Information Type specifications build upon the INSPIRE directive and describes the storing geospatial information, within a CS IP compatible Information Package;
- E-ARK SIARD (SIARD1, SIARD2 and SIARDDK): This Content Information Type specification describe the usage of the CS IP for the archiving, preservation and reuse of relational databases in one of the formats in the SIARD family (Software Independent Archiving of Relational Databases). Note, that SIARD1 and SIARDDK specifications are deemed outdated by the time of writing and are only intended to be used for packaging already available SIARD1 and SIARDDK packages in a CS IP compatible Information Packages. For new occurrences of archiving relational databases the use of the SIARD2 format and according Content Information Type Specification is recommended.

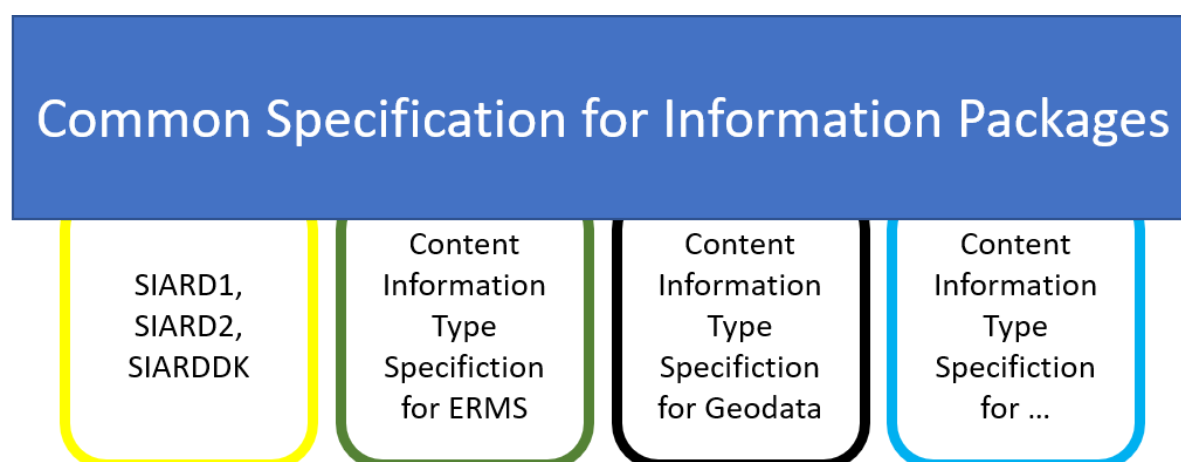


Figure 3: Common Specification for Information Packages and Content Information Type Specifications

The total number of Content Information Type specifications is, however, unlimited and the long-term commitment of the DILCIS Board is to keep the overall environment open and inclusive. As such, interested bodies are welcome to develop their own Content Information Type Specifications, for example for 3D building projects or electronic publications. An appropriate management regime to facilitate the creation and approval of additional Content Information Type specifications by anyone in the broader community is implemented by the DILCIS Board.

For more detailed information about the Content Information Type specifications please look also at [Section 6.1](#) below and check [www.dilcis.eu](http://www.dilcis.eu)!

### 1.3 Common Specification for Information Packages, OAIS Information Packages' specifications and Content Information Type Specifications

Following the discussions in the previous two Sections we can state that the overall ecosystem of E-ARK Common Specifications consists of 3-layers ([Figure 4](#)):

- The current document, the Common Specification for Information Packages, is the core which provides guidance which must be followed regardless of the process, data or lifecycle stage;
- The E-ARK SIP, AIP and DIP build on the CS IP and extend it with specific process-related aspects;
- The Content Information Type Specifications define detailed requirements for embedding and describing specific content types within a CS IP.

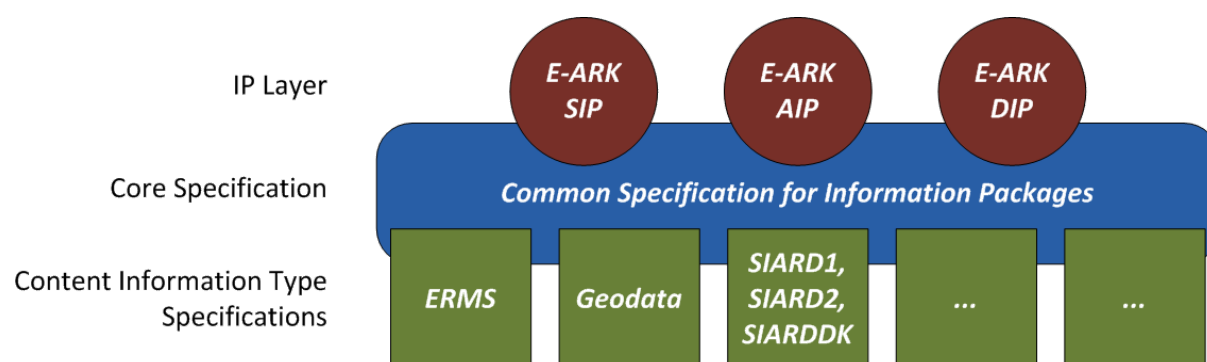


Figure 4: Relations between the Common Specification for Information Packages; E-ARK SIP, AIP and DIP specifications; and Content Information Type Specifications

Therefore the “thing encountered in the wild” is the E-ARK SIP, AIP or DIP including data according to one or many Content Information Type Specifications.

## 1.4. Relation to other documents

This Common Specification for Information Packages is related to the following documents:

### International standards and best-practices

- Reference Model for an Open Archival Information System (OAIS), 2012, [public.ccsds.org/publications/archive/650x0m2.pdf](http://public.ccsds.org/publications/archive/650x0m2.pdf) This specification has used the same terminology as introduced in the OAIS model and also the same division of information package types: Submission Information Package (SIP), Archival Information Package (AIP), Dissemination Information package (DIP).
- Producer-Archive Interface Specification (PAIS) – CCSDS, 2014, [public.ccsds.org/publications/archive/651x1b1.pdf](http://public.ccsds.org/publications/archive/651x1b1.pdf) We have investigated the structure of a SIP presented in OAIS, but as the implementation of this specification is not very comprehensive yet (only few prototypes exist), we decided to rely mainly on the best practices introduced in other reports (see below).

### E-ARK project (2014 – 2017) deliverables

- Deliverable D3.1, E-ARK Report on Available Best Practices
- Deliverable D4.1, Report on available formats and restrictions
- Deliverable D5.1, GAP report between requirements for access and current

access solutions

These three deliverables document the best-practice survey carried out during the first six months of the E-ARK project. Many of the core principles and requirements highlighted in the following Sections have been derived from these surveys.

## Other E-ARK specifications

- E-ARK SIP Specification
- E-ARK AIP Specification
- E-ARK DIP Specification

The E-ARK SIP, AIP and DIP specifications build on the Common Specification for Information Packages and extend it in regard to requirements derived from pre-ingest and ingest, archival storage, and access processes.

## 1.5. Structure of the document

---

The rest of this document describes the CS IP and its practical implementation. The document is divided into two logical parts.

The first part ([Section 2](#) and [Section 3](#)) describes the generic principles of the CS IP. The main aim of these Sections is to first identify a common set of needs and thereafter present a series of requirements which an Information Package needs to follow regardless of the implementation at any given point in time:

- [Section 2](#) provides an explanation of the need for a CS IP. The Section therefore presents some practical use cases which highlight the potential savings and increased functionality of digital archives when following internationally standardised approaches.
- [Section 3](#) presents the core principles which need to be met in order to achieve the interoperability goal described in Section 2. Based on these requirements a set of high-level solutions are introduced regarding, for example, the structure and use of metadata within any implementation of an Information Package.

The second part of this document ([Section 4](#), [Section 5](#) and [Section 6](#)) presents a practical implementation of the principles described in previous Sections, as implemented according to current state-of-the-art technologies. As such, this part of the document describes the requirements which are needed to achieve practical IP interoperability:



- [Section 4](#) presents a detailed description of the structure which must be implemented in any CS IP Information Package.
- [Section 5](#) presents a detailed overview of metadata requirements within CS IP Information Packages with a special focus on the use of metadata elements which are needed for the automation and interoperability of archival validation and identification tasks
- [Section 6](#) describes additional (optional) components extending the practical implementation in regard to specific aspects
  - How to create new Content Information Type specifications
  - How to split large content objects between multiple physical IPs
  - Generic guidelines on adding (any) descriptive metadata into a CS IP Information Package

Finally, in addition to this document full examples of IPs conforming to the Common Specification for Information implementation details are available at

<https://github.com/DILCISBoard/E-ARK-CSIP>.



# PART I: Common Specification for Information Packages

---

In this part of the document we build the argument for a Common Specification for Information Packages and present the main concepts and principles for the purpose.

## 2 Need for establishing common ground

---

The vision: *All digital preservation systems receive, store and provide access to information, regardless of its size, type or format, according to a set of agreed principles which allow institutions to identify, verify and validate the information in a uniform way.*

The goal: *Interoperability between data sources, archives and reuse environments is improved to a point where digital preservation tools can be reused across borders and institutions. This opens up new possibilities for collaboration and limits greatly the need for development resources for any single institution.*

The amount of digital information being created, held and exchanged is continuously growing. This information is created with the help of numerous software tools and systems, comes in a variety of technical formats, and covers most aspects of our daily lives. Regardless of the formats and systems in question we always need to consider whether the information is needed to be retained and managed for longer periods of time. The reasons for this might be, for example:

- to meet legal and regulatory obligations
- to provide for efficient reuse
- to satisfy historical, cultural, scientific and business interest.

As of now, most tools and systems used to create information are not built for coping with long-term requirements of keeping information safe and accessible. Instead, implementations separate the short- term and long-term management of information into different systems, for example business and records systems on one hand and archival systems on the other (Figure 5).

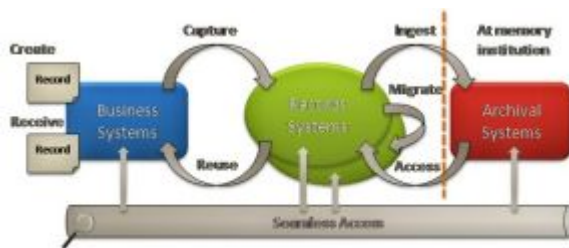


Figure 5: Information flow between live and archival systems

The implication for data owners and system managers is that information which has to be kept for extended time periods needs to be exchanged between a set of different locations, including archival systems:

- as effectively as possible,
- without endangering the authenticity and integrity of the information,
- and without limiting the possibilities for discovering and reusing the information.

As such, what we need in order to make the long-term availability of crucial information possible under (usually limited) resources is a set of principles which allow exchanging information in a common way across the systems participating in archival workflows and processes, i.e. create a set of interoperability specifications. For archival information packages we have identified the following interoperability scenarios (Figure 6):

- Export of data and metadata from source systems and transfer to SIP creation tools (or directly, as an SIP, into preservation systems);
- Transfer of SIPs from SIP creation tools to preservation systems;
- Exchange of preservation system where all AIPs need to be transferred (ingested) into a new technological platform;
- Distributed storage and synchronisation of AIPs between multiple (technologically different) preservation systems;
- Exchange of DIPs between preservation systems and access platforms or portals;
- Exchange of DIPs between various access platforms of portals.

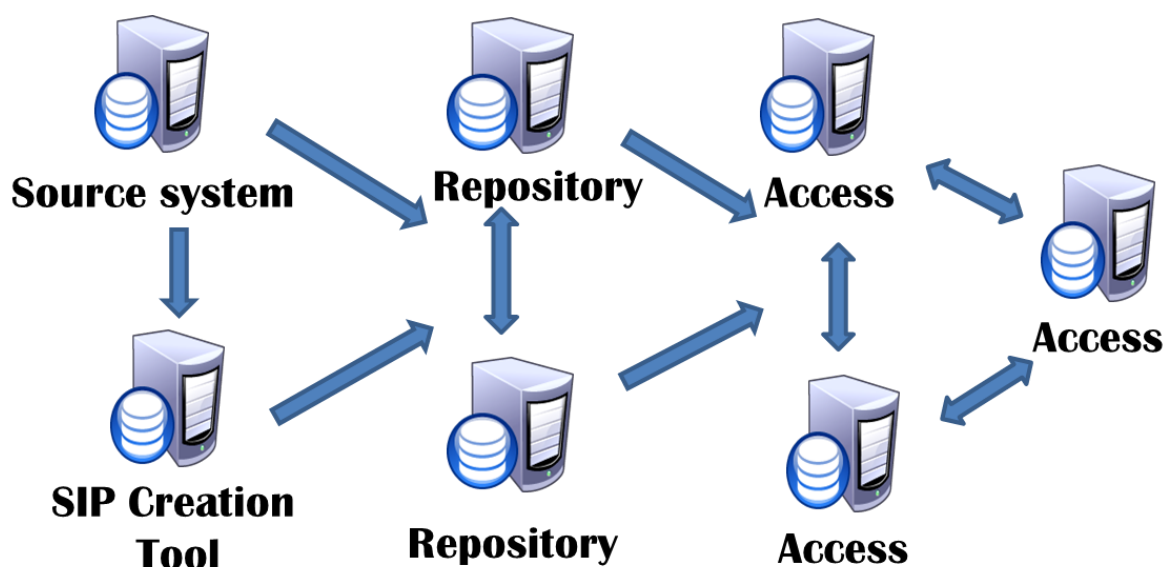


Figure 6: Archival workflow and tool ecosystem

As of 2014 (the start of the development of this specification) the state of interoperability in digital preservation was rather poor. While national and institutional practical implementation-level specifications existed to serve the need for data and metadata packaging and exchange, these were by large not interoperable with each other. On the contrary, available and widely used international specifications (most notably METS and PREMIS ) lack the necessary implementation-level detail, needed in order to serve as an authoritative source for practical interoperability.

This situation has a remarkable effect on the cost of digital preservation. Namely, the tools developed in individual institutions are not reusable across institutional and state borders and therefore need to be redeveloped at each single location. Globally, this raises the cost of digital preservation to a level which makes it not affordable for smaller institutions and, at the same time, does often not allow developing tools which would be sufficiently mature, user-friendly and prone to errors. As well, the multitude of national or institutional specifications does not allow internationally active source system providers (e.g. Oracle, Microsoft) to build a single native archiving functionality into their products, meaning that there is a need for bespoke development (and therefore added cost) for each installation of these source systems across all sectors and countries.

To overcome these limitations this document proposes a universal common specification, which can be implemented across borders, for how data and metadata should be structured and packaged when transferred to archival systems, ingested and preserved in these, and re-used. Such a specification will allow data owners to build standardised interfaces for the export of their data regardless of the archives in question; and digital archives to build standardised interfaces for data ingest and

access, regardless of the data providers and users in question.

Further, the aim of the common specification is to be sufficiently detailed and technical to allow for extended collaboration in regard to software development and pooling. Ideally the tools which implement the common specification for data export, transfer, ingest, preservation and reuse are exchangeable between institutions and administrations with minimal effort. This in turn shall lead to a significant decrease in resources needed from any single institution and at the same time opens up an extended market for commercial software providers.

## 3 Principles for interoperable Information Packages

---

At the heart of any standardisation activity has to be a clear understanding of the needs and aims which have to be addressed. This is also the goal of this Section, which presents a series of high level principles to guide the technical details delivered in Part II of this specification.

Most of the principles are driven by the aim of interoperability –Information Packages shall be easy to exchange, identify, validate and (re)use with a wide variety of software tools and systems.

Another crucial factor to take into account is long-term sustainability. Practical technical and semantic interoperability is possible only when a certain set of technologies have been agreed upon and implemented. However, any technology will become outdated sooner or later and previously agreed-upon approaches have to be updated to accommodate new, better and more efficient technologies and standards. Because of this, the developers of this Common Specification for Information Packages have reused, as much as possible, existing powerful, standardised and well-established best practices for the technical implementation of an Information Package (Part II of this document). This does not mean that the technical implementation details will not need to be changed in future, only that the need will arise later rather than sooner. To achieve long-term sustainability of the Common Specification for Information Packages, we present below a set of generic principles which must be followed when updating any specific implementation details at any point in time.

The principles present a conceptual view of an Information Package, including an overall IP data model, and use of data and metadata. An implementation of this conceptual view is presented later, in Part II of this document.

Each principle has a sequential number and a short description. The description includes always a MoSCoW (MUST/MUST NOT, SHOULD/SHOULD NOT, COULD, WOULD) prioritisation statement. The short description of each principle is followed by a rationale which describes the reason and background for the principle.

## 3.1 General principles

---

### Principle 1.1

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

This is one of the most crucial principles of the CSIP. In order to be truly “common”, technical implementations of the CSIP MUST NOT introduce limitations or restrictions which are only applicable to certain data or metadata types. If an Information Package implementation fails to meet this principle it is not possible to use it across different sectors and tools, thereby limiting practical interoperability.

### Principle 1.2:

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

Tools and methods for transferring Information Packages between locations are constantly evolving. It is also possible that different methods are preferred for packages of varying sizes. In order to achieve that a CSIP Information Package is truly interoperable across different platforms it therefore MUST NOT introduce limitations or restrictions which would be impossible to be met by specific information exchange tools or channels.

As such the CSIP does also not define the principle to use a particular transfer package or envelope. The scope of the CSIP is limited to the structure and requirements for data and metadata within the package. Different implementers are welcome to choose their own methods on top of the CSIP.

### Principle 1.3

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

One of the fundamental principles of the CSIP is that it MUST allow each individual repository to define the (intellectual) scope of an Information Package and its relations to real life entities. As such, any implementation of the CSIP MUST be equally usable for packaging, for example, the whole content of an ERMS as an single IP; or for extracting each record and its metadata from the ERMS individually and packaging each as a separate IP.

Out of the previous we can also derive that a CSIP specification **MUST NOT** define that, for example, a SIP should conform to exactly one AIP. Instead the CSIP **MUST** allow for the inclusion of “anything that the implementer wants to define as a SIP, AIP or DIP” and allow for “any relationships (1-1; 1-n; n-1; n-m) between SIPs, AIPs and DIPs”.

## **Principle 1.4:**

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

One of the practical concerns for Information Packages is their size. Many digital repositories have problems with data objects and metadata of increasing sizes, making it especially difficult to carry out tasks related to data or metadata validation, and identification and modification. For example, Information Packages including relational databases or born-digital 3D movies can easily reach TB sizes.

Consequently, any current or future implementation of the CSIP is required to provide for appropriate scalability mechanisms (for example: mechanisms for splitting large-scale data or metadata).

## **Principle 1.5:**

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

To support the goal of automating ingest, preservation and access workflows each of the implementations of the CSIP must be machine-actionable. This means that decisions about the use of metadata syntax and semantics as well as the physical structure must be expressed explicitly and in a clear way. This, in turn, allows the specification to be implemented in the same way across different tools and environments.

## **Principle 1.6:**

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

In long-term preservation we also need to take into account that “forgotten” Information Packages might be found long after details about the implementation are gone and no tools to access the package are available. For these scenarios it is crucial to ensure that the structure and metadata of the Information Package are understandable with minimal effort by using simple tools like text editors and file viewers.

In practice this means that any implementation of the CSIP should ensure that folder

and file naming conventions allow for the human identification of package components, and that the semantics of the package is explicit.

## Principle 1.7:

*The Information Package MUST support the preservation method best suited for the data.*

Different preservation institutions and different types of data need to use different methods for long-term preservation; migration and emulation being the most usual choices. A CSIP Information Package implementation MUST NOT prescribe the use of a specific preservation method but instead allow to document and/or add any data or metadata which is needed for any method.

## 3.2 Identification of the Information Package

---

### Principle 2.1:

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

One of the first tasks in analysing any Information Package is to identify its current status in the overall archival process. Therefore, any Information Package must explicitly and uniformly include metadata which identifies it as a SIP, AIP or DIP.

### Principle 2.2:

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

As stated in Principle 1.1 any Information Package MUST be able to include any kind of data and metadata. At the same time we have introduced in earlier Sections the concept of Content Information Types which allow users to achieve more detailed control and fine-grained interoperability. As such, any CSIP Information Package MUST include a statement about which Content Information Type specification(s) has been followed within the Information Package, or on the contrary, indicate clearly that no specific Content Information Type Specification has been followed.

The practical implication of principles 1.1, 2.1 and 2.2 is that, once these have been followed in implementations, we can in fact develop modular identification and validation tools and workflows. While generic components can carry out high level tasks regardless of the Content Information Type, it is possible to detect



automatically which additional content-aware modules need to be executed.

### **Principle 2.3:**

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

In order to manage a digital repository and provide access services each Information Package stored in the repository MUST be identified uniquely at least within the repository. At the same time a CSIP implementation MUST NOT limit the choice of the exact identification mechanism, as long as the mechanism is implemented consistently throughout the repository.

### **Principle 2.4:**

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

In addition to the previous principle, it is recommended that the identification mechanism used at the repository provides for global uniqueness and persistence of Information Package IDs. The application of globally unique and persistent identifiers allows repositories to participate more easily in cross-institutional information exchange and reuse scenarios (for example participation in national or international portals, or cross-repository duplication of AIP preservation). However, the CSIP MUST NOT limit the choice of the exact identification mechanism.

### **Principle 2.5:**

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

As stated above, a Information Package MUST be flexible enough to allow for the inclusion of any data or metadata depending on the needs of the repository and its users. As well, an Information Package might include additional support documentation like metadata schemas, user guidelines, contextual documentation etc. Regardless of which and how many components constitute a full Information Package, all components MUST have a unique and persistent identifier which allows for the appropriate linking of data, metadata and all other components. This, in turn, is one of the most crucial aspects towards achieving an interoperable way towards maintaining package integrity.

It is also worth mentioning that in any implementation it is only necessary to achieve identifier uniqueness and persistence within an individual Information Package. If this

is the case, repository-wide uniqueness is easily achieved when combining the package ID (unique according to principle 2.3) and the component ID.

The components of a Information Package are explained in more detail in the following section.

## 3.3 Structure of the Information Package

---

### Principle 3.1:

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

At the highest level each Information Package can be divided into data and metadata. In order to minimise the effort needed for the identification and validation of both, and to simplify long-term preservation actions it is reasonable to clearly separate data and metadata. This allows, for example, ingest tools to streamline and separate metadata identification and validation tasks, and file format identification and normalisation. Throughout long-term preservation such a separation allows also The most crucial (MUST) aspect of such separation is that it is achieved on the logical level of the Information Package.

### Principle 3.2:

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

In addition to the logical separation of components it is beneficial to have data and metadata physically separated (i.e. formatted as individual computer files or clearly separated bitstreams). This allows digital preservation tools and systems to update respective data or metadata portions of an Information Package without endangering the integrity of the whole package.

### Principle 3.3:

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

In addition to the previous principle it is recommended to explicitly divide metadata into more specific components. While the definitions of metadata types vary a lot between implementations it is our recommendation to divide metadata logically and

physically at least into descriptive and preservation metadata.

### **Principle 3.4:**

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

The concept of representations is one of the fundamental building blocks in digital preservation. As technologies evolve and get obsolete, data and metadata is constantly updated in order to ensure long-term accessibility, therefore creating new versions or representations of the data and metadata.

Expressing representations within the logical and physical structure of an Information Package helps institutions to explicitly understand the various states of the information throughout its lifecycle, therefore improving also the ease of long-term management and reuse of the information.

### **Principle 3.5:**

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

In addition to data and metadata, institutions might have the need to include additional information in an Information Package. For example, implementers might decide that XML Schemas about metadata structures and additional binary documentation about the original IT environment have to be added to the package.

If this is the case, the CSIP Information Package MUST NOT limit which components can constitute an Information Package, and MUST offer clearly defined extension points for the inclusion of these additional components into the Information Package. At the same time these extension points MUST be defined in a way which does not interfere with other components (i.e. the extension points MUST be clearly separated from other components of an Information Package).

### **Principle 3.6:**

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

Based on principles 3.1 – 3.4 we now present a common structure for any CSIP Information Package ([Figure 7](#)).

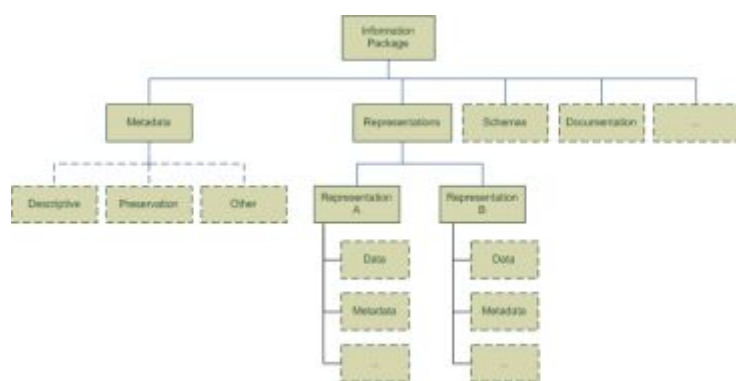


Figure 7: Conceptual structure of the Common Specification

Following Principle 3.4 the structure separates explicitly the representations of data and metadata into a separate structural component.

Following Principle 3.1 the package **MUST** include a high-level structural component for metadata which includes at least relevant metadata for the whole package. In addition the representations **MUST** internally separate between data and metadata (though note that the CSIP does not mandate that both data and metadata must be available in all representations).

In addition we highly recommend dividing the metadata portion of the Information Package to separate different types of metadata (SHOULD Principle 3.3).

Following Principle 3.5 repositories and their users have the possibility to add any additional components (as an example for schemas and binary support documentation) either as extensions to the whole Information Package or into a specific representation.

This common structure **MUST** be followed throughout all specific physical implementations of the CSIP.

## Principle 3.7:

*The Information Package **MUST** be implemented by **ONLY ONE** implementation at any point in time.*

The conceptual structure presented above can be implemented in various ways – for example the components might be defined by accompanying package metadata or explicitly through a physical structure. However, it is not reasonable to have multiple implementations available at once as this would lead to unnecessary complexity in developing interoperable tools for creating, processing and managing Information Packages. In CSIP the implementation at the time being, mandated to use is a fixed physical folder structure (see Section 4) as the implementation of this in combination

with the previous requirements.

At the same time it is clear that any given technical implementation will become obsolete in time, for example as new transfer methods and storage solutions emerge. As such this requirement does not prohibit the take-up of any emerging logical or physical technical solutions but merely requires to have one and only one of these to be implemented at any given point in time.

## 3.4 Information Package Metadata

---

### Principle 4.1:

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

In order to exchange, validate, process and reuse Information Packages in an interoperable and automated way we need to standardise how crucial metadata are presented in the package. “Crucial metadata”, is defined in this specification as the core information about how the package content has been created and managed (administrative and preservation metadata), explicit descriptions about of the structure of the package (structural metadata) and the technical details of the data themselves (technical metadata).

In order to ensure that these metadata are understood and implemented in a common and interoperable way in any Information Package, the use of established and widely used metadata standards is highly recommended. In the current implementation a large proportion of such metadata is covered by the widely used METS and PREMIS standards (see Section 5).

### Principle 4.2:

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

Many metadata standards support multiple options for describing specific details of an Information Package. However, such interpretation possibilities can also lead to different implementations and ultimately to the loss of interoperability.

To overcome this risk the CSIP requires that, while developing a specific implementation, the chosen metadata standard MUST be reviewed in regard to potential ambiguity. If needed, the selected metadata standard MUST be further refined to meet the needs of interoperability and automation.

## Principle 4.3:

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

Previous principles state the importance of controlled metadata for interoperability purposes. At the same time the opposite applies for other types of metadata, most prominently for resource discovery (also called descriptive) or Content Information Type specific technical and structural metadata. In order to not limit the widespread adoption of the CSIP it has to be possible for any implementer to add any metadata next to the mandatory metadata components needed for package level automation and interoperability.

In case organisations need to prescribe further details about descriptive or Content Information Type specific metadata for a deeper level of interoperability it is possible to use the mechanism of Content Information Type Specifications described above.

To summarise the requirements above from a more technical perspective, the CSIP foresees a modular approach towards Information Package metadata:

- All Information Packages share a common core of metadata which allows for the common development of high-level package creation, validation, identification and reuse tools;
- The rest of the metadata in the Information Package might follow additional agreements which have been made in order to develop specific tools such as, for example, tools to manage archival descriptions in EAD, or for specific Content Information Types like relational databases in the SIARD2 format.

## PART II: Implementation of the CSIP

---

In this part of the document we present an implementation of the requirements and principles presented in Part I of the specification for CS IP. The implementation consists of two core elements: a fixed physical structure of a CS IP Information Package (Section 4) and the exact use of metadata using the “Metadata Encoding & Transmission Standard” (METS) <http://www.loc.gov/standards/mets/> and “PREservation Metadata Implementation Strategies” (PREMIS) <http://www.loc.gov/standards/premis/> format (Section 5).

As explained earlier, any implementation using a metadata standard is will inevitably become obsolete. However, the authors have reused available best practices and established standards, and held discussions with the digital preservation community to ensure that the implementation is as future proof as possible.

### 4. CSIP structure

---

The preferred implementation of the conceptual model described in [Principle 3.6](#) is a fixed physical (folder) structure which follows exactly the conceptual structure. While the CS IP doesn't prohibited alternative implementations of the conceptual model such implementations aren't recommended.

The main reason for such an implementation decision is that a fixed physical folder structure makes it clear for both human users and tools where to find what. The main benefit of such a clear decision is that many archival tasks (for example file format risk analysis) can be executed directly on the data portion of the package structure, as opposed to first processing potentially large amounts of metadata for the locations of the files. This, in turn, allows for more efficient processing which is valuable in the case of large collections and bulk operations. In short, we believe that a fixed folder structure allows for more efficiency and scalability.

The authors of this specification are well aware that there are multiple data storage solutions which do not support explicit folder structures but use other means for structuring and storing (the content of) AIPs. However, we would like to note that the purpose of this specification is to support Information Package interoperability. As such we believe that even if a storage solution does not allow implementing the physical folder structure as the native AIP storage structure, it is still possible to implement the physical structure described below for SIPs, DIPs and the import/export of AIPs. While the repository needs to support an extra transformation (i.e. Common Specification IP to internal AIP and vice versa), it still allows the use of



tools created by other users of the common specification, easy transfer of AIPs to a new repository systems or storage solutions and to establish cross-repository duplicated storage solutions.

## 4.1. Folder structure of the CSIP

The CSIP folder structure is presented in Figure 8 below. The structure follows directly the principles of the conceptual data model by dividing the components of the package into stand-alone folders for representations, metadata, and other components. All folders described here are supposed to be present even if they are empty.



Figure 8: CSIP Information Package folder structure

The implementation requirements of the CSIP Information Package structure are:

**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.

**CSIPSTR2:** The Information Package root folder **SHOULD** be named with the ID or name of the Information Package.

**CSIPSTR3:** The Information Package root folder **CAN** be compressed (for example by using TAR or ZIP).

**CSIPSTR4:** The Information Package root folder **MUST** include a metadata file named `METS.xml`, which **MUST** include information about the identity and structure of the package and its components at a minimum down to a general description or pointer to each representation.

**CSIPSTR5:** The Information Package root folder **MUST** include a folder named `metadata`, which **SHOULD** include metadata relevant to the whole package.

**CSIPSTR6:** If preservation metadata are available, they **SHOULD** be included in



sub-folder `preservation`.

**CSIPSTR7:** If descriptive metadata are available, they SHOULD be included in sub-folder `descriptive`.

**CSIPSTR8:** If any other metadata are available, they CAN be included in separate sub-folders, for example an additional folder named `other`.

**CSIPSTR9:** The Information Package folder MUST include a folder named `representations`.

**CSIPSTR10:** The `representations` folder MUST include a sub-folder for each individual representation (i.e. the “representation folder”) named with a string uniquely identifying the representation within the scope of the package (for example the name of the representation and/or its creation date could be good examples for an representation sub-folder).

**CSIPSTR11:** The representation folder MUST include a sub-folder named `data` which includes all data constituting the representation.

**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.

**CSIPSTR13:** The representation folder MUST include a sub-folder named `metadata` which CAN include all metadata about the specific representation.

**CSIPSTR14:** The Information Package folder and representation folder CAN be extended with additional sub-folders.

**CSIPSTR15:** We recommend including all 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.

**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.

## 4.2. Implementing the structure

The requirements presented in Section 4.1 leave room for decisions during the implementation. For the sake of clarity we provide examples for two extremes – the simplest and the full use of the structure.

In the simplest case the structure can be implemented following mostly just the MUST requirements. An example of this is visible on Figure 9.

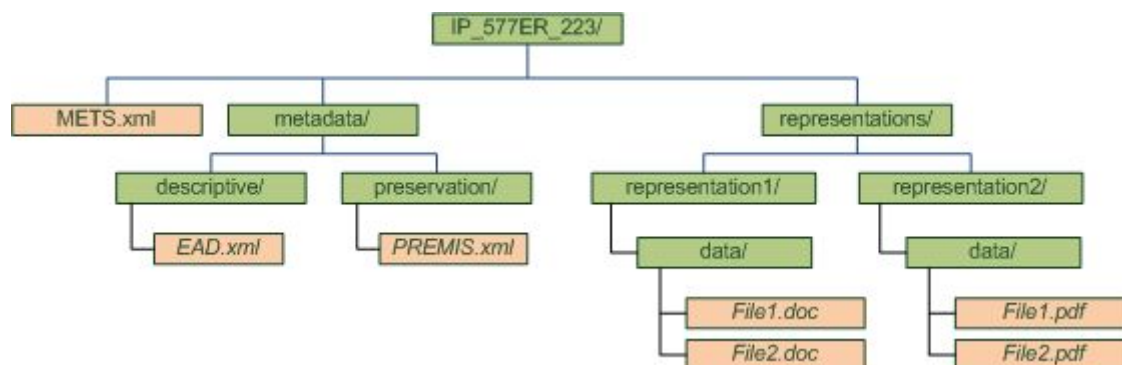


Figure 9: Example of a simple use of the CSIP structure

The main point to highlight with such a simple use is that the representations have been kept as simple as possible. All metadata about both the package and the representations (in this example METS, EAD and PREMIS metadata) are located in the Information Package folder and none of these components are available within the representation folders.

Such a simple implementation is reasonable in scenarios where the amount of data and metadata is limited. However, in the case of large Information Packages (for example, a package including three representations and 1,000,000 files in one representation) the size of both the `METS.xml` file and preservation metadata can grow too large to manage efficiently. Especially in such large data scenarios it might prove necessary to implement all the capabilities of the structure presented in the previous Section.

An example of the full implementation is delivered in Figure 10. The main difference between the simple and full use of the structure is that each representation does essentially repeat the simple structure. Especially structural and preservation metadata in METS and PREMIS formats is available in both the Information Package folder (for package level descriptions) and within representation folders (for representation level descriptions). As such the full structure allows for easier management of single representations and brings further benefits like more straight-forward metadata versioning. It is worth to note that, in order to avoid

confusion, it is recommended to have a common approach towards adding metadata into representations or not. In other words, we recommend having all representation-relevant metadata either in the root metadata folder or the representation metadata folder, but not to have a mixed approach (i.e. some representation metadata in the root metadata folder and some within the representation). Further, we do not recommend the duplication of any metadata or the content of optional folders (schemas, documentation, etc.) between the Information Package folder and representation folders.

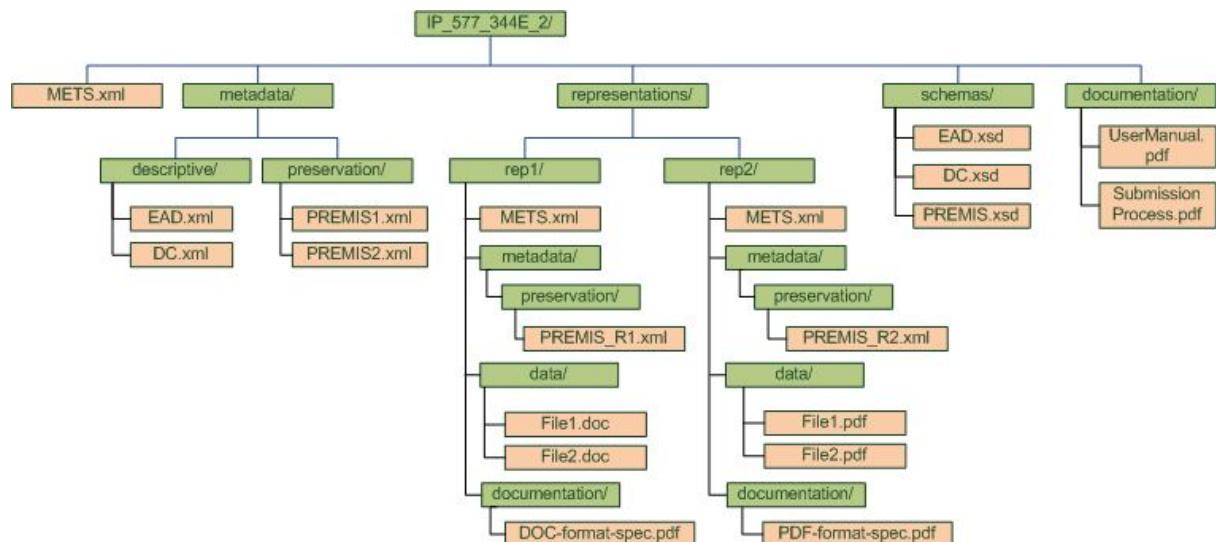


Figure 10: Example of the full use of the CSIP structure

## 5. Use of metadata

---

### 5.1. General requirements for metadata in a CS IP Information Package

---

The number one consideration when discussing metadata requirements is, as with the rest of this specification, the need for interoperability. In more detail, the focus is on high-level technical interoperability and tasks which allow an Information Package to be prepared, transferred and received regardless of the institutions and tools involved. These tasks include:

- Identifying uniquely an Information Package and its components;
- Validating an Information Package;
- Validating the contents of an Information Package;
- Proving the authenticity of the Information Package;
- Accessing the contents of an Information Package.

In more technical terms the CS IP makes an effort to control metadata which allows any tool or user to negotiate the data and metadata components of the package (i.e. packaging metadata), to validate that no component has come to harm during transfer or preservation (i.e. fixity information), to understand the processes behind the creation and management of the package (i.e. provenance and preservation metadata) and finally to understand how the data within the package could be accessed (i.e. representation information).

Most crucially, we regard descriptive metadata and most of detailed technical metadata to not belong in the scope of the CS IP. As such, the CS IP itself does not aim to provide detailed semantic interoperability between different systems. However, as noted in Section 1.2, implementers are welcome to use the construct of Content Information Type Specifications to achieve an even higher level of interoperability.

We implement the core metadata requirements with METS (Metadata Encoding & Transmission Standard, <http://www.loc.gov/standards/mets/> ). In this specification we describe the core elements used, more elements are available in the METS standard and can be used in the own implementation.

Some of the core metadata requirements are already visible from the structure presented in the previous Section. As seen in the previous section one or more METS files can be present. The METS file describing the whole package is from now called “Root METS” and the METS file present in the Representation folder is called

“Representation METS” in the rest of this document. The detailed specification of using METS within the CS IP is available in Section 5.3.

In addition to the METS files the CS IP recommends the inclusion of PREMIS metadata (PREservation Metadata Implementation Strategies, <http://www.loc.gov/standards/premis/> ) in appropriate preservation metadata folders. This is especially relevant when aiming for an interoperable approach towards provenance and access to Information Packages. However, we recognise that, especially in the case of SIPs, appropriate preservation metadata is not always available. As such this is also not an absolute requirement though highly desirable. The detailed specification of the use of PREMIS within the CS IP is available in Section 5.4.

The use of any additional metadata is not restricted in CS IP Information Packages.

## 5.2 General requirements for the use of metadata

---

Before we describe the detailed requirements for the use of METS and PREMIS we would like to highlight some general aspects which need to be implemented commonly across all metadata.

### The use of identifiers

The ID data type in XML ( <https://www.w3.org/TR/xml-id/> ) states that a valid ID must begin with a letter, or an underscore character ('\_'), and contain no characters other than letters, digits, hyphens, underscores, full stops, and certain combining and extension characters. To overcome this limitation and allow for interoperable package identification all identifiers within Common Specification metadata MUST start with a prefix, followed by the value of the identifier.

#### Examples:

Example 1: using a prefix which consists of the abbreviation of the identifier and a hyphen.

```
<dmdSec ID="uuid-906F4F12-BA52-4779-AE2C-178F9206111F" CREATED="2018-04-24T14:37:49.609+01:00">
```

Example 2: using a fixed prefix “ID”

```
<dmdSec ID="ID906F4F12-BA52-4779-AE2C-178F9206111F" CREATED="2018-04-24T14:37:49.609+01:00">
```

Note that identifier-type elements and attributes specified within the CS IP are mainly used for internal referencing between the components of an Information Package. As

such there is no need to require the use of any specific prefix syntax but it is required that any selected prefix is used consistently throughout the package.

## Referencing between files within a CS IP Information Package

This specification recommends strongly to format all components of the information package (i.e. all data, metadata and other parts) as distinct computer files within the package. While such an approach simplifies the overall management of the Information Package and makes it easier to include, validate and modify the package, it also brings the need for a clear method for referencing between these various files.

For example, when using the CS IP utilized with the METS specification referencing can occur to and between:

- descriptive and administrative metadata files described in the `amdSec` and `dmdSec` elements;
- content (data) files, components of documentation and schemas described in the `fileSec` element;
- representation METS files described in the root METS file's `fileSecas` as well as the `structMap` element.

A common approach towards referencing between metadata, and between metadata and other components of the package, is one of the core needs in Information Package validation and integrity checking. Different technical solutions are available for referencing and not all of these are supported across all digital preservation tools.

In order to guarantee interoperability, all references within a CS IP Information Package must follow the requirements stated in this specification.

## Referencing other packages

It is important that external references to related packages, like internal references, are expressed consistently. All external references MUST USE `mets/@OBJID` attribute of the package.

## 5.3 Use of METS

---

The main requirement for METS files in a CSIP Information Package is that these need to follow the official METS Schema version 1.12 (by CSIP used version in November 2018) and the extension schema developed for the CSIP and published by the DILCIS Board. As new versions of METS Schema become available the DILCIS Board will evaluate these and, if necessary, update the CSIP respectively.

The following text assumes knowledge of the principles of the METS specification. If this is not the case, please consult the official documentation before continuing.

METS allows metadata to be both embedded and referenced. The CSIP itself allows both the embedding of metadata within the METS.xml file but note that for scalability concerns the CSIP only recommends the use of referencing. This means that the CSIP only describes referencing of metadata.

The rest of this Section is structured according to the METS elements: `mets`, `header`, `dmdSec`, `amdSec`, `fileSec` and `structMap`. In each of these sections we explain in a concise way limitations imposed by the CSIP implementation when compared to the official METS documentation. When an implementation of the CSIP is created a choice can be made to extend the limitations with limitations needed by the implementation. If this is the case follow the official documentation and create an implementation which base is CSIP.

Differences between creating a root METS file and representation METS file are described when relevant.

All names of elements and attributes below are expressed using the XPath notation (i.e. `element/sub-element/@attribute`)

### 5.3.1. Use of the METS root element (element `mets`)

The purpose of the METS root element is to describe the container for the information being stored and/or transmitted. The implementation of the root element for a METS document conformant with CSIP uses attributes from the METS specification and attributes added for the purposes of the CSIP.

In addition to the attributes the METS root element `mets` MUST define all relevant namespaces and locations of XML schemas using the `@xmlns` and `@xsi:schemaLocation` attributes.

In case XML schemas have been included into the package (i.e. placed into the `schemas` folder) it is recommended to link to the schemas using the relative path of the

schema file (i.e. `schemas/mets.xsd`).

The specific requirements for the root element and its attributes are described in the following table .

ID	Name & Location	Description & usage	Cardinality & Level
CSIP 1	Content Identification  <code>mets/@OBJID</code>	It is mandatory to use a content ID which is expressed with <code>@OBJID</code> . The value should be the same as the name or ID of the package (the name of the root folder) for the root METS document or the name and folder name for the representation. The OBJID must meet the principle of being unique at least across the repository.	1..1  MUST
CSIP 2	General content type  <code>mets/@TYPE</code>	The <code>@TYPE</code> attribute must be used for identifying the general type of the package (genre). A vocabulary is used. The vocabulary is going to evolve under the care of the DILCIS Board as additional content information type specifications are developed.  See also: <a href="#">Content information type declaration</a>	1..1  MUST
CSIP 3	Other general content type  <code>mets/@csip:OTHERTYPE</code>	The <code>@csip:OTHERTYPE</code> attribute must be used for stating the general type of the package (genre) when <code>@TYPE</code> has the value	0..1  SHOULD



		<p>“OTHER”</p> <p>See also: &lt;ul&gt; &lt;li&gt; <a href="#">Content information type declaration</a>&lt;/li&gt; &lt;/ul&gt;</p>	
CSIP 4	<p>Specific content type</p> <p><code>mets/@csip:CONTENTINFORMATIONTYPE</code></p>	<p>An added attribute which describes the specific content information type specification used for the transferred content. The attribute is mandatory to use when the METS document describes a representation. The vocabulary is going to evolve under the care of the DILCIS Board as additional content information type specifications are developed.</p> <p>See also: &lt;ul&gt; &lt;li&gt; <a href="#">Content information type specification name</a> &lt;/li&gt; &lt;/ul&gt;</p>	<p>1..1</p> <p>SHOULD</p>
CSIP 5	<p>Other specific content type</p> <p><code>mets/@csip:OTHERCONTENTINFORMATIONTYPE</code></p>	<p>When the <code>@csip:CONTENTINFORMATIONTYPE</code> uses the value “OTHER” the <code>@csip:OTHERCONTENTINFORMATIONTYPE</code> must describe the content.</p>	<p>0..1</p> <p>MAY</p>
CSIP 6	<p>METS Profile</p> <p><code>mets/@PROFILE</code></p>	<p>The PROFILE attribute has to have as its value the URL of the profile used for describing the package.</p>	<p>1..1</p> <p>MUST</p>

**Example:** METS root element example with content not in the value list being transferred

```
<mets:mets OBJID="uuid-4422c185-5407-4918-83b1-7abfa77de182" LABEL="Sample CSIP
Information Package" TYPE="OTHER" OTHERTYPE="Patterns"
PROFILE="https://earkcsip.dilcis.eu/profile/CSIP.xml"
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://dilcis.eu/XML/METS/CSIPExtensionMETS/DILCISExtensionMETS.xsd">
</mets:mets>
```

**Example:** METS root element example of content with a type from the value list and representation content not in the value list

```
<mets:mets OBJID="uuid-4422c185-5407-4918-83b1-7abfa77de182" LABEL="Sample CSIP
Information Package" TYPE="Datasets" CONTENTINFORMATIONTYPE="OTHER"
OTHERCONTENTINFORMATIONTYPE="FGS Personal, version 1"
PROFILE="https://earkcsip.dilcis.eu/profile/CSIP.xml"
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://dilcis.eu/XML/METS/CSIPExtensionMETS/DILCISExtensionMETS.xsd">
</mets:mets>
```

### 5.3.2. Use of the METS header (element metsHdr)

The purpose of the METS header section is to describe the METS document itself, for example information about the creator of the IP. The requirements for the metsHdr element, its sub-elements and attributes are presented in the following table.

ID	Name & Location	Description & usage	Cardinality & Level
CSIP 7	Package creation date  metsHdr/@CREATEDATE	@CREATEDATE describes the date of creation of the package.	1..1  MUST
CSIP 8	Package last modification date  metsHdr/@LASTMODDATE	@LASTMODDATE is mandatory if the package has been modified.	0..1  SHOULD
CSIP 9	OAIS Package type information  metsHdr/@csip:OAISPAC KAGETYPE	@csip:OAISPAC KAGETYPE is an attribute added by the CSIP for describing the type of the IP.  See also:  <ul><li> <a href="#">OAIS Package type</a> </li></ul>	1..1  MUST
CSIP 10	Agent  metsHdr/agent	One mandatory agent is used to describe the software used for creating the package. Other uses of agents are described in the own implementations extending profile.	1..n  MUST
CSIP 11	Agent role  metsHdr/agent/@ROLE	The role of the mandatory agent is “CREATOR”.	1..1  MUST

CSIP 12	Agent type  metsHdr/agent/@TYPE	The type of the mandatory agent is "OTHER".	1..1  MUST
CSIP 13	Agent other type  metsHdr/agent/@OTHERTYPE	The other type of the mandatory agent is "SOFTWARE".  See also: <ul> <li> <a href="#">Other agent type</a> </li> </ul>	1..1  MUST
CSIP 14	Agent name  metsHdr/agent/name	The name of the mandatory agent is the name of the software tool which was used to create the IP.	1..1  MUST
CSIP 15	Agent additional information  metsHdr/agent/note	The mandatory agent has a note providing the version information for the tool which was used to create the IP.	1..1  MUST
CSIP 16	Classification of the agent additional information  metsHdr/agent/note/@csip:NOTETYPE	The mandatory agent note is typed with the fixed value of "SOFTWARE VERSION".  See also: <ul> <li> <a href="#">Note type</a> </li> </ul>	1..1  MUST

Example: METS example of the mandatory agent

```
<mets:metsHdr CREATEDATE="2018-04-24T14:37:49.602+01:00"
LASTMODDATE="2018-04-24T14:37:49.602+01:00" RECORDSTATUS="NEW"
OAISPACKAGETYPE="SIP">
  <mets:agent ROLE="CREATOR" TYPE="OTHER" OTHERTYPE="SOFTWARE">
    <mets:name>RODA-in</mets:name>
    <mets:note NOTETYPE="SOFTWARE VERSION">2.1.0-beta.7</mets:note>
  </mets:agent>
</mets:metsHdr>
```

### 5.3.3 Use of the METS descriptive metadata section (element dmdSec)

The purpose of the METS descriptive data section is to embed or refer to files containing descriptive metadata. CSIP is only using referencing of files containing descriptive metadata.

The CSIP as such does not make any assumptions on the use of specific descriptive metadata schemas. As such, implementers are welcome to use descriptive metadata following any standards inside a CS IP package.

Specific elements for which the exact use is fixed within this specification are highlighted in the following table.

ID	Name & Location	Description & usage	Cardinality & Level
CSIP 17	Descriptive metadata  dmdSec	Must be used if descriptive metadata for the package content is available. Each descriptive metadata section (dmdSec) contains one description and thus is repeated when more descriptions are available.  It is possible to transfer metadata in a package using just the descriptive metadata section/or administrative metadata section.	0..n  SHOULD
CSIP 18	Descriptive metadata identifier  dmdSec/@ID	An identifier for the descriptive metadata section (dmdSec) used for referencing inside the package. It must be unique within the package.  The ID must follow the rules for xml:id described in the chapter of the textual description of CSIP named "General requirements for the use of metadata"	1..1  MUST

CSIP 19	Descriptive metadata creation date  dmdSec/@CREATED	Creation date of the descriptive metadata in this section.	1..1  MUST
CSIP 20	Status of the descriptive metadata  dmdSec/@STATUS	Status of the metadata. Used to indicate the currency of the package. If used the two values “SUPERSEDED” or “CURRENT” from the vocabulary is used.  See also: <ul> <li> <a href="#">dmdSec status</a> </li> </ul>	0..1  SHOULD
CSIP 21	Reference to the document with the descriptive metadata  dmdSec/mdRef	Reference to the descriptive metadata file located in the “metadata” section of the IP.	0..1  SHOULD
CSIP 22	Type of locator  dmdSec/mdRef/@L OCTYPE	The locator type is always used with the value “URL” from the vocabulary in the attribute.	1..1  MUST
CSIP 23	Type of link  dmdSec/mdRef/@x link:type	Attribute used with the value “simple”. Value list is maintained by the xlink standard	1..1  MUST
CSIP 24	Resource location  dmdSec/mdRef/@x link:href	The actual location of the resource. This specification recommends recording a URL type filepath within this attribute.	1..1  MUST
CSIP 25	Type of metadata	Specifies the type of metadata in the linked file. Values are taken from the	1..1

	dmdSec/mdRef/@MDTYPE	list provided by the standard.	MUST
CSIP 26	File mime type  dmdSec/mdRef/@MIMETYPE	The IANA mime type for the linked file.  See also: <ul> <li> <a href="#">IANA media types</a> </li> </ul>	1..1  MUST
CSIP 27	File size  dmdSec/mdRef/@SIZE	Size of the linked file in bytes.	1..1  MUST
CSIP 28	File creation date  dmdSec/mdRef/@CREATED	The date the linked file was created.	1..1  MUST
CSIP 29	File checksum  dmdSec/mdRef/@CHECKSUM	The checksum of the linked file.	1..1  MUST
CSIP 30	File checksum type  dmdSec/mdRef/@CHECKSUMTYPE	The type of checksum following the value list in the standard which used for the linked file.	1..1  MUST

Example: METS example of referencing descriptive metadata in the for of an EAD document

```
<mets:dmdSec ID="uuid-906F4F12-BA52-4779-AE2C-178F9206111F"
CREATED="2018-04-24T14:37:49.609+01:00">
  <mets:mdRef LOCTYPE="URL" MDTYPE="EAD" type="simple"
href="metadata/descriptive/ead2002.xml" mimetype="application/xml" SIZE="903"
CREATED="2018-04-24T14:37:49.609+01:00"
CHECKSUM="F24263BF09994749F335E1664DCE0086DB6DCA323FDB6996938BCD28EA9E8153"
CHECKSUMTYPE="SHA-256">
    </mets:mdRef>
  </mets:dmdSec>
```

### 5.3.4. Use of the METS administrative metadata section (element amdSec)

The purpose of the METS administrative data section is to embed or refer to files containing administrative metadata about the IP content objects. CSIP is only using referencing of files containing administrative metadata. The CSIP (and METS) categorises preservation metadata as administrative metadata, specifically Digital Provenance metadata (following the available guidelines), hence all preservation metadata should be referenced from a digiprovMD element within the amdSec.

The METS amdSec element must include references to all relevant metadata located in the folder “metadata/preservation”. This means also that the root level METS.xml file must refer only to the root level preservation metadata and the representation METS.xml file must refer only to the representation level preservation metadata.

Decision regarding placement of PREMIS in this section is following the guide lines available from PREMIS EC

<http://www.loc.gov/standards/premis/guidelines2017-premismets.pdf>.

The specific requirements for the amdSec element, its sub-elements and attributes are presented in the following table.

ID	Name & Location	Description & usage	Cardinality & Level
CSIP 31	Administrative metadata  amdSec	If administrative / preservation metadata is available, it must be described using the administrative metadata section (amdSec) element.  It is possible to transfer metadata in a package using just the descriptive metadata section/or administrative metadata section.	0..n  SHOULD



CSIP 32	Digital provenance metadata  amdSec/digiprovMD	For recording information about preservation events the standard PREMIS is used. The PREMIS metadata must be either embedded or linked in a digital provenance metadata (digiprovMD) element. It is mandatory to include one digiprovMD element for each external PREMIS file placed in the “metadata/preservation” section, or for each embedded set of PREMIS metadata.	0..n  SHOULD
CSIP 33	Digital provenance metadata identifier  amdSec/digiprovMD/@ID	An identifier for the digital provenance metadata section (digiprovMD) used for referencing inside the package. It must be unique within the package.  The ID must follow the rules for xml:id described in the chapter of the textual description of CSIP named “General requirements for the use of metadata”	1..1  MUST
CSIP 34	Status of the digital provenance metadata  amdSec/digiprovMD/@STATUS	Status of the metadata. Used to indicate the currency of the package. If used the two values “SUPERSEDED” or “CURRENT” from the vocabulary is used.  See also:  <ul><li> <a href="#">dmdSec status</a> </li></ul>	0..1  SHOULD

CSIP 35	Reference to the document with the digital provenance metadata  amdSec/digiprovMD/mdRef	Reference to the digital provenance metadata file stored in the “metadata” section of the IP.	0..1  SHOULD
CSIP 36	Type of locator  amdSec/digiprovMD/mdRef /@LOCTYPE	The locator type is always used with the value “URL” from the vocabulary in the attribute.	1..1  MUST
CSIP 37	Type of link  amdSec/digiprovMD/mdRef /@xlink:type	Attribute used with the value “simple”. Value list is maintained by the xlink standard	1..1  MUST
CSIP 38	Resource location  amdSec/digiprovMD/mdRef /@xlink:href	The actual location of the resource. This specification recommends recording a URL type filepath within this attribute.	1..1  MUST
CSIP 39	Type of metadata  amdSec/digiprovMD/mdRef /@MDTYPE	Specifies the type of metadata in the linked file. Values are taken from the list provided by the standard.	1..1  MUST
CSIP 40	File mime type  amdSec/digiprovMD/mdRef /@MIMETYPE	The IANA mime type for the linked file.  See also: <ul> <li> <a href="#">IANA media types</a> </li> </ul>	1..1  MUST
CSIP 41	File size  amdSec/digiprovMD/mdRef /@SIZE	Size of the linked file in bytes.	1..1  MUST

CSIP 42	File creation date  amdSec/digiprovMD/mdRef /@CREATED	Date the linked file was created.	1..1  MUST
CSIP 43	File checksum  amdSec/digiprovMD/mdRef /@CHECKSUM	The checksum of the linked file.	1..1  MUST
CSIP 44	File checksum type  amdSec/digiprovMD/mdRef /@CHECKSUMTYPE	The type of checksum following the value list in the standard which used for the linked file.	1..1  MUST
CSIP 45	Rights metadata  amdSec/rightsMD	For describing an overall access status for the package a simple rights statement may be used.  as well as own local rights statements in use.	0..1  MAY
CSIP 46	Rights metadata identifier  amdSec/rightsMD/@ID	An identifier for the rights metadata section (rightsMD) used for referencing inside the package. It must be unique within the package.  The ID must follow the rules for xml:id described in the chapter of the textual description of CSIP named "General requirements for the use of metadata"	1..1  MUST
CSIP 47	Status of the rights metadata  >amdSec/rightsMD/@STATUS	Status of the metadata. Used to indicate the currency of the package. If used the two values "SUPERSEDED" or "CURRENT" from the	0..1  SHOULD

		<p>vocabulary is used.</p> <p>See also:</p> <ul style="list-style-type: none"> <li><a href="#">dmdSec status</a></li> </ul>	
CSIP 48	<p>Reference to the document with the rights metadata</p> <p><code>amdSec/rightsMD/mdRef</code></p>	<p>Reference to the rights metadata file stored in the “metadata” section of the IP.</p>	<p>0..1</p> <p>SHOULD</p>
CSIP 49	<p>Type of locator</p> <p><code>amdSec/rightsMD/mdRef/@LOCTYPE</code></p>	<p>The locator type is always used with the value “URL” from the vocabulary in the attribute.</p>	<p>1..1</p> <p>MUST</p>
CSIP 50	<p><code>amdSec/rightsMD/mdRef/@xlink:type</code></p>	<p>Attribute used with the value “simple”. Value list is maintained by the xlink standard</p>	<p>1..1</p> <p>MUST</p>
CSIP 51	<p>Resource location</p> <p><code>amdSec/rightsMD/mdRef/@xlink:href</code></p>	<p>The actual location of the resource. We recommend recording a URL type filepath within this attribute.</p>	<p>1..1</p> <p>MUST</p>
CSIP 52	<p>Type of metadata</p> <p><code>amdSec/rightsMD/mdRef/@MDTYPE</code></p>	<p>Specifies the type of metadata in the linked file. Value is taken from the list provided by the standard.</p>	<p>1..1</p> <p>MUST</p>
CSIP 53	<p>File mime type</p> <p><code>amdSec/rightsMD/mdRef/@MIMETYPE</code></p>	<p>The IANA mime type for the linked file.</p> <p>See also: <ul style="list-style-type: none"> <li><a href="#">IANA media types</a></li> </ul></p>	<p>1..1</p> <p>MUST</p>

CSIP 54	File size  amdSec/rightsMD/mdRef/@ SIZE	Size of the linked file in bytes.	1..1  MUST
CSIP 55	File creation date  amdSec/rightsMD/mdRef/@ CREATED	Date the linked file was created.	1..1  MUST
CSIP 56	File checksum  amdSec/rightsMD/mdRef/@ CHECKSUM	The checksum of the linked file.	1..1  MUST
CSIP 57	File checksum type  amdSec/rightsMD/mdRef/@ CHECKSUMTYPE	The type of checksum following the value list in the standard which used for the linked file.	1..1  MUST

Example: METS example of referencing preservation metadata in the form of PREMIS metadata for describing the preservation objects and the events pertaining to the objects

```
<mets:amdSec>
  <mets:digiprovMD ID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F"
    CREATED="2018-04-24T14:37:52.783+01:00">
    <mets:mdRef LOCTYPE="URL" type="simple"
      href="metadata/preservation/premis1.xml" MDTYPE="PREMIS:EVENT" MDTYPEVERSION="3.0"
      MIMETYPE="text/xml" SIZE="1211" CREATED="2018-04-24T14:37:52.783+01:00"
      CHECKSUM="8aa278038dbad54bbf142e7d72b493e2598a94946ea1304dc82a79c6b4bac3d5"
      CHECKSUMTYPE="SHA-256" LABEL="premis1.xml">
    </mets:mdRef>
  </mets:digiprovMD>
  <mets:digiprovMD ID="uuid-48C18DD8-2561-4315-AC39-F941CBB138B3"
    CREATED="2018-04-24T14:47:52.783+01:00">
    <mets:mdRef LOCTYPE="URL" type="simple"
      href="metadata/preservation/premis2.xml" MDTYPE="PREMIS:OBJECT"
      MDTYPEVERSION="3.0" MIMETYPE="text/xml" SIZE="2854"
      CREATED="2018-04-24T14:37:52.783+01:00"
      CHECKSUM="d1dfa585dcc9d87268069dc58d5e47956434ec3db4087a75a3885d287f15126f">
```

```
CHECKSUMTYPE="SHA-256" LABEL="premis2.xml">
  </mets:mdRef>
</mets:digiprovMD>
</mets:amdSec>
```

### 5.3.5. Use of the METS file section (element fileSec)

Use of the METS fileSec element is highly recommended by the CSIP (although not mandatory). It should describe all components of the IP which have not been already included in the amdSec and dmdSec elements. For all files the location and checksum need to be available. Therefore the main purpose of the METS file section is to serve as a “table of contents” or “manifest” and allow validating the integrity of the files included into the package.

The main requirement of the CSIP is that the file section of both the root and representation METS files includes at least one file group (element fileGrp) grouping together files. CSIP structures the different files into structural components (i.e. documentation, schemas, data) which are described by its own fileGrp element. Representations including their own METS files, the components (including data files) of a representation should be described only in the representation METS. The root METS file should still include a fileGrp for each representation but only reference the METS.xml file of the representation.

The specific requirements for elements, sub-elements and attributes are listed in the following table. Note that use of the stream and transformFile elements are not discussed below. Implementers wishing to use either of these METS elements should follow the requirements in the METS documentation.

ID	Name & Location	Description & usage	Cardinality & Level
CSIP 58	File section  fileSec	<p>When the section is used only one file section (fileSec) element is present.</p> <p>It is possible to transfer just descriptive metadata and/or administrative metadata without files placed in this section.</p>	0..1  SHOULD

CSIP 59	File section identifier  fileSec/@ID	<p>An identifier for the file section used for referencing inside the package. It must be unique within the package.</p> <p>The ID must follow the rules for xml:id described in the chapter of the textual description of CSIP named “General requirements for the use of metadata”</p>	1..1  MUST
CSIP 60	File grouping  fileSec/fileGrp	<p>There are one or more file group (fileGrp) elements present grouping the transfered files in the main categorization of; Documentation, Schemas and Representations.</p> <p>In one or more file groups with the categorization of “Documentation” all documentation pertaining to the transfered information is present.</p> <p>In one or more file groups with the categorization of “Schemas” all XML-schemas pertaining to the transfered XML documents is present.</p> <p>In one or more file groups with the categorization of “Representations” the data being transfered is present or in one file group the data for each representation is present.</p>	1..n  MUST

		<p>To make the catagorization easier the different files being transfered should be placed in folders with names folowing the catagorization</p> <p>See also:</p> <ul style="list-style-type: none"> <li><a href="#">File group names</a></li> </ul>	
CSIP 61	<p>Reference to administrative metadata</p> <p>fileSec/fileGrp/@ADMID</p>	<p>If administrative metadata is has been provided on the file group (fileGrp) level this attribute points to the correct administrative metadata section.</p>	<p>0..1</p> <p>MAY</p>
CSIP 62	<p>Specific content type</p> <p>fileSec/fileGrp/@csip:CONTENTINFORMATIONTYPE</p>	<p>An added attribute which describes the specific content information type specification used for the transferred content. The attribute is mandatory to use when the file group catagorization is Representations. The vocabulary is going to evolve under the care of the DILCIS Board as additional content information type specifications are developed.</p> <p>See also: <ul style="list-style-type: none"> <li><a href="#">Content information type specification name</a></li> </ul></p>	<p>1..1</p> <p>SHOULD</p>
CSIP 63	<p>Other specific content type</p> <p>fileSec/fileGrp/@csip:</p>	<p>When the @csip:CONTENTINFORMATIONTYPE uses the value "OTHER" the</p>	<p>0..1</p> <p>MAY</p>



	OTHERCONTENTINFORMATIONTYPE	@csip:OTHERCONTENTINFORMATIONTYPE must describe the content.	
CSIP 64	Description of the use of the file group  fileSec/fileGrp/@USE	The value in the @USE is the name of the whole folder structure to the data, e.g “Documentation”, “Schemas”, “Representations/preingest” or “Representations/submission/data”	1..1  MUST
CSIP 65	File group identifier  fileSec/fileGrp/@ID	An identifier for the file group used for referencing inside the package. It must be unique within the package.  The ID must follow the rules for xml:id described in the chapter of the textual description of CSIP named “General requirements for the use of metadata”	1..1  MUST
CSIP 66	File  fileSec/fileGrp/file	The lowest level file group (fileGrp) contains the file elements which describe the transferred file objects.  When the file element is categorised as “Representations” each representation file group contains one file which is the reference to the METS document describing the representation	1..1  MUST

CSIP 67	File identifier  fileSec/fileGrp/file/@ ID	A unique identifier for this file across the package.  The ID must follow the rules for xml:id described in the chapter of the textual description of CSIP named “General requirements for the use of metadata”	1..1  MUST
CSIP 68	File mimetype  fileSec/fileGrp/file/@ MIMETYPE	The IANA mime type for the linked file.  See also:  <ul><li> <a href="#">IANA media types</a> </li></ul>	1..1  MUST
CSIP 69	File size  fileSec/fileGrp/file/@ SIZE	Size of the linked file in bytes.	1..1  MUST
CSIP 70	File creation date  fileSec/fileGrp/file/@ CREATED	Date the linked file was created.	1..1  MUST
CSIP 71	File checksum	The checksum of the linked file.	1..1  MUST
CSIP 72	File checksum type  fileSec/fileGrp/file/@ CHECKSUMTYPE	The type of checksum following the value list in the standard which used for the linked file.	1..1  MUST
CSIP 73	File original identification  fileSec/fileGrp/file/@	If an original ID for the file has been given by the owner it can be saved in this	0..1  MAY

	OWNERID	attribute.	
CSIP 74	File reference to administrative metadata  fileSec/fileGrp/file/@ADMID	If administrative metadata has been described for the file this attribute points to the file's administrative metadata.	0..1  MAY
CSIP 75	File reference to descriptive metadata  fileSec/fileGrp/file/@DMDID	If descriptive metadata has been described per file this attribute points to the file's descriptive metadata.	0..1  MAY
CSIP 76	File locator reference  fileSec/fileGrp/file/FLocat	The location of each external file must be defined by the file location (FLocat) element using the same rules as for referencing metadata files. All references to files should be made using the XLink href attribute and the file protocol using the relative location of the file.	1..1  MUST
CSIP 77	Type of locator  fileSec/fileGrp/file/FLocat/@LOCTYPE	The locator type is always used with the value "URL" from the vocabulary in the attribute.	1..1  MUST
CSIP 78	Type of link  fileSec/fileGrp/file/FLocat/@xlink:type	Attribute used with the value "simple". Value list is maintained by the xlink standard	1..1  MUST
CSIP 79	Resource location  fileSec/fileGrp/file/FLocat/@xlink:href	The actual location of the resource. We recommend recording a URL type filepath	1..1  MUST

		within this attribute.	
--	--	------------------------	--

Example: METS example of structuring the data in the file section

```
<mets:fileSec ID="uuid-CA580D47-8C8B-4E91-ABD5-142EBBE15B84">
  <mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86H"
USE="Documentation">
    <mets:file ID="uuid-0C0049CA-6DE0-4A6D-8699-7975E4046A81"
MIMETYPE="application/vnd.openxmlformats-officedocument.wordprocessingml.document"
SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00"
CHECKSUM="91B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557"
CHECKSUMTYPE="SHA-256">
      <mets:FLocat LOCTYPE="URL" type="simple" href="Documentation/File.docx">
      </mets:FLocat>
    </mets:file>
    <mets:file ID="uuid-0C0049CA-6DE0-4A6D-8699-7975E4046A82"
MIMETYPE="application/vnd.openxmlformats-officedocument.wordprocessingml.document"
SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00"
CHECKSUM="91B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557"
CHECKSUMTYPE="SHA-256">
      <mets:FLocat LOCTYPE="URL" type="simple" href="Documentation/File2.docx">
      </mets:FLocat>
    </mets:file>
  </mets:fileGrp>
  <mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86F" USE="Schemas">
    <mets:file ID="uuid-A1B7B0DA-E129-48EF-B431-E553F2977FD6" MIMETYPE="text/xsd"
SIZE="123917" CREATED="2018-04-24T14:37:49.617+01:00"
CHECKSUM="0BF9E16ADE296EF277C7B8E5D249D300F1E1EB59F2DCBD89644B676D66F72DCC"
CHECKSUMTYPE="SHA-256">
      <mets:FLocat LOCTYPE="URL" type="simple" href="schemas/ead2002.xsd">
      </mets:FLocat>
    </mets:file>
  </mets:fileGrp>
  <mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86G"
USE="Representations/Submission/Data" CONTENTINFORMATIONTYPE="SIARDDK">
    <mets:file ID="uuid-EE23344D-4F64-40C1-8E18-75839EF661FD" MIMETYPE="xml"
SIZE="1338744" CREATED="2018-04-24T14:37:49.617+01:00"
CHECKSUM="7176A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
CHECKSUMTYPE="SHA-256" ADMID="uuid-48C18DD8-2561-4315-AC39-F941CBB138B3
uuid-9124DA4D-3736-4F69-8355-EB79A22E943F">
```

```

    <mets:FLocat LOCTYPE="URL" type="simple"
href="representations/Submission/Data/SIARD.xml">
  </mets:FLocat>

```

```

</mets:file>

```

```

</mets:fileGrp>

```

```

</mets:fileSec>

```

Example: METS example of structuring the data in the file section when there are representatins present

```

<mets:fileSec ID="uuid-CA580D47-8C8B-4E91-ABD5-142EBBE15B84">

```

```

  <mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86H"
USE="Documentation">

```

```

    <mets:file ID="uuid-0C0049CA-6DE0-4A6D-8699-7975E4046A81"
MIMETYPE="application/vnd.openxmlformats-officedocument.wordprocessingml.document"
SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00"
CHECKSUM="91B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557"
CHECKSUMTYPE="SHA-256">

```

```

      <mets:FLocat LOCTYPE="URL" type="simple" href="documentation/File.docx">

```

```

      </mets:FLocat>

```

```

    </mets:file>

```

```

    <mets:file ID="uuid-0C0049CA-6DE0-4A6D-8699-7975E4046A82"
MIMETYPE="application/vnd.openxmlformats-officedocument.wordprocessingml.document"
SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00"
CHECKSUM="91B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557"
CHECKSUMTYPE="SHA-256">

```

```

      <mets:FLocat LOCTYPE="URL" type="simple" href="documentation/File2.docx">

```

```

      </mets:FLocat>

```

```

    </mets:file>

```

```

  </mets:fileGrp>

```

```

  <mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86F" USE="schemas">

```

```

    <mets:file ID="uuid-A1B7B0DA-E129-48EF-B431-E553F2977FD6" MIMETYPE="text/xsd"
SIZE="123917" CREATED="2018-04-24T14:37:49.617+01:00"
CHECKSUM="0BF9E16ADE296EF277C7B8E5D249D300F1E1EB59F2DCBD89644B676D66F72DCC"
CHECKSUMTYPE="SHA-256">

```

```

      <mets:FLocat LOCTYPE="URL" type="simple" href="schemas/ead2002.xsd">

```

```

      </mets:FLocat>

```

```

    </mets:file>

```

```

  </mets:fileGrp>

```

```

  <mets:fileGrp ID="uuid-5811D494-6045-4741-924C-A1CFA340C277"
USE="Representations/preingest" CONTENTINFORMATIONTYPE="OTHER"

```

```

OTHERCONTENTINFORMATIONTYPE="Access database">
  <mets:file ID="uuid-EE23344D-4F64-40C1-8E18-75839EF661FE" MIMETYPE="xml"
SIZE="1338744" CREATED="2018-04-24T14:37:49.617+01:00"
CHECKSUM="7176A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
CHECKSUMTYPE="SHA-256" ADMID="uuid-48C18DD8-2561-4315-AC39-F941CBB138B3
uuid-9124DA4D-3736-4F69-8355-EB79A22E943F">
  <mets:FLocat LOCTYPE="URL" type="simple"
href="representations/preingest/METS.xml">
  </mets:FLocat>
</mets:file>
</mets:fileGrp>
<mets:fileGrp ID="uuid-5811D494-6045-4741-924C-A1CFA340C278"
USE="Representations/submission/data" CONTENTINFORMATIONTYPE="SIARDDK"
ADMID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F
uuid-48C18DD8-2561-4315-AC39-F941CBB138B3">
  <mets:file ID="uuid-EE23344D-4F64-40C1-8E18-75839EF661FF"
MIMETYPE="application/xml" SIZE="1338744" CREATED="2018-04-24T14:37:49.617+01:00"
CHECKSUM="7176A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
CHECKSUMTYPE="SHA-256" ADMID="uuid-48C18DD8-2561-4315-AC39-F941CBB138B3
uuid-9124DA4D-3736-4F69-8355-EB79A22E943F">
  <mets:FLocat LOCTYPE="URL" type="simple"
href="representations/Submission/METS.xml">
  </mets:FLocat>
</mets:file>
</mets:fileGrp>
<mets:fileGrp ID="uuid-5811D494-6045-4741-924C-A1CFA340C279"
USE="Representations/ingest/data" CONTENTINFORMATIONTYPE="SIARD1"
ADMID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943G
uuid-48C18DD8-2561-4315-AC39-F941CBB138B4">
  <mets:file ID="uuid-EE23344D-4F64-40C1-8E18-75839EF661FG"
MIMETYPE="application/xml" SIZE="1338744" CREATED="2018-04-24T14:37:49.617+01:00"
CHECKSUM="7176A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
CHECKSUMTYPE="SHA-256" ADMID="uuid-48C18DD8-2561-4315-AC39-F941CBB138B3
uuid-9124DA4D-3736-4F69-8355-EB79A22E943F">
  <mets:FLocat LOCTYPE="URL" type="simple"
href="representations/ingest/METS.xml">
  </mets:FLocat>
</mets:file>
</mets:fileGrp>
</mets:fileSec>

```

### 5.3.6. Use of the METS structural map (element structMap)

The METS structural map section is the only element mandatory in the METS specification and it is intended to provide an overview of components described in the METS document. It can also link the elements of that structure to associated content files and metadata. In CSIP the structMap describes the higher level structure of all the content in the root and may link to representations.

The CSIP requires the inclusion of one mandatory structural map according to the principles described below. However, implementers are welcome to define additional structural maps for their internal purposes by repeating the structMap element. The most crucial requirements for the CS IP mandated structural map are as follows:

- The structMap element has a mandatory attribute @LABEL which has the fixed value of “CSIP StructMap”. The @LABEL attribute is used to distinguish the Common Specification mandated structural map occurrence from any other, user-defined, structural maps. As such we can also derive the requirement, that any user-defined structural maps must not use the LABEL value of “CS IP StructMap”;
- The internal structure of the structural map (expressed by div elements) follows the CSIP high level physical structure as described in Section 4, therefore grouping together metadata, representations, schemas, documentation and user-defined folders into their own div elements;
  - All div elements must use the attribute LABEL with the value being the name of the folder (as an example “metadata”)
- In case both root and representation METS files exist, the structural map in the root METS file
  - Reference the fileGrp which describes all files in all folders with the exception of the content of the representation folders
  - Lists all representations (as separate div elements)
  - Lists only the appropriate representation METS file using the mptr element as the content of the representation
- The structural map in the representations METS file use the structural map the same way with no exceptions

The specific requirements for elements, sub-elements and attributes are listed in the following table. Note that the area, seq and par elements are not discussed below.

ID	Name & Location	Description & usage	Cardinality & Level
CSIP 80	Structural description of the package  structMap	Each METS file must include ONE structural map (structMap) element used exactly as described here. Institutions can add their own additional custom structural maps as separate structMap sections.	1..n  MUST
CSIP 81	Type of structural description  structMap/@TYPE	The type attribute of the structural map (structMap) is set to value “PHYSICAL” from the vocabulary.  See also:  <ul><li> <a href="#">Structural map typing</a> </li></ul>	1..1  MUST
CSIP 82	Name of the structural description  structMap/@LABEL	The label attribute is set to value “CSIP StructMap” from the vocabulary.  See also:  <ul><li> <a href="#">Structural map label</a> </li></ul>	1..1  MUST
CSIP 83	Structural description identifier  structMap/@ID	An identifier for the structural description (structMap) used for referencing inside the package. It must be unique within the package.  The ID must follow the rules for xml:id described in the chapter of the textual description of CSIP named “General requirements for the use of	1..1  MUST



		metadata”	
CSIP 84	Main structural division  structMap/div	The structural map consist of one main division.	1..1  MUST
CSIP 85	Main division identifier  structMap/div/@ID	Mandatory, identifier must be unique within the package.  The ID must follow the rules for xml:id described in the chapter of the textual description of CSIP named “General requirements for the use of metadata”	1..1  MUST
CSIP 86	Main structural division label  structMap/div/@LABEL	The main division (div) element in the package uses the package ID as the value for the attribute LABEL.	1..1  MUST
CSIP 87	Sub structural division  structMap/div	Each categorization “Documentation”, “Schemas” as well as each “Representation” within the package must be represented by an occurrence of the division (div) element.  Metadata in the administrative and descriptive metadata section has its own division	1..n  MUST
CSIP 88	Metadata division  structMap/div/div	The metadata referenced in the administrative and/or descriptive metadata section is described in the structural map with one sub division  When the transfer consist of	1..1  MUST

		only administrative and/or descriptive metadata this is the only sub division that occurs	
CSIP 89	<p>Metadata division identifier</p> <p>structMap/div/div/@ID</p>	<p>Mandatory, identifier must be unique within the package.</p> <p>The ID must follow the rules for xml:id described in the chapter of the textual description of CSIP named “General requirements for the use of metadata”</p>	<p>1..1</p> <p>MUST</p>
CSIP 90	<p>Metadata division label</p> <p>structMap/div/div/@LABEL</p>	<p>The metadata division (div) element in the package uses the value “Metadata” as the value for the attribute LABEL.</p> <p>See also:</p> <p>&lt;ul&gt;&lt;li&gt;<a href="#">File group names</a>&lt;/li&gt;&lt;/ul&gt;</p>	<p>1..1</p> <p>MUST</p>
CSIP 91	<p>Metadata division administrative metadata referencing</p> <p>structMap/div/div/@ADMID</p>	<p>All administrative metadata described in the package are referenced via the administrative sections different identifiers.</p>	<p>0..1</p> <p>MUST</p>
CSIP 92	<p>Metadata division descriptive metadata referencing</p> <p>structMap/div/div/@DMDID</p>	<p>All descriptive metadata described in the package are referenced via the descriptive section identifiers.</p>	<p>0..1</p> <p>MUST</p>

CSIP 93	Documentation division  structMap/div/div	The documentation referenced in the file section file groups is described in the structural map with one sub division	0..1  SHOULD
CSIP 94	Documentation division identifier  structMap/div/div/@ID	Mandatory, identifier must be unique within the package.  The ID must follow the rules for xml:id described in the chapter of the textual description of CSIP named “General requirements for the use of metadata”.	1..1  MUST
CSIP 95	Documentation division label  structMap/div/div/@LABEL	The documentation division (div) element in the package uses the value “Documentation” as the value for the attribute LABEL.  See also:  <ul><li> <a href="#">File group names</a> </li></ul>	1..1  MUST
CSIP 96	Documentation file referencing  structMap/div/div/@CONTENTID	All file groups containing documentation described in the package are referenced via the relevant file group identifiers.	1..1  MUST
CSIP 97	Schema division  structMap/div/div	The schemas referenced in the file section file groups is described in the structural map with one sub division	0..1  SHOULD
CSIP 98	Schema division identifier  structMap/div/div/@	Mandatory, identifier must be unique within the package.  The ID must follow the rules for	1..1  MUST

	ID	xml:id described in the chapter of the textual description of CSIP named “General requirements for the use of metadata”.	
CSIP 99	Schema division label  structMap/div/div/@LABEL	The schema division (div) element in the package uses the value “Schemas” as the value for the attribute LABEL.  See also:  <ul><li> <a href="#">File group names</a> </li></ul>	1..1  MUST
CSIP 100	Schema file referencing  structMap/div/div/@CONTENTID	All file groups containing schemas described in the package are referenced via the relevant file group identifiers.	1..1  MUST
CSIP 101	File division  structMap/div/div	When the transfer consist of only data and no representations there are one representation div present  The transferred files referenced in the file section file group is described in the structural map with one sub division	0..1  SHOULD
CSIP 102	File division identifier  structMap/div/div/@ID	Mandatory, identifier must be unique within the package.  The ID must follow the rules for xml:id described in the chapter of the textual description of CSIP named “General requirements for the use of	1..1  MUST

		metadata”.	
CSIP 103	File division label  structMap/div/div/@ LABEL	The file division (div) element in the package uses the value “Representations” as the value for the attribute LABEL.  See also:  <ul><li> <a href="#">File group names</a> </li></ul>	1..1  MUST
CSIP 104	File division file referencing  structMap/div/div/@ CONTENTID	The file group containing the files described in the package are referenced via the relevant file group identifier.	1..1  MUST
CSIP 105	Representation divisions  structMap/div/div	When the transfer consist of representations there are one representation div present for each representation	0..n  SHOULD
CSIP 106	Representation division identifier  structMap/div/div/@ ID	Mandatory, identifier must be unique within the package.  The ID must follow the rules for xml:id described in the chapter of the textual description of CSIP named “General requirements for the use of metadata”.	1..1  MUST
CSIP 107	Representation division label  structMap/div/div/@ LABEL	The representation division (div) element in the package uses the path to the METS document as the value for the attribute LABEL.  See also:	1..1  MUST

		<ul><li> <a href="#">File group names</a> </li></ul>	
CSIP 108	Representations division file referencing  structMap/div/div/@CONTENTID	The file group containing the files described in the package are referenced via the relevant file group identifier.	1..1  MUST
CSIP 109	Representation METS pointer  structMap/div/div/mptr	The division (div) of the specific representation includes one occurrence of the METS pointer (mptr) element, pointing to the appropriate representation METS file.	1..1  MUST
CSIP 110	Resource location  structMap/div/div/mptr/@xlink:href	The actual location of the resource. We recommend recording a URL type filepath within this attribute.	1..1  MUST
CSIP 111	Type of link  structMap/div/div/mptr/@xlink:type	Attribute used with the value "simple". Value list is maintained by the xlink standard	1..1  MUST
CSIP 112	Type of locator  structMap/div/div/mptr/@LOCTYPE	The locator type is always used with the value "URL" from the vocabulary in the attribute.	1..1  MUST

Example: METS example of the mandatory structural map

```
<mets:structMap ID="uuid-1465D250-0A24-4714-9555-5C1211722FB8" TYPE="PHYSICAL"
LABEL="CSIP StructMap">
  <mets:div ID="uuid-638362BC-65D9-4DA7-9457-5156B3965A18"
LABEL="uuid-4422c185-5407-4918-83b1-7abfa77de182">
    <mets:div ID="uuid-A4E1C5B6-CD9B-43EF-8F0C-3FD3AB688F81" LABEL="Metadata"
ADMID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F
uuid-48C18DD8-2561-4315-AC39-F941CBB138B3"
```

```

DMDID="uuid-906F4F12-BA52-4779-AE2C-178F9206111F">
  </mets:div>
  <mets:div ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86I" LABEL="Documentation"
CONTENTIDS="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86H">
  </mets:div>
  <mets:div ID="uuid-26757DC2-4C0F-4431-85B5-5943D1AB5CA3" LABEL="Schemas"
CONTENTIDS="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86F">
  </mets:div>
  <mets:div ID="uuid-35CB3341-D731-4AC3-9622-DB8901CD6736"
LABEL="Representations" CONTENTIDS="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86G">
  </mets:div>
</mets:div>
</mets:structMap>

```

**Example:** METS example of the mandatory structural map when there are representations present

```

<mets:structMap ID="uuid-1465D250-0A24-4714-9555-5C1211722FB8" TYPE="PHYSICAL"
LABEL="CSIP StructMap">
  <mets:div ID="uuid-638362BC-65D9-4DA7-9457-5156B3965A18"
LABEL="uuid-4422c185-5407-4918-83b1-7abfa77de182">
    <mets:div ID="uuid-A4E1C5B6-CD9B-43EF-8F0C-3FD3AB688F81" LABEL="Metadata"
ADMID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F
uuid-48C18DD8-2561-4315-AC39-F941CBB138B3
uuid-9124DA4D-3736-4F69-8355-EB79A22E943G
uuid-48C18DD8-2561-4315-AC39-F941CBB138B4"
DMDID="uuid-906F4F12-BA52-4779-AE2C-178F9206111F">
    </mets:div>
    <mets:div ID="uuid-26757DC2-4C0F-4431-85B5-5943D1AB5CA3" LABEL="Schemas"
CONTENTIDS="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86F">
    </mets:div>
    <mets:div ID="uuid-35CB3341-D731-4AC3-9622-DB8901CD6737"
LABEL="representations/preingest"
CONTENTIDS="uuid-5811D494-6045-4741-924C-A1CFA340C277">
      <mets:mptr LOCTYPE="URL" type="simple"
href="representations/preingest/METS.xml">
      </mets:mptr>
    </mets:div>
    <mets:div ID="uuid-35CB3341-D731-4AC3-9622-DB8901CD6736"
LABEL="representations/submission"
CONTENTIDS="uuid-5811D494-6045-4741-924C-A1CFA340C278">

```

```
<mets:mptr LOCTYPE="URL" type="simple"
href="representations/submission/METS.xml">
</mets:mptr>
</mets:div>
<mets:div ID="uuid-35CB3341-D731-4AC3-9622-DB8901CD6738"
LABEL="representations/ingest"
CONTENTIDS="uuid-5811D494-6045-4741-924C-A1CFA340C279">
<mets:mptr LOCTYPE="URL" type="simple"
href="representations/ingest/METS.xml">
</mets:mptr>
</mets:div>
</mets:div>
</mets:structMap>
```



## 5.4. Use of PREMIS

---

The CS IP recommends and advocates the use of the PREservation Metadata Implementation Strategies (PREMIS, information available at <http://www.loc.gov/standards/premis/>) metadata standard for recording preservation and technical metadata about digital objects contained within CS IP Information Packages. The CS IP implements version 3.0 of the PREMIS Data Dictionary. Note that use of PREMIS is not mandatory.

We strongly recommend keeping PREMIS metadata in discrete PREMIS XML files inside the IP. The PREMIS metadata can be included in the IP in separate files, and there is no convention regarding the naming and numbering of the PREMIS files. Implementations can choose to either store all preservation metadata in a single PREMIS file or split them into multiple files. The only requirement in this case is that all PREMIS files must be listed in the appropriate METS file, i.e. root PREMIS files from the root METS file and representation PREMIS files from the representation METS files, and referenced in the METS file(s) using the mdRef attributes and elements.

Therefore, the main recommendation of the CS IP is that preservation metadata are included in the information package in PREMIS format. Although this is not mandatory, all tools claiming to be able to validate CS IP compliant Information Packages must also be able to validate PREMIS metadata once it exists within the package. The two high level requirements for use of PREMIS in Common Specification IPs are that:

- All preservation metadata is created according to official PREMIS guidelines ;
- All PREMIS metadata is either embedded in or referenced from the amdSec/digiprovMD element of the appropriate METS file.

Further, to enhance the interoperability scope of the CS IP and to strengthen management of IPs in an archive, this specification imposes additional requirements with regard to use of PREMIS for describing Information Packages. The principles adopted in the CS IP for deciding the additional PREMIS semantic units required are:

- PREMIS should be used to record detailed technical metadata;
- Technical information should be included in PREMIS metadata by using the possibility of extension schemas in PREMIS;
- Information about agents carrying out preservation actions must be recorded in the PREMIS metadata. (The METS agents describe agents who are relevant for generic IP level events, such as the creation or submission of the

package, for example);

- Event descriptions should be included in PREMIS metadata. Use of the official PREMIS event vocabulary ( <http://id.loc.gov/vocabulary/preservation/eventType.html> ) is recommended;
- Detailed rights information should be included in PREMIS. High level rights information in METS indicates restrictions. Detailed, object-specific, rights information will be included in the PREMIS metadata;
- File format information for all files should be included as PUID values in the appropriate PREMIS semantic units.

## Vocabularies

This specification does not present a definitive list of vocabularies for use with PREMIS semantic units but does recommend the use of the Library of Congress vocabularies developed specifically to provide values for various PREMIS semantic units. All relevant vocabularies is presented in the PREMIS Data Dictionary.

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 recognizes that the use of identifier types is an implementation specific issue and does not recommend or require particular vocabularies for identifier types. The Library of Congress has developed its own identifier type vocabulary and the CS IP recommends its use in lieu of implementation specific identifier type vocabularies, where these have not yet been developed.

## 6. Implementation considerations

---

This Section touches on some additional issues which are relevant in respect to implementing the CS IP in real-life scenarios.

### 6.1 Content Information Type Specifications

---

#### 6.1.1 What is a Content Information Type Specification?

The concept of Content Information Type Specification is essentially an extension method which allows for widening the interoperability scope of the CS IP into a content specific level.

As defined by the OAIS Reference Model, Content Information is “A set of information that is the original target of preservation or that includes part or all of that information. It is an Information Object composed of its Content Data Object and its Representation Information”.

A Content Information Type can therefore be understood as a category of Content Information, for example relational databases, scientific data or digitised maps. And finally a Content Information Type Specification defines in technical terms how data and metadata (mainly in regard to the Information Object) must be formatted and placed within a CS IP Information Package in order to achieve interoperability in exchanging specific Content Information.

As such, the following elements can be at the core of a Content Information Type Specification:

- The required file format of data;
- Description of how data must be placed and structured within the CS IP folder structure (i.e. a sub-structure for the “Data” folder);
- Clearly defined requirements for specific representation metadata that needs to be available in PREMIS for rendering and understanding the Content Data Object appropriately;
- Clearly defined list of specific (binary) documentation or other components (like software, emulators, etc.) which have to be available for rendering and understanding the Content Data Object appropriately.

However, for practical purposes it is not sufficient to only deal with the Information Object. Especially for complex Content Information Types and large IPs it might also be relevant to describe explicitly requirements for other metadata (descriptive,

administrative) which are relevant and crucial only for this specific content type. For example, the ERMS Content Information Type Specification, developed within the E-ARK project, does set specific requirements for how data (i.e. computer files) need to be referenced from descriptive metadata (in ERMS format) in order to guarantee the integrity of data and metadata. Setting these requirements in a central specification will allow archival institutions to receive SIPs including ERMS extracts or whole systems and still be able to understand and validate the potentially complex structure of the whole data and metadata composition within it.

Concluding from the previous we can also see that Content Information Type Specification can potentially also be sector specific, and that there might be multiple specifications to cover a single content type. For example, archival institutions would be able to define a Content Information Type specification for archiving web sites along with descriptive metadata in EAD format, while libraries might define a specification for archiving web sites along with metadata in MARC.

### **6.1.2 Maintaining Content Information Type Specifications**

The number of possible Content Information Type Specifications is potentially unlimited. As well, it is the intention of the authors of the CS IP to allow everybody in the wider community to create new specifications.

The maintenance of such a living environment is the role of the DILCIS Board. The core principles of the maintenance regime are as follows:

- The DILCIS Board is responsible for establishing reasonable guidelines and quality requirements for new Content Information Type specifications, and publishing these on the Board website;
- The Board has the responsibility and mandate to manage a registry of available Content Information Type specifications which meet the guidelines and quality requirements;
- The Board does NOT take ownership of or have responsibility for maintaining and sustaining any Content Information Type specifications;
- There shall be no limitations to who is allowed to propose additional Content Information Type specifications;
- To ensure good quality of available specifications, the Board validates each proposed specification against the guidelines and quality requirements mentioned above. The validation shall be carried out free of charge and within a reasonable timeframe.

## 6.2. Handling large packages

---

By default a Common Specification IP is supposed to reside in a single folder or file (in case compression has been applied). However, the amount of data and metadata within a single IP can easily grow into sizes of several GB or even TB and as such can become difficult to manage and inefficient to process because, for example, of lacking media capacity.

The Common Specification itself can in principle be extended in multiple ways to support the segmenting of large packages into more manageable physical pieces. This Section describes one way which exploits the Common Specification “representation METS” concept and extends it into a physical segmentation scenario.

However, it is worth noting that this is a “recommended approach” and is, at this point in time, not a part of the core Common Specification, as such it is also not expected that all tools support such a mechanism.

### 6.2.1 The structure for IP, their representations and their segments

According to the E-ARK Common Specification for IPs an IP can have several representations. All representations contain the same intellectual content, but as the name implies is another representation; in its most simple form this could be another file format such as TIFF instead of JPEG.

The segmenting approach described here is based on the following considerations:

- Most of the size of an IP is the content (data) which according to the Common Specification resides in the representations folder of the IP. As such also any segmenting should take place within the representations layer of the Common Specification;
- According to the Common Specification each representation is essentially a Common Specification IP itself, as it can consist of a METS metadata file, data, metadata, and any additional components;
- A segment of an IP must also be in the Common Specification format, i.e. it shall be possible to validate each individual segment as a Common Specification IP;
- Each IP shall consist of a parent segment (including at least the root METS file) and any number of child segments;
- It shall be possible to add new physical child segments (as an example a new

representation) to the whole IP without having to update other child segments.

### 6.2.2 Using METS to refer from parent IP to child IP(s)

The method used to refer from parent to child is based on the ID of the IP of the child.

One reason for using ID and not URL or other more direct references to a location of the referenced METS file is the flexibility it gives to move the segmented IPs around in different storage locations. This is a flexibility often needed for segmented IPs that accumulated can be very large.

The value of the xlink:href attribute in the element in the METS file of the parent IP is used.

This value is to be set to the value of the OBJID attribute of the element in the METS file of the child IP. According to the Common Specification, the OBJID attribute must have the value of the ID of the IP. This is therefore sufficient for having the parent know the ID of the child, but the parent does not know the exact child location.

### 6.2.3 Using METS to refer from child IP to parent IP

The optional reference from child to the parent is based on the ID of the IP of the parent.

The value of the xlink:href attribute in element in the METS file of the child IP is used.

This value is to be set to the value of the OBJID attribute of the element in the METS file of the parent IP. According to the Common Specification, the OBJID attribute must have the value of the ID of the IP.

This is therefore sufficient for having the child know the ID of the parent, but the child does not know the exact parent location.

### 6.2.4 An example for the Northwind database

Here follows a partial example, where the value of the xlink:href attribute in the <mptr> element (inside the <div> element inside the <structMap> element) is ID.AVID.RA.18005.rep0(seg0 after the urn NID part (urn:<NID>:<NSS>).

The value ID.AVID.RA.18005.rep0(seg0 must now match the value of the OBJID attribute for the <mets> element in the child IP root METS file. (Note that in order to save space in this example the CS mandatory ID attribute for the <div> elements

have been left out.) Parent METS file

```
<!-- this top root level METS.xml IP only refers to the root level METS files in
the representations using the <mptr> element -->
<div LABEL="representations">
  <!-- the value of the attribute LABEL is the ID of the representation -->
  <div LABEL="representations/ID.AVID.RA.18005.rep0" ORDER="0" >
    <!-- we use the attribute LABEL value 'child IP' in the 'div' element for
    representations in accordance with the AIP spec.3.3.1.9 -->
    <div LABEL="child IP" TYPE="representation child">
      <!-- each root level METS file in the representations refer to its own METS files
      in the segments and in the representations folder using
      the <mptr> element -->
      <!-- this is a METS reference to another METS file, and this file is in another
      segment -->
      <mptr xlink:href="urn:sa.dk:ID.AVID.RA.18005.rep0.seg0" xlink:title="root
      level METS file for representation 0" xlink:type="simple"
      LOCTYPE="URN"/>
    </div>
  </div>
  <!-- the value of the attribute LABEL is the ID of the representation -->
  <div LABEL="representations/ID.AVID.RA.18005.rep1" ORDER="1">
    <div LABEL="child IP" TYPE="representation child">
      <!-- this is an indirect METS reference to another METS file, and this file is in
      another segment -->
      <mptr xlink:href="urn:sa.dk:ID.AVID.RA.18005.rep1.seg0" xlink:title="root
      level METS file for representation 1" xlink:type="simple"
      LOCTYPE="URN"/>
    </div>
  </div>
</div>
```

Child METS file

```
<mets xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.loc.gov/METS/"
xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.loc.gov/METS/ schemas/mets.xsd"
PROFILE="http://www.ra.ee/METS/v01/IP.xml" TYPE="Database segment child"
OBJID="ID.AVID.RA.18005.rep0.seg0" LABEL="root
```



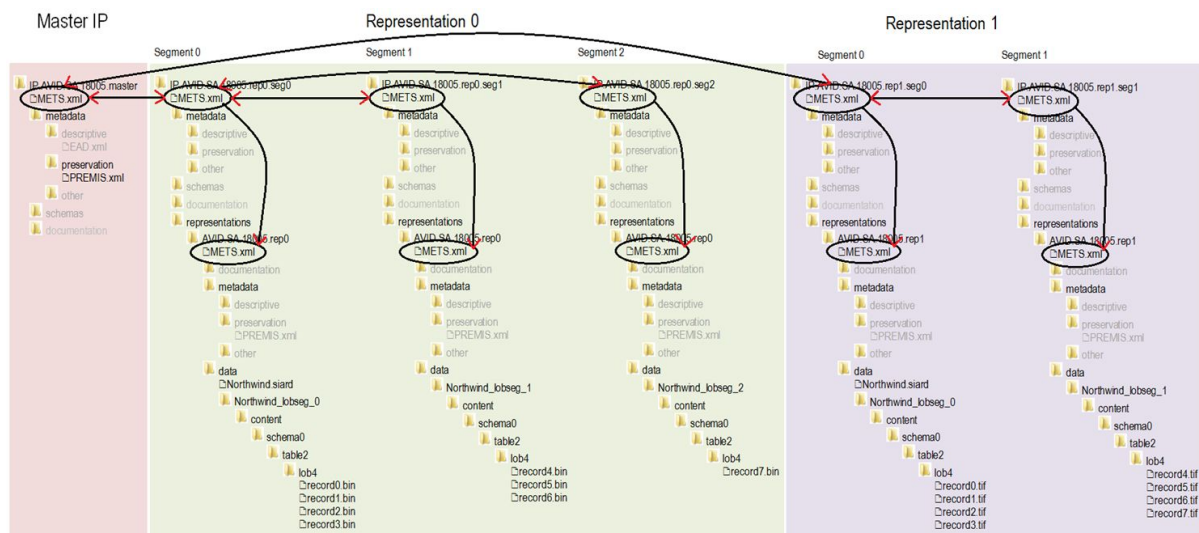
```

level METS file for a representation segment">
..
..
<div LABEL="parent IP" TYPE="Godfather IP"> <!-- working title - maybe master
IP is more appropriate -->
<!-- this is an indirect METS reference to another METS file. However, the
referenced file is in another segment -->
<mptr xlink:href="urn:sa.dk:ID.AVID.RA.18005.godfather" xlink:title="root
level METS file for godfather IP" xlink:type="simple"
LOCTYPE="URN"/>
</div>

```

### 6.2.5 Illustration of references between METS files in a segmented IP

We need to segment an IP at the data folder in the representations level, but according to the Common Specification this can only be done at the IP level. Therefore this IP has been segmented at the top IP level, and not at the representations level.



Please note the following about the example:

- The Master IP MUST NOT contain representations
- A representation MAY be segmented
- The IDs are not just unique but have implicit value for example purposes only
- In representation 0 the limits on folder size and amount of files requires three segments (0, 1 and 2)



- In representation 1 these limits have been increased and we only need two segments. Further the .bin files have been migrated to .tif.

## 6.3 Handling descriptive metadata within the Common Specification

Descriptive metadata are used to describe the intellectual contents of archival holdings, and they support finding and understanding individual information packages. The CS IP allows essentially for the inclusion of any kind of descriptive metadata in the IP. However, it is required that all descriptive metadata must be placed into the “metadata” folder of the IP, and that it is recommended (should) to also exploit the possibility of creating a specific sub-folder “descriptive” as seen in Figure 11 below (cf. EAD.xml).

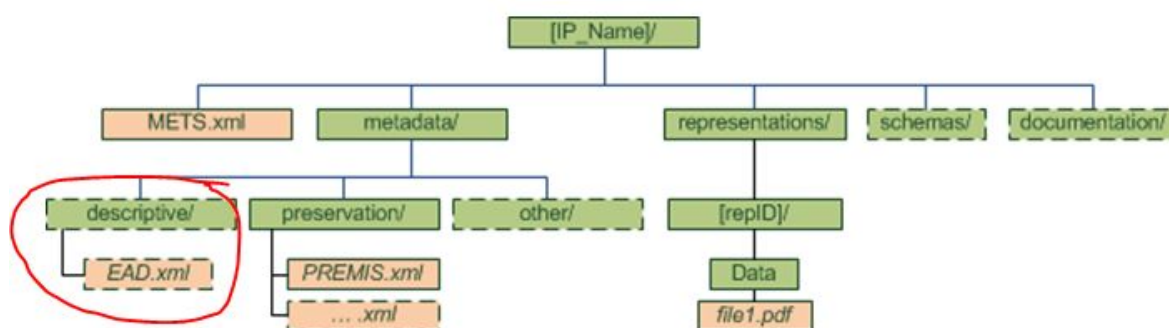


Figure 11: E-ARK IP descriptive metadata

Further, all descriptive metadata need itself to be described in and referenced from METS metadata (i.e. the METS.xml file) using the element `<dmdSec>` (Figure 12) and as such descriptive metadata are not to be embedded into the METS file directly.

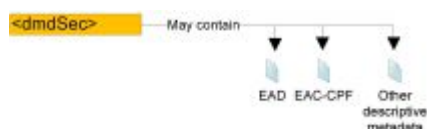


Figure 12: METS descriptive metadata

Following the requirement of explicitly and physically separating descriptive metadata and data we would also like to note, that for interoperability purposes appropriate descriptive metadata elements must explicitly refer to the data content they describe (unless the whole data portion is a single intellectual unit described as a discrete set of descriptive metadata). For example, in the case of EAD elements and `` shall be used to refer to content files from the descriptive metadata. However, regardless of the descriptive metadata standard in question the references from

descriptive metadata must always follow the requirement posed in Section 5.1 above (i.e. create references according to the format defined in RFC 3986, or to express references as a relative path to the data files).

Finally we would also note that the recommendation of the CS IP is to always include detailed metadata about intellectual access restrictions and copyright into descriptive metadata (i.e. not into the METS or PREMIS portions of the IP).

## Appendices

---

### Appendix A: E-ARK Information Package METS examples

---

Example 1: Example of a whole METS document describing an information package with no representations

```
<mets:mets OBJID="uuid-4422c185-5407-4918-83b1-7abfa77de182" LABEL="Sample CSIP
Information Package with no representations" TYPE="Database"
CONTENTINFORMATIONTYPE="SIARDDK"
PROFILE="https://eakcsip.dilcis.eu/profile/CSIP.xml"
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://dilcis.eu/XML/METS/CSIPExtensionMETS/DILCISExtensionMETS.xsd">
  <mets:metsHdr CREATEDATE="2018-04-24T14:37:49.602+01:00"
LASTMODDATE="2018-04-24T14:37:49.602+01:00" RECORDSTATUS="NEW"
OAISPACKAGETYPE="SIP">
    <mets:agent ROLE="CREATOR" TYPE="OTHER" OTHERTYPE="SOFTWARE">
      <mets:name>RODA-in</mets:name>
      <mets:note NOTETYPE="SOFTWARE VERSION">2.1.0-beta.7</mets:note>
    </mets:agent>
  </mets:metsHdr>
  <mets:dmdSec ID="uuid-906F4F12-BA52-4779-AE2C-178F9206111F"
CREATED="2018-04-24T14:37:49.609+01:00">
    <mets:mdRef LOCTYPE="URL" MDTYPE="EAD" MDTYPEVERSION="2002" type="simple"
href="metadata/descriptive/ead2002.xml" SIZE="903"
CREATED="2018-04-24T14:37:49.609+01:00"
CHECKSUM="F24263BF09994749F335E1664DCE0086DB6DCA323FDB6996938BCD28EA9E8153"
```

```

CHECKSUMTYPE="SHA-256">
  </mets:mdRef>
</mets:dmdSec>
<mets:amdSec>
  <mets:digiprovMD ID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F"
    CREATED="2018-04-24T14:37:52.783+01:00">
    <mets:mdRef LOCTYPE="URL" type="simple"
      href="metadata/preservation/premis1.xml" MDTYPE="PREMIS:EVENT" MDTYPEVERSION="3.0"
      MIMETYPE="text/xml" SIZE="1211" CREATED="2018-04-24T14:37:52.783+01:00"
      CHECKSUM="8aa278038dbad54bbf142e7d72b493e2598a94946ea1304dc82a79c6b4bac3d5"
      CHECKSUMTYPE="SHA-256" LABEL="premis1.xml">
      </mets:mdRef>
    </mets:digiprovMD>
    <mets:digiprovMD ID="uuid-48C18DD8-2561-4315-AC39-F941CBB138B3"
      CREATED="2018-04-24T14:47:52.783+01:00">
      <mets:mdRef LOCTYPE="URL" type="simple"
        href="metadata/preservation/premis2.xml" MDTYPE="PREMIS:OBJECT"
        MDTYPEVERSION="3.0" MIMETYPE="text/xml" SIZE="2854"
        CREATED="2018-04-24T14:37:52.783+01:00"
        CHECKSUM="d1dfa585dcc9d87268069dc58d5e47956434ec3db4087a75a3885d287f15126f"
        CHECKSUMTYPE="SHA-256" LABEL="premis2.xml">
        </mets:mdRef>
      </mets:digiprovMD>
    </mets:amdSec>
    <mets:fileSec ID="uuid-CA580D47-8C8B-4E91-ABD5-142EBBE15B84">
      <mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86H"
        USE="Documentation">
        <mets:file ID="uuid-0C0049CA-6DE0-4A6D-8699-7975E4046A81"
          MIMETYPE="application/vnd.openxmlformats-officedocument.wordprocessingml.document"
          SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00"
          CHECKSUM="91B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557"
          CHECKSUMTYPE="SHA-256">
          <mets:FLocat LOCTYPE="URL" type="simple" href="Documentation/File.docx">
          </mets:FLocat>
        </mets:file>
        <mets:file ID="uuid-0C0049CA-6DE0-4A6D-8699-7975E4046A82"
          MIMETYPE="application/vnd.openxmlformats-officedocument.wordprocessingml.document"
          SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00"
          CHECKSUM="91B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557"
          CHECKSUMTYPE="SHA-256">

```

```

        <mets:FLocat LOCTYPE="URL" type="simple" href="Documentation/File2.docx">
    </mets:FLocat>
</mets:file>
</mets:fileGrp>
    <mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86F" USE="Schemas">
        <mets:file ID="uuid-A1B7B0DA-E129-48EF-B431-E553F2977FD6"
MIMETYPE="text/xsd" SIZE="123917" CREATED="2018-04-24T14:37:49.617+01:00"
CHECKSUM="0BF9E16ADE296EF277C7B8E5D249D300F1E1EB59F2DCBD89644B676D66F72DCC"
CHECKSUMTYPE="SHA-256">
            <mets:FLocat LOCTYPE="URL" type="simple" href="schemas/ead2002.xsd">
        </mets:FLocat>
    </mets:file>
</mets:fileGrp>
    <mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86G"
USE="Representations/Submission/Data" CONTENTINFORMATIONTYPE="SIARDDK">
        <mets:file ID="uuid-EE23344D-4F64-40C1-8E18-75839EF661FD" MIMETYPE="xml"
SIZE="1338744" CREATED="2018-04-24T14:37:49.617+01:00"
CHECKSUM="7176A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
CHECKSUMTYPE="SHA-256" ADMID="uuid-48C18DD8-2561-4315-AC39-F941CBB138B3
uuid-9124DA4D-3736-4F69-8355-EB79A22E943F">
            <mets:FLocat LOCTYPE="URL" type="simple"
href="representations/Submission/Data/SIARD.xml">
        </mets:FLocat>
    </mets:file>
</mets:fileGrp>
</mets:fileSec>
    <mets:structMap ID="uuid-1465D250-0A24-4714-9555-5C1211722FB8" TYPE="PHYSICAL"
LABEL="CSIP StructMap">
        <mets:div ID="uuid-638362BC-65D9-4DA7-9457-5156B3965A18"
LABEL="uuid-4422c185-5407-4918-83b1-7abfa77de182">
            <mets:div ID="uuid-A4E1C5B6-CD9B-43EF-8F0C-3FD3AB688F81" LABEL="Metadata"
ADMID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F
uuid-48C18DD8-2561-4315-AC39-F941CBB138B3"
DMDID="uuid-906F4F12-BA52-4779-AE2C-178F9206111F">
                </mets:div>
            <mets:div ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86I"
LABEL="Documentation" CONTENTIDS="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86H">
                </mets:div>
            <mets:div ID="uuid-26757DC2-4C0F-4431-85B5-5943D1AB5CA3" LABEL="Schemas"
CONTENTIDS="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86F">

```

```

    </mets:div>
    <mets:div ID="uuid-35CB3341-D731-4AC3-9622-DB8901CD6736"
    LABEL="Representations" CONTENTIDS="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86G">
    </mets:div>
  </mets:div>
</mets:structMap>
</mets:mets>

```

Example 2: Example of a whole METS document describing an information package with representations

```

<mets:mets OBJID="uuid-4422c185-5407-4918-83b1-7abfa77de182" LABEL="Sample CSIP
Information Package with representations" TYPE="Database"
PROFILE="https://eakcsip.dilcis.eu/profile/CSIP.xml"
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://dilcis.eu/XML/METS/CSIPExtensionMETS/DILCISExtensionMETS.xsd">
  <mets:metsHdr CREATEDATE="2018-04-24T14:37:49.602+01:00"
  LASTMODDATE="2018-04-24T14:37:49.602+01:00" RECORDSTATUS="NEW"
  OAISPACKAGETYPE="SIP">
    <mets:agent ROLE="CREATOR" TYPE="OTHER" OTHERTYPE="SOFTWARE">
      <mets:name>RODA-in</mets:name>
      <mets:note NOTETYPE="SOFTWARE VERSION">2.1.0-beta.7</mets:note>
    </mets:agent>
  </mets:metsHdr>
  <mets:dmdSec ID="uuid-906F4F12-BA52-4779-AE2C-178F9206111F"
  CREATED="2018-04-24T14:37:49.609+01:00">
    <mets:mdRef LOCTYPE="URL" MDTYPE="EAD" MDTYPEVERSION="2002" type="simple"
    href="metadata/descriptive/ead2002.xml" SIZE="903"
    CREATED="2018-04-24T14:37:49.609+01:00"
    CHECKSUM="F24263BF09994749F335E1664DCE0086DB6DCA323FDB6996938BCD28EA9E8153"
    CHECKSUMTYPE="SHA-256">
    </mets:mdRef>
  </mets:dmdSec>
  <mets:amdSec>
    <mets:digiprovMD ID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F"
    CREATED="2018-04-24T14:37:52.783+01:00">
      <mets:mdRef LOCTYPE="URL" type="simple"

```

```

href="metadata/preservation/premis1.xml" MDTYPE="PREMIS:EVENT" MDTYPEVERSION="3.0"
MIMETYPE="text/xml" SIZE="1211" CREATED="2018-04-24T14:37:52.783+01:00"
CHECKSUM="8aa278038dbad54bbf142e7d72b493e2598a94946ea1304dc82a79c6b4bac3d5"
CHECKSUMTYPE="SHA-256" LABEL="premis1.xml">
  </mets:mdRef>
</mets:digiprovMD>
<mets:digiprovMD ID="uuid-48C18DD8-2561-4315-AC39-F941CBB138B3"
CREATED="2018-04-24T14:47:52.783+01:00">
  <mets:mdRef LOCTYPE="URL" type="simple"
href="metadata/preservation/premis2.xml" MDTYPE="PREMIS:OBJECT"
MDTYPEVERSION="3.0" MIMETYPE="text/xml" SIZE="2854"
CREATED="2018-04-24T14:37:52.783+01:00"
CHECKSUM="d1dfa585dcc9d87268069dc58d5e47956434ec3db4087a75a3885d287f15126f"
CHECKSUMTYPE="SHA-256" LABEL="premis2.xml">
    </mets:mdRef>
  </mets:digiprovMD>
  <mets:digiprovMD ID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943G"
CREATED="2018-04-24T14:37:52.783+01:00">
    <mets:mdRef LOCTYPE="URL" type="simple"
href="metadata/preservation/premis3.xml" MDTYPE="PREMIS:EVENT" MDTYPEVERSION="3.0"
MIMETYPE="text/xml" SIZE="1211" CREATED="2018-04-24T14:37:52.783+01:00"
CHECKSUM="8aa278038dbad54bbf142e7d72b493e2598a94946ea1304dc82a79c6b4bac3d5"
CHECKSUMTYPE="SHA-256" LABEL="premis1.xml">
      </mets:mdRef>
    </mets:digiprovMD>
    <mets:digiprovMD ID="uuid-48C18DD8-2561-4315-AC39-F941CBB138B4"
CREATED="2018-04-24T14:47:52.783+01:00">
      <mets:mdRef LOCTYPE="URL" type="simple"
href="metadata/preservation/premis4.xml" MDTYPE="PREMIS:OBJECT"
MDTYPEVERSION="3.0" MIMETYPE="text/xml" SIZE="2854"
CREATED="2018-04-24T14:37:52.783+01:00"
CHECKSUM="d1dfa585dcc9d87268069dc58d5e47956434ec3db4087a75a3885d287f15126f"
CHECKSUMTYPE="SHA-256" LABEL="premis2.xml">
        </mets:mdRef>
      </mets:digiprovMD>
    </mets:amdSec>
    <mets:fileSec ID="uuid-CA580D47-8C8B-4E91-ABD5-142EBBE15B84">
      <mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86H"
USE="Documentation">
        <mets:file ID="uuid-0C0049CA-6DE0-4A6D-8699-7975E4046A81"

```

```

MIMETYPE="application/vnd.openxmlformats-officedocument.wordprocessingml.document"
SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00"
CHECKSUM="91B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557"
CHECKSUMTYPE="SHA-256">
    <mets:FLocat LOCTYPE="URL" type="simple" href="documentation/File.docx">
    </mets:FLocat>
</mets:file>
<mets:file ID="uuid-0C0049CA-6DE0-4A6D-8699-7975E4046A82"
MIMETYPE="application/vnd.openxmlformats-officedocument.wordprocessingml.document"
SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00"
CHECKSUM="91B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557"
CHECKSUMTYPE="SHA-256">
    <mets:FLocat LOCTYPE="URL" type="simple" href="documentation/File2.docx">
    </mets:FLocat>
</mets:file>
</mets:fileGrp>
<mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86F" USE="Schemas">
    <mets:file ID="uuid-A1B7B0DA-E129-48EF-B431-E553F2977FD6"
MIMETYPE="text/xsd" SIZE="123917" CREATED="2018-04-24T14:37:49.617+01:00"
CHECKSUM="0BF9E16ADE296EF277C7B8E5D249D300F1E1EB59F2DCBD89644B676D66F72DCC"
CHECKSUMTYPE="SHA-256">
    <mets:FLocat LOCTYPE="URL" type="simple" href="schemas/ead2002.xsd">
    </mets:FLocat>
</mets:file>
</mets:fileGrp>
<mets:fileGrp ID="uuid-5811D494-6045-4741-924C-A1CFA340C277"
USE="Representations/preingest" CONTENTINFORMATIONTYPE="Access database">
    <mets:file ID="uuid-EE23344D-4F64-40C1-8E18-75839EF661FE" MIMETYPE="xml"
SIZE="1338744" CREATED="2018-04-24T14:37:49.617+01:00"
CHECKSUM="7176A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
CHECKSUMTYPE="SHA-256" ADMID="uuid-48C18DD8-2561-4315-AC39-F941CBB138B3
uuid-9124DA4D-3736-4F69-8355-EB79A22E943F">
    <mets:FLocat LOCTYPE="URL" type="simple"
href="representations/preingest/METS.xml">
    </mets:FLocat>
</mets:file>
</mets:fileGrp>
<mets:fileGrp ID="uuid-5811D494-6045-4741-924C-A1CFA340C278"
USE="Representations/submission/data" CONTENTINFORMATIONTYPE="SIARDDK"
ADMID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F

```

```

uuid-48C18DD8-2561-4315-AC39-F941CBB138B3">
  <mets:file ID="uuid-EE23344D-4F64-40C1-8E18-75839EF661FF"
MIMETYPE="application/xml" SIZE="1338744" CREATED="2018-04-24T14:37:49.617+01:00"
CHECKSUM="7176A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
CHECKSUMTYPE="SHA-256" ADMID="uuid-48C18DD8-2561-4315-AC39-F941CBB138B3
uuid-9124DA4D-3736-4F69-8355-EB79A22E943F">
  <mets:FLocat LOCTYPE="URL" type="simple"
href="representations/Submission/METS.xml">
  </mets:FLocat>
</mets:file>
</mets:fileGrp>
<mets:fileGrp ID="uuid-5811D494-6045-4741-924C-A1CFA340C279"
USE="Representations/ingest/data" CONTENTINFORMATIONTYPE="SIARD1"
ADMID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943G
uuid-48C18DD8-2561-4315-AC39-F941CBB138B4">
  <mets:file ID="uuid-EE23344D-4F64-40C1-8E18-75839EF661FG"
MIMETYPE="application/xml" SIZE="1338744" CREATED="2018-04-24T14:37:49.617+01:00"
CHECKSUM="7176A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
CHECKSUMTYPE="SHA-256" ADMID="uuid-48C18DD8-2561-4315-AC39-F941CBB138B3
uuid-9124DA4D-3736-4F69-8355-EB79A22E943F">
  <mets:FLocat LOCTYPE="URL" type="simple"
href="representations/ingest/METS.xml">
  </mets:FLocat>
  </mets:file>
</mets:fileGrp>
</mets:fileSec>
<mets:structMap ID="uuid-1465D250-0A24-4714-9555-5C1211722FB8" TYPE="PHYSICAL"
LABEL="CSIP StructMap">
  <mets:div ID="uuid-638362BC-65D9-4DA7-9457-5156B3965A18"
LABEL="uuid-4422c185-5407-4918-83b1-7abfa77de182">
    <mets:div ID="uuid-A4E1C5B6-CD9B-43EF-8F0C-3FD3AB688F81" LABEL="Metadata"
ADMID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F
uuid-48C18DD8-2561-4315-AC39-F941CBB138B3
uuid-9124DA4D-3736-4F69-8355-EB79A22E943G
uuid-48C18DD8-2561-4315-AC39-F941CBB138B4"
DMDID="uuid-906F4F12-BA52-4779-AE2C-178F9206111F">
  </mets:div>
  <mets:div ID="uuid-26757DC2-4C0F-4431-85B5-5943D1AB5CA3" LABEL="Schemas"
CONTENTIDS="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86F">
  </mets:div>

```



```

    <mets:div ID="uuid-35CB3341-D731-4AC3-9622-DB8901CD6737"
LABEL="representations/preingest"
CONTENTIDS="uuid-5811D494-6045-4741-924C-A1CFA340C277">
    <mets:mptr LOCTYPE="URL" type="simple"
href="representations/preingest/METS.xml">
    </mets:mptr>
  </mets:div>
  <mets:div ID="uuid-35CB3341-D731-4AC3-9622-DB8901CD6736"
LABEL="representations/submission"
CONTENTIDS="uuid-5811D494-6045-4741-924C-A1CFA340C278">
    <mets:mptr LOCTYPE="URL" type="simple"
href="representations/submission/METS.xml">
    </mets:mptr>
  </mets:div>
  <mets:div ID="uuid-35CB3341-D731-4AC3-9622-DB8901CD6738"
LABEL="representations/ingest"
CONTENTIDS="uuid-5811D494-6045-4741-924C-A1CFA340C279">
    <mets:mptr LOCTYPE="URL" type="simple"
href="representations/ingest/METS.xml">
    </mets:mptr>
  </mets:div>
</mets:div>
</mets:structMap>
</mets:mets>

```

## Appendix B: External Schema and Vocabularies

### External Schema

---

#### E-ARK CSIP METS Extension

Location:

<https://dilcis.eu/XML/METS/CSIPExtensionMETS/CSIPExtensionMETS.xsd>

Context: XML-schema for the attributes added by CSIP

Note:

An extension schema with the added attributes for use in this profile.

The schema is used with a namespace prefix of csip

#### PREMIS

Location: <http://www.loc.gov/standards/premis/>

Context: Used for preservation metadata

Note:

A rule set for use with this profile is under development.

### Controlled Vocabularies

---

#### Content information type specification name

Maintained By: DILCIS Board

Location: <http://earkcsip.dilcis.eu/schema/>

Context: Used in @csip:CONTENTINFORMATIONTYPE

Description:

Describes the specific E-ARK content information type names supported or maintained in this METS profile.

## **Content information type declaration**

Maintained By: DILCIS Board

Location: <http://earkcsip.dilcis.eu/schema/>

Context: Used in mets/@type

Description:

Describes the broad information type classification

## **OAIS Package type**

Maintained By: DILCIS Board

Location: <http://earkcsip.dilcis.eu/schema/>

Context: Used in @csip:OAISPACKAGETYPE

Description:

Describes the OAIS type the package belongs to in the OAIS reference model.

## **Note type**

Maintained By: DILCIS Board

Location: <http://earkcsip.dilcis.eu/schema/>

Context: Used in @csip:NOTETYPE

Description:

Describes the type of a note for an agent.

## **Other agent type**

Maintained By: DILCIS Board

Location: <http://earkcsip.dilcis.eu/schema/>

Context: Used in metsHdr/agent/@OTHERTYPE

Description:

Describes the other agent types supported by the profile

## **Identifier type**

Maintained By: Library of Congress

Location: <http://id.loc.gov/vocabulary/identifiers.html>

Context: Used in metsHdr/altRecordID/@TYPE

Description:

Describes the type of the identifier.

## **dmdSec status**

Maintained By: DILCIS Board

Location: <http://earkcsip.dilcis.eu/schema/>

Context: Used in dmdSec/@STATUS

Description:

Describes the status of the descriptive metadata section (dmdSec) which is supported by the profile.

## **IANA media types**

Maintained By: IANAs

Location: <https://www.iana.org/assignments/media-types/media-types.xhtml>

Context: Used in @MIMETYPE

Description:

Describes the mime type of a referenced file.

## **File group names**

Maintained By: DILCIS Board

Location: <http://earkcsip.dilcis.eu/schema/>

Context: Used in fileGrp/@USE

Description:

Describes the uses of the file group (fileGrp) that are supported by the profile.

Own names should be placed in an own extending vocabulary.

## **Structural map typing**

Maintained By: DILCIS Board

Location: <http://earkcsip.dilcis.eu/schema/>

Context: Used in structMap/@TYPE

Description:

Describes the type of the structural map (structMap) that is supported by the profile.

Own types should be placed in an own extending vocabulary.

## **Structural map label**

Maintained By: DILCIS Board

Location: <http://earkcsip.dilcis.eu/schema/>

Context: Used in structMap/@TYPE

Description:

Describes the label of the structural map that is supported by the profile.

Own labels should be placed in an own extending vocabulary.