# DELIVERABLE

**Project Acronym:** **E-ARK**
**Grant Agreement Number:** **620998**
**Project Title:** **European Archival Records and Knowledge Preservation**

**DELIVERABLE DETAILS**

| | |
|---|---|
| **DELIVERABLE REFERENCE NO.** | **4.3** |
| **DELIVERABLE TITLE** | **E-ARK AIP pilot specification** |
| **REVISION** | **1.10 January 29th 2016** |

**AUTHOR(S)**

| Name(s) | Organisation(s) |
|---|---|
| Sven Schlarb | AIT, Austrian Institute of Technology |
| Jan Rörden | AIT, Austrian Institute of Technology |
| João Cardoso | IST, Instituto Superior Técnico |
| Ricardo Vieira | IST, Instituto Superior Técnico |
| Janet Delve | UoB, University of Brighton |
| Richard Healey | UPHEC, University of Portsmouth |
| Gregor Zavrsnik | MC, Slovenian National Archives |

**REVIEWER(S)**

| Name(s) | Organisation(s) |
|---|---|

| | |
|---|---|
| Tarvo Kärberg | NAE, National Archives of Estonia |
| Kuldar Aas | NAE, National Archives of Estonia |
| Janet Delve | UoB, University of Brighton |

| **Project co-funded by the European Commission within the ICT Policy Support Programme**<br>**Dissemination Level** | | |
|---|---|---|
| **P** | **Public** | **x** |

# REVISION HISTORY AND STATEMENT OF ORIGINALITY

**Submitted Revisions History**

| Revision No. | Date | Authors(s) | Organisation | Description |
|---|---|---|---|---|
| 0.01 | July 15th, 2015 | Sven Schlarb | AIT, Austrian Institute of Technology | Original version submitted for internal review |
| 0.02 | December 1st, 2015 | Ricardo Vieira, João Cardoso | IST, Instituto Superior Técnico | Initial version of MoReq section |
| 0.03 | December 7th, 2015 | Janet Delve | UoB, University of Brighton | Initial version of OLAP section |
| 0.04 | January 12th, 2016 | Jan Rörden | AIT, Austrian Institute of Technology | Appendix with examples added |
| 1.00 | January 12th, 2016 | Sven Schlarb | AIT, Austrian Institute of Technology | Draft version for review |
| 1.01 | January 12th, 2016 | Sven Schlarb | AIT, Austrian Institute of Technology | Contributions to section 4 adapted |
| 1.02 | January 20th, 2016 | Sven Schlarb | AIT, Austrian Institute of Technology | Minor correction of code example in section 3.3.2.10. |
| 1.03 | January 21th, 2016 | Sven Schlarb | AIT, Austrian Institute of Technology | Adaptions according to comments from reviewer Tarvo Kärberg. |
| 1.04 | January 25th, 2016 | Sven Schlarb | AIT, Austrian Institute of Technology | Adaptions according to comments from reviewer Kuldar Aas. |
| 1.05 | January 26th, 2016 | Jan Rörden | AIT, Austrian Institute of Technology | Minor changes corrections of code examples. |
| 1.06 | January 27th, 2016 | Janet Delve | UoB, University of Brighton | OLAP section extended. |
| 1.07 | January 27th 2016 | Richard Healey | UPHEC, University of Portsmouth | OLAP section extended. |
| 1.08 | January 29th 2016 | Gregor Zavrsnik | MC, Slovenian National Archives | Geodata section extended |
| 1.09 | January 29th 2016 | Janet Delve | UoB, University of Brighton | Final Proof Read |
| 1.10 | January 29th 2016 | Sven Schlarb | AIT, Austrian Institute of Technology | Integrating final comments |

# TABLE OF CONTENTS

**1 Introduction .... 9**

**2 Preliminary definitions and remarks .... 10**

- 2.1 Representations .... 10
- 2.2 Logical and physical container of the AIP .... 10
- 2.3 Structural division of the AIP .... 11
- 2.4 Authenticity of the original submission .... 11
- 2.5 Version of an AIP .... 12
- 2.6 Cardinality of the SIP to AIP transformation .... 12

**3 AIP format specification .... 13**

- 3.1 E-ARK IP structure .... 13
  - 3.1.1 General IP structure .... 13
  - 3.1.2 Representations structure .... 15
- 3.2 E-ARK AIP structure .... 18
  - 3.2.1 AIP representations .... 18
  - 3.2.2 Changing metadata of the original submission .... 20
  - 3.2.3 Parent-Child relationship .... 21
- 3.3 E-ARK AIP metadata .... 23
  - 3.3.1 Structural metadata .... 23
  - 3.3.2 Preservation metadata .... 30
- 3.4 E-ARK Packaging format .... 40
  - 3.4.1 Package manifest .... 40

**4 E-ARK digital object classes .... 42**

- 4.1 OLAP (Online Analytical Data Processing) .... 42
- 4.2 MoReq2010® .... 49
  - 4.2.1 MoReq2010 Interoperability .... 51
  - 4.2.2 MCRS in relation to the AIP format .... 52
- 4.3 Geodata .... 53
  - 4.3.1 Geodata in relation to the AIP .... 55

**5 Appendices .... 61**

- 5.1 Appendix A - METS.xml referencing representation METS.xml files .... 61
- 5.2 Appendix B - METS.xml describing a representation .... 63
- 5.3 Appendix C - PREMIS.xml describing events on package level .... 64
- 5.4 Appendix D - PREMIS.xml describing migration events (representation level) .... 70

# List of Figures

Figure 1: General E-ARK IP structure ........ 14
Figure 2: Minimum E-ARK IP structure requirements ........ 15
Figure 3: One METS.xml file in the root of the IP references all metadata and data files ........ 15
Figure 4: Root METS.xml file references METS files of the different representations ........ 16
Figure 5 Example of an E-ARK IP ........ 17
Figure 6: AIP representations ........ 18
Figure 7: AIP representations ........ 19
Figure 8: AIP using representation-based division of METS.xml files ........ 20
Figure 9: METS.xml files in the AIP's "Metadata/submission" directory have priority over the ones contained in the original submission ........ 21
Figure 10: Parent-child relationship between AIPs ........ 21
Figure 11: Parent-child aggregation submission process ........ 22
Figure 12: New version of a parent-AIP ........ 23
Figure 14: Entity Relationship Diagram (ERD) ........ 43
Figure 15: Simple Fact Table ........ 44
Figure 16: Simple OLAP cube ........ 45
Figure 17: Slicing an OLAP cube ........ 46
Figure 18: Dicing an OLAP cube ........ 46
Figure 19: Drill (Roll) up / Drill down an OLAP cube ........ 47
Figure 20: MoReq2010 System Services ........ 50
Figure 21: Structured content of an XML export datafile ........ 51
Figure 22: A Large XML export may be split across multiple data files ........ 52
Figure 23: Geodata is made of graphical representation and attribute tables ........ 54
Figure 24: Geodata representation within the AIP ........ 55

# List of Tables

Table 1: New version of a parent-AIP ........ 24
Table 2: Attributes of the file element ........ 25

# Code Listings

Listing 1: METS root element example with namespace and namespace location definitions ........ 25
Listing 2: Example of a file in the fileSec as child of a fileGroup element (long attribute values replaced by "..." for better readability) ........ 25
Listing 3: Compressed file ........ 26
Listing 4: Linking to an EAD XML descriptive metadata file ........ 27
Listing 5: Linking to an EAD XML descriptive metadata file ........ 28
Listing 6: Structural map referencing METS.xml files of the different representations ........ 28
Listing 7: Obligatory E-ARK structural map ........ **Error! Bookmark not defined.**
Listing 8: Using a structMap to reference the parent AIP ........ **Error! Bookmark not defined.**
Listing 9: Using a structMap to reference the parent AIP ........ 30
Listing 10: Object identifier ........ 33
Listing 11: Hashsum (value shortened) ........ 34
Listing 12: Pronom Unique Identifier ........ 34
Listing 13: JHove digital object characterisation ........ 34
Listing 14: Original name ........ 35
Listing 15: Storage description ........ 35

Listing 16: Relationship ........ 35
Listing 17: Rights statement ........ 36
Listing 18: Event identifier ........ 36
Listing 19: Event date/time ........ 36
Listing 20: Link to agent/object ........ 37
Listing 21: Migration event ........ 38
Listing 23: Premis rights statement ........ 39
Listing 22: Software as an agent ........ 40
Listing 24: Manifest file ........ 41
Listing 25: Example of an SLD file ........ 58

# Executive Summary

This deliverable D4.3 “E-ARK AIP pilot specification” presents the E-ARK AIP format specification as it will be used by the pilots (implementations in pilot organizations). The deliverable is a follow-up version of E-ARK deliverable D4.2 “E-ARK AIP draft specification”.[1]

The document describes the structure, metadata, and physical container format of the E-ARK AIP, a container which is the result of converting an E-ARK Submission Information Package (SIP) into the E-ARK Archival Information Package (AIP). The conversion will be implemented in the Integrated Platform[2] as part of the component *earkweb*.[3]

---

[1] http://www.eark-project.com/resources/project-deliverables/18-d42-e-ark-aip-draft-specification

[2] See E-ARK deliverable 6.2 “Integrated Platform Reference Implementation”, section 6.

[3] https://github.com/eark-project/earkweb; the actual SIP to AIP conversion component consists of a set of tasks in https://github.com/eark-project/earkweb/blob/master/workers/tasks.py.

# 1 Introduction

To briefly recall the three types of information packages according to OAIS[4], there is the Submission Information Package (SIP) which is used to submit digital objects to a repository system; the Archival Information Package (AIP) which allows the transmission of a package to the repository, and its storage over the long-term; and the Dissemination Information Package (DIP) which is used to disseminate digital objects to the requesting user.

The current document D4.3 "E-ARK AIP pilot specification" is the follow-up version of E-ARK deliverable D4.2 "E-ARK AIP draft specification"[5] and will be followed by version D4.4 which is the final iteration of the E-ARK AIP format specification. The E-ARK AIP[6] format is "a specialization of the Information Package"[7], the E-ARK IP. The format is supported by the E-ARK SIP to AIP conversion component implemented as part of the integrated prototype implementation *earkweb*.[8]

The format specification consists of a set of conventions allowing the storage of digital objects together with metadata in a repository system, with the purpose of safeguarding digital objects over the long term.

Key objectives are:

- To define this format in a way that it is suitable for a wide variety of data types, such as document and image collections, archival records, databases or geographical data.
- To ensure the format is also suitable to store large quantities of data.
- To include mechanisms to support integrity and authenticity of data and metadata

As already pointed out in the previous iteration of this document[9], the AIP format is not only a description of the set of rules specifying how an AIP is stored, but as the AIP actually represents an entity that can be augmented or changed over time, it is necessary to stipulate how these changes can be tracked. For this reason, the question of how metadata can be used to record preservation information over time plays a key role for the AIP.

The structural parts of the AIP format presented here are defined in an abstract manner and can be implemented in different ways in a variety of systems. However, there are also decisions and recommendations regarding the concrete format implementation, especially as the use of structural and preservation metadata is concerned. In particular, this means that the hierarchical structure of the AIP and the way that

---

[4] Reference Model for an Open Archival Information System (OAIS); Retrieved from http://public.ccsds.org/publications/archive/650x0m2.pdf, in the following we are always referring to this version 2 of the OAIS which was published in June 2012 by CCSDS as "magenta book" (ISO 14721:2012). The document is cited using the abbreviation OAIS.

[5] Deliverable D4.2: "E-ARK AIP Draft Specification", available at http://www.eark-project.com/resources/project-deliverables/18-d42-e-ark-aip-draft-specification

[6] In the following the abbreviation AIP refers to the E-ARK AIP unless otherwise stated.

[7] OAIS, p. 4.

[8] https://github.com/eark-project/earkweb; the actual SIP to AIP conversion component consists of a set of tasks in https://github.com/eark-project/earkweb/blob/master/workers/tasks.py.

[9] Deliverable D4.2

content and metadata are organized is defined by a directory hierarchy in a file system. Furthermore, it means that package metadata is prescribed in the form of concrete structural and preservation metadata formats, together with prescriptive and optional guidelines which show how the corresponding metadata elements have to be used.

# 2 Preliminary definitions and remarks

## 2.1 Representations

The concept of “representations” is crucial in the context of E-ARK and it is generally used according to the definition given in the context of the PREMIS digital preservation metadata standard[10]:

> *"The set of files, including structural metadata, needed for a complete and reasonable rendition of an Intellectual Entity. For example, a journal article may be complete in one PDF file; this single file constitutes the representation. Another journal article may consist of one SGML file and two image files; these three files constitute the representation. A third article may be represented by one TIFF image for each of 12 pages plus an XML file of structural metadata showing the order of the pages; these 13 files constitute the representation."*[11]

The significance of this concept for the AIP format is due to the fact that a representation can, on the one hand, be created and submitted to the repository as part of the SIP, and, on the other hand, it can be created during ingest when the AIP is generated or when additional representations are added to an existing AIP.

It should also be mentioned that representations can be derived from each other; this is typically the case if digital objects of a representation are migrated and the derived representation consists of the digital objects in another format. However, a new representation is, in our understanding, not necessarily the result of a file format migration, but it can also be a set of instructions presented as part of representation metadata on how to create an emulation environment to render a set of files.

## 2.2 Logical and physical container of the AIP

In line with OAIS, we call the logical container of the AIP the complete set of digital objects and metadata representing the conceptual entity as a whole. The conceptual entity must be distinguished from the physical representation of one or possibly more physical containers which represent one conceptual entity.

From the point of view of preserving the integrity of the AIP, the ideal case is that the logical AIP representing the intellectual entity is packaged as one single physical container. In this way, in case of a disaster recovery, the physical container has all the information required to interpret and render the contained representations. In reality this is not always possible because the size of the physical container could become

---

[10] Preservation Metadata: Implementation Strategies, http://www.loc.gov/standards/premis

[11] Introduction and Supporting Materials from PREMIS Data Dictionary, http://www.loc.gov/standards/premis/v2/premis-report-2-1.pdf, p. 7.

extremely large, and this is the reason for proposing the divided METS structure described in section 3.1.2.2 which makes it easier to manage representations or representation parts separately. For both cases, the representation-based and the size-based splitting, described in section 3.1.2.3, the purpose is to store each of the constituent parts as one physical container.

## 2.3 Structural division of the AIP

One of the basic decisions in the E-ARK project is to use *METS*[12] as the metadata standard to describe the structure of the AIP – in line with the decision related to the other package types, the SIP and the DIP.

On the one hand, E-ARK pays special attention to the situation that it might not be possible to store all representations of an intellectual entity in one physical container, or that even a single representation must be divided so that it can be stored on long-term storage media. In order to make it easier to manage representations or representation parts separately, E-ARK proposes a divided METS structure by using a *METS.xml* file in the root directory of the AIP which references *METS.xml* files of the individual representations. This structure lays the ground for addressing the practical requirement of distributing parts of the intellectual entity over a sequence of physical containers representing a logical AIP. Even though this puts the integrity of the AIP at risk - because in case of disaster recovery the physical container does not represent the complete intellectual entity and dependencies to another (lost) physical container can potentially make it impossible to interpret, understand, or render the content - it is a necessary measure if the amount of data exceeds the capacity limitation of long-term storage media.

On the other hand, there might be the need to put a constraint on the size of the physical containers of the AIPs to allow storing them on long-term storage media more easily. In this case, E-ARK proposes a "compound" structure using a single *METS.xml* file that contains all references to the data and metadata of the package.

## 2.4 Authenticity of the original submission

One of the core requirements regarding the SIP to AIP conversion is to safeguard the authenticity of the original submission. This does not necessarily mean that the SIP content stored in the AIP must be identical to the original submission, as it is also clearly stated in OAIS:

> *"An OAIS is not always required to retain the information submitted to it in precisely the same format as in the SIP. Indeed, preserving the original information exactly as submitted may not be desirable."*[13]

However, in the E-ARK project it was decided to keep the original submission "as is" and not to change the structure or any data or metadata contained in the SIP. Instead, the E-ARK project defines a structure and provides a set of rules as to how changes to metadata or content can be applied and tracked without having to change the original submission. The main argument for this approach is that the authenticity of the original submission can be guaranteed more easily.

---

[12] Metadata Encoding and Transmission Standard, http://www.loc.gov/standards/mets/

[13] OAIS, p. 4-52 (95).

## 2.5 Version of an AIP

While the AIP always describes the same unaltered conceptual entity, the way in which this information is represented may change. Therefore the E-ARK AIP format describes the means to record the provenance from the time of the first submission, and also during the life-cycle of the AIP.

For the purpose of the AIP format specification, the "AIP version" concept is used in E-ARK as defined by OAIS:

> *"AIP Version: An AIP whose Content Information or Preservation Description Information has undergone a Transformation on a source AIP and is a candidate to replace the source AIP. An AIP version is considered to be the result of a Digital Migration."*[14]

A new version of an AIP contains one or more new representations which according to the E-ARK understanding can be either the result of a digital migration or information that enables the creation of an emulation environment to render a representation. The result of this operation is the creation of a new version of the AIP. [15]

## 2.6 Cardinality of the SIP to AIP transformation

While "within the OAIS one or more SIPs are transformed into one or more Archival Information Packages (AIPs) for preservation"[16], the E-ARK SIP to AIP conversion will only consider the transformation of one single SIP into one single AIP. As a consequence, if a given package needs to be divided into smaller parts, this must be done before the actual SIP to AIP conversion starts.

The separation of parts into distinguishable units that belong to the same intellectual entity is achieved by creating a series of SIPs which are aggregated in a later step by creating an AIP[17] that logically defines the relationship of the AIPs belonging together. This is independent from the fact that the separation of AIPs occurs because independent parts are conceptually grouped together, or there is the technical necessity to separate either different representations or parts of a representation into different AIPs.

---

[14] OAIS, p. 1-9 (19).

[15] The concept "generation" which was introduced in D4.2 is no longer used. In D4.2 this concept denoted the case that "[...] the AIP may after some time be repackaged and result in another physical container. Two containers, which contain representations of the information originating from one SIP, we call generations of the information package", Deliverable D4.2: "E-ARK AIP Draft Specification", http://www.eark-project.com/resources/project-deliverables/18-d42-e-ark-aip-draft-specification/file, p. 14. Note that according to the OAIS, only migration leads to a new version of the AIP. As adding an "emulation representation" basically means adding a set of metadata files without actually changing the content, this would be an "edit" according to OAIS terminology and therefore not result in a new version of the AIP. In E-ARK, however, the "emulation representation" is equivalent to the migration action in the sense that it adds a new representation to the AIP, and therefore leads to a new version.

[16] Reference Model for an Open Archival Information System (OAIS); Retrieved from http://public.ccsds.org/publications/archive/650x0m2.pdf, p. 2-8 (34).

[17] Ibidem, p. 4-41 (84).

# 3 AIP format specification

The purpose of the AIP format is to provide a logical and physical container for E-ARK Information Packages (IP)[18] which is adequate for preserving digital objects over a long time period while keeping track of any changes by recording the provenance of the IP.

The E-ARK AIP format will be explained with the help of textual descriptions, figures, and examples. The actual AIP format specification is defined as a set of requirements[19] that is derived from assumptions made in the corresponding context of this document.

The AIP format specification is divided into two parts. On the one hand, there is the structure and metadata specification which defines how the AIP is conceptually organized by means of a directory hierarchy in a file system and a set of metadata standards. And on the other hand, there is the physical container specification which defines the bit-level manifestation of the transferable entity. Both parts of the specification can be regarded as independent from each other which means that either can be used with another counterpart.

Before dealing with the actual AIP format we will describe the E-ARK IP structure which is the basis on top of which the AIP format is defined.

## 3.1 E-ARK IP structure

In this section we briefly describe the structure of the E-ARK IP which is defined by common rules for all information package types (SIP, AIP, DIP) specified by the "E-ARK common specification for information packages" [20] with a focus on the parts that are relevant for the AIP.

### 3.1.1 General IP structure

As already mentioned in section 2.1, the E-ARK IP format relies on the concept of "representations". Given an example of two distinguishable representations, Figure 1 shows the general structure of an E-ARK IP according to which metadata (folder "metadata") and data (folder "data") are strictly separated. It also shows that in this example the two data folders are divided into two different branches. These two branches separate "representations" of the same intellectual entity.

---

[18] In the following, the abbreviation "IP" is used to refer to a common set of standards and structure definitions which are valid for all types of information packages, i.e. independently from the fact if it is a Submission Information Package (SIP), Archival Information Package (AIP), or Dissemination Information Package (DIP).

[19] The requirements terminology is based upon RFC2119, "Key words for use in RFCs to indicate requirement levels", RFC 2119, S. Bradner, March 1997. Available at: http://www.ietf.org/rfc/rfc2119.txt

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

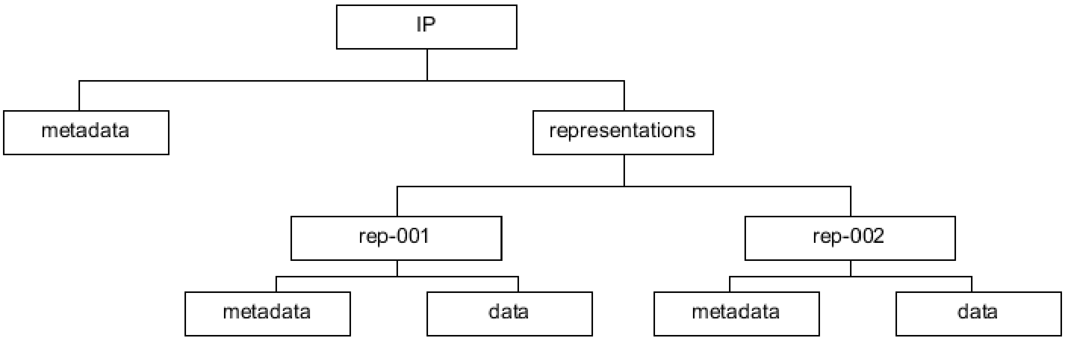


*Figure 1: General E-ARK IP structure*

Furthermore, Figure 1 shows that metadata can be stored either at the representation level or at the IP level. In section 3.2 it will be explained more in detail where different types of metadata can be stored. For now, it is sufficient to mention that descriptive, technical, preservation, and rights metadata can relate either to the IP as a whole or to individual representations.

Based on these assumptions, the following requirements to the general IP structure need to be taken into account in the context of the AIP format:

> ***Requirement 1****: An IP MUST contain a "representations" directory.*
>
> ***Requirement 2****: Representations MUST be stored in subdirectories of the "representations" directory. The names of these subdirectories can be chosen freely.*
>
> ***Requirement 3****: An IP MUST contain a "Metadata" directory.*
>
> ***Requirement 4****: The "Metadata" directory SHOULD be divided into "descriptive", "preservation", and "other" subdirectories for storing the corresponding category of metadata.*
>
> ***Requirement 5****: The root directory of the package must contain a METS.xml file which is created according to the rules defined in section 3.3.1.*

Following these requirements, the obligatory structure of an E-ARK IP is essentially as shown in Figure 2.[21]

---

[21] The variable name in curly brackets means that the name of this directory can be chosen freely.

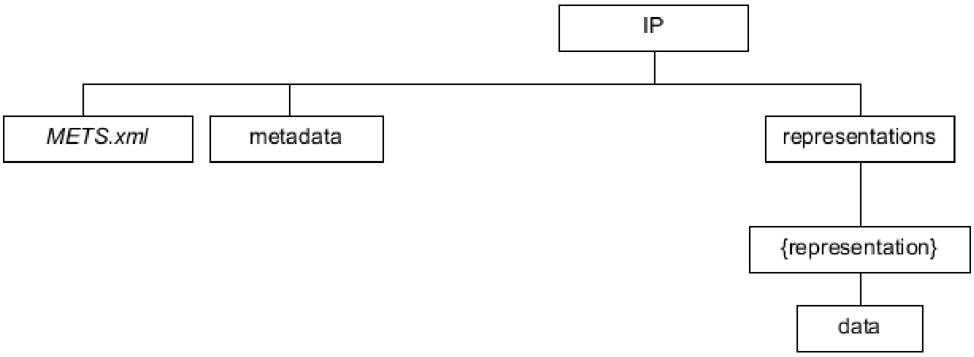


Figure 2: Minimum E-ARK IP structure requirements

## 3.1.2 Representations structure

The use of other components of the IP format depends on institutional preferences related to the use of structural metadata, and generally the type and amount of content that needs to be archived using this structure.

In the following we will use a concrete example to describe the two alternatives of using either a compound or a divided METS structure in more detail.

Let us assume that we have an IP with two representations, each of which consists of a set of three files. In the first representation all data files are in the Open Document Format (ODT) and in the second one - as a derivative of the first representation - all files are in the Portable Document Format (PDF).

### 3.1.2.1 Compound METS structure

The first option, as shown in Figure 3, is to have a single *METS.xml*[22] file that contains all references to metadata and data files available of the IP which is called “compound” or “simple” METS structure.

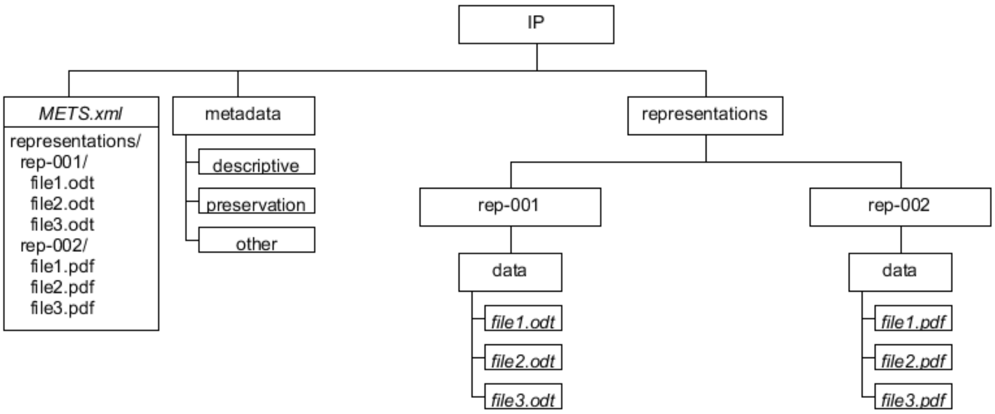


Figure 3: One METS.xml file in the root of the IP references all metadata and data files

Even though the number suffix of the directories “rep-001” and “rep-002” of the example shown in Figure 3 suggests an order of representations, there are no requirements regarding the naming of directories con-

---

[22] For the sake of simplicity, the examples show only references to data files and not to metadata files.

taining the representations. The order of representations, and the relations between them, is defined by the structural and preservation metadata, which will be, described more in detail in section 0. The "representations" directory is mandatory, also for IPs which contain only one representation.

#### 3.1.2.2 Divided METS structure

The second option, called "divided" or "full" METS structure, as shown in Figure 4, is to have separate *METS.xml* files for each representation and the *METS.xml* in the IP's root points to the *METS.xml* files of the representations. More concretely, the example shown in Figure 4 has a *METS.xml* file in the IP's root, which points to the *METS.xml* files "Representations/Rep-001/METS.xml" and "Representations/Rep-002/METS.xml". We will explain in section 3.3.1 more in detail how the referencing of *METS.xml* files must be implemented if this alternative is chosen.

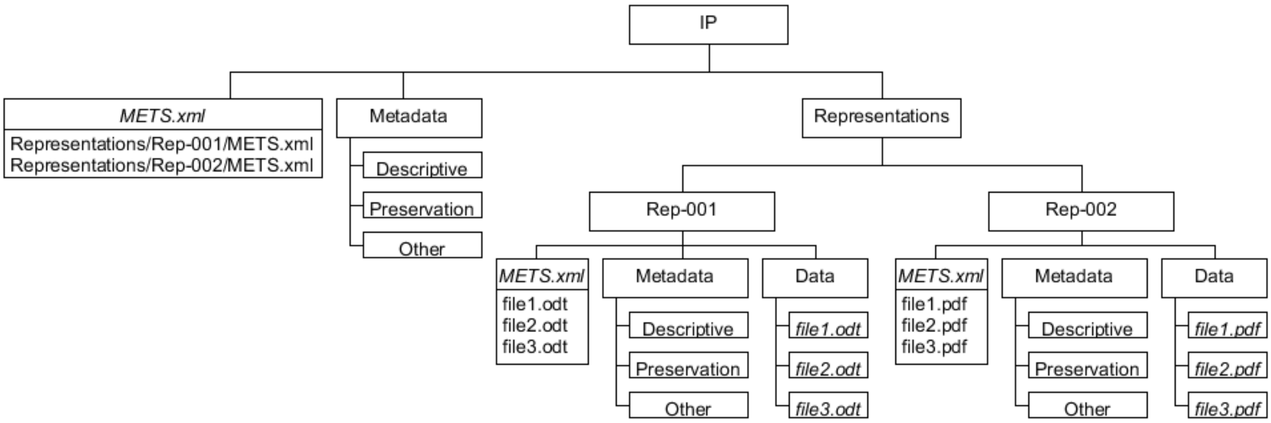


Figure 4: Root METS.xml file references METS files of the different representations

The reason why this alternative was introduced is that it makes it easier to manage very large representations (e.g. a binary database and the database in a vendor independent XML format as two separate representations) independently from each other.

#### 3.1.2.3 Representation-based vs. size-based division

As a corollary of this division method, we define, on the one hand, a *representation-based division* as the separation of representations in different directories under the "representations" folder as shown in the example of Figure 4. And, on the other hand, we define a *size-based division* as the separation of representation parts. To illustrate this, figure 5 shows an example where a set of files belongs to the same representation (here named "binary") and is referenced in two separate physical containers (here named {C1} and {C2} respectively). A key requirement when using size-based division of a representation is that there must not be any overlap in the structure of the representations, and that each sub-directory path must be unique across the containers where the representation parts together constitute a representation entity. Note that for this reason a numerical suffix is added to the representation METS.xml files, to avoid overriding representation METS.xml files when automatically merging the divided representation back into one single physical representation.

> ***Requirement 6****: If a representation is divided into parts, the representation component MUST use the same name in the different containers.*

> ***Requirement 7**: If a representation is divided into parts, there MUST not be any overlap in the structure of the representations and that each sub-directory path MUST be unique across the containers*

It must be noted that this size-based division method assumes that the separation into parts is based on criteria determined by the archivist.

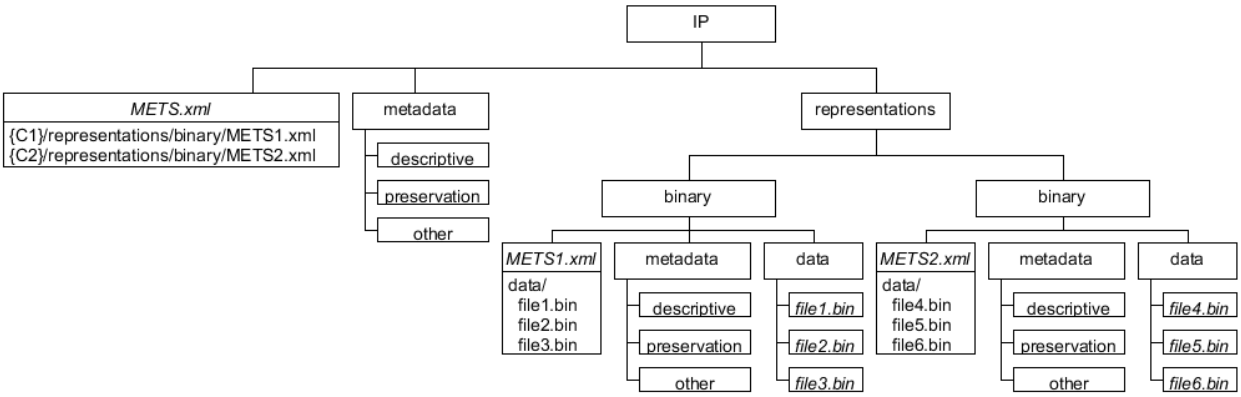


Figure 5 Example of an E-ARK IP

### 3.1.2.4 Requirements

There is the basic requirement that the root of an IP contains a *METS.xml* file, but it can be freely chosen if this *METS.xml* file is broken down into *METS.xml* parts for the different representations.

> ***Requirement 8**: The package root MUST contain a METS.xml file that either references metadata or data files or references other METS.xml files located in the corresponding representation directories under the "representations" directory.*

As already stated, only the "data" directory is obligatory for each representation. Additionally, there can be other directories at the representation level which are specified by the following requirements:

> ***Requirement 9**: Each representation MUST contain a "data" directory. The structure within this directory can be freely chosen.*
>
> ***Requirement 10**: A representation CAN contain a "Metadata" directory.*
>
> ***Requirement 11**: If a representation directory contains a "Metadata" directory, it SHOULD be divided into "descriptive", "preservation", and "other" subdirectories for storing the corresponding category of metadata.*
>
> ***Requirement 12**: The representation directory CAN contain a "Documentation" directory to store additional documents that explain the content available in the "data" directory.*
>
> ***Requirement 13**: The representation directory CAN contain a "Schemas" directory to store additional XML Schema files that are needed to validate XML documents contained in a representation.*

## 3.2 E-ARK AIP structure

Based on the IP format described in the previous section, the AIP format specifies a logical structure and guidelines for using METS and PREMIS metadata element to create E-ARK AIPs.

### 3.2.1 AIP representations

As described in section 3.1 in relation to an IP, one or several representations can be part of an SIP which is submitted to the repository. Additionally, the AIP must be able to include further representations either added during SIP to AIP conversion, or because a preservation measure needs to be performed on the AIP.

In principle, the AIP format offers a structure for storing the metadata and data of a SIP for the first time during ingest, and it allows the creation of documentation for any transformation of the content that might be applied during the lifecycle of the AIP that was created from the SIP.

To illustrate this with the help of an example, let Figure 6 shows the structure of an AIP where the original submission consists of two representations. The “submission” directory of the AIP contains the original submission “as is”, which means that neither data nor metadata is changed.

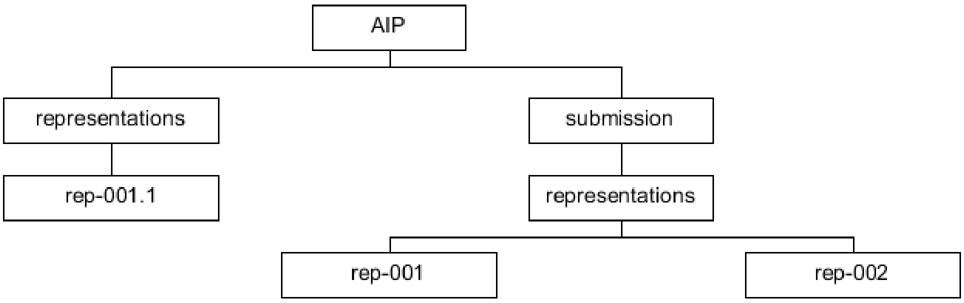


*Figure 6: AIP representations*

The following rule applies:

> ***Requirement 14****: The root directory of the AIP MUST contain a “submission” directory which contains the original submission.*

Let us now assume that during SIP to AIP conversion an additional representation is added to the AIP. Figure 7 illustrates an example where an additional “representations” directory exists as a sibling of the “submission” directory, which contains a new representation (rep-001.1), derived from one of the representations contained in the original submission (rep-001).

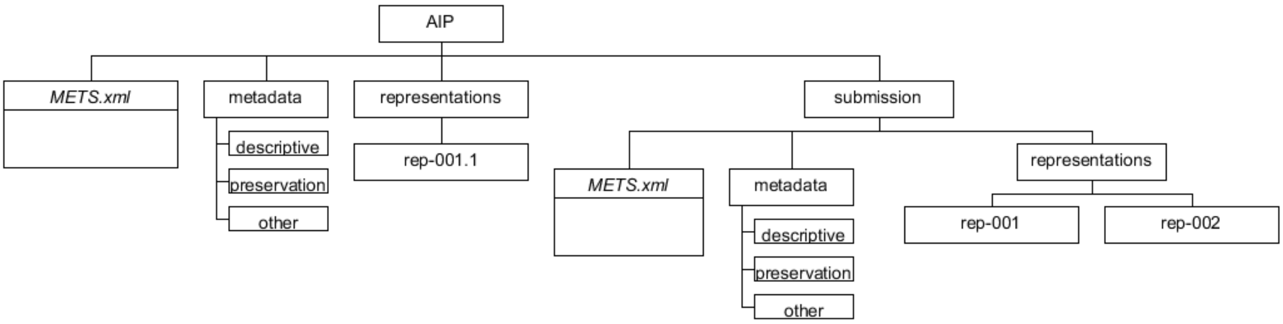


*Figure 7: AIP representations*

This leads to the following requirement regarding representations which are added during SIP to AIP conversion.

> ***Requirement 15****: If a new representation is added during ingest (SIP to AIP conversion) or created as an AIP preservation measure (AIP to AIP conversion), the root directory of the AIP MUST contain a "representations" directory. The same requirements as for the representations of an IP apply, namely requirements 9 to 13.*

Note that the three-digit number suffix following the name "Rep-" used in the example of Figure 7 indicates the order in time in which the representation of the original submission was created. And the additional number suffix after the dot indicates that the representation is a derivative of the representation identified by the three-digit number before the dot, i.e. "rep-001.1" is the first derivative of representation "rep-001". This is however for illustration purposes only; the naming of representations does not have to follow such logic.

The AIP is an extension of the IP format; therefore it must follow the basic structure of an IP. Figure 7 shows that the IP components, consisting of *METS.xml* file, "Metadata" and "representations" directories, are repeated on the AIP level. The extension of the AIP format is basically given by the fact that the AIP is an IP which can contain another IP (here an SIP) in the "submission" directory.

Note that the "representations" directory in the AIP root directory is optional; this is why requirement 15 is formulated as a conditional requirement. It means that this directory must only exist in case representations other than the ones originally submitted are added to the AIP.

> ***Requirement 16****: The AIP is an IP, therefore requirement 8 applies and the AIP root MUST contain a METS.xml file that either references all metadata and data files or it references other METS.xml files located in the corresponding representation directories of the AIPs or of the original submission's "representations".*

As a concrete example let us assume a policy stating that PDF documents must generally be converted to PDF/A.[23] Taking the premise formulated in section 2.4 into account that the original submission is not to be changed, the additional representation is added in a "Representation" directory in the root of the AIP as shown in figure 7. Note that this example uses a representation-based division of *METS.xml* files.

---

[23] http://pdf.editme.com/PDFA

Analogously to Figure 6 there are also two representations in the original submission shown in Figure 8. The first representation ("Rep-001") consists of a set of files in the Open Document Format (ODT) and the second one ("Rep-002") is a derivative of the first set of files in the Portable Document Format (PDF). As an example we assume that an institutional policy prescribes that every PDF document must be converted to PDF/A during SIP to AIP conversion. Therefore the second representation ("Rep-002") was converted to a set of PDF/A files and added to the AIP as an additional representation ("Rep-002.1").

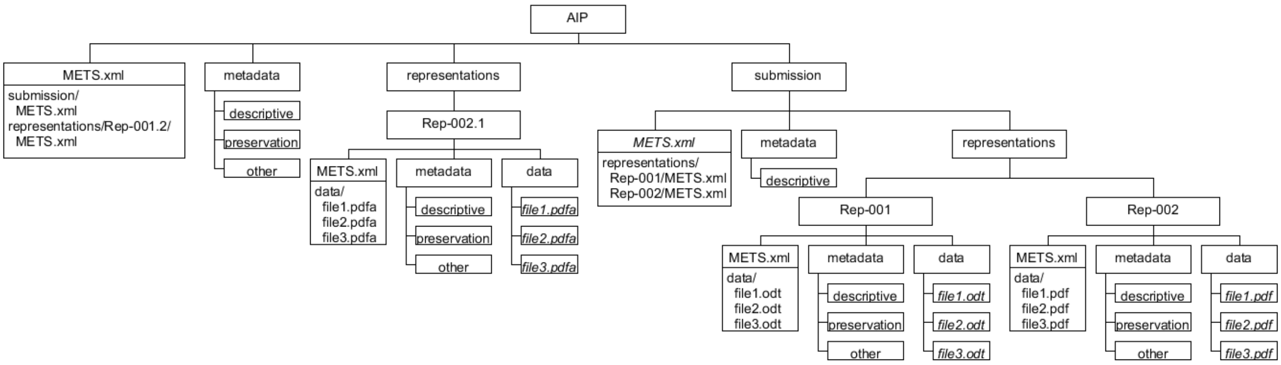


*Figure 8: AIP using representation-based division of METS.xml files*

The two representations of the original submission are located in the "submission/representations" directory of the AIP and the *METS.xml* file of the submission references the corresponding representation *METS.xml* files using a relative path to be resolved within the SIP. The root level *METS.xml* file of the AIP references the *METS.xml* file of the original submission (submission/representations/METS.xml) and the *METS.xml* file of the new representation (representations/Rep-002.1/METS.xml).

### 3.2.2 Changing metadata of the original submission

One consequence of the premise that the originally submitted SIP is not supposed to change is that the AIP level metadata directory might also contain metadata that relates to representations contained in the original submission.

As an example let us assume that we have to recalculate the checksum during SIP to AIP conversion and that the checksum is recorded as an attribute of the METS file element. As shown in Figure 9, the AIP's "Metadata" directory can - additionally to the existing metadata category directories - contain a "submission" directory with metadata files (here *METS.xml*) that by definition have priority over the ones contained in the original submission. This means that in case metadata needs to be updated, they must be placed into the root level metadata directory because metadata of the original submission is not allowed to be changed.

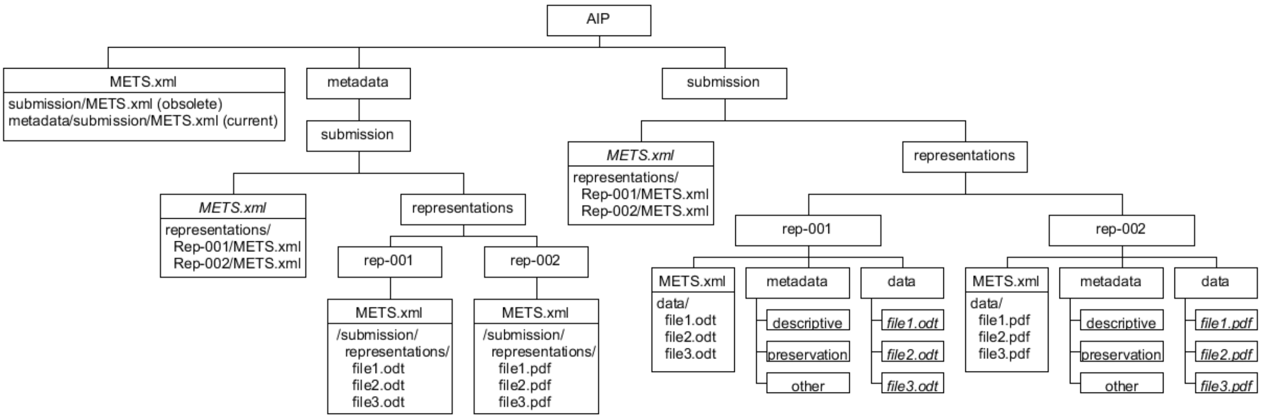


*Figure 9: METS.xml files in the AIP's "Metadata/submission" directory have priority over the ones contained in the original submission*

> ***Requirement 17**: Let <MDPath> be a sub-directory-path to a metadata file, then a metadata file under the "AIP/metadata/submission" directory MUST have priority over a metadata file under the "AIP/submission" directory so that **AIP/metadata/submission/<MDPath>** has priority over **AIP/submission/<MDPath>**.*

An example is shown in Figure 9 where the *METS.xml* file in the root of the AIP references an obsolete *METS.xml* file of the original submission and a current *METS.xml* file under "`metadata/submission`", i.e. the metadata file `AIP/metadata/submission/METS.xml` has priority over the metadata file `AIP/submission/METS.xml`. In this way users have the possibility to both consult initial metadata and the updated metadata.

### 3.2.3 Parent-Child relationship

As already pointed out, the divided METS structure was introduced to make the separation of representations or representation parts easier and allow the distribution of these components over a sequence of AIPs.

E-ARK allows the composition of a logical AIP to be expressed by a parent-child relationship between AIPs. It is a bidirectional relationship where each child-AIP bears the information to which parent-AIP they belong and, vice versa, the parent-AIP references the child-AIPs.

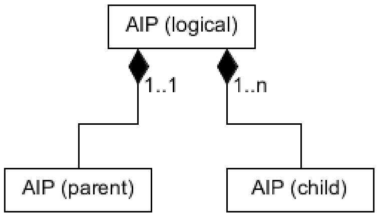


*Figure 10: Parent-child relationship between AIPs*

Even though this parent-child relationship could be used to create a hierarchical graph of AIPs, E-ARK only uses this method to aggregate representations or representation parts of the logical AIP.

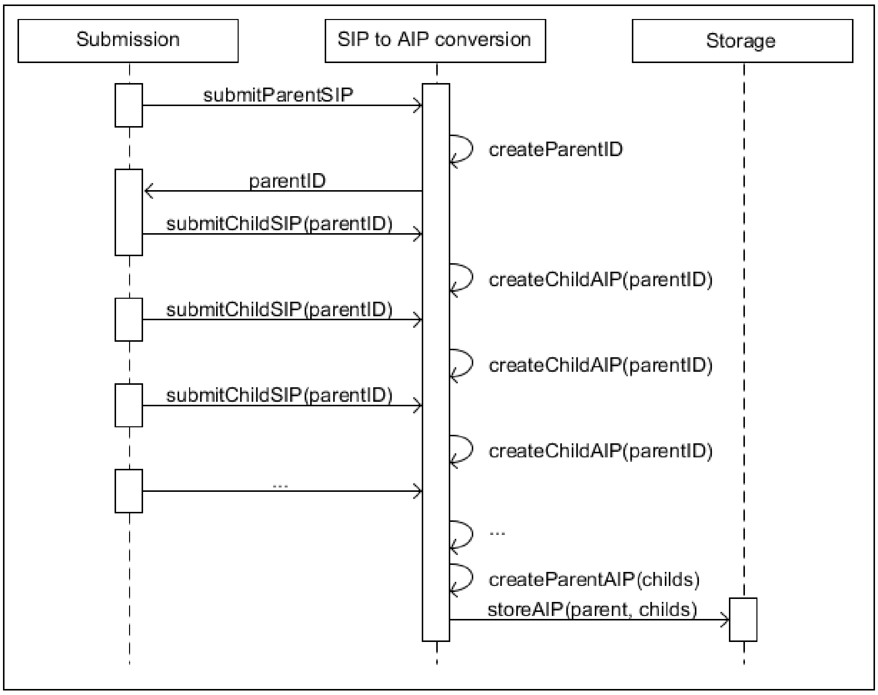


*Figure 11: Parent-child aggregation submission process*

Figure 11 illustrates how the creation of packages governed by a parent-child relationship can be achieved:

1. The submission of a package is processed up to the point where the identifier of the parent AIP (“parentID”) is created (operation “submitParentSIP”).
2. This SIP to AIP creation process remains open until the last child AIP is created.
3. A sequence of child SIPs can then be submitted and created as child-AIPs by indicating the parent identifier (“parentID”) generated previously.
4. After the last child-AIP was processed, the parent-AIP is created (operation “createParentAIP”) by indicating which child-AIPs belong to the AIP (“childs”).
5. Finally, the consistency of the logical AIP composition is verified and parent-AIP (“parent”) and all child-AIPs (“childs”) are stored (operation “storeAIP”).

Assuming that a new AIP (e.g. containing an additional representation) needs to be added after parent- and child-AIPs have been stored, the recreation of the whole logical AIP might be inefficient, especially if the AIPs are very large. For this reason, existing child-AIPs remain unchanged in case a new version of the parent-AIP is created. Only the new version of the parent-AIP has references to all child-AIPs as illustrated in Figure 12. As a consequence, in order to find all siblings of a single child-AIP it is necessary to get the latest version of the parent-AIP which implies the risk that the integrity of the logical AIP is in danger if the latest version of the parent-AIP is lost.

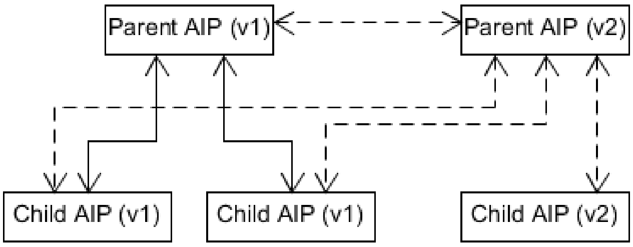


Figure 12: New version of a parent-AIP

The result of this process is a sequence of physical containers of child-AIPs plus one additional parent-AIP. The relation of the AIPs is expressed by means of structural metadata in the METS.xml files as described in the sections 3.3.1.6 and 3.3.1.7.

## 3.3 E-ARK AIP metadata

The AIP format specifies the use of structural and preservation metadata. Any type of additional metadata, such as descriptive metadata using Dublin Core or EAD, can be used.

In the following, XML elements are either enclosed between angle brackets (e.g. `<fileSpec>`) or addressed using XPath syntax (e.g. `/mets/metsHdr`). In the latter case a leading slash selects a node from the XML document root and the double slash ('//') selects nodes in the document from the current node that match the selection, no matter where they are. Also in line with the XPath syntax, element attributes have a leading '@' character. For example `//mets:file/@USE` denotes the '`USE`' attribute of a `<file>` element.

### 3.3.1 Structural metadata

Structural metadata is expressed by means of the METS standard. Some of the high level functions which the standard fulfils in the context of the AIP are the following.

- It provides an overview about the hierarchical structure of the AIP.
- It is an entry point for the AIP, i.e. the first entity to consult to know what an AIP contains and what entities it references.
- It references any metadata files describing the package as a whole as well as individual content files.
- It contains a complete list of digital objects contained in a package together with basic information to ensure the integrity of the digital objects.
- It establishes links between digital objects and metadata entities (both structural metadata and preservation metadata entities).
- It can hold information about different representations or representation parts belonging to the same intellectual entity.

This section has a focus on METS, therefore, if no namespace prefix is given, the element belongs to the METS default namespace[24].

### 3.3.1.1 METS root element

#### *3.3.1.1.1 METS identifier*

Each METS document must be assigned a persistent and (ideally globally) unique identifier. Possible identifier schemes are amongst others: OCLC Purls[25], CNRI Handles[26], DOI[27]. Alternatively, it is possible to use a UUID as a locally unique identifier.[28].

Using this identifier, the system must be able to retrieve the corresponding package from the repository.

As stated in Deliverable 4.2, it is recommended "using as a prefix an internationally recognized standard identifier for the institution from which the SIP originates. This may lead to problems with smaller institutions, which do not have any such internationally recognized standard identifier. We propose in that case, to start the prefix with the internationally recognized standard identifier of the institution, where the AIP is created, augmented by an identifier for the institution from which the SIP originates."[29]

In the context of the implementation as part of the E-ARK Integrated Platform, a UUID is used as identifier of an intellectual entity. The prefix "`urn:uuid:`" is used to state that the identifier can be used to locate physical packages as well. For example, if the package identifier value is "123e4567-e89b-12d3-a456-426655440000" this would be the value of the METS root element's 'OBJID' attribute:

```
/mets/@OBJID="urn:uuid:123e4567-e89b-12d3-a456-426655440000"
```

#### *3.3.1.1.2 Namespace and namespace schema definitions*

> ***Requirement 18**: The METS document MUST use at least the namespace and namespace schema definitions defined in table 1.*

*Table 1: Attributes of the METS root element*

| Attribute | Description | Value |
|---|---|---|
| //mets/@xmlns | METS-Namespace | http://www.loc.gov/METS/ |
| //mets/@xmlns:xlink | Xlink-Namespace | http://www.w3.org/1999/xlink |
| //mets/@xmlns:xsi | Schema-Instance | http://www.w3.org/2001/XMLSchema-instance |
| //mets/@xsi:schemaLocation | Schema-Location | http://www.loc.gov/METS/ |

---

[24] http://www.loc.gov/METS/

[25] http://purl.org/docs/index.html

[26] http://www.handle.net/

[27] http://www.do i.org

[28] Universally Unique Identifier according to RFC 4122, http://tools.ietf.org/html/rfc4122.html

[29] Deliverable D4.2, p. 17.

| | | |
|---|---|---|
| | | http://www.loc.gov/standards/mets/mets.xsd<br>http://www.w3.org/1999/xlink schemas/xlink.xsd |

An example of a root element with namespace and namespace location definitions is shown in Listing 1.

```
<mets
    xmlns:mets="http://www.loc.gov/METS/"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.loc.gov/METS/ http://www.loc.gov/standards/mets/mets.xsd
    http://www.w3.org/1999/xlink schemas/xlink.xsd">...</mets>
```

*Listing 1: METS root element example with namespace and namespace location definitions*

### 3.3.1.2 Digital objects

Digital objects are described in a file section (`<fileSec>`) of the METS document. Listing 2 shows an example of a file section with one file.

```
<fileSec>
    <fileGroup>
        <file ID="ID77146c6c-c8c3-4406-80b5-b3b41901f9d0"
            ADMID="..." MIMETYPE="text/x-sql" SIZE="2862064"
            CHECKSUMTYPE="SHA-256" CHECKSUM="..."
            CREATED="2015-0501T01:00:00+01:00">
            <FLocat LOCTYPE="URL"
                xlink:href="file://./submission/data/content/file.ext"
                xlink:type="simple"/>
        </file>
    <fileGroup>
</fileSec>
```

*Listing 2: Example of a file in the fileSec as child of a fileGroup element (long attribute values replaced by "..." for better readability)*

Table 2 lists the attributes of the `<file>` element with an example value. The `/file/FLocat` element provides the link to the actual file.

*Table 2: Attributes of the file element*

| Attribute | Description | Value |
|---|---|---|
| //file/@ID | File identifier; must be unique and start with the prefix "ID" | ID77146c6c-c8c3-4406-80b5-b3b41901f9d0 |
| //file/@ADMID | Used to link it to relevant administrative metadata sections that relate to the digital object | ID4566af74-0f7b-11e5-a6c0-1697f925ec7b<br>ID4566af74-0f7b-11e5-a6c0-1697f925ec7c |

| | | |
|---|---|---|
| | described. Can be a white space separated list of identifiers. | |
| //file/@CHECKSUMTYPE | Hash-sum calculator algorithm | SHA-256 |
| //file/@CHECKSUM | Hash-sum | 977fb584d53cd64662dfba427f3<br>5190813dfc58979f51a2703f8621<br>b9e1bc274 |
| //file/@CREATED | Date when the file entry was created. | 2014-05-01T01:00:00+01:00 |
| //file/@SIZE | Size of the file | 2498 |
| //file/@MIMETYPE | Mime-type | application/pdf |

Table 2 lists the attributes of the file element with an example value. The `/file/FLocat` element provides the link to the actual file.

The following rules apply for the URL attribute of the `<FLocat>` element:

> ***Requirement 19****: The local file paths CAN indicate the protocol, the file protocol (“`file://`”) is assumed if it is omitted.*

Additionally, the following requirement applies for compressed files:

> ***Requirement 20****: If a file is compressed, the transformFile element (`//file/transformFile`) SHOULD indicate how the packages have to be processed by means of the attributes ’TRANSFORMTYPE’, ’TRANSFORMALGORITHM’, and ’TRANSFORMORDER’.*

```
<file ...>
    <FLocat xlink:href="file://../compressed.tar.gz" xlink:type="simple" LOCTYPE="URL"/>
    <transformFile TRANSFORMORDER="1"
        TRANSFORMTYPE="decompression" TRANSFORMALGORITHM="gzip"/>
    <transformFile TRANSFORMORDER="2"
        TRANSFORMTYPE="decompression" TRANSFORMALGORITHM="tar"/>
</file>
```

*Listing 3: Compressed file*

#### 3.3.1.3 Referenced Metadata

Generally, the use of embedded metadata by using the `<mdWrap>` element is not recommended by E-ARK.

> ***Requirement 21***: *Metadata files such as EAD[30] or PREMIS files MUST be referenced by means of the `<mdRef>` element. Its "`xlink:href`" attribute value must be either a URL relative to the location of the METS root or an absolute URL.*

##### *3.3.1.3.1 Descriptive metadata*

The descriptive metadata section (`<dmdSec>`) references descriptive metadata contained in the AIP. Multiple `<dmdSec>` elements are allowed so that descriptive metadata can be referenced for each separate item within the METS object.

Descriptive metadata is referenced by means of the `<mdRef>` element as part of the descriptive metadata element `<dmdSec>`. Listing 4 shows an example linking to an EAD XML metadata file.

```
<dmdSec ID="ID550e8400-e29b-41d4-a716-44665544000a">
       <mdRef LOCTYPE="URL" MDTYPE="EAD" MIMETYPE="text/xml"
           CREATED="2015-04-26T12:00:00+01:00" xlink:type="simple"
           xlink:href="file://./metadata/EAD.xml"
           CHECKSUMTYPE="SHA-256" CHECKSUM="..." SIZE="2321"/>
</dmdSec>
```

*Listing 4: Linking to an EAD XML descriptive metadata file*

##### *3.3.1.3.2 Administrative Metadata*

> ***Requirement 22***: *The AIP METS must have a single `<amdSec>` element which contains one or several `<digiprovMD>` elements. The `<mdRef>` child of at least one of these elements must be of type "PREMIS" (`@MDTYPE="PREMIS"`) with the reference to a PREMIS file in the "Metadata" directory of the AIP root.*

Listing 5 shows an example with a link to a *PREMIS.xml* file:

---

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

```
<amdSec ID="...">
    <digiprovMD ID="...">
        <mdRef CHECKSUM="..." CHECKSUMTYPE="SHA-256"
            CREATED="..." LOCTYPE="URL" MDTYPE="PREMIS"
            MIMETYPE="text/xml" SIZE="1109"
            LABEL="Active PREMIS file"
            xlink:href="metadata/preservation/premis.xml" xlink:type="simple"/>
    </digiprovMD>
</amdSec>
```

*Listing 5: Linking to an EAD XML descriptive metadata file*

***Requirement 23**: The @LABEL attribute value of the `<mdRef>` element SHOULD start with “Obsolete” if the PREMIS file is obsolete and only included in the AIP to ensure traceability.*

***Requirement 24**: The @LABEL attribute value of the `<mdRef>` element CAN start with “Active” to make explicit that the PREMIS file is active.*

#### 3.3.1.4 Structural map

##### *3.3.1.4.1 Structural map of a divided METS structure*

***Requirement 25**: A structural map of type “physical” with the “LABEL” attribute value “E-ARK structural map” MUST exist for each METS file.*

***Requirement 26**: When an AIP uses the divided METS structure, i.e. the different representations have their own METS.xml file, the mandatory `<structMap>` MUST organize those METS.xml files through `<mptr>` and `<fptr>` entries, for each representation. The `<mptr>` node MUST reference the `/<representation>/METS.xml` and point at the corresponding `<file>` entry in the `<fileSec>` using the `<fptr>` element.*

```
<structMap TYPE="physical" LABEL="E-ARK structural map">
   <div LABEL="d7ef386d-275b-4a5d-9abf-48de9c390339">
     <div LABEL="representations/images_mig-1">
       <mptr xlink:href="file://./representations/images_mig-1/METS.xml" xlink:title="Mets
file describing representation: images_mig-1 of AIP: urn:uuid:d7ef386d-275b-4a5d-9abf-
48de9c390339." LOCTYPE="URL" ID="IDc063ebaf-e594-4996-9e2d-37bf91009155"/>
       <fptr FILEID="IDfb9c37e7-1c90-4849-a052-1875e67853d5"/>
     </div>
     <div LABEL="representations/docs_mig-1">
       <mptr xlink:href="file://./representations/docs_mig-1/METS.xml" xlink:title="Mets file
describing representation: docs_mig-1 of AIP: urn:uuid:d7ef386d-275b-4a5d-9abf-48de9c390339."
LOCTYPE="URL" ID="ID335f9e55-17b2-4cff-b62f-03fd6df4adbf"/>
       <fptr FILEID="ID3f2268cd-7da9-4ad8-909b-4f17730dacaf"/>
     </div>
   </div>
</structMap>
```

*Listing 6: Structural map referencing METS.xml files of the different representations*

#### 3.3.1.5 Metadata representation of the AIP structure

> ***Requirement 27**: One `<structMap>` with the LABEL attribute value "`E-ARK structural map`" MUST be present in the METS.xml file.*

Listing 7 shows a structural map with the `LABEL` attribute value "`E-ARK structural map`".

```
<structMap ID="IDf413c073-5b03-4499-830e-8ef724613bef" TYPE="physical" LABEL=" E-ARK structur-
al map">
    <div>
        <div LABEL="submission">
              ...
       </div>
       <div LABEL="representations">
           <div LABEL="rep-001">
              ...
           </div>
       </div>
   </div>
</structMap>
```

*Listing 7: Obligatory E-ARK structural map*

#### 3.3.1.6 Child AIP references parent AIP

The optional reference to a parent AIP is expressed by a structural map with the LABEL attribute value "`Parent`". Listing 8 shows an example where a UUID is used as the package identifier and the `xlink:href` attribute has the UUID identifier value of the referenced parent AIP as value. This identifier implicitly references the *METS.xml* file of the corresponding package. If other locator types, such as URN, URL, PURL, HANDLE, or DOI are used, the LOCTYPE attribute can be set correspondingly.

```
<structMap TYPE="logical" LABEL="parent AIP">
    <div LABEL="AIP parent identifier">
      <mptr xlink:href="urn:uuid:3a487ce5-63cf-4000-9522-7288e208e2bc"
            xlink:title="Referencing the parent AIP of this AIP
                         (URN:UUID:3218729b-c93c-4daa-ad3c-acb92ab59cee)."
            LOCTYPE="OTHER" OTHERLOCTYPE="UUID"
            ID="ID755d4d5f-5c5d-4751-9652-fcf839c7c6f2"/>
    </div>
</structMap>
```

*Listing 8: Using a structMap to reference the parent AIP*

#### 3.3.1.7 Parent AIP references child AIPs

The parent AIP which is referenced by child AIPs must have a structural map listing all child AIPs.

Listing 9 shows the structural map of a parent AIP listing four child AIPs.

```xml
<structMap TYPE="logical" LABEL="child AIPs">
    <div LABEL="child AIPs">
        <div LABEL="child AIP">
            <mptr xlink:href="urn:uuid:cea73348-741d-4594-ab8f-0b9e652c1099"
                  xlink:title="Referencing a child AIP."
                  LOCTYPE="OTHER" OTHERLOCTYPE="UUID"
                  ID="IDd98e416f-55a7-4237-8d45-59c22d221669"/>
        </div>
        <div LABEL="child AIP">
            <mptr xlink:href="urn:uuid:cea73348-741d-4594-ab8f-0b9e652c1099"
                  xlink:title="Referencing a child AIP."
                  LOCTYPE="OTHER" OTHERLOCTYPE="UUID"
                  ID="ID70f8ec28-23f1-4364-9163-b3e99165b6e6"/>
        </div>
        <div LABEL="child AIP">
            <mptr xlink:href="urn:uuid:3218729b-c93c-4daa-ad3c-acb92ab59cee"
                  xlink:title="Referencing a child AIP."
                  LOCTYPE="OTHER" OTHERLOCTYPE="UUID"
                  ID="ID77373d7f-e241-481b-bf89-675335beb049"/>
        </div>
        <div LABEL="child AIP">
            <mptr xlink:href="urn:uuid:cea73348-741d-4594-ab8f-0b9e652c1099"
                  xlink:title="Referencing a child AIP."
                  LOCTYPE="OTHER" OTHERLOCTYPE="UUID"
                  ID="ID3f0cc05c-f27d-499d-a6fd-63bdfed13cb0"/>
        </div>
    </div>
</structMap>
```

*Listing 9: Using a structMap to reference the parent AIP*

## 3.3.2 Preservation metadata

As already mentioned, PREMIS (version 2.2[31]) is used to describe technical metadata of digital objects, rights metadata to define the rights status in relation to specific agents or for specific objects, and to record events that are relevant regarding the digital provenance of digital objects.

In the following, the PREMIS format and the way that it relates to the METS elements is described in detail. NOTE: in the listings showing PREMIS code parts, the prefix "premis" is omitted (default namespace is the PREMIS namespace[32]) while the "mets" prefix is explicitly added if a relation to the METS file is explained.

### 3.3.2.1 Vocabulary

The definition of a vocabulary for PREMIS is an ongoing process, therefore E-ARK does not define an exhaustive list of vocabulary that is to be used exclusively.

---

[31] http://www.loc.gov/standards/premis/v2/premis-2-2.pdf

[32] Namespace: info:lc/xmlns/premis-v2, namespace schema location: http://www.loc.gov/standards/premis/v2/premis-v2-2.xsd

The basis of the preservation vocabulary used in E-ARK is the preservation schemes provided by the Library of Congress.[33] Additionally, recent contributions by the PREMIS Implementers Group are taken into consideration.[34] And, finally, there are contributions made by E-ARK project partners based on experience and best practices.

The vocabulary listed in the following sections is therefore to be seen as a core vocabulary which is continuously being extended.

##### *3.3.2.1.1 Identifier type*

Values of `*IdentifierType` elements.

- local = Scope is the PREMIS file.
- internal = Scope is the AIP.
- repository = Scope is the Repository (e.g. UUID known by the repository).
- uri = Form of identifier that is a unique resource identifier.
- filepath = AIP scope file path.

##### *3.3.2.1.2 Event type*

Values of `eventType` elements.

- adding emulation information - Adding emulation information.
- AIP validation - Validation of the AIP.
- archive - Archiving the AIP.
- capture - Obtaining a digital object.
- compression - compression of files.
- creation - Creating a new digital object.
- decompression - decompression of files.
- deaccession - The process of removing an object from the inventory of a repository.
- digital signature validation - The process of determining that a decrypted digital signature matches an expected value.
- decryption - Decryption of data.
- deletion - Removing a digital object from the repository.
- digital signature validation - Validating a digital signature.
- fixity check - Checking file hashsums.
- format identification - File format identification (e.g. PRONOM PUID).
- format validation - File format validation (e.g. using JHove).
- identifier assignment - Assignment of an identifier.
- ingestion - Adding digital objects to the repository.
- message digest calculation - Calculate the message digest of digital objects.

---

[33] http://id.loc.gov/vocabulary/preservation.html

[34] http://premisimplementers.pbworks.com/w/page/102413902/Preservation%20Events%20Controlled%20Vocabulary

- migration - File format migration.
- replication - Creating an exact copy of a digital object.
- SIP creation - Creation of the SIP.
- SIP validation - Validation of the SIP.
- storage migration - Moving digital object to another storage medium.
- validation - Structure and compliance validation of the AIP.
- virus check - Virus check

##### *3.3.2.1.3 Event outcome*

Values of `eventOutcome` elements.

- success - Process was applied successfully
- failure - An error occurred

##### *3.3.2.1.4 Agent Type*

Values of `agentType` elements.

- software - Software agent
- person - Person agent
- organisation - Organisation agent
- hardware - Hardware agent

##### *3.3.2.1.5 Rights granted - act*

Values of `rightsGranted/act` elements

- discover
- display
- copy
- migrate
- publish
- modify
- delete
- print

##### *3.3.2.1.6 Rights granted - restriction*

Values of `rightsGranted/restriction` elements

- GENERAL - Right statement must not be explicitly stated for each file object, action is allowed to be performed on each digital object. In this case it is sufficient to know that an agent has a specific right.
- PER_FILE - Right statement must be explicitly stated for each digital object. It is not sufficient to know if an agent has a specific right, it must be verified for each individual object if the specific right is given.

#### *3.3.2.1.7 Relationship*

Values of `relationshipSubType` elements. Logical relations to other AIPs.

- part of
- has part
- is sibling
- is version of

### 3.3.2.2 PREMIS object

The PREMIS object contains technical information about a digital object.

#### *3.3.2.2.1 Object identifier*

> ***Requirement 28****: An object identifier of type "filepath" MUST be used.*
>
> ***Requirement 29****: If an identifier of type "local" is used, this identifier SHOULD be used in the scope of the PREMIS document to specify links to the PREMIS object.*
>
> ***Requirement 30****: Other object identifiers of the allowed types CAN be used additionally to the identifier of type "filepath".*

The example shown in Listing 10 has one identifier which is valid in the scope of the PREMIS file of type "local" and one identifier of type "filepath" which is valid in the scope of the AIP. If only the identifier of type "filepath" is used, the file path must be used to link to the PREMIS object.

```
<objectIdentifier>
       <objectIdentifierType>local</objectIdentifierType>
       <objectIdentifierValue>fileId001</objectIdentifierValue>
</objectIdentifier>
<objectIdentifier>
       <objectIdentifierType>filepath</objectIdentifierType>
       <objectIdentifierValue>metadata/file.xml</objectIdentifierValue>
</objectIdentifier>
```

*Listing 10: Object identifier*

#### *3.3.2.2.2 Fixity*

Fixity information is provided as a descendant of the `objectCharacteristics` element information in form of a SHA-256 hashsum, a fixed size 256-bit value. An example is shown in Listing 11.

```
<fixity>
       <messageDigestAlgorithm>SHA-256</messageDigestAlgorithm>
       <messageDigest>3b1d00f7871d9102001c77f...</messageDigest>
       <messageDigestOriginator>/usr/bin/sha256sum</messageDigestOriginator>
</fixity>
```

*Listing 11: Hashsum (value shortened)*

#### *3.3.2.2.3 File format*

The Pronom Unique Identifier is provided as a as a descendant of the objectCharacteristics element in the form Persistent Unique Identifier (PUID)[35] based on the PRONOM technical registry.[36] An example is shown in Listing 12.

```
<format>
       <formatRegistry>
       <formatRegistryName>PRONOM</formatRegistryName>
       <formatRegistryKey>fmt/101</formatRegistryKey>
       <formatRegistryRole>identification</formatRegistryRole>
       </formatRegistry>
</format>
```

*Listing 12: Pronom Unique Identifier*

#### *3.3.2.2.4 Object characterisation*

The JHOVE[37] technical characterisation result (XML format) is embedded as a descendant of the objectCharacteristicsExtension element. An example is shown in Listing 13.

```
<objectCharacteristicsExtension>
       <mdSec ID="ID426087e8-0f79-11e3-847a-34e6d700c47b">
           <mdWrap MDTYPE="OTHER" OTHERMDTYPE="JHOVE">
              <xmlData>
              <jhove>
                     ...
              </jhove>
              </xmlData>
           </mdWrap>
       </mdSec>
</objectCharacteristicsExtension>
```

*Listing 13: JHove digital object characterisation*

---

[35] http://www.nationalarchives.gov.uk/aboutapps/pronom/puid.htm

[36] http://www.nationalarchives.gov.uk/PRONOM

[37] http://sourceforge.net/projects/jhove/

#### *3.3.2.2.5 Original name*

The original name is the absolute path of the file name in the original SIP file, an example is shown in Listing 14.

```
<originalName>D:///data/file.ext</originalName>
```

*Listing 14: Original name*

#### *3.3.2.2.6 Storage*

The `storage` element should contain information about the physical location of the digital object. Ideally this is a resolvable URI, but it can also generally hold information needed to retrieve the digital object from the storage system (e.g. access control). An example is shown in Listing 15.

```
<storage>
    <contentLocation>
        <contentLocationType>URI</contentLocationType>
        <contentLocationValue>
          metadata/file.xml
        </contentLocationValue>
    </contentLocation>
    <storageMedium>unix file system</storageMedium>
</storage>
```

*Listing 15: Storage description*

#### *3.3.2.2.7 Relationship*

This element contains the "part-of" relationship of the digital object. For digital objects included in the AIP, the value "is included in" of the `relationshipSubType` element means that a digital object is part of an AIP which is just making an explicit statement about a fact. More importantly, in the case of "segmented AIPs", it is used to express that an AIP is part of a parent AIP which is not expressed otherwise. An example of the latter case is shown in Listing 16.

```
<relationship>
    <relationshipType>structural</relationshipType>
    <relationshipSubType>is included in</relationshipSubType>
       <relatedObjectIdentification>
           <relatedObjectIdentifierType>repository</relatedObjectIdentifierType>
           <relatedObjectIdentifierValue>
               ID123e4567-e89b-12d3-a456-426655440000
        </relatedObjectIdentifierValue>
    </relatedObjectIdentification>
</relationship>
```

*Listing 16: Relationship*

#### 3.3.2.2.8 Linking rights statement

If a `linkingRightsStatementIdentifier` child element object exists, there is a rights statement attached to the object. For example, only files which have the "discovery right" are being indexed in order to allow these files to be retrievable by the full-text search. An example of the latter case is shown in Listing 17.

```
<linkingRightsStatementIdentifier>
    <linkingRightsStatementIdentifierType>
        filepath
    </linkingRightsStatementIdentifierType>
    <linkingRightsStatementIdentifierValue>
        metadata/file.xml
    </linkingRightsStatementIdentifierValue>
</linkingRightsStatementIdentifier>
```

*Listing 17: Rights statement*

### 3.3.2.3 PREMIS event

#### 3.3.2.3.1 Event identifier

The event identifier is an identifier that is valid in the scope of the PREMIS file. An example is shown in Listing 18.

```
<eventIdentifier>
       <eventIdentifierType>local</eventIdentifierType>
       <eventIdentifierValue>ingest</eventIdentifierValue>
</eventIdentifier>
```

*Listing 18: Event identifier*

#### 3.3.2.3.2 Event date/time

Combined date and time in UTC format (ISO 8601), example shown in Listing 19.

```
<eventDateTime>2014-05-01T01:00:00+01:00</eventDateTime>
```

*Listing 19: Event date/time*

#### 3.3.2.3.3 Link to agent/object

The event is linked to an agent and an object. In the example shown in listing 20 the SIP to AIP conversion software is linked as agent with identifier value ’Sip2Aip’ and the corresponding object is linked by the local UUID value. An example is shown in

Listing 20.

```
<linkingAgentIdentifier>
    <linkingAgentIdentifierType>local</linkingAgentIdentifierType>
        <linkingAgentIdentifierValue>
            earkweb-sip2aip
        </linkingAgentIdentifierValue>
    </linkingAgentIdentifier>
    <linkingObjectIdentifier>
    <linkingObjectIdentifierType>filepath</linkingObjectIdentifierType>
        <linkingObjectIdentifierValue>
            metadata/file.xml
        </linkingObjectIdentifierValue>
</linkingObjectIdentifier>
```

*Listing 20: Link to agent/object*

#### *3.3.2.3.4 Migration event type*

The migration event (value of element `eventIdentifierType` is "migration") needs to be related to the event that created the source object by means of the `relatedEventIdentification`. An example is shown in Listing 21.

```
<event>
    <eventIdentifier>
        <eventIdentifierType>local</eventIdentifierType>
        <eventIdentifierValue>migration-001</eventIdentifierValue>
    </eventIdentifier>
    <eventType>MIGRATION</eventType>
    <eventDateTime>2015-09-01T01:00:00+01:00</eventDateTime>
    <eventOutcomeInformation>
        <eventOutcome>success</eventOutcome>
    </eventOutcomeInformation>
    <linkingAgentIdentifier>
        <linkingAgentIdentifierType>local</linkingAgentIdentifierType>
        <linkingAgentIdentifierValue>
            FileFormatConversion001
        </linkingAgentIdentifierValue>
    </linkingAgentIdentifier>
    <linkingObjectIdentifier>
        <linkingObjectIdentifierType>filepath</linkingObjectIdentifierType>
        <linkingObjectIdentifierValue>
            metadata/file.xml
        </linkingObjectIdentifierValue>
    </linkingObjectIdentifier>
    <relatedEventIdentification>
        <relatedEventIdentifierType>local</relatedEventIdentifierType>
        <relatedEventIdentifierValue>
            ingest-001
        </relatedEventIdentifierValue>
    </relatedEventIdentification>
</event>
```

*Listing 21: Migration event*

The event shown in Listing 21 expresses the fact that the object "`metadata/file.xml`" is the result of the migration event "migration-001"and the event which created the source object is "ingest-001".

### 3.3.2.4 PREMIS rights

The rights element holds information about the rights status of individual digital objects or about agents. *As an example for a rights statement,*

Listing 22 describes a discovery right. This statement is used in the following Listing 23 to define the right of agent, in this case the right to perform indexing.

```
<rights>
       <rightsStatement>
           <rightsStatementIdentifier>
              <rightsStatementIdentifierType>
              local
              </rightsStatementIdentifierType>
              <rightsStatementIdentifierValue>
              discovery-right-001
              </rightsStatementIdentifierValue>
           </rightsStatementIdentifier>
           <rightsBasis>Statute</rightsBasis>
           <rightsGranted>
              <act>Discovery</act>
              <restriction></restriction>
           </rightsGranted>
       </rightsStatement>
</rights>
```

*Listing 22: Premis rights statement*

#### 3.3.2.5 PREMIS agent

The agent element holds information about agents (people, organizations or software).

As an example for an agent, listing shows the software Lily[38] which in the E-ARK project is used to index the text content of archival information packages. There is the "discovery right" assigned to this agent. An example is shown in Listing 23.

---

[38] http://www.ngdata.com/press-piece/lily-1-0-smart-data-at-scale-made-easy/

```
<agent>
    <agentIdentifier>
        <agentIdentifierType>local</agentIdentifierType>
        <agentIdentifierValue>Lily</agentIdentifierValue>
    </agentIdentifier>
    <agentName>E-ARK Lily</agentName>
    <agentType>Software</agentType>
    <linkingRightsStatementIdentifier>
        <linkingRightsStatementIdentifierType>
            local
        </linkingRightsStatementIdentifierType>
        <linkingRightsStatementIdentifierValue>
            discovery-right-001
        </linkingRightsStatementIdentifierValue>
    </linkingRightsStatementIdentifier>
</agent>
```

*Listing 23: Software as an agent*

## 3.4 E-ARK Packaging format

Part of the E-ARK AIP format is the specification which shows how the AIP is packaged into a transferable and storable entity.

As a convention, in the context of the E-ARK Integrated Prototype implementation, there is the assumption of a one-to-one relationship between an SIP and the corresponding AIP as described in section 2.6. As a consequence, merging or splitting of AIPs is not supposed to happen during SIP to AIP conversion. For archived AIPs it means that the operation of changing the partition of a logical AIP is done by retrieving the corresponding packages and creating the desired division of SIPs which is then one-by-one converted into the result AIP(s).

In the E-ARK project, SIPs must be of a manageable size. There is no fixed size which defines what "manageable" means, because this depends on the institutional environment and especially on the storage media which is used for long-term preservation. But to get an idea derived from current practice in various archival institutions, it can roughly mean that a delivery of Terabyte dimension must be divided into Gigabyte-sized SIPs.

### 3.4.1 Package manifest

The E-ARK AIP contains a manifest file (manifest.txt) with a complete list of files with MD5 and SHA-256 hashsum.

The manifest file is a text file containing a list of records separated by two line breaks (two carriage return characters (hexadecimal 0D0D) or two times carriage return/line feed (hexadecimal 0D0A0D0A). A record is a list of named fields, the minimum fields are:

- Name := File path relative to the AIP root
- Size := Size in bytes

- SHA256 := SHA-256 Checksum
- MD5 := MD5 Checksum

An example is shown in Listing 24.

```
Name: METS.xml
Size: 12135
SHA256: d7dec534d2ba5f455391e2ed0cb89db89a2780e0531c83def79b0b0abcb38679
MD5: e94dd23e792bd7e49721a863ad8ed769
Name: metadata/PREMIS.xml
Size: 53719
SHA256: ef01bc59a21f6e99ad3d87b0d25b89d6e8b4915c63dadb8791d9490739fe26d4
MD5: 96b85205a9b4b0b5d3c88e2e51b0dc4c
```

*Listing 24: Manifest file*

#### 3.4.1.1 Naming of the physical container of the AIP

As stated in Deliverable D4.2, "when an AIP is created during ingest, it receives an unalterable identifier, which defines the AIP as one consistent logical entity.

Using the following variables of which the identifier will be composed:

- identifier := Identifier of the Intellectual entity assigned by the repository
- version := Version is an increment-able zero-filled number where the highest number always represents the latest "version".
- package_format := Package file format

and the following file naming scheme for the AIP file:

`<identifier>_<version>.<package_format >`

Assuming that we have an intellectual entity with the identifier "426087e8-0f79-11e3-847a-34e6d700c47b". The first submission (version=00001) using package format "tar", in summary, the following variable values:

- id := "426087e8-0f79-11e3-847a-34e6d700c47b"
- version:= "00001"
- package_format := tar

would result in the following package name:

`426087e8-0f79-11e3-847a-34e6d700c47b_00001.tar`

If the AIP is changed, because, for example, a new representation was added to the AIP, a new physical container will be created as a new version of the AIP. As a result, only the numerical suffix of the generation part will be incremented (_00002) and the identifier for the intellectual entity remains the same (`426087e8-0f79-11e3-847a-34e6d700c47b`):

`426087e8-0f79-11e3-847a-34e6d700c47b_00002.tar`

# 4 E-ARK digital object classes

The E-ARK AIP format relates to all kinds of digital object types, but here we draw special attention to some digital object classes that were especially considered in the context of the E-ARK project.

## 4.1 OLAP (Online Analytical Data Processing)

Current database archiving relies mostly on archiving the data according to the original data model in an open format like SIARD (Software Independent Archiving of Relational Databases)[39]. Databases archived in such a manner are often difficult to reuse because of their highly complex data structures being hard to understand for users. To simplify access to archived databases, E-ARK is looking into methods to prepare de-normalized AIPs ready for dimensional analysis via OLAP. The topics: normalization and denormalization, OLAP, dimensional analysis and the related data warehousing are now summarized for those who are new to the area.

Relational databases are built upon normalization practice as set out by Codd[40], which process involves the steps of identifying the main entities (things of importance) to be modelled, then ensuring that the attributes (characteristics) of each entity are housed in that entity's own database table, so that each entity can be altered if need be, *independently* of all the other entities. For decades, normalization has been a major influence on database design, because it optimizes database performance, especially for OnLine Transaction Processing (OLTP) systems such as managing day-to-day banking transactions (Connolly and Begg[41]). However, creating a separate table for each entity can soon result in a large and unwieldy data model (ERD – Entity Relational Diagram), leading to many joins (links) across the tables in order to gather together all the related data to satisfy each and every particular query. Also, referential integrity rules are necessary in relational data modelling to ensure that the joins across tables are robust, correct and permanent. All this information: entities, attributes, the relations between them, referential integrity rules showing how the tables are linked are held together in a data dictionary. The example in Figure 13[42] illustrates the kind of complex topography to be found in an ERD (the general format / layout of this ERD is what is of interest here, not the individual entities or their attributes).

---

[39] SIARD http://www.ech.ch/vechweb/page?p=dossier&documentNumber=eCH-0165 (N.B. This is version 1.0, and version 2.0 is under review in cooperation with E-ARK http://www.eark-project.com/resources/specificationdocs/32-specification-for-siard-format-v20 )

[40] Codd, E.F. 1972, "Further normalization of the data base relational model". In *Data Base Systems* (Rustin R., ed.), Englewood Cliffs, NJ: Prentice Hall

[41] Connolly, T. and Begg, C. 2014, *Database Systems: A Practical Approach to Design, Implementation, and Management*, 6th edition, Harlow: Pearson.

[42] https://en.wikipedia.org/wiki/Data_model#/media/File:B_5_1_IDEF1X_Diagram.jpg

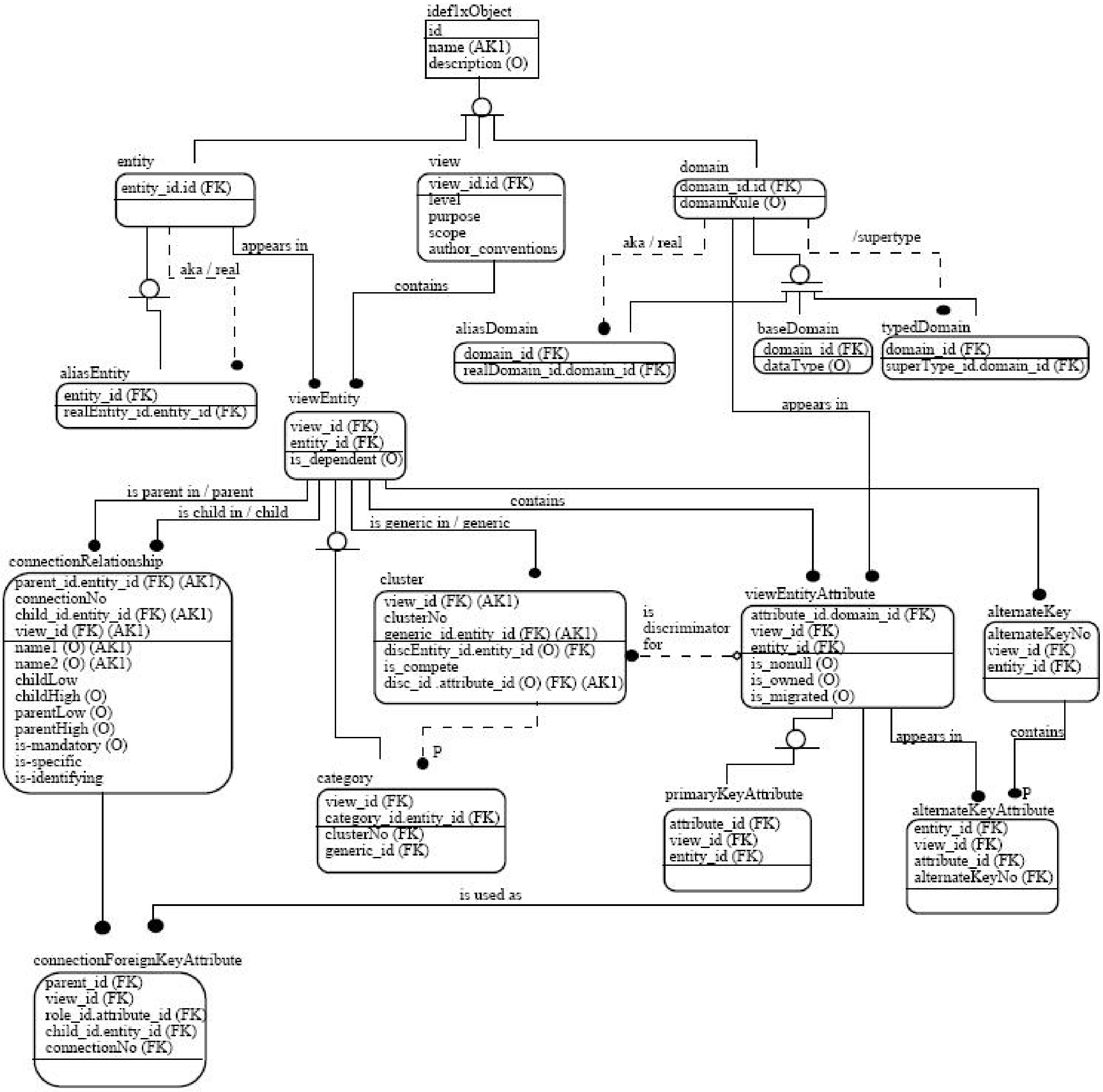


*Figure 13: Entity Relationship Diagram (ERD)*

Over the last two to three decades, there has been an emphasis on creating analytical databases that are designed to facilitate complex querying in a much easier fashion than creating joins across many tables[43]. Thus dimensional modelling has grown up, which is based on the star schema instead of the traditional ERD. From the ERD of a database, a major entity of interest is identified as the centre or *fact table* of the star

---

[43] Inmon, W.H., 2005. *Building the Data Warehouse*. 4th edition. Indianapolis: Wiley.

schema, and the other tables can then be denormalized and loosely combined / fashioned as *dimension tables* which surround the fact table. The dimension tables contain highly descriptive data and enable rich and powerful analysis. The dimension tables could be normalized if desired (snowflake schema), or both normalized and de-normalized (starflake schema). Dimensional models are much easier to navigate, much more user-friendly and intuitive than the traditional ERDs, as can be seen by comparing their general structures in Figure 13 and Figure 14. Figure 14[44] presents the layout of the basic star schema dimensional model.

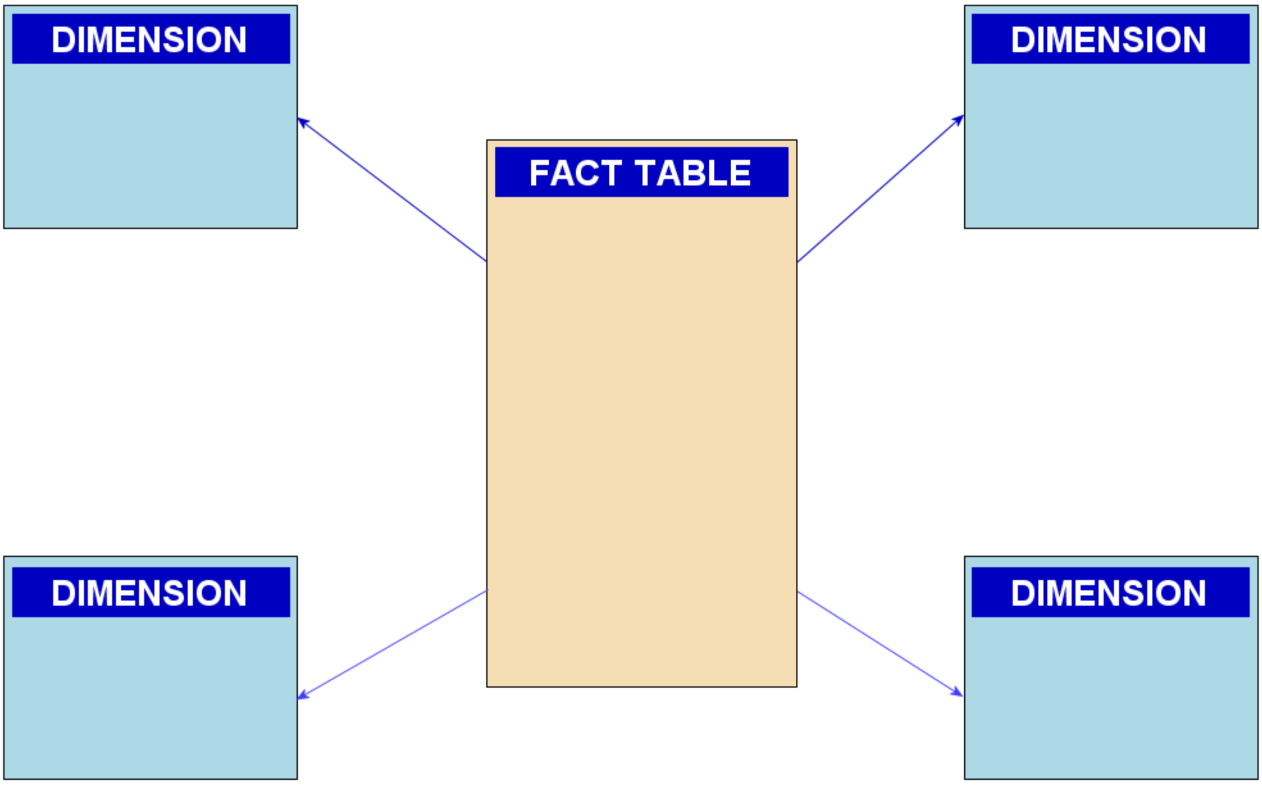


*Figure 14: Simple Fact Table*

Massive data warehouses can be built based on dimensional modelling, their primary purpose being large-scale data storage and analysis[45]. Data warehouses typically take in snapshots of databases taken over time, integrated with other relevant data from diverse sources. These huge data workhorses have been used over several decades for business analysis and forecasting by many companies, exemplars being Google[46] and

---

[44] https://en.wikipedia.org/wiki/Data_model#/media/File:Star-schema.png

[45] Inmon, W.H., 2005. *Building the Data Warehouse*. 4th edition. Indianapolis: Wiley.

[46] https://gigaom.com/2014/08/07/google-shows-off-mesa-a-super-fast-data-warehouse-that-runs-across-data-centers/

Amazon, who now offer a general data warehousing service called redshift[47].

OLAP is concerned with analysis and involves aggregating the dimensional data and visualizing them as data cubes (also called hyper cubes or data marts[48]). Again, these data cubes enhance understandability of the data, and they facilitate complex analysis as it is possible to: travel up hierarchies in a dimension (roll / drill up), drill down dimensional hierarchies to locate specific data items, and slice and dice these cubes to highlight data. Figure 15[49] depicts a simple OLAP cube. Figure 16[50] delineates slicing an OLAP cube, Figure 17[51] dicing an OLAP cube, and Figure 18[52] drill (roll) up and drill down.

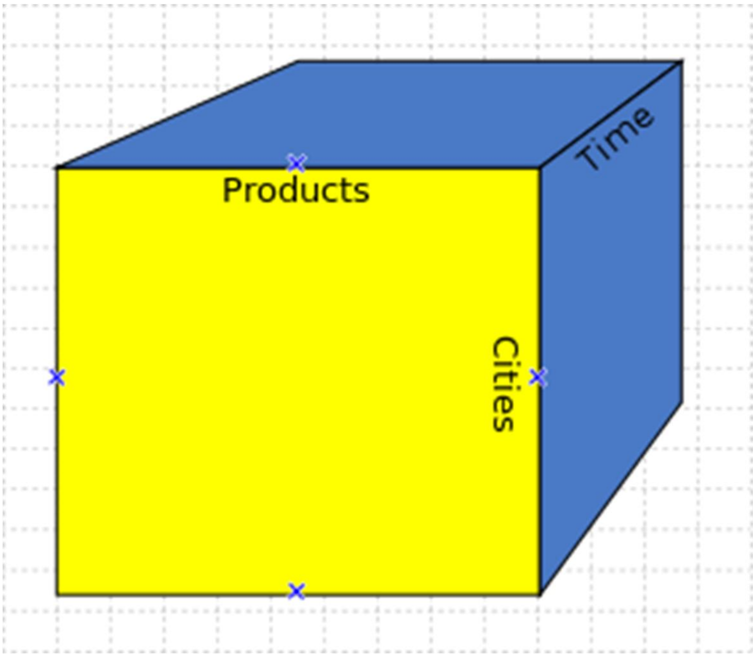


*Figure 15: Simple OLAP cube*

---

[47] https://aws.amazon.com/redshift

[48] Kimball, R., Ross, M., 2013, The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modelling, 3rd edition, Indianapolis: Wiley.

[49] https://en.wikipedia.org/wiki/OLAP_cube#/media/File:OLAP_Cube.svg

[50] https://en.wikipedia.org/wiki/OLAP_cube#/media/File:OLAP_slicing.png

[51] https://en.wikipedia.org/wiki/OLAP_cube#/media/File:OLAP_dicing.png

[52] https://en.wikipedia.org/wiki/OLAP_cube#/media/File:OLAP_drill_up%26down.png

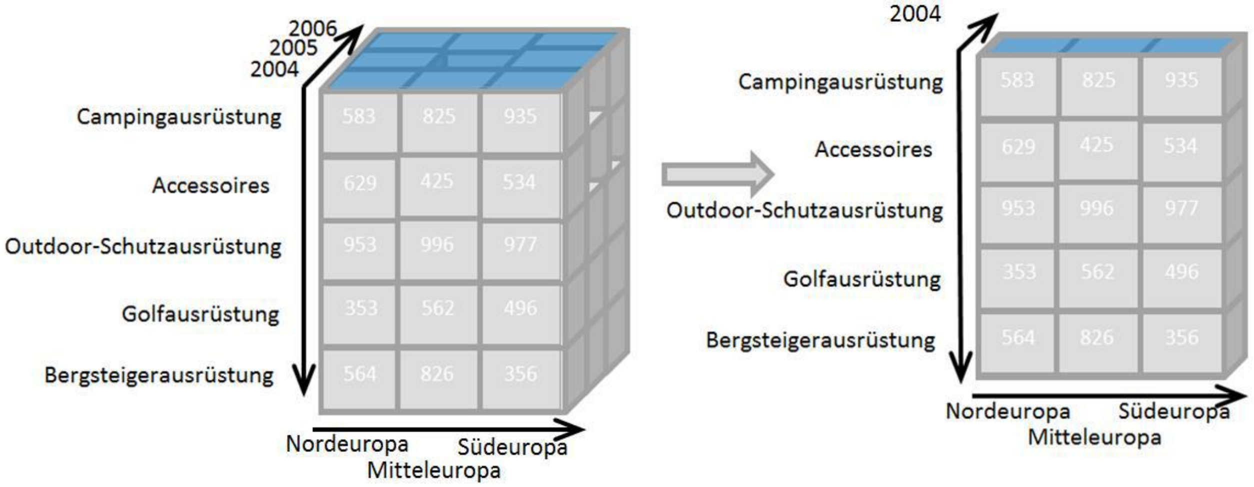


*Figure 16: Slicing an OLAP cube*

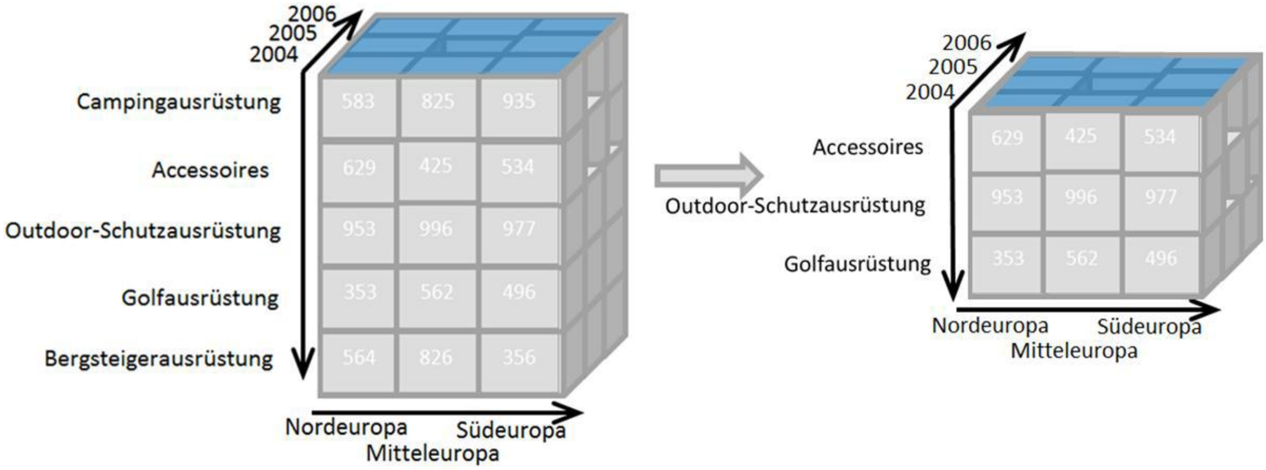


*Figure 17: Dicing an OLAP cube*

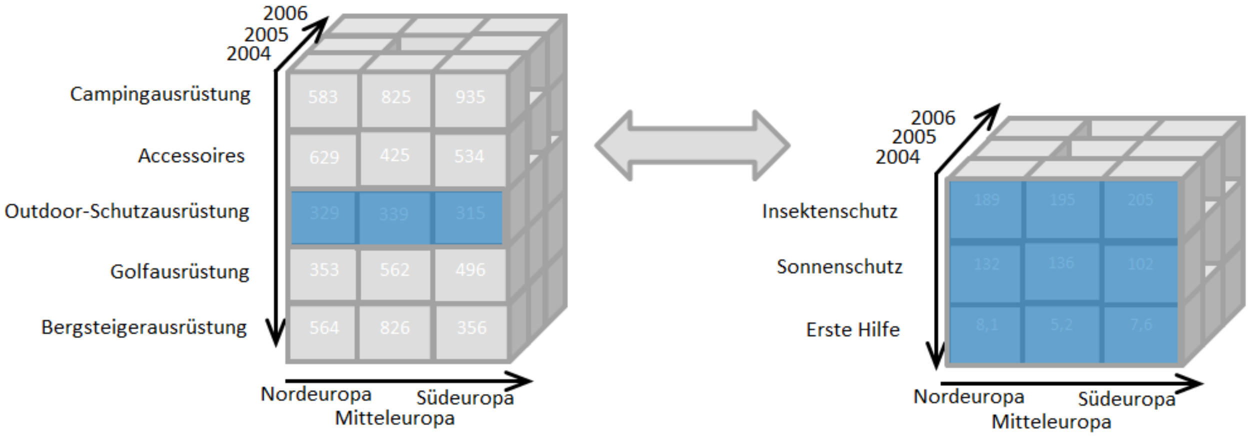


*Figure 18: Drill (Roll) up / Drill down an OLAP cube*

There are also different varieties of OLAP: Relational OLAP (ROLAP) based on normalized relational tables which are used to produce OLAP cubes on the fly; Multidimensional OLAP (MOLAP) which has fixed cube structures, and HOLAP which is a hybrid version of both ROLAP / MOLAP. In the following discussions we will be using ROLAP.

In the following we present advanced database archiving that prepares specifically de-normalized AIPs ready for dimensional analysis via OLAP and other tools. Essentially this is adding value to the AIPs by making them more discovery-ready, allowing complex analysis to be carried out on data within an archive, or even across several archives from different countries (compatibility being ensured by harmonization provided by the E-ARK specifications).

A number of different scenarios can be envisaged at the start of the SIP to AIP conversion process for standard relational database tables:

- Where the database tables have been de-normalized prior to conversion into SIARD format, in which case they can be designated 'OLAP cube ready' within the SIP (by means of a suitable metadata flag)
- Where no de-normalization has taken place prior to SIARD conversion, but the referential integrity constraint information has been provided in standardized digital format, so the de-normalization can take place in an automated fashion, in the course of SIP to AIP conversion.
- Where no de-normalization has taken place prior to SIARD conversion and no machine-readable referential integrity information has been provided. In this case, manual intervention is required to convert the referential integrity constraint information into digital form, prior to input to the automated de-normalization process.
- Where tables are provided in SIARD format but there is no referential integrity constraint information available at the time SIP to AIP conversion takes place. However, subsequently, this information becomes available (either through discovery of old documentation or by means of analogy with later, better documented versions of the same or very similar groupings of tables). In this case, following conversion of the constraint information into digital form, a new de-normalized representation of the older non-normalized data can be generated and added to the AIP.

It should be noted that the intention is to develop a vendor-independent database table structure to hold the referential integrity constraint information in a standard form, since there is currently no accepted standard for the structure and naming of data dictionary tables/views that contain this information. When this standard is agreed, it should be possible to develop automated procedures for abstracting the required information from different vendor systems, if it has previously been provided at the time when the original tables were created. Since this vendor-independent table will have the form of a standard relational table, it will clearly be possible for this to be converted into SIARD format at the appropriate stage and then unpacked from that format at the same time as the other tables that will be used as part of the de-normalization process.

It is important to note that although data warehouses and OLAP cubes are "new" analytical tools / architectures, they are still based on database tables, and so come under the umbrella of needing database archiving themselves. Since in a ROLAP implementation of a data warehouse, all the fact/cube tables and the dimensions also have a standard table structure, in principle there is no reason why an extension of the same approach could not be used to allow the archiving of data warehouses by means of their component tables. Specifying the referential integrity constraints between measures (attributes) in a cube and the relevant primary keys (unique identifiers) in dimension tables is the same process as required for standard relational tables described above, so no additional functionality is required to achieve this, though it may be useful for the constraint table (which hold details of all the constraints affecting the database) to include a flag to indicate that the constraint in question links a cube to a dimension. Clearly, however, no further de-normalization is required in this case, since this has already been undertaken as part of the original dimensional modelling and cube creation process.

However, to provide a complete description of the dimensional model underlying the specific data warehouse structure, it will be necessary to develop a second vendor-independent table structure to describe the hierarchical structure of aggregation levels in each dimension. The reason is the same as before – there is no agreed standard for the structure or naming of data dictionary views in commercial systems that contain this information. Fortunately, since the setting up of dimensions implies a much higher degree of data modelling skill than simply creating tables, and the structure of a dimension is based on a number of standard rules that OLAP extensions to SQL (Structured Query Language, used to interrogate (particularly relational) databases) need to use in an automated fashion, it is to be expected that there will be at least the detailed metadata present in the source database's data dictionary, which can be output in the standard vendor-independent form. The corollary of this is that this same vendor-independent information can be used to generate new 'create dimension' statements that would enable the data warehouse structure to be re-established, if required, at least semi-automatically, in a commercial system different to that from which it originated. Extending the logic of this process, would potentially mean that data from multiple vendors' data warehouses could be merged in the course of archiving it all in a single data warehouse, with the addition of further de-normalized data that had not previously been held in data warehouse form. As before, the data warehouse specific metadata table can be converted into SIARD format and handled in the same way as the other tables containing the actual dimension and cube data.

A final consideration is that the output of an SQL OLAP query against a data warehouse necessarily also has the form of a table. Hence such output could also be archived in SIARD format, conceptualized either as stored queries (e.g. commonly requested results) or as data aggregations for specific purposes, such as census summary tables, or for subsequent conversion into specialist statistical formats, such as SDMX, for storage or onward transmission to other agencies.

Work is now in progress to examine the technical feasibility of the proposed approach to the handling of OLAP-related resources within E-ARK. While it is not possible to survey all the possible ways in which data dictionaries in different systems may hold referential integrity information, alternative technical solutions to the problem of developing a vendor-independent table structure for holding this information are now being evaluated. This work is being undertaken with an eye to industry standards such as SQL92, though commercial DBMS systems vary substantially in their compliance with the latter standard, for example. Once a draft solution has been identified and circulated for comment, a test implementation/proof-of-concept will be undertaken using ORACLE, as arguably the leading commercial system, which is also in use by multiple partners in the E-ARK project.

A similar approach will subsequently be adopted to develop the vendor-independent table structure for holding metadata on data warehouse dimensions. It is recognised that the level of standardisation between vendors in this area is much less than for referential integrity information, so there may need to be a greater reliance on an ORACLE-based approach, since it is believed that data warehouse metadata handling is more effectively developed in this system than in many others.

In both cases, it will be necessary to provide guidelines for users/analysts on how to populate the resulting vendor -independent table structures, to maximise the benefits that can be derived from automation of the relevant metadata handling.

## 4.2 MoReq2010®

Another objective of the E-ARK project is to standardize the way that content from Electronic Records Management Systems (ERMSs) can be archived. The E-ARK approach reuses the MoReq specification and gives guidance on how to reuse it for SIPs and AIPs.

The original MoReq specification was published in 2001 as an outcome of the cooperation between the DLM Forum and the European Commission. MoReq aimed at providing a specification for systems that manage electronic records. In 2008, the DLM Forum published MoReq2 which is the result of updating and extending the original MoReq specification. MoReq2 introduced a testing and certification regime. Suppliers could now have their solutions tested at a MoReq2 testing center and receive independent certification that their products were compliant with the specification. In order to support testing and certification, MoReq2 introduced a metadata model into the specification, as well as an XML schema that was intended to define a common import/export format across different products and implementations.

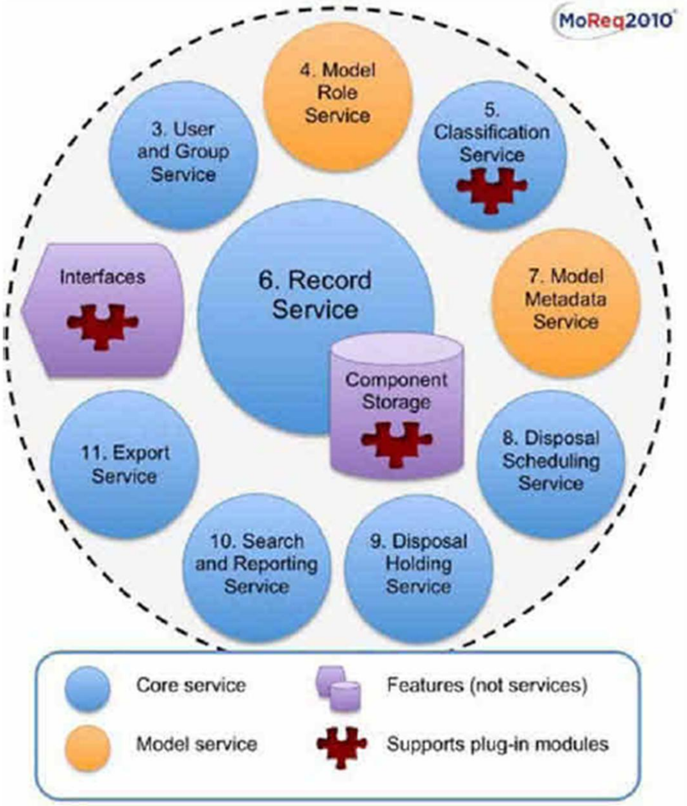

MoReq2010
4. Model Role Service
3. User and Group Service
5. Classification Service
Interfaces
6. Record Service
7. Model Metadata Service
Component Storage
11. Export Service
8. Disposal Scheduling Service
10. Search and Reporting Service
9. Disposal Holding Service
Core service
Features (not services)
Model service
Supports plug-in modules

*Figure 19: MoReq2010 System Services*

In 2011, and marking the 10th anniversary of the MoReq specification, DLM Forum® launched MoReq2010. In MoReq2010 the concept of "model" requirements has been replaced by that of "modular" requirements. The new modular approach allowed consumers to easily specify a flexible yet comprehensive and cohesive set of organizational requirements simply by choosing a suitable combination from a selection of modules that correspond to their organizational needs. The objective was to have the number and variety of modules that build on the platform of the MoReq2010 core services steadily increase over time, and extend coverage. The new specification defined that a MoReq2010 Compliant Records System (MCRS) should consist of the following set of services (Figure 19):

- **User and Group Service** responsible for managing users and groups of users of the system;
- **Model Role Service** for managing the roles that can be assigned to users and groups;
- **Classification Service** responsible for managing classes and aggregations - entities necessary to define a records management classification scheme;
- **Record Service** responsible for managing the records and their content (components);
- **Model Metadata Service** that allows users to define and assign metadata to the different entities of MoReq2010;
- **Disposal Scheduling Service** responsible for managing records lifecycle through the definition and management of disposal schedules;

- **Disposal Holding Service** capable of holding the disposal of records, a service imposed by the potential need to audit records of an organization;
- **Search and Reporting Service** that assures that every record can be found if the user searching has the necessary view rights; and
- **Export Service** to ensure the interoperability between records management systems and archives.

### 4.2.1 MoReq2010 Interoperability

One of the goals of MoReq2010 is to assure that the needs for interoperability between records management systems are met. Therefore, MoReq2010 introduced the MoReq2010 Export Service to assure that records in a MCRS can be exported to other systems. According to MoReq2010, a record can be exported due to: being relocated to a different system, organization or archive; a migration where records are moved from one MCRS to another; need to assure secondary hosting for the records, and for replication purposes. As such, in MoReq2010, the MCRS export operations resort to a MCRS description based in XML data file, (see Figure 20), which can be validated against MoReq2010's XML schema.

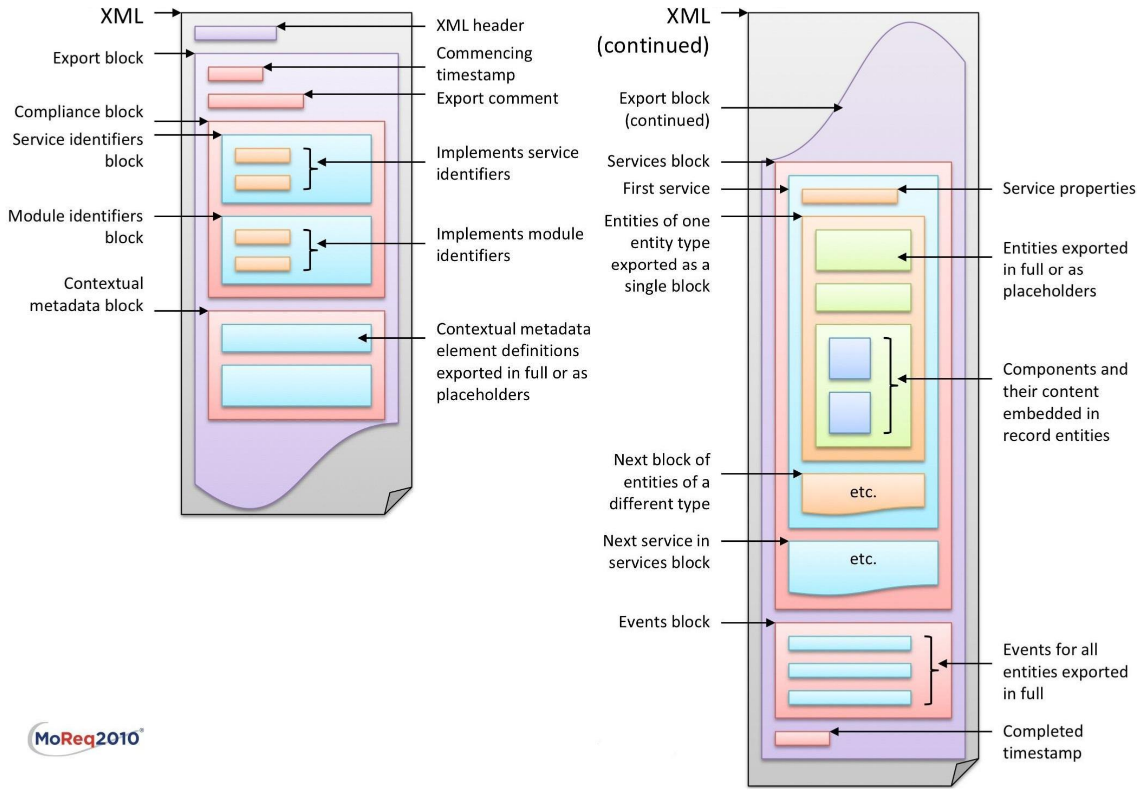

XML
Export block
Compliance block
Service identifiers block
Module identifiers block
Contextual metadata block
XML header
Commencing timestamp
Export comment
Implements service identifiers
Implements module identifiers
Contextual metadata element definitions exported in full or as placeholders
MoReq2010
XML (continued)
Export block (continued)
Services block
First service
Entities of one entity type exported as a single block
Next block of entities of a different type
Next service in services block
Events block
Service properties
Entities exported in full or as placeholders
Components and their content embedded in record entities
etc.
etc.
Events for all entities exported in full
Completed timestamp

*Figure 20: Structured content of an XML export datafile*

In order to get an overview on how MCRS is exported, it is important to understand how its data is structured within the XML export data file. There are four main blocks that are relevant for any MCRS export. They are: the compliance block, the contextual metadata block, the services block and the event block.

- The compliance block consists of a list of all services and modules that are implemented by the MCRS exporting the data;
- The contextual metadata block holds the definitions of the contextual metadata elements in use by the entities being exported;
- The services block contains a series of individual service blocks, with each service block being a representation of the data from a single service or bundle of services.
- The event block is an aggregation of all of the events for every entity that is being exported completely. The reason why events are not listed in the services block is that several entities may participate in the same event. This includes entities from multiple services.

It is also important to mention that regarding MoReq2010 export operation, its MCRS export was conceived as a stream of entry data to ensure that an import can be processed in a single sequential pass through the data. As such, MoReq2010's approach to handling large MCRS XML exports is to split the exported data into multiple data files, as can be seen in Figure 21. However MoReq2010 does not specify how these exports should be split, only requiring that the original export should be able to be recreated from its comprising data files.

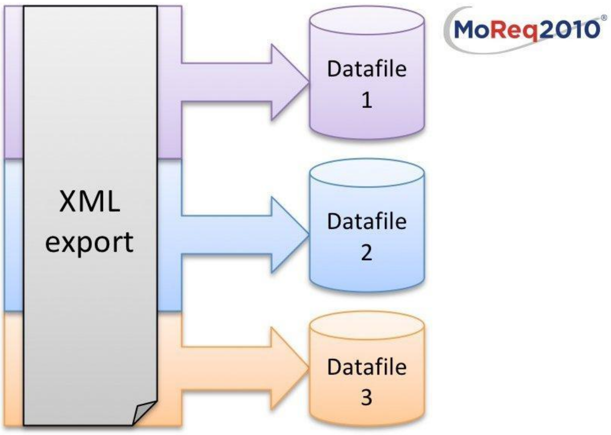


*Figure 21: A Large XML export may be split across multiple data files*

Another important aspect to highlight is that MoReq2010 defines that the components contents of a record, i.e. the physical object or digital sequence that constitutes the record, are exported through XML by converting the components to Base64 Binary Stream of Data.

### 4.2.2 MCRS in relation to the AIP format

MoReq2010 allows the export of system metadata elements and their values, contextual metadata elements and their values, record's disposal schedule, components of a records, the access control list and its access control entries, the users, groups and roles referred by these access control entities, and the events in the record's event history. The reason why MoReq2010 allows the export of all its different entities is because, as stated above, records can be exported for different reasons. Mostly the entities are exported along with the record to allow another MCRS to replicate the functionality and context associated with the record(s) being exported.

In the context of E-ARK and this deliverable, we are focused on the export due to a transfer to an archive. Therefore, entities exported with a record in MoReq2010 such as users, groups, roles, access control list, etc., lose their relevance as a mean to replicate functionality and context and assume a more historical relevance, i.e. they represent the history of the record but they are not necessary to properly preserve the record as intended in an archive. As a conclusion, in a MoReq2010 export to an archive we are interested in assuring: (1) that records metadata is preserved, and (2) that the record's component contents are properly managed to assure long-term preservation.

Therefore, using the E-ARK SIP (including SMURF (Semantically Marked Up Record Format) profile[53]) and AIP specification, a MoReq2010 export can be aligned to an E-ARK IP structure by:

- Exporting records, classes and aggregations from MCRS by following the mapping rules stated in the SMURF profile (items ①, ② on the diagram illustrating the SMURF SIP[54]);
- Including the EAD (and if needed also a subset of MoReq2010) elements as metadata in the "metadata" folder defined in the IP structure (items ③, ④, ⑤ on the diagram);
- Convert the record's components contents Base64 Binary Stream of Data to its original form and representation (not shown on the diagram);

The files and metadata from the previous steps constitute a representation of the record and consequently should be stored in the "representation" folder defined in the IP structure (item ⑨ on the diagram).

## 4.3 Geodata

This section is about Geographical data (short: geodata) [55] and how they are archived using the AIP format.

Geographic Information Systems (GISs) are used to model objects and their relationships in space. GIS enables us to store, query, edit, display, and analyse data in the system.

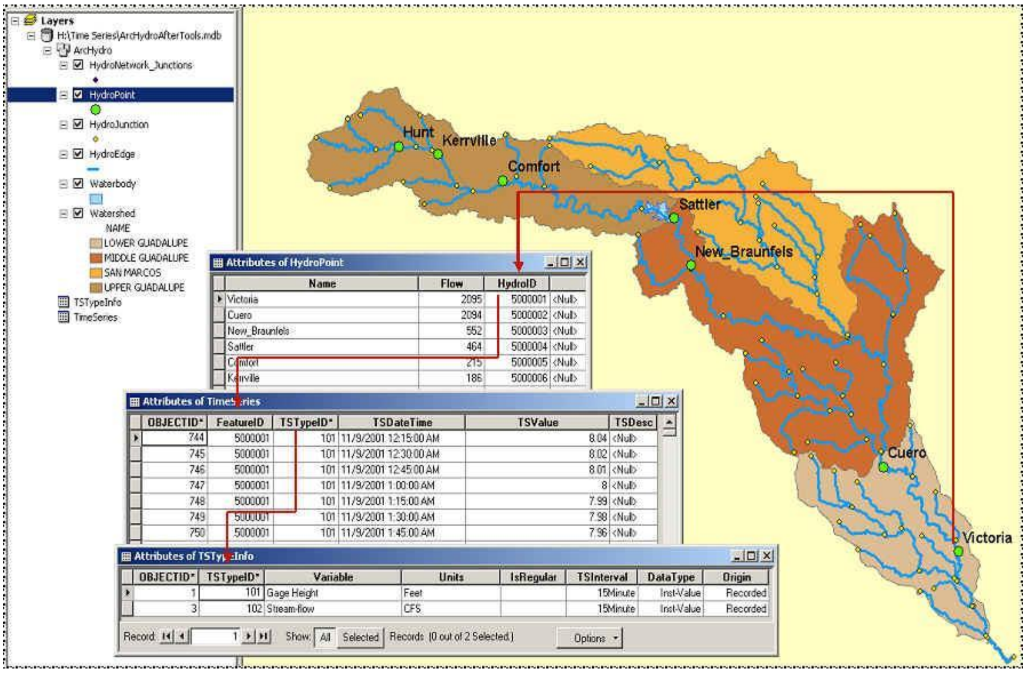
Geodata is a combination of the graphical representation of objects in space and their descriptions or attributes. Increasingly, geospatial formats include geospatially focused datasets or databases that contain primary information about a geographic location. In addition, ancillary and supplementary data – that can be either included or derived using spatial analysis – are considered necessary for the interpretation and re-use of the data.

---

[53] The SMURF profile for ERMS and SFSB can be found at http://eark-project.com/resources/project-deliverables as part of the official deliverable D.3.3 named E-ARK SMURF (semantically marked up record format)e (part of D3.3).

[54] Ibidem, p. 31 (Figure 12).

[55] Throughout this section we use the abbreviation "geodata" for geographical data.



Attributes of HydroPoint

| Name | Flow | HydroID | |
|---|---|---|---|
| Victoria | 2095 | 5000001 | <Null> |
| Cuero | 2094 | 5000002 | <Null> |
| New_Braunfels | 552 | 5000003 | <Null> |
| Sattler | 464 | 5000004 | <Null> |
| Comfort | 215 | 5000005 | <Null> |
| Kerrville | 186 | 5000006 | <Null> |

Attributes of TimeSeries

| OBJECTID* | FeatureID | TSTypeID* | TSDateTime | TSValue | TSDesc |
|---|---|---|---|---|---|
| 744 | 5000001 | 101 | 11/9/2001 12:15:00 AM | 8.04 | <Null> |
| 745 | 5000001 | 101 | 11/9/2001 12:30:00 AM | 8.02 | <Null> |
| 746 | 5000001 | 101 | 11/9/2001 12:45:00 AM | 8.01 | <Null> |
| 747 | 5000001 | 101 | 11/9/2001 1:00:00 AM | 8 | <Null> |
| 748 | 5000001 | 101 | 11/9/2001 1:15:00 AM | 7.99 | <Null> |
| 749 | 5000001 | 101 | 11/9/2001 1:30:00 AM | 7.98 | <Null> |
| 750 | 5000001 | 101 | 11/9/2001 1:45:00 AM | 7.96 | <Null> |

Attributes of TSTypeInfo

| OBJECTID* | TSTypeID* | Variable | Units | IsRegular | TSInterval | DataType | Origin |
|---|---|---|---|---|---|---|---|
| 1 | 101 | Gage Height | Feet | | 15Minute | Inst-Value | Recorded |
| 3 | 102 | Stream-flow | CFS | | 15Minute | Inst-Value | Recorded |

Record: 1 Records (0 out of 2 Selected.)

*Figure 22: Geodata is made of graphical representation and attribute tables*

The basis of geodata is generally either vector or raster which can be stored as a set of files or a database.

*Vector data* represents spatial objects as points, lines or polygons or, if complex, a combination of them. Descriptive or derived attributes are stored in tables (one row per spatial object).

*Raster data* represents spatial objects as cells in a matrix. Those cells can contain different types of values from binary to decimal numbers.

But for proper interpretation of geodata we also need some other elements that go beyond coordinates and attribute tables. The main elements are georeferencing, geoprocessing and visualization.

*Georeferencing* – Is a process of establishing a relationship between information (e.g., documents, datasets, maps, images, biographical information) and geographic locations through mechanisms such as the addition of place labels (e.g., place codes or toponyms) or the assignment of geographic coordinates. Geodata are defined by the Coordinate Reference System (CRS). Geographic coordinates depend very much on the coordinate system and cartographic projection used for the data in question. This data is essential and must be part of the archived geodata.

*Geoprocessing* – In order to derive useful information from a set of geodata, we need to use geo functions. This new information can be in the form of tables, maps, or list and is usually used for common queries or common reports. It can also be a basis for decision making like issuing building permits, setting real-estate tax rates, responding to changing demography trends. So that is why we need to document these processes.

*Visualization* – geodata visualization is a basis for the interpretation of data, since we can use different visual modalities (e.g. size, colour maps, patterns) to represent multiple values at the same time. The same source of geodata can be used to display different maps, and therefore does not present a unique representation of the data it contains. For instance, if we take a geodata layer of municipalities in Slovenia and classify them by the number of citizens, and then change the classification to citizens normalized by area, we get different maps from the same data. Geodata may or may not have visualisation data associated with it.

### 4.3.1 Geodata in relation to the AIP

Within the AIP geodata will be structured according to the representation-based model as shown in Figure 23.

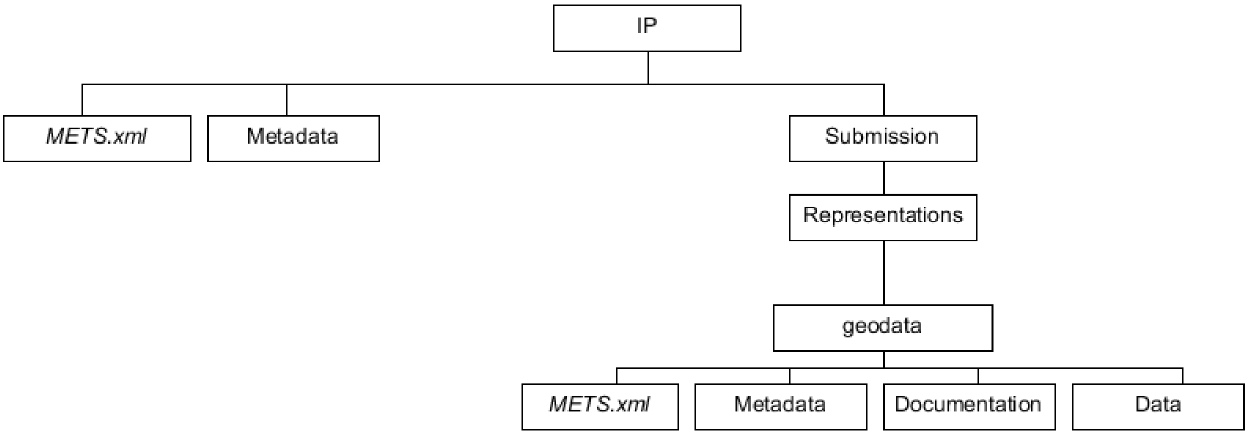


*Figure 23: Geodata representation within the AIP*

This means that geodata will be organised in the three directories “Metadata”, “Data”, and “Documentation”. The representation’s directory “geodata” bundles these directories. Note that the name of the “geodata” directory can be chosen freely.

#### 4.3.1.1 Data directory

The data directory contains one representation of geodata in a long term preservation format (GML for vector and GeoTIFF for Raster) and all information that is needed to properly render the information. Information needed for proper rendering of geodata can contain the elements described in the following sections.

##### *4.3.1.1.1 Graphical information*

Graphical information can be in vector or raster format.

Geodata in a vector format is organized in vector datasets which contain only one type of geometry (point, line, polygon …) per dataset. A dataset also contains a set of features (representations of real world objects) that are of the same topic or project. For instance, a roads dataset has all the roads in one dataset.

Vector data can be stored in different formats, such as SHP[56], KML[57], DXF[58], GML[59], etc.; the GML format as defined by the ISO19136:2007 standard was chosen as the long term preservation format.

It is also possible to store a representation of the data in the original format if that format is still commonly used and well documented as is the case with the ESRI[60] Shapefile (SHP) format. It is widely accepted and supported by most GIS software today.

Geodata in a raster format is stored in a "geo-enabled" raster file. Some common formats for raster geodata are: GeoTIFF[61], JPEG2000[62], BIM[63], GRID[64], ASCII GRID[65], TIFF[66], etc.; for long term preservation format we chose GeoTIFF or ordinary TIFF with a GML bounding box. Both GeoTIFF and GML bounding box need to contain the mandatory georeferencing information.

#### *4.3.1.1.2 Attribute information*

Attribute information in GML is already a part of the GML itself.

Attribute information in an ESRI Shapefile is stored in the mandatory *.dbf[67] file. Attribute definitions are required.

Attribute information in a Raster File can be stored in the value of the pixel itself. Attribute definitions are required if pixel value is not a grayscale value of an image.

#### *4.3.1.1.3 Georeferencing information - Coordinate Reference System (CRS)*

CRS tells us how to locate objects in the geodata on earth's surface. Elements of spatial reference system are projection, geodetic datum, and unit of measure. All the elements can be defined by an EPSG code[68] .

Georeferencing information in GML is a mandatory part of the file itself and it is embedded in the geodata file itself.

---

[56] http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf
[57] http://www.opengeospatial.org/standards/kml
[58] http://www.autodesk.com/techpubs/autocad/acad2000/dxf
[59] http://www.opengeospatial.org/standards/gml
[60] http://www.esri.com
[61] http://www.remotesensing.org/geotiff/spec/geotiffhome.html
[62] http://jpeg.org/jpeg2000
[63] http://www.buildingsmart.org
[64] ESRI Grid Format, http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=About_the_ESRI_Grid_format (Informal specification published by ESRI)
[65] http://desktop.arcgis.com/de/desktop/latest/manage-data/raster-and-images/esri-ascii-raster-format.htm
[66] http://partners.adobe.com/public/developer/tiff
[67] http://www.dbase.com/Knowledgebase/INT/db7_file_fmt.htm
[68] http://www.epsg.org

```
<gml:boundedBy>
   <gml:Envelope srsName="urn:x-ogc:def:crs:EPSG:6.6:4326">
       <gml:lowerCorner>50.23 9.23</gml:lowerCorner>
       <gml:upperCorner>50.31 9.27</gml:upperCorner>
   </gml:Envelope>
</gml:boundedBy>
```

The attribute “`srsName`” holds the value of the coordinate reference system code.

*Georeferencing information in ESRI Shapefile (shp)*

ESRI shapefile needs a <shapefilename>.prj file in order to be properly georeferenced. A Prj file is a txt file, containing a definition of the coordinate reference system and all of its elements.

```
PROJCS["NAD_1983_UTM_Zone_10N",GEOGCS["GCS_North_American_1983",
DATUM["D_North_American_1983",SPHEROID["GRS_1980",6378137,298.257222101]],
PRIMEM["Greenwich",0],UNIT["Degree",0.0174532925199433]],
PROJECTION["Transverse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",
0.0],PARAMETER["Central_Meridian",-123.0],PARAMETER["Scale_Factor",0.9996],
PARAMETER["Latitude_of_Origin",0.0],UNIT["Meter",1.0]]
```

*Georeferencing information in GeoTIFF*

GeoTIFF contains the CRS information in its header. A GeoTIFF raster needs to be validated for the existence of CRS metadata within its header. Otherwise CRS needs to be added as for an ordinary TIFF format.

*Georeferencing information in ordinary TIFF with GML bounding box*

If geodata comes in an ordinary TIFF files it should be accompanied by an additional “world” file. For example a D240143.tif file would be accompanied by a D240143.tfw file. That is a txt file, containing information about the coordinates and size of the first top left pixel.

```
0.42333
0.0
0.0
-0.42333
394250.00
```

CRS information is then provided within the GML vector file that represents the location of the raster file in space and accompanies an ordinary TIFF.

#### *4.3.1.1.4 Visualization information*

If geodata needs visualization information in order to be properly interpreted, we also require this information. A large number of GIS software enable an export of the symbology definitions to a special file, however they may be in a proprietary XML structure (like the *.qml in QGIS, *.tab in MapInfo…) or even in a binary format (*.lyr in ArcGIS). The only OGC standard for symbology is the OGC Styled Layer Description XML format (sld files). If the producer cannot provide the Archive with SLD files, these can be recreated from the description which is provided in the documentation in QGIS. Raster files can have a colour map associated with the pixel value.

The SLD standard is used for rendering geodata in OGC web services and is therefore an appropriate input for an easier DIP creation.

```
<StyledLayerDescriptor xmlns="http://www.opengis.net/sld"
       xmlns:ogc="http://www.opengis.net/ogc"
       xmlns:xlink="http://www.w3.org/1999/xlink"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       version="1.0.0"
       xsi:schemaLocation="http://www.opengis.net/sld StyledLayerDescriptor.xsd">
   <NamedLayer>
       <Name>Simple Point</Name>
       <UserStyle>
           <Title>SLD Cook Book: Simple Point</Title>
           <FeatureTypeStyle>
               <Rule>
                   <PointSymbolizer>
                       <Graphic>
                           <Mark>
                               <WellKnownName>circle</WellKnownName>
                               <Fill>
                                   <CssParameter name="fill">#FF0000</CssParameter>
                               </Fill>
                           </Mark>
                           <Size>6</Size>
                       </Graphic>
                   </PointSymbolizer>
               </Rule>
           </FeatureTypeStyle>
       </UserStyle>
   </NamedLayer>
</StyledLayerDescriptor>
```

*Listing 25: Example of an SLD file*

#### 4.3.1.2 Documentation directory

In the folder »Documentation« information is stored which will not be already included in the »Metadata« or »Data« folders. It will help us to understand the data-set in a wider social context and will provide a better understanding of the meaning, use and structure of the spatial data. The goal is to provide the end-user with all the necessary information for a proper understanding and interpretation of the geo-data set.

##### *4.3.1.2.1 Feature Catalogue*

The feature catalogue represents a logical structure of attributes. It provides a better understanding of the meaning, use and structure of the spatial data and provides a unified classification of spatial data in feature types (classes). Feature types are distinguished by their attributes (properties), by importance and by the relations between them.

##### *4.3.1.2.2 Visualisation and cartographic representation*

Data visualization provides an illustration and representation of spatial data. The catalogue of cartographic symbols is a collection of agreed cartographic symbols, which are used during the process of visualising spatial data sets to display objects in space. Cartographic symbols are shown in the legend, which explains their meaning.

For certain spatial data the visualization is already made by the producer or owner of a spatial data set in the form of (geo-located) raster images or paper maps. In these cases, it is reasonable to archive the visualization. For each spatial data set it is possible to produce any number of different visualizations with the appropriate software.

##### *4.3.1.2.3 The logical structure of layers and attribute definition*

The logical structure of layers shows the organization of the data layers at the level of logical tables contained in the database or in a connection of unstructured objects and their attributes organised in a GIS application. It also contains the descriptions of the attributes of each data layer and the code values for each attribute. This information is similar to information contained in Feature Catalogue, but it is described in a different manner. We can choose what to archive in the valorisation process.

##### *4.3.1.2.4 Table relations*

In the case of a complex system of tables, the documentation directory should contain diagrams of relations between tables in a database or within a GIS project in order to enable the reconstruction of queries and provide greater understanding of the usage of tables.

##### *4.3.1.2.5 Common queries*

A list and a description of the most common queries provides additional information about how the data set was used from the end user’s point of view. The main goal is to enable the re-creation of tools, which enable users to get information in the form of common queries and common reports, similarly to how they were used in the original production environment.

##### *4.3.1.2.6 Other contextual documentation*

This chapter combines all the documents (links) to the relevant documentation describing the lineage and provenance of the spatial data set. The list of documentation includes: user manuals, related practices in EU and worldwide, methodological rules, scientific articles, publications, etc.

#### 4.3.1.3 AIP Metadata Directory

We expect the usual metadata, like PREMIS and METS to be available here. Geodata specific contextual metadata will be within the EAD3 file. They are based on the EC Directive INSPIRE and were mapped to EAD3 elements.

#### *4.3.1.3.1 What is INSPIRE?*

The INSPIRE Directive[69] establishes an infrastructure for spatial information in Europe to support Community environmental policies, and policies or activities which may have an impact on the environment. Since, for the proper functioning of a Spatial Information Infrastructure, it is necessary for a user to be able to find spatial data sets and services and to establish whether they may be used and for what purpose, data owners should provide descriptions in the form of metadata for those spatial data sets and services. Since such metadata should be compatible and usable in a Community and trans-boundary context, the European Commission adopted Commission Regulation (EC) No. 1205/2008[70], implementing the INSPIRE Directive in regards to the metadata. INSPIRE Metadata Implementing Rules[71] refers to the above-mentioned Regulation and defines how the requirements of the Implementing Rules for Metadata can be applied using EN ISO 19115 and EN ISO 19119.

Selected INSPIRE Metadata elements could be used as archival descriptive metadata. Therefore the E-ARK project team mapped some of them onto the EAD3[72] metadata in order to enable automatic capture of descriptive metadata from the INSPIRE description of the spatial data sets. The INSPIRE XML should be a part of the “Medatata” folder, if it exists.

---

[69] Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE) was published in the official Journal on the 25th April 2007. The INSPIRE Directive entered into force on the 15th May 2007

[70] http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:32008R1205

[71] INSPIRE Metadata Implementing Rules: Technical Guidelines based on EN ISO 19115 and EN ISO 19119, version 1.3, 2013-10-29, http://inspire.jrc.ec.europa.eu/documents/Metadata/MD_IR_and_ISO_20131029.pdf

[72] http://www2.archivists.org/sites/all/files/TagLibrary-VersionEAD3.pdf

# 5 Appendices

## 5.1 Appendix A - METS.xml referencing representation METS.xml files

```
<?xml version='1.0' encoding='UTF-8'?>
<mets xmlns:ext="ExtensionMETS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns="http://www.loc.gov/METS/"
PROFILE="http://www.eark-project.com/METS/IP.xml" TYPE="AIP" OBJID="urn:uuid:03c870ee-da0b-4a1c-
8700-b8148a0e8b2b" LABEL="METS file describing the AIP matching the OBJID."
xsi:schemaLocation="http://www.loc.gov/METS/ schemas/mets_1_11.xsd http://www.w3.org/1999/xlink
schemas/xlink.xsd">
  <metsHdr RECORDSTATUS="NEW" CREATEDATE="2016-01-12T12:50:22">
        <agent TYPE="OTHER" ROLE="CREATOR" OTHERTYPE="SOFTWARE">
        <name>E-ARK earkweb</name>
        <note>VERSION=0.0.1</note>
        </agent>
        <metsDocumentID>METS.xml</metsDocumentID>
  </metsHdr>
  <amdSec ID="ID201aa3b0-2ec6-4387-9ac2-b84e58b35e5d">
        <digiprovMD ID="ID800ac4dc-1654-45fa-afa6-ace2646c9732">
        <mdRef MIMETYPE="text/plain" xlink:href="file://./metadata/earkweb.log" LOCTYPE="URL"
CREATED="2016-01-12T12:50:22"
CHECKSUM="379f3c6cb806ae964f2c18ee1d6b0d471502f416898431088e36b1168361f0d8" xlink:type="simple"
ID="IDda80fdbb-727e-4d80-aef6-9422b9a1b4af" MDTYPE="OTHER" CHECKSUMTYPE="SHA-256"/>
        </digiprovMD>
        <digiprovMD ID="ID6dce6498-6941-494f-82cf-ff186ecc55e6">
        <mdRef MIMETYPE="application/xml" xlink:href="file://./metadata/preservation/premis.xml"
LOCTYPE="URL" CREATED="2016-01-12T12:50:22"
CHECKSUM="64f1d04ed53946d8071f5af30730938a41d0f8b5fdb8ccea28ba6ac00ee64978" xlink:type="simple"
ID="ID338f76e4-9aaa-468e-849d-6398093b750b" MDTYPE="PREMIS" CHECKSUMTYPE="SHA-256"/>
        </digiprovMD>
  </amdSec>
  <fileSec>
        <fileGrp USE="general filegroup" ID="ID7cf197b6-9cb9-4afe-a740-b3e33a1a9f5e">
        <file MIMETYPE="application/xml" USE="Datafile" CHECKSUMTYPE="SHA-256" CREATED="2016-01-
12T12:49:10" CHECKSUM="a7b2dc0dba5e7c09ec91a19d35f6cac35eab6f8aaa7cd4257c0791c912721553"
ID="ID6b9255fe-69f8-44e6-9fd8-3c39168908af" SIZE="6307">
        <FLocat xlink:href="file://./submission/METS.xml" xlink:type="simple" LOCTYPE="URL"/>
        </file>
        <file MIMETYPE="application/xml" USE="Datafile" CHECKSUMTYPE="SHA-256" CREATED="2016-01-
12T12:50:22" CHECKSUM="dc8133202b61715fa2e52229bda041aed4e586f4b3b21109a40222391ac614f8"
ID="IDd40357c4-9729-40d0-8631-035376b092bf" SIZE="145430">
        <FLocat xlink:href="file://./schemas/IP.xsd" xlink:type="simple" LOCTYPE="URL"/>
        </file>
        <file MIMETYPE="application/xml" USE="Datafile" CHECKSUMTYPE="SHA-256" CREATED="2016-01-
12T12:50:22" CHECKSUM="a20b879caff4e304ee8900aa9652a8a532c54c450e89381ba034d1f7cf1ab1c9"
ID="ID88320c0f-b512-4e7a-9534-e6bd527f7430" SIZE="133018">
        <FLocat xlink:href="file://./schemas/mets_1_11.xsd" xlink:type="simple" LOCTYPE="URL"/>
        </file>
        <file MIMETYPE="application/xml" USE="Datafile" CHECKSUMTYPE="SHA-256" CREATED="2016-01-
12T12:50:22" CHECKSUM="0d47e53f21e2e44f48794287afae759bace86a0622a1dfb1fa44a1a982abddd7"
ID="ID464556fc-f41a-4c77-a54a-cf5d3852d843" SIZE="65126">
        <FLocat xlink:href="file://./schemas/premis-v2-2.xsd" xlink:type="simple" LOCTYPE="URL"/>
        </file>
        <file MIMETYPE="application/xml" USE="Datafile" CHECKSUMTYPE="SHA-256" CREATED="2016-01-
12T12:50:22" CHECKSUM="f1f5bb6003165cdd8f6c1fcc32f8fd1f965e1681010f3b9806d9460bcffa8a3c"
ID="IDa9d18c84-782f-4777-8864-9f589d759811" SIZE="3180">
        <FLocat xlink:href="file://./schemas/xlink.xsd" xlink:type="simple" LOCTYPE="URL"/>
```

```
       </file>
       <file MIMETYPE="application/xml" USE="Datafile" CHECKSUMTYPE="SHA-256" CREATED="2016-01-
12T12:50:18" CHECKSUM="04a9710df82684f55a753717856b3229e0e443b1b01e8d038b14beca61e6df70"
ID="IDf785002f-4d72-4370-a224-8b5b26db72c8" SIZE="2834">
       <FLocat xlink:href="file://./representations/image_mig-1/METS.xml" xlink:type="simple"
LOCTYPE="URL"/>
       </file>
       <file MIMETYPE="application/xml" USE="Datafile" CHECKSUMTYPE="SHA-256" CREATED="2016-01-
12T12:50:18" CHECKSUM="c00987ba3f9944117757c479292358832865c872033918c7715c648cc2ae6afd"
ID="ID0874e8e4-8c54-4fb9-ae00-c0f891aebaab" SIZE="2860">
       <FLocat xlink:href="file://./representations/docs_mig-1/METS.xml" xlink:type="simple"
LOCTYPE="URL"/>
       </file>
       </fileGrp>
  </fileSec>
  <structMap TYPE="physical" LABEL="earkstructmap">
       <div LABEL="03c870ee-da0b-4a1c-8700-b8148a0e8b2b">
       <div LABEL="submission">
       <mptr xlink:href="file://./submission/METS.xml" xlink:title="Mets file describing
representation: submission of AIP: urn:uuid:03c870ee-da0b-4a1c-8700-b8148a0e8b2b." LOCTYPE="URL"
ID="ID588691e6-03f6-4ef6-9acc-d05baefa1ad7"/>
       <fptr FILEID="ID6b9255fe-69f8-44e6-9fd8-3c39168908af"/>
       </div>
       <div LABEL="metadata">
       <fptr FILEID="IDda80fdbb-727e-4d80-aef6-9422b9a1b4af"/>
       <fptr FILEID="ID338f76e4-9aaa-468e-849d-6398093b750b"/>
       </div>
       <div LABEL="schemas">
       <fptr FILEID="IDd40357c4-9729-40d0-8631-035376b092bf"/>
       <fptr FILEID="ID88320c0f-b512-4e7a-9534-e6bd527f7430"/>
       <fptr FILEID="ID464556fc-f41a-4c77-a54a-cf5d3852d843"/>
       <fptr FILEID="IDa9d18c84-782f-4777-8864-9f589d759811"/>
       </div>
       <div LABEL="representations"/>
       <div LABEL="representations/image_mig-1">
       <mptr xlink:href="file://./representations/image_mig-1/METS.xml" xlink:title="Mets file
describing representation: image_mig-1 of AIP: urn:uuid:03c870ee-da0b-4a1c-8700-b8148a0e8b2b."
LOCTYPE="URL" ID="IDb639b428-3185-4a17-9152-999d7a23fa27"/>
       <fptr FILEID="IDf785002f-4d72-4370-a224-8b5b26db72c8"/>
       </div>
       <div LABEL="representations/docs_mig-1">
       <mptr xlink:href="file://./representations/docs_mig-1/METS.xml" xlink:title="Mets file
describing representation: docs_mig-1 of AIP: urn:uuid:03c870ee-da0b-4a1c-8700-b8148a0e8b2b."
LOCTYPE="URL" ID="IDf99cca1f-cfe2-4112-b86e-64ba5e741100"/>
       <fptr FILEID="ID0874e8e4-8c54-4fb9-ae00-c0f891aebaab"/>
       </div>
       </div>
  </structMap>
  <structMap TYPE="logical" LABEL="Simple AIP structuring">
       <div LABEL="Package structure">
       <div LABEL="metadata files">
       <fptr FILEID="IDda80fdbb-727e-4d80-aef6-9422b9a1b4af"/>
       <fptr FILEID="ID338f76e4-9aaa-468e-849d-6398093b750b"/>
       </div>
       <div LABEL="schema files">
       <fptr FILEID="IDd40357c4-9729-40d0-8631-035376b092bf"/>
       <fptr FILEID="ID88320c0f-b512-4e7a-9534-e6bd527f7430"/>
       <fptr FILEID="ID464556fc-f41a-4c77-a54a-cf5d3852d843"/>
       <fptr FILEID="IDa9d18c84-782f-4777-8864-9f589d759811"/>
       </div>
       </div>
  </structMap>
</mets>
```

## 5.2 Appendix B - METS.xml describing a representation

```
<?xml version='1.0' encoding='UTF-8'?>
<mets xmlns:ext="ExtensionMETS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns="http://www.loc.gov/METS/"
PROFILE="http://www.eark-project.com/METS/IP.xml" TYPE="AIP" OBJID="docs_mig-1" LABEL="METS file
describing the AIP matching the OBJID."
xsi:schemaLocation="http://www.loc.gov/METS/ ../../schemas/mets_1_11.xsd
http://www.w3.org/1999/xlink ../../schemas/xlink.xsd">
  <metsHdr RECORDSTATUS="NEW" CREATEDATE="2016-01-12T12:50:18">
       <agent TYPE="OTHER" ROLE="CREATOR" OTHERTYPE="SOFTWARE">
       <name>E-ARK earkweb</name>
       <note>VERSION=0.0.1</note>
       </agent>
       <metsDocumentID>METS.xml</metsDocumentID>
  </metsHdr>
  <amdSec ID="ID5851970c-6156-48da-afbc-008f03e7e682">
       <digiprovMD ID="IDabf152e2-c7ec-4055-be62-4863d471ebee">
       <mdRef MIMETYPE="application/xml" xlink:href="file://./metadata/preservation/premis.xml"
LOCTYPE="URL" CREATED="2016-01-12T12:50:18"
CHECKSUM="086b5716c54aea78f42863349a0054c6e8acb1f4ff289440e7e94b1cfce905d6" xlink:type="simple"
ID="ID825b8faf-24ab-4c44-b9a6-e5a02629ec59" MDTYPE="PREMIS" CHECKSUMTYPE="SHA-256"/>
       </digiprovMD>
  </amdSec>
  <fileSec>
       <fileGrp USE="general filegroup" ID="ID24ef7182-f8e0-4fa4-adfb-97006df2d94a">
       <file MIMETYPE="application/pdf" USE="Datafile" CHECKSUMTYPE="SHA-256" CREATED="2016-01-
12T12:49:24" CHECKSUM="2f89b9d05d642075d99078095b3c82128981ea0ca562d25810c1f23dfb835a7b"
ID="IDdcfc4cdd-3661-4dfb-81b1-ab8dd35d4e51" SIZE="2530049">
       <FLocat xlink:href="file://./data/Suleiman the Magnificent.pdf" xlink:type="simple"
LOCTYPE="URL"/>
       </file>
       <file MIMETYPE="application/pdf" USE="Datafile" CHECKSUMTYPE="SHA-256" CREATED="2016-01-
12T12:49:30" CHECKSUM="baede90f382a3ef0ad5a9fe6c42b1dfc0510fca667f5a424c0e478397f6024e3"
ID="ID935b7149-08a5-40b8-b020-273986aa4bdc" SIZE="7603618">
       <FLocat xlink:href="file://./data/Charlemagne.pdf" xlink:type="simple" LOCTYPE="URL"/>
       </file>
       </fileGrp>
  </fileSec>
  <structMap TYPE="physical" LABEL="earkstructmap">
       <div LABEL="docs_mig-1">
       <div LABEL="metadata">
       <fptr FILEID="ID825b8faf-24ab-4c44-b9a6-e5a02629ec59"/>
       </div>
       <div LABEL="data">
       <fptr FILEID="IDdcfc4cdd-3661-4dfb-81b1-ab8dd35d4e51"/>
       <fptr FILEID="ID935b7149-08a5-40b8-b020-273986aa4bdc"/>
       </div>
       </div>
  </structMap>
  <structMap TYPE="logical" LABEL="Simple AIP structuring">
       <div LABEL="Package structure">
       <div LABEL="metadata files">
       <fptr FILEID="ID825b8faf-24ab-4c44-b9a6-e5a02629ec59"/>
       </div>
       <div LABEL="schema files"/>
       <div LABEL="files from /data">
       <fptr FILEID="IDdcfc4cdd-3661-4dfb-81b1-ab8dd35d4e51"/>
       <fptr FILEID="ID935b7149-08a5-40b8-b020-273986aa4bdc"/>
       </div>
       </div>
  </structMap>
</mets>
```

## 5.3 Appendix C - PREMIS.xml describing events on package level

```
<?xml version='1.0' encoding='UTF-8'?>
<premis xmlns="info:lc/xmlns/premis-v2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
version="2.0"
       xsi:schemaLocation="info:lc/xmlns/premis-v2 ../../schemas/premis-v2-2.xsd">
       <object xmlID="IDefbef34b-c39c-4f93-8acd-86d12e8dab15" xsi:type="representation">
       <objectIdentifier>
              <objectIdentifierType>repository</objectIdentifierType>
              <objectIdentifierValue>03c870ee-da0b-4a1c-8700-b8148a0e8b2b</objectIdentifierValue>
       </objectIdentifier>
       </object>
       <event>
       <eventIdentifier>
              <eventIdentifierType>local</eventIdentifierType>
              <eventIdentifierValue>IDc804890d-1760-4673-84a4-7a599ef763c0</eventIdentifierValue>
       </eventIdentifier>
       <eventType>currently not in vocabulary</eventType>
       <eventDateTime>2016-01-12T12:48:59</eventDateTime>
       <eventOutcomeInformation>
              <eventOutcome>success</eventOutcome>
       </eventOutcomeInformation>
       <linkingAgentIdentifier>
              <linkingAgentIdentifierType>software</linkingAgentIdentifierType>
              <linkingAgentIdentifierValue>SIPDeliveryValidation</linkingAgentIdentifierValue>
       </linkingAgentIdentifier>
       <linkingObjectIdentifier>
              <linkingObjectIdentifierType>repository</linkingObjectIdentifierType>
              <linkingObjectIdentifierValue>03c870ee-da0b-4a1c-8700-
b8148a0e8b2b</linkingObjectIdentifierValue>
       </linkingObjectIdentifier>
       </event>
       <event>
       <eventIdentifier>
              <eventIdentifierType>local</eventIdentifierType>
              <eventIdentifierValue>ID1a03b3dd-3bd8-47e3-9fcd-9a1fddbc8f98</eventIdentifierValue>
       </eventIdentifier>
       <eventType>identifier assignment</eventType>
       <eventDateTime>2016-01-12T12:49:02</eventDateTime>
       <eventOutcomeInformation>
              <eventOutcome>success</eventOutcome>
       </eventOutcomeInformation>
       <linkingAgentIdentifier>
              <linkingAgentIdentifierType>software</linkingAgentIdentifierType>
              <linkingAgentIdentifierValue>IdentifierAssignment</linkingAgentIdentifierValue>
       </linkingAgentIdentifier>
       <linkingObjectIdentifier>
              <linkingObjectIdentifierType>repository</linkingObjectIdentifierType>
```

```
                <linkingObjectIdentifierValue>03c870ee-da0b-4a1c-8700-
b8148a0e8b2b</linkingObjectIdentifierValue>
        </linkingObjectIdentifier>
        </event>
        <event>
        <eventIdentifier>
                <eventIdentifierType>local</eventIdentifierType>
                <eventIdentifierValue>IDedb3dfcd-ead5-4db0-8433-c5c2b4dd4c12</eventIdentifierValue>
        </eventIdentifier>
        <eventType>currently not in vocabulary</eventType>
        <eventDateTime>2016-01-12T12:49:06</eventDateTime>
        <eventOutcomeInformation>
                <eventOutcome>success</eventOutcome>
        </eventOutcomeInformation>
        <linkingAgentIdentifier>
                <linkingAgentIdentifierType>software</linkingAgentIdentifierType>
                <linkingAgentIdentifierValue>SIPExtraction</linkingAgentIdentifierValue>
        </linkingAgentIdentifier>
        <linkingObjectIdentifier>
                <linkingObjectIdentifierType>repository</linkingObjectIdentifierType>
                <linkingObjectIdentifierValue>03c870ee-da0b-4a1c-8700-
b8148a0e8b2b</linkingObjectIdentifierValue>
        </linkingObjectIdentifier>
        </event>
        <event>
        <eventIdentifier>
                <eventIdentifierType>local</eventIdentifierType>
                <eventIdentifierValue>ID89438e13-d69e-4467-bc01-f36eef4e8bc9</eventIdentifierValue>
        </eventIdentifier>
        <eventType>currently not in vocabulary</eventType>
        <eventDateTime>2016-01-12T12:49:11</eventDateTime>
        <eventOutcomeInformation>
                <eventOutcome>success</eventOutcome>
        </eventOutcomeInformation>
        <linkingAgentIdentifier>
                <linkingAgentIdentifierType>software</linkingAgentIdentifierType>
                <linkingAgentIdentifierValue>SIPRestructuring</linkingAgentIdentifierValue>
        </linkingAgentIdentifier>
        <linkingObjectIdentifier>
                <linkingObjectIdentifierType>repository</linkingObjectIdentifierType>
                <linkingObjectIdentifierValue>03c870ee-da0b-4a1c-8700-
b8148a0e8b2b</linkingObjectIdentifierValue>
        </linkingObjectIdentifier>
        </event>
        <event>
        <eventIdentifier>
                <eventIdentifierType>local</eventIdentifierType>
                <eventIdentifierValue>ID339ba266-8dbe-4a18-b06b-e9c5de0c0824</eventIdentifierValue>
        </eventIdentifier>
        <eventType>SIP validation</eventType>
        <eventDateTime>2016-01-12T12:49:14</eventDateTime>
        <eventOutcomeInformation>
                <eventOutcome>success</eventOutcome>
        </eventOutcomeInformation>
        <linkingAgentIdentifier>
                <linkingAgentIdentifierType>software</linkingAgentIdentifierType>
                <linkingAgentIdentifierValue>SIPValidation</linkingAgentIdentifierValue>
        </linkingAgentIdentifier>
        <linkingObjectIdentifier>
```

```
                <linkingObjectIdentifierType>repository</linkingObjectIdentifierType>
                <linkingObjectIdentifierValue>03c870ee-da0b-4a1c-8700-
b8148a0e8b2b</linkingObjectIdentifierValue>
        </linkingObjectIdentifier>
        </event>
        <event>
        <eventIdentifier>
                <eventIdentifierType>local</eventIdentifierType>
                <eventIdentifierValue>ID75897c91-1ca1-4f83-a1be-4ad95ff4dc20</eventIdentifierValue>
        </eventIdentifier>
        <eventType>migration</eventType>
        <eventDateTime>2016-01-12T12:49:20</eventDateTime>
        <eventOutcomeInformation>
                <eventOutcome>success</eventOutcome>
        </eventOutcomeInformation>
        <linkingAgentIdentifier>
                <linkingAgentIdentifierType>software</linkingAgentIdentifierType>
                <linkingAgentIdentifierValue>MigrationProcess</linkingAgentIdentifierValue>
        </linkingAgentIdentifier>
        <linkingObjectIdentifier>
                <linkingObjectIdentifierType>repository</linkingObjectIdentifierType>
                <linkingObjectIdentifierValue>03c870ee-da0b-4a1c-8700-
b8148a0e8b2b</linkingObjectIdentifierValue>
        </linkingObjectIdentifier>
        </event>
        <event>
        <eventIdentifier>
                <eventIdentifierType>local</eventIdentifierType>
                <eventIdentifierValue>ID02c40401-6a8e-4959-a6c0-6de2f56fd4d8</eventIdentifierValue>
        </eventIdentifier>
        <eventType>migration</eventType>
        <eventDateTime>2016-01-12T12:49:20</eventDateTime>
        <eventOutcomeInformation>
                <eventOutcome>success</eventOutcome>
        </eventOutcomeInformation>
        <linkingAgentIdentifier>
                <linkingAgentIdentifierType>software</linkingAgentIdentifierType>
                <linkingAgentIdentifierValue>MigrationProcess</linkingAgentIdentifierValue>
        </linkingAgentIdentifier>
        <linkingObjectIdentifier>
                <linkingObjectIdentifierType>repository</linkingObjectIdentifierType>
                <linkingObjectIdentifierValue>03c870ee-da0b-4a1c-8700-
b8148a0e8b2b</linkingObjectIdentifierValue>
        </linkingObjectIdentifier>
        </event>
        <event>
        <eventIdentifier>
                <eventIdentifierType>local</eventIdentifierType>
                <eventIdentifierValue>ID981ee060-bb14-482d-8704-7a043cb3c70c</eventIdentifierValue>
        </eventIdentifier>
        <eventType>currently not in vocabulary</eventType>
        <eventDateTime>2016-01-12T12:49:20</eventDateTime>
        <eventOutcomeInformation>
                <eventOutcome>success</eventOutcome>
        </eventOutcomeInformation>
        <linkingAgentIdentifier>
                <linkingAgentIdentifierType>software</linkingAgentIdentifierType>
                <linkingAgentIdentifierValue>AIPMigrations</linkingAgentIdentifierValue>
        </linkingAgentIdentifier>
```

```
       <linkingObjectIdentifier>
              <linkingObjectIdentifierType>repository</linkingObjectIdentifierType>
              <linkingObjectIdentifierValue>03c870ee-da0b-4a1c-8700-
b8148a0e8b2b</linkingObjectIdentifierValue>
       </linkingObjectIdentifier>
       </event>
       <event>
       <eventIdentifier>
              <eventIdentifierType>local</eventIdentifierType>
              <eventIdentifierValue>ID27c70b58-e4de-4c03-a03a-3878e1b73598</eventIdentifierValue>
       </eventIdentifier>
       <eventType>migration</eventType>
       <eventDateTime>2016-01-12T12:49:24</eventDateTime>
       <eventOutcomeInformation>
              <eventOutcome>success</eventOutcome>
       </eventOutcomeInformation>
       <linkingAgentIdentifier>
              <linkingAgentIdentifierType>software</linkingAgentIdentifierType>
              <linkingAgentIdentifierValue>MigrationProcess</linkingAgentIdentifierValue>
       </linkingAgentIdentifier>
       <linkingObjectIdentifier>
              <linkingObjectIdentifierType>repository</linkingObjectIdentifierType>
              <linkingObjectIdentifierValue>03c870ee-da0b-4a1c-8700-
b8148a0e8b2b</linkingObjectIdentifierValue>
       </linkingObjectIdentifier>
       </event>
       <event>
       <eventIdentifier>
              <eventIdentifierType>local</eventIdentifierType>
              <eventIdentifierValue>ID96391e4e-d7c6-4a80-bf1f-a63891ee1bdc</eventIdentifierValue>
       </eventIdentifier>
       <eventType>migration</eventType>
       <eventDateTime>2016-01-12T12:49:30</eventDateTime>
       <eventOutcomeInformation>
              <eventOutcome>success</eventOutcome>
       </eventOutcomeInformation>
       <linkingAgentIdentifier>
              <linkingAgentIdentifierType>software</linkingAgentIdentifierType>
              <linkingAgentIdentifierValue>MigrationProcess</linkingAgentIdentifierValue>
       </linkingAgentIdentifier>
       <linkingObjectIdentifier>
              <linkingObjectIdentifierType>repository</linkingObjectIdentifierType>
              <linkingObjectIdentifierValue>03c870ee-da0b-4a1c-8700-
b8148a0e8b2b</linkingObjectIdentifierValue>
       </linkingObjectIdentifier>
       </event>
       <event>
       <eventIdentifier>
              <eventIdentifierType>local</eventIdentifierType>
              <eventIdentifierValue>ID17b1808a-dfb6-4076-96af-6d0c58343d9c</eventIdentifierValue>
       </eventIdentifier>
       <eventType>currently not in vocabulary</eventType>
       <eventDateTime>2016-01-12T12:50:06</eventDateTime>
       <eventOutcomeInformation>
              <eventOutcome>success</eventOutcome>
       </eventOutcomeInformation>
       <linkingAgentIdentifier>
              <linkingAgentIdentifierType>software</linkingAgentIdentifierType>
             <linkingAgentIdentifierValue>AIPCheckMigrationProgress</linkingAgentIdentifierValue>
```

```
        </linkingAgentIdentifier>
        <linkingObjectIdentifier>
                <linkingObjectIdentifierType>repository</linkingObjectIdentifierType>
                <linkingObjectIdentifierValue>03c870ee-da0b-4a1c-8700-
b8148a0e8b2b</linkingObjectIdentifierValue>
        </linkingObjectIdentifier>
        </event>
        <event>
        <eventIdentifier>
                <eventIdentifierType>local</eventIdentifierType>
                <eventIdentifierValue>IDd9ddf486-c510-4f0d-9ccb-7660dc64ef24</eventIdentifierValue>
        </eventIdentifier>
        <eventType>currently not in vocabulary</eventType>
        <eventDateTime>2016-01-12T12:50:06</eventDateTime>
        <eventOutcomeInformation>
                <eventOutcome>success</eventOutcome>
        </eventOutcomeInformation>
        <linkingAgentIdentifier>
                <linkingAgentIdentifierType>software</linkingAgentIdentifierType>
                <linkingAgentIdentifierValue>MigrationsComplete</linkingAgentIdentifierValue>
        </linkingAgentIdentifier>
        <linkingObjectIdentifier>
                <linkingObjectIdentifierType>repository</linkingObjectIdentifierType>
                <linkingObjectIdentifierValue>03c870ee-da0b-4a1c-8700-
b8148a0e8b2b</linkingObjectIdentifierValue>
        </linkingObjectIdentifier>
        </event>
        <event>
        <eventIdentifier>
                <eventIdentifierType>local</eventIdentifierType>
                <eventIdentifierValue>ID61ef786d-3fee-4f8c-876a-5f496f5f2254</eventIdentifierValue>
        </eventIdentifier>
        <eventType>currently not in vocabulary</eventType>
        <eventDateTime>2016-01-12T12:50:11</eventDateTime>
        <eventOutcomeInformation>
                <eventOutcome>success</eventOutcome>
        </eventOutcomeInformation>
        <linkingAgentIdentifier>
                <linkingAgentIdentifierType>software</linkingAgentIdentifierType>

        <linkingAgentIdentifierValue>CreatePremisAfterMigration</linkingAgentIdentifierValue>
        </linkingAgentIdentifier>
        <linkingObjectIdentifier>
                <linkingObjectIdentifierType>repository</linkingObjectIdentifierType>
                <linkingObjectIdentifierValue>03c870ee-da0b-4a1c-8700-
b8148a0e8b2b</linkingObjectIdentifierValue>
        </linkingObjectIdentifier>
        </event>
        <event>
        <eventIdentifier>
                <eventIdentifierType>local</eventIdentifierType>
                <eventIdentifierValue>IDbce51331-b032-4662-bd96-6cd91fbf0711</eventIdentifierValue>
        </eventIdentifier>
        <eventType>currently not in vocabulary</eventType>
        <eventDateTime>2016-01-12T12:50:18</eventDateTime>
        <eventOutcomeInformation>
                <eventOutcome>success</eventOutcome>
        </eventOutcomeInformation>
        <linkingAgentIdentifier>
```

```
                <linkingAgentIdentifierType>software</linkingAgentIdentifierType>

        <linkingAgentIdentifierValue>AIPRepresentationMetsCreation</linkingAgentIdentifierValue>
        </linkingAgentIdentifier>
        <linkingObjectIdentifier>
                <linkingObjectIdentifierType>repository</linkingObjectIdentifierType>
                <linkingObjectIdentifierValue>03c870ee-da0b-4a1c-8700-
b8148a0e8b2b</linkingObjectIdentifierValue>
        </linkingObjectIdentifier>
        </event>
        <event>
        <eventIdentifier>
                <eventIdentifierType>local</eventIdentifierType>
                <eventIdentifierValue>ID771bed8b-a5d1-495c-b354-423b5254aab5</eventIdentifierValue>
        </eventIdentifier>
        <eventType>not in vocabulary</eventType>
        <eventDateTime>2016-01-12T12:50:22</eventDateTime>
        <eventOutcomeInformation>
                <eventOutcome>success</eventOutcome>
        </eventOutcomeInformation>
        <linkingAgentIdentifier>
                <linkingAgentIdentifierType>software</linkingAgentIdentifierType>
                <linkingAgentIdentifierValue>AIPPackageMetsCreation</linkingAgentIdentifierValue>
        </linkingAgentIdentifier>
        <linkingObjectIdentifier>
                <linkingObjectIdentifierType>repository</linkingObjectIdentifierType>
                <linkingObjectIdentifierValue>03c870ee-da0b-4a1c-8700-
b8148a0e8b2b</linkingObjectIdentifierValue>
        </linkingObjectIdentifier>
        </event>
        <event>
        <eventIdentifier>
                <eventIdentifierType>local</eventIdentifierType>
                <eventIdentifierValue>ID2d634dac-82c6-40c2-af4c-aa340e132a83</eventIdentifierValue>
        </eventIdentifier>
        <eventType>AIP validation</eventType>
        <eventDateTime>2016-01-12T12:50:28</eventDateTime>
        <eventOutcomeInformation>
                <eventOutcome>success</eventOutcome>
        </eventOutcomeInformation>
        <linkingAgentIdentifier>
                <linkingAgentIdentifierType>software</linkingAgentIdentifierType>
                <linkingAgentIdentifierValue>AIPValidation</linkingAgentIdentifierValue>
        </linkingAgentIdentifier>
        <linkingObjectIdentifier>
                <linkingObjectIdentifierType>repository</linkingObjectIdentifierType>
                <linkingObjectIdentifierValue>03c870ee-da0b-4a1c-8700-
b8148a0e8b2b</linkingObjectIdentifierValue>
        </linkingObjectIdentifier>
        </event>
        <agent>
        <agentIdentifier>
                <agentIdentifierType>LOCAL</agentIdentifierType>
                <agentIdentifierValue>earkweb</agentIdentifierValue>
        </agentIdentifier>
        <agentName>E-ARK AIP to DIP Converter</agentName>
        <agentType>Software</agentType>
        </agent>
</premis>
```

## 5.4 Appendix D - PREMIS.xml describing migration events (representation level)

```xml
<?xml version='1.0' encoding='UTF-8'?>
<premis xmlns="info:lc/xmlns/premis-v2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
version="2.0" xsi:schemaLocation="info:lc/xmlns/premis-v2 ../../schemas/premis-v2-2.xsd">
  <object xmlID="IDea32f707-2158-41ca-a96c-59f33bc76acf" xsi:type="representation">
       <objectIdentifier>
       <objectIdentifierType>repository</objectIdentifierType>
       <objectIdentifierValue>package-id-goes-here-?</objectIdentifierValue>
       </objectIdentifier>
  </object>
  <object xmlID="ID860cc897-b43f-4b1b-a4f0-b71a90e54bb9" xsi:type="file">
       <objectIdentifier>
       <objectIdentifierType>filepath</objectIdentifierType>
       <objectIdentifierValue>file://./data/Suleiman the Magnificent.pdf</objectIdentifierValue>
       </objectIdentifier>
       <objectCharacteristics>
       <compositionLevel>0</compositionLevel>
       <fixity>
       <messageDigestAlgorithm>SHA-256</messageDigestAlgorithm>

       <messageDigest>2f89b9d05d642075d99078095b3c82128981ea0ca562d25810c1f23dfb835a7b</messageDi
gest>
       <messageDigestOriginator>hashlib</messageDigestOriginator>
       </fixity>
       <size>2530049</size>
       <format>
       <formatRegistry>
       <formatRegistryName>PRONOM</formatRegistryName>
       <formatRegistryKey>fmt/18</formatRegistryKey>
       <formatRegistryRole>identification</formatRegistryRole>
       </formatRegistry>
       </format>
       </objectCharacteristics>
       <relationship>
       <relationshipType>derivation</relationshipType>
       <relationshipSubType>has source</relationshipSubType>
       <relatedObjectIdentification>
       <relatedObjectIdentifierType>filepath</relatedObjectIdentifierType>
       <relatedObjectIdentifierValue>file://./../../submission/representations/docs/data/Suleiman
the Magnificent.pdf</relatedObjectIdentifierValue>
       <relatedObjectSequence>0</relatedObjectSequence>
       </relatedObjectIdentification>
       <relatedEventIdentification>
       <relatedEventIdentifierType>local</relatedEventIdentifierType>
       <relatedEventIdentifierValue>IDdd7fe35b-3722-449f-aa22-
019549b43b40</relatedEventIdentifierValue>
       <relatedEventSequence>1</relatedEventSequence>
       </relatedEventIdentification>
       </relationship>
  </object>
  <object xmlID="IDf0972940-60b9-4e4e-bc37-12b41c1253b5" xsi:type="file">
       <objectIdentifier>
       <objectIdentifierType>filepath</objectIdentifierType>
       <objectIdentifierValue>file://./data/Charlemagne.pdf</objectIdentifierValue>
       </objectIdentifier>
       <objectCharacteristics>
       <compositionLevel>0</compositionLevel>
       <fixity>
       <messageDigestAlgorithm>SHA-256</messageDigestAlgorithm>

       <messageDigest>baede90f382a3ef0ad5a9fe6c42b1dfc0510fca667f5a424c0e478397f6024e3</messageDi
gest>
       <messageDigestOriginator>hashlib</messageDigestOriginator>
       </fixity>
       <size>7603618</size>
       <format>
       <formatRegistry>
       <formatRegistryName>PRONOM</formatRegistryName>
       <formatRegistryKey>fmt/18</formatRegistryKey>
       <formatRegistryRole>identification</formatRegistryRole>
       </formatRegistry>
       </format>
       </objectCharacteristics>
       <relationship>
       <relationshipType>derivation</relationshipType>
       <relationshipSubType>has source</relationshipSubType>
       <relatedObjectIdentification>
       <relatedObjectIdentifierType>filepath</relatedObjectIdentifierType>

       <relatedObjectIdentifierValue>file://./../../submission/representations/docs/data/Charlema
gne.pdf</relatedObjectIdentifierValue>
       <relatedObjectSequence>0</relatedObjectSequence>
       </relatedObjectIdentification>
       <relatedEventIdentification>
       <relatedEventIdentifierType>local</relatedEventIdentifierType>
       <relatedEventIdentifierValue>IDb4b9fa6a-a147-49f6-9901-
18e104221cb1</relatedEventIdentifierValue>
       <relatedEventSequence>1</relatedEventSequence>
       </relatedEventIdentification>
       </relationship>
  </object>
  <event>
       <eventIdentifier>
       <eventIdentifierType>local</eventIdentifierType>
       <eventIdentifierValue>IDdd7fe35b-3722-449f-aa22-019549b43b40</eventIdentifierValue>
       </eventIdentifier>
       <eventType>migration</eventType>
       <eventDateTime>2016-01-12T12:49:20</eventDateTime>
       <eventOutcomeInformation>
       <eventOutcome>success</eventOutcome>
       </eventOutcomeInformation>
       <linkingAgentIdentifier>
       <linkingAgentIdentifierType>software</linkingAgentIdentifierType>
       <linkingAgentIdentifierValue>should probably come from
migrations.xml</linkingAgentIdentifierValue>
       </linkingAgentIdentifier>
       <linkingObjectIdentifier>
       <linkingObjectIdentifierType>filepath</linkingObjectIdentifierType>
       <linkingObjectIdentifierValue>file://./data/Suleiman the
Magnificent.pdf</linkingObjectIdentifierValue>
       </linkingObjectIdentifier>
  </event>
  <event>
       <eventIdentifier>
       <eventIdentifierType>local</eventIdentifierType>
       <eventIdentifierValue>IDb4b9fa6a-a147-49f6-9901-18e104221cb1</eventIdentifierValue>
       </eventIdentifier>
       <eventType>migration</eventType>
       <eventDateTime>2016-01-12T12:49:20</eventDateTime>
       <eventOutcomeInformation>
       <eventOutcome>success</eventOutcome>
       </eventOutcomeInformation>
       <linkingAgentIdentifier>
       <linkingAgentIdentifierType>software</linkingAgentIdentifierType>
       <linkingAgentIdentifierValue>should probably come from
migrations.xml</linkingAgentIdentifierValue>
       </linkingAgentIdentifier>
       <linkingObjectIdentifier>
       <linkingObjectIdentifierType>filepath</linkingObjectIdentifierType>
       <linkingObjectIdentifierValue>file://./data/Charlemagne.pdf</linkingObjectIdentifierValue>
       </linkingObjectIdentifier>
  </event>
  <agent>
       <agentIdentifier>
       <agentIdentifierType>LOCAL</agentIdentifierType>
       <agentIdentifierValue>earkweb</agentIdentifierValue>
       </agentIdentifier>
       <agentName>E-ARK AIP to DIP Converter</agentName>
       <agentType>Software</agentType>
  </agent>
</premis>
```