Welcome to this live webinar on The eArchiving validation framework: its rationale and how it is linked to validating against specifications

Start 10:00

15 February 2024

Audience notes for the Live Webinar

Your **cameras have been turned off** and **microphones muted**.

If you have any technical issues during the event, please use the chat function.

Please **use the Q&A for questions to speakers**. These will be addressed at the end of the event.

Please note that this webinar is recorded. No attendee personal information will be captured in these recordings.

Agenda

10:00 – 10:05

eArchiving Initiaitve welcome

Jaime Kaminski – eArchiving Initiative training activity lead

10:05 – 10:50

The eArchiving validation framework: its rationale and how it is linked to validating against specifications

Sven Schlarb – Austrian Institute of Technology (AIT)

10:50 – 11:00

Q&A

The eArchiving Validation Framework

Sven Schlarb, AIT Austrian Institute of Technology

eArchiving Initiative Webinar

Overview

- Motivation
- Similar EC activities
  - Looking at similar activities of the European Commission: Eurostat, eDelivery
- Framework overview
  - Specifications, Requirements, Test Corpus, Validators
- eArchiving Conformance
- Summary & Acknowledgements

Motivation

Why do we need specifications, and what is the benefit of validating conformance?

Interoperability

- **Interoperability** can be considered as the ability of systems to exchange data (information packages) and use it
- **Standards** to define **Formats** for exchanging data between archival systems and services
- Achieve standards through defining **Best Practices** in **Specifications** and apply **Validation** to enforce **Conformance**

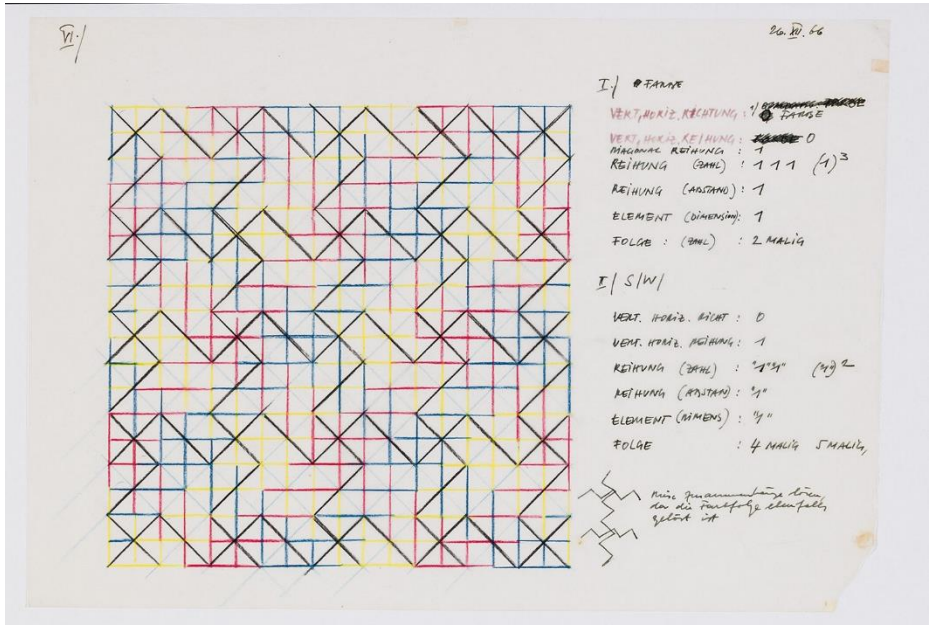
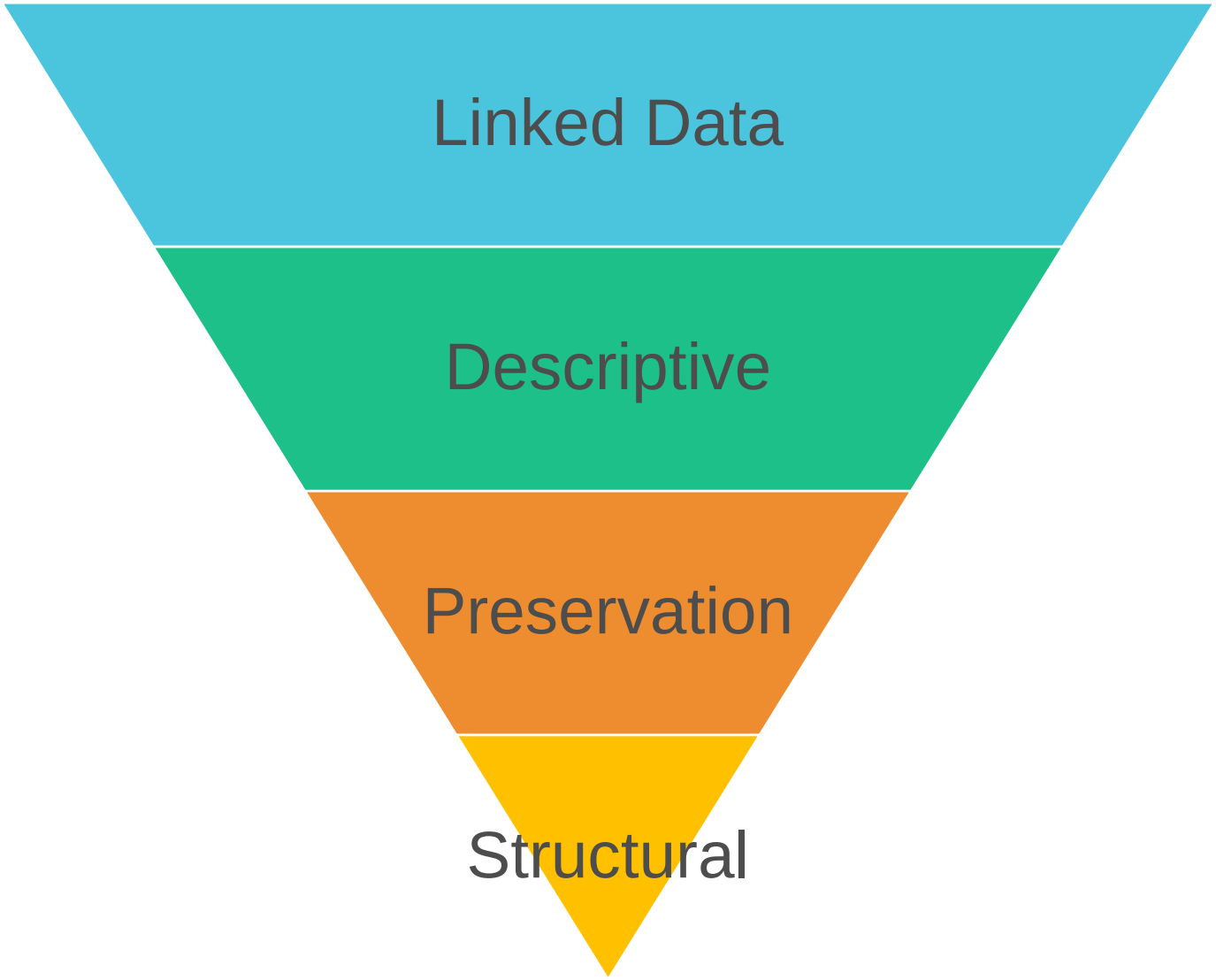


Image Attribution: Europeana, Skizze (Numerische Systeme) - State Collections of Lower Austria, Austria - In Copyright - Educational Use Permitted. https://www.europeana.eu/item/548/KS_23752_3

European Commission

Metadata Interoperability



RDF OWL SKOS RiC

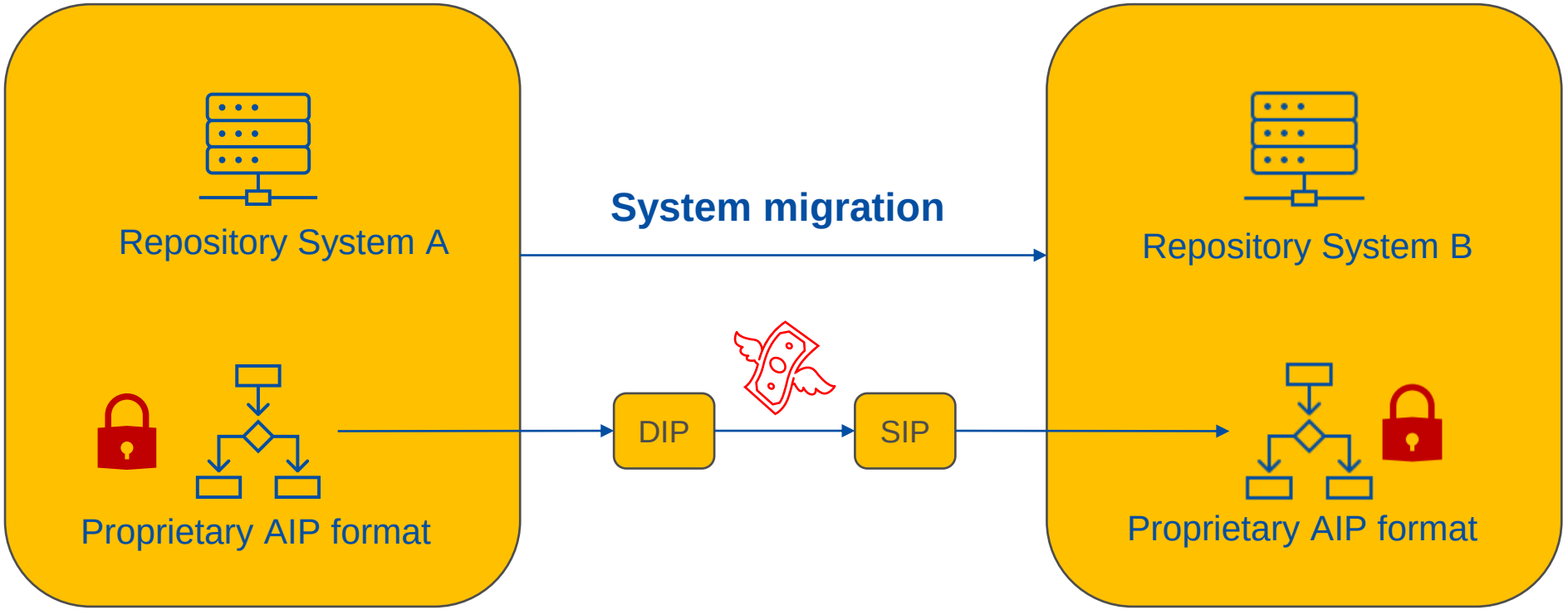
ead

PREMIS

METS

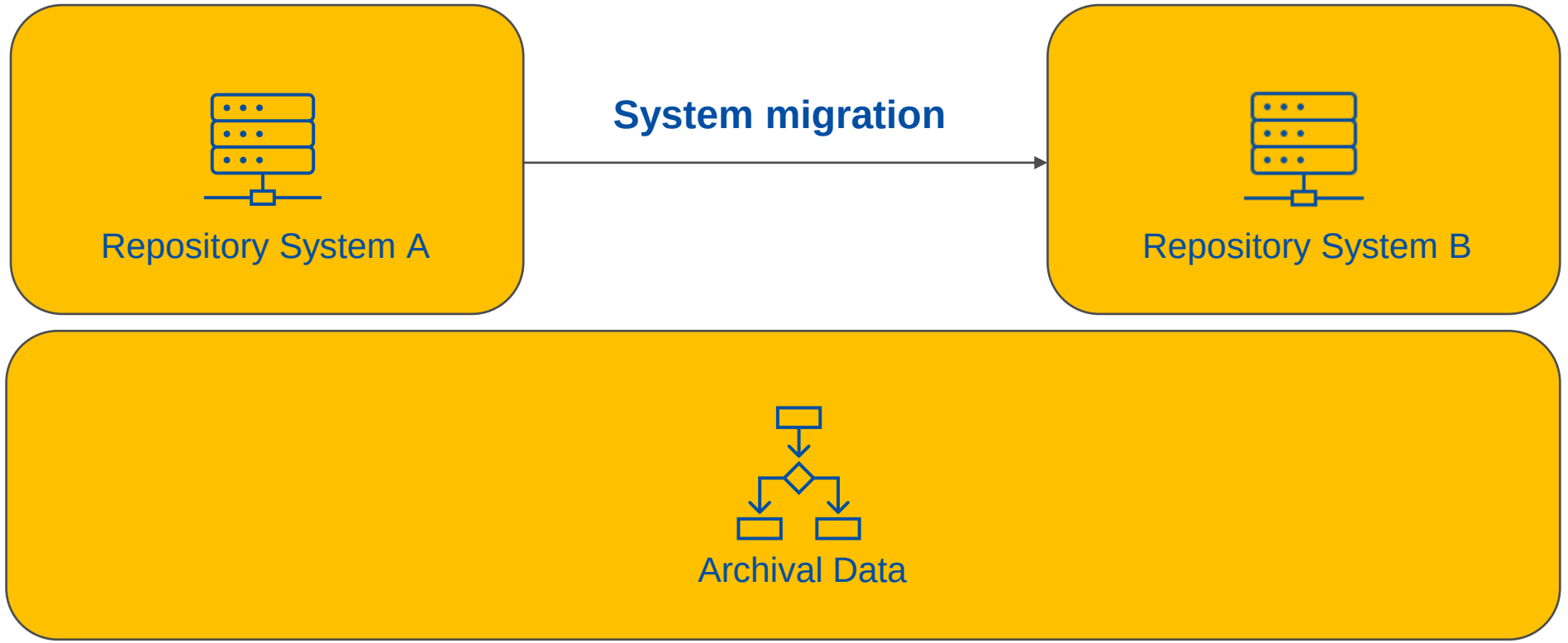
European Commission

Avoid Vendor Lock-in



- Data transformation processes are complex and tie up human resources
- System migration is therefore associated with high costs

Pan-European Information Package Format



Using instead a standardised format for the Archival Information Package (AIP) …

- makes system migration easier, and …
- enables shared functionality and synergies between solution providers if features are based on standards and conventions.

European Commission

Similar EC activities

Looking at similar activities of the European Commission:

- Eurostat
- eDelivery

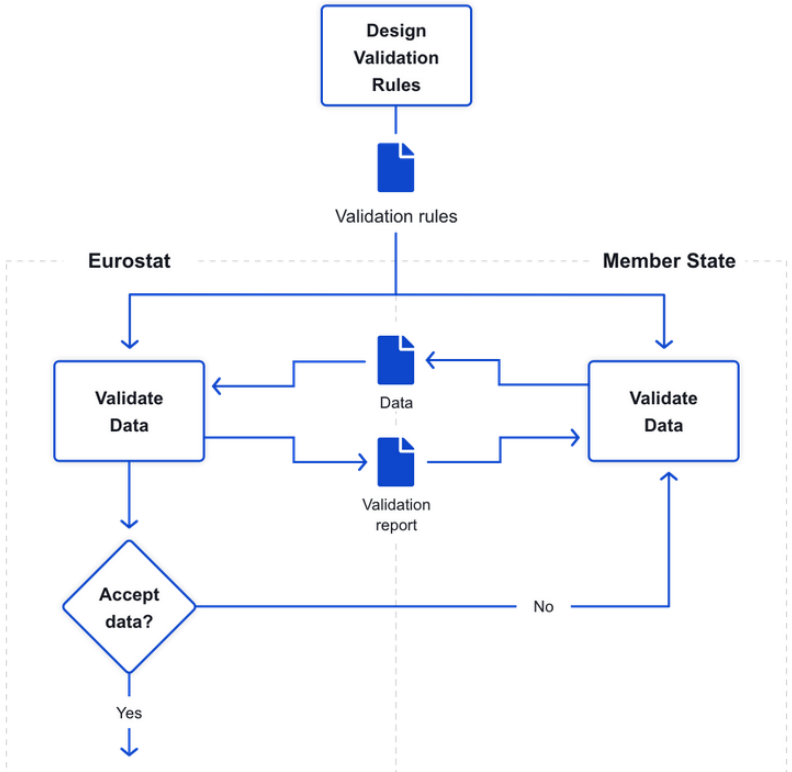
European Commission

Data Validation

- Defining **standards** for **validation** is a common approach to achieve **integration**
  - see Eurostat, for example
- Additionally:
  - providing **validation tools or services**
  - providing **support**.

eurostat

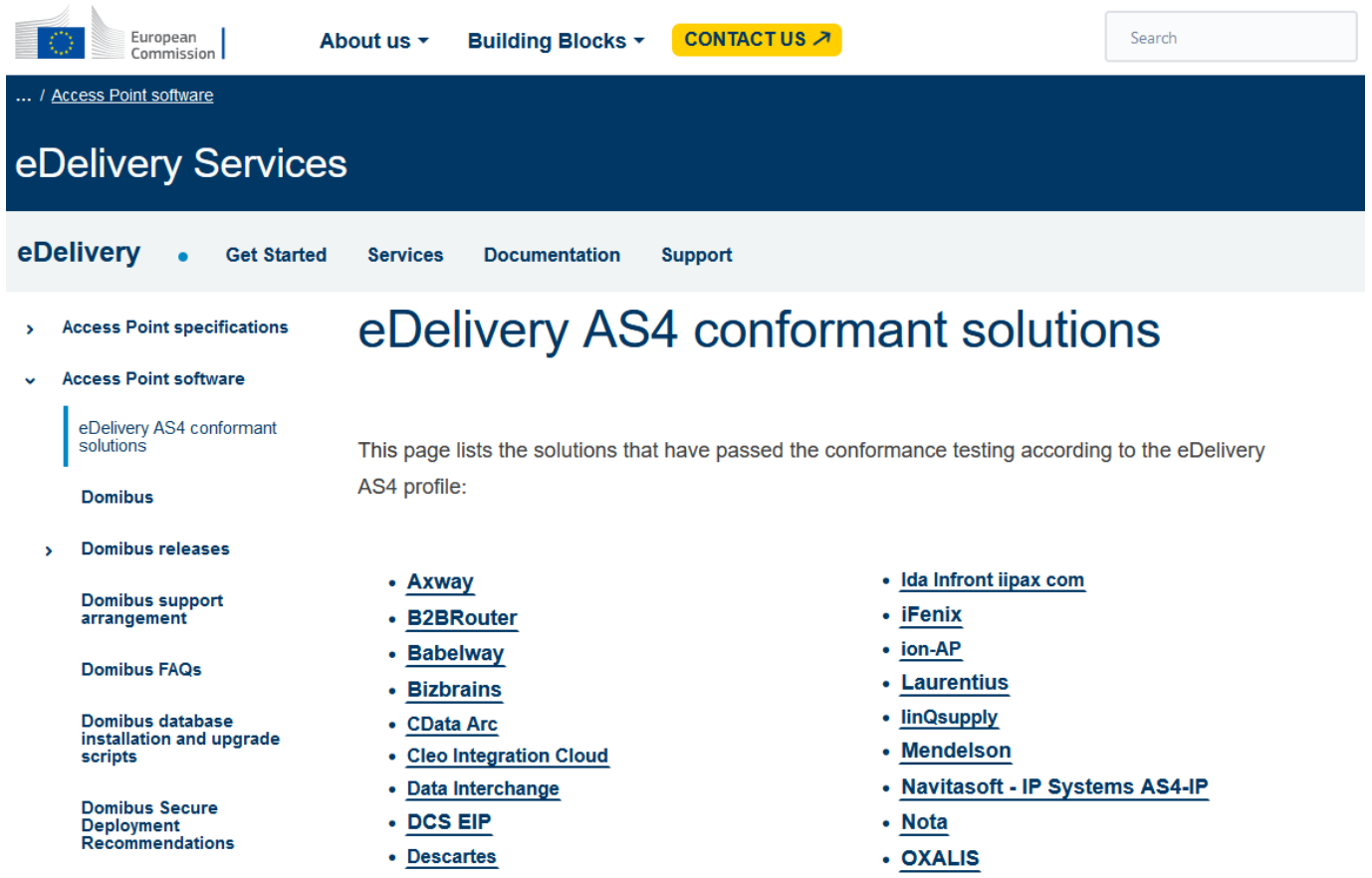
Definition: Data validation is an activity verifying whether or not a combination of values is a member of a set of acceptable combinations.



https://ec.europa.eu/eurostat/data/data-validation

European Commission

eDelivery – Conformant Solutions



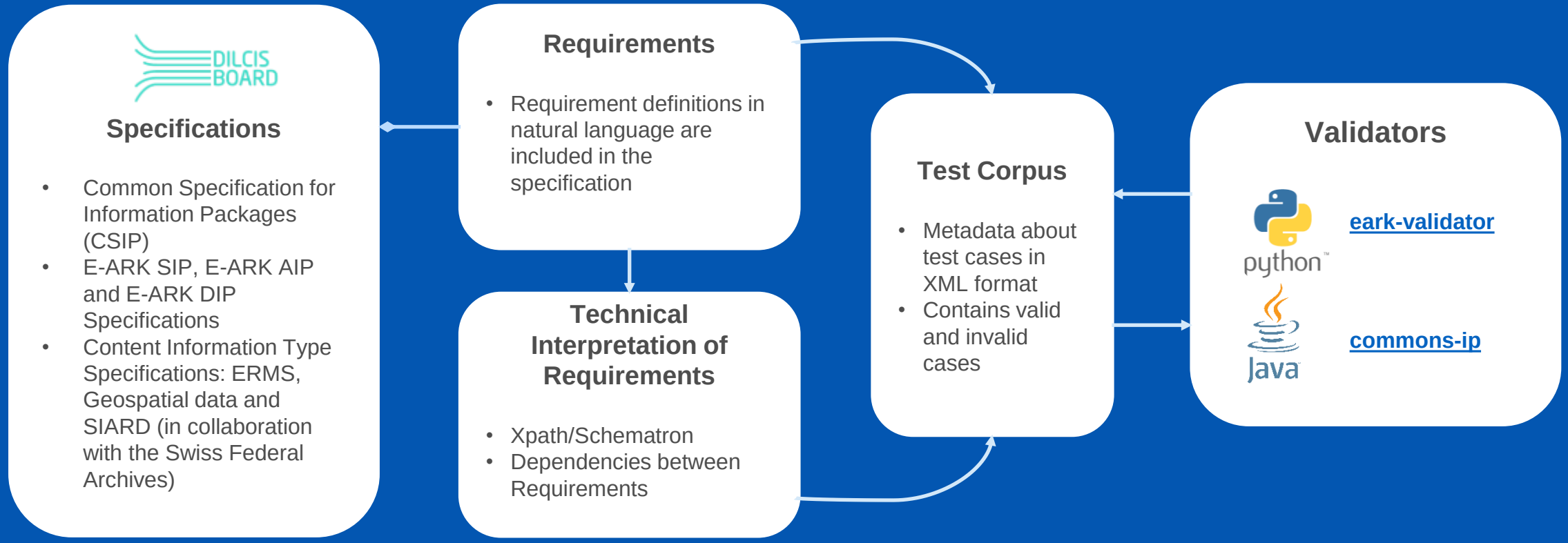
https://ec.europa.eu/digital-building-blocks/sites/display/DIGITAL/eDelivery+AS4+conformant+solutions

European Commission

Framework overview

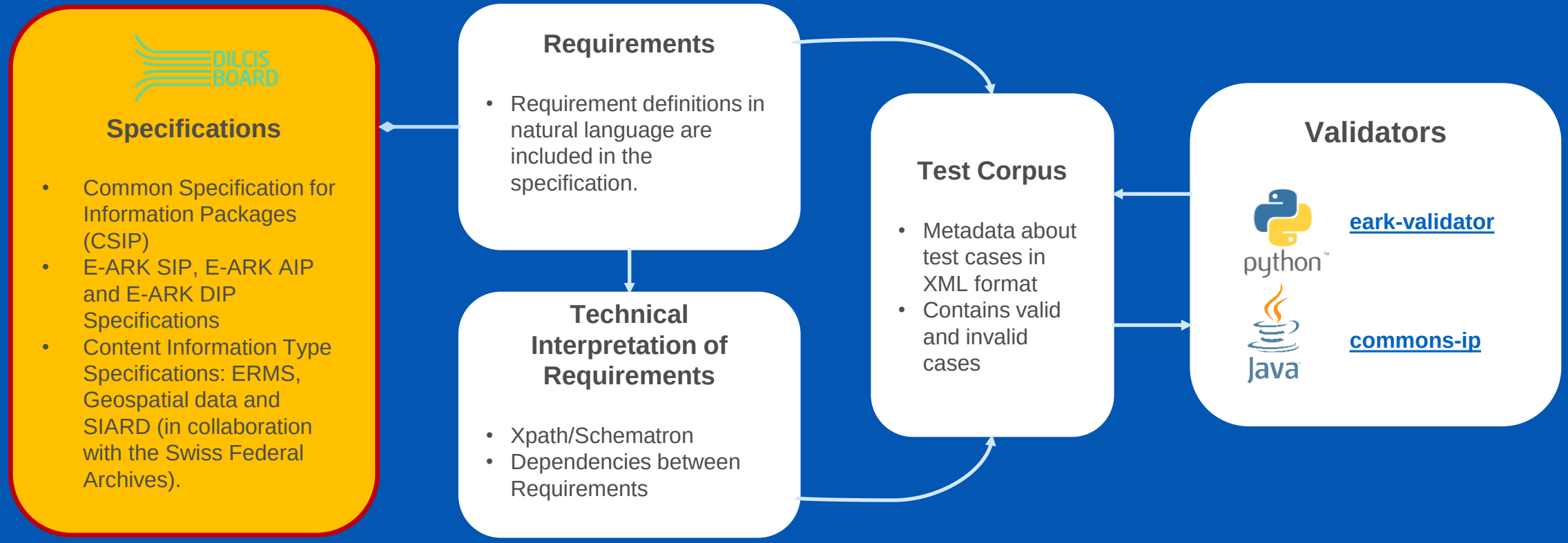
- Specifications including Requirements
- Metadata Profiles
- Test Corpus
- Validators

Validation Framework



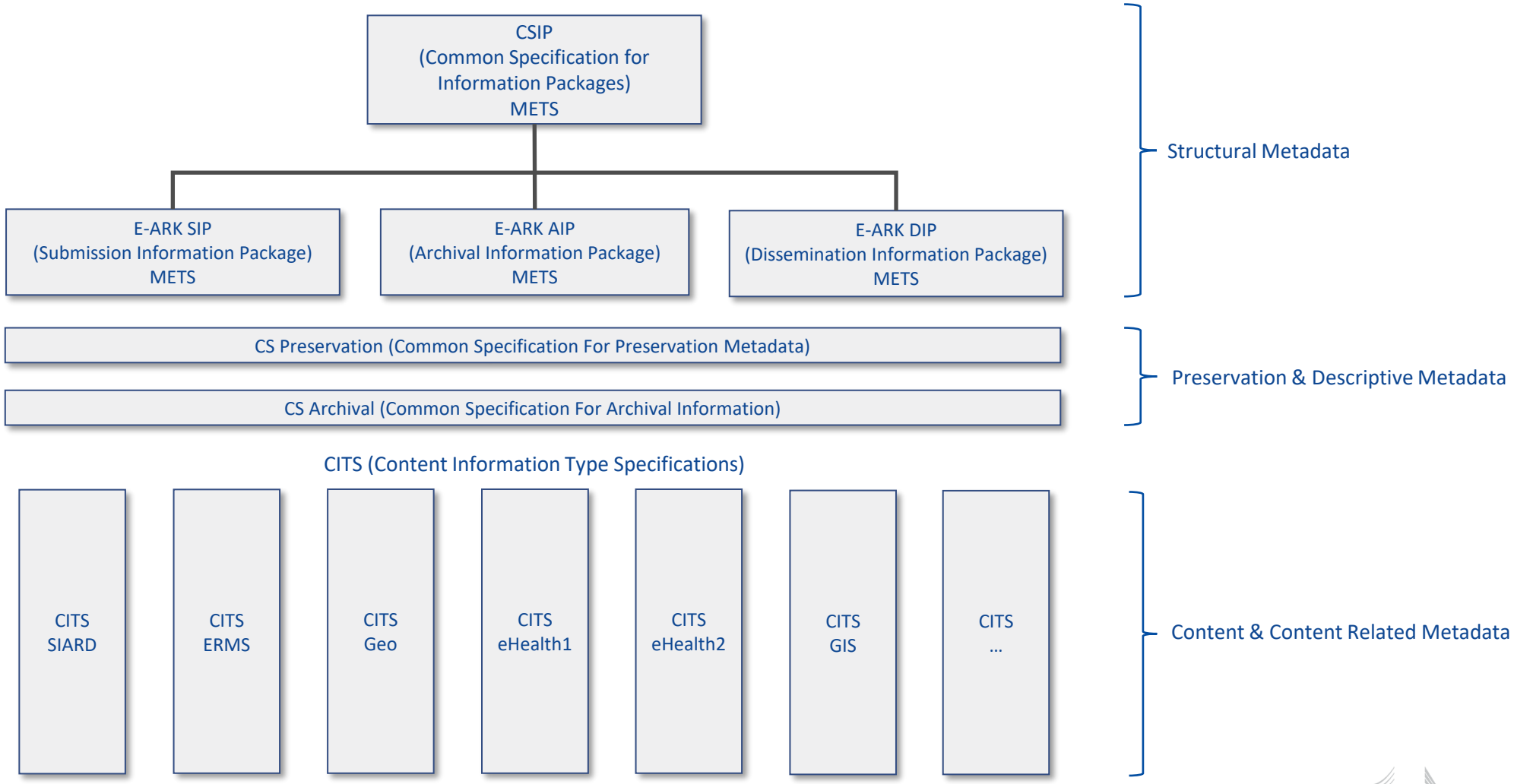
European Commission

Validation Framework



European Commission

Specifications

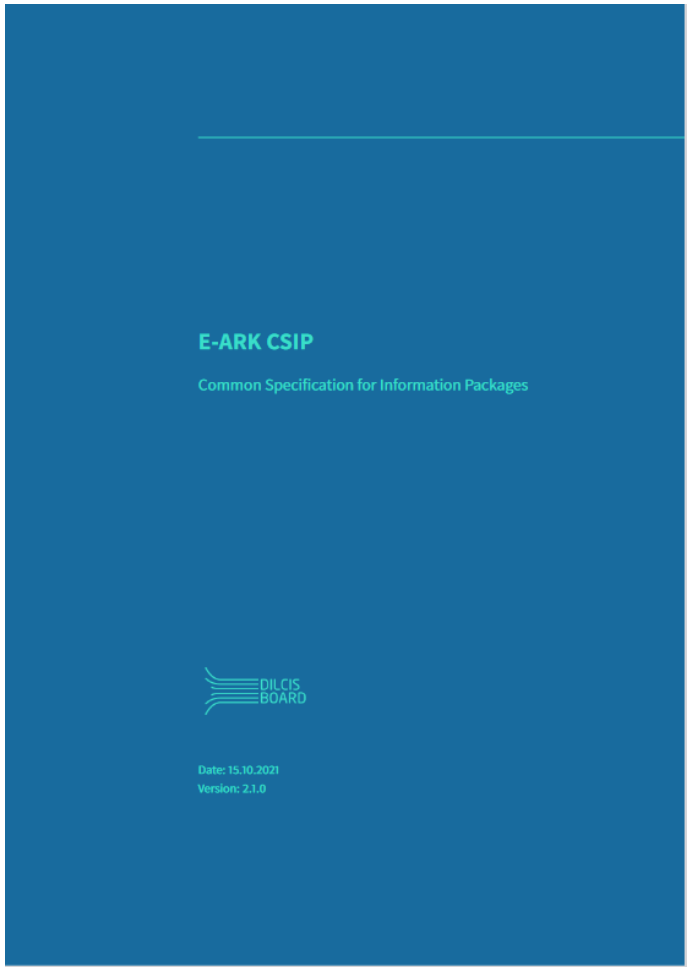


DILCIS BOARD

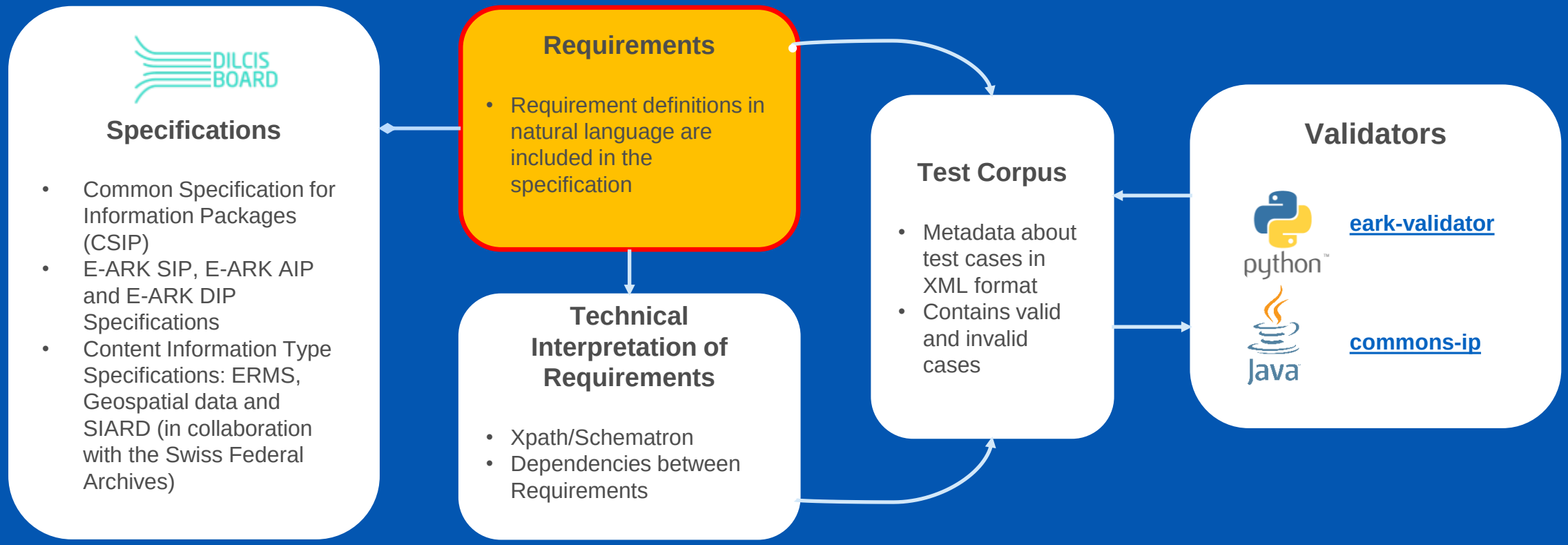
European Commission

Specification document, e.g. Common Specification for Information Packages

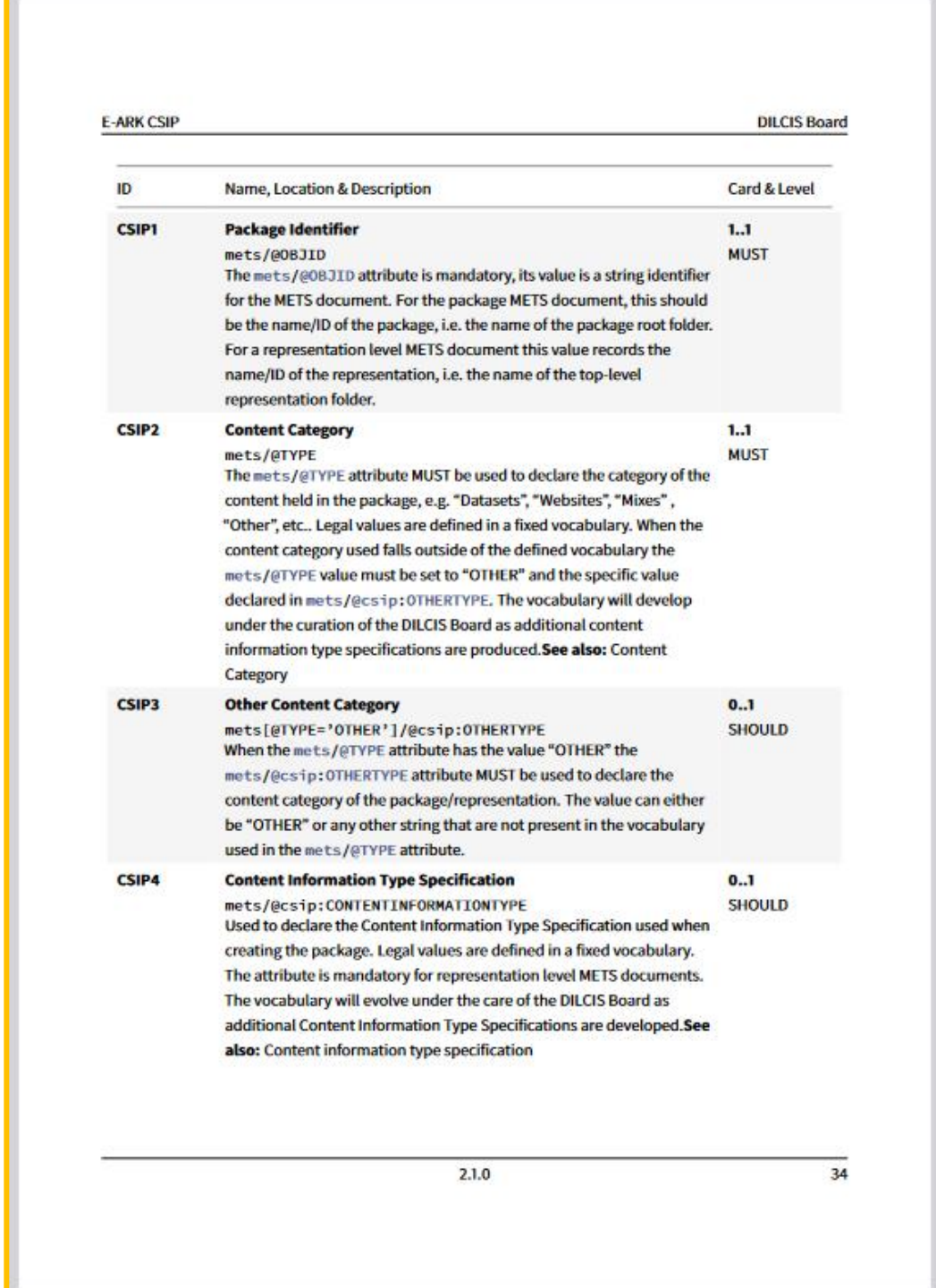
- High-Level Principles
  - Provide overarching guidance and direction for the technical specification details
  - Alignment with the goal to achieve interoperability
- Requirements
  - Requirements are related to a concrete implementation
    - Define a general folder structure for the information package
    - XML-based implementation of the requirements using standards which are widely used in international digital preservation



Validation Framework



European Commission



E-ARK CSIP DILCIS Board

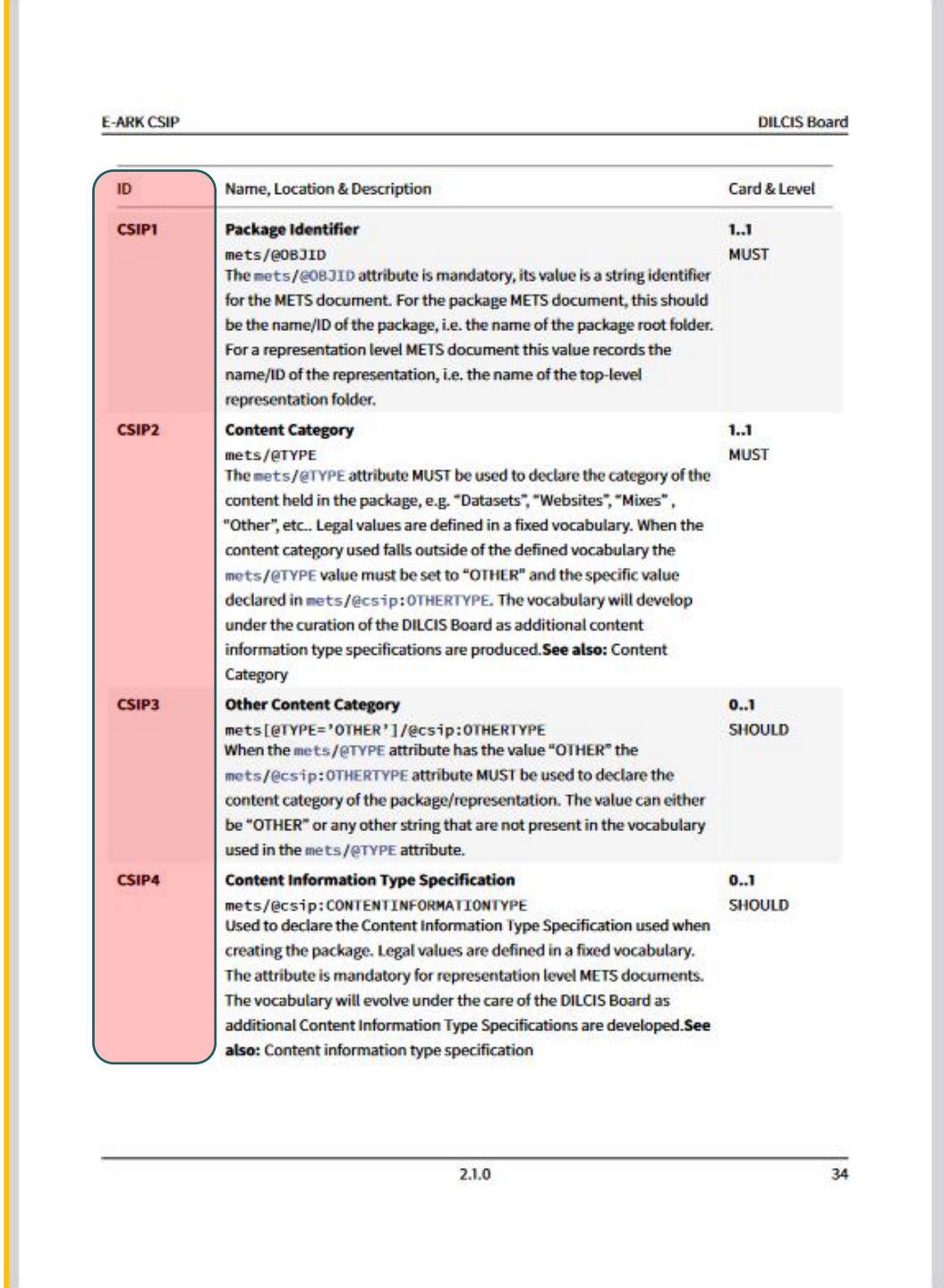
ID	Name, Location & Description	Card & Level
CSIP1	**Package Identifier** `mets/@OBJID` The `mets/@OBJID` attribute is mandatory, its value is a string identifier for the METS document. For the package METS document, this should be the name/ID of the package, i.e. the name of the package root folder. For a representation level METS document this value records the name/ID of the representation, i.e. the name of the top-level representation folder.	**1..1** MUST
CSIP2	**Content Category** `mets/@TYPE` The `mets/@TYPE` attribute MUST be used to declare the category of the content held in the package, e.g. "Datasets", "Websites", "Mixes" , "Other", etc.. Legal values are defined in a fixed vocabulary. When the content category used falls outside of the defined vocabulary the `mets/@TYPE` value must be set to "OTHER" and the specific value declared in `mets/@csip:OTHERTYPE`. The vocabulary will develop under the curation of the DILCIS Board as additional content information type specifications are produced.**See also:** Content Category	**1..1** MUST
CSIP3	**Other Content Category** `mets[@TYPE='OTHER']/@csip:OTHERTYPE` When the `mets/@TYPE` attribute has the value "OTHER" the `mets/@csip:OTHERTYPE` attribute MUST be used to declare the content category of the package/representation. The value can either be "OTHER" or any other string that are not present in the vocabulary used in the `mets/@TYPE` attribute.	**0..1** SHOULD
CSIP4	**Content Information Type Specification** `mets/@csip:CONTENTINFORMATIONTYPE` Used to declare the Content Information Type Specification used when creating the package. Legal values are defined in a fixed vocabulary. The attribute is mandatory for representation level METS documents. The vocabulary will evolve under the care of the DILCIS Board as additional Content Information Type Specifications are developed.**See also:** Content information type specification	**0..1** SHOULD

2.1.0 34

Requirements

- Requirements are part of the specification document
- All requirements have
  - an ID
  - a Name, Location & Description
  - A Cardinality and RFC2119 level (MUST, SHOULD, MAY)

European Commission



E-ARK CSIP DILCIS Board

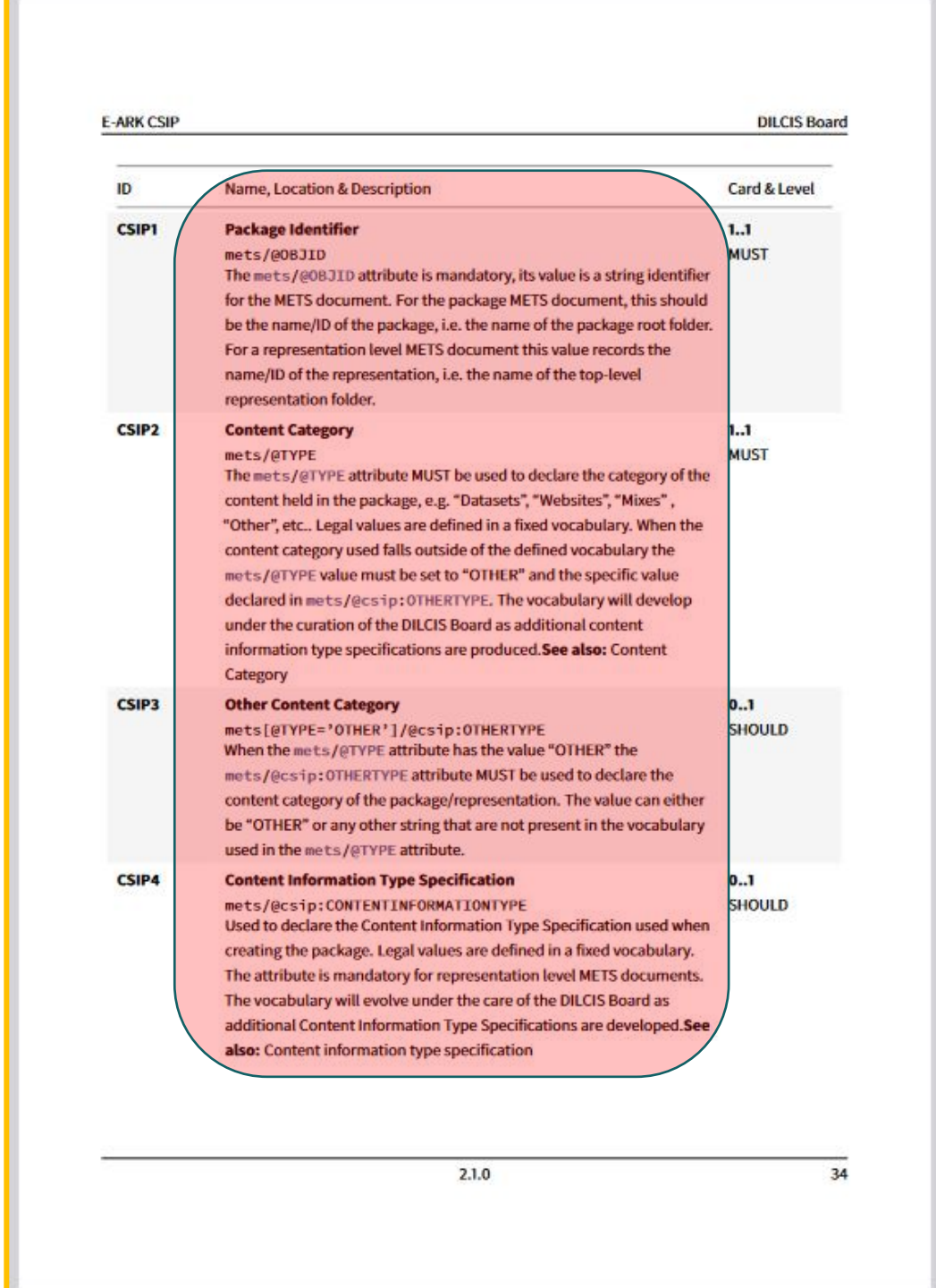
ID	Name, Location & Description	Card & Level
CSIP1	**Package Identifier** `mets/@OBJID` The `mets/@OBJID` attribute is mandatory, its value is a string identifier for the METS document. For the package METS document, this should be the name/ID of the package, i.e. the name of the package root folder. For a representation level METS document this value records the name/ID of the representation, i.e. the name of the top-level representation folder.	**1..1** MUST
CSIP2	**Content Category** `mets/@TYPE` The `mets/@TYPE` attribute MUST be used to declare the category of the content held in the package, e.g. "Datasets", "Websites", "Mixes" , "Other", etc.. Legal values are defined in a fixed vocabulary. When the content category used falls outside of the defined vocabulary the `mets/@TYPE` value must be set to "OTHER" and the specific value declared in `mets/@csip:OTHERTYPE`. The vocabulary will develop under the curation of the DILCIS Board as additional content information type specifications are produced.**See also:** Content Category	**1..1** MUST
CSIP3	**Other Content Category** `mets[@TYPE='OTHER']/@csip:OTHERTYPE` When the `mets/@TYPE` attribute has the value "OTHER" the `mets/@csip:OTHERTYPE` attribute MUST be used to declare the content category of the package/representation. The value can either be "OTHER" or any other string that are not present in the vocabulary used in the `mets/@TYPE` attribute.	**0..1** SHOULD
CSIP4	**Content Information Type Specification** `mets/@csip:CONTENTINFORMATIONTYPE` Used to declare the Content Information Type Specification used when creating the package. Legal values are defined in a fixed vocabulary. The attribute is mandatory for representation level METS documents. The vocabulary will evolve under the care of the DILCIS Board as additional Content Information Type Specifications are developed.**See also:** Content information type specification	**0..1** SHOULD

2.1.0 34

Requirements

- Requirements are part of the specification document
- All requirements have
  - **an ID**
  - a Name, Location & Description
  - A Cardinality and RFC2119 level (MUST, SHOULD, MAY)

European Commission



E-ARK CSIP DILCIS Board

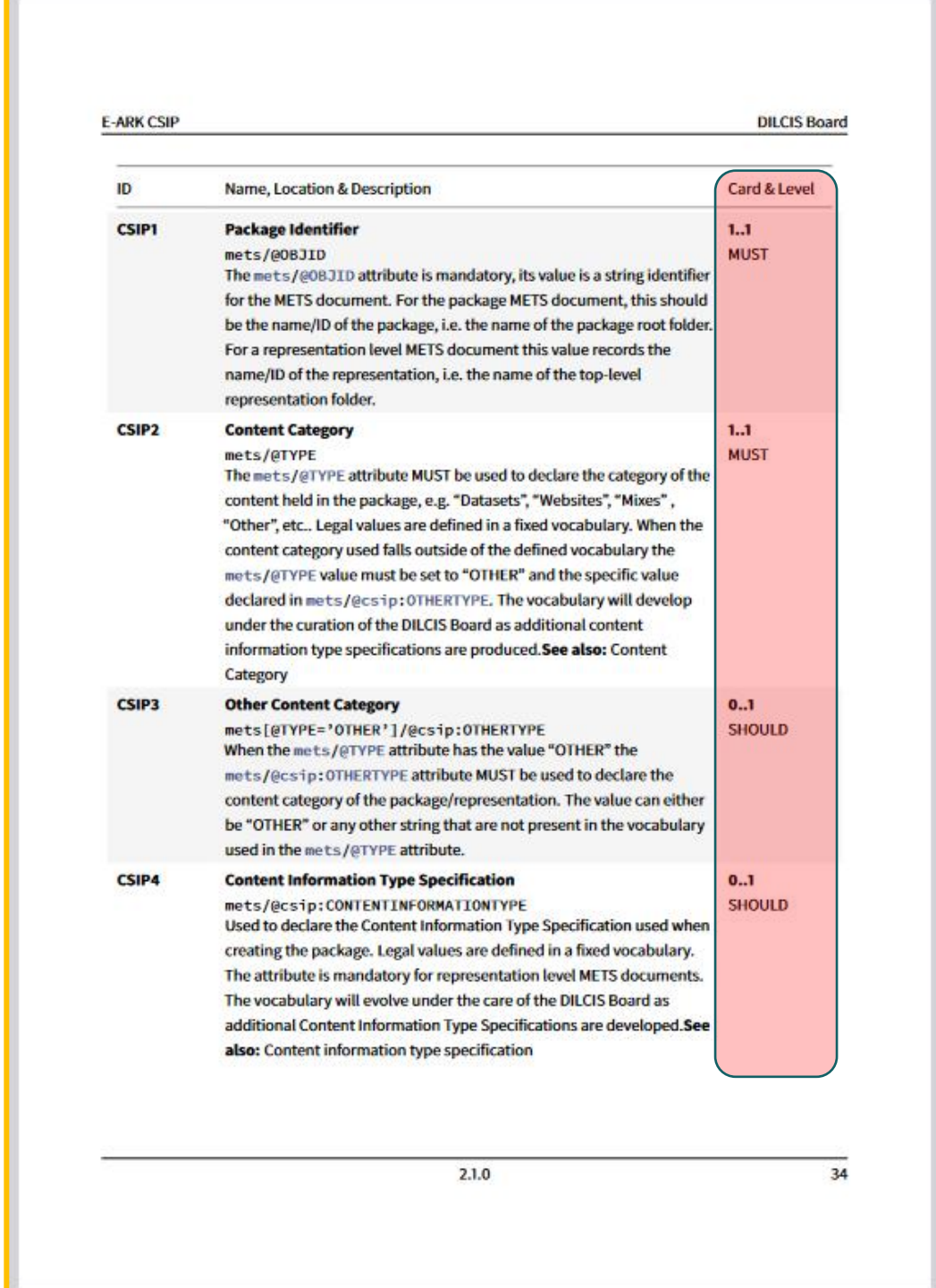
ID	Name, Location & Description	Card & Level
CSIP1	**Package Identifier** `mets/@OBJID` The `mets/@OBJID` attribute is mandatory, its value is a string identifier for the METS document. For the package METS document, this should be the name/ID of the package, i.e. the name of the package root folder. For a representation level METS document this value records the name/ID of the representation, i.e. the name of the top-level representation folder.	1..1 MUST
CSIP2	**Content Category** `mets/@TYPE` The `mets/@TYPE` attribute MUST be used to declare the category of the content held in the package, e.g. "Datasets", "Websites", "Mixes" , "Other", etc.. Legal values are defined in a fixed vocabulary. When the content category used falls outside of the defined vocabulary the `mets/@TYPE` value must be set to "OTHER" and the specific value declared in `mets/@csip:OTHERTYPE`. The vocabulary will develop under the curation of the DILCIS Board as additional content information type specifications are produced.**See also:** Content Category	1..1 MUST
CSIP3	**Other Content Category** `mets[@TYPE='OTHER']/@csip:OTHERTYPE` When the `mets/@TYPE` attribute has the value "OTHER" the `mets/@csip:OTHERTYPE` attribute MUST be used to declare the content category of the package/representation. The value can either be "OTHER" or any other string that are not present in the vocabulary used in the `mets/@TYPE` attribute.	0..1 SHOULD
CSIP4	**Content Information Type Specification** `mets/@csip:CONTENTINFORMATIONTYPE` Used to declare the Content Information Type Specification used when creating the package. Legal values are defined in a fixed vocabulary. The attribute is mandatory for representation level METS documents. The vocabulary will evolve under the care of the DILCIS Board as additional Content Information Type Specifications are developed.**See also:** Content information type specification	0..1 SHOULD

2.1.0 34

Requirements

- Requirements are part of the specification document
- All requirements have
  - an ID
  - **a Name, Location & Description**
  - A Cardinality and RFC2119 level (MUST, SHOULD, MAY)

European Commission



E-ARK CSIP DILCIS Board

ID	Name, Location & Description	Card & Level
CSIP1	**Package Identifier** `mets/@OBJID` The `mets/@OBJID` attribute is mandatory, its value is a string identifier for the METS document. For the package METS document, this should be the name/ID of the package, i.e. the name of the package root folder. For a representation level METS document this value records the name/ID of the representation, i.e. the name of the top-level representation folder.	**1..1** MUST
CSIP2	**Content Category** `mets/@TYPE` The `mets/@TYPE` attribute MUST be used to declare the category of the content held in the package, e.g. "Datasets", "Websites", "Mixes" , "Other", etc.. Legal values are defined in a fixed vocabulary. When the content category used falls outside of the defined vocabulary the `mets/@TYPE` value must be set to "OTHER" and the specific value declared in `mets/@csip:OTHERTYPE`. The vocabulary will develop under the curation of the DILCIS Board as additional content information type specifications are produced.**See also:** Content Category	**1..1** MUST
CSIP3	**Other Content Category** `mets[@TYPE='OTHER']/@csip:OTHERTYPE` When the `mets/@TYPE` attribute has the value "OTHER" the `mets/@csip:OTHERTYPE` attribute MUST be used to declare the content category of the package/representation. The value can either be "OTHER" or any other string that are not present in the vocabulary used in the `mets/@TYPE` attribute.	**0..1** SHOULD
CSIP4	**Content Information Type Specification** `mets/@csip:CONTENTINFORMATIONTYPE` Used to declare the Content Information Type Specification used when creating the package. Legal values are defined in a fixed vocabulary. The attribute is mandatory for representation level METS documents. The vocabulary will evolve under the care of the DILCIS Board as additional Content Information Type Specifications are developed.**See also:** Content information type specification	**0..1** SHOULD

2.1.0 34

Requirements

- Requirements are part of the specification document
- All requirements have
  - an ID
  - a Name, Location & Description
  - **A Cardinality and RFC2119 level (MUST, SHOULD, MAY)**

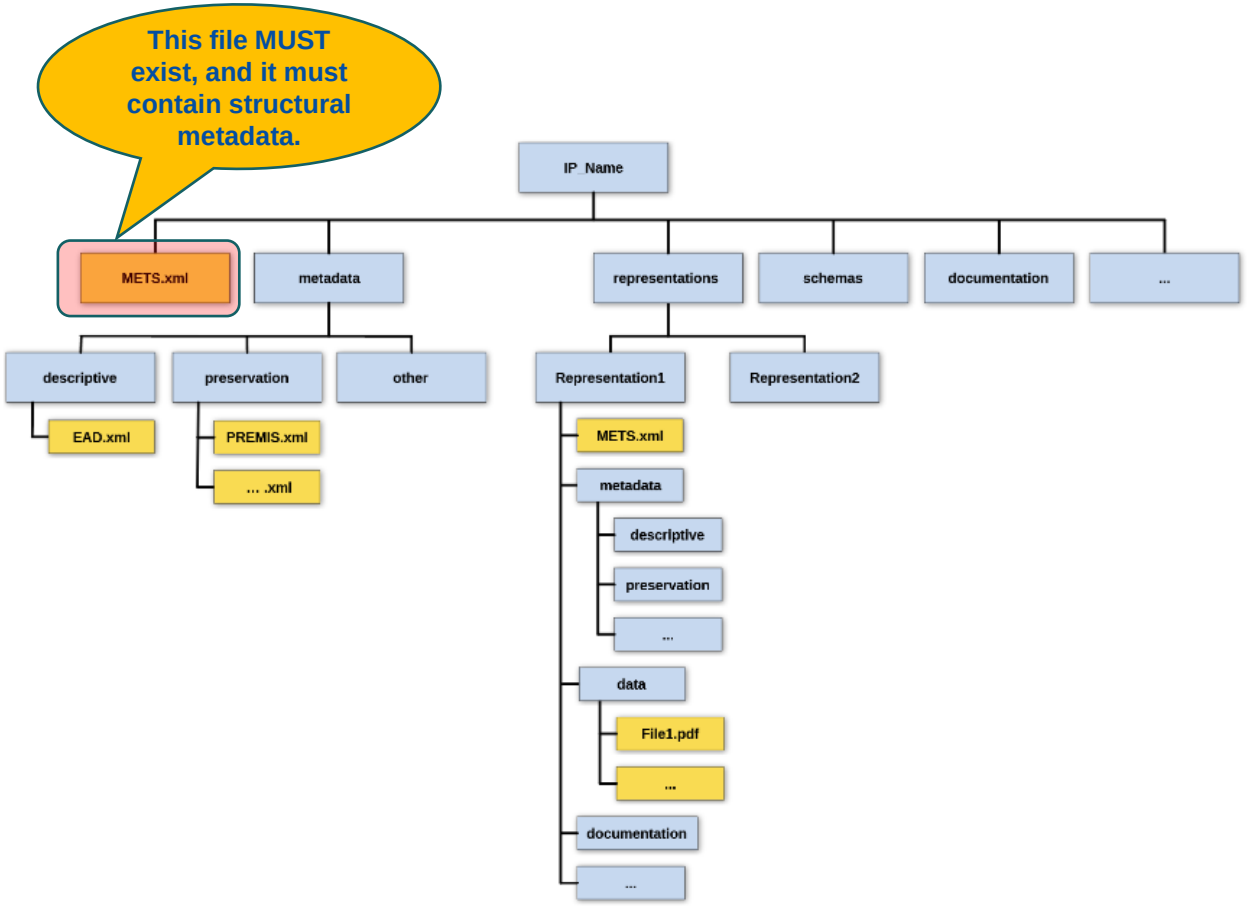
European Commission

CSIP Requirement Example – CSIPSTR4

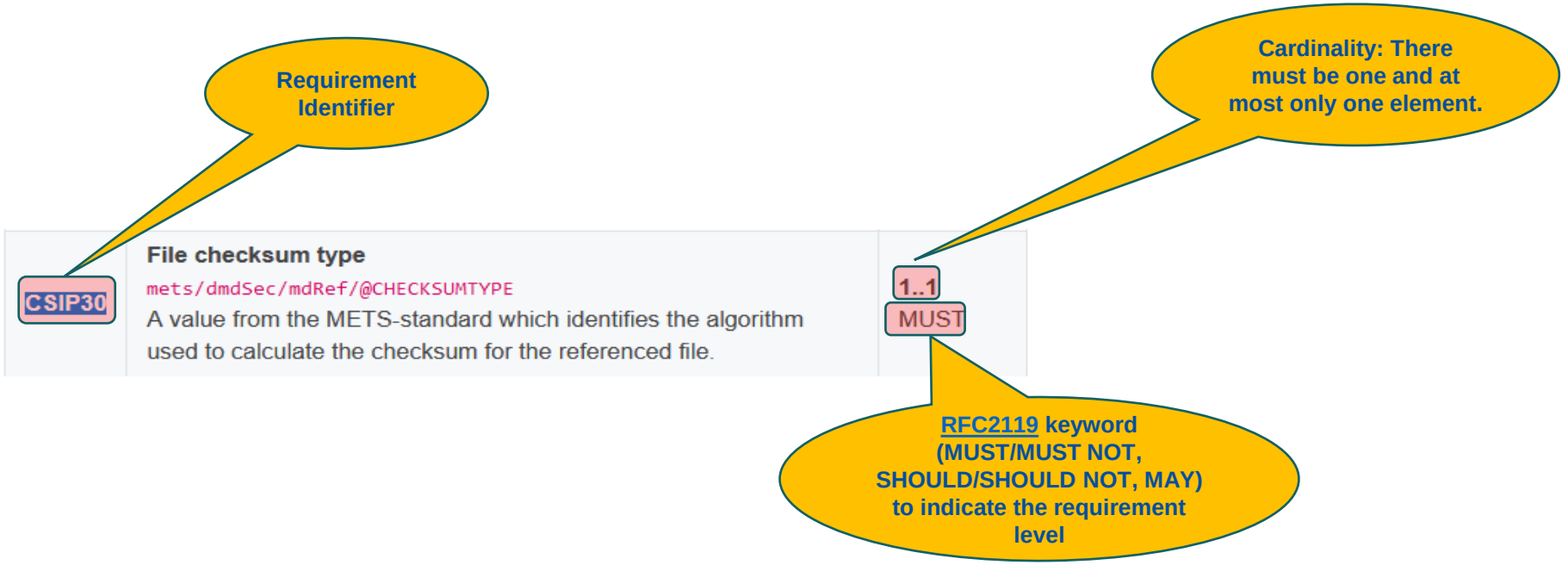
Requirement Identifier

RFC2119 keyword (MUST/MUST NOT, SHOULD/SHOULD NOT, MAY) to indicate the requirement level

CSIPSTR4: The Information Package root folder **MUST** include a file named `METS.xml`. This file **MUST** contain metadata that identifies the package, provides a high-level package description, and describes its structure, including pointers to constituent representations.



CSIP Requirement Example – CSIP30




```
<Example ID="dmdSecExample1" LABEL="METS example of referencing the descriptive metadata which is described with an EAD document">
        <mets:dmdSec ID="dmd-ead-file" CREATED="2018-04-24T14:37:49.609+01:00">
            <mets:mdRef LOCTYPE="URL" MDTYPE="EAD" xlink:type="simple"  xlink:href="metadata/descriptive/ead2002.xml"
              MIMETYPE="application/xml" SIZE="903" CREATED="2018-04-24T14:37:49.609+01:00"
              CHECKSUM="F24263BF09994749F335E1664DCE0086DB6DCA323FDB6996938BCD28EA9E8153" CHECKSUMTYPE="SHA-256"/>
        </mets:dmdSec>
</Example>
```

The CHECKSUMTYPE element must exist and there must be a value for the algorithm type.

European Commission

Requirements list

Structure

CSIPSTR1
CSIPSTR2
CSIPSTR3
CSIPSTR4
CSIPSTR5
CSIPSTR6
CSIPSTR7
CSIPSTR8
CSIPSTR9
CSIPSTR10
CSIPSTR11
CSIPSTR12
CSIPSTR13

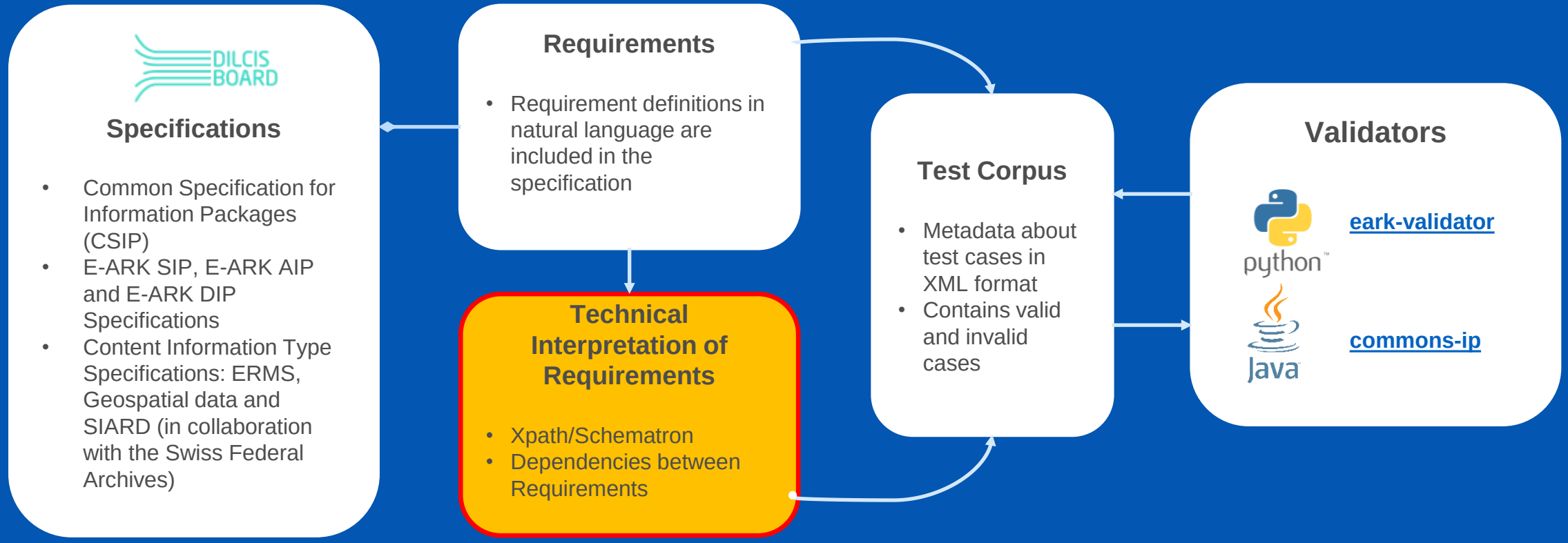
Metadata

CSIP1	CSIP21	CSIP41	CSIP61	CSIP81	CSIP101
CSIP2	CSIP22	CSIP42	CSIP62	CSIP82	CSIP102
CSIP3	CSIP23	CSIP43	CSIP63	CSIP83	CSIP103
CSIP4	CSIP24	CSIP44	CSIP64	CSIP84	CSIP104
CSIP5	CSIP25	CSIP45	CSIP65	CSIP85	CSIP105
CSIP6	CSIP26	CSIP46	CSIP66		CSIP106
CSIP7	CSIP27	CSIP47	CSIP67		CSIP107
CSIP8	CSIP28	CSIP48	CSIP68	CSIP88	CSIP108
CSIP9	CSIP29	CSIP49	CSIP69	CSIP89	CSIP109
CSIP10	CSIP30	CSIP50	CSIP70	CSIP90	CSIP110
CSIP11	CSIP31	CSIP51	CSIP71	CSIP91	CSIP111
CSIP12	CSIP32	CSIP52	CSIP72	CSIP92	CSIP112
CSIP13	CSIP33	CSIP53	CSIP73	CSIP93	CSIP113
CSIP14	CSIP34	CSIP54	CSIP74	CSIP94	CSIP114
CSIP15	CSIP35	CSIP55	CSIP75	CSIP95	
CSIP16	CSIP36	CSIP56	CSIP76	CSIP96	CSIP116
CSIP17	CSIP37	CSIP57	CSIP77	CSIP97	CSIP117
CSIP18	CSIP38	CSIP58	CSIP78	CSIP98	CSIP118
CSIP19	CSIP39	CSIP59	CSIP79	CSIP99	CSIP119
CSIP20	CSIP40	CSIP60	CSIP80	CSIP100	

Section references

REF_METS_1
REF_METS_2

Validation Framework



European Commission

Requirement in METS Profile

```
<requirement ID="CSIP30" REQLEVEL="MUST" EXAMPLES="dmdSecExample1">
        <description>
            <head>File checksum type</head>
            <p xmlns="http://www.w3.org/1999/xhtml">
               The type of checksum following the value list present in the METS-standard which has been used for
               calculating the checksum for the referenced file.</p>
            <dl xmlns="http://www.w3.org/1999/xhtml">
        <dt>dependsOn</dt><dd>CSIP21</dd>
            </dl>
        </description>
        <tests>
            <test ID="TEST30-1" TESTLANGUAGE="XPath" TESTLANGUAGEVERSION="3.1">
        <testWrap>
            <testXML>/mets:mets/mets:dmdSec/mets:mdRef/@CHECKSUMTYPE</testXML>
        </testWrap>
            </test>
            <test ID="TEST30-2" TESTLANGUAGE="Schematron" TESTLANGUAGEVERSION="ISO" TESTLANGUAGEURI="http://purl.oclc.org/dsdl/schematron">
        <testWrap>
            <testXML>
                <iso:rule context="/mets:mets/mets:dmdSec/mets:mdref">
                    <iso:assert id="CSIP30" role="ERROR" test="@CHECKSUMTYPE">
                      MUST hold the algorithm type of checksum of the referenced file.</iso:assert>
                </iso:rule>
            </testXML>
        </testWrap>
            </test>
        </tests>
</requirement>
```

Requirement in METS Profile

```
<requirement ID="CSIP30" REQLEVEL="MUST" EXAMPLES="dmdSecExample1">
        <description>
            <head>File checksum type</head>
            <p xmlns="http://www.w3.org/1999/xhtml">
               The type of checksum following the value list present in the METS-standard which has been used for
               calculating the checksum for the referenced file.</p>
            <dl xmlns="http://www.w3.org/1999/xhtml">
        <dt>dependsOn</dt><dd>CSIP21</dd>
            </dl>
        </description>
        <tests>
            <test ID="TEST30-1" TESTLANGUAGE="XPath" TESTLANGUAGEVERSION="3.0">
        <testWrap>
            <testXML>/mets:mets/mets:dmdSec/mets:mdRef/@CHECKSUMTYPE</testXML>
        </testWrap>
            </test>
            <test ID="TEST30-2" TESTLANGUAGE="Schematron" TESTLANGUAGEVERSION="ISO" TESTLANGUAGEURI="http://purl.oclc.org/dsdl/schematron">
        <testWrap>
            <testXML>
                <iso:rule context="/mets:mets/mets:dmdSec/mets:mdref">
                    <iso:assert id="CSIP30" role="ERROR" test="@CHECKSUMTYPE">
                      MUST hold the algorithm type of checksum of the referenced file.</iso:assert>
                </iso:rule>
            </testXML>
        </testWrap>
            </test>
        </tests>
</requirement>
```

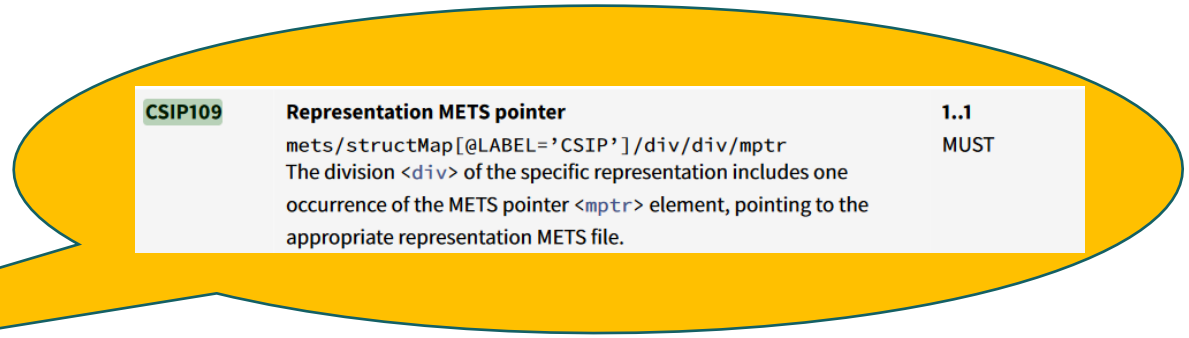
XPath expression

Requirement in METS Profile

```
<requirement ID="CSIP30" REQLEVEL="MUST" EXAMPLES="dmdSecExample1">
        <description>
            <head>File checksum type</head>
            <p xmlns="http://www.w3.org/1999/xhtml">
               The type of checksum following the value list present in the METS-standard which has been used for
               calculating the checksum for the referenced file.</p>
            <dl xmlns="http://www.w3.org/1999/xhtml">
        <dt>dependsOn</dt><dd>CSIP21</dd>
            </dl>
        </description>
        <tests>
            <test ID="TEST30-1" TESTLANGUAGE="XPath" TESTLANGUAGEVERSION="3.1">
        <testWrap>
            <testXML>/mets:mets/mets:dmdSec/mets:mdRef/@CHECKSUMTYPE</testXML>
        </testWrap>
            </test>
            <test ID="TEST30-2" TESTLANGUAGE="Schematron" TESTLANGUAGEVERSION="ISO" TESTLANGUAGEURI="http://purl.oclc.org/dsdl/schematron">
        <testWrap>
            <testXML>
                <iso:rule context="/mets:mets/mets:dmdSec/mets:mdref">
                    <iso:assert id="CSIP30" role="ERROR" test="@CHECKSUMTYPE">
                      MUST hold the algorithm type of checksum of the referenced file.</iso:assert>
                </iso:rule>
            </testXML>
        </testWrap>
            </test>
        </tests>
</requirement>
```

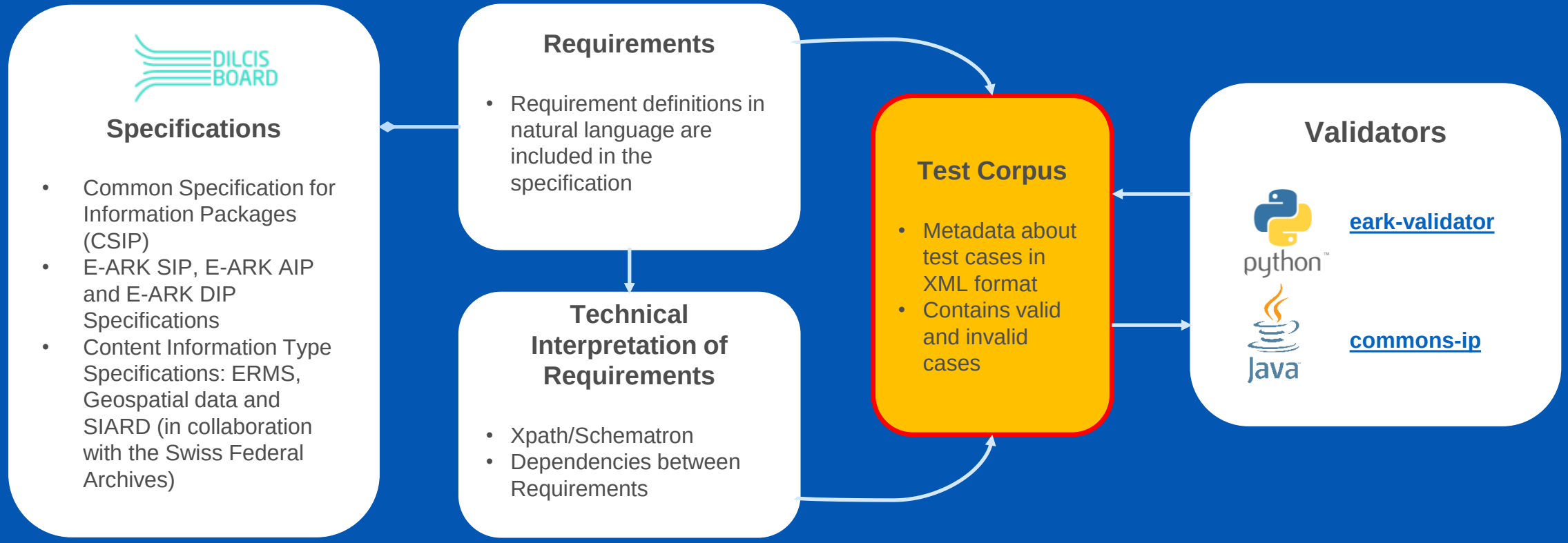
Schematron

Dependencies




```
<requirement ID="CSIP112" REQLEVEL="MUST" EXAMPLES="structMapExample2">
        <description>
            <head>Type of locator</head>
            <p xmlns="http://www.w3.org/1999/xhtml">
                        The locator type is always                with the value "URL" from the vocabulary in the attribute.</p>
            <dl xmlns="http://www.w3.org/1999/xhtml">
                <dt>dependsOn</dt><dd>CSIP109</dd>
                <dt>since</dt><dd>v2.1.0</dd>
            </dl>
        </description>
        <tests>
            <test ID="TEST112-1" TESTLANGUAGE="XPath" TESTLANGUAGEVERSION="3.1">
                <testWrap>
                    <testXML>/mets:mets/mets:structMap/mets:div/mets:div/mets:mptr[@LOCTYPE='URL']</testXML>
                </testWrap>
            </test>
            <test ID="TEST112-2" TESTLANGUAGE="Schematron" TESTLANGUAGEVERSION="ISO" TESTLANGUAGEURI="http://purl.oclc.org/dsdl/schematron">
                <testWrap>
                    <testXML>
                        <iso:rule context="/mets:mets/mets:structMap[@LABEL = 'CSIP']/mets:div/mets:div[@LABEL = 'Representations']/mptr">
                            <iso:assert id="CSIP112" role="ERROR" test="@LOCTYPE = 'URL'”">
                                The locator type is always used with the value “URL” from the vocabulary in the attribute.
                            </iso:assert>
                        </iso:rule>
                    </testXML>
                </testWrap>
            </test>
        </tests>
</requirement>
```

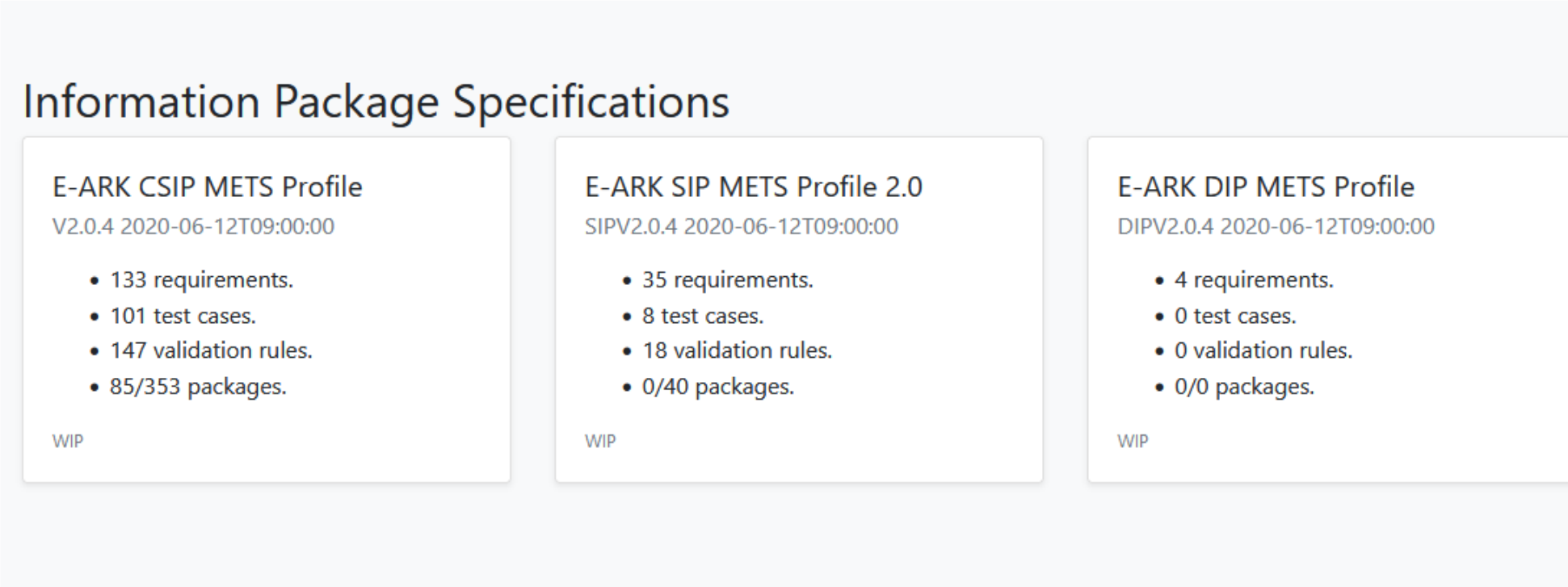
Validation Framework



European Commission

E-ARK Specification Test Coverage

Test coverage for the E-ARK Information Package Specifications.

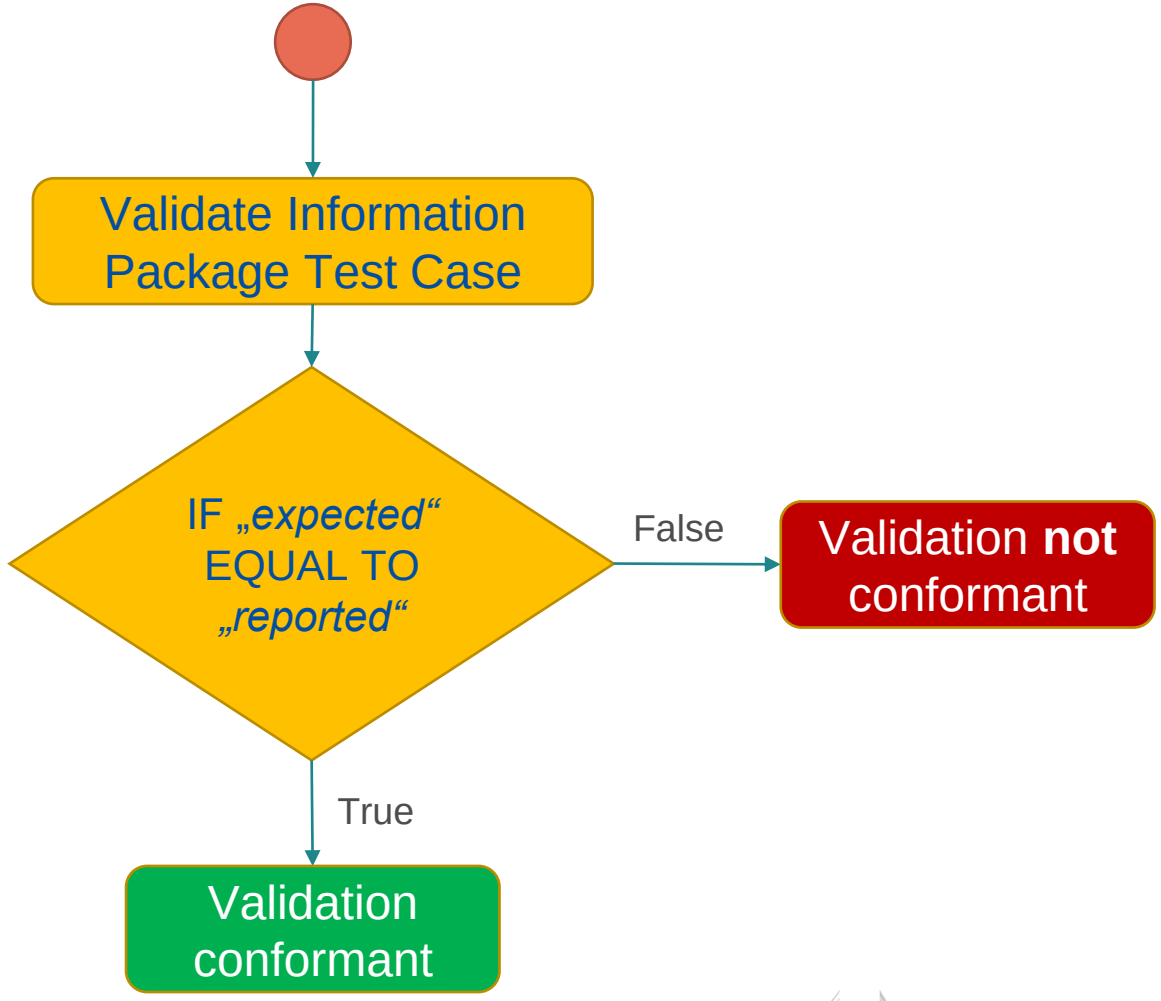


https://dilcisboard.github.io/eark-ip-test-corpus/

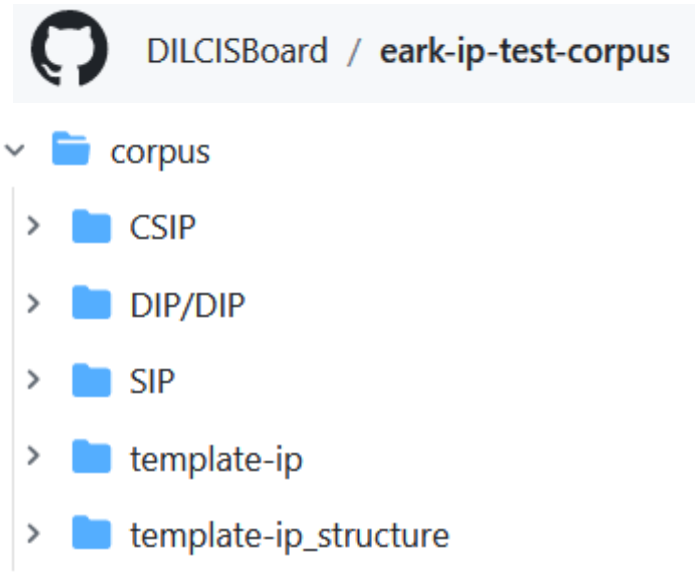
European Commission

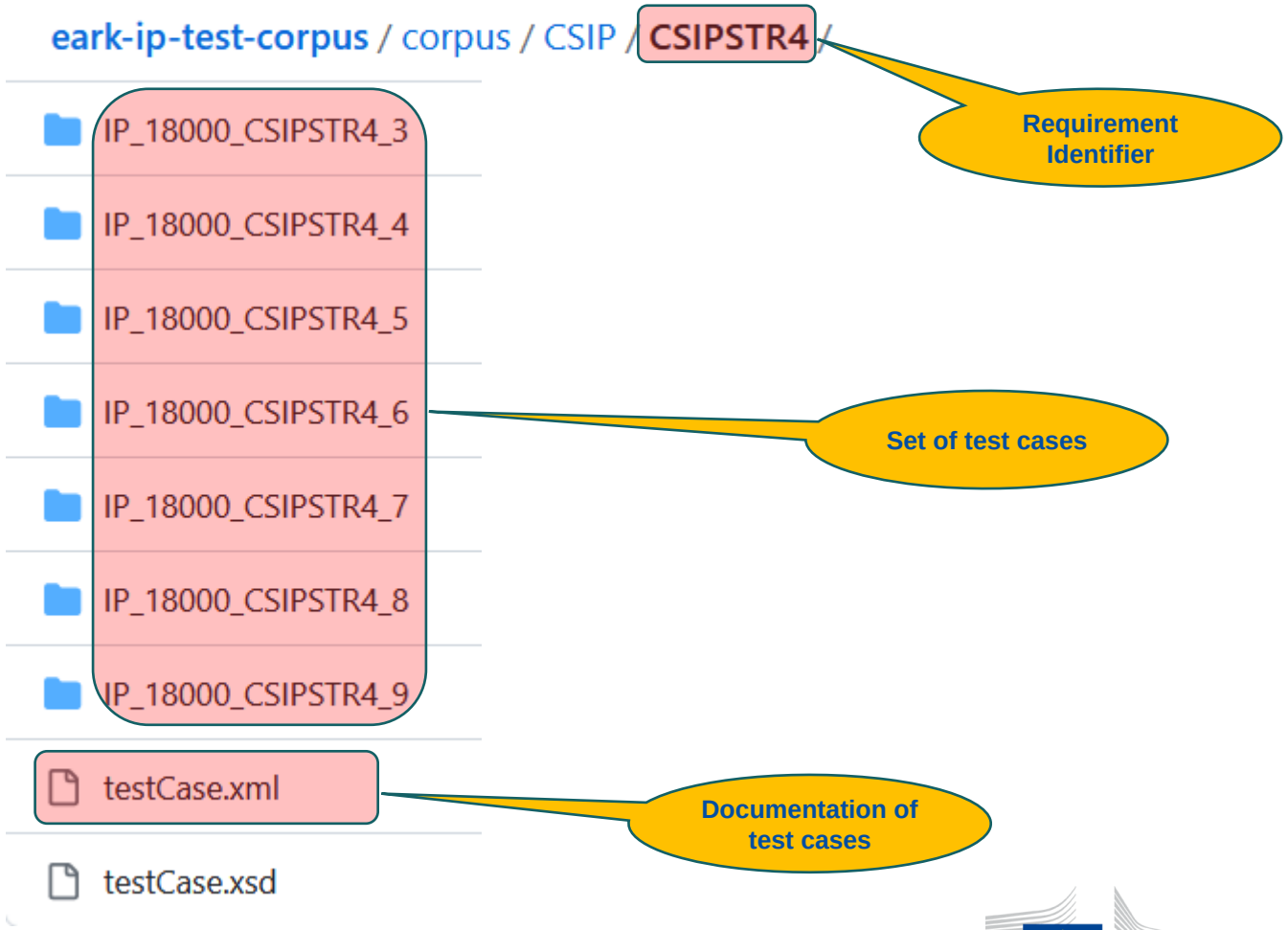
What is the test corpus?

- The test corpus comprises a set of human readable and machine-readable test cases
- Each test case is described within a xml-file named `testcase.xml`
- The `testcase.xml` defines a minimum of tests to be executed to verify conformance with a specific requirement



Structure of the E-ARK IP Test Corpus





European Commission

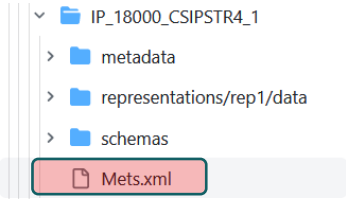
Documentation of test cases (testCase.xml)

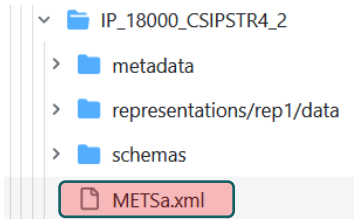
```
<testCase xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="testCase.xsd" testable="TRUE">
<id specification="E-ARK CSIP" version="2.0-DRAFT" requirementId="CSIPSTR4"/>
<references>
    <reference requirementId="CSIPSTR4" URL="http://earkcsip.dilcis.eu/#CSIPSTR4" />
  </references>
<rules>
<rule id="1">
<description>The Information Package folder MUST include a metadata file named `METS.xml` </description>
      <!-- Details of the validation error generated by the message, can be one of: [ ERROR | WARNING | INFO ]-->
      <error level="ERROR">
        <!-- Expected error message when the rule is violated. -->
        <message>The Information Package folder does not include METS.xml file. It MUST include a metadata file named `METS.xml` </message>
      </error>
      <corpusPackages>
        <!-- Details of corpus packages designed to test this validation rule. -->
        <package name="IP_18000_CSIPSTR4_1" isValid="FALSE">
          <path>/corpus/structure/CSIPSTR4/IP_18000_CSIPSTR4_1</path>
          <!-- Full description of corpus package features. -->
          <description>A package based on the minimal IP with schemas. File METS.xml named in camelcase (Mets.xml)</description>
        </package>
        <package name="IP_18000_CSIPSTR4_2" isValid="FALSE">
          <path>/corpus/structure/CSIPSTR4/IP_18000_CSIPSTR4_2</path>
          <!-- Full description of corpus package features. -->
          <description>A package based on the minimal IP with schemas. File METS.xml has extra lowercase character at the end</description>
        </package>
        ...
      </corpusPackages>
    </rule>
  </rules>
</testCase>
```

Expected Outcome

Test Case Location

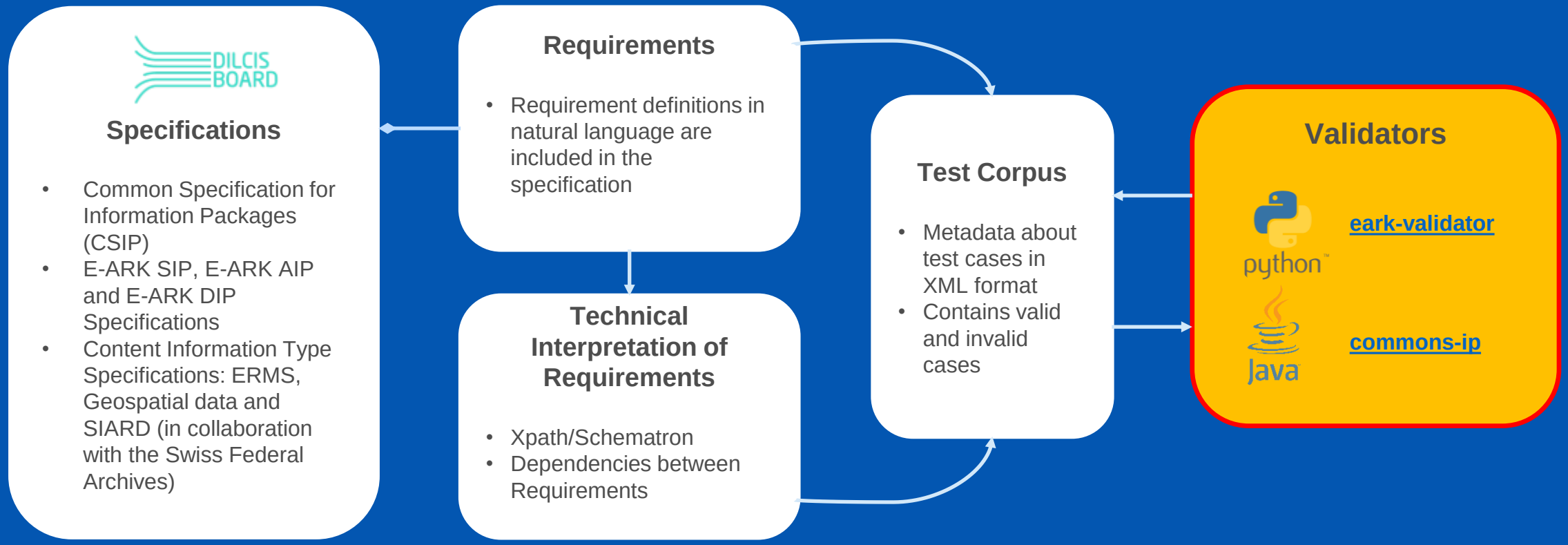
Test Case Identifier





European Commission

Validation Framework



Validate **CSIPSTR4**

```
IP
├── metadata
├── METS.xml
└── representations
    └── rep1
        └── data
            └── example.txt
```

```
$ file IP/METS.xml

IP/METS.xml: XML 1.0 document, Unicode
text, UTF-8 text
```

```
IP
├── metadata
└── representations
    └── rep1
        └── data
            └── example.txt
```

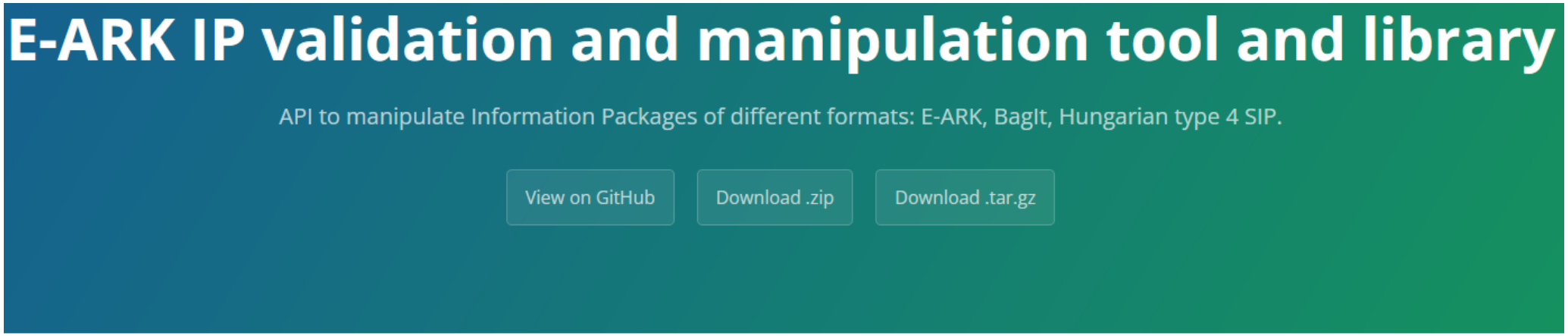
```
$ file IP/METS.xml

IP/METS.xml: cannot open `IP/METS.xml' (No
such file or directory)
```

Validator JSON output


```
{
    "specification" : "CSIP-2.0.4",
    "id" : "CSIPSTR4",
    "name" : "CSIP Information Package folder structure",
    "location" : "",
    "description" : "The Information Package root folder MUST include a file named METS.xml.
        This file MUST contain metadata that identifies the package, provides a high-level package
        description, and describes its structure, including pointers to constituent representations.",
    "cardinality" : "",
    "level" : "MUST",
    "testing" : {
      "outcome" : "PASSED",
      "issues" : [ ],
      "warnings" : [ ],
      "notes" : [ ]
    }
  }
```

E-ARK Validator (Java) – commons-ip



E-ARK IP validation and manipulation tool and library

This project provides a command-line interface and Java library to validate and manipulate OAIS Information Packages of different formats: E-ARK (version 1, 2.0.4, 2.1.0), BagIt, Hungarian type 4 SIP.

The E-ARK Information Packages are maintained by the Digital Information LifeCycle Interoperability Standards Board (DILCIS Board). DILCIS Board is an international group of experts committed to maintain and sustain maintain a set of interoperability specifications which allow for the transfer, long-term preservation, and reuse of digital information regardless of the origin or type of the information.

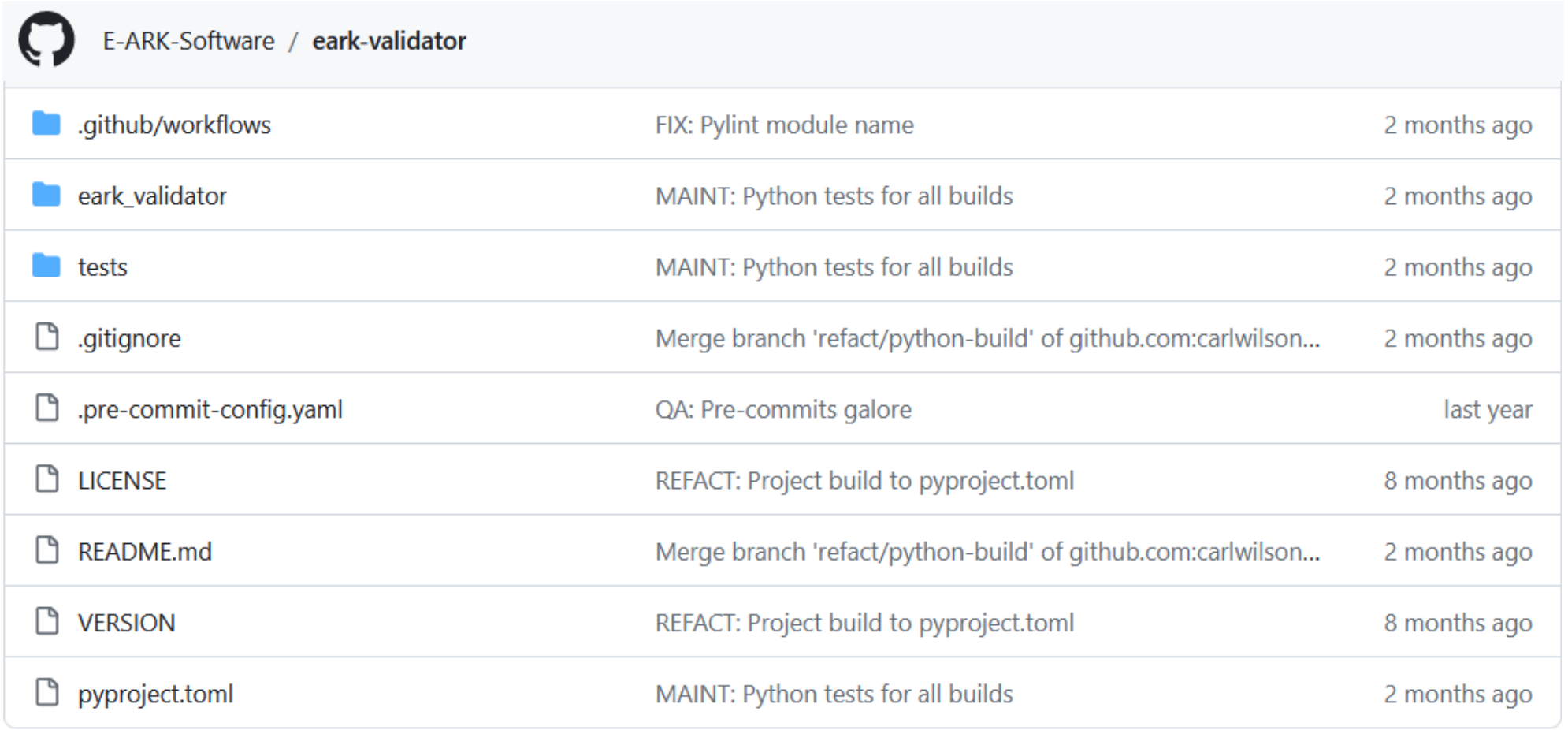
https://keeps.github.io/commons-ip/

Use commons-ip

- Requires Java (>= 17)
- Download the latest release
- Command: `java -jar commons-ip-cli-2.X.Y.jar validate -i sip1.zip sip2.zip -o output/`

More information: https://github.com/keeps/commons-ip/blob/master/README.md

E-ARK Validator (Python) – eark-validator



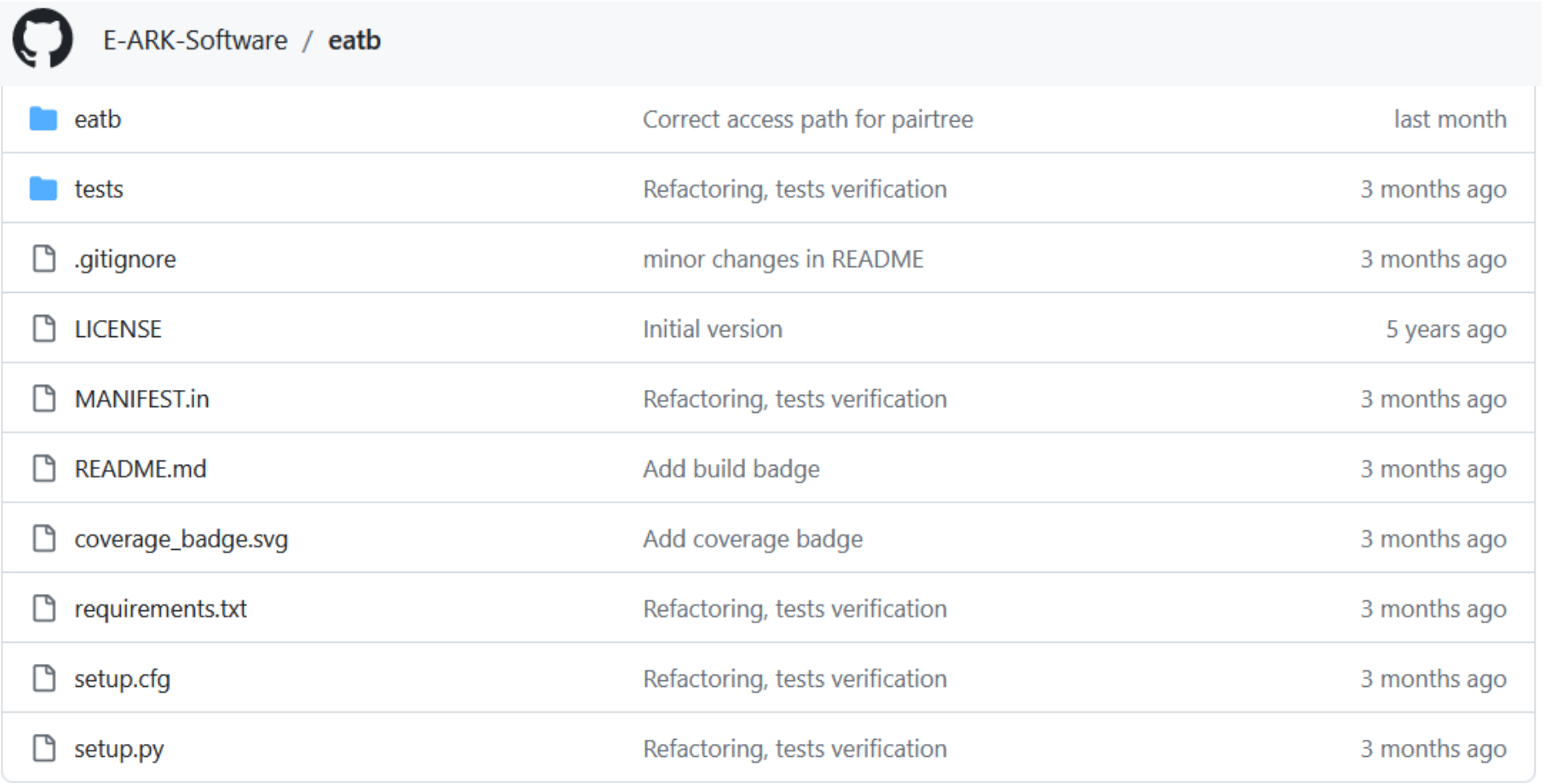
https://github.com/E-ARK-Software/eark-validator

Use commons-ip

- eark-validator on pypi.org: https://pypi.org/project/eark-validator/
- `pip install eark-validator`
- `eark-validator <path_to_directory_or_package>`

More information: https://github.com/E-ARK-Software/eark-validator

E-ARK Package Creation (Python) – eatb



E-ARK-Software / **eatb**

eatb	Correct access path for pairtree	last month
tests	Refactoring, tests verification	3 months ago
.gitignore	minor changes in README	3 months ago
LICENSE	Initial version	5 years ago
MANIFEST.in	Refactoring, tests verification	3 months ago
README.md	Add build badge	3 months ago
coverage_badge.svg	Add coverage badge	3 months ago
requirements.txt	Refactoring, tests verification	3 months ago
setup.cfg	Refactoring, tests verification	3 months ago
setup.py	Refactoring, tests verification	3 months ago

https://github.com/E-ARK-Software/eatb

eArchiving Conformance

What is the eArchiving Conformance Seal and how to get it

European Commission

EC’s eArchiving Conformance Seal Website

https://digital-strategy.ec.europa.eu/en/activities/earchiving-conformance-seal

eArchiving - Conformance Seal

The eArchiving Conformance Seal is a sign of quality of digital archiving, long term preservation, and data management following standards and best practices.

The eArchiving Conformance Seal demonstrates that a digital repository or archival solution is conformant with the E-ARK specifications. Below are the basic steps to acquire the eArchiving Conformance Seal.

There are several steps which need to be carried out by a designated representative from an organisation (digital repository maintainer) or company (solution provider).

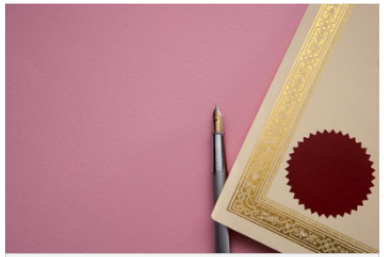
1. Validation
Validation is the process of **testing a single Information Package (IP)** file for conformance. You can start by verifying if individual information packages of your digital repository or archival solution are valid by using the **Validation Service** to test if these are conformant with E-ARK specifications.

2. Self-assessment conformance
eArchiving conformance of a digital archive or archiving solution includes **meeting the requirements for ingest and processing of Information Packages**, in addition to the validation of a representative sample of information packages.

3. Review
A review with the applicant by the eArchiving team will be performed to verify conformance against the **E-ARK specifications** and **requirements for ingest and processing of Information Packages**. This is a condition for acquiring the eArchiving Conformance seal.

4. Seal
After successful review the **eArchiving Conformance Seal** is awardedby the European Commission.

You can get more information on the eArchiving Conformance Seal website.



©Gettyimages

Technical specifications >

Knowledge Centre >

Training >

Support desk >

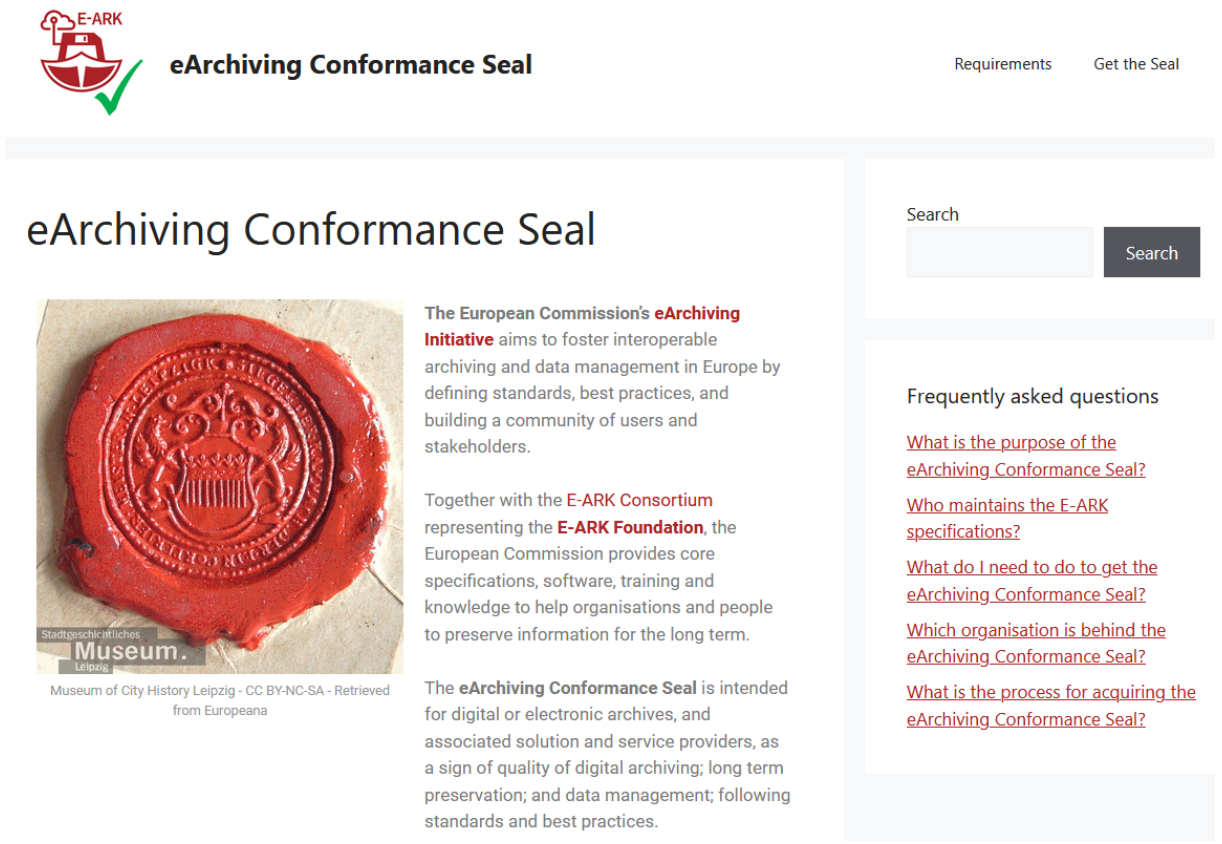
Get involved >

Back to eArchiving main page >

European Commission

E-ARK Consortium's Conformance Seal Website

https://seal.e-ark-foundation.eu

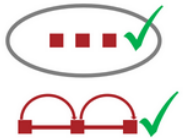


European Commission

Process for getting the eArchiving Conformance Seal

1. Validation
Validation is the process of **testing a single Information Package (IP)** file for conformance. You can start by verifying if individual information packages of your digital repository or archival solutions are valid by using the **Validation Service** to test if your information packages are conformant with E-ARK specifications.

Validation Service



2. Self-assessment conformance
eArchiving conformance of a digital archive or archiving solution
includes **to meet the E-ARK requirements for ingest and processing of Information Packages**, in addition to the validation of a representative sample of information packages.

Self-Assessment

3. Conformance application review
A brief interview with the eArchiving Team is arranged to review the evidence provided and verify the implementation of E-ARK **requirements for ingest and processing of Information Packages**.

Apply for a Review

4. Seal
After successful review the **eArchiving Conformance Seal** is **awarded by the European Commission.**

European Commission

Summary & Acknowledgements

Key statements and contributors

European Commission

Summary of key statements

- The eArchiving Validation Framework consists of:
  - ***Specifications*** including ***Requirements*** with corresponding ***Metadata Profiles***
  - ***Test Corpus***
  - ***Validators***
- Validation is key to ensure interoperability by introducing rules concerning the structure and metadata of information packages
- Validation rules are derived from digital preservation best practices

Acknowledgement

- Karin Bredenberg, Sydarkivera
  - DILCIS Board Lead & Management of E-ARK Specifications
- Carl Wilson & Darren Dignam, OPF
  - Specification Profiles & eark-validator (Python) Development & Deployment
- Phillip Tømmerholt, KMD
  - E-ARK Test Corpus
- Miguel Ferreira & Luís Faria, Keep Solutions
  - commons-ip (Java) Validator Development

European Commission

Thank you

European Commission