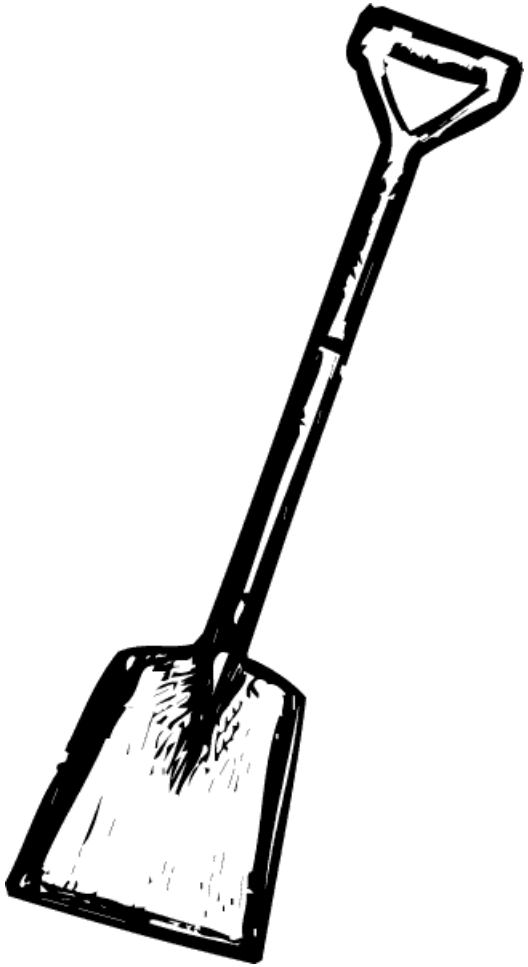
IMPLEMENTATION ISSUES



How PREMIS can be used

- For systems in development
  - as a basis for metadata definition
- For existing repositories
  - as a checklist for evaluation

"It seems that often people say they aren't ready to implement PREMIS yet, but they don't seem to realise they are already collecting some of the same information that PREMIS describes. The metadata is the same because it is often common sense that it is needed in a repository system. PREMIS can be useful to point out a few extra areas they perhaps hadn't thought of yet."

Deborah Woodyard-Robinson

Implementation issues: Reconciling data models

- PREMIS data model is for convenience of aggregation
- Context-dependent decisions
  e.g. an anomaly discovered during validation:
  - a property of the object or
  - an outcome of the validation event?
- Other data models equally valid
  e.g. NLNZ has Process, Object, File, Metadata
- However: PREMIS encourages consistent application of preservation metadata across different categories of objects (representation, file, bitstream)

Implementation issues: Implementation in relational databases

- PREMIS data model is not entity-relationship model

Implementation issues: obtaining values

- What values to use for controlled vocabularies?
  - In version 1, PREMIS has not had a semantic unit to indicate what controlled vocabulary is used
  - Version 2 introduces a mechanism to document controlled vocabularies
  - LC set up registries with starter lists (taken from “suggested values”)

Controlled vocabularies databases

- Library of Congress is establishing databases with controlled vocabulary values for standards that it maintains
- Controlled lists are represented using SKOS as well as alternative syntaxes

Controlled vocabularies databases

- Lists currently in progress:
  - ISO 639-2 and MARC language code list
  - MARC geographic area codes
  - MARC country code list
  - MARC relators
  - PREMIS controlled value lists
  - Thesaurus of Graphic Material
- Other possibilities
  - Enumerated values in MODS schema
  - Coded and uncoded value lists in MARC

Controlled vocabularies in SKOS: example

```
<rdf:Description rdf:about=
  "http://www.loc.gov/standards/registry/vocabulary
  /preservationEvents/creation">
   <rdf:type rdf:resource=
     "http://www.w3.org/2008/05/skos#Concept" />
   <skos:prefLabel xml:lang="en-latn">
     creation</skos:prefLabel>
   <skos:narrower rdf:resource=
     "http://www.loc.gov/standards/registry/vocabul
     ary/preservationEvents/migration" />
   <skos:narrower rdf:resource=
     "http://www.loc.gov/standards/registry/vocabul
     ary/preservationEvents/normalization" />
   <skos:definition xml:lang= "en-latn">the act of
     creating a new object</skos:definition>
   <skos:inScheme rdf:resource=
     "http://www.loc.gov/standards/registry/vocabul
     ary /preservationEvents" />
</rdf:Description>
```

Using controlled vocabularies in PREMIS

- Semantic units that specify a controlled vocabulary: realized as “concept scheme”
- Each value: realized as SKOS instance
- Implementers add their values within a concept scheme
- Mechanism to import the values into the PREMIS XML schema to enable validation
- A concept in multiple standards may be established for broad usage in a concept scheme
- LC is exploring an RDF version of PREMIS for semantic web applications

Those wishing to experiment: http://id.loc.gov/

Implementation issues: conformance

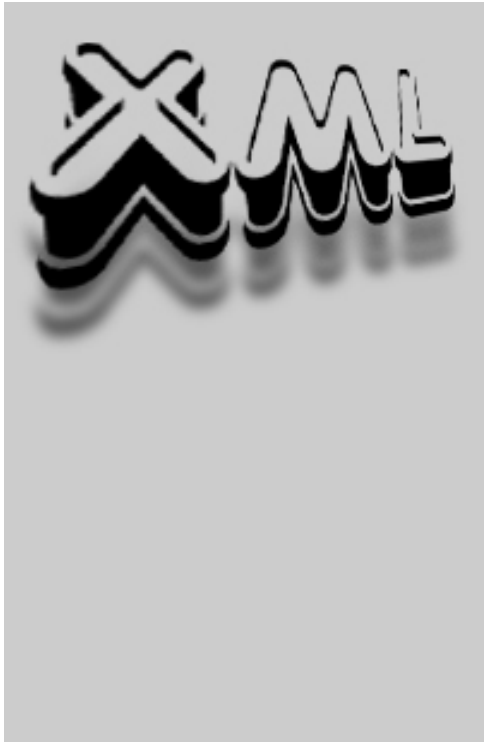
- Conformance is defined in PREMIS Final Report
  - if you use the name, use the definition
  - local metadata can supplement but not modify PREMIS
  - can define more stringent repeatability and obligation, but not more liberal

- Meaning of mandatory:
  - you have to know it, and you have to be able to supply it if exporting for exchange
  - you don't have to record it in the repository

Implementation issues: additional metadata

- preservation metadata that is not “core”
  - core = all objects, all preservation strategies
  - example of non-core = installation requirements
- more detailed information on Agents
- metadata describing Intellectual Entity
- business rules of the repository
- information about the metadata itself (e.g., who obtained or recorded a value, when last changed…)

XML issues



W3C Recommendation

A Brief Introduction to XML

Extensible Markup Language (XML) 1.0 (Fourth Edition)

W3C Recommendation 16 August 2006, edited in place 29 September 2006

This version:
http://www.w3.org/TR/2006/REC-xml-20060816
Latest version:
http://www.w3.org/TR/xml
Previous version:
http://www.w3.org/TR/2006/PER-xml-20060614
Editors:
Tim Bray, Textuality and Netscape <tbray@textuality.com>
Jean Paoli, Microsoft <jeanpa@microsoft.com>
C. M. Sperberg-McQueen, W3C <cmsmcq@w3.org>
Eve Maler, Sun Microsystems, Inc. <eve.maler@east.sun.com>
François Yergeau

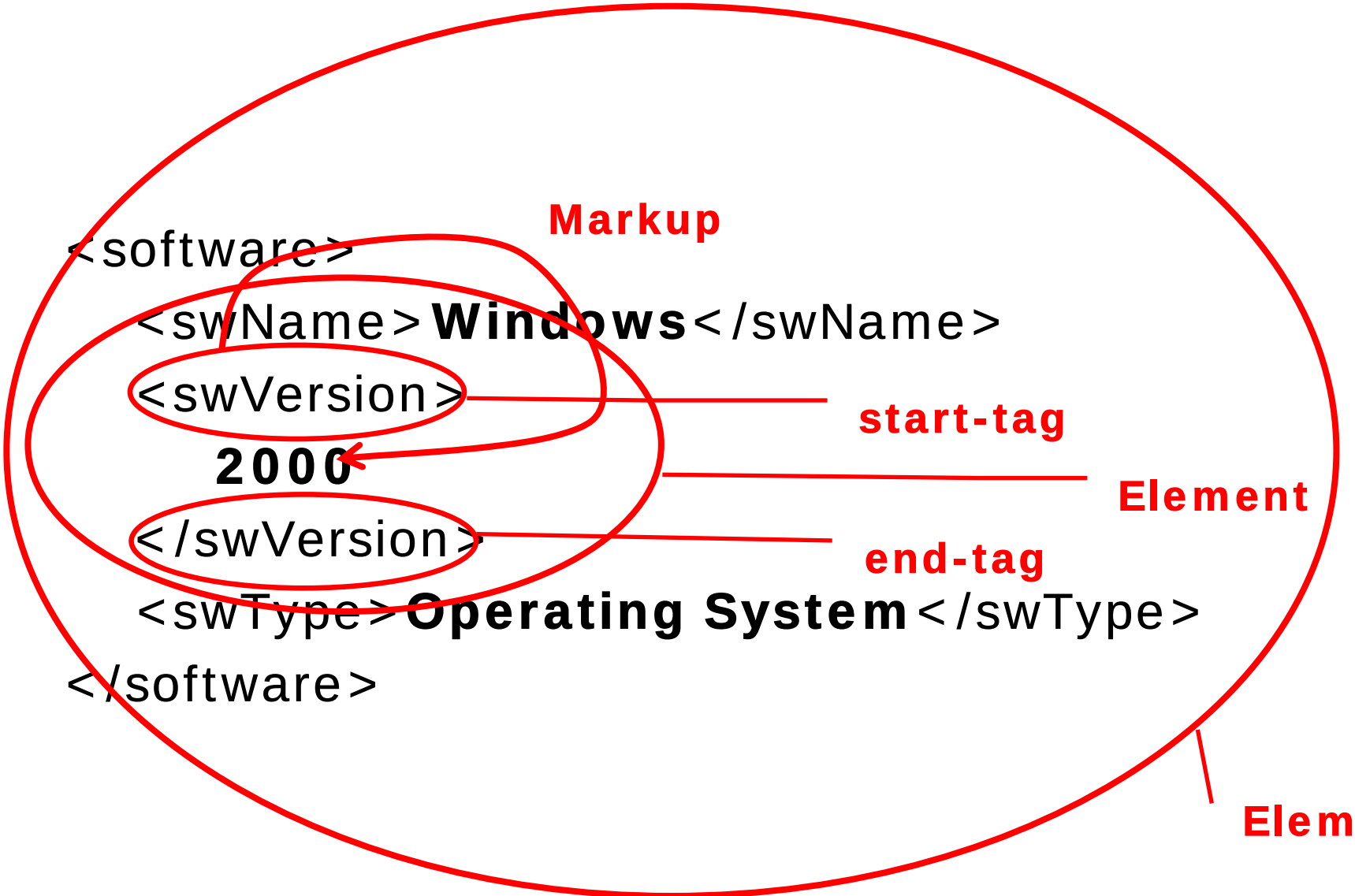
Please refer to the **errata** for this document, which may include some normative corrections.

The previous errata for this document, are also available.

See also **translations**.

This document is also available in these non-normative formats: XML and XHTML with color-coded revision inc

XML Example: Data + Meaning = Information?



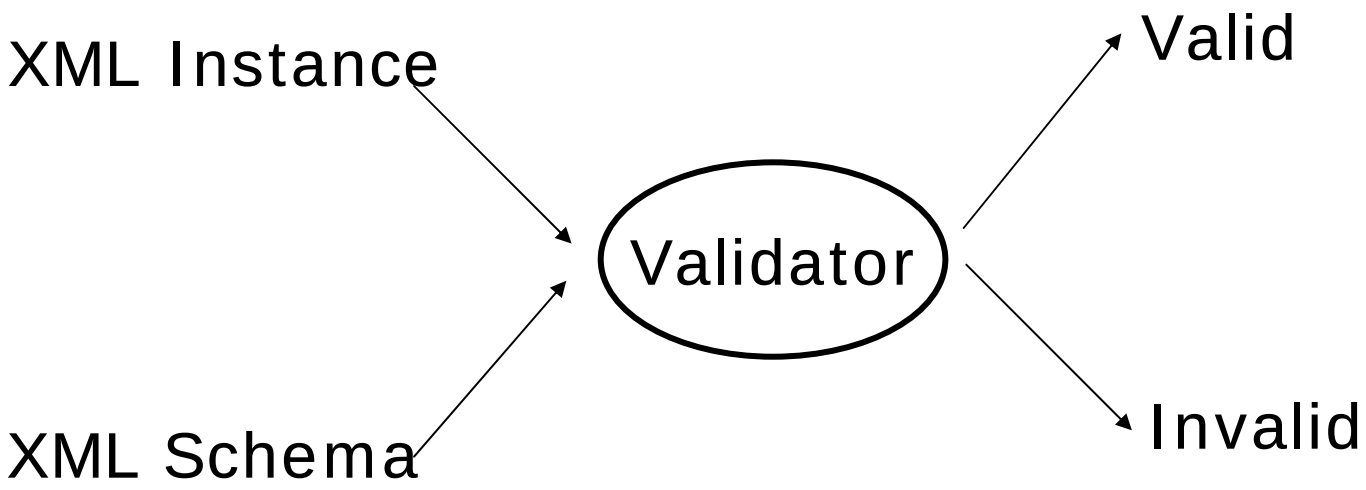
XML: Extensible Markup Language

- A technical approach to convey meaning with data
- Not a natural language, uses natural languages
  - <name>Louis Armstrong</name>
- Not a programming language
- A limited set of tags defines the vocabularies that can be used to markup data
- The set of tags and their relationships need to be explicitly defined (e.g., in XML schema)
- We can build software that uses XML as input and process them in a meaningful way
- You can define your own markups and schemas

XML Schema Defines:

- What elements may be used?
- Of which types?
- Any attributes?
- In which order?
- Optional or compulsory?
- Repeatable?
- Subelements?
- Logic?
- ...

XML Validation



PREMIS Publishes official schemas for validating the XML implementations.

XML Schema Examples

```
<xs:element name="software" minOccurs="0"
   maxOccurs="unbounded">
   <xs:complexType>
        <xs:sequence>
             <xs:element name="swName"
                  minOccurs="1" maxOccurs="1"
                  type="xs:string"></xs:element>
             <xs:element name="swOtherInformation"
                  minOccurs="0" maxOccurs="unbounded"
                  type="xs:string"> </xs:element>
        </xs:sequence>
   </xs:complexType>
</xs:element>
```

Will the following XML validate?

```
<software>
    <swName>Windows</swName>
    <swOtherInformation>Operating System
                        </swOtherInformation>
</software>
```

PREMIS XML schemas

- In version 1: 5 schemas, one for each PREMIS entity in the data model and a container schema

- In version 2 an instance is
  - (1) One or more of <object>, <event>, <agent>, <rights> all wrapped within a <premis> container; or
  - (2) any one of <object>, <event>, <agent>, <rights> by itself.

  - Thus the root element is one of the following: <premis>, <object>, <event>, <agent>, <rights>

PREMIS XML schemas

- Semantic units in PREMIS schemas
  - XML is faithful to data dictionary
  - Semantic units for objects may be validated according to the level for which they are applicable (i.e. representation, file, bitstream)

http://www.loc.gov/standards/premis/premis.xsd

Significant changes in XML schema v 2.0

- Extensibility mechanism is provided for further structure or for schemas from other namespaces
  - significantProperties
  - objectCharacteristics
  - creatingApplication
  - environment
  - signatureInformation
  - eventOutcomeDetail
  - Rights

Significant changes in XML schema v 2.0

- An abstract object type allows for better validation of object category; objectCategory is not an element
- Defining main elements globally allow for reuse
- Includes definitions for types of date expressions not in W3CDTF,
  including ISO 8601 basic format (without hyphens)
  and conventions for special types of dates (e.g. open-ended or questionable dates)

  http://www.loc.gov/standards/datetime/

Date and time formats

- Use of a structured form to aid machine processing
  - To be implementation independent, no particular standard specified
- Conventions are needed to express other aspects of a time period, such as an open-ended or questionable date.
- Semantic units that may include a date or date and time:
  - preservationLevelDateAssigned
  - dateCreatedByApplication
  - eventDateTime
  - copyrightStatusDeterminationDate
  - statuteInformationDeterminationDate
  - startDate
  - endDate

Implementing PREMIS using XML in METS

METS Introduction - Extensibility

- METS is open source and developed by open discussion, mainly cultural heritage community
- Describes complex and compound objects
- Encapsulates administrative, structural, and descriptive metadata about digital objects

METS Introduction - Extensibility

- XML based (xml schema)
- Modular & extensible
  - elements from other schemas can be plugged in
  - uses the XML Schema facility for combining vocabularies from different Namespaces
- METS uses extension “wrappers” or “sockets” where elements from other schemas can be plugged in (called extension schemas)

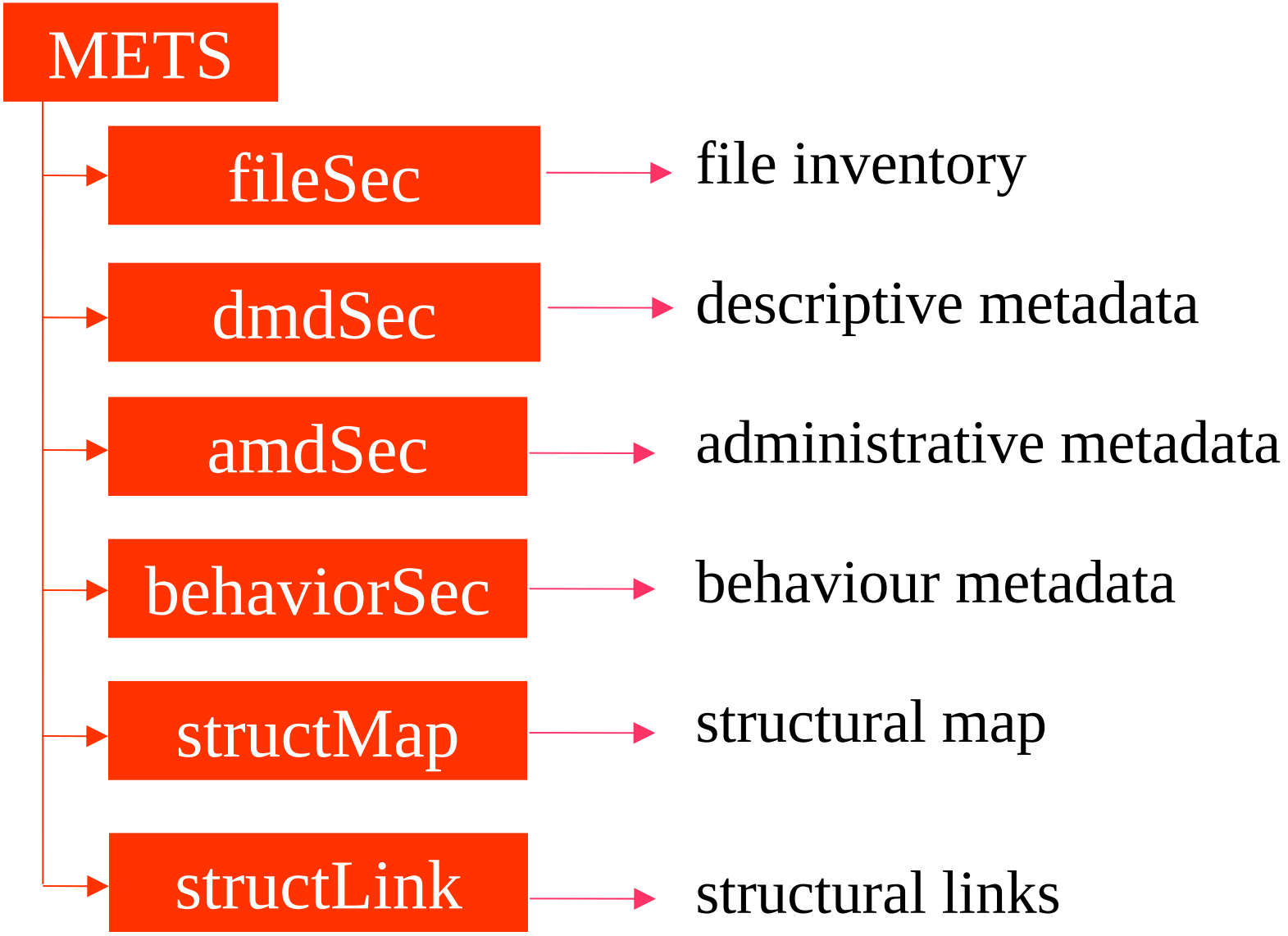
METS Introduction - Extensibility

- Many institutions trying to use PREMIS within the METS context
- The METS Editorial Board has endorsed PREMIS as an extension schema
- Endorsed extension schemas:
  - Descriptive: MODS, DC, MARCXML
  - Technical metadata: MIX (image); textMD (text)
  - Preservation related: PREMIS

METS Introduction

- Records the structure of digital objects
- Records the names and locations of the files that comprise those objects.
- Records relationships among the metadata and among the pieces of the complex objects
- Describes and attaches executable behaviour appropriate for content
- A unit of storage (e.g. OAIS AIP) or a transmission format (e.g. OAIS SIP or DIP)
- Content-type independent
- Batch processing for creation, processing, retrieval, and presentation
- Text editor, XML editor, or a forms-based user interface

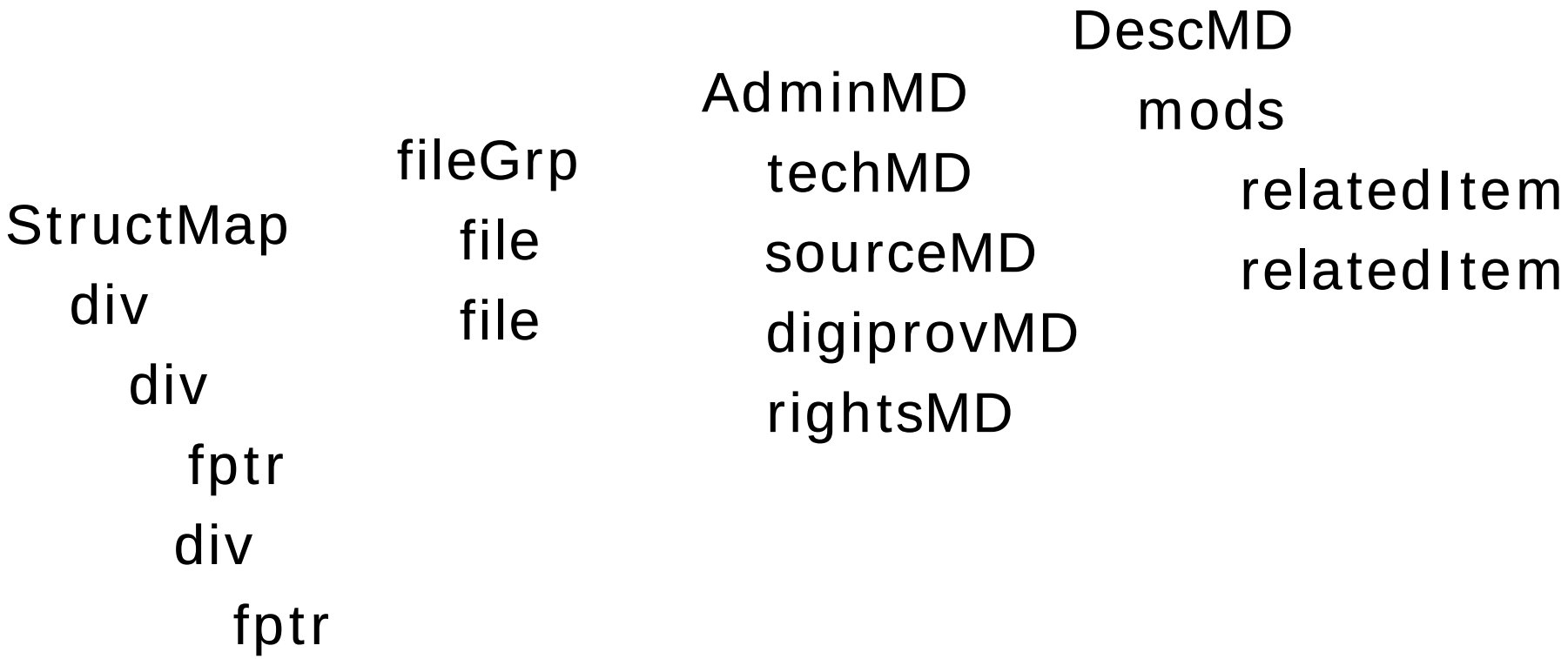
The structure of a METS file



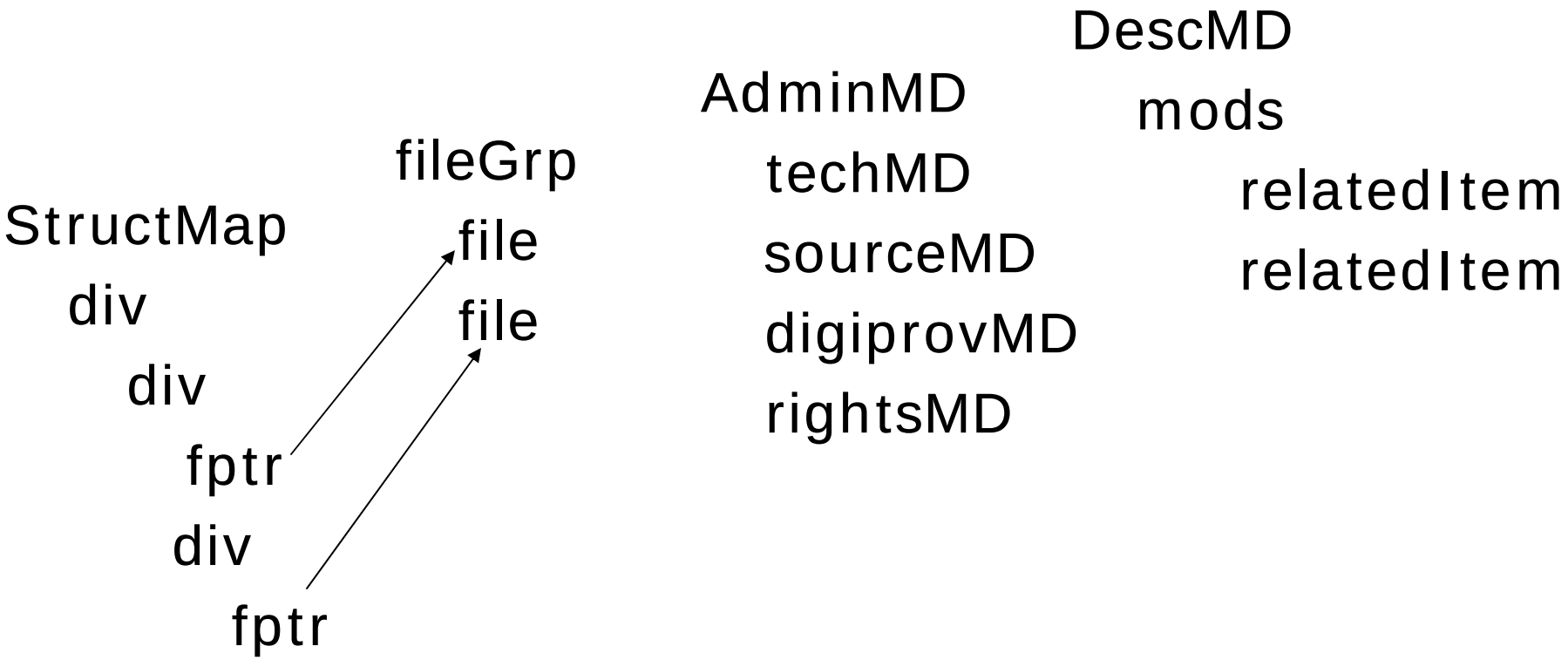
Inserting technical metadata in a METS Document

```
<mets>
  <amdSec>
    <techMD>
      <mdWrap>
        <xmlData>
          <!--  insert data from different namespace here  -->
        </xmlData>
      </mdWrap>
    </techMD>
  </amdSec>
  <fileSec />
  <structMap />
</mets>
```

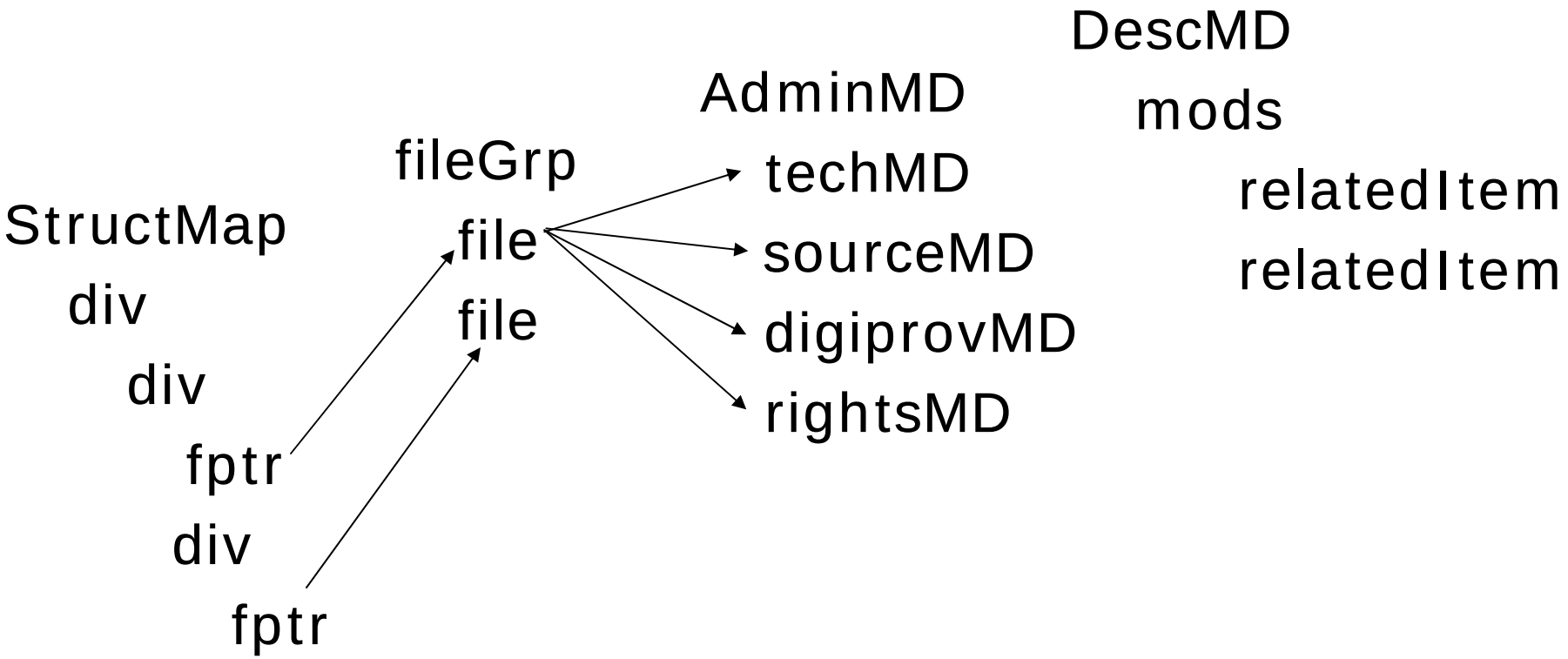
Linking in METS Documents (XML ID/IDREF links)



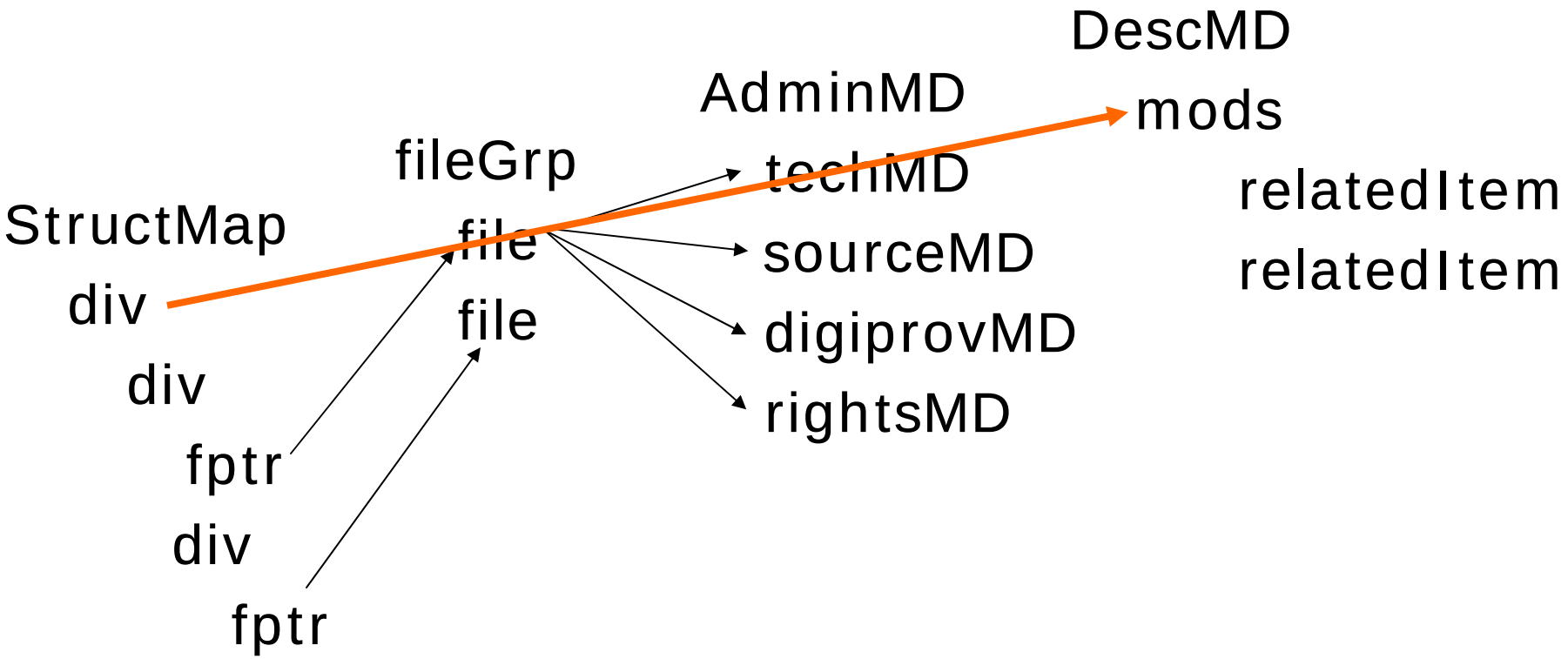
Linking in METS Documents (XML ID/IDREF links)



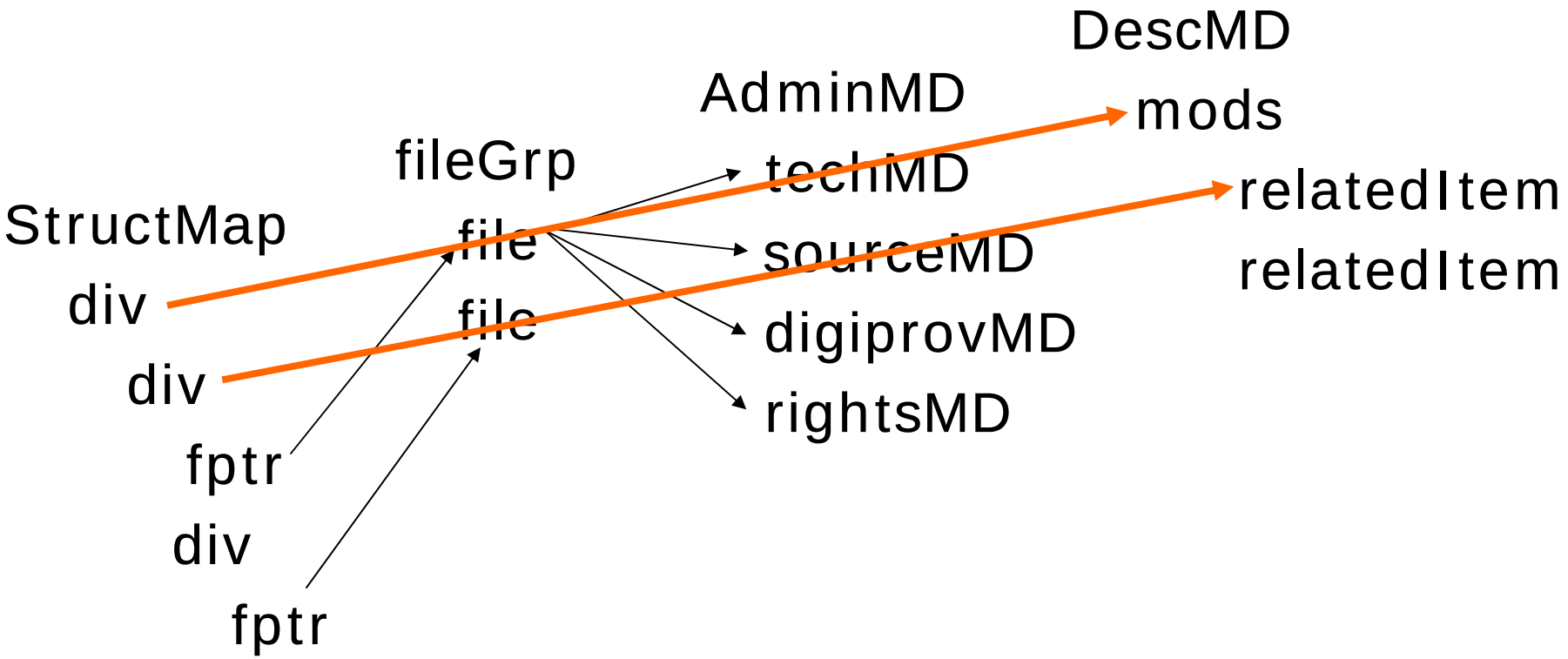
Linking in METS Documents (XML ID/IDREF links)



Linking in METS Documents (XML ID/IDREF links)



Linking in METS Documents (XML ID/IDREF links)



Issues in using PREMIS with METS

- Flexibility of METS requires implementation decisions:
  - Which METS sections to use
  - How many administrative MD sections to use?
  - Use PREMIS container or separate packages?
  - Whether to record elements redundantly in PREMIS and METS
  - How to record elements that are also part of a format specific technical metadata schema (e.g. MIX)
  - Where to store structural relationships?
  - How to deal with locally controlled vocabularies

Experimentation will result in best practices – guidelines might help

PREMIS and METS sections

- You can't put all PREMIS metadata directly under amdSec
- What sections to use for PREMIS metadata?
  - Alternative 1
    - Object in techMD
    - Event in digiProvMD
    - Rights in rightsMD
    - Agent with event or rights
  - Alternative 2
    - Everything in digiProvMD
  - Alternative 3
    - Everything in techMD
- How many administrative MD sections to use?

PREMIS and METS sections

- **Guidelines:**

http://www.loc.gov/standards/premis/guidelines-premismets.pdf

PREMIS and METS sections

- **Guidelines: number of sections**
  - Use one amdSec with repeating subelements (techMD, etc.) **OR** repeating amdSec for each subelement
  - Agent in conjunction with an event or right should be stored in its own digiProvMD or rightsMD section to avoid redundancy
  - Technical metadata from different schemas should be stored in separate techMD sections or can be embedded into PREMIS' objectCharacteristicsExtension.

PREMIS and METS sections

- **Guidelines: PREMIS in METS sections**
  - Object under techMD or digiProvMD
    - Files/bitstream: techMD
    - Representation: digiProvMD
  - Event in digiProvMD
  - Rights in rightsMD
  - Agent in digiProvMD or rightsMD (depending if attached to event or rights)

Local decisions may vary depending on processing model

PREMIS and METS sections

- **Guidelines: PREMIS container?**
  - If an implementation wants to keep all PREMIS metadata together the PREMIS container is used.
  - In this case the PREMIS package must go into digiProvMD

PREMIS and METS sections

- **Guidelines: structural relationships?**

  - Hierarchical relationships: <mets:div> elements should be used (richer than PREMIS semantic units).
  - Store the PREMIS relationship elements in the Object schema redundantly, if the scope of exchanging objects is preservation
  - Other, derivative types of relationships should always be stored in PREMIS relationship

PREMIS and METS sections

- **Guidelines: ID/IDREF referencing?**
  - PREMIS and METS are using ID/IDREF to link elements:
    - METS: <amdSec ID=“”/>
    - PREMIS: linkingEventIdentifier, LinkEventXmlID etc
- METS’ IDREF attributes must not link to PREMIS elements
- PREMIS linking-attributes must not link to METS elements

ID/IDREF links are only valid within the same schema

PREMIS and METS sections

- **Guidelines: ID/IDREF referencing?**
  - If it is intended to use the PREMIS outside of the METS container, redundant linking is necessary as METS ID/IDREF mechanism might break
  - Links from METS to PREMIS sections should be made on the highest level possible – usually pointing to the first level subelement under amdSec (digiProvMD, techMD etc.)

Elements defined in both METS and PREMIS:

- **METS: CHECKSUM, CHECKSUMTYPE**
  - attribute of <file>
  - not repeatable
- **PREMIS: fixity**
  - also includes messageDigestOriginator
  - allows multiples

- **METS: SIZE**
  - attribute of <file>
- **PREMIS: size**

```
<fileSec>
    <fileGrp>
        <file    ID="FID1"
                 SIZE="184302"
                 ADMID="TMD1PREMIS TMD1MIX DP1EVENT “
                 CHECKSUM="4638bc65c5b9715557…2ecbf"
                 CHECKSUMTYPE="SHA-1">
                 <FLocat LOCTYPE="OTHER" xlink:href="BXF22.JPG" />
</file></fileGrp></fileSec>
<techMD ID="TMD1PREMIS">
    <mdWrap MDTYPE="PREMIS">
        <xmlData>
            <premis:object >
                <objectCharacteristics>
                    <fixity>
                        <messageDigestAlgorithm>
                            SHA-1
                        </messageDigestAlgorithm>
                        <messageDigest>
                            4638bc65c5b97155…2ecbf
                        </messageDigest>
                        <messageDigestOriginator>
                            EchoDep
                        </messageDigestOriginator>
                    </fixity>
                    <size>184302</size>
                </objectCharacteristics>
```

Elements defined both in METS and PREMIS:

- **METS: MIMETYPE**
  - attribute of <file>
  - optional
- **PREMIS: <format>**
  - more granular; includes name and version (although name may be MIMETYPE)
  - mandatory

```
<fileSec>
   <fileGrp>
        <file    ID="FID1"
                 ADMID="TMD1PREMIS DP1EVENT DP1AGENT“
                 MIMETYPE="image/jpeg“>
                 <FLocat LOCTYPE="OTHER" xlink:href="BXF22.JPG"/>
</file></fileGrp></fileSec>

<techMD ID="TMD1PREMIS“
   <mdWrap MDTYPE="PREMIS">
        <xmlData>
                 <premis:object>
                          <objectCharacteristics>
                                   <format>
                                            <formatDesignation>
                                                     <formatName>
                                                              image/jpeg
                                                     </formatName>
                                                     <formatVersion>
                                                              1.02
                                                     </formatVersion>
                                            </formatDesignation>
                                   </format>
                          </objectCharacteristics>
```

Elements defined both in METS and PREMIS:

- **METS ID/IDref:**
  - used to associate metadata in different sections and for different files
- **PREMIS identifiers**:
  - explicit linking between entity types

```
<fileSec>
    <fileGrp>
        <file    ID="FID1"
                 ADMID="TMD1PREMIS TMD1MIX DP1EVENT DP1AGENT">
...
<techMD ID="TMD1PREMIS">
    <linkingEventIdentifier>
        <linkingEventIdentifierType>
                ECHODEP</linkingEventIdentifierType>
        <linkingEventIdentifierValue>
                echo12345</linkingEventIdentifierValue>
    </linkingEventIdentifier>

<digiprovMD ID="DP1EVENT">
    <premis:event>
        <eventIdentifier>
                <eventIdentifierType>
                        ECHODEP</eventIdentifierType>
                <eventIdentifierValue>
                        echo12345</eventIdentifierValue>
        </eventIdentifier>
        <eventType>ingestion</eventType>
```

Elements defined both in METS and PREMIS:

- **METS: structMap**
  - details structural relationships and is the heart of the METS document
  - hierarchical, so may be more expressive than PREMIS semantic units
  - links the elements of the structure to content files and metadata
- **PREMIS: <relationship>**
  - details all kinds of relationships, including structural
  - data dictionary says that implementations may record by other means

```
<structMap TYPE=“physical”>
    <div ORDER="1" TYPE="text">
        <:fptr FILEID="FID9"/>
        <div ORDER="1" TYPE="page" LABEL=" Page [1]">
                <fptr FILEID="FID1"/></mets:div>
        <div ORDER="2" TYPE="page" LABEL=" Page [2]">
                <fptr FILEID="FID2"/></mets:div>
    </div>

<relationship>
    <relationshipType>structural</relationshipType>
    <relationshipSubType>is sibling of </relationshipSubType>
    <relatedObjectIdentification>
        <relatedObjectIdentifierType>
                UCB</relatedObjectIdentifierType>
        <relatedObjectIdentifierValue>
                FID2</relatedObjectIdentifierValue>
        <relatedObjectSequence>1</relatedObjectSequence>
```

Should semantic units be recorded redundantly?

- Various options are possible when there is overlap between PREMIS and METS or PREMIS and other technical metadata schemas
  - Record only in METS
  - Record only in PREMIS
  - Record in both
- Are there advantages in using PREMIS semantic units?
- Is it important to keep PREMIS metadata together as a unit? There may be an advantage for reuse and maintenance purposes

How to record elements from 2 different technical metadata schemas

- Format specific metadata may be included in addition to PREMIS general technical metadata
- Use multiple techMD sections and specify source in MDType attribute and/or namespace declaration
  - e.g. MDTYPE="NISOIMG" or "PREMIS"
  - Give MIX schema declaration in METS document
- MIX was recently revised to correspond with the revision of the Z39.87 technical metadata for digital still images standard; names harmonized with corresponding PREMIS semantic units
- For digital still images:
  use PREMIS for general semantic units defined in PREMIS and MIX for format specific units without redundancy

Examples of PREMIS in XML

- **PREMIS in METS:**
  - Portrait of Louis Armstrong (XML) (Library of Congress)

    Web Presentation of this object
  - Peoria County, Illinois aerial photograph (ECHO Depository, UIUC Grainger Engineering Library)

Examples of PREMIS in XML

- **MATHARC implementation:**
  http://pigpen.lib.uchicago.edu:8888/pigpen/uploads/13/asset_descr_mets_premis_02v2.xml

- **UC examples using PREMIS**
  - **Stanford (geospatial and "transfer manifest")**
  - **UCSD (complex object)**
  - **UCB (general METS profile)**

- **British Library Examples**
  - **eJournals, newspapers, WebArchiving**

Application Profiles (1)

- Documenting the structure of an information package
  - (Preservation) metadata is part of the information package
- Use/purpose of semantic units/metadata elements
  - Values: where do they come from? controlled vocabulary used?
  - Purpose of storing this information (how is it used?)
  - May be use case specific (e.g. in case of migration…)
- METS profile(s): just a template
  - some shortcomings, but at least a start
  - Re-use profiles

Application Profiles (2)

- Checklist for documenting PREMIS-METS decisions in a METS profile : (Sally Vermaaten, OCLC)
  - What schemas are used? (MODS? PREMIS? MIX?)
  - How does the profile relate to other profiles?
  - What controlled vocabularies are used
  - Is PREMIS wrapped or referenced?
  - PREMIS bundled or distributed?
  - Separate amdSec elements OR amdSec subelements?

Application Profiles (3)

- Checklist for documenting PREMIS-METS decisions in a METS profile : (Sally Vermaaten, OCLC)
  - What PREMIS semantic units does the profile require/recommend?
  - Technical metadata: in separate techMD or premis:objectCharacteristicExtension?
  - How are relationships expressed? (METS div elements? Or premis:relationships?
  - What level of objects does PREMIS describe?

Application Profiles (4)

- Checklist for documenting PREMIS-METS decisions in a METS profile : (Sally Vermaaten, OCLC)
  - How are linking identifiers, IDREFs and premis identifiers used?
  - PREMIS-METS redundancies
  - Metadata tools or applications used?

Summary: container formats

- A container format is needed to package all forms of metadata (of which PREMIS is one) and digital content
- Use of a container is compatible with and an implementation of the OAIS information package concept
- Co-existence with other types of metadata requires best practices for both approaches; redundancy seems to be preferred

Summary: container formats

- Changes to the next version of the PREMIS XML schemas will facilitate a phased approach to full PREMIS implementation
- Development of registries for controlled vocabularies will benefit implementation
- Tools are being/were developed to facilitate implementation
- Application profiles are important for documenting the use of metadata in an information package