

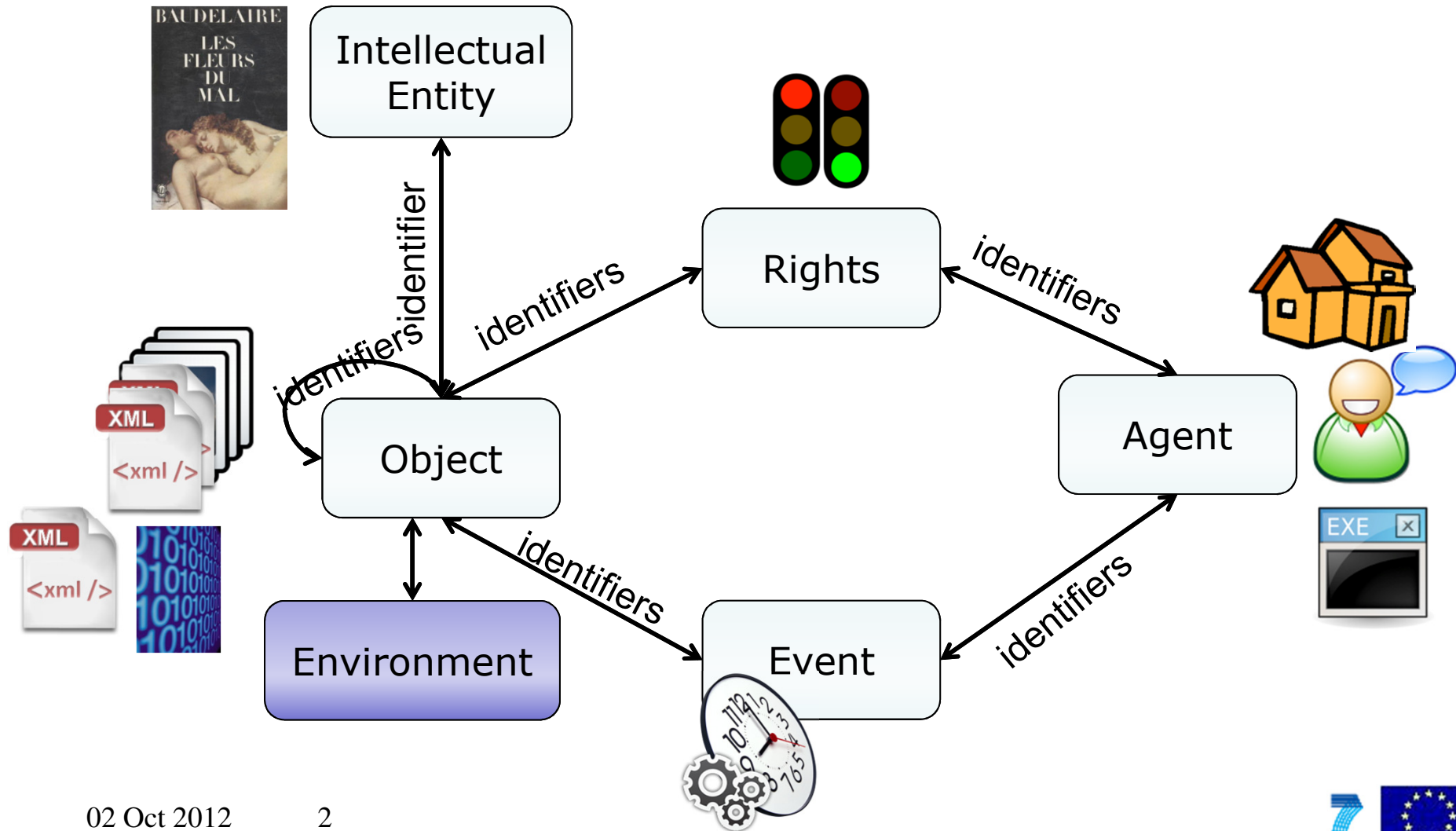
Proposed Data Model Changes for PREMIS 3.0

Angela Dappert

Digital Preservation Coalition

The PREMIS Data Model

Slide by S. Peyrard



Intellectual Entities

- A set of content that is considered a single intellectual unit for purposes of management and description
- For example, a particular book, map, photograph, or database.
- An Intellectual Entity can include other Intellectual Entities; for example, a Web site can include a Web page; a Web page can include an image.
- An Intellectual Entity may have one or more digital representations.

Intellectual Entities Implementation

- Can capture descriptive metadata.
- Assumed to be held in a container metadata schema.
- PREMIS Objects link to it.

IntellectualEntities have been used as Objects to

- ★ Represent a collection, FRBR work, FRBR expression, fonds, series, files (in the archival sense)
 - Represent a batch of files which share environments and events in order to avoid repetition of this information
 - Represent transport packages (TIPR)
 - ★ Capture versioning information and metadata update events for intellectualEntities, such as articles and issues.
 - Describe the output of a complex event such as a web crawl.
- Some of these are shortcuts for “cleaner” but more verbose solutions

Represent a collection, FRBR work, FRBR expression, fonds, series, files ... in order to

- capture descriptive metadata
 - to have business requirements associated with them or to be referenced in business requirements (such as significant characteristics, risk definitions, guidelines for preservation actions, etc.)
 - structural and derivative relationships
 - rights and preservation rights information
 - events and agents
-
- This can only partially be accommodated by container metadata systems and their associated descriptive or administrative metadata.
 - Core preservation metadata (provenance aspects)

Capture versioning information and metadata update events for intellectualEntities, such as articles and issues

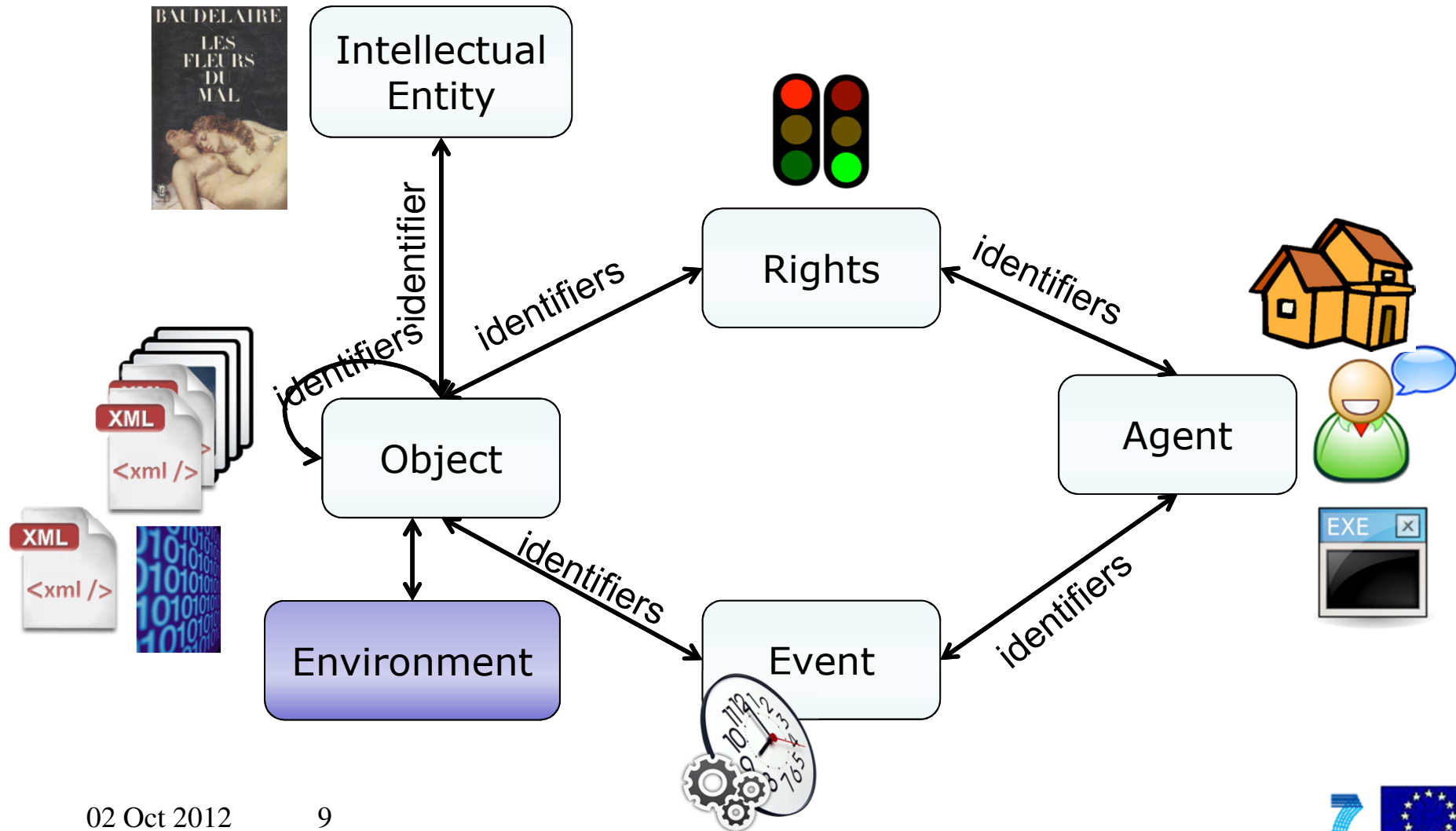
- This should be accommodated on the container metadata level.
METS does not record metadata of metadata
- This could be addressed by treating the metadata as a file that can have its own metadata which can record modification information.
=> awkward

Treat IntellectualEntity as Object Type

- Data model will be more compact.
- Data dictionary will be more self-contained.
- Simplify the dictionary
 - drop linkingIntellectualIdentifier
- Directly attach events, rights, indirectly attach agents to intellectual entities

The PREMIS Data Model

Slide by S. Peyrard



Required Changes

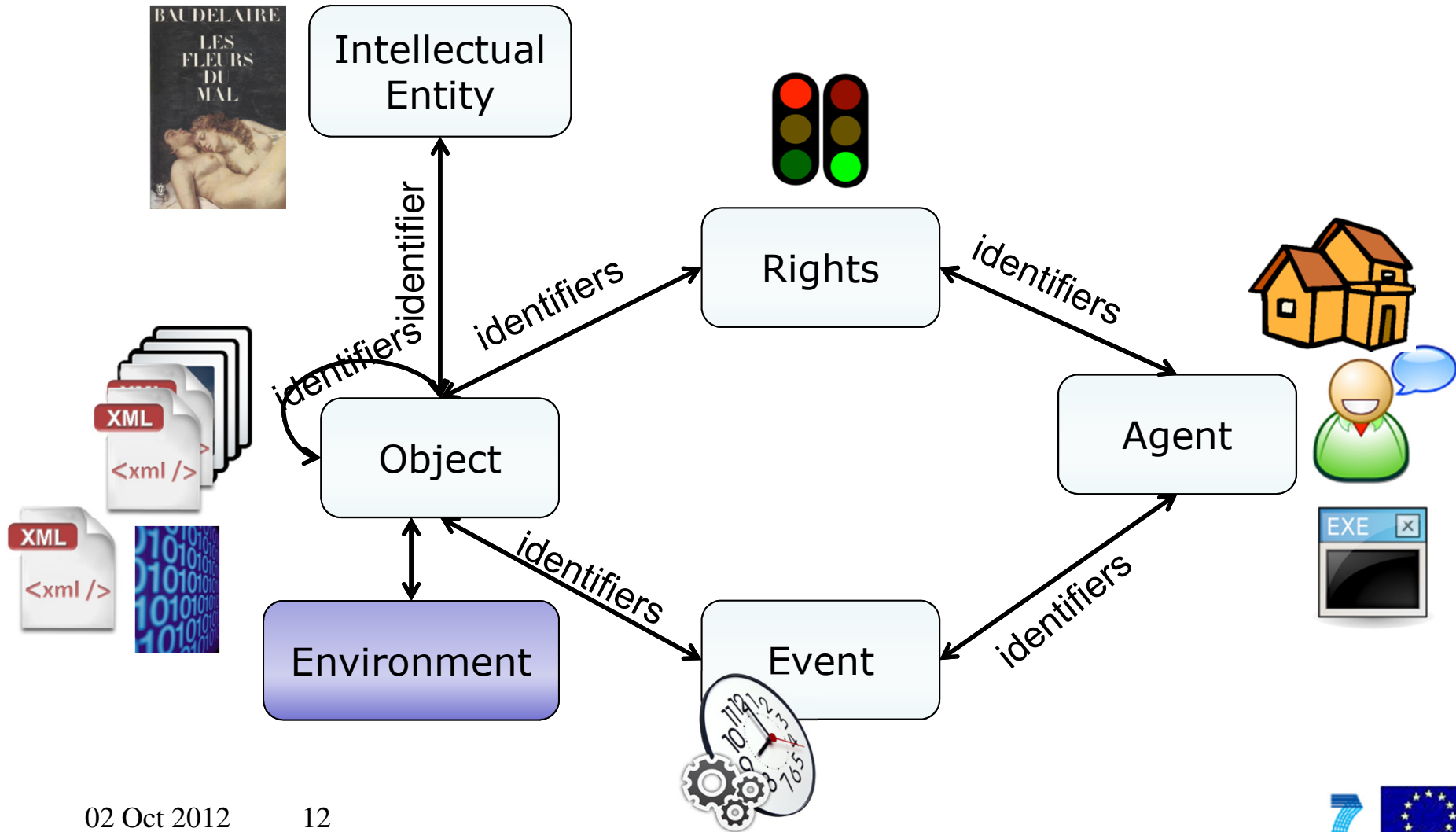
- 1. Remove IntellectualEntities as stand-alone entity
- 2. Rename IntellectualEntities -> IntellectualEntity
- 3. Add IntellectualEntity as Object type
- 4. Define semantic units -> as for Representation
- 5. Decide whether "environment" semantic unit is considered applicable to IntellectualEntity.
 - No HW and SW (technical) environments.
 - Default technical environment for all representations
 - Policy environments (such as in which reading room it should be accessible, etc.)

Required Changes

- 6. Update relationship:
structural relationships may be used to record logical containment (e.g. between an article and an issue)
- 7. Remove linkingIntellectualEntityIdentifier:
-> Use "relationship" instead.
- 8. Update definition of Object entity:
need not be related to any digital object.
- Not:
 - 9. Add a semantic unit to store its type. <div> TYPE attribute: article, monograph etc...
 - 10. Add a semantic unit to store the FRBR-level (works, expressions and manifestations) or archival categorizations.
=> use descriptive metadata instead

Environments

Slide by S. Peyrard



Environment

- Software
- Hardware
- A format
- A document
 - A policy document
 - A manual
 - Documentation
- A cheat sheet
- A user behaviour study
- A process
- “Other representation information”

Goal

- High-level data model for Environments**
- Capture the required relationships to other DP entities**
- Capture desirable characteristics**
- Standardized way of treating Environments**
- Information sharing / exchange**
- Repositories and registries**

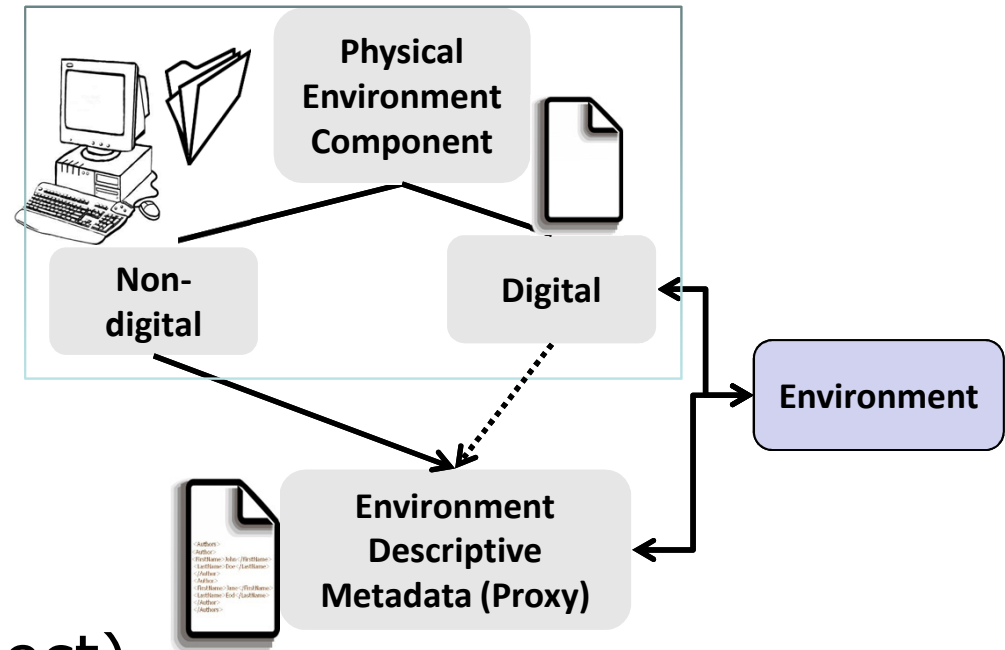
- Not:**
modelling the internals of a given Environment category – as e.g. TOTEM

Guidelines

- ❑ **Backward compatibility**
- ❑ **Compliance with OAIS**
- ❑ **Straightforward semantics**
 - ❖ **easy to implement**
- ❑ **Clear mapping of historic to new Environment features**
PREMIS 2 -> PREMIS 3

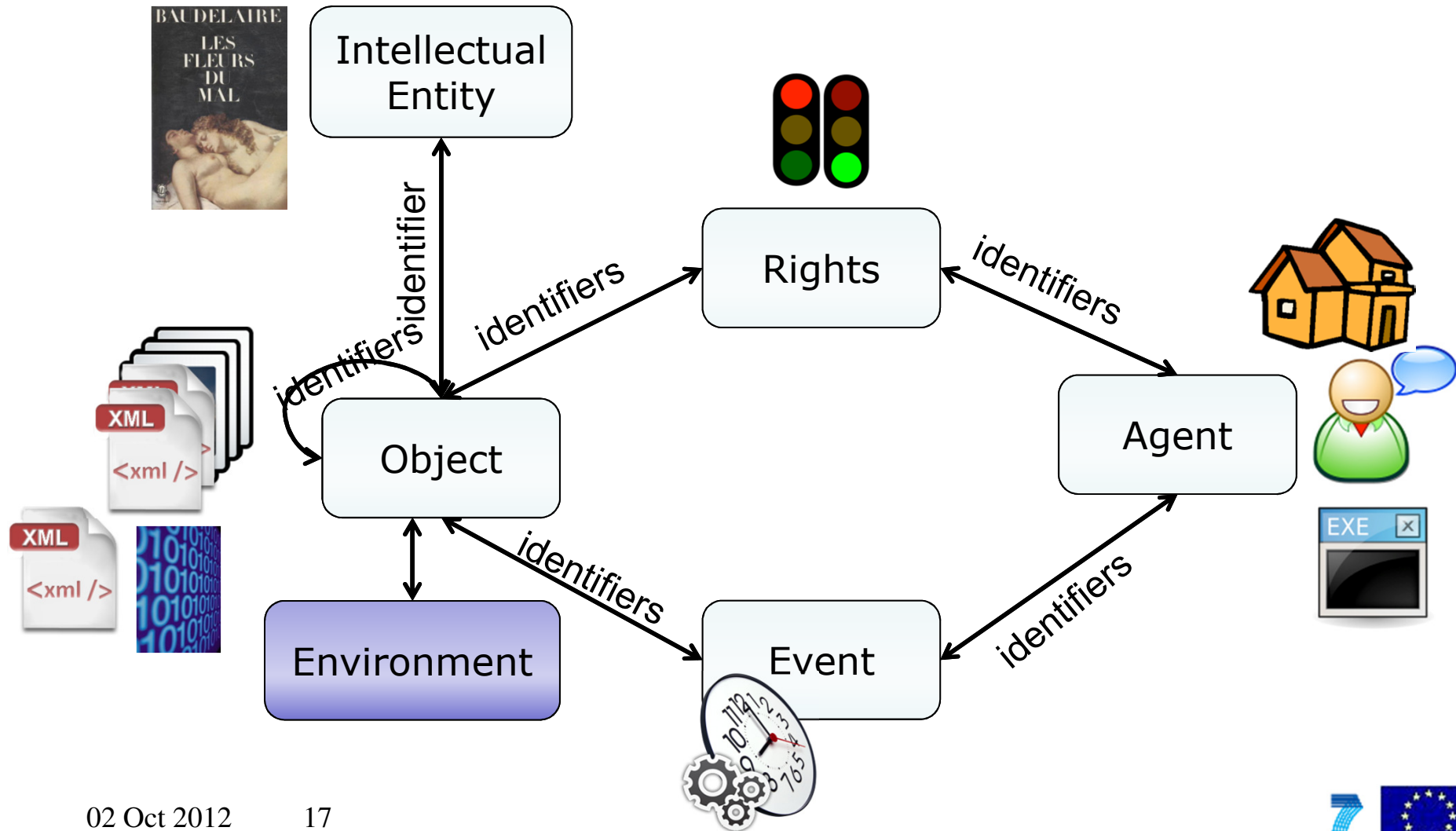
Requirements

- ❑ **Environments may be digital or non-digital**
- ❑ **Environments may be generic or instances**
(abstract description or concrete digital object)
- ❑ **Environments may be tools or services**
- ❑ **Environments have no simple software / hardware distinction**
(Virtual machines blur the distinction)



The PREMIS Data Model

Slide by S. Peyrard



Example: Object Entity

Main types of information

- ❖ identifier
- ❖ technical object characteristics
- ❖ creation information
- ❖ software and hardware environment
- ❖ digital signatures
- ❖ relationships to other objects
- ❖ links to other types of entity

PREMIS – Environment Metadata

1.5.5 creatingApplication

1.5.5.1 creatingApplicationName

1.5.5.2 creatingApplicationVersion

1.5.5.3 dateCreatedByApplication

1.5.5.4 creatingApplicationExtension

Gap Analysis

□ OAIS focus on Object:

- ❖ **Creating Applications are Environments**
- ❖ **Life-cycle view treating Environments uniformly**

Semantic Unit: Environment

- **What is needed to render or use an object**
 - ❖ **Operating system**
 - ❖ **Application software**
 - ❖ **Computing resources**

PREMIS – Environment Metadata

1.8 environment

- 1.8.1 environmentCharacteristic
- 1.8.2 environmentPurpose
- 1.8.3 environmentNote

1.8.4 dependency

- 1.8.4.1 dependencyName
- 1.8.4.2 dependencyIdentifier
 - 1.8.4.2.1 dependencyIdentifierType
 - 1.8.4.2.2 dependencyIdentifierValue

1.8.5 software

- 1.8.5.1 swName
- 1.8.5.2 swVersion
- 1.8.5.3 swType
- 1.8.5.4 swOtherInformation
- 1.8.5.5 swDependency

1.8.6 hardware

- 1.8.6.1 hwName
- 1.8.6.2 hwType
- 1.8.6.3 hwOtherInformation

- 1.8.7 environmentExtension

Environment Example: PDF File

environmentCharacteristic = known to work
environmentPurpose = render

hardware/hwName = Intel Pentium II
hardware/hwType = processor

dependency/dependencyName =
Mathematica 5.2
True Type math fonts

software/swName = Adobe Acrobat Reader
software/swVersion = 6.1
software/swType = renderer
software/swDependency = Windows NT

software/swName = Windows NT
software/swVersion = 5.0
software/swType = operatingSystem

Gap Analysis – Environment Subordinate to Object

❑ **Solution too specific**

Too complex to handle in an Object repository.

❑ **Solution too redundant**

Rarely specific to a single Object.

Redundancy results in

- Unnecessary verbosity;
- Cumbersome management of Environment descriptions as they evolve.

❑ **Unable to describe stand-alone Environments**

Independent of Objects -> Registry

Repositories and registries need to speak the same language

➤ Solution: Environment as first class entities

Gap Analysis – Scope

- ❑ **Primarily applicable to computing environments** (technical level).
- ❑ **No representation information in the broader sense.**
- ❑ **No explicit possibility to document the nature of dependencies** (e.g. operating systems to hardware).
- ❑ **No links to registry descriptions other than file formats.**
- ❑ **Specification of versions for software, but not for hardware.**

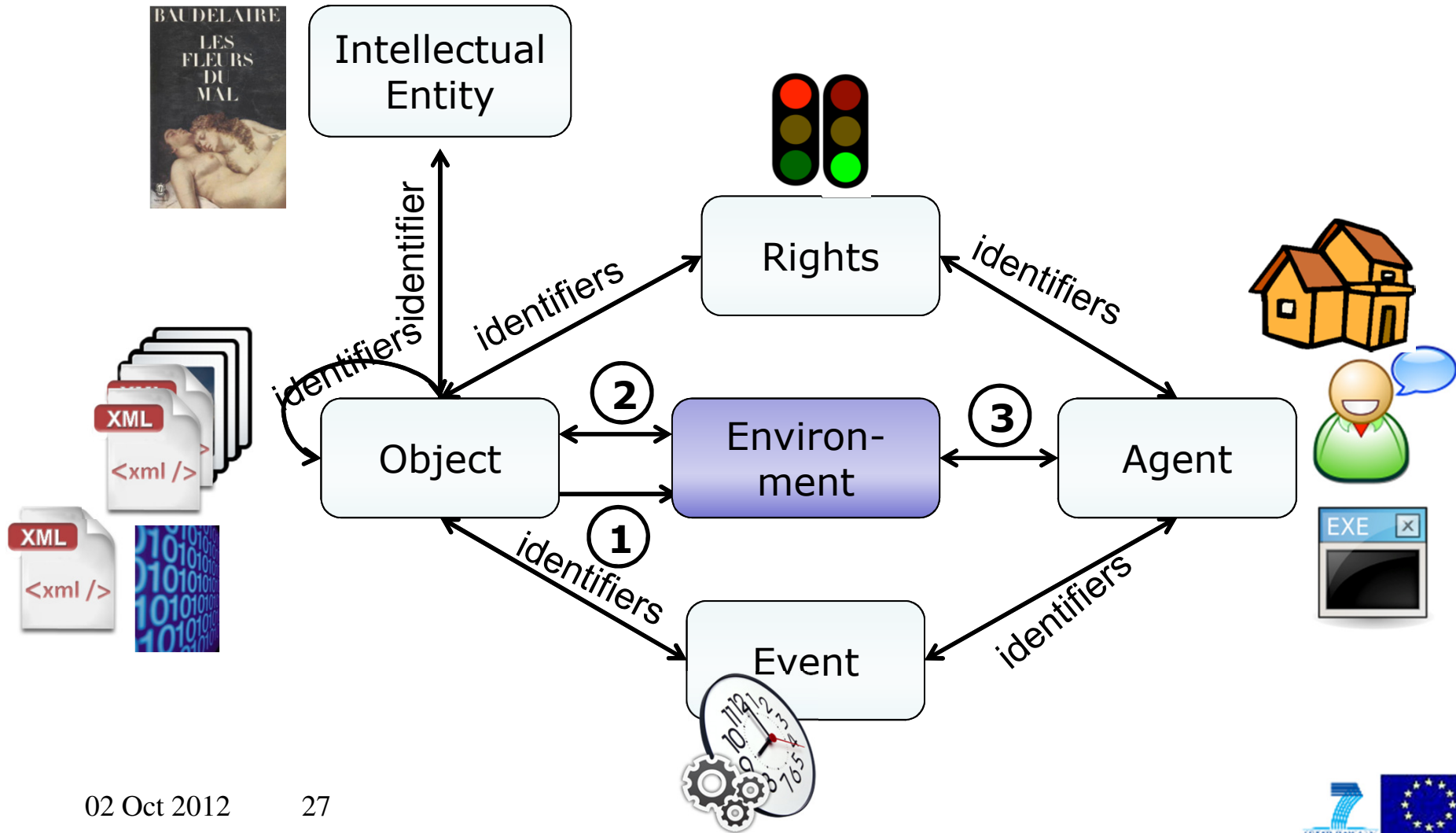
Gap Analysis - Relationships

❑ Not generic enough: Environments

- ❖ Can be related to Objects
- ❖ Can be Objects that need to be preserved
- ❖ Can be software Agents in an Events object

The PREMIS Data Model

Slide adapted from S. Peyrard



Gap Analysis - Relationships

❑ **Not generic enough: Environments**

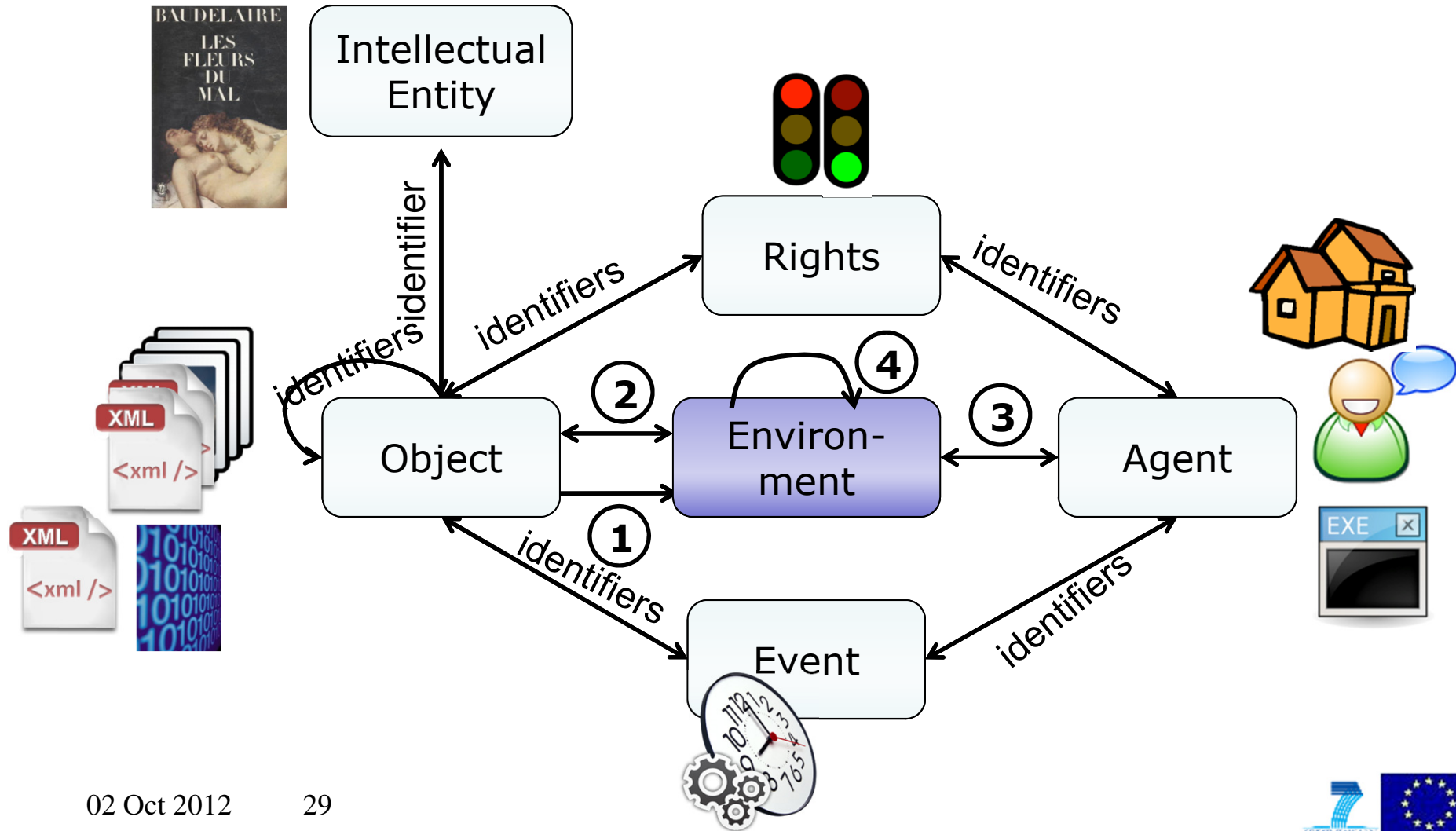
- ❖ Can be related to Objects
- ❖ Can be Objects that need to be preserved
- ❖ Can be software Agents in an Events object

❑ **Environments may link to other Environments**

- ❖ E.g. software application linking to its hardware platform

The PREMIS Data Model

Slide adapted from S. Peyrard



Gap Analysis - Relationships

❑ **Not generic enough: Environments**

- ❖ Can be related to Objects
- ❖ Can be Objects that need to be preserved
- ❖ Can be software Agents in an Events object

❑ **Environments may link to other Environments**

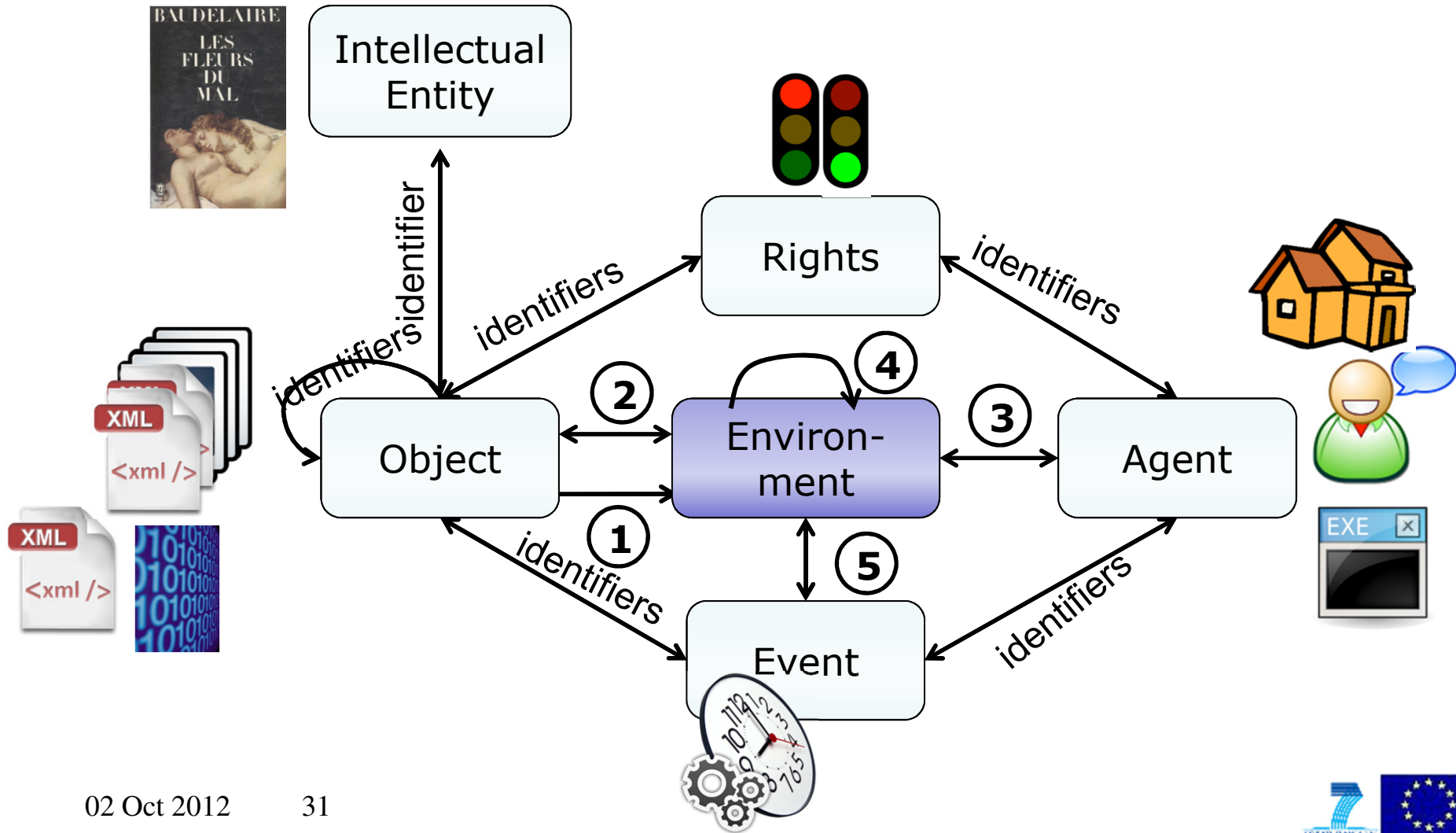
- ❖ E.g. software application linking to its hardware platform

❑ **Environments may link to Events**

- ❖ Creation, adding memory, ...
- ❖ Environments may need to be versioned

The PREMIS Data Model

Slide adapted from S. Peyrard



Requirements

- **Environment Type and SubType**

- **Environment Identifier**

- **Environment Designation**
(Name and Version)

- **Environment Registry**

- **Environment Storage**

all identical or generalizations

Requirements

- ❑ **Relationship to other Environment:**
structural, replacement, dependency,
generalization, reference, ...
- ❑ **Link to Object, Agent and Identifier**
- ❑ **Dependencies become a relationship type
(in addition to structural and derivation)**
- ❑ **Link from Object, Agent and Identifier to
Environment**

Thank you!