

# OAI-PMH Basics

Simeon Warner (Cornell University)

`simeon@cs.cornell.edu`



OAI5, CERN, Geneva, 18 April 2007.



# Schedule

9:00 **I. Introduction** (who we are / scope / objectives)

9:15 **II. OAI-PMH foundations**

- The basics of protocol basics
- Sets, datestamps, character encoding
- Hands-on trial using OAI-PMH requests

10:15 **III. Validation and compliance of an OAI data provider**

- Common problems / What to watch out for
- Validation services

10:45 **IV. Pointers to sharable metadata**

11:00 **V. Other topics and examples**

- Resource harvesting, mod\_oai, rights, discussion

11:30 **Close**

# Acknowledgments

- I have stolen slides from many friends and collaborators including: Carl Lagoze, Herbert Van de Sompel, Michael Nelson, and Tim Cole.
- This tutorial is an evolution from various tutorials given with the above.
- A disclaimer: I've been doing this for far too long so I may forget things that should be in an “OAI-PMH Basics” tutorial – please stop me if I go too fast, or if you can't understand my accent!

# Who are you?

- You signed up for a tutorial called “OAI Basics” so perhaps you want to know the basics of OAI?
- Implementing or deploying data-provider?
- Implementing or deploying harvester?
- Programming languages?
- Experience in metadata creation? dc/qdc, MARC flavors, METS, MODS, MAB, LOM, MPEG21 DIDL?
- Plan only to use Dublin Core in OAI? If so, why?
- Harvesting experience?
- XML, XSLT and/or W3C Schema experience?

# A very brief history

- Roots in e-print community (arXiv, cogprints, NCSTRL, RePEc)
- Idea: e-print servers will have greater impact if connected together → Sante Fe UPS meeting 1999
- Rapidly evolved into Open Archives Initiative (OAI) → technical committee → alpha phase → v1.0
- v1.1 revision as W3C XML schema spec. evolved
- Significant reworking, based on experience to create stable v2.0 (2002-06)
- New work on OAI-ORE, OAI-PMH to remain stable.

# Cross archive search

- This was one motivating example
- Bad experience with attempts at distributed search in the library community – nice in theory but troublesome in practice
- As storage became cheaper it seemed that it might be simpler to create union catalogues (aggregations of metadata) and then search locally (witness Google if you have scaling concerns)
- OAI provided a way to build that union catalogue
- Now OAIster etc., even Google doing some OAI harvesting

## II. OAI-PMH foundations

# OAI-PMH foundations

- Only talking about v2.0, not 1.x (pre-2002). v1.x is dead, don't accommodate it, move away from it.
- Reference

<http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm>

<http://www.openarchives.org/OAI/2.0/guidelines.htm>

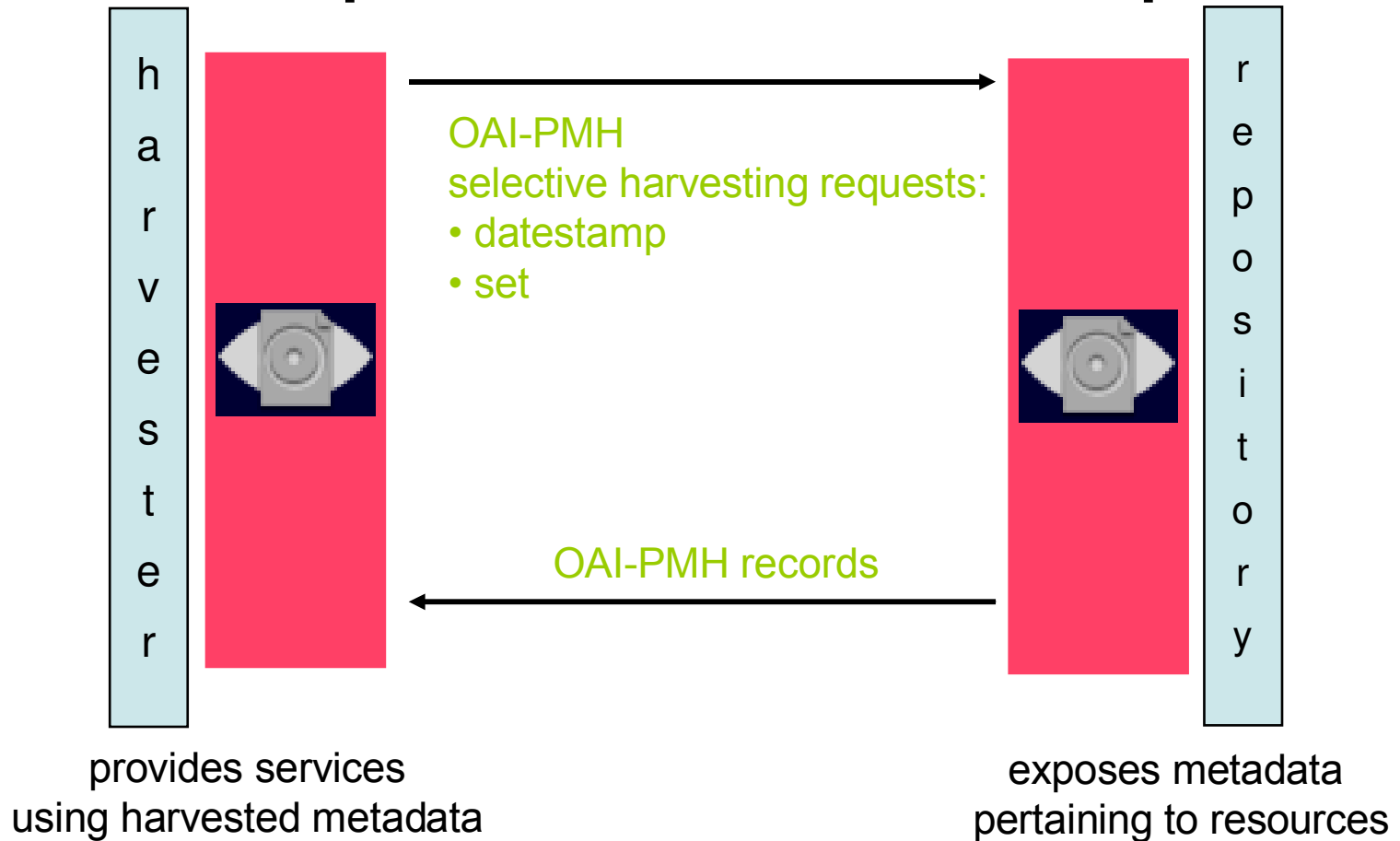
- Help:

[oai-implementers list](#) at

<http://www.openarchives.org/mailman/listinfo/oai-implementers>

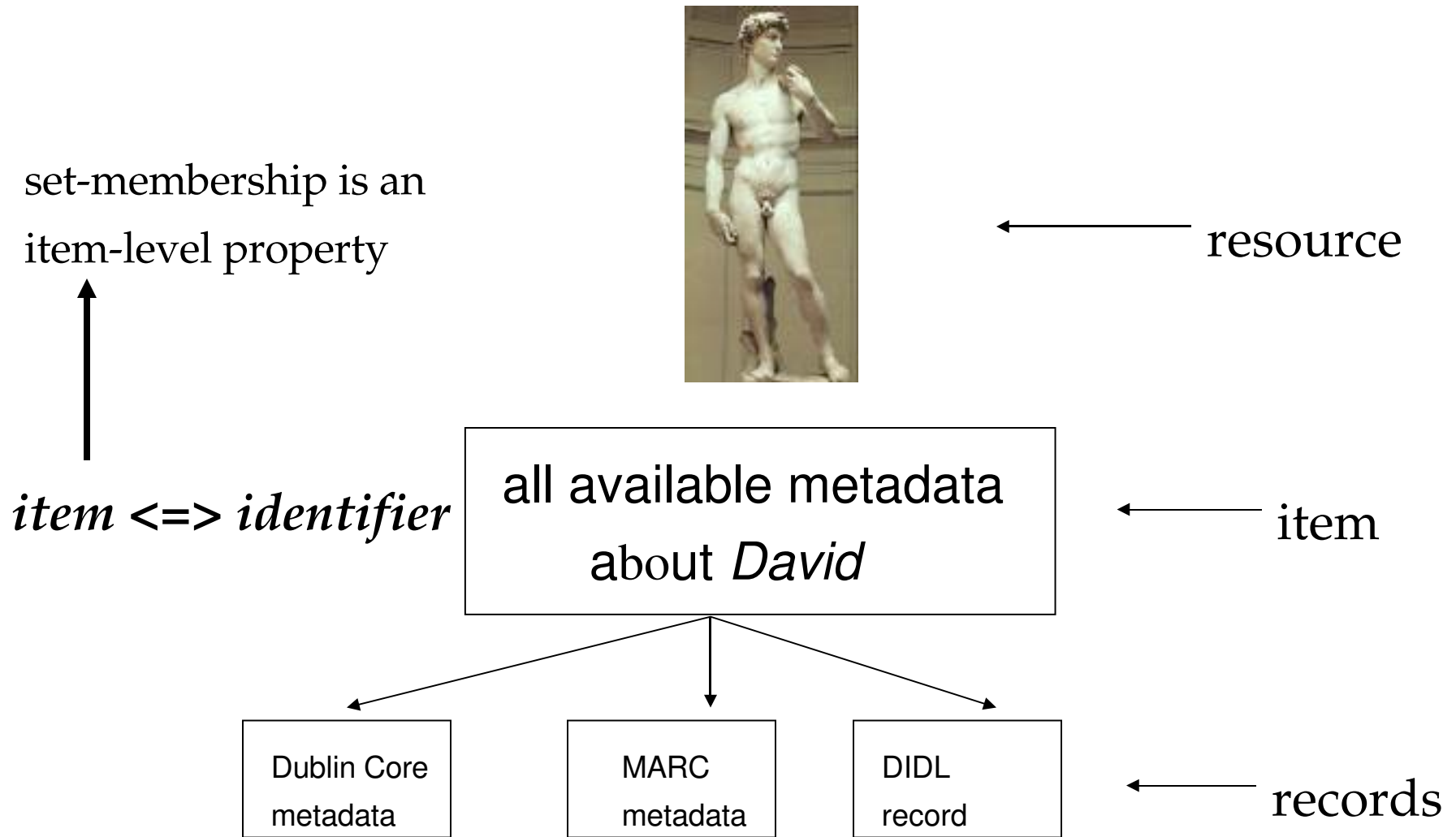


# Service-provider / Data-provider



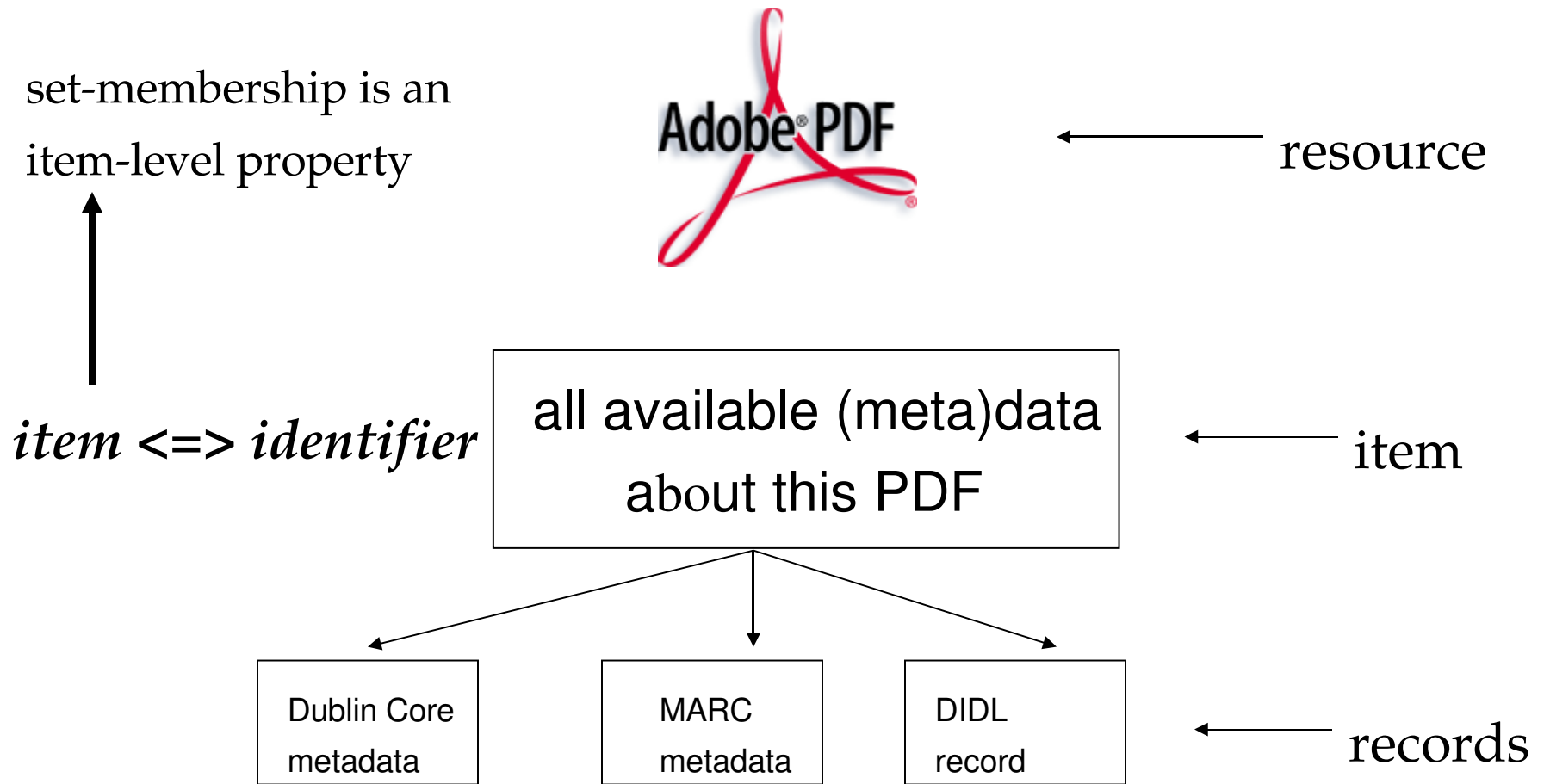
**OAI-PMH provides a way for a service-provider to efficiently keep an up-to-date copy of (some of) the metadata (or any XML data) exposed by a data-provider. Build services on top of this metadata.**

# Data model: **resource-item-record**



*record*  $\Leftrightarrow$  *identifier* + *metadataPrefix* + *datestamp*

# Data model: **resource-item-record**



*record*  $\Leftrightarrow$  *identifier* + *metadataPrefix* + *datestamp*

# Evolution of an “Item”

- Initial focus of Santa Fe convention was **an e-print**
- OAI-PMH v1.x assumed an item was a **document-like object**
- OAI-PMH v2.0 considers the item to be all information related to an arbitrary **resource** (which may or may not itself be electronic)

Compare with the reworked notions of the web architecture:

- Resource = something identified with URI

→ OAI-PMH Item, has identifier

- Representation = representation of a resource with a particular MIME type

→ OAI-PMH Record in a particular metadata format

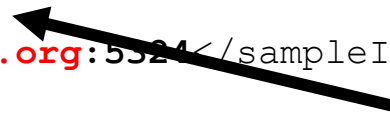
# Records and identifiers

- In OAI-PMH a record is uniquely identified *within a repository* by
  - ***identifier + metadataPrefix + timestamp***
- ***identifier*** here NOT the identifier of resource
  - resource identifier goes in metadata record
  - pick appropriate scheme to make globally unique (e.g. oai-identifier, info: URI, handle)
- ***metadataPrefix*** codes for a namespace, only oai\_dc can be assumed to tie globally
- ***timestamp*** is UTC time of last update in repository's granularity (globally meaningful)

# oai-identifier

- revision of **oai-identifier** from v1.x
- separate guidelines, both still used with OAI-PMH v2.0
- any new use of **oai-identifier** should use v2.0

```
<description>
  <oai-identifier xmlns="http://www.openarchives.org/OAI/2.0/oai-identifier"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai-identifier
    http://www.openarchives.org/OAI/2.0/oai-identifier.xsd">
    <scheme>oai</scheme>
    <repositoryIdentifier>oai-stuff.foo.org</repositoryIdentifier>
    <delimiter>:</delimiter>
    <sampleIdentifier>oai:oai-stuff.foo.org:5524</sampleIdentifier>
  </oai-identifier>
</description>
```



domain based  
repository  
identifiers

# Six verbs

	<b>Verb</b>	<b>Function</b>
metadata about the repository	Identify	Description of repository
	ListMetadataFormats	Metadata formats supported by repository
	ListSets	Sets defined by repository
harvesting verbs	ListIdentifiers	List OAI unique identifiers contained in repository
	ListRecords	List of many records
	GetRecord	List a single record

Most verbs take arguments: datestamps, sets, id, metadata format and resumption token (for flow control)

# Identify

- Arguments
  - none
- Errors
  - badArgument - if any argument is given

“Tell me about yourself..”



# ListMetadataFormats

- Arguments
  - identifier (OPTIONAL)
- Errors
  - badArgument - extra or unparsable arguments
  - noMetadataFormats - instead of empty reply
  - idDoesNotExist - more specific than just badArgument

“What metadata formats do you support? What internal names correspond to their XML namespaces?”

# ListSets

- Arguments
  - resumptionToken (EXCLUSIVE)
- Errors
  - badArgument
  - badResumptionToken
  - noSetHierarchy

“What sets are items organized in, if any? How are they identified and described?”

# ListIdentifiers

- Arguments
  - from (OPTIONAL), until (OPTIONAL), set (OPTIONAL)
  - resumptionToken (EXCLUSIVE)
  - metadataPrefix (REQUIRED)
- Errors
  - badArgument
  - cannotDisseminateFormat
  - badResumptionToken
  - noSetHierarchy
  - noRecordsMatch

“What are the identifiers (headers) of records available in this set/date-range/metadata format from this repository?”

# ListRecords

- Arguments

- from (OPTIONAL), until (OPTIONAL), set (OPTIONAL)
- resumptionToken (EXCLUSIVE)
- metadataPrefix (REQUIRED)

- Errors

- noRecordsMatch
- cannotDisseminateFormat
- badResumptionToken
- noSetHierarchy
- badArgument

“Give me all the records available in this set/date-range/metadata format from this repository”

# GetRecord

- Arguments
  - identifier (REQUIRED)
  - metadataPrefix (REQUIRED)
- Errors
  - badArgument
  - cannotDisseminateFormat
  - idDoesNotExist

“Give me this specific record from the given item  
in the requested format”

# Protocol vs periphery

- Protocol
  - Protocol document
  - oai\_dc
- Periphery
  - HTTP
  - XML
  - Other metadata formats (MARC, qdc, DIDL, METS...)
  - Extension schemas
  - Community guidelines

# OAI-PMH vs HTTP

- clear separation of OAI-PMH and HTTP
  - OAI-PMH error handling
    - all OK at HTTP level? => 200 OK
    - something wrong at OAI-PMH level? => OAI-PMH error (e.g. badVerb)
  - HTTP codes 302, 503, etc. still available to implementers, but they don't represent OAI-PMH events
- (except perhaps in baseURL terminology)

# Response with no errors

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH>
<responseDate>2002-02-08T08:55:46Z</responseDate>
<request verb="GetRecord"... ..>http://arxiv.org/oai2</request>
  <GetRecord>
    <record>
      <header>
        <identifier>oai:arXiv.org:cs/0112017</identifier>
        <datestamp>2001-12-14</datestamp>
        <setSpec>cs</setSpec>
        <setSpec>math</setSpec>
      </header>
      <metadata>
        . . .
      </metadata>
    </record>
  </GetRecord>
</OAI-PMH>
```

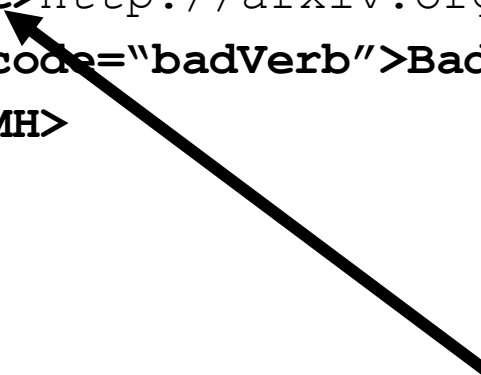
*Note no HTTP encoding  
of the OAI-PMH request*





# Response with error

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH>
<responseDate>2002-02-08T08:55:46Z</responseDate>
<request>http://arxiv.org/oai2</request>
<error code="badVerb">Bad verb. 'ShowMe' not implemented</error>
</OAI-PMH>
```



*In error case, only the correct  
attributes are echoed as attributes in  
<request>*

*Request was*

<http://arXiv.org/oai2?verb=ShowMe>

# Datestamp and granularity

- all dates/times are in UTC, encoded in ISO8601, in Z-notation:

1999-03-20T20:30:00Z

or just with year, month, day:

1999-03-20

Z means UTC = GMT



- harvesting granularity
  - mandatory support of YYYY-MM-DD
  - optional support of YYYY-MM-DDThh:mm:ssZ
  - granularity of `from` and `until` must be the same

# Set membership in header

The header contains the set membership of item

```
<record>
  <header>
    <identifier>oai:arXiv.org:cs/0112017</identifier>
    <datestamp>2001-12-14</datestamp>
    <setSpec>cs</setSpec>
    <setSpec>physics:hep-th</setSpec>
  </header>
  <metadata>
    . . .
  </metadata>
</record>
```

- Components of set path separated with colons [:]
- Super-sets do not need to be included, e.g. no **physics** if there is **physics:hep-th**

# ListIdentifiers

ListIdentifiers returns headers (should really have been called ListHeaders)

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH>
<responseDate>2002-02-08T08:55:46Z</responseDate>
<request verb="..." ...>http://arxiv.org/oai2</request>
<ListIdentifiers>
  <header>
    <identifier>oai:arxiv.org:hep-th/9801001</identifier>
    <datestamp>1999-02-23</datestamp>
    <setSpec>physics:hep-th</setSpec>
  </header>
  <header>
    <identifier>oai:arXiv.org:hep-th/9801002</identifier>
    <datestamp>1999-03-20</datestamp>
    <setSpec>physics:hep-th</setSpec>
    <setSpec>physics:hep-ex</setSpec>
  </header>
  .....
```

# metadataPrefix and setSpec

- The character set for **metadataPrefix** and **setSpec** is the following set of URL-safe characters:

A-Z a-z 0-9 - \_ . ! ~ \* ' ( )

(defined in the schema pattern match)

- This character set does not include % so URL encoding of other characters is not allowed. A number of repositories use URL encoding with the %. However, harvesters should treat **setSpec** as opaque (except for : semantics).

# Be honest with timestamps!

- A change in the process of dynamic generation of a metadata format that changes the output *really does mean all records have been updated!*
  - **If you get this wrong, updates will be missed by incremental harvests**
- Possible internal logic for updates to dynamic disseminations:

```
if (internalItemDatestamp >
    disseminationInterfaceDatestamp) {
    datestamp = internalItemDatestamp
} else {
    datestamp = disseminationInterfaceDatestamp
}
```

# Not hiding updates

- OAI-PMH is designed to allow incremental harvesting
- Updates must be available by the end of the period of the datestamp assigned, i.e.
  - Day granularity => during same day
  - Seconds granularity => during same second
- Reason: harvesters need to overlap requests by just one datestamp interval (one day or one second)

# resumptionToken

The only defined use of `resumptionToken` is as follows:

- a repository **must** include a `resumptionToken` element as part of each response that includes an incomplete list;
- in order to retrieve the next portion of the complete list, the next request **must** use the value of that `resumptionToken` element as the value of the `resumptionToken` argument of the request;
- the response containing the incomplete list that completes the list **must** include an empty `resumptionToken` element.



# State in resumptionTokens

- HTTP is stateless
- `resumptionTokens` allow state information to be passed back to the repository to create a complete list from sequence of incomplete lists:

EITHER – all state in `resumptionToken`

OR – cache result set in repository

(in which case there are limitations of expected lifetime of `resumptionToken`, Can express `expirationDate`.)

# All state in the resumptionToken

- Arrange that remaining items/headers in complete list response can be specified with a new query and encode that in `resumptionToken`
- One simple approach is to return items/headers in id order and make the new query specify the same parameters and the last id return (or by date)
  - simple to implement, but possibly inefficient
- Can encode parameters very simply:

```
<resumptionToken>metadataPrefix=oai_dc  
from=1999-02-03&until=2002-04-01&  
lastid=fghy:45:123</resumptionToken>
```

# Caching the result set

- Repository caches results of initial request, returns only incomplete list. This is the natural approach for DB based repositories.
- `resumptionToken` does not contain all state information, it includes:
  - a session id
  - offset information, necessary for idempotency
- `resumptionToken` allows repository to return next incomplete list
- increased complexity due to cache management
  - but a potential performance win

# resumptionToken & idempotency

- idempotency of List requests: return same incomplete list when `resumptionToken` is re-issued
  - while no changes occur in the repository: strict
  - while changes occur in the repository: all items with unchanged datestamp
- Means that harvester can recover from a bad transmission by repeating request at any point in a long response sequence

**IMPLICATION:** data-provider must accept both the most recent `resumptionToken` issued and the previous one.

# Flow control

How to respond to a harvester -- normal, too fast and problematic/bad:

1. HTTP status code 200; response to OAI-PMH request with a `resumptionToken`.
2. HTTP status code 503; with the `Retry-After` header set to an appropriate value if subsequent request follows too quickly or if the server is heavily loaded.
3. HTTP status code 403; with an appropriate reason specified if subsequent requests do not adhere to `Retry-After` delays.

# Error reporting

In general more detail is better...

```
<error code="badArgument">Illegal argument `foo' </error>
```

```
<error code="badArgument">Illegal argument `bar' </error>
```

is preferred over:

```
<error code="badArgument">Illegal arguments `foo',  
  `bar' </error>
```

which is preferred over:

```
<error code="badArgument">Illegal arguments</error>
```

# Scope of error reporting

- the OAI-PMH error / exception conditions are for OAI-PMH events (with semantics given in spec.)
- they are not for situations when:
  - the database is down
  - a record is malformed
    - remember: record = id + datestamp + metadataPrefix
    - if you're missing one of those, you don't have an OAI record!
  - and other conditions that occur outside the OAI scope
    - use HTTP codes 500, 503 or other appropriate values to indicate non-OAI-PMH problems

# Hands-on: OAI-PMH requests

- Pick a data-provider from the OAI registry:
  - <http://www.openarchives.org/Register/BrowseSites>
- Click on “Identify”, examine output, find `adminEmail`
- Edit the URL to replace `baseURL?verb=Identify` with
  - `baseURL?verb=ListMetadataFormats`
  - `baseURL?verb=ListSets`
  - `baseURL?verb=ListIdentifiers&metadataPrefix=oai_dc`
  - `baseURL?verb=GetRecord&metadataPrefix=oai_dc&identifier=<id>`
  - `baseURL?verb=ListRecords&metadataPrefix=oai_dc&from=2007-04-01&until=2007-04-18`



### III. Validation and compliance of an OAI data provider

# History of validation

- Validation service launched coincident with initial protocol release in 2001 (work of Donna Bergmark, Cornell)
- Updated with release of versions 1.1 and 2.0 (also by Donna Bergmark)
- Revamp to correct some problems in Jan 2004 (Simeon Warner)
- Continued corrections/additions as and when needed
- <http://www.openarchives.org/Register/ValidateSite>

# Registration

- Optional after validation (>600 sites, 2007-04-01)

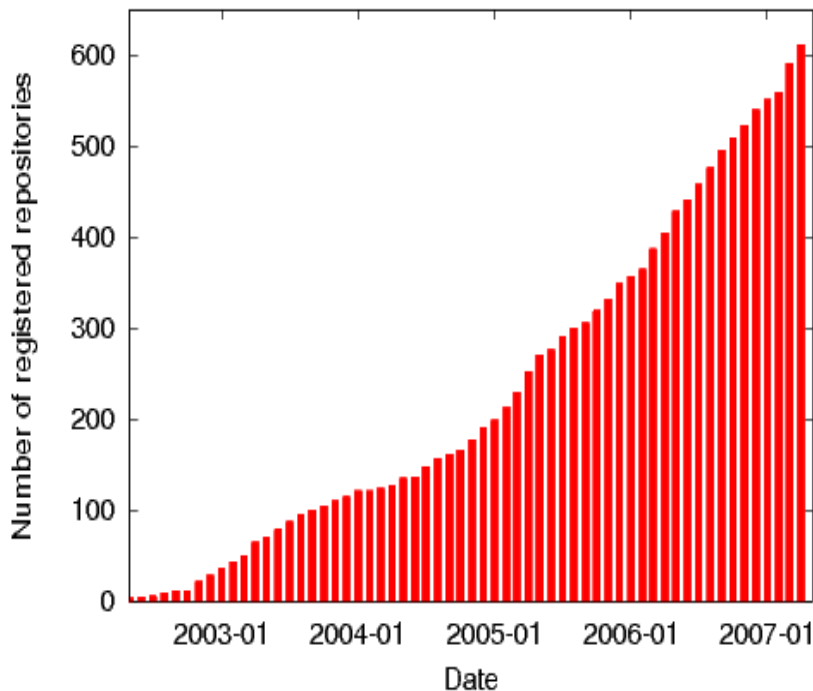
[http://www.openarchives.org/  
...org/Register/BrowseSites](http://www.openarchives.org/...org/Register/BrowseSites)

There are other registries, most notably the one run by Tom Habing at UIUC:

<http://gita.grainger.uiuc.edu/registry/>

also ROAR run by Tim Brody at Southampton:

<http://roar.eprints.org/>



# Step 1 – Identify response

- Fundamental to protocol, typically first request made by a harvester
- Check values needed by protocol
- Extract and check **adminEmail** used by validator
- Insist that **baseURL** returned in response is identical to that entered
- Email sent to **adminEmail** with code to continue, avoids DoS attack launched from openarchives site.

# Step 2 - the rest

- Get one response from each verb and validate XML against schema
- Check schema and namespace use, oai\_dc use
- Check use of datestamps in ListRecords
- Check responses to bad input conditions.
- Check correct use of resumptionToken (if used)
  
- INCOMPLETE TESTING -- under gradual improvement when a new problem case presents itself.

# Common problems (1)

- Analyzed validation 2004 logs for validator:  
<http://www.openarchives.org/Register/ValidateSite>  
(paper [arXiv:cs.DL/0506010](http://arxiv.org/abs/cs.DL/0506010) describes in more detail)
- 1893 requests with sensible baseURL
- 18% no Identify response
- 21% of cases returned invalid XML (Xerces output)
- 7% bad **adminEmail**, 0.3% bad protocol version
- 24% other errors with Identify -- usually quickly fixed
- 1% excessive (>5 in a row) 503 Retry-after
- 3% no identifiers from ListIdentifiers
- 2.5% no datestamp in sample record - fundamental problem!

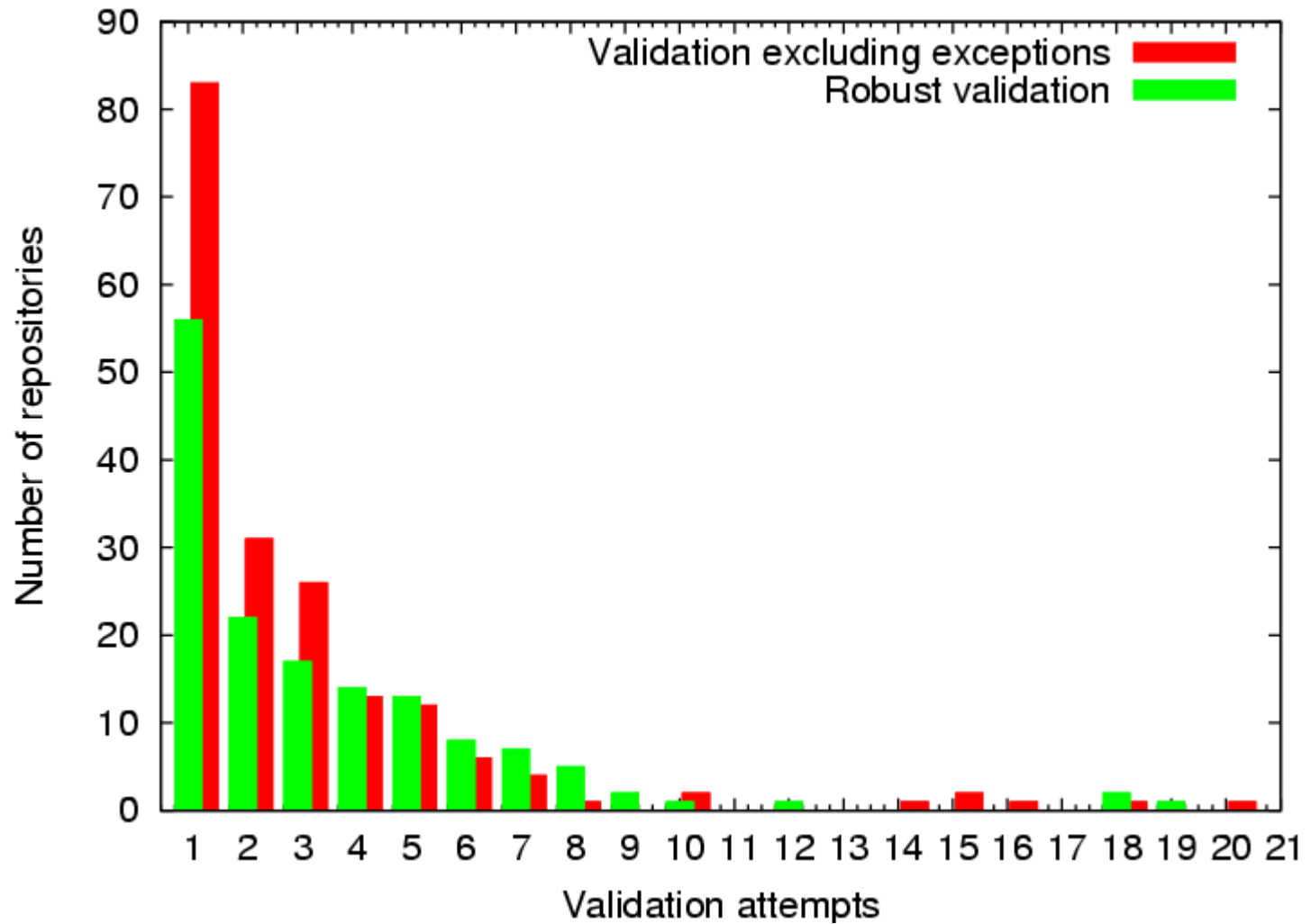
# Common problems (2)

- 927 completed validation requests
- 34% successful
- 22% errors in handling exception conditions
- 44% other (more serious) errors

Most common errors:

- ◆ Failed schema validation
- ◆ Empty response with known good **from** and **until**
- ◆ Empty **resumptionToken** to request without **resumptionToken**
- ◆ Malformed response if identifier is **invalid"i**d
- ◆ Granularity of **earliestDatestamp** doesn't match **granularity** value

# Validation attempts to success





# How hard was it to validate?

- 38% of cases successful first time (often deployments of standard software, e.g. eprints.org)
- Average of ~3 attempts/repository
- Ignore 238 sites with just one attempt (test sites?). Still 24 sites tried >5 times but never succeeded.
- 30% of those successful had errors in exception handling after otherwise OK.

# XML / Schema / Namespace

- Primary XML problem is character encoding (later...)
- OAI-PMH response must specify the correct namespaces and schemaLocations for the OAI-PMH schema and the oai\_dc schema, e.g.

```
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/  
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
```

and

```
<oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"  
  xmlns:dc="http://purl.org/dc/elements/1.1/"  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/  
    http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
```

(Hint: just copy from spec.)

- Use standard namespaces and schemas for other formats where possible

# Tricky datestamp and timezone

- One useful test is to check that a given header/record is returned when the **from** and **until** dates of a ListIdentifiers/ListRecords are set to its **datestamp**.
- Second most “popular” error after parsing failures.
- Usually quickly corrected.
- (One then unsolved case with a DSpace instance in Australia, operating in a timezone with a half-hour offset from UTC/GMT. The **from** and **until** must be set half a hour off to get the correct record, clearly broken!)

# identifier=invalid" id

- The most common responses to this input condition are:
  1. invalid XML returned
  2. 500 server error
- Particularly troubling as these cases imply
  1. lack of systematic parameter checking (should have checks at least as strict as OAI spec, perhaps more so to limit to local context)
  2. lack of systematic output encoding (plain " can't go in an XML attribute even if one mistakenly wants to include it, use &quot; instead)
- Such failures are asking for trouble!

# XML character encoding (1)

**YOU MUST GET IT RIGHT - NO EXCUSES!**

The whole XML framework falls apart if you don't have valid character encodings, harvesters **will** fail.

- OAI-PMH mandates UTF-8.
- UTF-8 is an encoding of Unicode where code points (characters) above 127 are encoded using multi-byte sequences.
- The code points for Latin-1 are identical in Unicode but those above 127 must have special encoding.
- Non ASCII (>127) characters must use either **multi-byte sequences (UTF8)** or **numeric entities**:  
e.g. decimal `&#241;`; hex `&#xF1;` (NO `&ntilde;` for ñ)

# XML character encoding (2)

- Allowed code points for XML1.0 (XML1.1 slightly different)  
#x9 | #xA | #xD | [#x20-#xD7FF]  
[#xE000-#xFFFFD] | [#x10000-#x10FFFF]
- These restrictions are tighter than plain Unicode/UTF8 restrictions. For example, including either character 15 or the numeric entity &#xF; will give illegal XML since the numeric entities are decoded before parsing.
- **BOTTOM LINE: Anyone implementing an OAI-PMH data-provider should make illegal responses impossible, irrespective of the input data. Should probably report internal problems to administrator.**

# Debugging UTF-8 encodings

- One option is a small program I wrote (and have used to help many data-providers) --

**utf8conditioner**

<http://www.cs.cornell.edu/people/simeon/software/utf8conditioner/>

(Current version [2005-10-25] checks UTF-8 with XML restrictions and tests numeric entities)

- NSDL harvester uses this code to attempt to clean responses that cannot be parsed
- There is also a Java port from the Kopal project, koLibRI:  
[http://kopal.langzeitarchivierung.de/index\\_koLibRI.php.en](http://kopal.langzeitarchivierung.de/index_koLibRI.php.en)
- This is the real nitty-gritty of encodings, best to avoid it all by carefully using existing implementations and/or good libraries to start with.

# utf8conditioner (-x)

Example output run on test/testfile with -x flag (output in red):

```
01: $Id: testfile,v 1.3 2001/08/01 20:59:43 simeon Exp $
02: Test file for utf8conditioner, Simeon Warner 1Aug2001
03: 0xXX are the hex values of the bytes that follow
04: -----
05: valid 2 byte (0xCF 0x8F) <CF><8F>
06: valid 3 byte (0xEF 0x8F 0x8F) <EF><8F><8F>
Line 7, char 323, byte 326: byte 2 isn't continuation: 0xCF 0x61,
  restart at 0x61, substituted 0x3F
07: invalid 2 byte (0xCF a) ?a
Line 8, char 359, byte 363: byte 3 isn't continuation: 0xEF 0x81,
  0x61, restart at 0x61, substituted 0x3F
08: invalid 3 byte (0xEF 0x81 a) ?a
Line 9, char 395, byte 399: illegal byte: 0xB0, substituted 0x3F
09: illegal byte in UTF-8 (0xB0) ?
Line 10, char 428, byte 432: code not allowed in XML1.0: 0x000B,
  substituted 0x3F
10: not allowed in XML (0x0B) ?
11: bye
```



# IV. Pointers to sharable metadata

# OAI Best Practices effort

- OAI Best Practices effort was organized by the Digital Library Federation (DLF) and the National Science Digital Library (NSDL), starting 2004.

- All work available at

<http://oai-best.comm.nsdlib.org/cgi-bin/wiki.pl?TableOfContents>

- Has guidelines in a number of areas (data provider competencies and best practices, tools and extensions) but area least covered elsewhere is:

- Best practices for sharable metadata

<http://oai-best.comm.nsdlib.org/cgi-bin/wiki.pl?PublicTOC>

# What is sharable metadata?

- By “sharable metadata” we mean metadata that is useful when shared. Metadata that one can hope to harvest via the OAI-PMH and build services on without needing to know too many details of the particular system it came from.
- The OAI Best Practices recites text from Bruce and Hillman which outlines what makes “quality metadata” and then refines that to “sharable metadata”. I quote this on the next four slides.

# Quality metadata (1)

- **Completeness.** Two aspects of this characteristic are described: choosing an element set allowing the resources in question to be described as completely as is economically feasible, and applying that element set as completely as possible.
- **Accuracy.** This characteristic is defined as the metadata being correct and factual, and conforming to syntax of the element set in use.
- **Provenance.** Here provenance refers to the provision of information about the expertise of the person(s) creating the original metadata, and its transformation history.
- **Conformance** to expectations. Metadata elements, use of controlled vocabularies, and robustness should match the expectations of a particular community. This aspect of metadata quality is particularly problematic for OAI data providers, as sharing metadata via OAI allows it to be used by a wider variety of communities than previously targeted.

[Bruce and Hillmann, see full reference next page]

# Quality metadata (2)

- **Logical consistency and coherence.** This characteristic is defined as element usage matching standard definitions, and consistent application of these elements.
- **Timeliness.** Two concepts make up this characteristic of metadata quality. Currency refers to metadata keeping up with changes to the resource it describes. Lag refers to a resources availability preceding the availability of its metadata.
- **Accessibility.** Proper association of metadata with the resource it describes and readability by target users contribute to this characteristic.

[“The Continuum of Metadata Quality”, in the book *Metadata in Practice*, ed. Diane I. Hillmann and Elaine L. Westbrook, Chicago: American Library Association, 2004, Thomas R. Bruce and Diane I. Hillmann]

# Sharable metadata (1)

In addition to “quality”, add:

- **Proper context.** In a shared environment, metadata records will become separated from any high-level context applying to all records in a group, and from other records presented together in a local environment. It is therefore essential that each record contain the context necessary for understanding the resource the record describes, without relying on outside information.
- **Content coherence.** Metadata records for a shared environment need to contain enough information such that the record makes sense standing on its own, yet exclude information that only makes sense in a local environment. This can be described as sharing a 'view' of the native metadata.
- **Use of standard vocabularies.** The use of standard vocabularies enables the better integration of metadata records from one source with records from other sources.

[Bruce and Hillmann, see full reference next page]

# Sharable metadata (2)

- **Consistency.** Even high-quality metadata will vary somewhat among metadata creators. All decisions made about application of elements, syntax of metadata values, and usage of controlled vocabularies, should be consistent within an identifiable set of metadata records so those using this metadata can apply any necessary transformation steps without having to process inconsistencies within such a set.
- **Technical conformance.** Metadata should conform to the specified XML schemas and should be properly encoded.

[“The Continuum of Metadata Quality”, in the book *Metadata in Practice*, ed. Diane I. Hillmann and Elaine L. Westbrook, Chicago: American Library Association, 2004, Thomas R. Bruce and Diane I. Hillmann]

# Sharable metadata examples

- **Context:** the classic example is a picture of Theodore Roosevelt described as “On a horse”. This is not very helpful outside the context of the collection of pictures of Roosevelt.
- **Dates:** expose dates that are meaningful for discovers. Use standard formats (e.g. ISO8601) and exclude non-parsable information. i.e.  
`<dc:date>2005-01-01</dc:date>`  
and not  
`<dc:date>Created  
1Jan2005</dc:date>`



V. Others topics and examples

# V.1 Resource harvesting

[ slides from Herbert Van de Sompel, see also:

“Resource Harvesting within the OAI-PMH Framework”,

Herbert Van de Sompel, Michael L. Nelson, Carl Lagoze, Simeon Warner

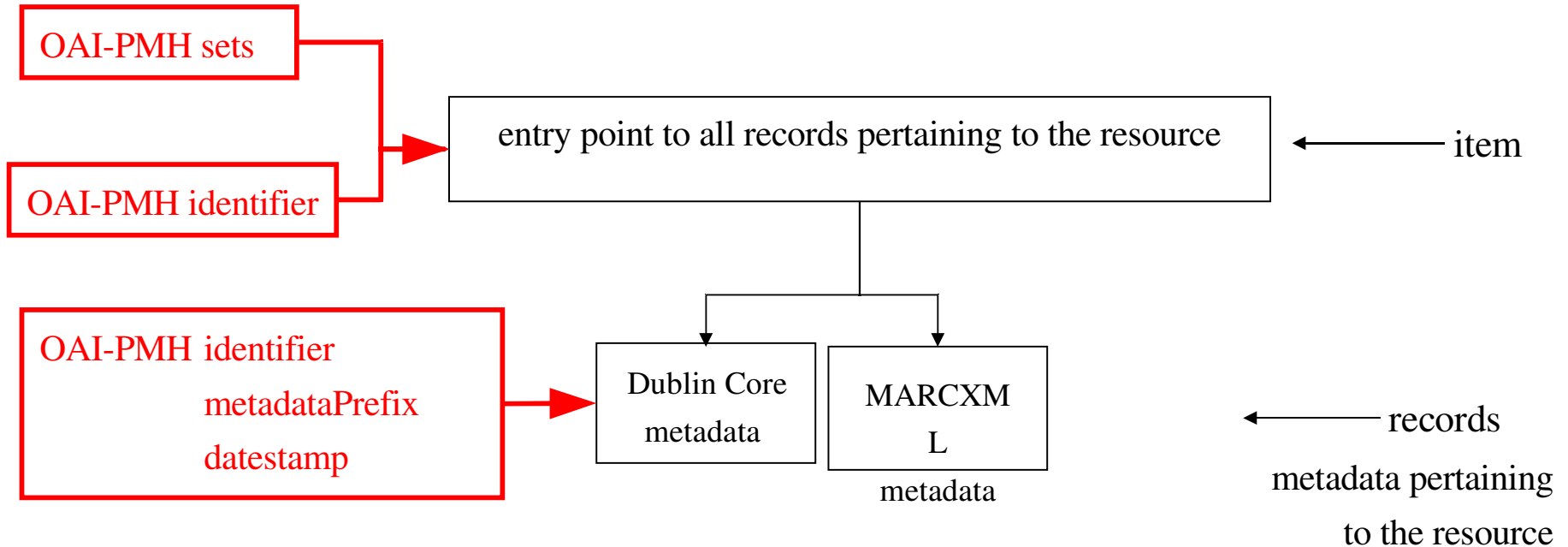
<http://www.dlib.org/dlib/december04/vandesompel/12vandesompel.htm>

]

# OAI-PMH data model



← resource



# Discovery use case

**Discovery:** use content itself in the creation of services

- search engines that make full-text searchable
- citation indexing systems that extract references from the full-text content
- browsing interfaces that include thumbnail versions of high-quality images from cultural heritage collections

## **Examples**

- Institutional Repository & Digital Library Projects: UK JISC, DARE, DINI
- Web search engines: competition for content (c.f. Google Scholar)

# Preservation use case

## **Preservation:**

- periodically transfer digital content from a data repository to one or more trusted digital repositories
- trusted digital repositories need a mechanism to automatically synchronize with the originating data repository

## **Examples:**

- Institutional Repository & Digital Library Projects: UK JISC, DARE, DINI
- Library of Congress NDIPP Archive Export/Ingest

# Various OAI-PMH based approaches

Typical scenario:

1. An OAI-PMH harvester harvests Dublin Core records from the OAI-PMH repository.
2. The harvester analyzes each Dublin Core record, extracting dc.identifier information in order to determine the network location of the described resource.
3. A separate process, out-of-band from the OAI-PMH, collects the described resource from its network location.

# Various OAI-PMH based approaches : Issue 1

- Locate resource based on information provided in dc.identifier
  - dc.identifier used to convey a variety of identifier: (simultaneously) URL DOI, bibliographic citation, ... Not expressive enough to distinguish between identifier, locator.
    - Several dereferencing attempts required
  - URI provided in dc.identifier is commonly that of a bibliographic “splash page”
    - How to know it is a bibliographic “splash page”, not the resource?
    - If it is a bibliographic “splash page”, where is the resource?

# Various OAI-PMH based approaches : Issue 2

- Using the OAI-PMH datestamp of the Dublin Core record to trigger incremental harvesting:
  - Datestamp of DC record does not necessarily change when resource changes

	DC record datestamp no change	DC record datestamp change
	no metadata update	metadata update
no resource update	OK	unnecessary resource download
resource update	missed resource update	OK



# Various OAI-PMH based approaches : Conventions

- Conventions address Issue 1; Issue 2 can not really be addressed.
- First dc.identifier is location of the resource
  - what if the resource is not digital?
- Use of dc.format and/or dc.relation to convey location

# Various OAI-PMH based approaches : Conventions


```
<oai_dc:dc>
  <dc:title>A Simple Parallel-Plate Resonator Technique for Microwave.
    Characterization of Thin Resistive Films</dc:title>
  <dc:creator>Vorobiev, A.</dc:creator>
  <dc:subject>ING-INF/01 Elettronica</dc:subject>
  <dc:description>A parallel-plate resonator method is proposed for
    non-destructive characterisation of resistive films used in
    microwave integrated circuits. A slot made in one ... </dc:description>
  <dc:publisher>Microwave engineering Europe</dc:publisher>
  <dc:date>2002</dc:date>
  <dc:type>Documento relativo ad una Conferenza o altro Evento</dc:type>
  <dc:type>PeerReviewed</dc:type>
  <dc:identifier>http://amsacta.cib.unibo.it/archive/00000014/</dc:identifier>
  <dc:format>pdf
    http://amsacta.cib.unibo.it/archive/00000014/01/GaAs_1_Vorobiev.pdf
  </dc:format>
</oai_dc:dc>
```

splash page

location of resource

# Various OAI-PMH based approaches : Conventions

```
...  
<dc:identifier>http://amsacta.cib.unibo.it/archive/00000014/</dc:identifier>  
<dc:relation>  
  http://amsacta.cib.unibo.it/archive/00000014/01/GaAs_1_Vorobiev.pdf  
</dc:relation>  
...
```

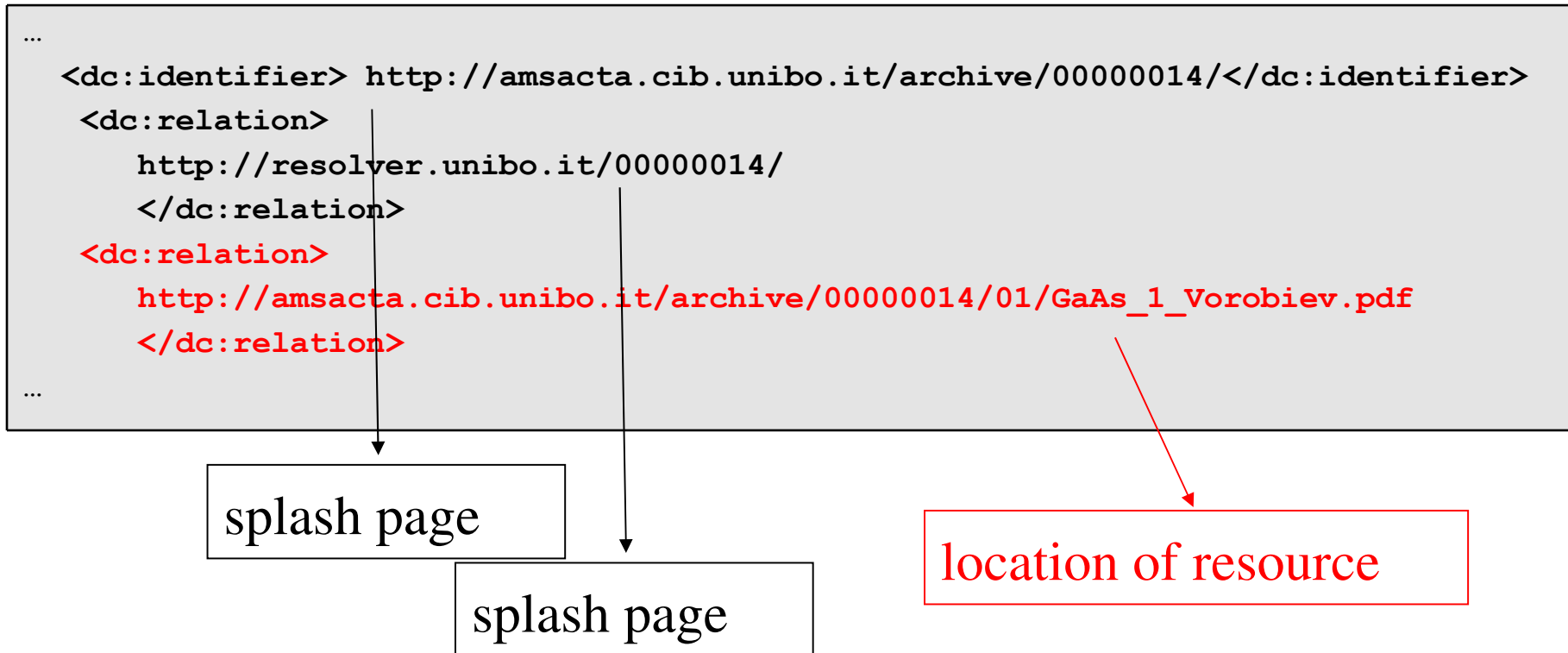


splash page



location of resource

# Various OAI-PMH based approaches : Conventions



# Various OAI-PMH based approaches : Other attempts

- dc.identifier leads to splash page & splash page contains special purpose XHTML link to resource(s)
  - What if there is no splash page?
  - How does a harvester know he is in this situation?
- OA-X: protocol extension
  - OK in local context, strategic problem to generalize
  - How to consolidate with OAI-PMH data model
  - Relies upon Qualified Dublin Core
- Could bring expressiveness to distinguish between location & identifier
  - But what with datestamp issue?

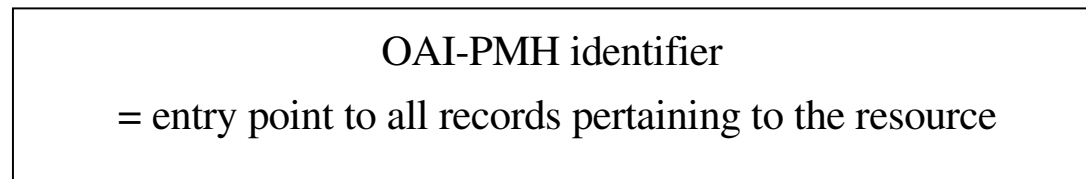
# An approach within OAI-PMH

- Use “metadata” formats that were specifically created for representation of digital objects:
  - Complex Object Formats as OAI-PMH metadata formats
  - MPEG-21 DIDL, METS, ..

# OAI-PMH data model

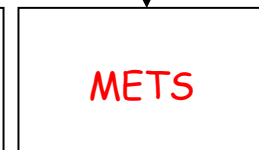
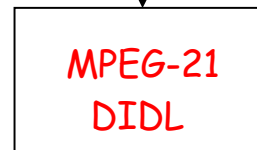
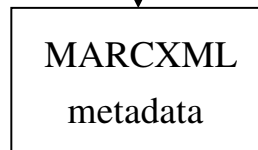
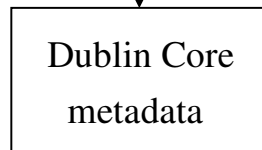


← resource



← item

metadata pertaining  
to the resource



← records

simple

more  
expressive

highly  
expressive

highly  
expressive

# Complex Object Formats : characteristics

- Representation of a digital object by means of a wrapper XML document.
- Represented resource can be:
  - simple digital object (consisting of a single datastream)
  - compound digital object (consisting of multiple datastreams)
- Unambiguous approach to convey identifiers of the digital object and its constituent datastreams.
- Include datastream:
  - By-Value: embedding of base64-encoded datastream
  - By-Reference: embedding network location of the datastream
  - not mutually exclusive; equivalent
- Include a variety of secondary information
  - By-Value
  - By-Reference
  - Descriptive metadata, rights information, technical metadata, ...



```
<didl:DIDL>
<didl:Item>
  <didl:Descriptor><didl:Statement mimeType="text/xml; charset=UTF-8">
    <dii:Identifier>
      http://amsacta.cib.unibo.it/archive/00000014/
    </dii:Identifier>
  </didl:Statement></didl:Descriptor>
  <didl:Descriptor><didl:Statement mimeType="text/xml; charset=UTF-8">
    <oai_dc:dc>
      <dc:title>A Simple Parallel-Plate Resonator Technique for
        Microwave. Characterization of Thin Resistive Films
      </dc:title>
      <dc:creator>Vorobiev, A.</dc:creator>
      <dc:identifier>
        http://amsacta.cib.unibo.it/archive/00000014/</dc:identifier>
      <dc:format>application/pdf</dc:format>
      ...
    </oai_dc:dc>
  </didl:Statement></didl:Descriptor>
  <didl:Component>
    <didl:Resource mimeType="application/pdf"
      ref="http://amsacta.cib.unibo.it/archive/00000014/01/GaAs_1_Vorobiev.pdf"/>
  </didl:Component>
</didl:Item>
</didl:DIDL>
```

# Complex Object Formats & OAI-PMH

- Resource represented via XML wrapper => OAI-PMH  
**<metadata>**
- Uniform solution for simple & compound objects
- Unambiguous expression of locator of datastream
- Disambiguation between locators & identifiers
- OAI-PMH datestamp changes whenever the resource (datastreams & secondary information) changes
- OAI-PMH semantics apply: “about” containers, set membership

# OAI-PMH based approach using Complex Object Format

Typical scenario:

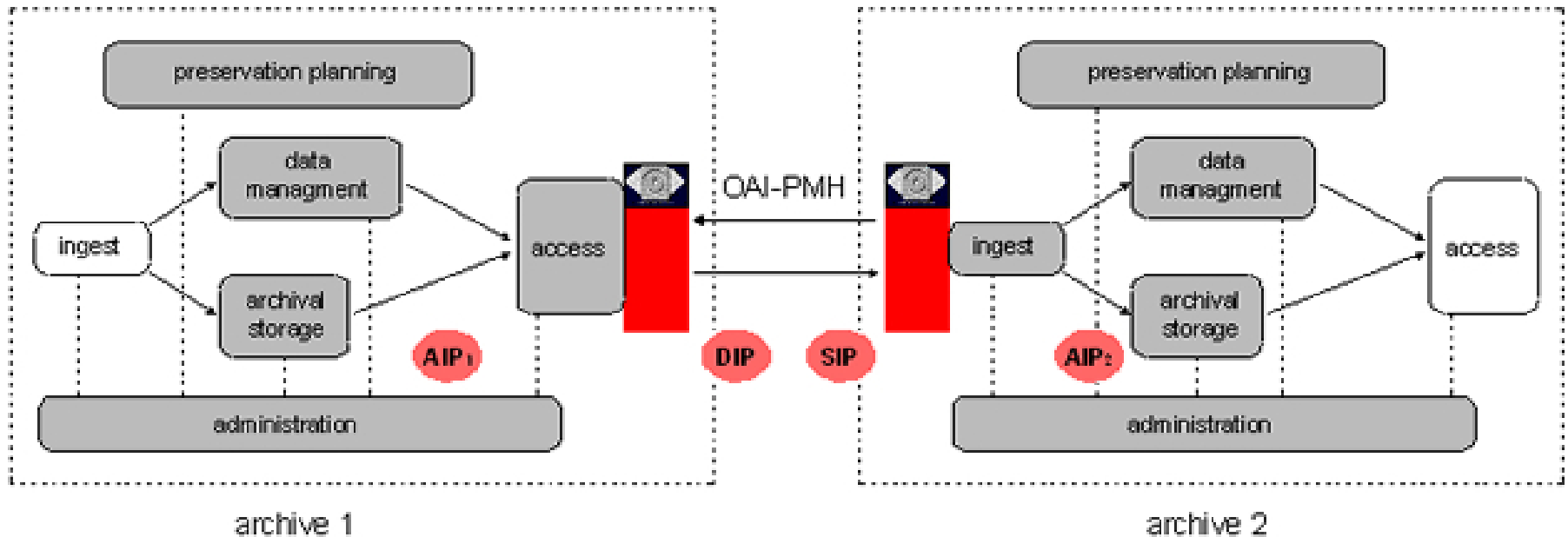
1. An OAI-PMH harvester checks for support of a locally understood complex object format using the ListMetadataFormats verb
2. The harvester harvests the complex object metadata. Semantics of the OAI-PMH datestamp guarantee that new and modified resources are detected.
3. A parser at the end of the harvesting application analyzes each harvested complex object record:
  - The parser extracts the bitstreams that were delivered By-Value.
  - The parser extracts the unambiguous references to the network location of bitstreams delivered By-Reference.
4. A separate process, out-of-band from the OAI-PMH, collects the bitstreams delivered By-Reference from the extracted network locations.

# Complex Object Formats & OAI-PMH : existing implementations

- LANL Repository
  - Assets stored as MPEG-21 DIDL documents
  - DIDL documents made accessible to downstream applications via the OAI-PMH
- Mirroring of American Physical Society collection at LANL
  - Maps APS document model to MPEG-21 DIDL Transfer Profile
  - Exposes MPEG-21 DIDL documents through OAI-PMH infrastructure
  - Includes digests/signatures
- DSpace & Fedora plug-ins
  - Maps DSpace/Fedora document model to MPEG-21 DIDL Transfer Profile
  - Exposes MPEG-21 DIDL documents through OAI-PMH infrastructure
- mod\_oai

# Complex Object Formats & OAI-PMH

## PMH : archive export/ingest



# Complex Object Formats & OAI-PMH : issues

- Which Complex Object Format(s)
- How to Profile Complex Object Format(s) for OAI-PMH Harvesting
- Large records
- Making resources re-harvestable
- Because the resource is represented as **<metadata>**, can rights pertaining to the resource be expressed according to the “rights for metadata” OAI-rights guideline?
- Tools:
  - Software library to write compliant complex objects
  - Integration of this library with repository systems (Fedora, DSpace, eprints.org, ....)

# V.2 OAI Rights

# Why OAI-rights?

OAI has matured beyond e-prints and is used to convey metadata about resources for which the ability to express rights is a factor limiting dissemination

⇒ Encourage participation by allowing assertion of rights and restrictions

Even in the open access world it may be important to express permissions

⇒ Work inspired by the RoMEO project (Oppenheim, Proberts, Gadd, 2002-2003)



# How?

“The usual OAI way”:

- Assemble group of knowledgeable and interested parties (the OAI-rights group)
- Distribute first-stab white paper
- Discuss via conference call, scope work
- Email and conference call discussions, develop alpha specification (Jun 2004), revise
- Release beta specification (Nov 2004)
- Release specification (May 2005)

<http://www.openarchives.org/OAI/2.0/guidelines-rights.htm>

# Who?

## The OAI-rights group:

**Caroline Arms** (Library of Congress), **Chris Barlas** (Rightscom), **Tim Cole** (University of Illinois at Urbana-Champaign), **Mark Doyle** (American Physical Society), **Henk Ellerman** (Erasmus Electronic Publishing Initiative), **John Erickson** (Hewlett Packard & DSpace), **Elizabeth Gadd** (Loughborough University & RoMEO), **Brian Green** (EDItEUR), **Chris Gutteridge** (Southampton University & eprints.org), **Carl Lagoze** (Cornell University & OAI), **Mike Linksvayer** (Creative Commons), **Uwe Müller** (Humboldt University), **Michael Nelson** (Old Dominion University & OAI), **John Ober** (California Digital Library), **Charles Oppenheim** (Loughborough University & RoMEO), **Sandy Payette** (Cornell University), **Andy Powell** (UKOLN, University of Bath), **Steve Proberts** (Loughborough University & RoMEO), **Herbert Van de Sompel** (Los Alamos National Laboratory & OAI), and **Simeon Warner** (Cornell University, arXiv & OAI)

# Scope

- No new rights expression language
  - Don't restrict to specific language(s)
  - Don't get bogged down in rights vs permissions vs enforcement, OAI-PMH is about transferring XML data
  - Right about metadata a separate problem from rights about resources
    - Tackle rights about metadata first
    - Postpone work on rights about resources (note overlap with resource harvesting work)
- ? Issues with rights expressions for aggregations of items (OAI sets; whole repositories)
- ? Issues with whether and how changes in rights expressions should be picked up in selective harvesting (datestamps)

# Creative Commons as example language

- Felt we should pick one as an example
  - RoMEO aligned with Create Commons (CC)
  - CC fits well with interests of many of the original OAI participants (e.g. arXiv considering use of CC)
  - CC is a “good thing” to promote
- Picking CC turned out to be a little complicated because of RDF formulation.
  - No XML schema
  - Refer to only by-reference
- CC really is just an example, can use any XML rights expression language (REL)
  - Will likely add appendices with other example languages later

# OAI-PMH data model

Data model elements:

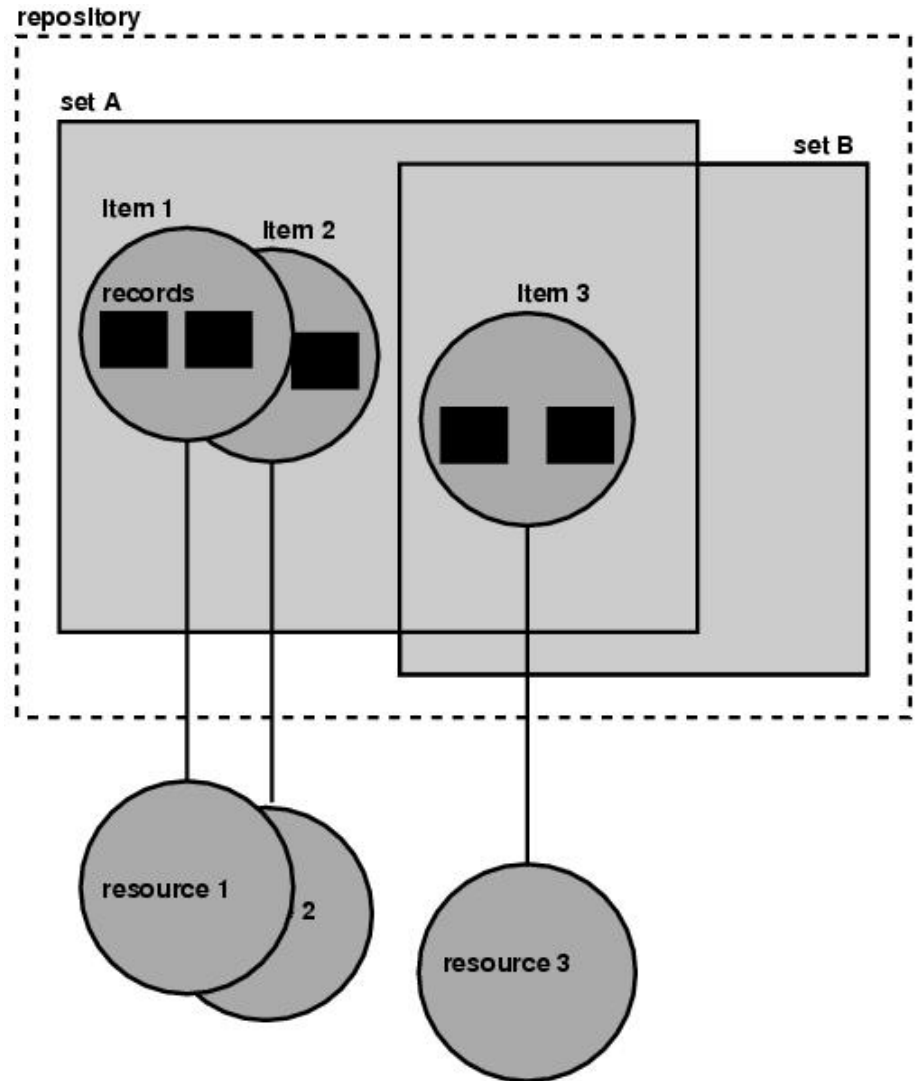
**repository**

**item** - all metadata about a resource, has identifier

**record** - metadata in a particular format, plus header and information about the metadata

**set** - optional, overlapping, hierarchical groupings of items

**resource** outside scope of OAI-PMH



# Different aggregation levels

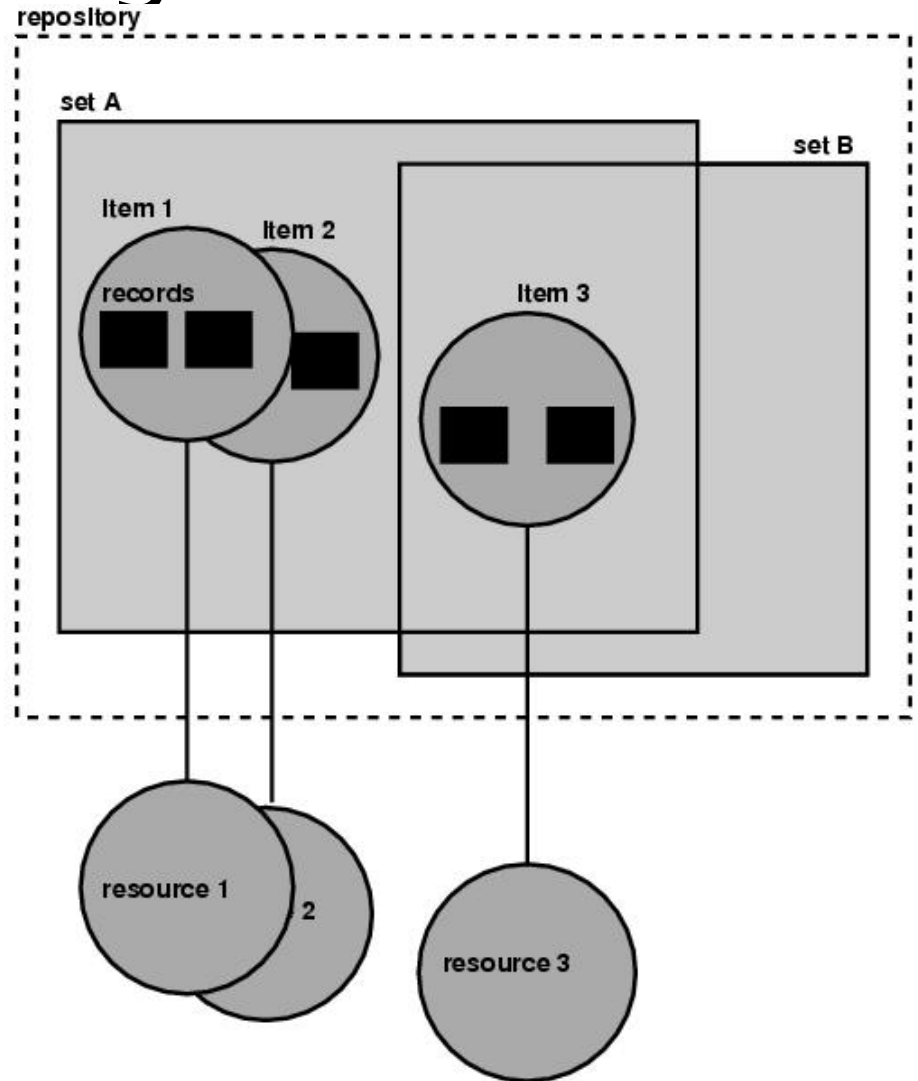
Aggregation levels:

**record** - Rights about an individual record

**repository** - Manifests of rights about all records (all metadata formats from each item) in a repository

**set** - Manifests of rights about all records (all metadata formats from each item) in a set

**Record level** expression is authoritative. Other levels are optional



# record level rights expressions

- W3C XML schema defines format for `<rights>` package to be included in `<about>` container

```
<record>
  <header> id, datestamp, sets </header>
  <metadata> metadata: DC, MARCXML, ... </metadata>
  <about> <rights>...</rights> </about>
  <about> provenance, branding etc. </about>
</record>
```

# record level rights expressions

- Actual rights expression may be in-line (must be valid XML) or by-reference (at given URL, XML recommended)
- In-line method recommended for truly static rights expressions.  
Avoids possible ambiguity with delayed de-referencing

```
<record>
  <header> id, datestamp, sets </header>
  <metadata> metadata: DC, MARCXML, ... </metadata>
  <about> <rights>...</rights> </about>
  <about> provenance, branding etc. </about>
</record>
```



# set and repository level expressions

- These are **optional** and **non-authoritative**
- W3C XML schema defines **<rightsManifest>** package which contains a sequence of **<rights>** elements (as used at the **record** level)
- **<rightsManifest>** included in
  - For **repository** level: **<description>** in Identify
  - For **set** level: **<setDescription>** in ListSets response
- Useful when there is a small set of expressions within the particular aggregation
- Should be accurate and complete but this is not enforced by specification

# Rights about resources

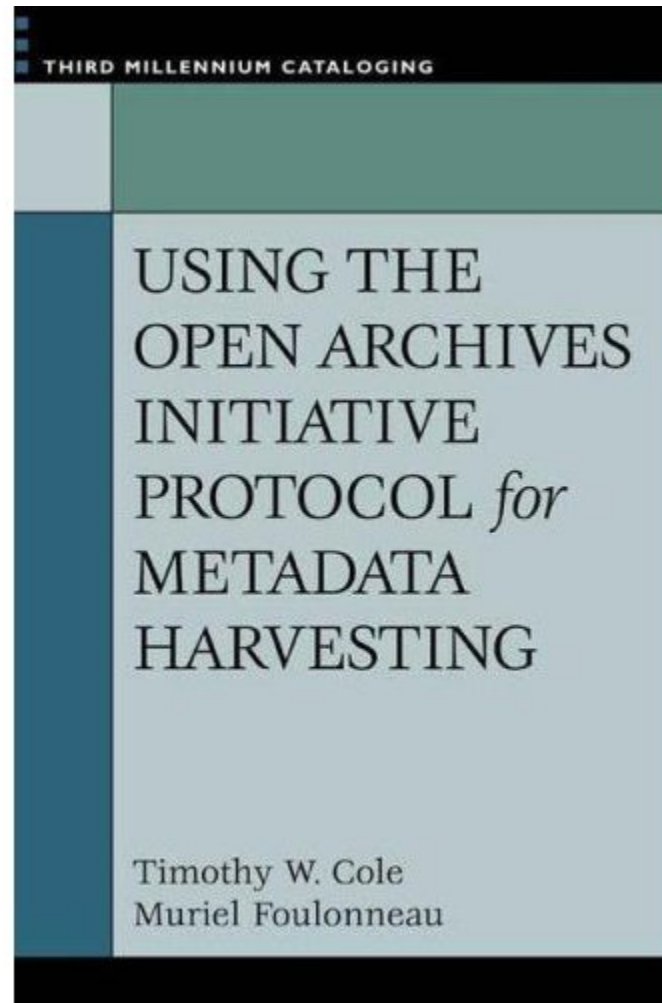
- Can already be done: use an appropriate metadata format as one of the parallel metadata formats from an item. But:
  - Too much choice: need profile
  - Issues with identification of resources
- Overlap with resource harvesting work

<http://www.openarchives.org/OAI/2.0/guidelines-rights.htm>

# V.3 Other resources

# OAI, the book

By Tim Cole and  
Muriel  
Foulonneau,  
available soon.



# Other tutorials online

Previous CERN OAI meetings, all online:

- OAI4 (<http://oai4.web.cern.ch/OAI4/>)
- OAI3 (<http://oai3.web.cern.ch/OAI3/>)
- also links back to OAI2, OAI1

OAForum tutorial

- <http://www.oaforum.org/tutorial/>