The Open Archives Initiative (OAI) and the Protocol for Metadata Harvesting (OAI-PMH)

CS431 guest lecture

Simeon Warner

3 March 2004

Origins of the OAI

*"The Open Archives Initiative has been set up to create a forum to discuss and solve matters of **interoperability** between electronic **preprint** solutions, as a way to promote their global acceptance. "*

(Paul Ginsparg, Rick Luce & Herbert Van de Sompel - 1999)

What is the OAI now?

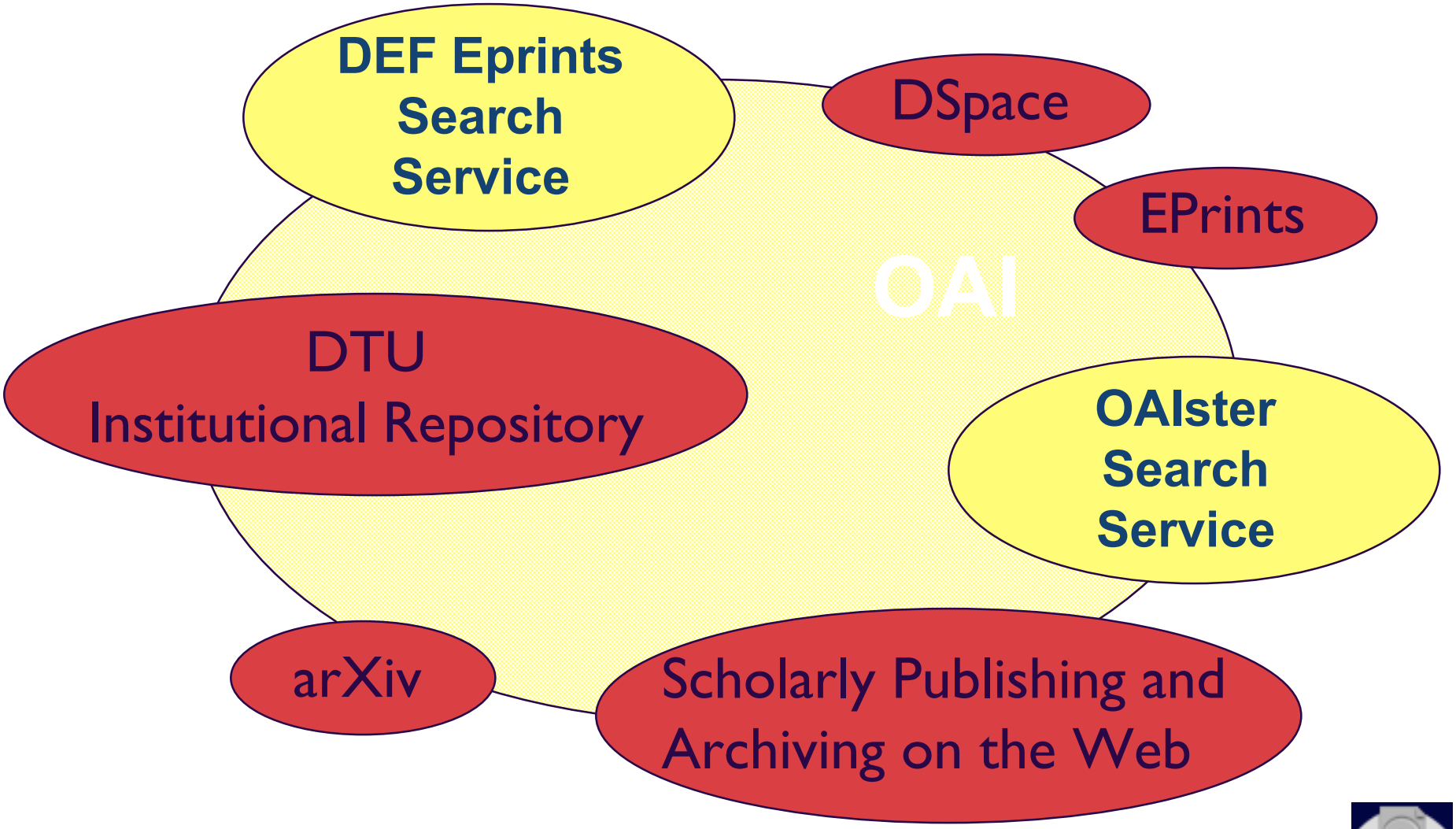
"The OAI develops and promotes interoperability standards that aim to facilitate the efficient dissemination of content." *(from OAI mission statement)*

- Technological framework around OAI-PMH protocol
- Application independent
- Independent of economic model for content

Also … a community and a "brand"

(and you need it for an assignment due in April)

Where does the OAI fit?



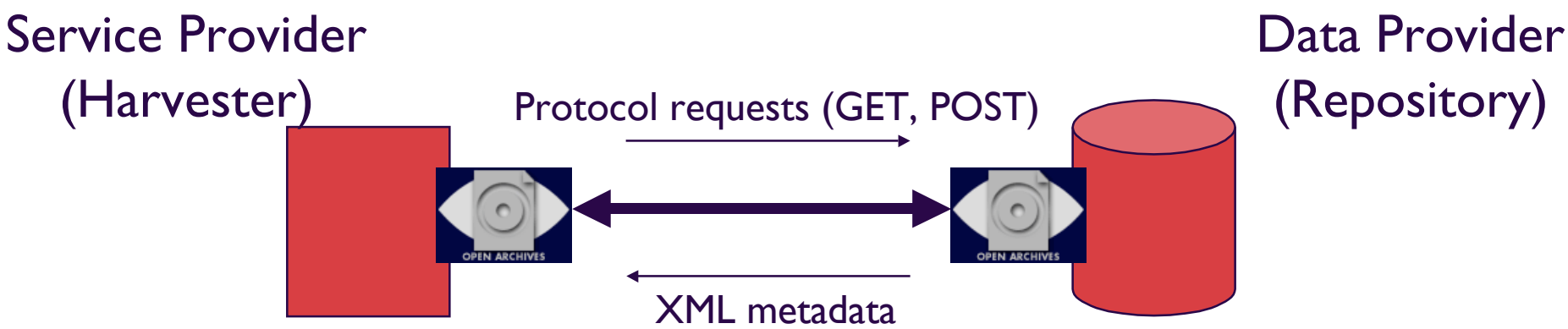
OAI and Open Access

- There is "A" difference
  - Open Archives Initiative
  - Open Access
- The OAI is not tied to a particular political agenda - technical focus
- BUT… the OAI provides functionality that is essential for many Open Access proposals

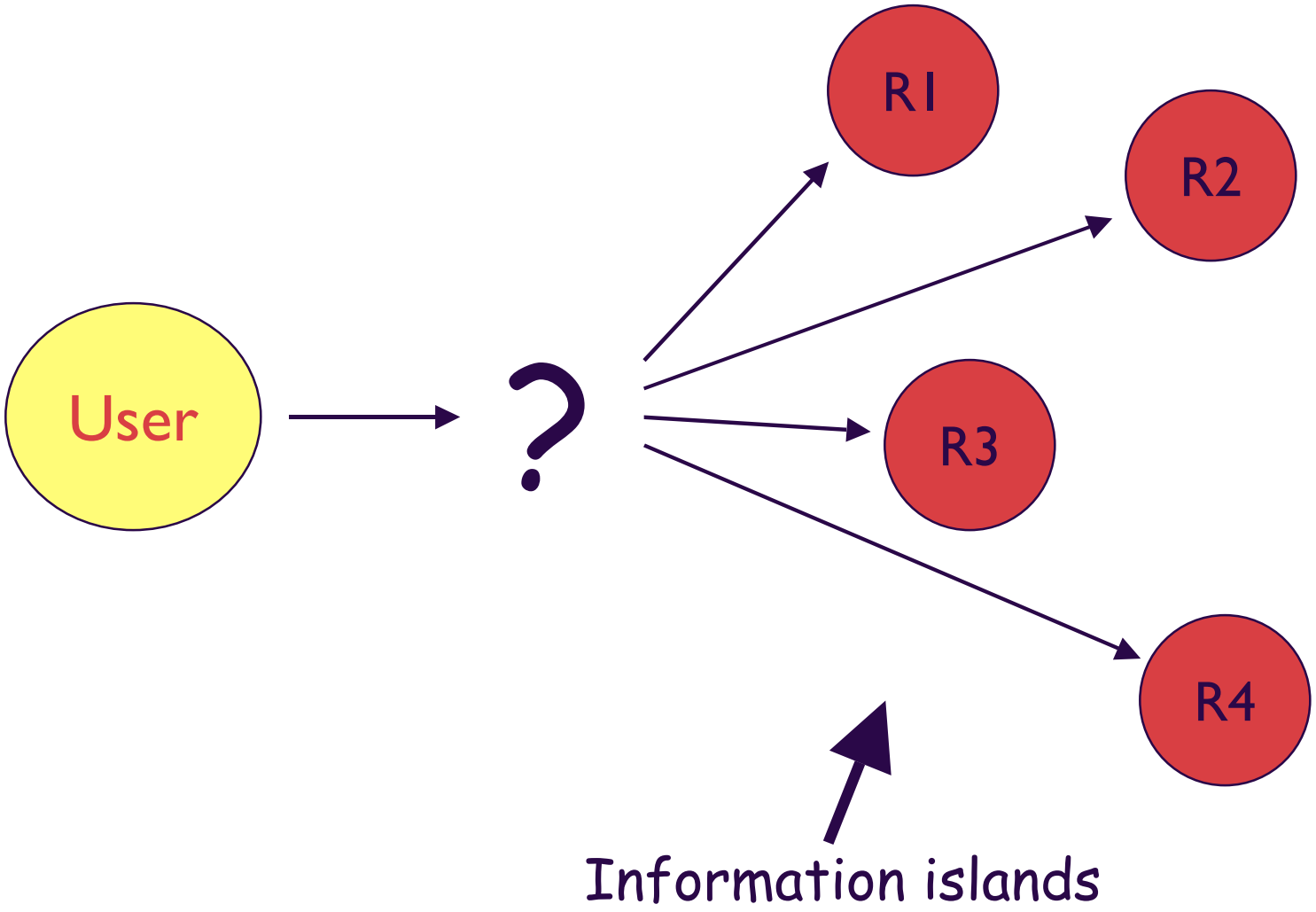
OAI-PMH

⇒ PMH -> Protocol for Metadata Harvesting
http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm

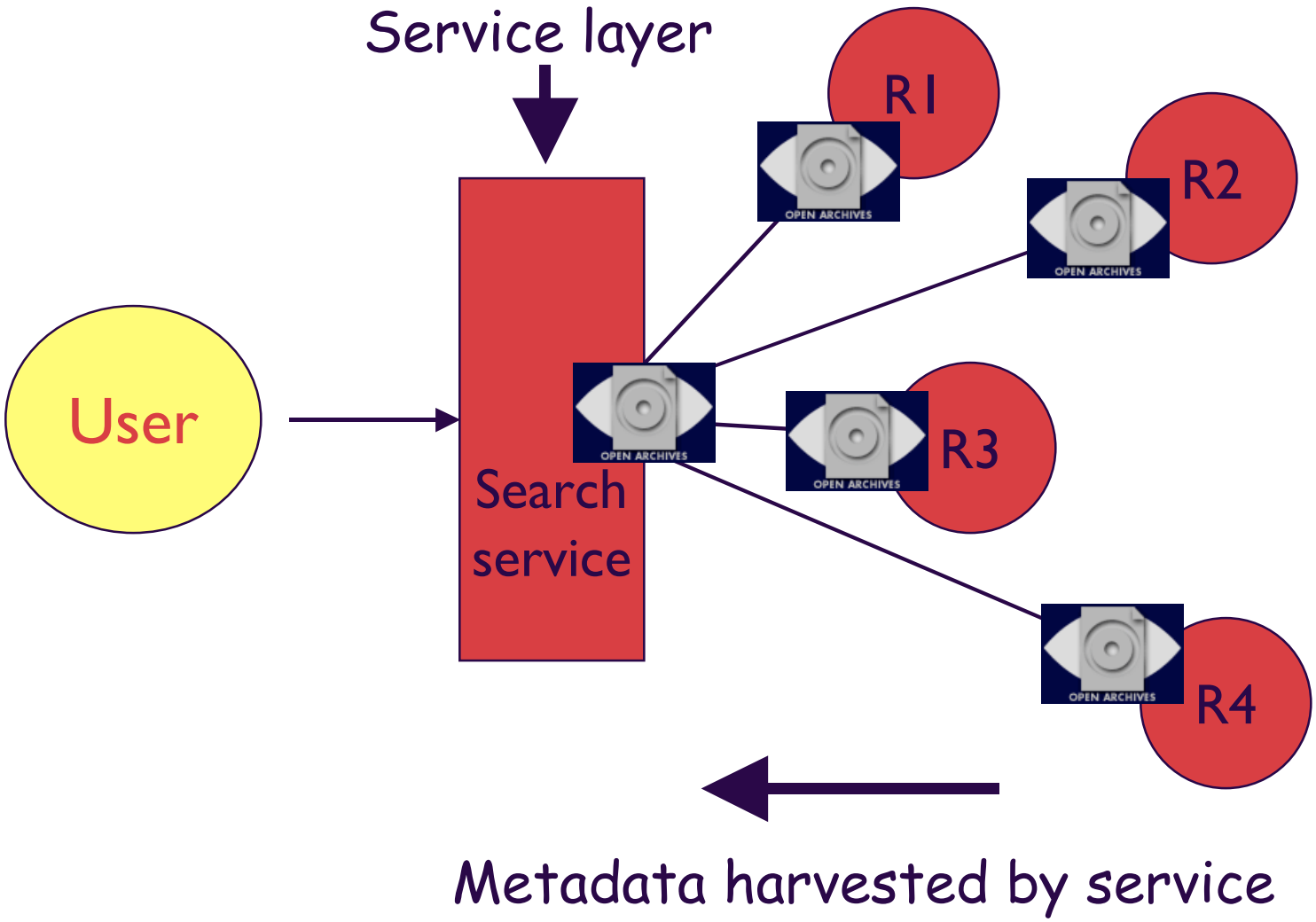
- Simple protocol, just 6 verbs
- Designed to allow harvesting of any XML metadata (schema described)
- For batch-mode not interactive use



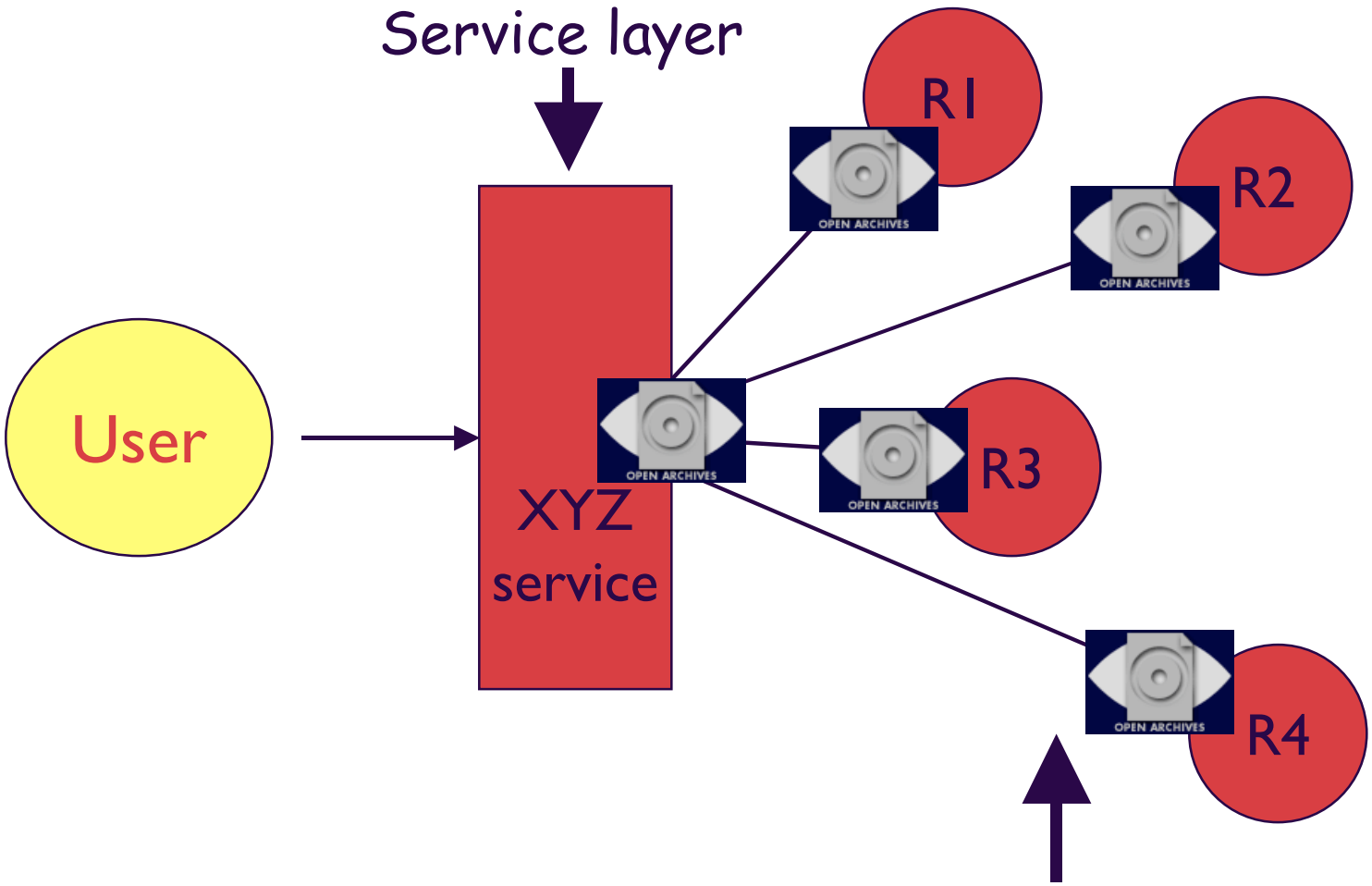
OAI for discovery



OAI for discovery



OAI for XYZ



Global network of resources exposing metadata

OAI-PMH Data Model

resource

item has identifier

all available metadata about this sculpture

item

Dublin Core metadata

MARC21 metadata

branding metadata

records

record has identifier + metadata format + datestamp

OAI-PMH and HTTP

- Clear separation of OAI-PMH and HTTP: OAI-PMH uses HTTP as transport
  - ✂ all OK at HTTP level? => 200 OK
  - ✂ something wrong at OAI-PMH level? => OAI-PMH error (e.g. badVerb)
- HTTP codes 302 (redirect), 503 (retry-after), etc. still available to implementers, but do not represent OAI-PMH events
- Not REST like

Normal response

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH>        ....namespace info not shown here
<responseDate>2002-0208T08:55:46Z</responseDate>
<request verb="GetRecord"… …>http://arXiv.org/oai2</request>
 <GetRecord>
  <record>
   <header>
    <identifier>oai:arXiv: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

Error/exception response

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH>
<responseDate>2002-0208T08:55:46Z</responseDate>
<request>http://arXiv.org/oai2</request>
<error code="badVerb">ShowMe is not a valid OAI-PMH verb</error>
</OAI-PMH>
```

Same schema for all responses, including error responses.

with errors, only the correct attributes are echoed in <request>

OAI-PMH verbs

	Verb	Function
metadata about the repository	Identify	description of archive
	ListMetadataFormats	metadata formats supported by archive
	ListSets	sets defined by archive
harvesting verbs	ListIdentifiers	OAI unique ids contained in archive
	ListRecords	listing of N records
	GetRecord	listing of a single record

most verbs take arguments: dates, sets, ids, metadata formats and resumption token (for flow control)

Identify verb

Information about the repository, start any harvest with Identify

```
<Identify>
    <repositoryName>Library of Congress 1</repositoryName>
    <baseURL>http://memory.loc.gov/cgi-bin/oai</baseURL>
    <protocolVersion>2.0</protocolVersion>
    <adminEmail>r.e.gillian@larc.nasa.gov</adminEmail>
    <adminEmail>rgillian@visi.net</adminEmail>
    <deletedRecord>transient</deletedRecord>
    <earliestDatestamp>1990-02-01T00:00:00Z</earliestDatestamp>
    <granularity>YYYY-MM-DDThh:mm:ssZ</granularity>
    <compression>deflate</compression>
```

Identifiers

- Items have identifiers (all records of same item share identifier)
- Identifiers must have URI syntax (defined by RFC, a type in XML schema)
- Unless you can recognize a global URI scheme, identifiers must be assumed to be local to the repository
- Complete identification of a record is baseURL+identifier+metadataPrefix+datestamp
- `<provenance>` container may be used to express harvesting/transformation history

Datestamps

- All dates/times are UTC, encoded in ISO8601, Z notation: **`1957-03-20T20:30:00Z`**
- Datestamps may be either fill date/time as above or date only (YYYY-MM-DD). Must be consistent over whole repository, 'granularity' specified in Identify response.
- Earlier version of the protocol specified "local time" which caused lots of misunderstandings. Not good for global interoperability!

Harvesting granularity

- mandatory support of `YYYY-MM-DD`
- optional support of `YYYY-MM-DDThh:mm:ssZ` (must look at Identify response)
- granularity of `from` and `until` agrument in ListIdentifier/ListRecords must match

Sets

- Simple notion of grouping at the item level to support selective harvesting
  - Hierarchical set structure
  - Multiple set membership permitted
  - E.g: repo has sets A, A:B, A:B:C, D, D:E, D:F

    If item1 is in A:B then it is in A

    If item2 is in D:E then it is in D, may also be in D:F

    Item3 may be in no sets at all
- Don't use sets unless you have a good reason (selective harvesting)

resumptionToken

- Protocol supports the notion of partial responses in a very simple way: Response includes a 'token' at the which is used to get the next chunk.
- Idempotency of `resumptionToken`: return same incomplete list when `resumptionToken` is reissued
  - while no changes occur in the repo: strict
  - while changes occur in the repo: all items with unchanged `datestamp`
  - optional attributes for the resumptionToken: `expirationDate, completeListSize, cursor`

Record headers

- `header` contains set membership of item

```
<record>
  <header>
   <identifier>oai:arXiv:cs/0112017</identifier>
   <datestamp>2001-12-14</datestamp>
   <setSpec>cs</setSpec>
   <setSpec>math</setSpec>
  </header>
  <metadata>
   …..
  </metadata>
 </record>
```

eliminates the need for the "double harvest" 1.x required to get all records and all set information

Deleted records

- What happens when a record (or item) is deleted from a repository? Would be nice if harvesters could find out.
- Not necessarily guaranteed in OAI that harvesters will find out. Support made optional because of problems with legacy repositories (practical constraint).
  - Level of support expressed in Identify (`no`, `persistent`, `transient`)
  - Status expressed in header element,

    `<header status="deleted">…</header>`

Harvesting strategy

- Issue Identify request
  - Check all as expected (validate, version, baseURL, granularity, comporession…)
- Check sets/metadata formats as necessary (ListSets, ListMetadataFormats)
- Do harvest, initial complete harvest done with no `from` and `to` parameters
- Subsequent incremental harvests start from datastamp that is `responseDate` of last response

Changing Scholarly Communication

- Traditional journal publishing combines functions: registration, certification, awareness, archiving.
- How about eprints being the starting point of a new value chain in which the raw material - the non-certified eprint - is open access?
- Other functions might be fullfilled by different networked parties. This requires a communication infrastructure: OAI-PMH may be part of this.
- Presentations on OAI and Scholarly Communication at http://www.cs.cornell.edu/people/simeon/talks