

Interoperability (cont.)

Metadata Syndication and Harvesting

CS 431 - March 8, 2006

Carl Lagoze - Cornell University

PageRank Algorithm (Google)

Concept:

The rank of a web page is higher if many pages link to it.

Links from highly ranked pages are given greater weight than links from less highly ranked pages.

PageRank with Damping Factor Intuitive Model

Ranking of a page is based on model of a user who:

1. Starts at a random page on the web
- 2a. With probability p , selects any random page and jumps to it
- 2b. With probability $1-p$, selects a random hyperlink from the current page and jumps to the corresponding page
3. Repeats Step 2a and 2b a very large number of times

Pages are ranked according to the relative frequency with which they are visited.

Information Retrieval Using PageRank

Simple Method

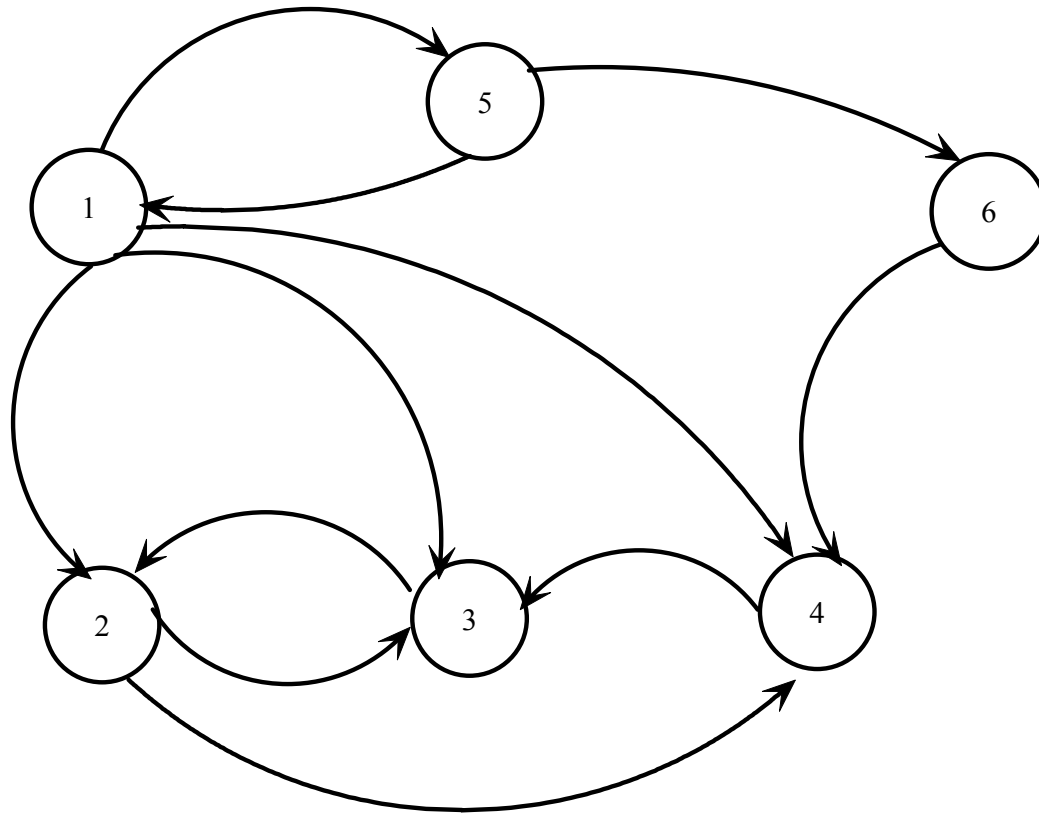
Consider all hits (i.e., all document vectors that share at least one term with the query vector) as equal.

Display the hits ranked by PageRank.

The disadvantage of this method is that it gives no attention to how closely a document matches a query

With **dynamic document sets**, references patterns are calculated for a set of documents that are selected based on each individual query.


Google Example



Adjacency Matrix

		Citing page (from)						Number
		P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	
Cited page (to)	P ₁					1		1
	P ₂	1		1				2
	P ₃	1	1		1			3
	P ₄	1	1			1	1	4
	P ₅	1						1
	P ₆					1		1
Number		4	2	1	1	3	1	

Normalize by Number of Links from Page

		Citing page					
		 P ₁	P ₂	P ₃	P ₄	P ₅	P ₆
Cited page	P ₁					0.33	
	P ₂	0.25		1			
	P ₃	0.25	0.5		1		
	P ₄	0.25	0.5			0.33	1
	P ₅	0.25					
	P ₆					0.33	
Number		4	2	1	1	3	1

= B

**Normalized
link matrix**

Iterate until convergence

Iterate: $\mathbf{w}_k = \mathbf{B}\mathbf{w}_{k-1}$

\mathbf{w}_1	\mathbf{w}_2	\mathbf{w}_3	\mathbf{w}_4	... converges to ...	\mathbf{w}
$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0.33 \\ 1.25 \\ 1.75 \\ 2.08 \\ 0.25 \\ 0.33 \end{bmatrix}$	$\begin{bmatrix} 0.08 \\ 1.83 \\ 2.79 \\ 1.12 \\ 0.08 \\ 0.08 \end{bmatrix}$	$\begin{bmatrix} 0.03 \\ 2.80 \\ 2.06 \\ 1.05 \\ 0.02 \\ 0.03 \end{bmatrix}$	$\begin{matrix} \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \end{matrix}$	$\begin{bmatrix} 0.00 \\ 2.39 \\ 2.39 \\ 1.19 \\ 0.00 \\ 0.00 \end{bmatrix}$

The PageRank Iteration

The basic method iterates using the normalized link matrix, B .

$$w_k = Bw_{k-1}$$

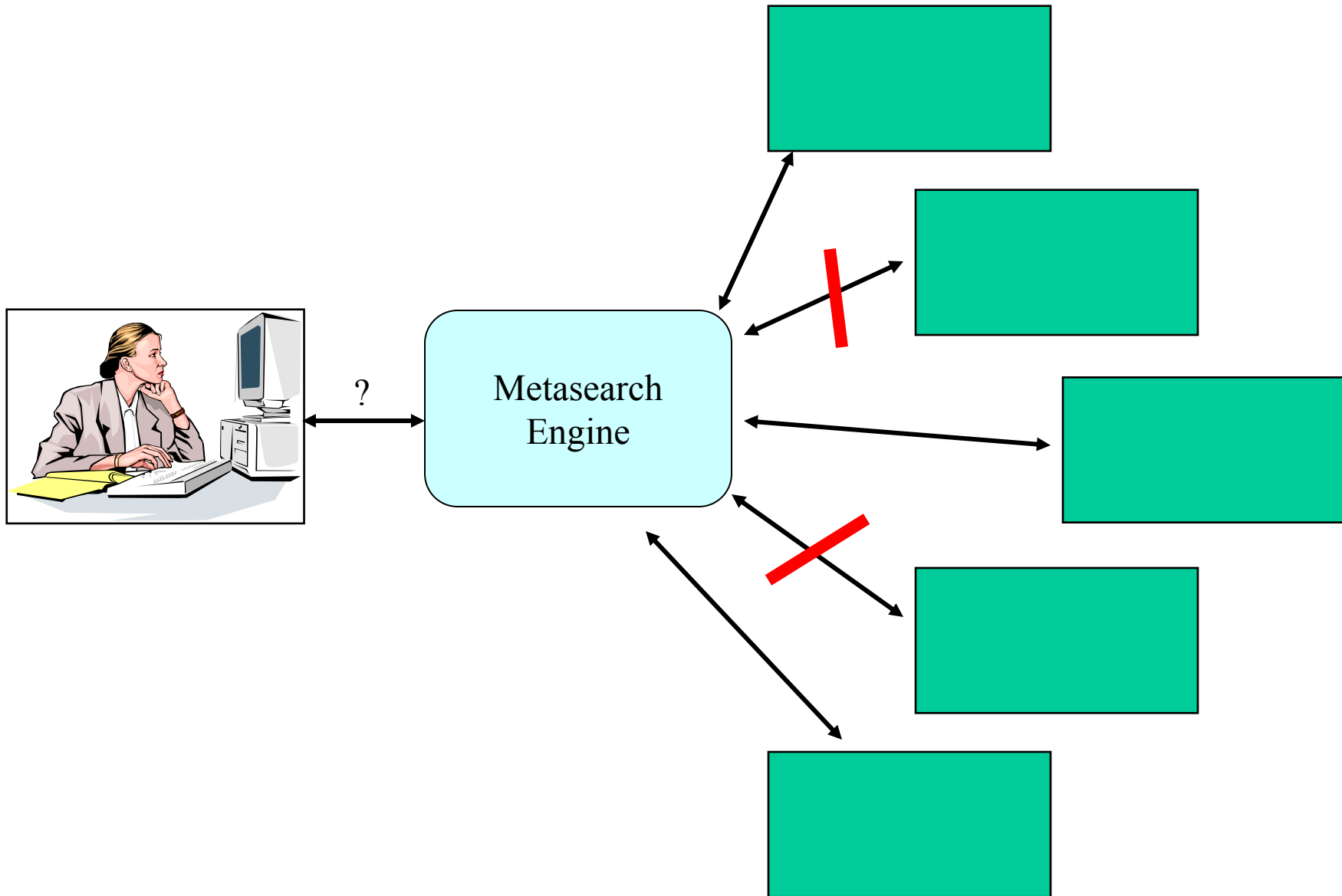
This w is the high order eigenvector of B

Google iterates using a damping factor to model the behavior of the surfer getting bored. The method iterates using a matrix B' , where:

$$B' = dN + (1 - d)B$$

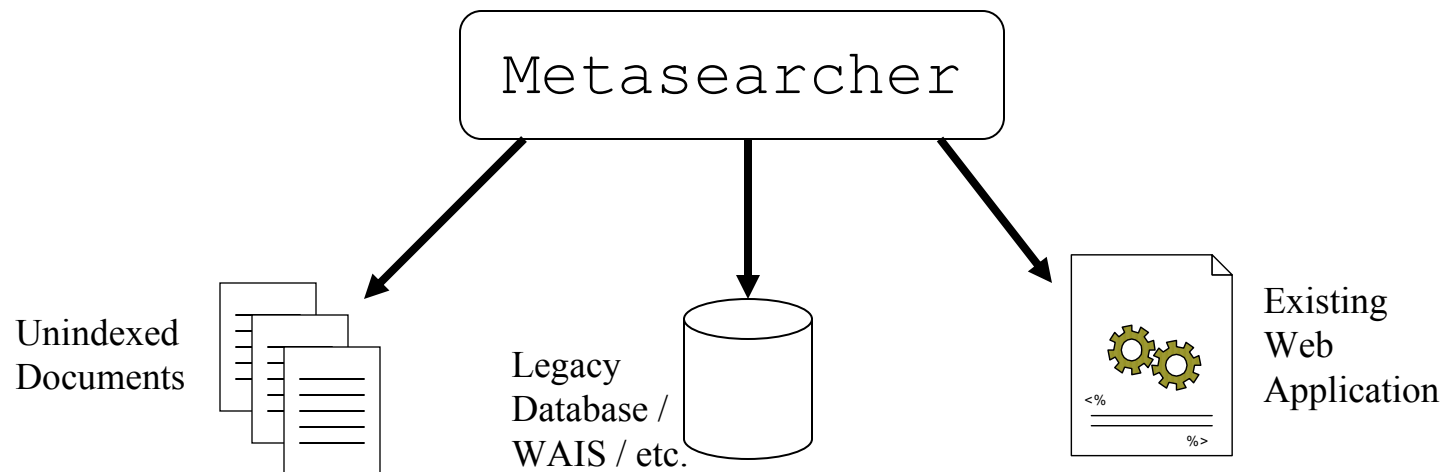
N is the matrix with every element equal to $1/n$.
 d is a constant found by experiment (.85 in original paper).

Web Search Strategies - Metasearching



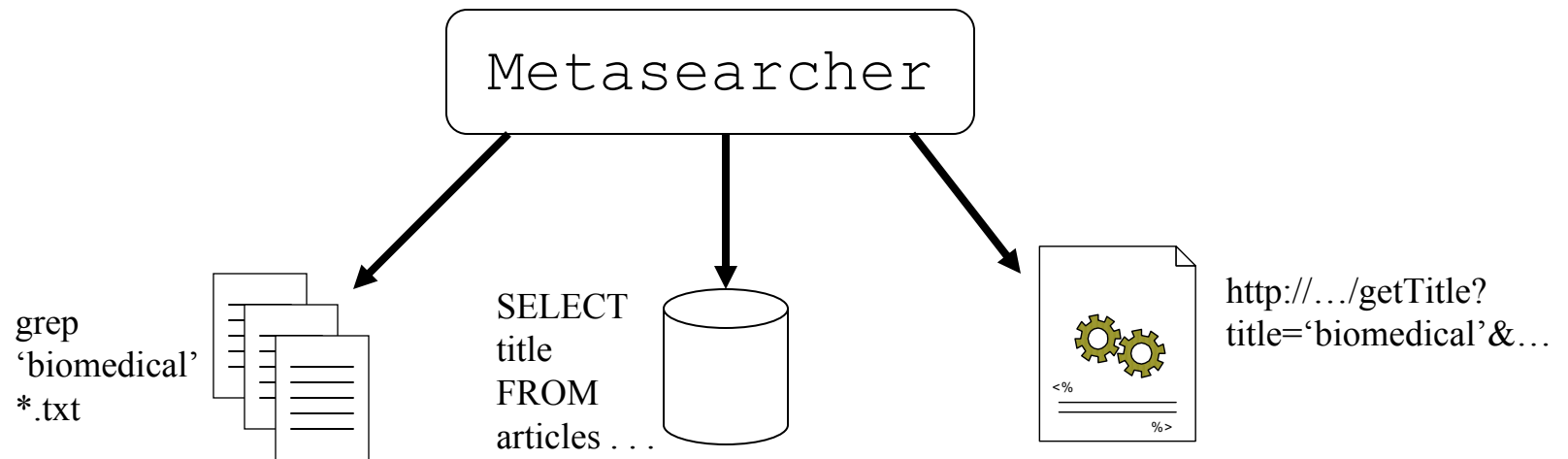
What is "Metasearching"?

- Given many document sources and a query, a metasearcher:
 - Finds the good sources for the query
 - Evaluates the query at these sources
 - Merges the results from these sources



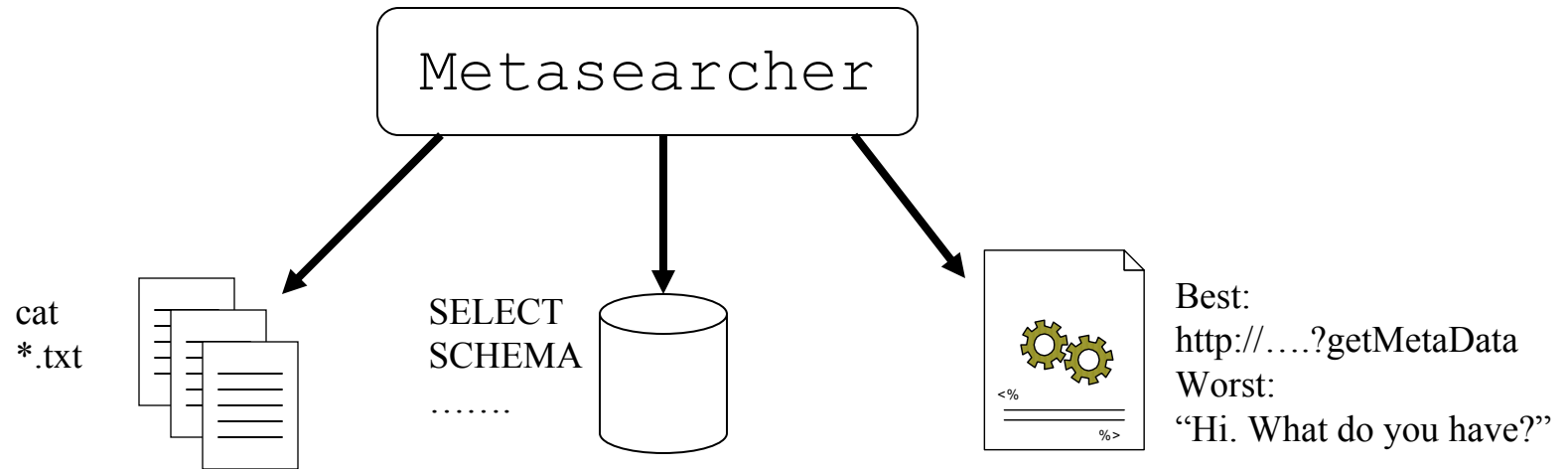
Metasearching Issues

- How to query different types of sources?
- How to combine results and rankings from multiple data sources?



Metasearching Issues . . . Cont'd

- How to choose among multiple data sources?
- How to get metadata about multiple data sources?



ZING

[http://www.loc.gov/z3950/agency/zing/zin
g-home.html](http://www.loc.gov/z3950/agency/zing/zin
g-home.html)

The problem

- Search syntax differs across engines
 - <http://www.google.com/search?hl=en&ie=ISO-8859-1&q=dogs+and+cats&btnG=Google+Search>
 - <http://search.yahoo.com/search?fr=fp-pull-web-t&p=dogs+and+cats>
 - <http://search.msn.com/results.aspx?FORM=MSNH&q=dogs%20and%20cats>
- Means of returning results sets differs
- Statefulness - sometimes it makes sense to manipulate result sets

Aims of ZING

- Common framework (implemented as protocol) for searching over multiple servers
- Builds on notion of metadata (attribute-based access points to information).
- Components
 - CQL – Common query syntax, keyword and attribute based
 - SRU – REST based transmission of requests
 - SRW – SOAP based transmission of requests

SRW/SRU services

- Explain
 - Return information about the database - search access points (e.g, title, author) metadata formats returned
- Scan
 - Return information about an index term (e.g., related terms)
- Search
 - Return search results

SRW Result Sets

- Server may support notion of persistent result sets
 - Return an ID of the set from query
- Client may perform operations on result sets
 - Refine searches
 - Chunk results
- Server makes "commitment" to retain result set but may change commitment.

Metadata aggregation and harvesting

- Crawling is not always appropriate
 - rights issues
 - focused targets
 - firewalls
 - deep web
- Other applications than search
 - Current awareness
 - Specialized surrogates/metadata
 - rights statement
 - structural description

Syndication - RSS and Atom

- Format to expose news and content of news-like sites
 - Wired
 - Slashdot
 - Weblogs
- "News" has very wide meaning
 - Any dynamic content that can be broken down into discrete items
 - Wiki changes
 - CVS checkins
- Roles
 - Provider *syndicates* by placing an RSS-formated XML file on Web
 - Aggregator runs RSS-aware program to check feeds for changes

RSS History

- Original design (0.90) for Netscape for building portals of headlines to news sites
 - Loosely RDF based
- Simplified for 0.91 dropping RDF connections
- RDF branch was continued with namespaces and extensibility in RSS 1.0
- Non-RDF branch continued to 2.0 release
- Alternately called:
 - Rich Site Summary
 - RDF Site Summary
 - Really Simple Syndication

RSS is in wide use

- All sorts of origins
 - News
 - Blogs
 - Corporate sites
 - Libraries
 - Commercial

RSS components

- *Channel*
 - single tag that encloses the main body of the RSS document
 - Contains metadata about the channel - *title, link, description, language, image*
- *Item*
 - Channel may contain multiple items
 - Each item is a "story"
 - Contains metadata about the story (*title, description, etc.*) and possible *link* to the story

RSS 2.0 Example

```
<rss version="2.0" xmlns:dc="http://purl.org/dc/elements/1.1/">
  <channel>
    <title>XML.com</title>
    <link>http://www.xml.com/</link>
    <description>XML.com features a rich mix of information and services
for the XML community.</description>
    <language>en-us</language>
    <item>
      <title>Normalizing XML, Part 2</title>
      <link>http://www.xml.com/pub/a/2002/12/04/normalizing.html</link>
      <description>In this second and final look at applying relational
normalization techniques to W3C XML Schema data modeling, Will Provost
discusses when not to normalize, the scope of uniqueness and the fourth
and fifth normal forms.</description>
      <dc:creator>Will Provost</dc:creator>
      <dc:date>2002-12-04</dc:date>
    </item>
    <item>
      <title>The .NET Schema Object Model</title>
      <link>http://www.xml.com/pub/a/2002/12/04/som.html</link>
      <description>Priya Lakshminarayanan describes in detail the use of
the .NET Schema Object Model for programmatic manipulation of W3C XML
Schemas.</description>
      <dc:creator>Priya Lakshminarayanan</dc:creator>
      <dc:date>2002-12-04</dc:date>
    </item>
  </channel>
</rss>
```


RSS 1.0 Example

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://purl.org/rss/1.0/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
>
  <channel rdf:about="http://example.com/news.rss">
    <title>Example Channel</title>
    <link>http://example.com/</link>
    <description>My example channel</description>
    <items>
      <rdf:Seq>
        <rdf:li resource="http://example.com/2002/09/01/">
        <rdf:li resource="http://example.com/2002/09/02/">
      </rdf:Seq>
    </items>
  </channel>
  <item rdf:about="http://example.com/2002/09/01">
    <title>News for September the Second</title>
    <link>http://example.com/2002/09/01</link>
    <description>other things happened today</description>
    <dc:date>2002-09-01</dc:date>
  </item>
  <item rdf:about="http://example.com/2002/09/01">
    <title>News for September the First</title>
    <link>http://example.com/2002/09/02</link>
    <dc:date>2002-09-02</dc:date>
  </item>
</rdf:RDF>
```

RSS applications

- <http://www.syndic8.com/>
- Automated discovery of RSS feeds
 - `<link rel="alternate" type="application/rss+xml" title="RSS 2.0" href="http://www.goals-2-go.com/?feed=rss2" />`
 - `<link rel="alternate" type="text/xml" title="RSS .92" href="http://www.goals-2-go.com/?feed=rss" />`
 - `<link rel="alternate" type="application/atom+xml" title="Atom 0.3" href="http://www.goals-2-go.com/?feed=atom" />`
- Aggregators
 - AmphetaDesk - <http://disobey.com/amphetadesk/>
 - NewsGator - <http://www.newsgator.com/home.aspx>
 - NetNewsWire - <http://ranchero.com/netnewswire/>

Atom

- Attempt to rationalize RSS 1.x, 2.x divergence
- Encoding is up-to-date with current XML standards
 - namespaces
 - schema
- Rationalizes the division between metadata and contained content

Atom Example

```
<feed xmlns="http://www.w3.org/2005/Atom"
      xml:lang="en"
      xml:base="http://www.example.org">
  <id>http://www.example.org/myfeed</id>
  <title>My Simple Feed</title>
  <updated>2005-07-15T12:00:00Z</updated>
  <link href="/blog" />
  <link rel="self" href="/myfeed" />
  <entry>
    <id>http://www.example.org/entries/1</id>
    <title>A simple blog entry</title>
    <link href="/blog/2005/07/1" />
    <updated>2005-07-15T12:00:00Z</updated>
    <summary>This is a simple blog entry</summary>
  </entry>
  <entry>
    <id>http://www.example.org/entries/2</id>
    <title />
    <link href="/blog/2005/07/2" />
    <updated>2005-07-15T12:00:00Z</updated>
    <summary>This is simple blog entry without a title</summary>
  </entry>
</feed>
```

Atom Enclosures and Content Support (podcast)

```
<feed xmlns="http://www.w3.org/2005/Atom">
  <id>http://www.example.org/myfeed</id>
  <title>My Podcast Feed</title>
  <updated>2005-07-15T12:00:00Z</updated>
  <author>
    <name>James M Snell</name>
  </author>
  <link href="http://example.org" />
  <link rel="self" href="http://example.org/myfeed" />
  <entry>
    <id>http://www.example.org/entries/1</id>
    <title>Atom 1.0</title>
    <updated>2005-07-15T12:00:00Z</updated>
    <link href="http://www.example.org/entries/1" />
    <summary>An overview of Atom 1.0</summary>
    <link rel="enclosure"
      type="audio/mpeg"
      title="MP3"
      href="http://www.example.org/myaudiofile.mp3"
      length="1234" />
    <link rel="enclosure"
      type="application/x-bittorrent"
      title="BitTorrent"
      href="http://www.example.org/myaudiofile.torrent"
      length="1234" />
    <content type="xhtml">
      <div xmlns="http://www.w3.org/1999/xhtml">
        <h1>Show Notes</h1>
        <ul>
          <li>00:01:00 -- Introduction</li>
          <li>00:15:00 -- Talking about Atom 1.0</li>
          <li>00:30:00 -- Wrapping up</li>
        </ul>
      </div>
    </content>
  </entry>
</feed>
```

RSS 2.0 and publish and subscribe

- <cloud> element of channel
- Specifies a web service that supports the rssCloud interface which can be implemented in HTTP-POST, XML-RPC or SOAP 1.1
- Allow processes to register with a cloud to be notified of updates to the channel via a callback
- <cloud domain="radio.xmlstoragesystem.com" port="80" path="/RPC2" registerProcedure="xmlStorageSystem.rssPleaseNotify" protocol="xml-rpc" />

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

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?

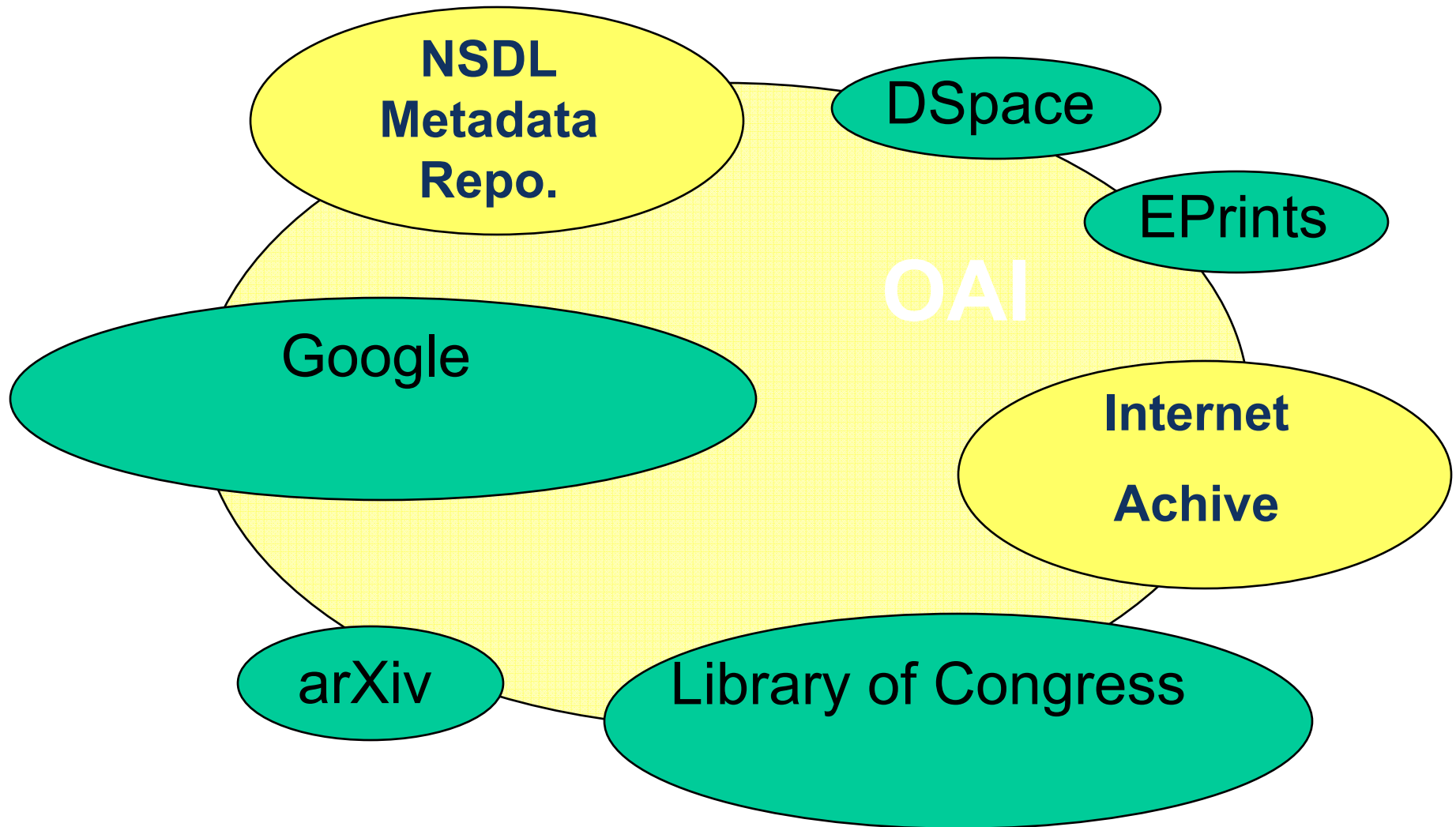
"The OAI develops and promotes interoperability standards that aim to facilitate the efficient dissemination of content." (from OAI mission statement)

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

Also ... a community and a "brand"

- Something you need to complete your project 1!

Where does the OAI fit?

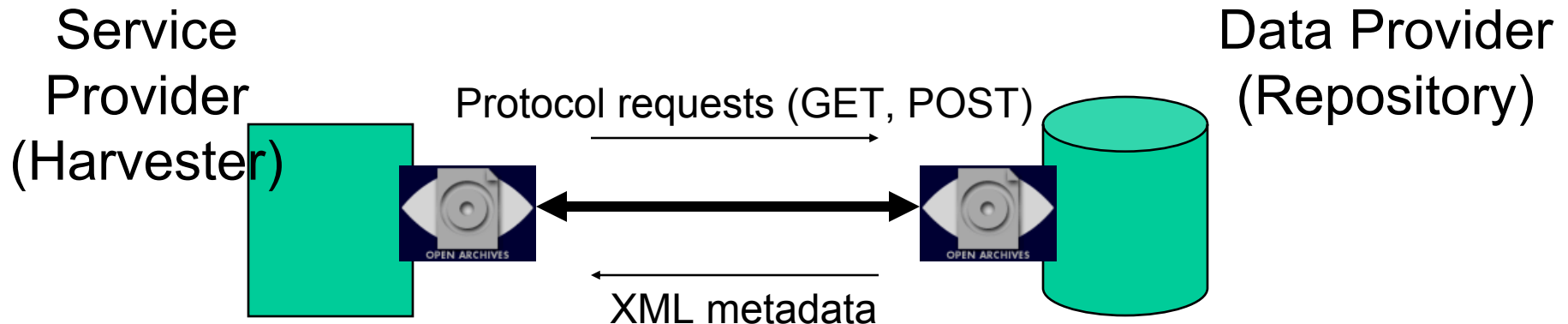


OAI-PMH

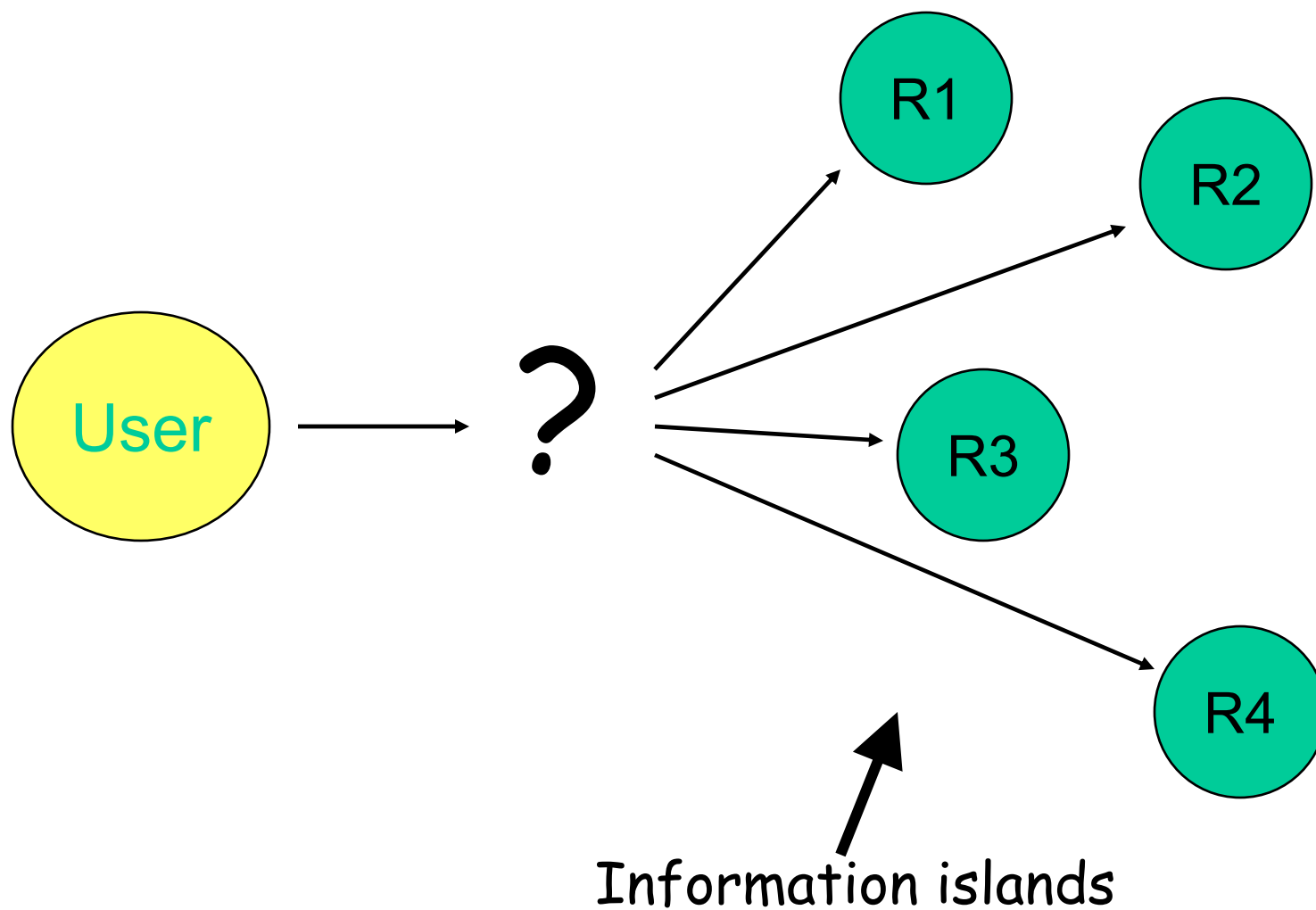
⇒ PMH -> Protocol for Metadata Harvesting

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

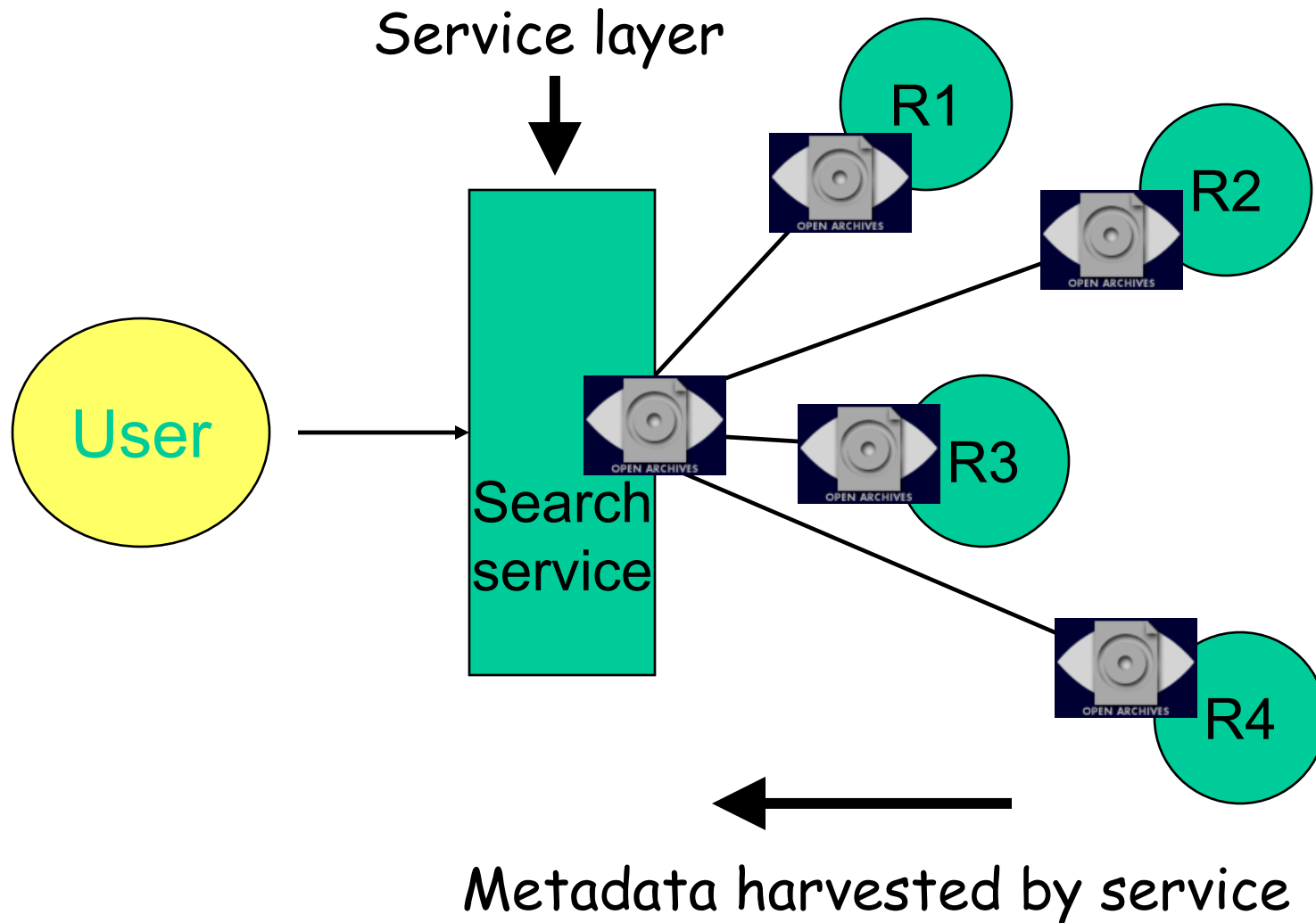
- Simple protocol, just 6 verbs
- Designed to allow harvesting of any XML (meta)data (schema described)
- For batch-mode not interactive use



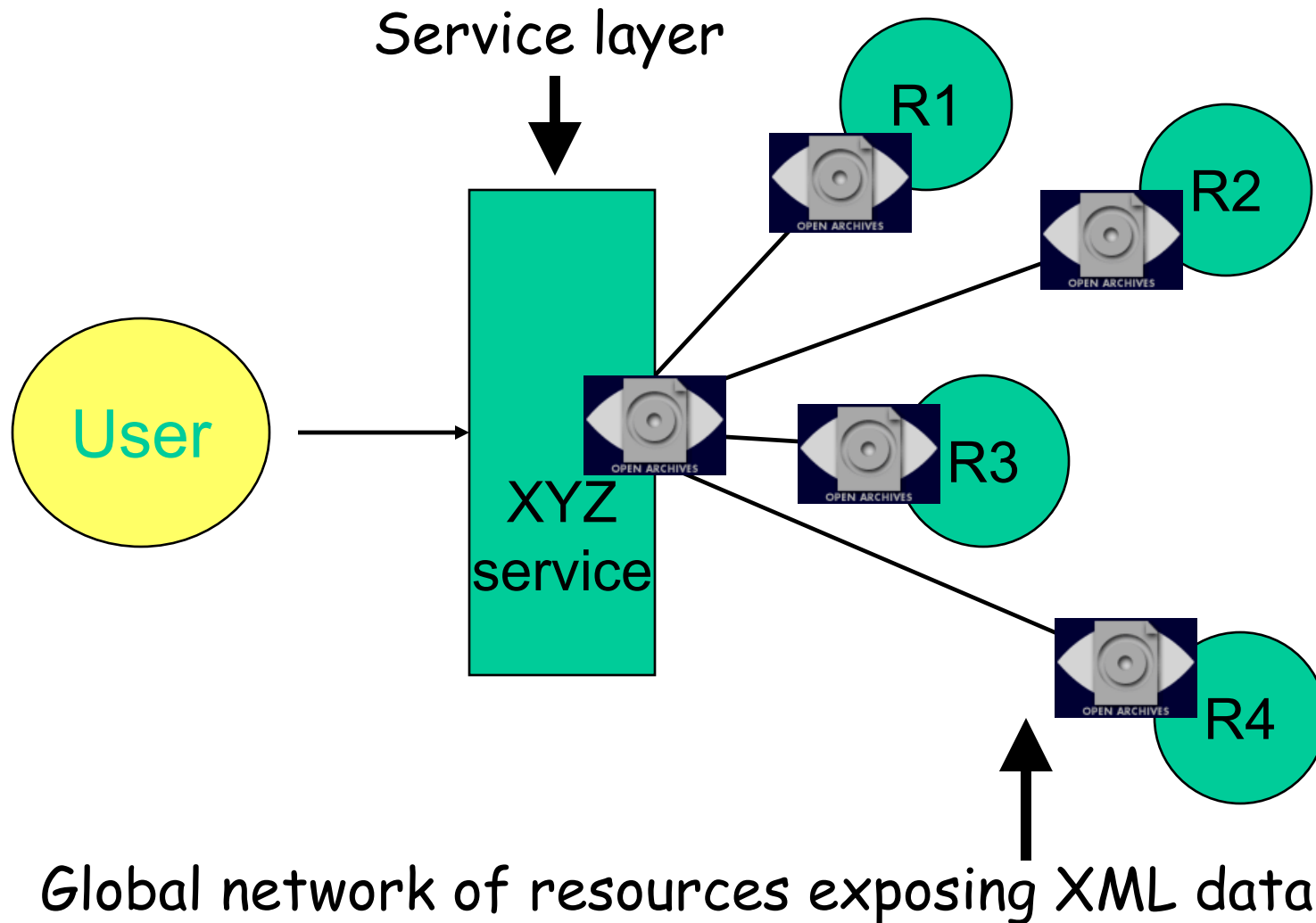
OAI for discovery



OAI for discovery



OAI for XYZ



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 and Metadata Formats

- Protocol based on the notion that a record can be described in multiple metadata formats
- Dublin Core is required for "interoperability"
- Extended to include XML compound object formats: e.g., METS, DIDL
 - <http://www.dlib.org/dlib/december04/vandesompe/12vandesompe.html>

OAI-PMH and HTTP

- OAI-PMH uses HTTP as transport
 - Encoding OAI-PMH in GET
 - `http://baseURL?verb=<verb>&arg1=<arg1Val>...`
 - Example: `http://an.oa.org/OAIscript?verb=GetRecord&identifier=oai:arXiv.org:hep-th/9901001&metadataPrefix=oai_dc`
- Error handling
 - 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

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>

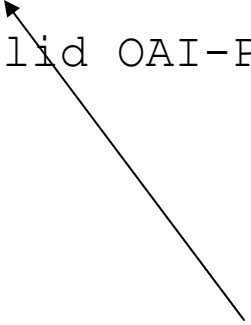
<compression>deflate</compression>

GetRecord - 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>
        <timestamp>2001-12-14</timestamp>
        <setSpec>cs</setSpec>
        <setSpec>math</setSpec>
      </header>
      <metadata>
        ....
      </metadata>
    </record>
  </GetRecord>
</OAI-PMH>
```

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>*

Identifiers

- Items have identifiers (all records of same item share identifier)
- Identifiers must have URI syntax 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

Selective Harvesting

- RSS is mainly a "tail" format
- OAI-PMH is more "grep" like
- Two "selectors" for harvesting
 - Date
 - Set
- Why not general search?
 - Out of scope
 - Not low-barrier
 - Difficulty in achieving consensus

Datestamps

- All dates/times are UTC, encoded in ISO8601, Z notation: 1957-03-20T20:30:00Z
- Datestamps may be either full 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` argument 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

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>
```

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`

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

OAI-PMH - Has it worked?

- Of course, yes...
 - Very wide deployment
 - "millions and millions of records served"
 - Incorporated into commercial systems
- But....
 - NSDL experience has shown "low barrier" is not always true
 - XML is hard
 - Incremental harvesting model is full of holes