A Modular Framework for Web-Based Resource Synchronization

Herbert Van de Sompel
Los Alamos National Laboratory
@hvdsomp

http://www.openarchives.org/rs
#resourcesync

ResourceSync was funded by the Sloan Foundation & JISC
This ResourceSync Presentation

- Problem Domain
- Scope
- Framework – Conceptual Overview
- Framework – Technology Overview
- Implementations, Tools, Pointers
Background - OAI-PMH

- Recurrent metadata exchange from a Data Provider to Service Providers
- XML metadata only
- Repository centric
- Devised 1999-2002, prior to REST, prior to dominance of web search engines
Revisit the Problem Domain - ResourceSync

- Synchronization of resources from a Source to Destinations
- Web resources, anything with an HTTP URI & representation
- Resource centric
- Devised 2012-2013, leverages key ingredients of web architecture, SEO practice
- Updated 2017
Problem Statement

• Consideration:
  • **Source** (server) A has resources that change over time: they get created, modified, deleted
  • **Destination** (servers) X, Y, and Z leverage (some) resources of Source A

• Problem:
  • Destinations want to keep in step with the resource changes at Source A
A Source’s Resources

resources

D

B

A

time
A Source’s Resources Evolve over Time

resources

D

B

A

updated

time
A Source’s Resources Evolve over Time
A Source’s Resources Evolve over Time
A Source’s Resources Evolve over Time

resources

D -> D (updated)

B -> B (updated) -> B (created)

A -> A (updated) -> A (updated)
A Source’s Resources Evolve over Time

(resources)

D → updated → D → deleted

B → created

A → updated

A → updated

A → updated

A → updated

B → updated

D → updated

D → created

D → updated

D → created

D → updated

D → created

D → updated

D → created

D → updated
A Source’s Resources Evolve over Time
Problem Statement

• Consideration:
  • **Source** (server) A has resources that change over time: they get created, modified, deleted
  • **Destination** (servers) X, Y, and Z leverage (some) resources of Source A

• Problem:
  • Destinations want to keep in step with the resource changes at Source A

• Goal:
  • An approach for web-based resource synchronization that has a fair chance of adoption by different communities
This ResourceSync Presentation

- Problem Domain
- Scope
- Framework – Conceptual Overview
- Framework – Technology Overview
Scope – Collection Size

• **Size** of a Source’s resource collection:
  
  • A few resources - small web sites, repositories
  
  • Millions of resources – large repositories, datasets, linked data collections
Scope – Change Frequency

- **Change frequency** of a Source’s resources:
  - Low – daily, weekly, monthly
  - High – seconds, minutes
Scope – Synchronization Latency

• Destination’s requirements regarding synchronization latency:
  • High latency acceptable
  • Low latency essential
Scope – Collection Coverage

• Destination’s requirements regarding the coverage of a Source’s resources:
  
  • Partial coverage of the Source’s resources acceptable
  
  • Full coverage of the Source’s resources verifiable
Scope – Bitstream Accuracy

• Destination’s requirements regarding bitstream accuracy:
  • Unverifiable bitstream accuracy acceptable
  • Verifiable bitstream accuracy essential
One to One Synchronization
One to Many – Master Copy
Many to One - Aggregator
Metadata Harvesting
This ResourceSync Presentation

• Problem Domain

• Scope

• Framework – Conceptual Overview

• Framework – Technology Overview

• Implementations, Tools, Pointers
A Source’s Resources Evolve over Time

- **D** updated
- **D** deleted
- **C** created
- **C** updated
- **B** updated
- **A** updated
- **A** updated

resources

time
Solution Perspective - Destination

- Destination needs regarding synchronization:
  - **Baseline synchronization**: Initial catch-up operation to align with the Source’s resources
  - **Incremental synchronization**: Remain synchronized as the Source’s resources evolve
  - **Audit**: Destination determines whether it effectively is in sync with the Source
    - Coverage of resources
    - Bitstream accuracy
Solution Perspective - Source

• Source communicates about the state of its resources:

  • **Publish inventory**: snapshot of the state of resources at a moment in time

  • **Publish changes**: enumeration of resource changes that occurred during a temporal interval

  • **Notify about changes**: send notifications as changes occur

  • **Communication payload**:
    • Minimal, e.g. HTTP URI of resource
    • Additional, e.g. datetime of change event, content-based hash of resource
Publish Inventory - Resource List

• In order to meet a Destination’s need for baseline synchronization, the Source may recurrently publish a Resource List

• A Resource List enumerates resources that exist at a given moment in time

• Per resource, it minimally provides the resource’s URI

• Process:
  - Destination obtains the Resource List
  - Destination obtains listed resources by their URI
Publish Inventory - Resource Dump

- In order to meet a Destination’s need for **baseline synchronization**, the Source may recurrently publish a **Resource Dump**
  - A Resource Dump provides access to packages of resources that exist at a given moment in time
  - A Resource Dump is a list of (URIs of) packages of resources

- Process:
  - Destination GETs the Resource Dump
  - Destination GETs the listed packages by their URI
  - Destination unpacks the packages
  - Package is ZIP format with manifest
Publish Resource List: Inventory at Tx

Resource List @Tx = { A ; B ; C }
Publish Changes - Change List

• In order to meet a Destination’s need for **incremental synchronization**, the Source may recurrently publish a **Change List**

• A Change List enumerates resources that underwent (a) change event(s) during a temporal interval

• For each event, it minimally lists URI of the resource, the nature of the change

• Process:
  - Destination obtains the Change List
  - Destination obtains created/updated resources by their URI, removes deleted resources
Publish Changes - Change Dump

• In order to meet a Destination’s need for **incremental synchronization**, the Source may recurrently publish a Change Dump

• A Change Dump provides access to packages of resources that underwent (a) change event(s) during a temporal interval

• A Change Dump is a list of (URIs of) packages of resources

• Process:
  • Destination GETs the Change Dump
  • Destination GETs the listed packages by their URI
  • Destination unpacks the packages
  • Package is ZIP format with manifest
Publish **Change List**: Resource Changes During Interval Ty-Tz

Change List \([\text{Ty,Tz}]\) = \{ A updated @Tc ; B updated @Tc ; C created @Td ; D deleted @Te ; C updated @Tf \}
Publish Changes - Change Notification

• In order to meet a Destination’s need for incremental synchronization and low latency, the Source may recurrently push out Change Notifications.

• A Change Notification enumerates resources that underwent (a) change event(s) during a temporal interval.

• For each event, it minimally lists URI of the resource, the nature of the change.

- Process:
  - Destination receives Change Notification
  - Destination obtains created/updated resources by their URI, removes deleted resources.
Send **Change Notification** – Resource Changes at Ta

\[ \text{Change Notification } [T_0, Ta] = \{ \text{A updated } @Ta \} \]
Send **Change Notification** – Resource Changes at Tb

Change Notification \([Ta, Tb]\) = \{ D updated @Tb \}
Send **Change Notification** – Resource Changes at Tc

Change Notification $[T_b, T_c] = \{ \text{A updated @} T_c ; \text{B updated @} T_c \}$
Send **Change Notification** – Resource Changes at Td

Change Notification \([Tc, Td]\) = \{ C created @Td \}
Send **Change Notification** – Resource Changes at Te

\[
Change \text{ Notification } [T_d, T_e] = \{ D \text{ deleted } @ T_e \}\]
Send **Change Notification** – Resource Changes at Tf

\[ \text{Change Notification } [T_e, T_f] = \{ \text{C updated } @ T_f \} \]
Communication Payload – Metadata & Links

- A Source may provide additional **metadata** and **links** pertaining to resources conveyed in Resource Lists, Change Lists, Change Notifications, …

- Metadata about a resource: content encoding, content length, mime type, content-based hash

- Linking to related resources: mirror copies, alternate representations, resource versions, diff between current and previous version, metadata-to-content link, content-to-metadata link, collection membership, persistent identifier, etc. Based on link relation types:
  - From IANA Link Relation Type Registry – max interop
  - URI minted by community – community interop
Further Framework Characteristics

- **Modular**: A Source does not have to implement all capabilities
  - Source decides which capabilities to support based on local and community requirements

- **Sets of Resources**: Division of a Source’s resource collection in logical groupings.
  - Supported capabilities can differ per set

- **Discovery**: Mechanisms for Destinations to determine whether and how a Source supports ResourceSync
  - Based on conventions for web discovery and documents that detail the level of support
Source: Modular Capabilities

- URI
- Metadata
  - fixity
  - links

- URI
- Bitstream
- Metadata
  - fixity
  - links
Resource List

at="t1"

Resource Dump

at="t1"

Change List

from="t1" until="t2"

Resource Dump

at="t3"

Change List

from="t2" until="t3"

Resource Dump

at="t7"

Change List

from="t3" until="t6"

Change Dump

from="t4" until="t7"
# Destination: Key Processes

<table>
<thead>
<tr>
<th>Baseline Synchronization</th>
<th>Incremental Synchronization</th>
<th>Audit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource List</td>
<td>Change List</td>
<td></td>
</tr>
<tr>
<td>URI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>URI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metadata</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bitstream</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metadata</td>
<td></td>
<td>fixity</td>
</tr>
<tr>
<td>links</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change List</td>
<td></td>
<td>fixity</td>
</tr>
<tr>
<td>Resource Dump</td>
<td>Change Dump</td>
<td></td>
</tr>
<tr>
<td>URI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>URI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metadata</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bitstream</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metadata</td>
<td></td>
<td></td>
</tr>
<tr>
<td>links</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource Dump</td>
<td>Change Dump</td>
<td></td>
</tr>
<tr>
<td>URI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>URI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metadata</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bitstream</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metadata</td>
<td></td>
<td>fixity</td>
</tr>
<tr>
<td>links</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource Dump</td>
<td>Change Dump</td>
<td>fixity</td>
</tr>
</tbody>
</table>
This ResourceSync Presentation

• Problem Domain

• Scope

• Framework – Conceptual Overview

• Framework – Technology Overview

• Implementations, Tools, Pointers
Technology Overview - Sitemaps
A Framework Based on Sitemaps

• Sitemap is the core format throughout the framework
  
  ○ Reuse Sitemap format for all capability documents: Resource List, Resource Dump, Change List, Change Dump, Change Notifications, and manifest in Dumps
  
  ○ Introduce extension elements and attributes:
    - In ResourceSync namespace (rs:) to accommodate synchronization needs
  
  ○ Utilize Sitemap Index format where needed
Sitemap Format

<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>http://example.com/res1</loc>
    <lastmod>2013-01-02T13:00:00Z</lastmod>
  </url>
  <url>
    <loc>http://example.com/res2</loc>
  </url>
  ...
</urlset>
Sitemap Index Format

```xml
<sitemapindex xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <sitemap>
    <loc>http://example.com/sitemap1.xml</loc>
    <lastmod>2013-01-02T13:00:00Z</lastmod>
  </sitemap>
  <sitemap>
    <loc>http://example.com/sitemap2.xml</loc>
    <lastmod>2013-01-02T14:00:00Z</lastmod>
  </sitemap>
  ...
</sitemapindex>
```
ResourceSync Sitemap Extensions

```xml
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9" xmlns:rs="http://www.openarchives.org/rs/terms/">  
  <rs:ln .../>  
  <rs:md .../>  
  <url>  
    <loc>http://example.com/res1</loc>  
    <lastmod>2013-01-02T13:00:00Z</lastmod>  
    <rs:ln .../>  
    <rs:md .../>  
  </url>  
  <url>  
    <loc>http://example.com/res2</loc>  
    <rs:ln .../>  
    <rs:md .../>  
  </url>  
  ...  
</urlset>
```
ResourceSync Sitemap Extensions

```xml
<sitemapindex xmlns=http://www.sitemaps.org/schemas/sitemap/0.9
xmlns:rs="http://www.openarchives.org/rs/terms/">
  <rs:ln …/>
  <rs:md …/>
  <sitemap>
    <loc>http://example.com/sitemap1.xml</loc>
    <lastmod>2013-01-02T13:00:00Z</lastmod>
  </sitemap>
  …
</sitemapindex>
```
## Resource Metadata Summary

<table>
<thead>
<tr>
<th>Element/Attribute</th>
<th>Description</th>
<th>Defined by</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;loc&gt;</td>
<td>Resource URI (identity)</td>
<td>sitemaps</td>
</tr>
<tr>
<td>&lt;lastmod&gt;</td>
<td>Timestamp of last change</td>
<td>sitemaps</td>
</tr>
<tr>
<td>&lt;changefreq&gt;</td>
<td>Expected update frequency</td>
<td>sitemaps</td>
</tr>
<tr>
<td><a href="">rs:md</a></td>
<td></td>
<td>ResourceSync</td>
</tr>
<tr>
<td>change</td>
<td>Change type (Change List &amp; Change Dump Manifest only)</td>
<td>ResourceSync</td>
</tr>
<tr>
<td>encoding</td>
<td>HTTP Content-Encoding header value</td>
<td>RFC2616</td>
</tr>
<tr>
<td>hash</td>
<td>One or more content digests (md5, sha-1, sha-256)</td>
<td>Atom Link Ext.</td>
</tr>
<tr>
<td>length</td>
<td>HTTP Content-Length header value</td>
<td>RFC4287</td>
</tr>
<tr>
<td>path</td>
<td>Path in ZIP package (Dump Manifests only)</td>
<td>ResourceSync</td>
</tr>
<tr>
<td>type</td>
<td>HTTP Content-Type header value</td>
<td>RFC4287</td>
</tr>
</tbody>
</table>
## Link Relation Type Summary

<table>
<thead>
<tr>
<th>Relation</th>
<th>Use in ResourceSync</th>
<th>Defined in</th>
</tr>
</thead>
<tbody>
<tr>
<td>rel=&quot;alternate&quot;</td>
<td>Link from generic to specific URI</td>
<td>HTML 5</td>
</tr>
<tr>
<td>rel=&quot;canonical&quot;</td>
<td>Link from specific to generic URI</td>
<td>RFC6596</td>
</tr>
<tr>
<td>rel=&quot;collection&quot;</td>
<td>Resource is member of collection</td>
<td>RFC6573</td>
</tr>
<tr>
<td>rel=&quot;contents&quot;</td>
<td>Link from dump to manifest</td>
<td>HTML4</td>
</tr>
<tr>
<td>rel=&quot;describedby&quot;</td>
<td>Has metadata</td>
<td>Protocol for Web Description Resources (POWDER): Description Resources</td>
</tr>
<tr>
<td>rel=&quot;describes&quot;</td>
<td>Is metadata for</td>
<td>The 'describes' Link Relation Type</td>
</tr>
<tr>
<td>rel=&quot;duplicate&quot;</td>
<td>Mirror or alternative copy</td>
<td>RFC6249</td>
</tr>
<tr>
<td>rel=&quot;.../rs/terms/patch&quot;</td>
<td>A patch -- efficient change information</td>
<td>This specification</td>
</tr>
<tr>
<td>rel=&quot;memento&quot;</td>
<td>Link to time-specific URI</td>
<td>Memento Internet Draft</td>
</tr>
<tr>
<td>rel=&quot;timegate&quot;</td>
<td>Link to timegate</td>
<td>Memento Internet Draft</td>
</tr>
<tr>
<td>rel=&quot;via&quot;</td>
<td>Provenance chain, came from</td>
<td>RFC4287</td>
</tr>
</tbody>
</table>
# Link Attribute Summary

<table>
<thead>
<tr>
<th>Element/Attribute</th>
<th>Description</th>
<th>Defined by</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="">rs:ln</a></td>
<td></td>
<td>ResourceSync</td>
</tr>
<tr>
<td>encoding</td>
<td>HTTP Content-Encoding header value</td>
<td>RFC2616</td>
</tr>
<tr>
<td>hash</td>
<td>One or more content digests (md5, sha-1, sha-256)</td>
<td>Atom Link Ext.</td>
</tr>
<tr>
<td>href</td>
<td>Related resource URI (identity)</td>
<td>RFC4287</td>
</tr>
<tr>
<td>length</td>
<td>HTTP Content-Length header value</td>
<td>RFC4287</td>
</tr>
<tr>
<td>modified</td>
<td>Timestamp of last change (c.f. &lt;lastmod&gt;)</td>
<td>Atom Link Ext.</td>
</tr>
<tr>
<td>path</td>
<td>Path in ZIP package (Dump Manifests only)</td>
<td>ResourceSync</td>
</tr>
<tr>
<td>pri</td>
<td>Priority of link</td>
<td>RFC6249</td>
</tr>
<tr>
<td>rel</td>
<td>Relation - IANA registered or URI</td>
<td>RFC4287</td>
</tr>
<tr>
<td>type</td>
<td>HTTP Content-Type header value</td>
<td>RFC4287</td>
</tr>
</tbody>
</table>
Technology Overview – Resource List & Change Notification Capabilities
Publish Inventory – Resource List

http://www.openarchives.org/rs/resourcesync#DescResources
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
   xmlns:rs="http://www.openarchives.org/rs/terms/>
<rs:md capability="resourcelist"
   at="2013-01-03T09:00:00Z" />
<ul>
   <loc>http://example.com/res1</loc>
   <lastmod>2012-10-02T13:00:00Z</lastmod>
   <rs:md hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      length="8876"
      type="text/html"/>
</url>
<url>
   ...
</url>
</urlset>
Resource List Index <resourcelist_index.xml>

<sitemapindex xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
               xmlns:rs="http://www.openarchives.org/rs/terms/">

  <rs:md capability="resourcelist"
          at="2013-01-02T09:00:02Z"/>

  <sitemap>
    <loc>http://example.com/resourcelist1.xml</loc>
    <rs:md type="application/xml"/>
  </sitemap>

  <sitemap>
    <loc>http://example.com/resourcelist2.xml</loc>
    <rs:md type="application/xml"/>
  </sitemap>

</sitemapindex>
Resource List <resourcelist1.xml>

<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
        xmlns:rs="http://www.openarchives.org/rs/terms/>

<rs:ln rel="index"
      href="http://example.com/resourcelist_index.xml"/>

<rs:md capability="resourcelist"
        at="2013-01-03T09:00:00Z"/>

<url>
  <loc>http://example.com/res1</loc>
  <lastmod>2012-10-02T13:00:00Z</lastmod>
  <rs:md hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
        length="8876"
        type="text/html"/>

</url>
...
</urlset>
Publish Changes – Change Notifications

http://www.openarchives.org/rs/notification
Motivation for Notifications

- Reduce synchronization latency by having the Source push out resource change information
- Ongoing publication of Change Notifications works nicely in combination with recurrent publication of Resource Lists
- Avoids continuous pull of Change Lists by Destinations
Notifications Channels

- Notification sent via subscription channel
  - One channel per set of resources

- Payload for notifications: <urlset> documents

- Transport protocol for notifications:
  - W3C WebSub (formerly known as PubSubHubbub) - https://www.w3.org/TR/websub/
Change Notification Payload

```xml
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
       xmlns:rs="http://www.openarchives.org/rs/terms/">
  <rs:ln rel="up" href="http://example.com/dataset1/capabilitylist.xml">
  <rs:md capability="changelist-notification"
         from="2013-01-03T00:00:00Z"
         until="2013-01-03T00:10:00Z"/>
  </rs:ln>
  <url>
    <loc>http://example.com/res2</loc>
    <lastmod>2012-10-02T09:07:00Z</lastmod>
    <rs:md change="created"
           hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
           type="application/pdf"
           datetime="2013-01-03T09:07:00Z"/>
  </url>
  ...
</urlset>
```
Technology Overview – Discovery of Capabilities
Discovery of Capabilities

```
<urlset>
  <sitemapindex>
    Source Description
    capabilitylist
    resourcelist
    changelist
    resourcedump
  
  Resource List
  Capability List
  Change List Index
  Resource Dump
```

@hvdsomp - ResourceSync
OAI10, Geneva, Switzerland, June 21 2017
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
   xmlns:rs="http://www.openarchives.org/rs/terms/">
  <rs:md capability="description"/>
  <rs:ln rel="describedby"
     href="http://example.com/info_about_source.xml"/>
  <url>
    <loc>http://example.com/dataset1/capabilitylist.xml</loc>
    <rs:md capability="capabilitylist"/>
    <rs:ln rel="describedby"
       href="http://example.com/dataset1/info_about_dataset1.xml"/>
  </url>
  <url>
    <loc>http://example.com/dataset2/capabilitylist.xml</loc>
    <rs:md capability="capabilitylist"/>
    <rs:ln rel="describedby"
       href="http://example.com/dataset2/info_about_dataset2.xml"/>
  </url>
</urlset>
Capability List

```xml
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
         xmlns:rs="http://www.openarchives.org/rs/terms/>
<rs:md capability="capabilitylist"/>
<rs:ln rel="up" href="http://example.com/.well-known/resourcesync"/>

<url>
  <loc>http://example.com/dataset1/resourcelist.xml</loc>
  <rs:md capability="resourcelist"/>
</url>

<url>
  <loc>http://example.com/dataset1/change/</loc>
  <rs:ln rel="hub" href="http://hub.example.org/pubsubhubbub"/>
  <rs:md capability="changelist-notification"/>
</url>

<url>
  <loc>http://example.com/dataset1/resourcedump.xml</loc>
  <rs:md capability="resourcedump"/>
</url>
</urlset>
```
Discovery of Capabilities

Source Description

Discovery (./well-known/resourcesync)

Capability List

resourcesync

Discovery (Link: rel="resourcesync")

Resource List

Sitemap

Sitemap-compatible

Discovery (./robots.txt)

http://www.openarchives.org/rs/resourcesync#Discovery
Framework Navigation

Source Description

Capability List

Resource List

Change List Index

Change List

Resource Dump

Change List

Change List

Change List

[urlset>]

<sitemapindex>
e.g. Capability List

```xml
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
    xmlns:rs="http://www.openarchives.org/rs/terms/">
    <rs:md capability="capabilitylist"/>
    <rs:ln rel="up"
        href="http://example.com/.well-known/resourcesync"/>
    <url>
        <loc>http://example.com/dataset1/resourcelist.xml</loc>
        <rs:md capability="resourcelist"/>
    </url>
    <url>
        <loc>http://example.com/dataset1/changelist.xml</loc>
        <rs:md capability="changelist"/>
    </url>
    <url>
        <loc>http://example.com/dataset1/resourcedump.xml</loc>
        <rs:md capability="resourcedump"/>
    </url>
</urlset>
```
Technology Overview – Linking to Related Resources
Cases Detailed in the Spec

Provide links to related resources to address specific resource synchronization needs.

1. Mirrored content with multiple download locations
2. Alternate representations of the same content
3. Patching content rather than replacing it
4. Resources and metadata about resources
5. Prior versions of resources
6. Collection membership of resources
7. Republishing synchronized resources

• All cases are handled with a <rs:ln> element referring to the linked resource
• Obviously, additional relationships can be expressed
Linking – Alternate Representations

Alternate representations of the same content

This may be of interest for:

- Resources subject to HTTP content negotiation
- Format migration for preservation reasons
- Different clients wanting different formats
- Multiple languages of the content

http://www.openarchives.org/rs/resourcesync#AltRep
Linking – Alternate Representations

```xml
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
        xmlns:rs="http://www.openarchives.org/rs/terms/">
  <rs:md capability="changelist"
           from="2013-01-02T09:00:00Z"
           until="2013-01-03T09:00:00Z"/>

  <url>
    <loc>http://example.com/res1</loc>
    <lastmod>2013-01-02T13:00:00Z</lastmod>
    <rs:md change="updated"/>
    <rs:ln rel="alternate"
           type="text/html"
           href="http://example.com/res1.html"/>
    <rs:ln rel="alternate"
           type="application/pdf"
           href="http://example.com/res1.pdf"/>
  </url>

</urlset>
```
Linking – Alternate Representations

```xml
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
        xmlns:rs="http://www.openarchives.org/rs/terms/">
  <rs:md capability="changelist"
           from="2013-01-02T09:00:00Z"
           until="2013-01-03T09:00:00Z"/>
  <url>
    <loc>http://example.com/res1.html</loc>
    <lastmod>2013-01-02T13:00:00Z</lastmod>
    <rs:md change="updated"/>
    <rs:ln rel="canonical"
           href="http://example.com/res1"/>
  </url>
</urlset>
```
Linking – Metadata about Resources

Resources and metadata about resources

• Metadata resources are resources like any other; they have a URI

This may be of interest when:

• Resources have associated descriptive metadata records, which are useful for understanding the resource
  • Such as cultural heritage images, audio, video

• Resources that have associated technical, administrative, rights metadata

http://www.openarchives.org/rs/resourcesync#ResMDLinking
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
    xmlns:rs="http://www.openarchives.org/rs/terms/">
    <rs:md capability="changelist"
        from="2013-01-02T09:00:00Z"
        until="2013-01-03T09:00:00Z"/>
    <url>
        <loc>http://example.com/res1</loc>
        <lastmod>2013-01-02T13:00:00Z</lastmod>
        <rs:md change="updated"/>
        <rs:ln rel="describedby"
            type="application/xml"
            href="http://example.com/metadata/res1.xml"/>
    </url>
</urlset>
Linking – Metadata about Resources

```xml
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
        xmlns:rs="http://www.openarchives.org/rs/terms/>
<rs:md capability="changelist"
        from="2013-01-02T09:00:00Z"
        until="2013-01-03T09:00:00Z">
  <url>
    <loc>http://example.com/metadata/res1.xml</loc>
    <lastmod>2013-01-02T13:00:00Z</lastmod>
    <rs:md change="updated"/>
    <rs:ln rel="describes"
           type="text/html"
           href="http://example.com/res1"/>
  </url>
</urlset>
```
This ResourceSync Presentation

- Problem Domain
- Scope
- Framework – Conceptual Overview
- Framework – Technology Overview
- Implementations, Tools, Pointers
CORE Use Cases

» Very scalable implementation on both the server and client side
» Interpretation of metadata happens using existing pipeline at the aggregator.
» 1.5 million OA publications from Elsevier, Springer and others already exposed.

Available at: https://publisher-connector.core.ac.uk/

@hvdsomp - ResourceSync
OAI10, Geneva, Switzerland, June 21 2017
CORE Use Cases

» Will be a game changer ...
» Advocated by COAR Next Generation Repositories WG

Key publishers (OA + hybrid OA)
- ELSEVIER
- Wiley-Blackwell
- Springer
- Taylor & Francis
- Many others

Publisher connector
- PDF
- XML

A range of bespoke APIs

OA Repositories

OA Journals

ResourceSync

Mostly OAI-PMH

OMTD-SHARE (over REST)

openMIN7ED

UNSILO

IRIS.AI

@hvdsomp - ResourceSync
OAI10, Geneva, Switzerland, June 21 2017
EHRI Use Case

- Aggregation of information about Holocaust collections
  - held by 1,800+ organizations worldwide
  - into a central service
  - EAD as exchange format

- Diversity of data sources and locations
  - databases, spreadsheets (“home collections”)

https://ehri-project.eu/
http://portal.ehri-project.eu
EHRI Use Case

- Special ResourceSync implementation
  - Bridges gap between local systems and ResourceSync capability documents on a web server
  - Filters local resources by subject, time period, etc
  - Set up by EHRI technical staff, operated by contributing party

- Baseline synchronization: Resource Lists
- Incremental synchronization: Change Lists
- Together with EAD files moved from local system to web server
  - Dropbox, FTP, USB stick

- Service: partners expose EADs, server collects and offers value-added services e.g., graph database

https://ehri-project.eu/
http://portal.ehri-project.eu
CLARIAH Use Case

• Various institutions host evolving collections
  • Make collection items uniformly available via RDF graph
  • Central registry holds description of all collections

• Researchers use Virtual Research Environment to
  • Discover collections (via registry)
  • Collect graphs from respective institution
  • Keep graphs up to date

https://www.clariah.nl/
CLARIAH Use Case

- Baseline synchronization
  - Download graph from DB
  - Serialized as one or more files, one RDF triple per line (+ s p o graph_name)
  - + stands for “add”
  - URIs of files listed in Resource List

- Incremental synchronization
  - Changes logged in one or more files, one change per line (+/- s p o graph_name)
  - + stands for “add”, “-” for delete
  - URIs of files listed in Change List

https://www.clariah.nl/
https://twitter.com/CLARIAH_NL
ResourceSync Tools

- Source implementation
  - Python
  - DANS & LANL & Open University CORE
  - Connectors to file system, Solr index
  - Exposes Resource Lists, Change Lists
  - OAI-PMH converter planned
  - Resource Dump, Change Dump planned
  - https://github.com/resourcesync/py-resourcesync
ResourceSync Tools

- Client implementation
  - Python
  - https://github.com/resync/resync

- Aggregator implementation
  - Phyton
  - https://github.com/EHRI/rs-aggregator
  - Will be documented, moved to https://github.com/resourcesync/py-resourcesync

- Notification implementation
  - W3C WebSub
  - https://github.com/resync/resourcesync_push