ReadingList

Background

This particular paper talks about identifiers, how they ought to be constructed and
maintained, and how to increase their network effects.
Architecture of the World Wide Web

Of course we have been thinking about this for a while; Adam, George, and
others presented at the Metadata Working Group:
2002 MDWG
and summarized the main issues, problems, and possibilities.

Issues in Permanent Identifiers.

Here is a workshop on PI?s.

[DCC Workshop on Persistent Identifiers
30 June ? 1 July 2005| http://www.dcc.ac.uk/training/pi-2005/]

Lots of excellent papers.

What are some use cases ? where do we need persistent identifiers?
1) Preservation ? finding the thing we have preserved.
2) Long term referencing ? independent of delivery system. For example, user should be able to reference particular ?May anti-slavery pamphlet? independent of what system delivers it.

Questions:

1) Are we talking about a permanent *locator* :something which will always get you to a particular resource?
2) Or, are we talking about an *identifier*,: something which will always identify a specific resource, but may, or may not, get you to the resource.

This distinction is made, and I think importantly, in this document:
http://www.cdlib.org/inside/diglib/ark/

3) The ERL reference talks about the trustworthiness of the reference.
Evidently their references include some sort of signature in the reference, which is then compared with the signature of the object. This proves that the user is pointed to what they planned to point to.

SYSTEMS

ARK

To quote: (http://www.cdlib.org/inside/diglib/ark/)
The Archival Resource Key (ARK) identifier is a naming scheme for persistent access to digital objects (including images, texts, data sets, and finding aids), currently being tested and implemented by the California Digital Library (CDL) for collections that it manages.
An identifier is an association between a string (a sequence of characters) and an information resource. That association is made manifest by a record (in the case of this service, a METS record) that binds th[e identifier string to a set of identifying resource characteristics. The ARK identifier is a specially constructed, globally unique, actionable URL. Each ARK links end-users to three things:

- Digital object metadata
- Digital object content files
- A commitment statement made by the CDL concerning the digital object.

And this:
http://www.sspnet.org/files/public/Kunze.htm

And the latest (Feb. '06) IETF Draft:
http://www3.ietf.org/internet-drafts/draft-kunze-ark-11.txt

PURL

http://purl.oclc.org/docs/inet96.html

But PURL are actually persistent locators ? that is after all their name. A PURL does not offer any service except that of resolution to its destination. There is no way to get metadata about the object, and no way to determine an institutions commitment to maintain a PURL. No way to store alternate, or backup URL?s, and no way to provide context specific resolution.

Handles

http://www.handle.net/introduction.html
The LOC has this writeup on Handles: http://lcweb2.loc.gov/ammem/award/docs/h-s2.html
http://www.handle.net/rfc/rfc3650.html
http://www.handle.net/rfc/rfc3651.html

DOI?s ? I think DOI?s are an implementation of handles, but commercially, so each DOI costs money. Other than that, I don?t see any difference between doi?s and handles.
To quote:
The DOI System is an application of the Handle System to intellectual property. The DOI is managed and developed by the International DOI Foundation, a not-for-profit membership organization. The DOI system adds to the Handle System an approach based on structured associated metadata; policies regarding scope and application; procedures for ensuring consistency and quality control across applications; business models; and specific application tools. Initial implementations are now being supplemented by increasingly sophisticated value-added tools for metadata management and content management, which will use the Handle System multiple resolution function. More information is available in the DOI Handbook.

Handles can store arbitrary data:
http://www.handle.net/hs_manual/server_manual_2.html#SEC3
says:
A handle has a set of values assigned to it and may be thought of as a record that consists of a group of fields. Each handle value must have a data type specified in its <type> field, that defines the syntax and semantics of its data, and a unique <index> value that distinguishes it from the other values of the set. A set of handle data types has been pre-defined for administrative use. (See Handle System Namespace and Service Definition.)
<type> can be any UTF8-string. Handle System users acknowledge, however, that there are potential conflicts for handle clients if users assign types that are not registered and recognized across the user community. How <types> should be defined and how they should be used is currently under discussion. The non-administrative types that have been registered and defined to date are listed below.

- URL: Values of type URL are UTF8-encoded URIs that specify the location of the object identified by a handle.
- EMAIL: Values of type EMAIL are UTF8-encoded email addresses.

PID Pros and Cons:

Characteristic	Store URL.	Store more than 1 URL.	Metadata other than URL.	Tools for administering /configuring	Integration into other tools, browsers, etc.
PURL	Y	N	N	Y	N
Handle	Y	Y ?	Y		
ARK	Y	Y	Y		
DOI	Y	Y	Y	Y	
OpenURL		Y	Y	Y	

ERL - Eternal Resource locators

The eternal resource locator
an alternative means of establishing
trust on the World Wide Web
Ross J Anderson, Vaclav Matyas, Fabien AP Petitcolas
University of Cambridge, UK
frja14, vm206, fapp2g@cl.cam.ac.uk

3rd USENIX workshop on electronic commerce, 31 August{3 September 1998,
Boston, Massachusetts, USA, pp. 141{153. ISBN 1-880-446-97-9.

Abstract. Much research on Internet security has concentrated so far
on generic mechanisms such as firewalls, IP authentication and protocols
for large scale key distribution. However, once we start to look at
specific applications, some quite different requirements appear. We set
out to build an infrastructure that would support the reliable electronic
distribution of books on which doctors depend when making diagnostic
and treatment decisions, such as care protocols, drug formularies and
government notices. Similar requirements will be essential for other areas
of human activities such as electronic commerce.
We initially tried to implement a signature hierarchy based on X.509 but
found that this had a number of shortcomings. We therefore developed
an alternative way to manage trust in electronic publishing, that has a
number of advantages which may commend it in other applications. It
does not involve the use of export-controlled cryptography; it uses much
less computational resources than digital signature mechanisms; and it
provides a number of features that may be useful in environments where
we are worried about liability. Yet another alternative involves use of one-time signatures. We have actually implemented one-time signatures for one version of the medical publishing system. This system initially used the familiar X.509 and RSA based signature mechanisms; the move to one-time signatures enabled considerable simplifcation, cost reduction and performance improvement. We believe that similar mechanisms may be appropriate for protecting other information that changes slowly and remains available over long time periods. Book and journal publishing or legal announcements
in general appear to be strong candidates.

POI PURL-based Object Identifier

http://www.ukoln.ac.uk/distributed-systems/poi/

This document describes the PURL-based Object Identifier (POI) - a simple specification for resource identifiers based on the PURL system. The use of the POI is closely related to the use of the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) and with the OAI identifier format (oai-identifiers) used within that protocol.

The POI has been developed with the following criteria in mind:

- the use of currently deployed technologies,
- simplicity of assignment,
- the ability to assign POIs in a distributed environment without compromising the uniqueness of assigned identifiers,
- the delivery of a 'resolver' service for POIs that (where possible) builds on the existing investment in OAI repositories.

The primary intention of the POI is as a relatively persistent identifier for resources that are described by metadata 'items' in OAI-compliant repositories. Where this is the case, POIs are not explicitly assigned to resources - a POI exists implicitly because an OAI 'item' associated with the resource is made available in an OAI-compliant repository. However, POIs can be explicitly assigned to resources independently from the use of OAI repositories and the OAI-PMH if desired. As such, the POI can be seen as a possible mechanism for implementing cool URIs.

A separate document provides some POI resolver guidelines 5. All POI assigners are strongly encouraged to configure the PURL system to resolve their POIs.

http://www.ukoln.ac.uk/distributed-systems/poi/

aDore

This paper describes the aDORe repository architecture designed and implemented for ingesting, storing, and accessing a vast collection of Digital Objects at the Research Library of the Los Alamos National Laboratory. The aDORe architecture is highly modular and standards-based. In the architecture, the MPEG-21 Digital Item Declaration Language is used as the XML-based format to represent Digital Objects that can consist of multiple datastreams as Open Archival Information System Archival Information Packages (OAIS AIPs). Through an ingestion process, these OAIS AIPs are stored in a multitude of autonomous repositories. A Repository Index keeps track of the creation and location of all the autonomous repositories, whereas an Identifier Locator reflects
in which autonomous repository a given Digital Object or OAIS AIP resides. A front-end to the complete environment?the OAI-PMH Federator?is introduced for requesting OAIS Dissmination Information Packages (OAIS DIPs). These OAIS DIPs can be the stored OAIS AIPs themselves, or transformations thereof. This front-end allows OAI-PMH harvesters to recurrently and selectively collect batches of OAIS DIPs from aDORe, and hence to create multiple, parallel services using the collected objects. Another front-end?the OpenURL Resolver?is introduced for requesting OAIS Result Sets. An OAIS Result Set is a dissemination of an individual Digital Object or of its constituent datastreams. Both front-ends make use of an MPEG-21 Digital Item Processing engine to apply those services to OAIS AIPs, Digital Objects, or constituent datastreams that were specified
in a dissemination request.
http://comjnl.oxfordjournals.org/cgi/rapidpdf/bxh114v1.pdf

OpenURL

OpenURL 1.0 for Reptilian Brains

http://outgoing.typepad.com/outgoing/2006/03/openurl_10_for_.html&nbsp;