pilin persistent identifier linking infrastructure

web: http://resolver.net.au/hdl/102.100.272/0N8J991QH
email: policy@pilin.net.au

Version History

Version	Date	Status & changes	Expression identifiers
V1.0	2007-09-24	Initial draft for public discussion.	PILIN/NQNDLPDQH hdl:102.100.272/NQNDLPDQH

PILIN Guideline

Identifier Service Guidelines

This document is a work in progress and may contain open questions not resolved during the timeline of the PILIN project. It represents the thinking of the PILIN team as at December 2007.

To cite the *latest* version of this work use http://resolver.net.au/hdl/102.100.272/1KKBLPDQH

To cite *this* version of this work, use http://resolver.net.au/hdl/102.100.272/NQNDLPDQH

1 Purpose/Issue

This guidelines document outlines considerations in setting up and presenting services operating on identifiers.

2 Background

2.1 Service

A service operating on an identifier is an action operating on that identifier through some defined protocol for requests and responses. Services are hosted by computer systems. The system hosting the service on an identifier is often the identifier management system, but this is not always the case. Services may be built on other services; in particular, value-added identifier services are services hosted and operating outside the identifier management system, which are built on top of services provided by the identifier management system.

- Example: A content delivery system uses identifiers as its retrieval keys. The data source delivering content is distinct from the identifier management system. In that case, the identifier management system provides a resolution service, mapping identifiers to retrieval keys. That retrieval key is then used on the data source to deliver content. So the content delivery is a value-added service, using a direct identifier service to enable a service specific to a local concern.

Service Request

A service request is an encoded parameterised request for a service from a service host. The parameters specify the entities involved in the request, and are themselves encodings. So a service request for an identifier service serialises encodings of identifiers for the host, for the service, and the identifier operated on. (The identifier for the host and the service are usually considered as a unit.) Other identifiers, such as authentication, are optional. The encodings for these three identifiers need not be consistent (and rarely are). The encoding for the identifier as a service parameter may not be similar to the encoding of the identifier in isolation; this is often because the service encoding constrains what identifiers it operates on, so a full specification of identifier context may be unnecessary.

- Example: The URL http://www.arrow.monash.edu.au/hdl/1951.1/2395 is a service request for identifier resolution:
  - The serialisation of the three identifiers required in the service request is slash-delimited concatenation.
  - *http://www.arrow.monash.edu.au* is the identifier for the service host, as a URL.
  - *hdl/* is the identifier for the service of Handle resolution. "hdl/" is a label; its context is supplied by the service host, so that the URL http://www.arrow.monash.edu.au/hdl fully identifies the resolution service.
  - *1951.1/2395* is the identifier to be resolved. It is not encoded as a URL. However, the identifier is constrained to be encoded with URL-safe encodings, as it is embedded within a URL. (It is an implementation choice whether the slash is encoded as %2F.) The context information that the identifier is a Handle (e.g. hdl: or info:hdl:) would be expected were the identifier presented in isolation; here it is left out: the Handle resolver already specifies that its parameters are Handles.

2.2 Actionable Identifier

An actionable identifier is an identifier that may have an action associated with it, and that action can be associated in the form of a service request. It is important to distinguish the actionable identifier from the service request. The service request binds the identifier to one particular service, encoding, and protocol. The identifier has broader application than the

service request, even if the service request is for a basic operation such as resolution.

- The Handle 102.100.272/0N8J991QH is actionable through the Handle system, which defines resolution services for it. But the service request http://hdl.handle.net/102.100.272/0N8J991QH binds the Handle to one particular instance of one particular service: it is not the only context in which the identifier can be used, and should therefore not be used interchangeably with the Handle itself. Other service requests are just as legitimate uses of the identifier with different services instances:
  - http://resolver.net.au/hdl/**102.100.272/0N8J991QH**
  - http://oai-pmh.example.com?verb=GetRecord& metadataPrefix=oai_dc& identifier=**info%3Ahdl %3A102.100.272%2F0N8J991QH**
  - http://openurl.example.com?url_ver=Z39.88-2004& rft_id=**info%3Ahdl%3A102.100.272%2F0N8J991QH**

  —restricting identifiers to the one service will prevent those other services using the identifier in their own way.

In many actionable identifier schemes, the identifier is so tightly bound to the corresponding resolution service that the two cannot be differentiated.

- URLs were originally defined as locators; they are now defined as URIs which "in addition to identifying a resource, provide a means of locating the resource by describing its primary access mechanism (e.g., its network 'location')" (RFC 3986). Both functions are fulfilled by the same URI, and there is no resolution service component ("means of locating the resource") in a URL that can be differentiated from a pure identifier component. The same applies to PURLs.

In such identifier schemes, the resolution service is implicit in the presentation of the identifier as a URL.

- URIs are not restricted to URLs, but can be names independent of any resolution service. To make sense of the distinction between name URIs and locator URIs, we can consider the URI to have no resolution service defined for it, and the locator URI to have an implicit resolution service defined for it through protocols like http. (RFC 3986: "The URI itself only provides identification; access to the resource is neither guaranteed nor implied by the presence of a URI. Instead, any operation associated with a URI reference is defined by the protocol element, data format attribute, or natural language text in which it appears.")
- If passed to an OpenURL or OAI-PMH service request, as illustrated above for Handles, a URL becomes actioned through that service rather than through HTTP; so the HTTP protocol is

ignored, and the URL is treated as a URI name independent of resolution.

3 Scope

These guidelines relate to services parameterised on identifiers, and presented for consumption to actors outside the curation boundary of the identifier management system. They are not constrained to services provided or hosted by the identifier management system. The guidelines are restricted to services realised through computer systems across a computer network, but are not restricted to Web Services, nor to any particular convention such as SOAP or REST.

4 Guidelines: Implicit and Explicit Services

4.1 Service Request vs. Actionable Identifier

The identifier and the service request for the identifier serve different purposes. For computer processes, identifiers are only meaningful if they are actioned through some defined service. So the identifier will not appear outside the context of those services, and will only be presented to processes through service requests. At a higher level, the identifier coordinates the various services that can act on it, and can be presented as an abstraction outside of any services acting on it. So the identifier itself is more important in information management; this includes archival use and other domains where the persistence of the identifier is not tied to any one service.

The differentiation between an identifier and a resolution service request on an identifier is unfamiliar to contemporary users of the Web, as URLs have conflated the two, using the locator as a non-persistent identifier. The expectation on the Web is therefore that any presentation of an identifier is actionable (in particular, resolvable) without any intervening steps. If an identifier is separated in the identifier system model from an identifier resolution requests, there are several ways to deal with this expectation:

- Continue to bind identifier presentation to a resolution service, whose behaviour may or may not be defined for a particular instance. This is the approach of PURL and ARK; in the case of ARK, though, the resolution service is only a default, and can be overridden through a suffix.
- Have an implicit resolution defined for any presentation of the identifier; this can be resolution but may change according to context. This is the policy now recommended for URI, as seen:

URIs can be presented as resolvable through HTTP, but need not be. Applied to a system differentiating identifiers from resolution more clearly (like Handle), this would mean presenting the identifier as is in citing it, but binding it to an explicit service when rendering the citation.

- Allow a choice of presentations between identifiers and identifier resolution requests. This is the policy PILIN has chosen to deal with Handles. Canonical citation of identifiers uses both a canonical presentation of the identifier itself, and a separate encoding of a resolution request for the identifier through a canonical host. This can lead to complexity and inefficiency, and PILIN allows for only one of the two presentations to be used in particular contexts.

4.2 Encoding Services

Identifiers are embedded in service requests in often idiosyncratic ways.

Even though URI queries (RFC 3986 §3.4) define a syntax for service requests, that syntax is not always observed in URI encodings of service requests, which may use simple concatenation or embedding instead.

- URI query: http://oai-pmh.example.com?verb=GetRecord& metadataPrefix=oai_dc& identifier=**info%3Ahdl %3A102.100.272%2F0N8J991QH**
- Concatenation: http://resolver.net.au/hdl/**102.100.272/0N8J991QH**

The advantage of URI queries is that they are explicit: they separate the identifier from the rest of the service encoding formally, and tend to use identifiers with an explicit, well-defined context (e.g. the Info-URI registry for OpenURL). XML encodings of service requests are even more explicit in this way. Simple concatenation or embedding is more succinct, and is not dependent on explicit registration of identifiers.

The flexibility of embedding can lead to some latitude in interpreting how the identifier is delimited from the service, and thereby the scope of the identifier. This flexibility is useful if the implementer has to deal with legacy identifier scopes inconsistent with the current system. In general, however, a well-delimited identifier encoding should be preferred, to prevent confusion for subsequent users.

- Example: in ARK, the string "??" concatenated to an identifier constitutes a request for the permanence .policy associated with that identifier. So in the ARK http://ark.nlm.nih.gov/ark:/12025/psbbantu?? , the identifier is ark:/12025/psbbantu , the service host ("name mapping authority hostport") is http://ark.nlm.nih.gov/ , and the service is "??". In this case, it is only the conventions of ARK, and not of URI itself, that allow us to separate "??" from the identifier.

- Example: The National Library of Australia has its own persistent identifier scheme for pictures, which are accessed through URI queries; e.g. http://www.nla.gov.au/apps/cdview?pi=nla.aus-vn4200235 . Particular transformations of pictures are accessed by appending suffixes to the URI query; e.g. http://www.nla.gov.au/apps/cdview?pi=nla.aus-vn4200235-s1-v (side 1, view), http://www.nla.gov.au/apps/cdview?pi=nla.aus-vn4200235-s2-e (side 2, examine). If the URI query is taken on face value, then the transformations are being identified separately through identifiers meaningfully related to the citation-level identifier.

  Because of the flexibility of URI presentations of services, one could also impose a different information model onto these URIs, in which identifiers are always arbitrary, and the suffixes present transformation services operating on a base identifier, and not novel identifiers (e.g. "-s2-e" is the service "examine side 2 of…"). However, because the URI syntax interprets nla.aus-vn4200235-s2-e as a single parameter, it is likeliest to have been intended as a single identifier, and other interpretations may lead to confusion. Since the URI query syntax is a well-established standard, its interpretation of service requests should be preferred.

4.3 Single vs. Multiple Services

Because of the strong traditional binding between identifiers and resolution, users often expect only one resolution service, with one type of behaviour, to be available for an actionable identifier. This is the way URLs and PURLs behave. This expectation ties in with the default information model that any thing that can be used on its own should have its own identifier—including transformations and realisations of other things. So a work will have one identifier, resolvable through a resolution service common to that identifier scheme; the PDF manifestation of the latest version of the work will have a different identifier, resolvable through the same resolution service, and discoverable from the work identifier through a relationship service.

If we are not constrained to a single resolution service, and allow service access to the transformations of a thing, then we need not commit our information model to separate identifiers for all transformations of a work and a relationship service. Instead, we can identify only the top-level, citable thing, and have all transformations of it accessed through exposed, well-defined services. This increases the value of the top-level citable identifier, and allows it to drive a range of business processes without needing the intermediate infrastructure of a relationship service. Using transformation services on a top-level identifier allows the benefits of data encapsulation present in object- and service-oriented design; not coincidentally, it is also consistent with current repository design, in which

the transformations are available from the top-level object as disseminations.

There are other reasons why we may choose to avoid the single resolution service for an identifier: institutional branding, load sharing, policy constraints on using external services (cf. RQF policy on Handle citation).

There may still be practical reasons why we are constrained to use a single resolution service instead of offering multiple resolution services.

- As discussed below, there is a strong notion of a canonical resolution service among users. Multiple resolution services undermine this notion, particularly if they are presented as resolutions with different authority, and may encounter resistance. (They do not give the "real" resolution, and do not have an obvious validating mechanism.)
- Multiple resolution services may require a level of cooperation with the identifier management system which might not be forthcoming. The distinct resolution services need not be hosted by the same identifier management system, but they will likely need access to information from the system.
- We do not have enough control over the presentation of identifiers to guarantee that our resolution service, rather than the default, will be invoked.
- Though multiple resolution services dispense with the need for a relationship service to act as an identifier directory, they may need management through a service directory, and they may not be readily discoverable.
- Protocols interacting with a system may be constrained to use identifiers instead of service calls. For instance OAI-PMH requires an identifier for anything harvested; a thumbnail harvest would require distinct identifiers for thumbnails and for content pictures to be harvestable.

4.4 Persistence of Services

As seen, an actionable identifier is typically presented as a service request for the action on the identifier. If the identifier is persistent, there is a user expectation that all presentations and qualities of the identifier are also persistent. So a service request for resolution of a persistent identifier is itself expected to be persistent. That means that the identifier for the service and the service host used should be persistent. But because the service may not be managed directly through the identifier management system, the persistence of the identifier does not guarantee the persistence of the service: the persistence of the service is guaranteed by a third party.

Some identifier schemes uncouple identifier association from identifier actionability, as seen (e.g. Handle). But someone still needs to guarantee persistence of identifier actionability for that identifier to remain usable, especially if the identifier will be cited as actionable (e.g. embedded in a PDF as a hyperlink). This separate guarantee is strongly emphasised in the design of the ARK identifier scheme, and is also a concern in Handle, as branded, institution-specific resolution services have started to be used.

This means that parties using actionable persistent identifiers should make arrangements for the services realising those actions to be persistent themselves, and to expose the accountability of those services the same way the accountability of the identifiers is exposed. The same general principles for persisting a service apply as for persisting an identifier: meaningful and branded names should be avoided; there should be handover arrangements in case the current host of the service becomes unavailable; a large institution or consortium of institutions provides a better guarantee of longevity than an individual or small institution; and so forth.

Because any identifier can have a third-party resolver out of the control of the identifier management system, the identifier system should also expose the fact that its resolution service is canonical, and allow users access to mechanisms to verify accurate resolution (such as message digesting).

5 Appendix

- RFC 3986: Berners-Lee, T., Fielding, R., Masinter, L. 2005. *Uniform Resource Identifier (URI): Generic Syntax*. RFC 3986. http://www.ietf.org/rfc/rfc3986.txt
- Kunze, J. 2007. *The ARK Persistent Identifier Scheme*. http://www.ietf.org/internet-drafts/draft-kunze-ark-14.txt

Copyright © Monash University

This work is licensed under the Creative Commons Attribution-Share Alike 2.5 Australia License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/2.5/au/

This work was created as part of the PILIN project. The PILIN project is funded by the Australian Commonwealth Department of Education, Science and Training, (DEST) under the Systemic Infrastructure Initiative (SII) as part of the Commonwealth Government's Backing Australia's Ability – An Innovation Action Plan for the Future (BAA) under the ARROW Project.