Re-discovering repository architecture; adding discovery as a key service

Peter Sefton

Australian Digital Futures Institute, University of Southern Queensland, Toowoomba, Australia

peter.sefton@usq.edu.au

This article proposes an architecture for institutional repositories (IRs) which is more service oriented and distributed than the typical view of a repository as a monolithic software application. The discussion is informed by first-hand case-studies from work conducted in the Australian IR scene. While this paper is grounded in real, mostly antipodean, examples it is relevant globally; the issues discussed and solutions proposed here are important to many repositories.

The discussion is in three parts. Firstly, there is a short outline of the core functions of IRs in the Australian context and how they are viewed and modeled by practising repository practitioners; repositories are widely perceived and discussed now as monolithic systems, a view which contrasts with work by repository and digital library strategists, that suggest they be viewed as a collection of services with an accompanying governance model. Secondly, selected key issues in repositories are illustrated with reference to experience working with Australian repository managers; in each of these cases I show how a reinterpretation of the monolithic architecture could help to make the repository better able to meet its goals. Finally, the architectural features suggested in the second part are drawn together into a proposed layered architecture for repository systems. The two key components of this model are (a) an *index* component, populated by a smart, scriptable indexer which can index multiple disparate systems on and off campus, together with (b) a discovery layer which consists of one or more web portals that use the index to provide a single point of access for both machine and human data consumers.

This exposition suggests more work on reconsidering the institutional repository as a kind of *institution* in its own right; analogous to a library, rather than the current habit of conflating the term 'repository' with a software application.

Keywords: institutional repositories, OAI-PMH, digital preservation

Introduction

As Clifford Lynch noted in a recent talk on Revisting Institutional Repositories, to talk about 'repositories' you need to know who you are talking to. For this paper the broad working definition of an Institutional Repository (IR) comes from the RUBRIC toolkit (RUBRIC was a project led by the University of Southern Queensland and funded by the Australian Government under the Systemic Infrastructure Initiative):

> IRs centralize, preserve and make accessible the knowledge generated by academic institutions, and form part of a larger global system of repositories which are indexed in a standardised way and searchable using a common interface. IRs store electronic resources regardless of type or format, for example text, images, sound, data and, being institutionally sponsored, provide ongoing storage and access beyond the life of an individual computer, research project or organisational unit.

Generally speaking the current focus of Institutional Repository *managers*, in Australia is above-all on compliance with government requirements for reporting of research activity, with an underlying commitment to open access to research output materials. The IR is usually owned by the library, with some integration to systems in the research office in some cases*.

In 2003 Lynch talked about an institutional repository as a 'set of services':

> In my view, a university-based institutional repository is a set of services that a university offers to the members of its community for the management and dissemination of digital materials created by the institution and its community members. It is most essentially an organizational commitment to the stewardship of these digital materials, including long-term preservation where appropriate, as well as organization and access or distribution.
> (Lynch)

This view is widely held by repository theorist and informs modelling such as the recent Open Library Environment (OLE) Reference Model (OLE Project) But for most working repository managers in Australia, for practical purposes, the repository is the *software application* installed at their institution, not the *organization* of people and systems that Lynch identifies as part of the repository, or even a mesh of services.

This article has two main points:

1. It will make a case, via concrete examples that we should work hard to broaden the view of the repository, back to something in the spirit of Lynch's definition as a set of services.
2. To that end the article will propose a flexible software architecture, with one key component, an over-arching index of the content in different systems which drives a discovery service, or discovery layer.

* The source for this evaluation is my involvement in the CAUL Australian Institutional Repository Support Service (CAIRSS) – where our staff have daily contact with repository managers and technical staff and have compiled lists of priorities for repository managers. Kate Watson and I will write-up our findings for publication soon.

This article is aimed at those who are working with the day to day realities of managing an IR for whom theoretical service-based models do not reflect daily business. The discussion starts with some small case studies, or vignettes of real-life repository issues which can be vexing for on-the-ground repository managers. These include issues around persistent identifiers for repository items and people, local versus federated repository views, and metadata standardization.

In each case, I argue that a more relaxed, inclusive definition of the 'repository' would help to improve the functionality of the repository. The approach in this paper, building an argument from case studies, complements the ongoing attempt to try to formalize the design process for software by building ontologies of service-types and standards as part of the e-Framework (eFramework contributors). An e-Framework based approach describing a similar architecture is presented in a paper by Warwick Cathro (Cathro) using the eFramework.

Case studies / vignettes

Persistent identifiers

One of the very vigorous debates in web-theory and in the repository world is around best-practice for naming things. To simplify the debate, on one hand there is a 'pure-web' position that HTTP URI's, that is URLs, are adequate for identifying resources, as argued by Norman Walsh (Walsh) On the other there are the proponents of schemes which attempt to abstract naming infrastructure further than the DNS system which underlies URLs, notably via the Handle system (Sun), (although in practice Handles are typically used to cite resources in the form of a URI which redirects to the resource upon resolution, thus handles are used very similar to URL redirection schemes in use). In Australia, the government-funded PILIN project produced some guidelines and a number of services for identifiers built around the Handle system.

Amongst institutional repositories the URL approach is exemplified by Southampton's ePrints repository where the URL is the main identifier. This item is a presentation by Leslie Carr with the identifier http://eprints.soton.ac.uk/12468/. If the goal of the repository is that the URIs used to name resources must *persist* then the maintainers at Southampton have to make sure that if their software changes, or their domain name changes, perhaps via a merger or re-branding of the institution, some kind of redirect service is put in place when people try to use the URL-based identifier.

On the other hand, visiting a repository that is in the Handles camp, a resource looks like this. In large text the page invites users to refer to the item using a handle-powered URL other than the one appearing in the address bar in their browser:

> Please use this identifier to cite or link to this item:
> http://arrow.monash.edu.au/hdl/1959.1/43305

Users may ignore the request to cite or link using the supplied URL and bookmark the URL they see in the address bar. At the time of writing, capturing the item into the Zotero reference manager also results in the deprecated URL being stored for the item. Thus, the Monash repository service needs to maintain the application-specific URL which people *will* use by copying from the address bar, but *also* keep up the handles subscription and associated management.

I speculate that one reason that this design has come about is a tendency to think of the repository as a computer application rather than a people-supported service, where the assumption is that adding the Handle 'feature' will ensure persistence. Trying to get users to refer to page using a URL other than the one showing in the address bar runs against normal web practice and increases the maintenance load on repository staff as handles must be maintained as well as redirect services for plain URLs. Thus using the Handle system in this particular way has actually increased the risk that the repository will not meet its goals by increasing the number of things that needs to be maintained,

I propose that two useful design principles can be abstracted from this, which reinforce the position put by members of the PILIN project that persistent Identity services are primarily a matter of policy and governance (Nicholas, Ward, & Blinco):

1 **Do not confuse governance issues and usability with technological solutions**.

2 **Avoid conflating the term "repository" with the application at the centre of your repository lest it lead you to add features that have unforeseen costs.** (Where the extra cost here is to do with maintaining service on two URLs instead of one).

While policy is all important, there are potential software services that might help ease the load on IR managers in managing URL persistence. Specifically, I am proposing that an index of the content of the the repository and other web applications could be created, populated by a web crawler or an OAI-PMH feed . This index would contain all the metadata housed in the repository, including the URLs on which items are served and any other identifying tokens. If the index were maintained beyond the life of individual applications such as Eprints this would effectively provide a map of all the content that has or might have been served by web systems over time.

An index could be used to build an important service, one which could locate content which has moved, or is not longer available in all the contexts in which it used to be. Such a service was described by the RIDIR project, a 'Broken link resolver service' (Green 46-47) which uses an index to attempt to find an authoritative match for a piece of content that has moved, and if that is not available offers the user a range of options based on searching for items with similar metadata. One mechanism that can assist in this approach is OpenUrl (Van de Sompel, Beit-Arie, & Van de Sompel, Herbert).

In a similar vein to RIDIR the PILIN project in Australia proposed the Persistent Citation Resolver Service, which does a reverse-lookup on the handles database to locate URLs; the handle system could be used to implement and index, as it is able to store metadata.

An important extra benefit of this index-driven approach to assisting in URL management would be the ability to associate one ID with more than one URL – there are cases where the same item might be served by an repository application on a number of different URLs as part of a different search or browse context. The context in which a paper is linked might vary between its use in courseware, in a portfolio or as part of an IR; an identifier management system should be able to register URLs and other identifying metadata from each of these contexts and be able to resolve the identifier to another home for a resource when one of the contexts is no longer available.

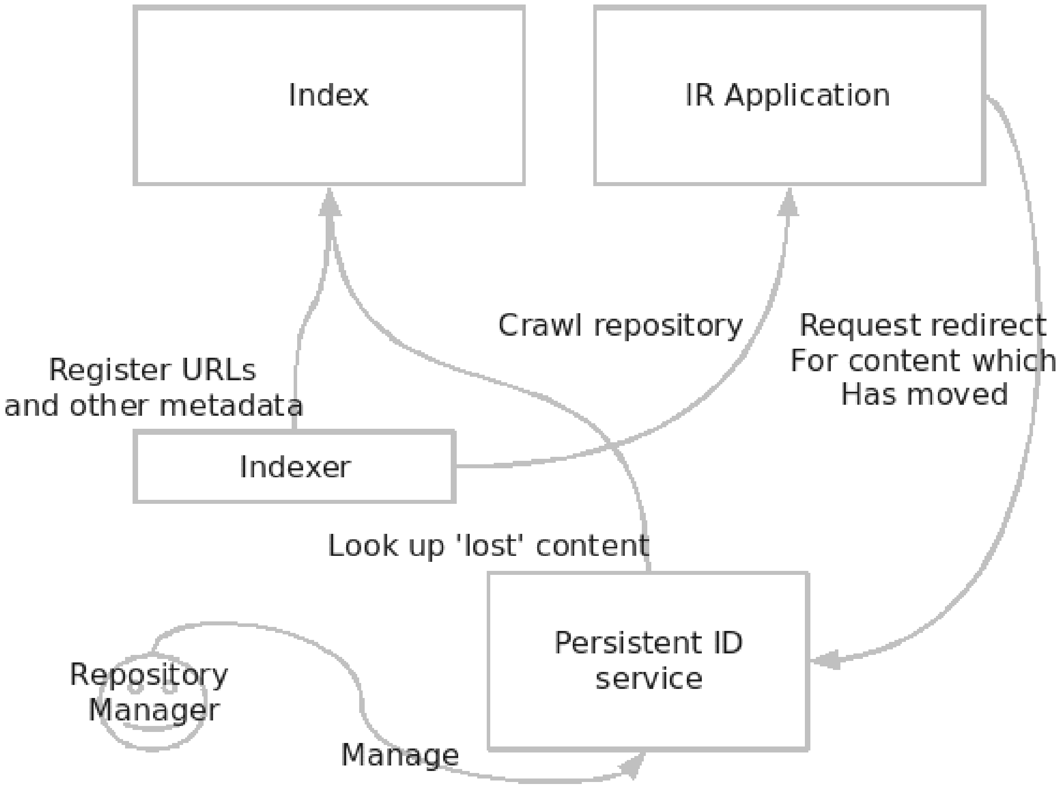

Illustration 1: An architectural suggestion, removing persistent ID concerns from the IR into a separate service informed by an index

Persistent Identifiers for people

Just as there is a widely recognized requirement for items in a repository to have reliable and persistent identifiers, so it is recognized that identifying people is also important. In the library world, this is usually done via name-authority files and the usual practice is to use a single canonical form of a name, making authority control the most expensive part of cataloguing (Tillett 24). In IRs usual practice is to record names as they appear on a work (Salo), but not all IR software has a way to manage name-ids for all people mentioned in metadata, being able to reliably associate the right identity with the works with which they are associated while preserving differing name-forms. In a survey of the state of the art in name-authority Salo also notes the cost of name-authority data; it must be managed by hand and it is usually is not present at all in IRs; she gives an example of the range of name-variants that appear in the OAISTER discovery service and describes the shortcomings of contemporary repositories form a usability point of view. Salo couches the problem in terms of a lack of name authority control, but it can also be looked at in terms of a lack of Identity management.

One repository software package that Salo does not discuss is Fez (Kortekaas), from the University of Queensland, this allows for both preservation of the form of the name used on a work, and an local identifier for an author. This shows the range of name-forms for one author:

> **Refine**
> **Names this author has published as**
> Hunter, Jane (51)
> Hunter, J. (24)
> Hunter, J (7)
> Hunter J. (4)
> Jane Hunter (1)
> http://vmdev-repo.library.uq.edu.au/view/UQ:65771

But as the ID for Jane Hunter (UQ:65771) is a local one, there is still a need for a way to match-up records held in other repositories, a point which is now widely recognized. A workshop in Amsterdam in early 2009 resulted in a working group to coordinate international efforts on distributed identity management– this wiki document, which is subject to change captures some of the recommendations and work needed: http://repinf.pbworks.com/Interoperable-identification-infrastructure.

For repositories where name-identity is not firmly established, an Australian project aims to assist; the NicNames application, which promises to allow repository managers to manage name-identities for the contributors in their repositories using an application which sits outside of the repository itself. The system will let a repository manager batch-load metadata from a repository and try to identify individuals uniquely. NicNames will use clusters of subject-codes and other cues in metadata to create a semi-automated identity service (Sefton, “NicNames and People Australia –

some thoughts for CAIRSS « CAIRSS Blog”). main options for repository integration.

Once established, the name data could be loaded back into the repository application but as Salo notes, a lot of repository software does not have ways to use the new identities. Repository staff at Swinburne University of Technology where the NicNames application is being developed tell me that they would not at this stage be able to manage a process where they lock-down their repository, use NicNames to establish name identity, reconfigure the repository to deal with the new ID information then re-load the data, because the repository software in use doesn't have an existing capacity to handle this. However, they still see value in collecting and creating the data for identifiers for use in new records, and in the future, their retrospective records (when they work out how).

The point of having name IDs as well as name strings is to be able to show records from the same author grouped together regardless of the spelling of their name in a citation. Technically, whatever the platform this involves setting up indexes on the name ID. At Swinburne the indexer configuration *is* open to the library staff but it cannot use data which might be in a name-authority service because it can only 'see' metadata in the repository storage layer.

I am proposing that a key component that should be included in a repository (in the broad sense) is a *smart* indexer which is scriptable, and *is* able to call out to other services. In the case of the Swinburne repository service the library staff would be able to include NicNames ID's if the indexer were smarter and configurable even if they were unable to integrate the name-identity service with their core repository software.

The following architectural sketch shows a potential practical solution to the problem – the library staff who maintain it would be able to configure their system so that tight integration between the IR application and the NicNames name management application was not necessary, however desirable. A smart indexer could be configured so that upon indexing each record in a repository it would call-out to the names system to get an ID based on a lookup of the item ID, and a name string.

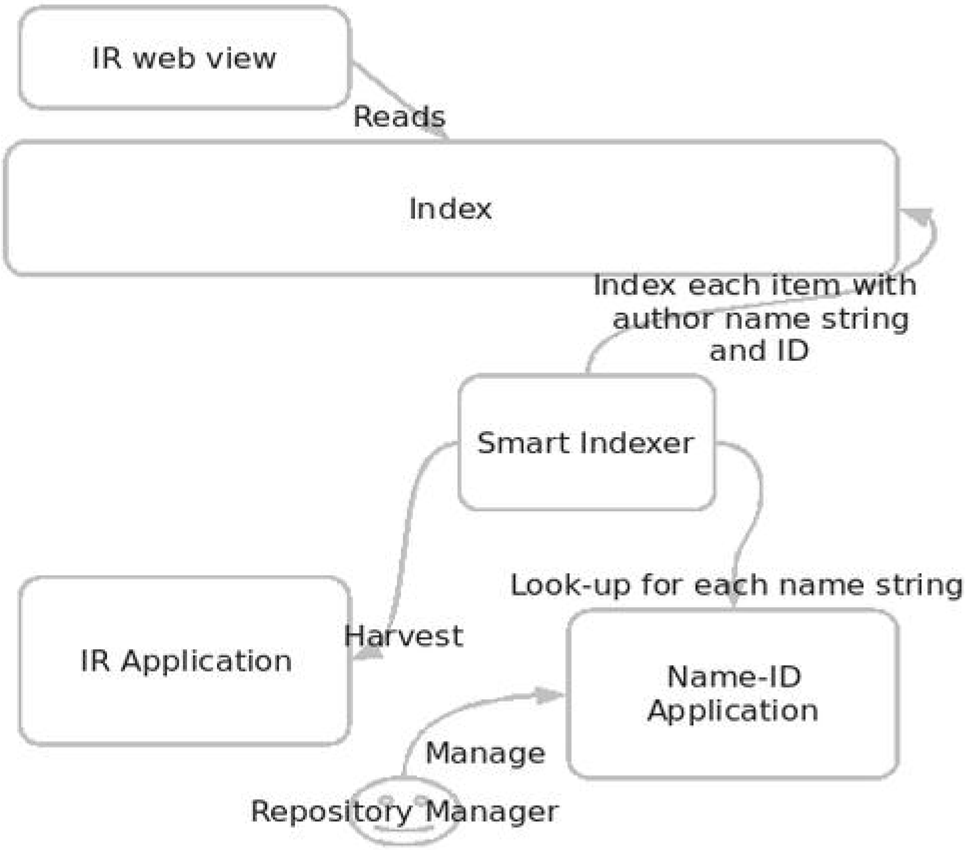

Illustration 2: Tying together an IR with a name-ID management system using a smart, configurable indexer to relate name strings to IDs

In the previous case, item identifiers the key component was the index, so that item requests could be serviced, but here another component is required, a web interface to the index so that web-users can see a view of the repository to is built from an amalgamation of item data and name data.

My team at USQ has built on the work of others to provide such a smart-indexer and web interface as part of a software application know as The Fascinator (Sefton & Lucido). It was originally conceived as a proof-of-concept application funded by the ARROW project (Treloar & Groenewegen), and took the form of an indexing component for the Fedora (Lagoze et al.) repository back-end. The Fascinator is built using an very impressive piece of open source software from the Apache Foundation called Solr ("Welcome to Solr") which is widely used in libraries to provide catalogue and other resource discovery services.

The Fascinator's indexer allows custom scripting in the Jython language, meaning that as each item is indexed it is easy to do calculations or look-up other data sources to generate index terms. It can be used to extract embedded metadata from images, or to generate metadata such as access control information based on flexible criteria; for example 'if the item is of type *thesis* or *working paper* then set the *access* field to be *on-campus'* (where the default value is *guest*). In portals sitting on top of the index, access control is handled by limiting off-campus users to guest access, via a limit-query/filter on the index. Current work on The Fascinator is focussed on making it more modular, so that it can be used to index any data source, including file system storage, web sites via a crawler and OAI-PMH feeds.

In the example of an author with multiple name-strings the item id of UQ: 65771 and name-string "Hunter, Jane" would return an author ID that could be used in constructing a name-aware view of repository contents. The entry in the `dc_creator` index could be structured like this:

```
/id/canonical-name string-/actual-string
```

/65771/Hunter, Jane/Jane Hunter

Using this scheme a lightweight web application could show the canonical name:

```
+ Hunter, Jane (87)
```

With a '+' sign to expand to the view:

```
+ Hunter, Jane (87)
   Hunter, Jane (51)
   Hunter, J. (24)
   Hunter, J (7)
   Hunter J. (4)
   Jane Hunter (1)
```

Local vs shared or federated views & Metadata standardization

Metadata from IRs is frequently aggregated into federations allowing people to search across repositories. In Australia, the most notable example is the Australian Research Online service, hosted by the National Library of Australia (NLA) using an Apache Solr index. This service uses the OAI-PMH protocol to pull metadata records from a number of repositories into a coherent view. For example a search for *climate change* shows:

Type

- journal article (749)
- conference paper (154)
- thesis (78)
- report (72)
- book chapter (68)
- more

There are a number of resource types, with 749 journal articles topping the list. But, the same search on a test server also running Apache Solr, over an out-of-date harvest of a subset of the ARROW service shows a very different picture:

Type

- Journal Article (184)
- PeerReviewed (105)
- Article (75)
- Thesis (66)
- Book chapter (65)
- NonPeerReviewed (62)
- Conference Paper (35)
- Journal Articles (Refereed Article) (30)
- c1 (28)
- techreport (27)
- Full-text link or file (26)
- Conference or Workshop Item (DEST Category E) (21)
- PhD Doctorate (20)
- Article (DEST Category C) (19)
- journal article (18)

  - Book Chapter (17)
  - text (14)
  - Book Section (10)
  - Report (9)
  - Conference Publications (Full Written Paper - Refereed) (8)
  - Conference or Workshop Item (8)
  - e1 (8)
  - Book Chapters (7)
  - b1 (7)
- Book Chapter (DEST Category B) (5)

The difference between these two views is that the ARO service is a normalized harvest with one descriptor for *journal article* whereas the other one shows the raw data where we have *journal article*, *Journal Article*, *Article* and obscure resource types such as *b1* which may turn out to be articles as well. This brings up one the most vexing problems facing a repository manager; dealing with both the local demands of their institution and its practices, and the national or global view. Locally there is a need to use terms that suit institutional practice, whereas nationally there is a set of terms agreed by a group of experts. The problem is that these views of the data are completely disconnected. The local view is under the control of local staff, while the normalization, if performed at all is configured by the host of the discovery service, in this case by NLA staff. In Lynch's terms, it would be reasonable to think of the discovery service as **part of** an IR (Lynch), but at the moment that is clearly not the case.

At least one voice has suggested that the ARO discovery service and similar is of greatest use to librarians and other 'meta users':

> Our consequent analysis suggests that metadata generation, and especially ‘perfect metadata’ should take a low priority. Author- or automatically-generated metadata may well be satisfactory. [...] Federated global gateways are the primary discovery tool. to is not relevant to research outputs are discoverable via search engines. (Sale)

But there are some problems with this approach:

1. Index coverage is not always complete, as described by this discussion from the NLA (Boston) and this Dlib article (Hagedorn & Santelli).

2 The level of granularity of external indices is too coarse to get maximum use from the data. For example, the resource types listed in the above example represent a metadata field that indexes like Google do not contain – so it is not possible to build a Google-driven portal to search theses unless an intermediary service like the ARO collates a set of web pages that represent the collection to be searched. So my analysis suggests the opposite of Sale's – metadata should be given priority, and repositories should be supplemented with easily configured portal software which allows repository owners to create harvestable, and syndicatable 'slices' of the repository based on metadata queries.

My suggestion is that current IR software be supplemented with an institutional index which overlays the core IR software and other services, driven by a smart indexer, as described above in relation to name-authorities. This index would be like the the one underpinning the new Single Business Discovery System being built and piloted by the NLA (Cathro) which also uses Apache Solr and provides a single searchable index of multiple collections. This class of system is designed to 'glue together' other computer systems into a web-view

Using a system like The Fascinator, which allows simple configuration of distinct portal-views of a resource-pool, the following configuration would be possible, so that both locally relevant metadata and be presented on an institutional web view and federation-ready metadata can be presented as a web-view, via configuration files that map data from the local format to a federation-ready format.

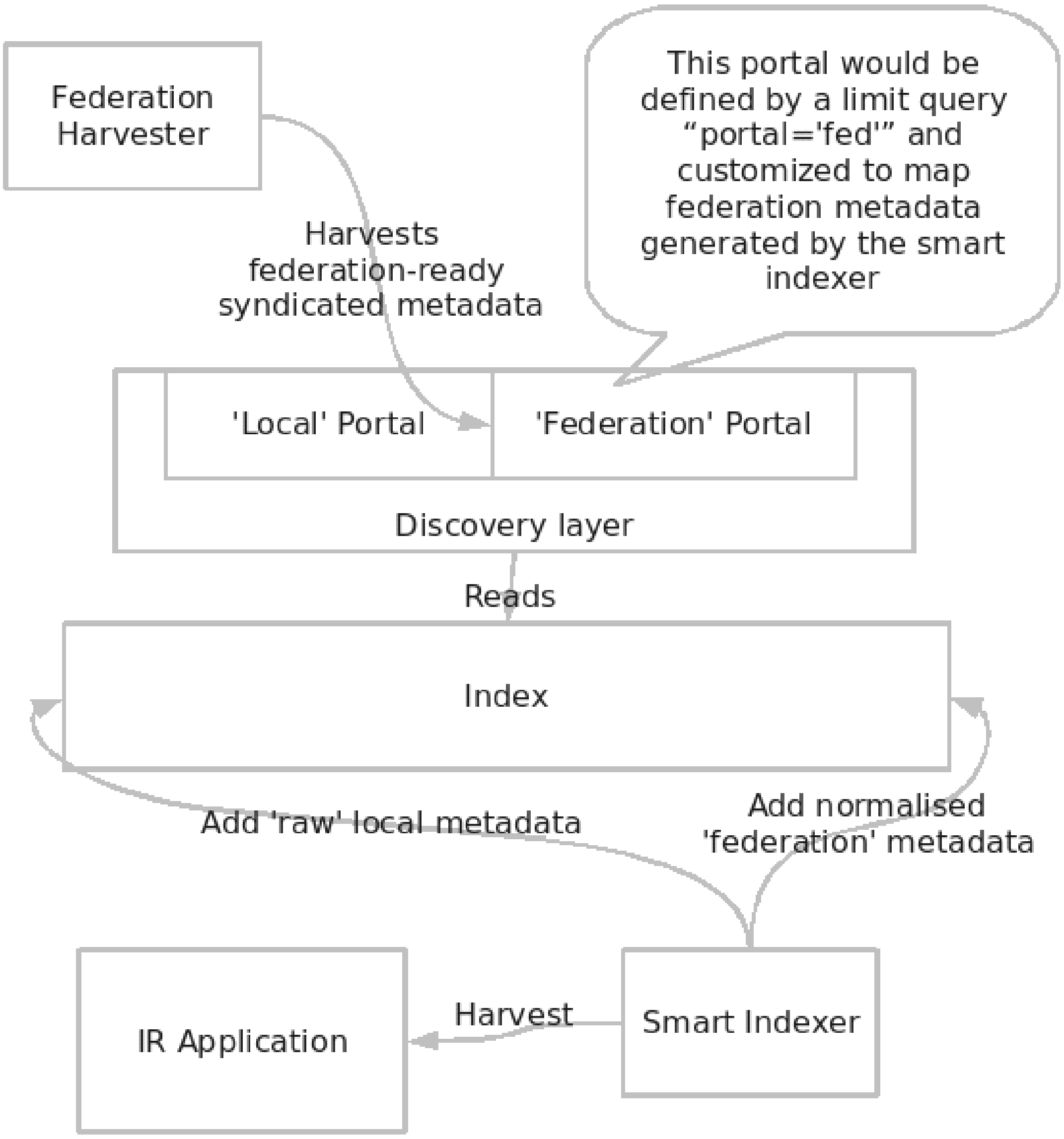
Federation Harvester
Harvests federation-ready syndicated metadata
This portal would be defined by a limit query “portal='fed'” and customized to map federation metadata generated by the smart indexer
'Local' Portal
'Federation' Portal
Discovery layer
Reads
Index
Add 'raw' local metadata
Add normalised 'federation' metadata
IR Application
Harvest
Smart Indexer

Illustration 3: An architecture to allow a local repository to have both a locally relevant portal and a federation-ready view of data under local control

A new architecture

In the above case studies I have looked at areas in which there area strong reasons to consider a distributed set of computational services working together.

The case studies above explored several points: (a) usability and governance problems caused by trying to make repository software take on 'features' that are really a matter of policy and process, and which are not particular to repository application software; (b) issues with rolling out new software or integrating systems because of resource constraints; (c) problems with local versus global metadata standardization. I have made a case for a new (to most universities) repository component consisting of an over-arching index driven by a configurable, 'smart' indexer which allows integration of data from disparate sources, such as name IDs from one system and bibliographic details in a repository software application, via an index populated by a smart indexer, configurable by library or repository IT staff without having to go through a long software update cycle, or rely on vendors.

True, this layer may represent a new application that needs to be installed, and as such represents a challenge for many repository owners with limited IT resources and protracted procurement processes, but once installed, perhaps as part of a repository-software upgrade it should allow for more flexible deployment and staged integration of services than we are seeing in Australia and avoid the issues I outlined above.

The key here is a discovery layer that can 'glue together' different services – an institutional index to supplement some of the other functions of repositories, not only helping repositories (in the broad sense) to become the collection of services they really should be, but for those services to be shared across institutional systems. So the final diagram here shows a potential architectural view of how a discovery layer might work at an institution like USQ, which has a distance education focus with course materials delivered via a Learning Management System. It is worth noting here, that the more access to institutional resources is mediated by a discovery layer the simpler it will be to manage persistent identifiers, and access control for web users; the discovery layer can act as a proxy for access to more specialised internal resources, thus minimising the number of different places that access control, syndication and integration with services like name authorities or persistent identification infrastructure need to be done, those integrations can be done that the discovery layer without having to touch individual applications.

What this means for institutional systems is:

- Individual interfaces such as an IR web site can still exist, but having a broader index behind them allows for discovery or resources which may have moved, also exist in a different context, or exist in a parallel system.

  > One good example of the latter would be at an institution like the University of Southampton which has two ePrints repositories, a fact which could present usability and management challenges, as described in a blog post of mine (Sefton, “A few discoveries”).

- For users who don't know which individual system to search an institution-wide search can direct them to where they should be looking.
- The opportunities for federating different views of an institution's content described above in the discussion about local versus global metadata would mean that discovery beyond the institution becomes more likely, via federations such as OAISTER, or generic internet search.
- The opportunity to integrate data from different systems via a configurable, programmable indexer would increase the chances that systems such as name authorities have a chance of being deployed, where the alternative is expensive direct integration of systems which may be very hard to resource.

All of the above point to a view of the repository in terms of services, rather than as a monolithic application. To a well-read repository or library specialist there is nothing new in this suggestion. The reason I am restating it is to show how it addresses the real issues being experienced by IR managers described in the vignettes above.

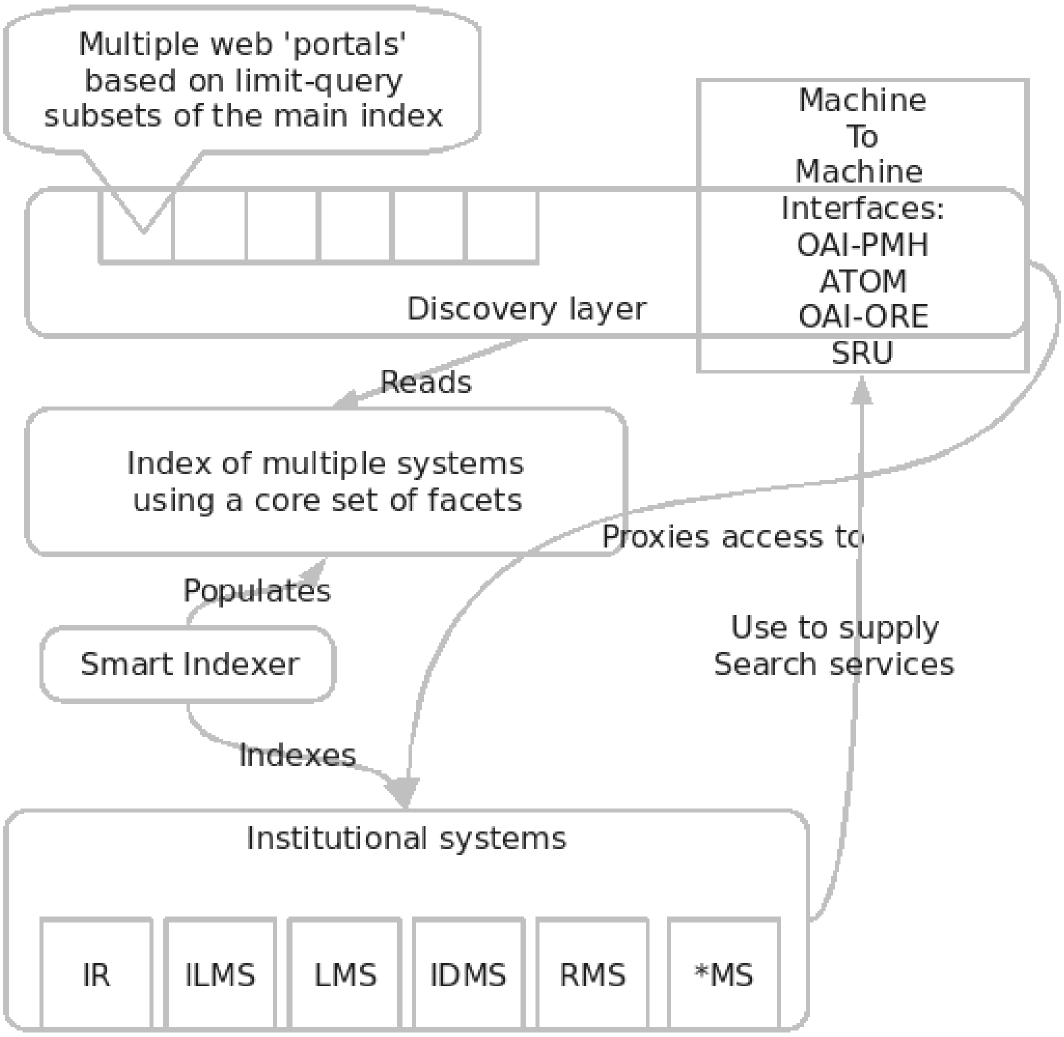
Multiple web 'portals' based on limit-query subsets of the main index
Machine To Machine Interfaces: OAI-PMH ATOM OAI-ORE SRU
Discovery layer
Reads
Index of multiple systems using a core set of facets
Proxies access to
Populates
Smart Indexer
Use to supply Search services
Indexes
Institutional systems
IR
ILMS
LMS
IDMS
RMS
*MS

Illustration 4: A discovery-layer architecture showing the IR as one of the Management Systems along with Integrated Library Management, Learning Management, Identity Management, Research Management etc.

In conclusion, it seems to me that the software architectures that are emerging from the library community (Dempsey), in which a common faceted index plays a central role, are a promising development for Institutional Repositories, and my team at the Australian Digital Futures Institute will explore this with the our library staff. Looking past software architecture to the more important issues of governance, writing this article has brought into sharp focus one of the key issues with IRs – there is a too-easy conflation of the term 'repository' with a single software application. It is very clumsy to resort to devices such as 'repository application' or 'repository-as-institution' as I have done here. Further work is needed on this with practising repository staff, rather than at the theoretical level at which debate and high-level policy setting takes place at the moment, possibly ceding the term repository to the prosaic data-storage sense, and seeking a new term for the functional, organizational, service-based view.. This will require some careful consideration of what the IR really means, and an examination of just where the repository-as-institution should sit within a broader institutional, national and global context.

References

Boston, T. "Exposing the deep web to increase access to library collections." *Eleventh Australasian World Wide Web Conference, Gold Coast, Australia. Retrieved*. 2006.

Cathro, W. "COLLABORATION IN BUILDING A SUSTAINABLE REPOSITORY ENVIRONMENT: A NATIONAL LIBRARY'S ROLE OPEN REPOSITORIES 2008, SOUTHAMPTON, 1-4 APRIL 2008."

Dempsey, L. "The library catalogue in the new discovery environment: some thoughts." *Ariadne* 48 (2006): 29.

eFramework contributors. *Overview of the e-Framework*. 2009. <http://www.e-framework.org/Default.aspx?tabid=934>.

Green, Richard. *RIDIR Final report*. Hull & London UK: University of Hull & Rightscom, 2008. <http://www.hull.ac.uk/ridir/Documents/ridir-final-report.pdf>.

Hagedorn, Kat, and Joshua Santelli. “Google Still Not Indexing Hidden Web URLS.” *D-Lib Magazine* 14.7/8 (2008). 2 Aug 2009 <http://www.dlib.org/dlib/july08/hagedorn/07hagedorn.html>.

Kortekaas, C. G. “Making Fedora easier to implement with Fez.”

Lagoze et al. “Fedora: an architecture for complex objects and their relationships.” *International Journal on Digital Libraries* 6.2 (2006): 124-138.

Lynch, C.A. “Institutional Repositories: Essential Infrastructure for Scholarship in the Digital Age.” *ARL* (2003). <http://www.arl.org/newsltr/226/ir.html>.

Nicholas, Nick, Nigel Ward, and Kerry Blinco. “A Policy Checklist for Enabling Persistence of Identifiers.” *D-Lib Magazine* 15.1/2 (2009). 14 Sep 2009 <http://www.dlib.org/dlib/january09/nicholas/01nicholas.html>.

OLE Project. “The OLE Project | Reference Model.” 2009. 16 Sep 2009 <http://oleproject.org/overview/ole-reference-model/>.

Sale, A. “A researcher's viewpoint.” *Open Access: key strategic, technical and economic aspects. Oxford: Chandos Publishing* (2006).

Salo, Dorothea. “Name Authority Control in Institutional Repositories - Cataloging & Classification Quarterly.” *Cataloging & Classification QuarterlyWhere* 47.3 & 4 (2009): 249 - 261.

Sefton, Peter. “A few discoveries.” *ptsefton* 18 Aug 2009. 18 Sep 2009 <http://ptsefton.com/2009/08/18/a-few-discoveries.htm>.

---. “NicNames and People Australia – some thoughts for CAIRSS « CAIRSS Blog.” *CAUL Australian Institutional Repository Support Service Blog* 2009. 9 Sep 2009 <http://caulcairss.wordpress.com/2009/07/21/nicnames-and-people-australia-some-thoughts-for-cairss/>.

Sefton, Peter, and Oliver Lucido. “The Fascinator: a lightweight, modular contribution to the Fedora-commons world.” Atlanta, Georgia, 2009. <http://eprints.usq.edu.au/5259/>.

Sun, S. “Establishing persistent identity using the handle system.” *Proceedings of the Tenth International World Wide Web Conference* (2001).

Tillett, B. B. “Authority control: State of the art and new perspectives.” *Cataloging & Classification Quarterly* 38.3 (2004): 23–41.

Treloar, A, and D Groenewegen. “ARROW, DART and ARCHER: A Quiver Full of Research Repository and Related Projects.” *Ariadne* 2007. 16 May 2007 <http://www.ariadne.ac.uk/issue51/treloar-groenewegen/>.

Van de Sompel, Herbert, Oren Beit-Arie, and Van de Sompel, Herbert. “Open Linking in the Scholarly Information Environment Using the OpenURL Framework.” *D-Lib Magazine* 7.3 (2001): 34826.

Walsh, Norman. “Names and addresses.” 2006. 1 Aug 2009 <http://norman.walsh.name/2006/07/25/namesAndAddresses>.

“Welcome to Solr.” 12 Sep 2007 <http://lucene.apache.org/solr/>.