Discovery Strategy Recommendation

September 2013

Sam Popowich, Discovery Systems Librarian

FINAL

Executive Summary

Discovery solutions have come a long way from the days of the OPAC and the “next-generation catalogue”. Their systems and design considerations have attempted to keep pace with the development of the web more generally, without ever being generally successful. This lack of success means that we must investigate ways in which users’ attention and workflow have changed since the development of the web. Serious consideration has to be given to exposing our data as widely as possible and making our services available within a user's workflow, rather than forcing the user to come to us. Our discovery strategies must allow for both outside-in and inside-out discovery. No single system is capable of doing this, so any strategy we undertake must combine disparate systems into a single overarching solution.

Given their lack of flexibility and customizability, as well as their cost and difficulty of installation and maintenance, none of the commercial discovery solutions are adequate on their own for a library-wide discovery strategy. However, an article-level knowledge base is indispensable for any discovery strategy to be viable and useful. The recommendation of this report, then, is to combine an existing commercial solution, preferably Ebsco Discovery Service, with a lightweight open-source discovery layer that will allow us extensive customizability as well as the ability to index any and all material we have available.

My recommendation is to continue to license Ebsco Discovery Service and use its API to present its knowledge base data as one pane of a “bento box” display driven by Blacklight. Within the Blacklight layer we will present the library website, the library catalogue, electronic journals, institutional repository and digitized content, alongside the EDS knowledge base. The library catalogue data can either be indexed within Blacklight or within Ebsco Discovery Service. My recommendation is to continue indexing and presenting the catalogue records through EDS, as users are familiar with the EDS interface for finding/accessing print materials. The discovery solution will be implemented using open-source software, lightweight and flexible design, and recognized standards and best-practices.

Recommendation

In order to move forward with the implementation of discovery strategy, I recommend that we adopt Blacklight as the basis of a discovery layer, combining Ebsco Discovery Service and Symphony catalogue content, as well as content from most if not all of our data silos. The choice of software for implementation should not be taken as determining any decisions relating to design or user-experience. Software development best-practices recommend a separation of concerns between content and presentation. I feel confident in recommending Blacklight as the system to manage and present content; presentation decisions will be made as we implement the design phase, with input from all stakeholders, including students and researchers.

Part 1. Context

The first large-scale online public access catalogues (OPACs) were implemented at Ohio State University in 1975 and the Dallas Public Library in 1978. Since then we have seen the development first of "next-generation catalogues" in the 1990s and now "web-scale discovery services". Each of these developments reflected growing trends away from the OPAC as an inventory control and location tool to a network-scale discovery and access service, a task for which the traditional OPAC was not suited.

The initial push towards "library automation" was concerned with migrating card-catalogue records to a computerized database, with the added benefit of using a patron-database to track inventory circulation. Both the creation/maintenance of catalogue cards and the management of patron and circulation data had previously been done by hand. Early interfaces to the computerized catalogue used basic pre-SQL database queries to return known-item results across a small number of indexes. These results were sometimes given a limited context by placing them in an alphabetic list of proximate records. As digital material began to be purchased or licensed, the computerized catalogue was no longer used simply to track physical inventory, but to provide search across both local and non-local material. Rather than redesign the database system to reflect changing needs and requirements, federated search was developed in an attempt to supplement catalogue searching with results from licensed platforms. One problem with federated search was its inability to integrate result sets (beyond basic sorting), but the main problem was speed: federated searching across databases using the z39.50 protocol was too slow to be a viable search option.

The next stage in the development of discovery systems was to scrap federated search in favour of a single index. Computing power achieved such low cost in the early 2000s, that creating single indexes of disparate material was possible, not only sidestepping the speed problems of federated search, but also allowing for contextual searching/browsing/faceting of different datasets. The only limit was what metadata could be supplied to the indexing system, which meant in effect the data the library had under its direct control. Publisher-supplied article metadata, for instance, was not generally included in next-generation catalogues. Even in a perfect next-generation system, then, at least two searches were needed to search local library content (print and digital) and licensed material from vendor databases. This opened up a space in which "web-scale" discovery systems could be developed.

Despite the widespread development of competing next-generation catalogues, the discovery service landscape is much smaller and homogeneous. In the January 2011 *Library Technology Report*, Jason Vaughan identified four discovery services that had risen to the top of the field: OCLC WorldCat Local, Serials Solutions Summon, Ebsco Discovery Service (EDS), and Ex Libris

Primo[1]. To a large extent, these remain the only viable all-in-one “web-scale” discovery platforms, due to their integrated article-level knowledge bases (containing both citation metadata and full-text documents). It is the knowledge-base that makes a discovery system web-scale (or network-scale) and differentiates it from a next-generation catalogue.

Many developments have taken place within the library and the world-wide-web at large. The eclipsing of federated search by unified indexing, the rise of mobile, and the advent of cloud ILS systems are three of the most significant. Participation in these trends is constrained by several factors, notably budgetary and staffing constraints. Decisions that are made regarding discovery will have to strike a balance between these two factors. In order to maintain that balance, this report will take a position that open-source software and software-development is of greater value than proprietary systems, wherever possible. Open-source software allows for greater agility and flexibility in implementation and maintenance, software reuse and modularity, and in many cases has a more responsive and open development community than proprietary software. I recognize that the lower cost (“free”) of open-source software must be balanced against greater staffing costs, but I will assume in this report that such a balance is zero-sum.

The primary requirement for a discovery service, then, is that it provide searching and faceted browsing across as many of the Libraries’ collections (both owned and licensed) as possible, presented in a single, unified interface. The interface and workflow of the system should, however, allow users the ability to quickly and easily focus their search on particular content areas. If possible, the system should be able to remember a user’s preference, and allow them to quickly limit to a particular collection or context.

The University of Alberta Libraries has, at one point or another, licensed and used three of the four discovery systems mentioned above. Currently, we continue to license and use Ebsco Discovery System, and until recently we also used OCLC WorldCat Local. Both of these systems were presented as alternative search tools to the traditional Symphony OPAC, which was presented as the default or primary search interface. In 2011, we purchased Ex Libris' Primo system and licensed access to the Primo Central knowledge base. In March 2013, the implementation of Primo was cancelled. Serials Solutions Summon is similar in many ways to Primo, in that the knowledge base was negotiated with individual serials publishers for access to metadata (as opposed to WorldCat which used the OCLC knowledge base and EDS which uses Ebscohost content).

In our case, decisions to license and implement these discovery systems were made without investigating the discovery needs of our users, the technical requirements of either system, or the best practices involved in implementing a discovery strategy. Now that we have “cleared the decks” this is a good opportunity to investigate our discovery requirements and make an informed decision regarding discovery. This report seeks to provide a basis and a framework for such a decision.

Part 2. Principles of Evaluation

[1] Vaughan, Jason. *Library Technology Reports: Web Scale Discovery Services*, January 2011.

Given the primary requirement stated above, we are left with evaluating the existing options according to a) the assessment of our own users' needs, requirements, search behaviour, and workflow and b) best-practices from the field. In the absence of a usability and discovery assessment project, I will rely on recent studies and articles to give us a sense of user behaviour as well as best-practices. The primary documents in this respect are the 2009 Discoverability Report from the University of Minnesota[2], Lorcan Dempsey's December 2012 article "Thirteen Ways of Looking at Libraries, Discovery, and the Catalog"[3], and a 2013 article out of North Carolina State University entitled "How Users Search the Library from a Single Search Box"[4]. I will supplement this material with other sources, as needed.

Trends

The University of Minnesota Discoverability Report identifies five trends drawn from user studies and statistical usage reports of their existing systems. The five trends are as follows:

"**Users are discovering relevant resources outside traditional library systems.**"
"Search, once one of the key skills and specialties of librarians, is now a daily activity for the vast majority of our users. Our users approach their research with an established history of search success that gives them confidence in their search skills." They come to the university *already knowing how* to search online. We shouldn't be trying to teach them a different way to do that.

"**Users expect discovery and delivery to coincide.**"
"Searchers do not distinguish between discovery and delivery in their web searches and increasingly find it discordant to experience this disconnect in the library environment." Our current solutions (SFX, GetIt) do very little to overcome this problem in an efficient and effective way.

"**Usage of portable Internet-capable devices is expanding.**"
"Rather than just supplementing the desktop computer, mobile devices are poised to become the primary means of Internet access for a critical mass of our users." This was in 2008, and has come true now.

"**Discovery increasingly happens through recommending.**"
"Facilitating discovery requires us to develop and implement systems that push relevant content to users and that allows users to share content with others." As we shall see, this trend

[2] Web Services Steering Committee, *Discoverability Phase 1 Final Report*. University of Minnesota, February 2009. Available at http://purl.umn.edu/48258.

[3] Dempsey, Lorcan. "Thirteen Ways of Looking at Libraries, Discovery, and the Catalog: Scale, Workflow, Attention." *Educause Online,* December 10, 2012. Available at http://www.educause.edu/ero/article/thirteen-ways-looking-libraries-discovery-and-catalog-scale-workflow-attention.

[4] Lown, Cory, Tito Sierra, and Josh Boyer. "How Users Search the Library from a Single Search Box." *C&RL*, 74:227-241 (May 2013). Available at: http://crl.acrl.org/content/74/3/227.

fits directly into Dempsey's *workflow* and *social object* points. One fundamental aspect of this is the provision of effective and intuitive permanent URLs for improved link-exchange.

“**Our users increasingly rely on emerging nontraditional information objects.”**
“The format of useful and discoverable information objects is much broader than those traditionally offered through libraries; users increasingly rely upon multimedia objects, data sets, blogs, and other 'grey' objects to meet their information needs.”

Attention and Workflow Shifts

In order to understand these trends and how they fit into the current discovery landscape, I'll turn to Lorcan Dempsey's 2012 *Thirteen Ways,* which ascribed the changes in library discovery to the rise of mobile devices and the prevalence of interoperability and search outside the library world. These wider developments led to two fundamental shifts in the behaviour of library users that libraries were not well-prepared to accommodate. Previously, information was scarce and attention abundant, but now it is the other way around (“attention switch”), which has led to a change of emphasis in search expectations from precision to recall. In addition to this, users bring search behaviour learned outside the library into the library, and they expect the library to conform to workflows learned at home, at work, and at school (“workflow switch”). Prior to the rise of the Internet, academic libraries could view their users as blank slates on which to inscribe “proper” searching techniques. Now resources and services need to be a part of the user's pre-existing workflow; we can't expect users to change successful behaviour in order to use our resources and services.

Given these shifts in attention and workflow, access and discovery expectations have also scaled up to the web- or network-level, but our data resources have not kept up. Part of the explanation for the decline of the library web-presence as the gateway to our resources (as borne out both by the Minnesota study, but also by vendor reports and our own Google Analytics) is that it is now at the wrong level or scale. Users expect the gateways to our resources to be at the network level (e.g. Google). Dempsey sees three strategies for moving our catalogue from institutional-scale to network-scale: consolidation (e.g. putting our records in HathiTrust or WorldCat), syndication and leveraging (e.g. hooking our resources into Google Scholar or Mendeley), and open-API/open-Data, where others can reuse our data and services. “While discovery and exploration may move elsewhere,” Dempsey notes, “such external discovery environments may link to the library environment for location and fulfillment”.

One barrier to moving our catalogue to the network-level is a tendency to still think of our resources and services as being consumed on a desktop or laptop computer, serving up pages of web-content in a traditional browser. The environment created by APIs and linked data means that our data and resources can be chopped up and selectively exposed in different formats and media. Libraries' attempts to move into this new environment have highlighted three issues: 1) the separation of discovery from delivery; 2) a reconfiguration of functions in app/mobile environments, where functionality is disembedded from the full website experience and re-embedded in smaller, more discrete environments; and 3) attention shift from the local to the full collection.

The library website is difficult to scale up because in fact it is a “thin integration layer” over the set of legacy and emergent management systems (ILS, resolver, repositories, etc.) and 2) the set of licensed databases, repositories, and knowledge bases. We need to provide both integrated and targeted access to both these sets of resources, which is made most difficult by the fact that some of this material can be locally hosted while the rest *must* exist only in the cloud. Providing a seamless user-interface between all of these systems is all but impossible, and it previously fell to the website and the catalogue to do the heavy lifting in this respect. Now, however, it is “apparent in the use of Drupal, VuFind, or Blacklight” that “a more integrated approach across such tools” is possible.

Single-Search, Rich Results

Another expectation users bring to library discovery is an assumption that a simple search tool should lead to a rich result experience. People no longer expect simply to be provided with a list of items in our inventory. Thanks primarily to Google, sophisticated ranking, targeted results, interjection (of related content), recommendations, and suggestions are all part of a user's expected experience. The user-experience (UX) transition from simple search to rich results has become the model for web content, and this has two challenges for libraries: 1) how do we better exploit the nature of our data to create navigable relations (e.g. making our data work harder through facets, FRBR, etc.) and 2) how do we recombine and repackage our record-based data into resources about entities of interest? From a UX perspective, libraries are trying to overcome these challenges through new ways of presenting discovery results, leveraging different device times and screen orientations to present data differently. According to Dempsey, the two primary models of presenting this information are the (traditional) single-stream method and the multiple-pane or “bento box” style (used, for example, by the National Library of Australia's *Trove* system).

Outside-In and Inside Out

All of these developments serve the purpose of trying to a) make the best use of people's attention and b) trying to conform to their workflows rather than making them conform to ours. The ability to repackage our data to make it usable outside the traditional library environment is a core element of discovery today, and this has engendered a dual-mode view of discovery. The first is “outside-in”, where material from outside the institution (books, journals, data sets, maps, etc.) are purchased or licensed and brought into the library for its users to access. This is traditionally what we mean by discovery. The new model, however, adds an “inside-out” component: making locally produced digital content (images, postcards, digitized materials) and associated metadata available outside the institution. By designing systems that allow for easy reuse of our data objects, we are able to a) place our data in the context of the entire network-level collection (i.e. make it findable in Google) and b) allow our data to be easily integrated into existing workflows that are not under the library's control.

Dempsey identifies three main areas where we can work to move our data closer to users' workflows: 1) institutional and personal curation (e.g. citation management, Goodreads, what

Dempsey refers to as “bibliographic playlists”); 2) syndication (RSS feeds, Moodle widgets, toolbars, etc.) and 3) leveraging, using a search engine or discovery environment that is outside the library's control to bring people back to the library environment (the classic example here is the LibX toolbar). For Dempsey, however, while “ideally these [systems and tools] link back through to library resources where it makes sense (...) the connective tissue to achieve this conveniently is not really in place.”

Searching Beyond the Local Collection

The desire to search beyond “the collection” raises some interesting challenges and new directions in discovery. The current trend is to integrate local holdings with cloud (or network-level) data (either through single indexing as in Primo or with registering local data with the network-provider as with EDS and WCL). The challenges with this approach are: how should integrated institution- and network-level results be presented? Other data created under different regimes will not have the same structure as catalogue data. How do we make decisions regarding mapping and normalization, authority control (or relinquishing of control) and linked data? Three possible directions alternative to “collection searching” are: 1) Curation environments (institutional or personal): e.g. resource guides, reading lists, social reading sites, citation managers, faculty profiles, etc. “Library resources sit outside of these at the moment, although it would be beneficial to be able to easily provide data to them and to be linked by them.”; 2) Services: “Library users are interested in other things than collections: library services, expertise, staff profiles, etc.”; 3) Scoped portals. “Some libraries are investing in additional local integration layers, which may allow customization to particular institutional interests. Look at the community of Blacklight users for example. Here, a local unifying layer may pull in results from a discovery layer product, a catalogue, local databases, the library website, or other resources. In this way, the library can more readily control a local experience, add additional resources, and so on.

Conclusions

Lorcan Dempsey's article provides a comprehensive way of looking at discovery, and the most important conclusions are the following:

1. Attention and Workflow switch: we need to be where our users are, and be what they are looking at.
2. We have to integrate with the network level. We can't be institution level anymore.
3. We have to make it easy for others to take and reuse our data (open-API and open-Data)
4. We have to relinquish control over our data when it leaves us (syndication and leveraging)
5. Simple Search should lead to Rich Results
6. Analytics and usability are key.
7. Outside-in and inside-out discovery
8. Various Data Structures, Services instead of Software, Scoped Portals

Clearly, any Discovery Strategy for the University of Alberta will have to focus on both outside-in and inside-out discovery, especially given the prominence and strategic importance of our digital initiatives. Discovery cannot be restricted to bringing outside materials in. However, outside-in discovery is the most pressing problem and will be the sole focus of the remainder of this report.

Part 3. Discovery Systems Environment

The systems that are currently in place that are implicated in discovery are: ILS, SFX (link-resolver), EZproxy, Library Website, Ebsco Discovery Service (incl. Ebsco Knowledge Base), LibGuides, campus mobile app, Moodle courseware widget, digital initiatives systems (Peel, ERA, ArchiveIt etc.), Non-canonical silos (Omeka, map database, etc).

We can think of discovery as a two tiered system: there's a presentation layer, and a content layer. The presentation layer is anything that is presented to the user, and the content layer is the data on which the presentation layer is based. Where possible we want to unify all the elements of the presentation layer into a single, seamless user-interface (UI).

Presentation Layer

Currently the public facing systems are: the Website, Libguides, some kind of discovery platform (here called Blacklight), Mobile, Relais, My Account, and GetIt (SFX). There are obviously, other public facing systems (databases, journals, ebook platforms, digital initiative platforms) but for the sake of this exercise, they have to be considered out of scope (they have their own native interfaces that are out of our control). Of these systems then, Libguides and Relais are hosted and so we are limited to the amount of UI design we can impose. The remainder (Website, Mobile, Blacklight, My Account) can and should be given a common look and feel using a single canonical CSS and Javascript library. NOTE: we can either integrate the website with the discovery platform or keep them as separate systems. What is important is that the user not notice that they are moving from one platform to another.

The reason the website could be integrated with the discovery platform is that it has no data model itself (other than its HTML pages). It would require no data merging to integrate it into the discovery system. My Account on the other hand, since it has its own data model, can't be integrated into the discovery platform. Again, it should have the common look and feel of the discovery platform and the website.

Content Layer

As much as possible, data should be taken out of the content providers so that it can be returned in the common-look-and-feel interface (discovery platform). The discoverable systems in the content layer are:

- ILS (Symphony)

- ERA
- Peel
- the EDS Knowledge Base
- the SFX Knowledge Base
- Libguides (surrogate records only)
- local siloed content

There are two routes to presenting this content in the discovery platform:

1. Index it all in Blacklight using the Solr index. We would not be able to index the EDS content, so this would require a separate search box for article content. The advantage here is that we have control over all the indexing of e.g. the Peel and ERA MODS records.
2. Send all the content we can to EDS to be indexed there, and we will simply provide a search interface. This will allow a unified search interface for all content (including the EDS knowledge base). The problem is that EDS will only index non-MARC content as Dublin Core, potentially resulting in a loss of metadata richness.

Mobile

One advantage of providing our own locally-hosted, lightweight, and unified web-presence is the ability to provide API hooks into the content, making mobile integration much easier and more effective.

Part 4. Components of a Discovery Strategy

Library Catalogue

The library catalogue (SirsiDynix Symphony) must remain the main tool for controlling inventory and circulation. However, the catalogue user-interface is no longer optimal for use in a modern website, and its indexing and searching capabilities are no longer fit for purpose, except among specialized groups of expert searchers. I recommend that the library catalogue records be reindexed either in the discovery layer or in the knowledge base platform for better presentation, searchability, and access.

Knowledge Base

A knowledge base is an index of citations and full-text articles and e-books licensed by the library. It is typically hosted in the cloud and accessed through a vendor's user interface, though in some cases an API is available to build a local search interface. Examples of a knowledge base are Ex Libris' Primo Central and the Ebsco collection of database content.

Non-indexed web content

Many libraries have taken to indexing their website as if it were simply another data silo. This allows users to use the same search box/search interface to find material on library hours, locations, policies, etc. Recent studies have shown that users expect the discovery search box to find this information, even when an alternative website search box is presented[5]. From a usability perspective, having two search boxes on the same page is confusing.

Index of locally held, curated, or created material
Some of the biggest silos in a library's collection are those of locally created or curated material (e.g. the institutional repository and digitized collections). These are a strategic component of the library, and their exclusion from discovery systems is problematic. In addition to providing for local discovery, an overall discovery strategy should enable the "inside-out" discovery of this material.

Search interface
The search interface is one of the trickiest discovery components to get right. Dempsey argues for a search interface that leads users "from simple-search to rich results", but recognizes that there is little consensus in the best way to present search results. One of the most modern approaches is to provide a "bento box" of search results, in which results of different types, or for different audiences, are presented in different parts of a grid. In essence, the discovery layer federates and repackages search results from various providers into the various panes of an intuitive grid.

Interoperability hooks
One of the major opportunities for libraries to improve both outside-in and inside-out discovery is the consumption or provision of an API. In consuming an API, library discovery systems can pull data in and either index that data along with data from other places, or simply repackage the data for display alongside search results (for example, using Google Maps alongside the search results for the library's location). Provisioning an API is fundamental to inside-out discovery, allowing other users and institutions to easily crawl, index, and repurpose data that the library has created or curates. This will also allow our users more easily to integrate our content into their workflow, whatever that may be.

Linking and authorization (SFX, EZproxy)
The University of Minnesota found that "users expect discovery and delivery to coincide" and they are confused when this does not happen. The simplest and most effective way to satisfy this expectation is to have a full text article either embedded or directly linked within a search result (as in a knowledge base). In practice, however, our licensing policies and our linking technologies require certain intermediate steps. These steps separate discovery and delivery in the workflow of our users, and we must try as much as possible to reduce this separation. As a result, fine-tuning of SFX (our link resolver) and EZproxy (our proxy) must be a component of any discovery strategy.

LibGuides
The same argument for indexing the library's website applies to libguides. Currently, finding a libguide on a given topic requires either direct linking, browsing, or using the libguides search interface. Including libguides in the results of the discovery layer will allow for immediate

[5] Lown, Sierra, and Boyer.

referral of users to a subject guide if that is what they need. Inclusion of the libguides will be an important component of our discovery strategy, but will require much thought and work around the questions of usability (presentation of libguide results) and metadata (appropriate subject tagging of guides for our central index).

A-Z Lists

Until now, both the e-journal A-Z and the database A-Z lists have been custom Cold Fusion applications. We need to move them to more up-to-date and dynamic systems which are easily customizable and require little or no day-to-day maintenance. There are various options available to us, but the simplest will be to provide a scoped portal within our chosen discovery layer (e.g. Blacklight).

SEO and "Inside-out" discovery (out of scope)

Finally, search-engine optimization and inside-out discovery are become integral components of a discovery system. For our immediate purposes, however, we need to implement a robust, flexible, single discovery system primarily for outside-in discovery. I recommend that inside-out discovery be placed out of this project's scope. It should form part of a future discovery project.

Part 5. Evaluation of Existing Solutions

Existing discovery solutions were rated on the following criteria:

Open Source or Proprietary: Is the system open-source (and if so, under what license) or proprietary? Is the software open to forking and pull-requests (i.e. Is it on GitHub)?

Hosting: Is the solution locally hosted, or hosted in the cloud. If locally hosted, how much hardware is required? If cloud-hosted, how responsive is the service to our change requests?

Maintainability: If locally hosted, how easy is the system to maintain? How often are updates and patches deployed, and how easy are they to install?

Customizability: What technology does the front-end use, and how easy is it for us to customize?

Flexibility of Indexing: How much of the indexing (ranking, facets, etc) is under our control? If cloud-hosted, how responsive is the vendor to indexing change requests?

Metadata: What metadata schemas are supported? Are they directly supported or do they require crosswalks?

Ongoing Development: Is the system still being developed? What is the development cycle?

Modern Technologies: What technologies and design does the system use (e.g. Java, ASP; MVC design, etc.)

Interoperability: Is there an API? Can the system consume data from other APIs? What kind of API (REST, SOAP, etc.)

Knowledge Base: does the system have a knowledge base? What is the coverage? Is the coverage actively being increased?

Cost: What is the monetary cost? What is the cost in human resources?

	EDS	Primo	Summon	WorldCat Local	Blacklight
Open-Source?	No	No	No	No	Yes
Hosting	Cloud	Cloud or Local	Cloud	Cloud	Local
Maintainability	N/A	Difficult	Unknown	N/A	Moderate
Customizability	Minimal	Minimal	Minimal	None	Fully customizable
Indexing	Inflexible	Extremely flexible, but complicated.	Inflexible	Inflexible	Extremely flexible, and easy to accomplish
Metadata	MARC and Dublin Core only	In theory can accommodate all metadata types, but in practice this is difficult to achieve	Unknown	MARC only	All metadata schemas can be implemented. Some work required.
Ongoing Development	Yes	Minimal	Minimal	Minimal	Yes
Modern Technologies	Java and Javascript	Java/JSP	Java/JSP and Javascript	Unknown	Rails and Solr
Interoperability	Yes, via an API	No	No	No	Yes
Knowledge	Yes	Yes	Yes	Yes	No

Base					
Cost	Just over $46,000 in 2013 (down from $84,000 in 2012 due to increased NEOS licensing)	Just under $75,000	Unknown, probably similar to Primo	$33,000	Free

Part 6. Conclusions and Recommendation

Given the lack of flexibility and customizability, as well as their cost and difficulty of installation and maintenance, none of the commercial discovery solutions are adequate on their own for a library-wide discovery strategy. However, an article-level knowledge base is indispensable for any discovery strategy to be viable and useful. The recommendation of this report, then, is to combine an existing commercial solution, preferably Ebsco Discovery Service, with a lightweight open-source discovery layer that will allow us extensive customizability as well as the ability to index any and all material we have available.

My recommendation is to continue to license Ebsco Discovery Service and use its API to present its knowledge base data as one pane of a “bento box” layer driven by Blacklight. Within the Blacklight layer we will present the library website, electronic journals, institutional repository and digitized content, alongside the EDS knowledge base. The library catalogue data can either be indexed within Blacklight or within Ebsco Discovery Service. My recommendation would be to continue indexing and presenting the catalogue records through EDS, as users are familiar with the EDS interface for finding/accessing print materials. The discovery solution will be implemented using open-source software, lightweight and flexible design, and recognized standards and best-practices.

In order to move forward with the implementation of discovery strategy, I recommend that we adopt Blacklight as the basis of a discovery layer, combining Ebsco Discovery Service and Symphony catalogue content, as well as content from most if not all of our data silos. The choice of software for implementation should not be taken as determining any decisions relating to design or user-experience. Software development best-practices recommend a separation of concerns between content and presentation. I feel confident in recommending Blacklight as the system to manage and present content; presentation decisions will be made as we implement Blacklight, with input from all stakeholders, including students and researchers.

Appendix 1: Design Considerations

Drawn from Bulock & Fields, “Discovery on a Budget” and Breeding, “Looking Forward to the Next Generation of Discovery Services”.

1. Names and Language: Use verbs, don't use names, especially proprietary names.
2. The order of items in lists matters, especially lists of links.
3. Be familiar: don't break common or standard UI (e.g. facets on the left).
4. Strive for comprehensive coverage: single index for everything, if possible.
5. Strive for improved relevancy.
6. Offer a single UI wherever possible.
7. Implement social discovery wherever possible.

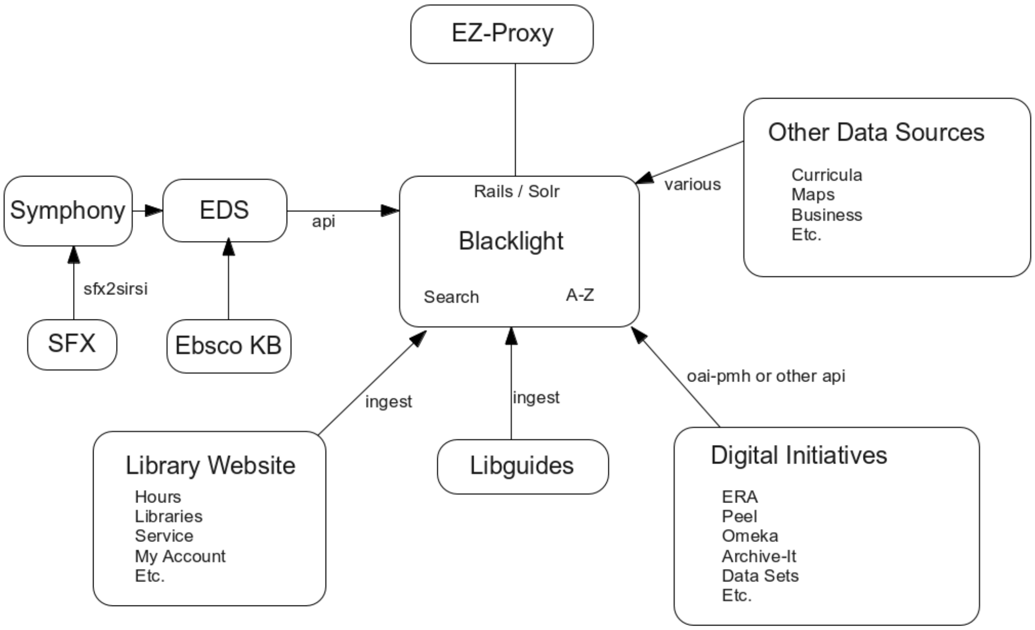
Appendix 2: Institutions with Bento Box design

Trove - The National Library of Australia. http://trove.nla.gov.au/
Stanford University - http://library.stanford.edu/
University of Calgary - http://library.ucalgary.ca/
Brown University - http://library.brown.edu/
University of Virginia - http://www.library.virginia.edu/
Johns Hopkins University - https://catalyst.library.jhu.edu/multi_search

Appendix 3: Institutions with Blacklight

Stanford University - http://library.stanford.edu/
University of Virginia - http://www.library.virginia.edu/
Johns Hopkins University - https://catalyst.library.jhu.edu/multi_search
WGBH OpenVault - http://openvault.wgbh.org/

Appendix 5: Schema of data-sources and technology



Appendix 6: Raw notes for discovery system comparison

Ebsco Discovery Service

Open-Source or Proprietary: Proprietary.

Hosting: Cloud-hosted. So far they have seemed very responsive to our change requests. However, there are constraints based on the fact that the software is the same for all institutions.

Maintainability: N/A

Customizability: Not much customization is available beyond basic branding and the adding of custom widgets (e.g. For expanding searches, libguides, chat box, etc.)

Flexibility of Indexing: EDS' indexing is not very flexible. However, they will implement indexing change requests if they are valid for most/all users. Most of the indexing is fairly black-box.

Metadata: Natively, EDS supports only MARC and Dublin Core. All other metadata schemas require crosswalking (which can result in some loss of granularity/richness).

Ongoing Development: EDS is still being vigorously updated, with a major development cycle being about a year, with some smaller cycles occurring every three months or so.

Modern Technologies: Yes, though the technologies are enterprise level. The main technology appears to be one of the Java web development frameworks.

Interoperability: EDS does provide an API, and that API is RESTful. EDS can consume data using OAI-PMH, but there's a limit to the number of feeds it can consume.

Knowledge Base: There is a knowledge base, and it covers all the Ebscohost content, as well as some other publisher metadata (e.g. Gale). The knowledge base is actively being increased.

Cost: Just over $46,000 in 2013 (down from $84,000 in 2012). Our cost goes down as other NEOS libraries implement EDS. Human cost is very little, as it is cloud hosted. It really just requires a single person to maintain the configuration, etc.

Additional notes: Anecdotally, EDS seems to be well-liked by public services staff, and Ebsco itself has always been responsive to our needs and requests for modifications and enhancements. For this reason, as well as its continuous improvement, I would recommend EDS as the web-scale knowledge base component of any discovery strategy.

Ex Libris Primo

Open-Source or Proprietary: Proprietary

Hosting: Primo can be locally hosted or cloud hosted. When locally hosted, the hardware requirements are quite extensive.

Maintainability: Difficult to maintain. Like SFX (also an Ex Libris project) the software is very large and complicated and requires knowledge of many custom scripts.

Customizability: When locally hosted, the front end is completely customizable; when cloud hosted, like EDS, only minimum customization is possible.

Flexibility of Indexing: Indexing is *extremely* flexible. However, custom indexing requires in-depth knowledge of a large and extensive set of normalization rules.

Metadata: In theory, Primo can accommodate any metadata schema, as long as you can implement the normalization rules properly. This is very difficult to do.

Ongoing Development: Again, like SFX, Primo does not seem to be undergoing constant development. Small patches and upgrades are regularly released (on a three-month development cycle), but no major changes to the software are being made.

Modern Technologies: Primo uses Java and JSP, and the design is Byzantine, to say the least. HTML, JSP, and scripting code is messy and problematic.

Interoperability: No API available; able to consume a limited number (2) of OAI-PMH feeds.

Knowledge Base: Yes, Primo Central. Also actively expanding the content.

Cost: Just under $75,000

Serials Solutions Summon

Open-Source or Proprietary: Proprietary

Hosting: Cloud hosted.

Maintainability: N/A

Customizability: Same as Primo.

Flexibility of Indexing: Similar to Primo.

Metadata: Similar to Primo.

Ongoing Development: Yes. Development cycle unknown.

Modern Technologies: Unknown – assume to be similar to Primo.

Interoperability: Unknown – assume to be similar to Primo.

Knowledge Base: Yes, also actively being expanded.

Cost: Unknown; probably similar to Primo.

OCLC WorldCat Local

Open-Source or Proprietary: Proprietary

Hosting: Cloud hosting only.

Maintainability: N/A

Customizability: Very little beyond basic branding.

Flexibility of Indexing: Not flexible; the index is controlled by OCLC.

Metadata: As far as I'm aware, Worldcat local only indexes MARC. They may be able to accommodate Dublin Core as well, but I haven't heard of that.

Ongoing Development: Yes, but infrequently. OCLC's priorities seem to lie elsewhere.

Modern Technologies: Yes, Java and JSP.

Interoperability: API available. Does not consume other web services, as it is based solely on the OCLC knowledge base.

Knowledge Base: Yes, the OCLC collection of records.

Cost: $33,000

Blacklight

Open-Source or Proprietary: Open-Source. Apache 2.0 license. GitHub repository.

Hosting: Locally hosted. Hardware costs are minimal as the software is much lighter-weight than the other locally hosted solutions.

Maintainability: Easy to maintain; the trickiest part will be keeping on top of updates/patches, but this is made easier by the fact that the software is checked in to a GitHub repository.

Customizability: Completely customizable. The front-end is Rails.

Flexibility of Indexing: Completely flexible. The index is Solr, which allows a high level of granularity in indexing. Solr is also the index behind the ERA and Peel sites, so we have local expertise with Solr indexing.

Metadata: Can ingest any metadata schema. It is up to us to map the data to the appropriate facets/fields.

Ongoing Development: Yes, Blacklight is a lively open-source project, to which developers contribute regularly.

Modern Technologies: Yes, the front end is Rails (currently being upgraded to the latest version, Rails 4.0) and the back end is the Solr index.

Knowledge Base: No. Implementation of Blacklight would require the licensing of one of the other systems for its knowledge base.

Cost: Free. Human resources costs are perhaps higher than with proprietary software, but Blacklight's light weight means that developer costs are minimal.