Thank you for the opportunity to review a pre-published version of your search comparison study article. We greatly appreciate the opportunity to provide feedback. I hope you find our response to be useful for your article and/or any discovery service evaluations occurring at Johns Hopkins.

We interpreted your goals for the study to cover some very important areas of academic library article searching services or “discovery services”: Usability, Content, API Tech. We want to address these three areas and relay some new features in our EDS API that may present a different perception of our service’s capabilities.

We also wish to clarify that the EBSCOhost “Traditional” API was built with the intent for use in federated search and single (or few) database access integration and the very specific needs for that database such as granular field data, advanced limiting, and authority/thesaurus browse. The EDS API from its conception was built for large content sets. Both APIs are able to take advantage of standard EBSCOadmin settings, such as: Search Mode (Find All, Phrase, etc.) which may not have been apparent when you conducted the study. Appendix A is a table that highlights the differences in feature sets between these two APIs. A library that would be comparing Summon or Primo to EDS would not be encouraged to utilize the EBSCOhost API in this manner, as the EDS API is the premium service they would use.

Your findings [Interpretation - page 17] were interesting to us as they show that the EBSCOhost databases themselves (including A&I only databases) are extremely important materials that should be part of any broad article discovery. EDS is the only service that incorporates these 100%.

Usability

In the early portion of your article you reference a goal “creating a better article search experience for our users” [Background and Goals – page 2], but add: “We do not know why participants preferred one tool to another” [Limitations page 11]. A study was recently completed that does go into depth to answer this “why” question. It was conducted by librarians at Illinois Wesleyan (an EDS customers) and Bucknell (a Summon customer) and is called: “Paths of Discovery: Comparing The Search Effectiveness of EBSCO Discovery Service, Summon, Google Scholar, and Conventional Library Resources” [http://crl.acrl.org/content/early/2012/05/07/crl-374.full.pdf+html]. This study focuses on student search behaviors, speed to success, quality of research in comparing a variety of services, etc. In addition to ranking EDS first, the following conclusions came from this study:

- Students using EDS required less time to complete the four searches than any of the other test groups
- Students using EDS also required fewer searches to find the information they needed and viewed fewer web pages before choosing resources than any of the other four test groups
Our qualitative observations suggest that Summon might, in fact, be leading students inadvertently to less appropriate resources. Once newspaper, magazine, and trade periodical results were removed from the scores for the EDS and Summon test groups, the EDS group still obtained higher average scores than the students using Summon on all four questions. EDS was the superior performing discovery system.

This last notation concerning content types might be a good segue to your focus on a “bento style” display. We noticed that the descriptions on bento style and the published source code weren’t entirely in use within the survey tool (as presented for example in the NCSU library site with its varied content powering the boxes).

It is important to convey that EBSCO built an EDS API “Bento Box” demo with source code that was released July 11. You can find access, documentation, and source code here:

- [http://edswiki.ebscohost.com/EBSCO_Discovery_Service_API_Sample_Applications](http://edswiki.ebscohost.com/EBSCO_Discovery_Service_API_Sample_Applications)
- [http://support.ebsco.com/eit/samples/BentoBox_demo/](http://support.ebsco.com/eit/samples/BentoBox_demo/)

The boxes are filled in via a configuration file where the library can define what goes into each box for a search; whether it’s a content type, top subject terms, top publications, etc.

We released this demo after you had begun developing the survey application, but it does exist and has existed for more than four months. We agree that Bento is a very good design when used with various content types, lists, etc. I guess an open question from the study is worth noting whether user preference would have fared differently if the bento style presentation was used. Please note that some of the very early adopters of this bento style display are now EDS customers.

Speed of result data to the browser is an important aspect of service usability. As mentioned for Summon, “like all the products, did exhibit some extremely slow responses (possibly due to network hiccups, or cold caches on the vendor’s side)” [Speed of response between products – page 18], occasional abnormally long search times even in EDS can happen for these reasons. Appendix B is some very precise data on causes of network latency. Another contributing factor is the authentication process if it is improperly implemented (i.e., calling it asynchronously in the background every 30 minutes or so rather than on every search). Other comments on authentication are below. Please note also that several enhancements to increase API performance have been released since this study began. EBSCO has employees on staff who have “performance engineering” in their titles, and we regularly do very in-depth performance tracing and analysis on a customer level if customers experience slow performance. We would like to offer this service to John Hopkins as well if required in the future.

**Content**

We admittedly need to better convey how our “Source Type” facet can be used effectively in this model. Source Type definitions within EDS are actually designed by a team of librarians at EBSCO and the
content makeup is tailored to a library’s custom EDS index (i.e., EDS’s search index can be fit to a library’s catalog, subscribed databases, preferred content, as well as their “library e-holdings”).

In mentioning OpenURL and products needing to work with the Johns Hopkins link resolver (SFX), we wanted to clarify EDS’ best practices and approach to link resolvers. By default, Primo works great with SFX, Summon is tied to SS360, but EDS is link resolver neutral (even though we do offer Link Source as a product) and through EBSCOadmin makes these template link configuration available such that non-API library administrators can effectively manage this critical full text access point without requiring API app code changes. In fact, almost all current EBSCOhost customers have these already set up and do not need to do anything new.

In reviewing the screen capture of the survey tool [The survey instrument - page 5] and the screen cast supplied, it did not relay enough information for us to comment on the equal disparity of metadata displayed to a user for evaluation of results. With EDS, we make a concerted effort to license and make available the world’s richest metadata sources, including the most comprehensive full-text searching and highest quality subject indexing. It is the subject headings from controlled vocabularies that user testing has shown to convey to users the relevance of an article and why it is shown on page 1 of a result list. As mentioned, “preferring results with ‘more information’” [What criteria do participants use to judge? - page 17], we feel strongly that users are hindered by leaving this metadata off of the results display. That relevancy ranking and data presentation is an EDS differentiator that was not fully included in the test.

Our relevancy ranking algorithm [http://www.ebscohost.com/discovery/technology/relevancy-ranking] weighs heavily on matches to subject headings from controlled vocabularies and leans toward academic materials, recent publication dates, etc. that have proved through many usability tests to be what users want in an academic setting.

Guaranteed full-text links exposed directly on the result lists (from the 100 Johns Hopkins EBSCOhost databases via SmartLinking [http://www.ebscohost.com/discovery/user-experience/full-text-experience]) would also have a dramatic effect on user perception as to the usefulness and direct access to full text of article results. When all full-text links are hidden behind a link resolver link and the user is always forced into a separate menu for “potential full text” instead of listing first “known trusted pre-qualified links”, this does affect the user experience.

One conclusion that could be drawn from the results of the test, where scores were on equal footing and there was no clear “winner”, is that that these areas of metadata quality and presentation, relevancy ranking, supporting the “why” are important factors that shouldn’t be ignored and that these results should be combined with quality assessments such as the Illinois Wesleyan/Bucknell study. We do understand the study’s ambitions to create a level playing field, but feel we needed to highlight the above aspects that more than 2,000 libraries worldwide used to evaluate and choose EDS as their discovery service.
API Tech
As you built most of this project in Ruby, we did want to address our current direction for Ruby and the EDS API. A sample EDS API Ruby Gem is in process. Our initial focus is to progress this effort for integrating with Blacklight. An early version of this should be available by end of December. We have a developer looking at bento_search and it has been very helpful. We want to thank you for making this code available as open source.

Most of the issues highlighted with the EDS API’s technical capabilities and flexibility, we interpreted in these areas (with supporting figures in Appendix C):

- Administration and Search
- Authentication
- New EDS API v2.1 Release: Granular Data and Developers Console
- Documentation Improvements and Roadmap

Administration and Search
One of the key integration elements of the EDS API is its ability to help bridge a customer’s administrative and implementation resource groups. By providing the familiar EBSCOadmin interface to library staff, all of the API’s “profile” settings can be easily configured just as they would for any other EBSCO interface. This allows technical resources to focus on the implementation with the confidence that the associated settings have been put in place. API profile settings that can all be controlled in the EBSCOadmin interface are:

- Search Modes
- Limiters
- Local Collections
- Content
- Linking (Custom Links, Persistent Links, Smart Links)
- Facets
- Session Timeout (up to 8 hours)

The EDS API takes full advantage of the strong Search Engine and Relevancy Ranking algorithm that has consistently set EBSCO apart in the Discovery landscape. All searches conducted by the API utilize the exact same search technology as EDS. There is no separation of platforms thus ensuring consistent results that users expect (i.e., it’s exactly the same search syntax as EBSCOhost so students, advanced users, and librarians do not have to learn a new syntax).

In addition, the EDS API supplies not only the relevancy ranked search results, but also the individual relevancy scores for each result. This can be useful if the customer wishes to merge results with another data set. The information is available and consumable if needed.

Authentication
Your frustration with the EDS API’s authentication complexity is noted. EDS has some unique requirements from protecting its content that other discovery services do not. EDS is the only discovery
service that can directly include 1) full text and 2) licensed databases (which sometimes have a simultaneous user license) both of which require extraordinary access protection/authentication.

With this in mind, the EDS API utilizes an industry standard two step authentication approach via Authentication Tokens and Session Management. We are constantly evaluating the delicate balance of requirements and efficiency and appreciate all of the input collected to assist us in our efforts.

Some general guidelines that should be considered are:

- Calls to the API should include:
  - Authentication Token: the authentication response contains the expiration time so a new Token can programmatically be maintained.
  - Client Session: session tokens stay valid even if the authentication token has expired.
  - Search: use the same session token without the need to get a new one.

- Token expiration times:
  - Auth token: currently 30 minutes with expiration time provided in response. We are reviewing this setting for future efficiencies.
  - Session token: this is configurable in EBSCOadmin up to 8 hours, but it stays alive as long as there is client search activity using this session token (the expiration only applies when the session is inactive).

We recognize that our current two step authentication approach, via Authentication Tokens and Session Management, may be overly conservative given implementation realities. We are currently evaluating a modified strategy that will remove all of the hurdles you encountered by moving to an IP based authentication, coupled with the removal of SSL and token expirations. These efforts are currently being looked at for an early 2013 release. We will also be increasing existing authentication token expiration limits significantly for more immediate short term ease of use.

**New EDS API 2.1 Release**

EBSCO has maintained its commitment to continuous improvement to the EDS API as the Discovery landscape evolves. Ensuring that the EDS API provides all of the necessary elements to be “consumable” is a top priority and is directly reflected in the latest release. Some items worth noting for this release included:

- Additional “granular” data available in Search and Retrieve responses.
- Developer’s console updates
- Publication type icons available
- Response structure changes
- Highlighted search terms in Retrieve response
- Facet updates
- Demo and sample application source code updates
- Documentation updates
**Granular Data**

In addition to the display ready data contained in the ‘Items’ element, the 2.1 release has up to 14 different granular data items available under the ‘RecordInfo’ element. These include:

- Record identifiers: ISSN, ISBN, DOI
- Source information: Publication Date (year, month, date), Cover date, Volume, Issue, Start Page, Page Count, Language (with code)
- Title information: Record title, Parent title (example: Journal name)
- Authors
- Subjects

A data element may be found in both display ready format as well as granular format. Which one to use depends on your needs. Sample uses of granular data:

- Use the granular identifiers to create an external link (example: to OCLC) for a record.
- Use specific elements from the granular level source information, instead of the display ready source ‘line’ in ‘Items’.
- Display parent title underneath the record title.
- Display non-hyperlinked authors and subjects, without having to strip out any display tags.

While all of this granularity is now available, only the desired elements need be applied by the developer. This allows the flexibility of a very detailed response and record display or a streamlined approach – all of the information is available and you can choose what to implement.

**Developer’s Console Updates**

We have made significant improvements to the provided developer’s console interface to assist developers in their efforts for both ease of integration and implementation efficiencies. These updates include:

- Simplified user interface
- Each method linked to relevant documentation
- One Login button: can also use individual Auth and Session methods
- GET and POST for all methods
- Ability to see request sent to the server
- Embedded and Collapsible responses

**Documentation Improvements and Roadmap**

Any analysis of the EDS API is viewed as extremely valuable to EBSCO. Your efforts were very much appreciated and have helped us: identify weaknesses in our documentation that we will address shortly and verify our current direction for the latest release. In addition, this gives support to our continued 2013 EDS API roadmap.
## APPENDIX A:

<table>
<thead>
<tr>
<th>Feature</th>
<th>EIT:WS</th>
<th>EDS API 2.1</th>
<th>EDS API 2.2</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display-ready response format with labels</td>
<td></td>
<td>✔</td>
<td></td>
<td>EIT returns only “raw” granular data without display attributes such as labels.</td>
</tr>
<tr>
<td>Granular Data such as ISBNs, DOIs, Issue and Volume</td>
<td>✔</td>
<td>✔</td>
<td>+</td>
<td>EDS API data is structured (eg: an article is a child entity of a magazine) and more parse-able XML elements (eg: 'Title' instead of 'atl')</td>
</tr>
<tr>
<td>Complete Detail Record</td>
<td></td>
<td>✔</td>
<td></td>
<td>EDS API returns the same data and labels that display in EDS.</td>
</tr>
<tr>
<td>Relevance Score returned in search response</td>
<td></td>
<td>✔</td>
<td></td>
<td>Relevance score can be used to interleave results</td>
</tr>
<tr>
<td>JSON Response Format</td>
<td></td>
<td>✔</td>
<td></td>
<td>Lightweight response protocol optimized for mobile</td>
</tr>
<tr>
<td>EDS Facets (Content Provider, Publisher, Subject, Source Type) with hit counts</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full limiter and expander support</td>
<td></td>
<td>✔</td>
<td></td>
<td>All EDS limiters such as the Local Collection limiters, Library location, and Available in Library Collection limiter supported.</td>
</tr>
<tr>
<td>Search terms highlighted in response</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search modes support</td>
<td></td>
<td>✔</td>
<td></td>
<td>Supports Find All, Find Any or Boolean search modes</td>
</tr>
<tr>
<td>Book jackets for eBooks and catalog records</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Status</td>
<td>Notes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Publication Type icons</td>
<td>❌</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guest Access Support</td>
<td>❌</td>
<td>The EDS API Guest Access support is consistent with EDS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action based syntax to simply client side</td>
<td>❌</td>
<td>Robust demo covering all API methods. Source module ready to plug in</td>
<td></td>
<td></td>
</tr>
<tr>
<td>coding in response to user action such as</td>
<td>✔️</td>
<td>to existing VuFind implementation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>facet or limiter clicks</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Robust console application</td>
<td>❌</td>
<td>Robust demo covering all API methods. Source module ready to plug in</td>
<td></td>
<td></td>
</tr>
<tr>
<td>makes it easy to evaluate the API and helps</td>
<td>✔️</td>
<td>to existing VuFind implementation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>developers debug</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VuFind demo and module</td>
<td>❌</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiple standalone demos and source code</td>
<td>✔️</td>
<td>EDS API demos are full featured UIs with facets, limiters, etc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>collaboration</td>
<td>✔️</td>
<td>FAQs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EBSCOhost Integrated Search</td>
<td>✔️</td>
<td>EDS API implementation optimized for performance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Library Admin control over display – labels,</td>
<td>❌</td>
<td>Enables librarians to change the user experience without programming.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>link orders, facet order, defaults.</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Easily build Advanced Search page using the</td>
<td>❌</td>
<td>Info method returns available limiters and expanders, including</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Info response</td>
<td>✔️</td>
<td>their possible values (for example, library locations)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTAC</td>
<td>❌</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eBook Availability</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX B:

Sources of Latency and End to End Performance:

<table>
<thead>
<tr>
<th>Request</th>
<th>End Latency</th>
<th>Host Latency</th>
<th>Network Latency</th>
<th>SSL Latency</th>
<th>Idle Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>40885</td>
<td>2.5 ms</td>
<td>1.4 ms</td>
<td>59.4 ms</td>
<td>23.5 ms</td>
<td>2.7 ms</td>
</tr>
</tbody>
</table>

Report for Sources of Latency on eda-epichooheost.com (search)

Nov 07 2012 00:00 GMT - Nov 14 2012 23:59 GMT
APPENDIX C:

EBSCOadmin session duration:

Granular Data:

Mechanism for distribution of acetamide: a novel pharmacological agent for the treatment of functional dyspepsia in rat stomach (main)

By Yoshi, K-city

[Abstract] The novel pharmacological agent acetamide improves gastric motility by inhibiting acetylcholinesterase activity in stomach. However, the mechanism of distribution of acetamide from blood to stomach has not been clarified. Here, the tissue distribution of acetamide was investigated in rats. The tissue-to-plasma concentration rate (K0,app) in stomach decreased from 1.2 to 2.4 mg/g of tissue at steady state with increasing plasma concentrations, whereas the K0,app in vivo for skeletal muscle was much lower and constant, regardless of the concentration of acetamide in plasma. In vitro binding to stomach tissue protein exhibited a linear profile with a predicted K0,app in vitro of 2.2 times from free fractions under linear conditions. Therefore, protein binding to stomach tissue might play a limited role in the stomach distribution of acetamide. The influx permeability (P = 50 ± 31 nM/sec) in the stomach exhibited dose-dependent saturation at the lowest range of examined blood concentration of acetamide, whereas that in skeletal muscle exhibited only minimal dose dependence. In addition, the urinary concentration ratio of stomach to plasma (2.8) at steady state was markedly higher than that. Taken together, these results suggest that carrier-mediated concentration uptake processes play an important role in the distribution of acetamide to the stomach but not to skeletal muscle. © 2011 Wiley-Liss, Inc. and the American Pharmacists Association.
EBSCOadmin link configuration and templates:

EDS API Developer’s Console Redesign:
The Search method is used to perform searches against the databases in the profile. It returns a result list along with the relevant facets and abstracts.

**Parameters:**
- **Query:** `wind energy`
- **Search Mode:** `all`
- **Results Per Page:** `20`
- **Page Number:** `1`
- **Sort:** `relevance`

**Default Values:**
- Required.
- The search terms.
- `(booleanOperator), (facetCode): (term)`, Example: `SU-Hiking`
- `all`
- The number of records retrieved with the search results (between 1-100).
- `1`
- Starting page number for the result set returned from a search (if results per page = 10, and page number = 3, this implies: I am expecting 10 records starting at page 3).
- `relevance`

**Notes:**
- Use the Info method for a list of common sort options, and
- `Content-Type: application/xml`
- `Accept: application/xml`
- `x-authenticationToken: 002af78d-e921-4f87-ad78-38a2b339df56`
- `x-sessionToken: b5fe352-c44b-467a-bb0c-790f2fa20c93.ePRbqyozgHQ7URxG9eFBA==`
- `Host: localhost`

**HTTP/1.1 200 OK**
- `Date: Thu, 15 Nov 2012 15:24:23 GMT`
- `Server: Microsoft-IIS/7.5`
- `X-AspNet-Version: 4.0.30319`
- `X-Powered-By: ASP.NET`

Open in new window

```xml
  <SearchRequest>
    <QueryString>query-1-AND.wind:energy</queryString>
    <includeFacets>false</includeFacets>
    <searchMode>all</searchMode>
    <view>brief</view>
    <resultsPerPage>20</resultsPerPage>
    <pageNumber>1</pageNumber>
    <highlight>true</highlight>
  </SearchRequest>
</SearchResponseMessageGet>
```