

Improving Authentication and Authorization

SeamlessAccess and GetFTR

Introduction

In chapter 2, we talked briefly about SAML-based methods for authentication and authorization. However, IP recognition, despite all its drawbacks, is still the more common way of providing access. Why is this so?

Why RA21 or SeamlessAccess?

Here is a recap of how SAML works. Refer to the section Single Sign-on with SAML in chapter 2 for a fuller description.

Imagine a user is on JSTOR and tries to read a specific article. To activate the SAML-based method, also known as single sign-on (SSO), the user clicks on the sign-on button or link and identifies the institution they are from. This is typically done by selecting from a list of institutions. This sign-in process is known as the Where-Are-You-From (WAYF) process.

JSTOR, which acts as the service provider (SP), then redirects the user to the identity provider (IdP) of the institution, which we will call Institution X, for authentication.

How does JSTOR know where the IdP of Institution X is? Either Institution X has worked directly with JSTOR to provide the address of the IdP, or both the SP and Institution X are in the same identity federation (see chapter 2 for a discussion on what an identity federation is) and JSTOR looks up the address in the SAML directory.

Either way, the user authenticates using their institutional credentials at the IdP of Institution X. Assuming this goes well, the user is redirected back to the SP with a SAML assertion. The assertion may contribute *attributes* that contain information about the user. (We'll discuss this more later.) The SP then gives access to the user (see figure 4.1).

What types of SAML assertions and attributes can be sent by the IdP? In one scenario, the IdP can simply assert the user is affiliated with Institution X and entitled to the rights from such an affiliation.¹ In such a scenario, the anonymous identifier is unique for every visit and the SP. This is the highest level of privacy provided to the user in this process, as the SP does not get any information at all about the user beyond their institutional status.

Sometimes users from the same institution may have different entitlements depending on the department they are from. Additional assertions could assert that the user is from Institution X and also is an associate professor from Department Y. In this situation, the SAML assertion from the IdP to the SP would include some information about the user, possibly identifying the user's department or user group in an attribute. There's a possibility, of course, that liberal release of such attributes might allow users to be identifiable in practice. In the most extreme case, personally identifiable information (PII)—such as campus e-mail, name, and position—can be asserted and the information sent as attributes to the SP, which of course would allow individual users to be tracked. Finally, the IdP may make assertions that involve release of pseudonymous identifiers. They are similar to anonymous identifiers, except that while anonymous identifiers are generated for each visit, pseudonymous identifiers are generated for each person/SP combination; they persist and are reused across visits. In other words, if pseudonymous identifiers are used in the SAML process, the SP will always know it is the same user who is accessing the service but will not know the real identity of the user based on the identifier alone. This approach can be useful in cases where the user wants personalization (e.g., saving the custom settings of their account) but not to be personally identifiable.

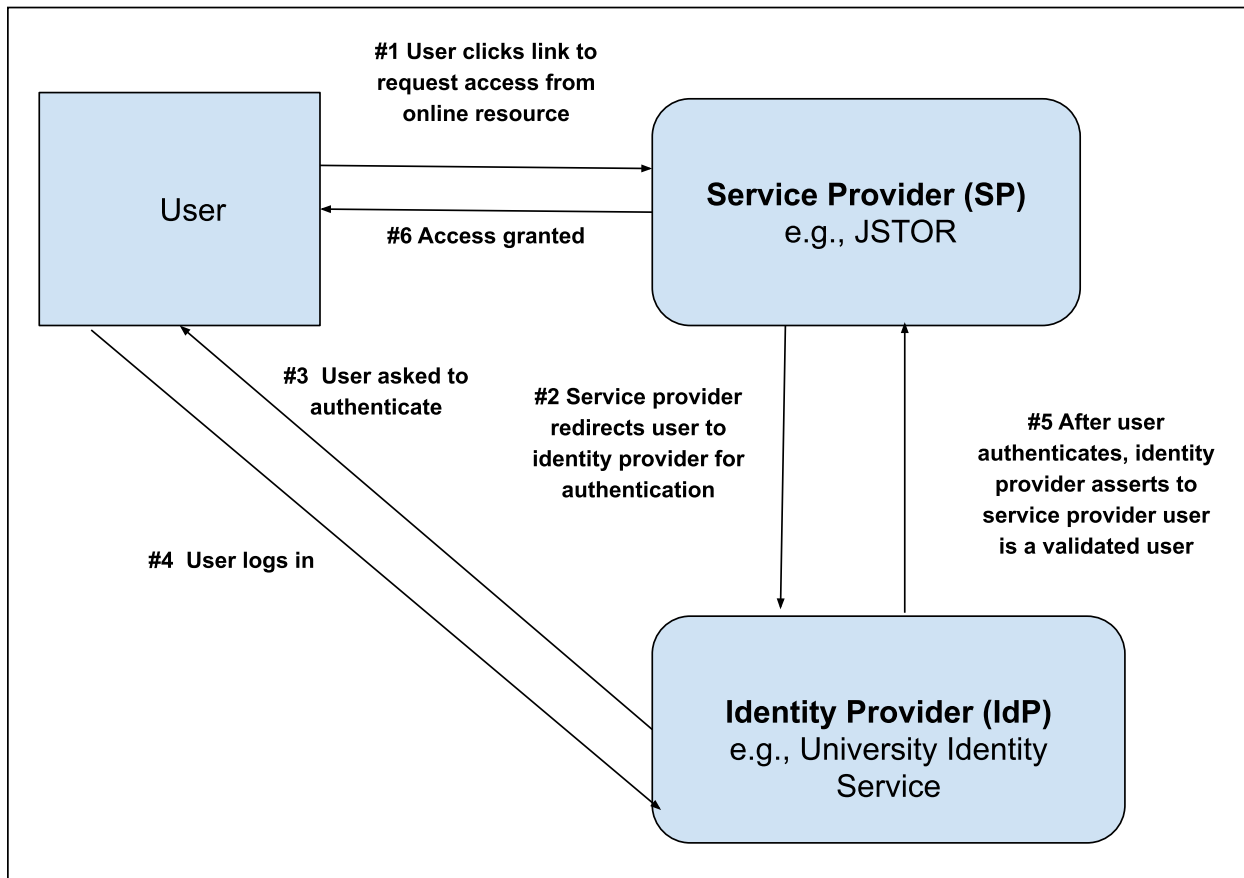


Figure 4.1
Diagram of the SAML SSO process

All this is fine as far as it goes. However, use of SAML-based methods is still not widespread.² Why? Besides the fact that not all library-licensed resources support SAML, there are a couple of reasons. First, while I have painted an ideal picture of how SAML-based federated access is supposed to work, in practice users find it unintuitive to use due to inconsistent and poor user interface (UI) elements during the IdP discovery phase. Second, the library may prefer IP authentication for a variety of reasons, including

- librarian lack of familiarity with federated access methods
- librarian concern about user privacy issues, perhaps driven by the lack of standardization in attribute release standards

RA21: Resource Access for the 21st Century was formed to explore solutions to some of these issues.

The IDP Discovery or the WAYF Problem

Launched in 2016, RA21 was a joint initiative of the International Association of Scientific, Technical, and

Medical Publishers (STM) and NISO with this mission: “to align and simplify pathways to subscribed content across participating scientific platforms. RA21 will address the common problems users face when interacting with multiple and varied information protocols.”³

Though this mission sounds somewhat vague, RA21 aimed to “explore pathways to move beyond IP-recognition as the primary authentication system.” In practice, this means RA21 focused quickly on improving user experience for researchers using SAML methods. It completed its initiative on June 30, 2019, with the publication of the NISO *Recommended Practices for Improved Access to Institutionally-Provided Information Resources*.⁴ Chief among the recommendations was setting up a new service, SeamlessAccess, to carry out the recommendations.

For the rest of the discussion, we will use just SeamlessAccess as a catchall term to describe the work of RA21 and the successor organization SeamlessAccess. SeamlessAccess identified several areas of improvement for SAML-based sign-ons but aims to improve and standardize sign-in UI across all platforms. It particularly focused on improving the WAYF,

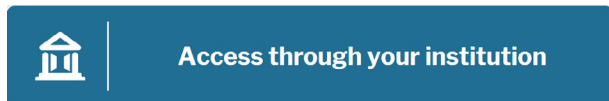


Figure 4.2
SeamlessAccess standardized log-in button (sample)

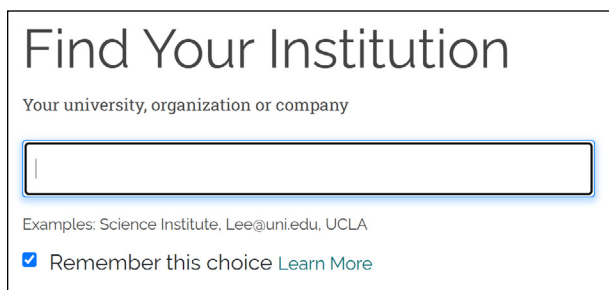


Figure 4.3
Search-based list box where users can identify their institution (sample)

or Where-Are-You-From, process as this was clearly a pain point. This is because prior to RA21 different journal publishers implemented this process in a non-standard, unintuitive, and often confusing way. For example, users would often be faced with multiple WAYF options resulting in two or more ways to search for their institutions.

By carefully analyzing the difficulty users had with this process and coming up with guidelines for a simpler, more intuitive, and consistent UI across all platforms, SeamlessAccess hoped to make this problem far less significant. See figure 4.2 for an example of the standardized log-in button that will be implemented consistently across all platforms that support SeamlessAccess.

As noted by researchers, SAML-based implementations prior to implementing SeamlessAccess could be extremely unintuitive. For example, a Georgetown University nursing student compared the experience she had using SAML log-in on Wiley before and after Wiley switched to supporting SeamlessAccess as “night and day.”⁵ Some of the problems identified in the old sign-in included an excessive number of clicks needed, the need to scroll down a long list of institutions, and unnecessary use of jargon like “selecting a federation.”

In comparison, the new SeamlessAccess UI, based on careful user testing, is a lot more intuitive with a consistent visual cue and call to action (“Access through your institution”). It normalizes the language used and provides a search-based list so users can easily find their institution (see figure 4.3).

There are also guidelines for responsive design to support mobile use. Because selecting the institution was often the slowest part of the process, SeamlessAccess also built in a browser-based mechanism that would allow the user’s browser to remember the last



 Add or change institution

Figure 4.4
Institution preselected by browser

used sign-in and auto-populate that option by default. This information is similar to a cookie but is instead stored in browser local storage. The browser remembers which institution the user previously chose to sign in and selects that institution across sessions, and SPs that supported SeamlessAccess did the same, which saved the user a lot of time. (See figure 4.4 for what a user might see if information is stored in the browser about the last selected institution.)

With more and more platforms supporting SeamlessAccess, consistency will help users know what to expect—the same way users today know what to do when faced with a Log in with Google or Facebook button. Still, for SeamlessAccess to work, platforms need to implement it.

Currently SeamlessAccess provides three types of integration for platforms that implement it—Limited, Standard, and Advanced methods. Please refer to the description of the service on the SeamlessAccess website for more details.

SeamlessAccess: The Service
<https://seamlessaccess.org/services/>

Publishers, such as IOPscience, that implemented SeamlessAccess have found large improvements. For example, IOPscience found that in 2021, total item requests via federated authentication increased by 82 percent after it implemented SeamlessAccess.⁶

Attribute Release and Privacy Concerns about SeamlessAccess

One of the major concerns librarians have about moving from IP recognition to SAML and the SeamlessAccess service is privacy and the attributes sent to SPs. The important thing to note is that the SP can receive only the attributes that the IdP releases. The question then becomes, Who decides what attributes should be sent? While the SP and IdP can mutually agree on the attributes to be sent (including, as noted earlier, none at all, in which case an anonymous identifier assertion is sent), it seems that it would be easier if there were some standards to follow.

This is where entity categories come into play, and a federation can define and use these categories to

set standards for SPs and IdPs in that federation. An example of such an entity category would be REFEDS's (Research and Education Federations) Research and Scholarship (R&S) entity category, which would apply a set of standard attribute release criteria for all SPs classed under this category, but this does not apply to library-licensed resources.

This has led to the development, under the auspices of SeamlessAccess, of the new Anonymous Authorization and Pseudonymous Authorization entity categories.⁷ Despite this development, many are still skeptical about supporting SAML-based methods and whether they should replace IP recognition and proxy servers.

In chapter 2, we saw IP recognition and proxy servers were extremely inconvenient for users, particularly off campus; however, they have one major advantage in that user's privacy is protected. Access via the proxy ensured that SPs would get practically no information on who was accessing the content as everything was filtered via the proxy.

While it is true that SAML access can be configured such that the usage is mostly anonymous and the SP is told only that the user is a legitimate user of the institution (through release of an anonymous identifier generated per visit), it is possible for the user to be tracked with other exchanges of attributes as discussed earlier.

All in all, protecting user privacy is an extremely technically complicated and tricky topic with much potential for missteps, and concerned librarians worry about making a mistake. Below are just some scenarios that may lead to privacy issues:

- The IdP server is often not controlled directly by the library itself but by the institution, typically the campus IT department.⁸ This might lead to misconfigurations that result in identifying information being leaked. Such misconfigurations are hard for the library to fix.⁹
- The people running the campus-wide IdP service may not share the same ethical and moral standard as libraries and may not see protecting user privacy as a concern.
- Even persistent pseudonymous identifiers might allow users to be tracked, as some have argued vendors can use a combination of web bugs and behavioral tracking to tie the persistent pseudonymous identifier to a real identity.¹⁰

Current Status of SeamlessAccess

At the time of writing in March 2022, SeamlessAccess is supported by about twenty service providers, including Elsevier's ScienceDirect, Wiley, Taylor & Francis, and the American Chemical Society (ACS), among others.¹¹ However, work needs to be done, not just from the publisher side, but also from the

institutional side. For institutions to benefit from SeamlessAccess, their users need to be encouraged to try to access content from content owners that support it. On the back end, the library or institution itself also needs to be registered in the appropriate identity federation with its IdP, and the publisher needs to be in the same federation. It is not unusual for libraries to have no say in the management of the IdP, as this is usually under the control of the institution's campus IT department, so setting this up may not be simple from the library point of view.

Leaving aside technical capabilities, there is still concern about privacy. While the development of the Anonymous Authorization and Pseudonymous Authorization entity categories creates some clarification on the attribute release bundles, some librarians are still skeptical of federated access compared to IP authentication for the privacy reasons discussed earlier.

It is important to note that RA21 and, presumably, its successor SeamlessAccess have stated that they do indeed have a long-term goal of eliminating accessing resources via IP.¹² So, in the long term, this may not be a case of having an additional option that you can choose to ignore. To be fair, supporters of RA21/SeamlessAccess point out that IP-based methods that generally also require the use of proxies have other drawbacks even if we are willing to accept the friction they involve when the user isn't on campus. Use of proxies on today's internet increases complexity and may even permit certain security risks.

We now turn our attention to GetFTR, a publisher-initiated project targeted at improving delivery from another angle.

What Is GetFTR?

On December 3, 2019, five major publishers—ACS, Elsevier, Springer Nature, Taylor & Francis, and Wiley—announced the launch of the GetFTR project.¹³ While RA21/SeamlessAccess focused on authentication, GetFTR focuses on streamlining access to journal content by means of real-time entitlement checks for users who discover such content via platforms other than the publisher website. This includes use of discovery tools (e.g., Scopus) and scholarly collaborative networks (e.g., Mendeley).

From the user's point of view, imagine being on a platform like Scopus and, after doing a search, seeing ten article results displayed and not knowing which article you have access to. The platform needs to reliably tell users which of these articles they have access to. Of course, part of this process involves the platform, known in GetFTR jargon as the *integrator service*, needing to confirm and authenticate the user. GetFTR does not do anything novel here. Instead, it uses the existing authentication mechanics—either IP

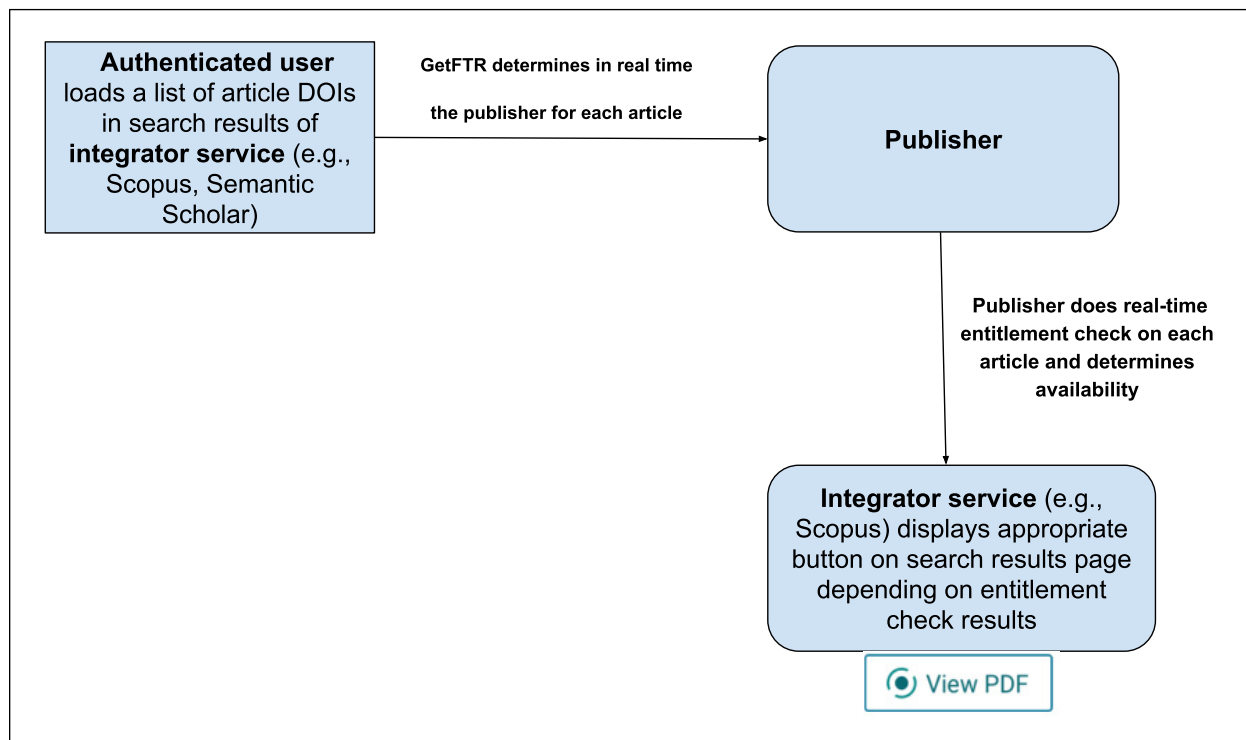


Figure 4.5
Workflow diagram for GetFTR

recognition or SAML-based methods, which include SeamlessAccess, as described earlier. What is novel is that once the platform authenticates the user, platforms that support GetFTR do real-time entitlement checks with GetFTR-supported publishers to get accurate and up-to-date information on whether the user can access the content and to display the information accordingly.

Note: Pre-authentication or lack of authentication will likely result in GetFTR displaying open-access or at least free-to-read full-text links.

GetFTR Explained Briefly

On paper, GetFTR works simply. See figure 4.5 for a workflow diagram.

GetFTR needs two pieces of information from the integrator service: the article DOI and the user's institution affiliation. The latter is obtained in a variety of ways, including

- SAML (or even SeamlessAccess) log-ins
- IP recognition

Next, for every article displayed with a DOI in the user's search, the GetFTR API will simply use the DOI to check which publisher it belongs to (via the Crossref API) and then route the query to the article publisher (publishers who are GetFTR partners) to see if the

user is entitled to it and display the availability next to each article. The process where it queries publishers to see if the user is entitled to a journal article is the real-time entitlement check. The publisher returns a corresponding *entitlement resource*, which contains the following pieces of information¹⁴:

- level of entitlement (e.g., yes, no)
- access type (e.g., open, free, paid)
- document type (e.g., version of record or alternative version)
- content type (e.g., HTML, PDF)
- a link to the actual resource

This information is then used by the integrator service to display the appropriate message on the search interface. Similar to SeamlessAccess, GetFTR has recommended callouts and labels to use consistently across all platforms. See figure 4.6 for the standardized GetFTR button that will be shown when the user is entitled to access the full text.

After a user has authenticated, the system is able to know by checking with the publisher whether the user is supposed to have access or not without the user even clicking through the link.¹⁵ GetFTR seems to work similarly to traditional library link resolvers by facilitating user access to links that they can use to access full text regardless of where the resource is. In other words, both provide a solution to the so-called

appropriate copy problem, so let's discuss that next. For a summary of how the following solutions to the appropriate copy problem stack up, see table 4.1.



Figure 4.6
Standard GetFTR button displayed when a user is entitled to access full text

GetFTR and Solutions to the Appropriate Copy Problem

As we discussed in chapter 2, one of the more interesting issues about digital copies is that there might be more than one online site where a user can get access to a journal article. While a journal article may be available at the publisher site—say, Wiley—it may also be available to some users at aggregator sites such as EBSCOhost. These days there might also be open-access copies (of varying versions) available at preprint servers and institutional or subject repository sites. A user from Institution A might have access via one site, while a user from Institution B might have access via another site due to different licensing agreements. How does the system know where to send the user? This in a nutshell is what librarians called the appropriate copy problem in the early 2000s.

The creation of the library link resolvers and the OpenURL standard was intended to solve this issue. Platforms that supported this standard and armed with the right institutional context would direct users to the appropriate institutional link resolver, and users would eventually be fed the right link to the appropriate copy.

Traditional Solution to the Appropriate Copy Problem: Link Resolvers and OpenURL

In chapter 2, we discussed how the traditional solution to the appropriate copy problem was via library link resolvers, which typically use OpenURL technology. However, both traditional link-resolvers using OpenURL and GetFTR require some way to check to see what the user (who is already authenticated) is entitled to access and from which sites. In fact, this is the key difference between the two. In the case of traditional link resolvers, the information on entitlements is drawn from the library—more specifically from the library's knowledge base. In comparison, GetFTR queries the publisher directly.

Another, smaller difference is the way the links are implemented. In most cases, link resolver links today are displayed as static buttons on web pages that support the link resolver. The user will see the same button or link for each article and will need to click on the link before they know if they have access. GetFTR links are designed to be dynamic. When the page loads, the system will do a check on the fly and display different links or buttons depending on the

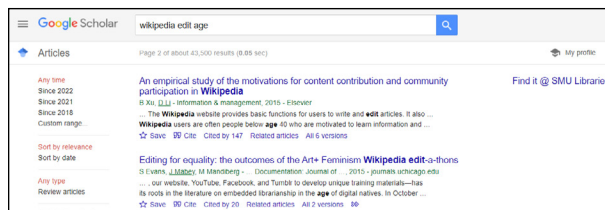


Figure 4.7
Google Scholar results where only one of two results has a link displayed

outcome of the entitlement check. This allows users to see what they have access to without first clicking on the link, which helps reduce frustration.

The closest traditional link resolvers come to supporting this feature is via the implementation of the Google Scholar Library Links program. When a user clicks on a displayed link in Google Scholar, they experience the same thing as they do on other platforms supporting library link resolvers. However, instead of displaying the same link next to every result, Google Scholar displays a link only for results where the user has access. See figure 4.7 for an example where one result has a link displayed (Find it @ SMU Libraries) and the other does not because Google Scholar knows the user is not entitled to access to the second resource via their institution. It can do so because Google Scholar obtains a local copy (which is updated periodically) of the institution's holdings or entitlements in advance and uses that to determine when to show a link.

Why GetFTR's Solution to the Appropriate Copy Problem Might Be Better

Why do we need GetFTR? How well do traditional library link resolvers work? After close to two decades' use of library link resolvers (which are largely but not entirely based on OpenURL technology), the main finding is that while OpenURL linking mostly works, it is not always reliable.¹⁶ There are various reasons for this, but here I will highlight two relevant ones.

The first problem is specific to the way knowledge bases and the link resolvers that use them are set up. Knowledge bases record entitlement at the journal or source level and not at the article level. In other words, the knowledge bases, if written in plain English, would say something like "We are entitled to access articles from Journal X, from volume 1 to volume 20, via JSTOR and from volume 25 onward via Business Source Complete." They are unable to reflect variance of access within the same issue or volume.

You might wonder why this could be an issue. After all, libraries generally subscribe to journal articles by year (or volume and issue) and not by article. However, with the rise of open access and in

Table 4.1. Different solutions to the appropriate copy problem

Solution	Source of entitlements/holdings check	Final link to full text	Type of links displayed on website
Library link resolver	Library knowledge base	Provided by library link resolver	Typically displays the same static button or link. Checks for availability only on user click.
Google Scholar Library Links program (library link resolver)	Google Scholar's local copy of institution holdings	Provided by library link resolver	Google Scholar uses local copy of institution holdings to selectively display links.
GetFTR	Publisher	Provided by GetFTR publisher	Dynamic link. Checks for availability on load of page and displays appropriate links.
LibKey.io	LibKey.io, with holdings obtained periodically from library	LibKey link resolver	Typically displays the same static button or link. Checks for availability only on user click.

particular hybrid open access, we are starting to see authors pay for some articles to be made open access so that only those articles are free to read while others in the same issue are not. Clearly, because library link resolvers work mostly with the entitlement data in the knowledge base (at the journal or source level), they are unable to handle the case where some articles are open access and hence available to use while others in the same issue are behind paywalls.

Beyond this problem, the biggest stumbling block when using link resolvers is that it is simply very labor-intensive for libraries to keep their entitlements in the knowledge base up-to-date. When a library signs a contract with a publisher that gives the institution access to a journal package, librarians need to update the library's knowledge base with information about these new journals. This allows the link resolver to correctly recognize that users have access to these journals.

Unfortunately, ensuring that the library's knowledge base is accurate can be difficult since it involves the librarian keep tracking of hundreds of packages consisting of thousands of journals with different coverage entitlements across dozens of publishers that all change across time. This is not made easier by the fact that libraries may have their own unique package of titles, all of which requires the librarian to ensure that this information is correctly reflected in their library's knowledge base. While there have been advancements that speed up the process of updating the library's knowledge base (in particular with NISO's *KBART Automation: Automated Retrieval of Customer Electronic Holdings*, which libraries can use to automate population of supported knowledge bases instantly using the publisher's API), this still relies on the librarian updating the system.¹⁷

So what does GetFTR bring to the table? Instead of querying the library knowledge base, GetFTR cuts out the intermediary by simply querying the publishers for entitlements directly. Therefore, even if the library is delayed in updating its knowledge base, the moment the ink on the contract is dry, the access rights are updated on the publisher side and the user gets full access to the new entitlements without requiring the library to do any work.

As mentioned earlier, GetFTR also has another advantage over traditional link resolvers in that access is checked on the article level. This enables GetFTR to flag a hybrid article as available even if not all the articles in the same issue are accessible. That said, modern link resolvers don't use just OpenURL technology for linking but may also use other methods, such as integration of open-access-finding services like Unpaywall or CORE Discovery, that can help mitigate this issue. In addition, journal publishers do occasionally turn on free access for limited periods for various reasons such as promotion. Even if this free promotion were given only at the issue and volume level and hence could be captured in the knowledge base, it is unlikely that any library would make the effort to update its knowledge base for temporary access. This would not be a problem if a library uses GetFTR, as entitlements are managed automatically at the publisher level.

GetFTR's Solution vs. LibKey

In chapter 3, we discussed Third Iron's LibKey infrastructure and suite of services, and of all the solutions discussed, it is most like GetFTR. Like GetFTR, it works on the article level (unlike traditional link resolvers using OpenURL) and, more specifically, works on DOIs. Hence it can detect hybrid articles directly while link resolvers cannot.

Unlike GetFTR, LibKey supports some aggregators, such as ProQuest and EBSCOhost. Also unlike GetFTR, it integrates with link resolvers by passing on requests when it fails to find any hits. On the other hand, like other library solutions, it relies on the library's record of entitlements (which it gets a local copy of periodically), which can be incomplete for reasons already stated.

One thing to note is that while we have generally acted as if getting entitlements from the publisher is superior compared to getting entitlements from the library knowledge base, publisher-provided entitlements can be wrong too! As a result, it is important not to blindly trust that the publishers have made no errors in turning on entitlements and to ensure that

procedures are in place to check on the accuracy of GetFTR linking.

Concerns about GetFTR

It is fair to say librarians expressed quite a bit of concern about GetFTR when it was first announced.¹⁸ GetFTR moved quickly to address some of these concerns. For example, on first launch, it seemed tied to RA21/SeamlessAccess for authentication, which as noted earlier has its own set of issues, and this was addressed by announcing it would also allow support of IP authentication as well. On the community governance front, GetFTR was initially started by and involved only publishers. The lack of representatives from various other stakeholders, such as the researcher and librarian communities, was concerning. This issue has since been addressed as well, with noted librarian Lisa Hinchliffe among others nominated to serve on GetFTR's advisory board. However, other concerns remain.

First, questions were raised on whether GetFTR would also cover third-party aggregators and not just publishers. This is important to many institutions that have coverage of important journals via aggregators such as EBSCO or ProQuest rather than via publishers. While GetFTR issued an announcement almost immediately after its launch stating that "GetFTR is fully committed to supporting third-party aggregators,"¹⁹ at the time of this writing in March 2022, this issue still has not been resolved and no aggregator is yet included. Related to this issue is the concern that GetFTR planned to show No Access labels (and might provide publisher-mandated options) if the GetFTR entitlement checks showed no access. Clearly this would be misleading since it did not take into account aggregators or sources other than publisher sites. As Lisa Hinchliffe noted, such a red No Access label actually means "there is no entitled access that is known to exist for all users at this institution on the publisher platform" rather than no access per se.²⁰ There has been discussion that GetFTR might be moving away from the idea of showing No Access labels and only show links for available items, but as of this writing in March 2022, it is still unclear how this will turn out.

Another hot-button issue is how or whether the GetFTR feature would appear alongside existing delivery mechanisms like link resolvers. The answers to this question might eventually differ depending on the platform.

Ultimately, though, librarians feel uneasy that GetFTR is a way for publishers to create their own alternative to library link resolvers and suspicious that this is a way to control where users end up and take away the choice from users and libraries.

Chris Bullock, an electronic resource librarian at California State University, reviews the appropriate

copy problem and compares library link resolvers and GetFTR. Bullock is of the view that "it [is] clear that the intent of GetFTR is not to connect researchers with the most appropriate copy for their needs, but to improve linking through channels that participating publishers control."²¹

From the point of view of many librarians, GetFTR, which is an API, should arguably be just part of the tool kit of linking techniques from which the link resolvers can choose rather than a separate independent one. GetFTR's response so far has been non-committal. In the FAQ on whether GetFTR works with library link resolvers, the response is "No, not yet," though this use case is being explored.²²

Finally, in terms of privacy, GetFTR claims that beyond DOIs and affiliations, "it does not require or capture any other information about the user," which is comforting.²³ Still, compared to traditional link resolvers in which the publisher isn't much involved, a move toward GetFTR over traditional link resolvers will make publishers a bigger part of the workflow, which might eventually lead to privacy issues. There's also a tiny caveat to the statement that GetFTR doesn't capture any information about the user, as "integrators can also share user's [sic] IP addresses with GetFTR, although this is optional. Those that choose to share user's [sic] IP addresses with GetFTR and participating publishers have to notify users via their privacy policy ahead of doing so."²⁴

Conclusion

Both SeamlessAccess and GetFTR, simply by the virtue of being heavily supported by major journal publishers, have the potential to completely change the way our users get access to resources and journal full text if they choose to adopt these systems on their platforms. At this point, though, both projects are still fairly early in their implementation, and it is unclear how things will pan out. Both projects have of course included other stakeholders, such as librarians and researchers, in the discussion, and it is important for librarians to engage with the issues.

Notes

1. SeamlessAccess, "Privacy, Attributes, and Why They're Important," YouTube video, 8:15, June 8, 2020, <https://www.youtube.com/watch?v=4xRqdc0DeJl>.
2. IOPscience reported only 3 percent of all usage was via federated access. (OpenAthens, "How IOP Publishing Simplified User Access to IOPscience," YouTube video, 56:26, July 20, 2021, <https://www.youtube.com/watch?v=khMp9t0hkZ8>.)
3. RA21: Resource Access for the 21st Century, "RA21 Position Statement on Access Brokers," August 23,

- 2018, <https://ra21.org/what-is-ra21/ra21-position-statement-on-access-brokers/>.
4. NISO, *Recommended Practices for Improved Access to Institutionally-Provided Information Resources: Results from the Resource Access in the 21st Century (RA21) Project*, recommended practice, NISO RP-27-2019 (Baltimore, MD: National Information Standards Organization, June 21, 2019), <https://www.niso.org/publications/rp-27-2019-ra21>.
 5. SeamlessAccess, “Seamless Access and the User Journey,” YouTube video, 2:39, August 10, 2021, <https://www.youtube.com/watch?v=V5xfPyaIMyI>.
 6. OpenAthens, “How IOP Publishing Simplified.”
 7. “REFEDS Community Announces Publication of New Entity Categories,” REFEDS, April 6, 2021, <https://refeds.org/a/2558>.
 8. This has led some libraries to run their own identity providers via OpenAthens service.
 9. This is not a theoretical issue. Observers such as Lisa Hinchliffe have shown that misconfigurations of SAML have led to exposure of e-mails and personal names. For example, see Lisa Janicke Hinchliffe, “What Will You Do When They Come for Your Proxy Server?,” *The Scholarly Kitchen* (blog), January 16, 2018, <https://scholarlykitchen.sspnet.org/2018/01/16/what-will-you-do-when-they-come-for-your-proxy-server-ra21/>.
 10. Dorothea Salo, “Single Sign-on, the Library, and Patron Privacy,” presentation slides, November 1, 2021, <https://speakerdeck.com/dsalo/single-sign-on-the-library-and-patron-privacy>.
 11. “Stakeholders > Service Providers,” SeamlessAccess, <https://seamlessaccess.org/stakeholders-for-service-providers/>.
 12. Todd A. Carpenter, “Myth Busting: Five Commonly Held Misconceptions about RA21 (and One Rumor Confirmed),” *Scholarly Kitchen* (blog), February 7, 2018, <https://scholarlykitchen.sspnet.org/2018/02/07/myth-busting-five-commonly-held-misconceptions-ra21/>.
 13. Roger C. Schonfeld, “Publishers Announce a Major New Service to Plug Leakage,” *Scholarly Kitchen* (blog), December 3, 2019, <https://scholarlykitchen.sspnet.org/2019/12/03/publishers-announce-plug-leakage/>.
 14. “GetFTR: Dataflows and User Privacy,” GetFTR, November 19, 2021, <https://www.getfulltextresearch.com/community/getftr-dataflows-and-user-privacy/>.
 15. GetFTR also offers “Deferred Authentication (federated authentication provided by publisher),” where the user authenticates only after clicking on the first GetFTR link, as well as authentication before or after a search. (“Usability Guidelines,” GetFTR, March 2020, <https://www.getfulltextresearch.com/for-integrators/usability-guidelines/>).
 16. Cindi Trainor and Jason Price, *Rethinking Library Linking: Breathing New Life into OpenURL* (Chicago: American Library Association, 2010).
 17. NISO, *KBART Automation: Automated Retrieval of Customer Electronic Holdings*, recommended practice, NISO RP-26-2019 (Baltimore, MD: National Information Standards Organization, June 18, 2019), <http://www.niso.org/publications/rp-26-2019-kbartautomation>.
 18. Lisa Janicke Hinchliffe, “Why Are Librarians Concerned about GetFTR?” *Scholarly Kitchen* (blog), December 10, 2019, <https://scholarlykitchen.sspnet.org/2019/12/10/why-are-librarians-concerned-about-getftr/>.
 19. “Updating the Community,” GetFTR, accessed November 19, 2021, <https://www.getfulltextresearch.com/community/updates-the-community/>.
 20. Hinchliffe, “Why Are Librarians Concerned?”
 21. Chris Bullock, “Get Full Text Research and the Search for Appropriate Copies,” *Serials Review* 46, no. 2 (2020): 160–62, <https://doi.org/10.1080/00987913.2020.1759361>.
 22. “Why GetFTR and FAQs,” GetFTR, accessed November 19, 2021, <https://www.getfulltextresearch.com/why-use-getftr/>.
 23. “GetFTR: Dataflows and User Privacy.”
 24. “GetFTR: Dataflows and User Privacy.”