Best Practices for Libraries and Library Service Providers

These best practices were developed by the InCommon Library Consortium in 2009. The consortium was formed to explore various potential solutions. The focus was to improve access to licensed electronic resources, identify user scenarios, document business practice and technology issues, and test proposed solutions.

Resource Providers

Resource providers deal with a variety of authentication challenges and often have to support several authentication schemes simultaneously. Shibboleth is just one of those schemes. These best practices should serve as a guideline for resource providers that are implementing Shibboleth. These best practices have been written by InCommon member institutions that have experience with integrating Shibboleth as an authentication option for library resources.

If all of these best practices are followed, libraries will be able to maintain the seamless access they can currently provide with IP-based authentication, and also have the groundwork laid for future developments such as personalized services. As there are many resource providers and all have different platforms and services, there is no expectation that all will be able to implement these best practices immediately. Some items tend to naturally build on others and in those cases they have been listed sequentially below. Each issue is explored more deeply following the list.

1. Authorization via eduPerson attributes
2. Implement WAYF-less URLs.
3. Implement authenticated direct links to resources.
4. Shibboleth/EZproxy hybrid compliance

Best practice #1: Authorization via eduPerson attributes

In a typical Shibboleth Single Sign-On scenario, the Shibboleth identity provider (IdP) performs the authentication, and the service provider (SP) handles the authorization. The identity provider delivers (only) those user attributes necessary for the service provider to make an informed authorization decision. These user attributes (their names and values) are completely variable and can be implemented in any agreed-on format. However, eduPerson provides the standard definitions for how these attributes should be formulated.

One of the main values of InCommon – or any federation – is an agreement to use a common set of user attributes. This simplifies implementation for both identity providers and service providers within the federation. InCommon strongly encourages the use of eduPerson attributes. For the purpose of this best practices document, the InCommon Library Services Collaboration makes specific recommendations concerning eduPerson attributes, with each party having certain responsibilities.

Libraries' responsibilities to e-resource vendors (Shibboleth SPs):

- to restrict the assignment of relevant entitlements to all (but ONLY) those members of their communities who meet the terms of licensing contracts with the vendors.

Libraries' responsibilities to their own users:

- to send only those attributes necessary for the vendor to make a proper authorization decision.
- to provide transparency to users about **which** identity attributes are being released to any given SP (and, when plausible, to actually allow user **control** over the attribute release). This becomes increasingly important as SPs move to support additional **personalization** features, which (by definition) **require** the ability to identify an individual as an individual rather than simply as the member of a certain group or class.

E-resource vendors' (SPs) responsibilities to libraries:

- to respect the agreed-upon entitlements when presented during the authorization phase of attempted resource access. It is understood that this means that the agreed-upon attribute must be released by the library IdP and must contain an appropriate value.

In order for single sign-on to be efficiently implemented across all vendors and libraries in the InCommon Federation, it is critical that there is standardization to the degree and in the ways that allow for the simplest solution that remains effective. To this end, we strongly recommend the use of **eduPersonEntitlement** as the standard attribute for authorization by e-resource vendors within InCommon. Furthermore, we strongly recommend that the relevant ***VALUE*** for the "eduPersonEntitlement" be "**urn:mace:dir:entitlement:common-lib-terms** " as described at http://middleware.internet2.edu/urn-mace/urn-mace-dir-entitlement.html*

By agreeing to and providing both a standard attribute and value:

- Libraries can efficiently manage access across scores of potential resources.
- Vendors can efficiently manage access for scores of libraries.

This presupposes standard contract terms (hence the "common-lib-terms"), of course, but by striving for this, we **encourage** those standard terms **and** ensure the feasibility of managing Shibboleth for both vendors and libraries.

Note, also, that by recommending the use of eduPersonEntitlement, we are intentionally indicating that it should be preferred to the eduPersonScopedAffiliation attribute, since (1) the former expressly allows standardization on a **single** attribute and standard value, and (2) the latter provides the libraries (and their parent institutions) less granularity in discriminating valid users from invalid ones in the standard LDAP taxonomy.

While the eduPersonEntitlement should be the initial focus for making single sign-on feasible on a widespread basis, it does not provide the user-specificity necessary for enhanced services such a personalization. For such personalization, we recommend use of either the eduPersonPrincipalName OR eduPersonTargetedID attributes.

In large part, the decision point between these two choices is weighing the relative importance of privacy vs. interoperability. The former tends to provide more human-

friendly identification of an individual (e.g, jsmith@somecollege.edu **vs.** kl83HlsnblqYskgh72KfqkI) and therefore it will likely be preferred for use across multiple vendors and resources. Identifying an individual in a more human-readable way, however, might elicit greater privacy concerns. In either case, it is important to realize that the identifier will still **uniquely** identify a given individual.

Best practice #2: Implement WAYF-less URLs

Simon McLeish coined the term WAYF-less URL and there is plenty of discussion of the term and its meaning on his wiki (see https://gabriel.lse.ac.uk/twiki/bin/view/Projects/WayfLess). In brief, WAYF-less URLs are URLs to resources that allow for bypassing the single sign-on **W**here **A**re **Y**ou **F**rom (WAYF) step. For our purposes, this means providing a URL syntax such that a resource URL could be created to navigate users through the authentication/SSO process without prompting users to identify their institution.

There are two acceptable forms of WAYF-less URLs. The first is Session Initiators implemented by the resource provider. The second is an SSO location provided by the identity provider. Each of these is described below, along with some indication of what is required to implement each by the resource provider.

Session Initiators
Session Initiators are URLs that exist on the resource provider site and can accept two parameters – a resource location and an identity provider's entityID – and properly direct the user through the appropriate identity provider for authN and then on to the resource location, without requiring the user to identify where they are from. The syntax of a session initiator URL is as follows:
http://resource-provider-site/session-initiator-url?entityID=IDENTITY-PROVIDER-ENTITYID&target=RESOURCE-LOCATION

With session initiators, the library creating WAYF-less URLs would need only know the location of the session initiator URL, their own campus' identity provider entityID, and the URL of the resource they are trying to link to. JSTOR is an example of a resource provider that has implemented Session Initiators:
http://www.jstor.org/start-session?entityID=provider-uri&target=target-url

The Shibboleth service provider software implements Session Initiator URLs natively. Information about their configuration is available on the Shibboleth 2 wiki, https://spaces.internet2.edu/display/SHIB2/NativeSPSessionInitiator, and there is some more basic discussion on the Shibboleth 1.x wiki:
https://spaces.internet2.edu/display/SHIB/SPMainConfig

For a resource provider, this is only part of the implementation; this must integrate and play well with other supported forms of authentication.

Identity Provider SSO
Due to compatibility and extensibility concerns, we recommend that NEW implementations of THIS type of WAYF-less URL *NOT* be pursued. (Instead, wherever possible, implement the "Session Initiator" style described above.) Information is provided here for completeness.

A very good description of this type of WAYF-less URL is given by Simon McLeish on his wiki page, WAYF-less URL, under the section, General Form: (https://gabriel.lse.ac.uk/twiki/bin/view/Projects/WayfLess)

We won't recapitulate all of the detail here, but will copy the basic elements of this type of WAYF-less URL. The general form of a WAYF-less URL for a Shibboleth protected resources is:

```
SSO-LOCATION?target=RESOURCE-LOCATION&shire=AC-SERVICE-
LOCATION&providerId=SERVICE-PROVIDER-ENTITYID
```

where:

- SSO-LOCATION is the URL of the single sign-on service of the Identity Provider to use for authentication (which can be found in the IdP configuration files or federation metadata)
- RESOURCE-LOCATION is the URL which is to be accessed following the establishment of a single sign-on session
- AC-SERVICE-LOCATION is the URL of the Assertion Consumer Service of the Service Provider (which can be found in the federation metadata)
- SERVICE-PROVIDER-ENTITYID is the identifier of the Service Provider within the federation used for access (which can be found in the federation metadata)

In order for resource providers to support this type of WAYF-less URL, the resource provider site must be able to establish their own sessions after someone has been authenticated via the SAML-compliant single sign-on software.

Best practice #3: Implement authenticated direct links to resources.

This is mainly an extension to best practice #2, WAYF-less URLs. The difference is that a resource provider can support WAYF-less URLs even if those WAYF-less URLs only work with their “start page” or their search page. To support best practice #3, resource providers are asked to extend the WAYF-less URL to work directly with resource URLs, so that WAYF-less URLs can be created directly to a resource (ex. full text of an article).

Best practice #4: Shibboleth/EZproxy hybrid compliance.

EZproxy has the ability to rewrite URLs and redirect browsers appropriately. This ability allows EZproxy to be configured to generate WAYF-less URLs for a target resource, if it is only given the target resource URL. This means libraries can continue to send all of their links from other applications (link resolvers, metasearch engines, database of databases, learning management systems, etc) through EZproxy. And then EZproxy will turn those links into WAYF-less URLs.

Again, this best practice builds on the one before; in order to support it, a vendor must already support best practices #2 and #3.

In order for a resource provider to be Shibboleth/EZproxy hybrid compliant, their Shibboleth authenticated direct links must be either (1) the exact same URL as the

unauthenticated link to the resource, or (2) capable of being rewritten consistently from an unauthenticated resource link to a Shibboleth authenticated direct link.

Two examples might make some sense of this.

Example 1
JSTOR provides persistent URLs such that the same persistent URL that is used by other authentication means (eg. IP authentication) also can be used as the RESOURCE-LOCATION URL when creating WAYF-less URLs as described above. For instance, `http://www.jstor.org/stable/3912141`, is a persistent link to The Science Newsletter. This same URL can be used directly to create a WAYF-less URL:
`http://www.jstor.org/start-session?entityID=urn:mace:incommon:university.edu&target=http%3A%2F%2Fwww.jstor.org%2Fstable%2F3912141`

The fact that the same URL can be used to create the WAYF-less URL makes implementation via EZproxy very straightforward. EZproxy provides a mechanism, called SPUEdit, for rewriting URLs and redirecting browsers to the new rewritten URL. This SPUEdit functionality allows EZproxy to create a WAYF-less URL from a persistent link. In the case of JSTOR, an SPUEdit directive in EZproxy can be defined as follows:

```
SPUEdit @^https*://.*\.jstor\.org.$@http://www.jstor.org/start-session?entityID=urn:mace:incommon:university.edu&target=$e0@irs
```

The SPUEdit is a regular expression rule. In this example, `^https*://.*\.jstor\.org.$`, is the URL pattern that is matched in the regular expression. And `http://www.jstor.org/start-session?entityID=urn:mace:incommon:university.edu&target=$e0` is the URL that is created from the regular expression. [The `$e0` in this example is the entire matched pattern encoded.] When EZproxy processes a URL, for instance, `http://ezproxy.university.edu/login?url=http://www.jstor.org/stable/3912141`,
the URL will match the SPUEdit pattern, and redirect the user's browser through a WAYF-less URL:
http://www.jstor.org/start-session?entityID=urn:mace:incommon:university.edy&target=http%3A%2F%2Fwww.jstor.org%2Fstable%2F3912141

Example 2
HW Wilson Web provides persistent URLs. However, the persisten URLs that are used to create WAYF-less URLs for HW Wilson Web are not the same persistent URLs that are used by other authentication methods. For instance, a persistent URL for Visual Arts: The State of the Art is

```
http://vnweb.hwwilsonweb.com/hww/jumpstart.jhtml?recid=0bc05f7a67b1790e22c948a92294d51df84446d22ac499870f2fecb1b0400d7c102b0ca3d14413a7&fmt=C
```

A Shibboleth authenticated direct link to the same resource, though, is a different URL structure,

https://vnweb.hwwilsonweb.com/cgi-bin/shibenv.pl?recid=0bc05f7a67b1790e22c948a92294d51df84446d22ac499870f2fecb1b0400d7c102b0ca3d14413a7&fmt=C

Notice that the difference between the two URLs:

http://vnweb.hwwilsonweb.com/hww/jumpstart.jhtml <=>
https://vnweb.hwwilsonweb.com/cgi-bin/shibenv.pl

It turns out that the jumpstart.jhtml pattern can be consistently replaced by the shibenv.pl pattern across the HW Wilson Web platform in order to create WAYF-less URLs. The process is slightly more complicated, but can be handled by a SPUEdit directive in EZproxy:

```
SPUEdit
@^https*://vnweb\.hwwilsonweb\.com/hww/jumpstart\.jhtml?(.*)$@https://shib.university.edu/shibboleth-idp/SSO?shire=https%3A%2F%2Fvnweb.hwwilsonweb.com%2FShibboleth.sso%2FSAML%2FPOST&providerId=https%3A%2F%2Fvnweb.hwwilsonweb.com%2Fshibboleth&target=https%3A%2F%2Fvnweb.hwwilsonweb.com%2Fcgi-bin%2Fshibenv.pl%3F$e1@irs
```

Notice that, in this case, `$e1` is used instead of `$e0`. This is because `$e1` is the portion of the original URL that matches (.*) in the pattern. Basically, this SPUEdit directive replaces the jumpstart.jhtml portion of the URL with the shibenv.pl portion, and then constructs a WAYF-less URL.

The importance of this is that it allows libraries to have a single point within the library environment – EZproxy – where WAYF-less URLs are created. There are several different use cases within a university and beyond for creating Shibboleth-authenticated direct URLs. As can be imagined, these can be complicated to create by hand, and also complicated for software systems within the university setting (which may or may not even be aware of Shibboleth) to create. However, it is much simpler to create an EZproxy link by simply appending the target URL to the EZproxy login URL.