Report from InCommon Linking SSO Working Group

Created by Etan Weintraub (johnshopkins.edu), last modified by Nicole Roy just a moment ago

Final report, approved by CACTI on 2023-08-16

Version: 1.0

Publication Date: 2023-08-25

DOI: http://doi.org/10.26869/TI.171.1

Introduction

A number of factors have, in recent years, led to increasing deployment of multiple single sign-on (SSO) solutions within individual organizations. Different consumers of SSO services may require different SSO protocols/APIs (eg., SAML vs. OIDC vs. WS-* vs. CAS); implementations of the same protocol may differ in ways that require different SSO providers due to variant interpretations of standards; some (primarily commercial) services may even provide their own self-contained SSO solutions. Apart from the expense of operating multiple SSO systems, this fragmentation of SSO services produces an undesirable, high-friction user experience, and can threaten the consistency and security of identity and access management (IAM) across disparate systems. Required to interact with multiple, unlinked SSO services, users may become confused as to what credentials to use when, and which “sign on” service(s) they should trust. They may quite rightly question how their experience can be termed “single” sign on at all.

A common, and in many cases the only viable approach to reducing friction and limiting the negative impact of SSO service fragmentation on users involves linking disparate SSO systems together, usually with the goal of providing a consistent point of authentication for the end user while allowing SSO consumers (relying parties) to integrate with different linked component services as necessary.

Multiple strategies for linking particular SSO systems may be used, each with different effects on the user experience, security, and federation capabilities. The choice, for example, of which SSO system will be responsible for end-user interaction, and how the integration between linked systems is accomplished, may expand or limit options for such important features as multi-factor authentication (MFA). No single linking strategy may be “optimal” for all sites and all scenarios, but each strategy has strengths and weaknesses which need to be considered when an organization designs a solution.

This report details the creation of the InCommon Linking SSO Working Group and it's attempt to document some of the known strategies for linking SSO systems and significant issues and benefits around each. After compiling the rational for the linked IdP scenarios it was very clear that a core value proposition for linking SSO systems is the ability to federate an otherwise non-InCommon-compatible IdP with InCommon, a multilateral federation. Other common value propositions on linking SSO systems involved less user friction due to consistent login experiences, and even the ability to create temporary transition states or migration paths between different SSO products or solutions.

The Working Group

The InCommon Linking SSO Working Group stemmed from conversations that occurred at ACAMP 2021 and the fact that there are many different SSO protocols and APIs that individual organizations have found a need to have that aren't always available within one SSO solution, which can require having multiple SSO solutions within the same organization. Of course, this leads to a question of how to link the different SSO providers so that there is still true Single Sign On, and not Multiple Single Sign On solutions. The working group was chartered with the intent to review existing methods members have used, in the past, present, and future plans, and to give a basic listing of scenarios for linkage strategies, as well as the benefits and issues with each. The group first convened in April 2022 to review the charter, determine the scope of the work the group planned to complete, and establish ground rules for the group. After this meeting, Brian Arkills (U Washington) and Etan Weintraub (Johns Hopkins) were elected as co-chairs for the working group. It was decided to target a completion date of end of September for the group's work.

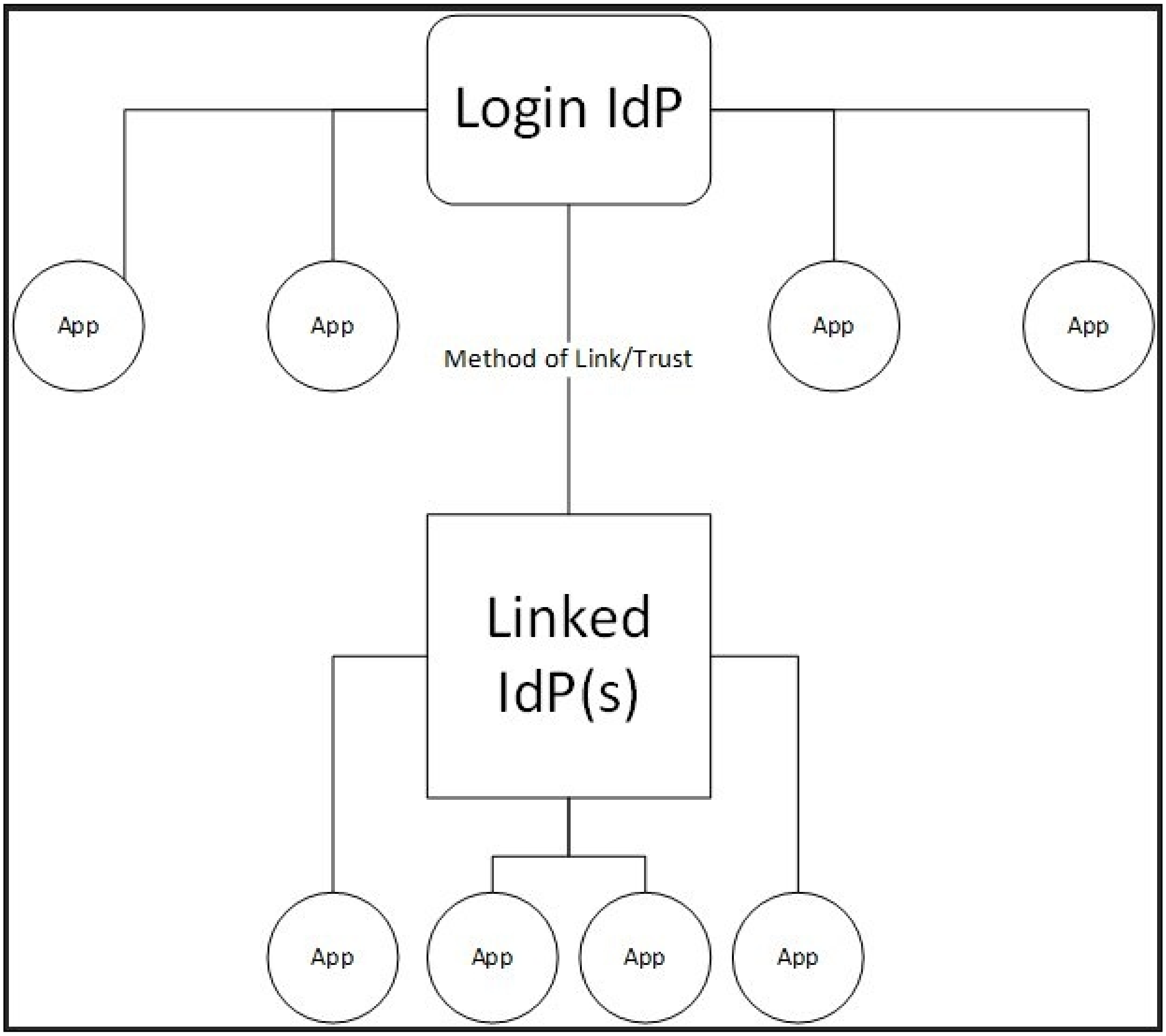
Over the next few months, the working group went through the process of gathering a list of scenarios members of the group were using and what issues and benefits they saw with each. A table was developed for the purpose of being a uniform way to provide information for each scenario, and members were asked to load the information for their scenarios into that table. The table is the expected final product of the group's work.

Visual Aid

The finished table has five columns defined as follows:

Login IdP	The IdP the users enter their credentials at
Linked IdP(s)	The additional IdP or IdPs that applications are connected to
Method of Link/Trust	The technology used to link the IdPs together
Significant Issues	A list of issues seen with using the scenario
Significant Benefits	A list of benefits seen with using the scenario

The first three columns may be visually described by filling them in to the following diagram:



IdP Glossary

There is also a “glossary” of IdP names that are used in the table that is shared here:

IdP	Full Name	Website	Supports Multilateral Federation Natively
ADFS	Microsoft Active Directory Federation Services	https://docs.microsoft.com/en-us/windows-server/identity/active-directory-federation-services	No (but there are tools available to make it work)
Azure AD	Microsoft Azure Active Directory	https://azure.microsoft.com/en-us/services/active-directory/	No
CAS	CAS Single Sign-On	https://www.apereo.org/projects/cas	Yes
CIB	Cirrus Identity Bridge	https://www.cirrusidentity.com/products/bridge	Yes
Google	Google Cloud	https://cloud.google.com/architecture/identity	No
OAM	Oracle Access Management	https://www.oracle.com/security/identity-management/access-management/	No

IdP	Full Name	Website	Supports Multilateral Federation Natively
Okta	Okta	https://www.okta.com	No
Shib	Shibboleth	https://www.shibboleth.net	Yes

The table is available at Key Linked IdP Scenarios and below. This is a list of Scenarios we have and had people willing to comment on. This is NOT an exhaustive list of possible scenarios, but rather the ones which the working group had collective experience. One requirement for all scenarios was that they supported Multilateral Federation. If any future readers of this report have experience with another linked IDP scenario (to achieve similar goals) not on the list below please consider reaching out to the Linking SSO Working Group mailing list on possibly getting the scenario added to this report.

The ability to not only parse, consume, and configure an IdP's behavior based on multilateral federation metadata, but also interpret and validate metadata signatures is critical to participating fully in a multilateral federation, and is integral to maintaining the trust on which such federations depend. In each of the selected scenarios below, it is worth noting that at least one of the linked IdPs or proxies supports behavioral autoconfiguration via signed multilateral federation metadata (whether that may be Shibboleth, the Cirrus Identity Bridge, or another IdP or proxy). Each of the scenarios below may be deployed in a fashion in keeping with the Kantara Initiative's SAML2 Interoperability Deployment Profile, which both ensures proper processing of federation metadata and prepares a deployment for participation in InCommon (and other multilateral federations). To ensure interoperability across multilateral federations in these scenarios, it is important that the IdP actually registered and participating in the federation be (the) one which supports saml2int (whether it's the login or linked IdP). Full documentation of current requirements for InCommon participation by IdPs (including the latest Baseline Expectations requirements) can be found in the InCommon Federation Library. The Working Group recommends that all deployers review and address these recommendations and requirements, regardless of the scenario(s) they may be implementing.

A note on Linking vs Proxying: It is understood that there can be confusion around linking vs proxying as it relates to this topic of linking sso. The working group agreed that proxying refers to the method of how the SSO systems are linked. A proxy method can involve multiple Idp systems linked together though a chained saml requests (IdP acts as an SAML sp to another SAML IdP). Other proxy methods can involve different protocols on each side of the linked systems. For example ADFS can act as an transparent proxy to translate the incoming WS-Trust request from Azure AD into a SAML request to Shibboleth. Not all linked SSO scenarios are achieved via a proxy method.

Table of Linked Scenarios

Login IdP	Linked IdP(s)	Method of Link/Trust	Significant issues	Significant benefits
ADFS	Shib	Shib natively proxied to ADFS vis SAML Auth Proxy	No one in the working group does this so we cannot provide issues or benefits.	No one in the working group does this so we cannot provide issues or benefits.
Azure AD	Shib	Shib natively proxied to AAD with SAML Auth Proxy	• Requires that accounts exist both in Azure AD and in an on-prem directory/data source with synchronization and connected identifiers (UPN) (it is possible to connect Shib to Azure AD for data source using some extra tools, but that drops your ability to have a back up authentication in case of Azure or Internet outage)	• Unified sign in page • Shib gets some AAD Conditional Access features • Unified sign-in logs • Shib gets some automated risk-based protections (CARTA)

Login IdP	Linked IdP(s)	Method of Link/Trust	Significant issues	Significant benefits
			• Requires Azure/Internet Connectivity to be available unless you set up a faildown authentication method for Shib	• Shib gets online fraud detection (OFD) • Allows for usage of Azure B2B accounts with Shib protected resources • Allows for setting up a second authentication method within Shib for usage when Azure/Internet has an outage
Azure AD	CIB	CIB federated to AAD	Additional cost. Why pay to have a 2nd IdP?	• Shib SPs can continue to use CIB w/o impact • CIB provides full AAD CA features & CARTA
Okta	Shib	Dependent Campus login forwarding to Okta for medical center accounts (button based)	• Requires heavy Shib code customization managed by a third party • Difficulty with AuthnContextRef (Okta has a static list) • Requires a data feed to map Okta account to campus account (Often different)	• Allows Med Center access to inCommon services
Shib	Azure AD	AAD federated to Shib	• AAD Device join broken; can't use Intune (WS-Fed needed) • Difference in token lifetimes leads to arcane issues	• Unified sign in page (with username required to be entered twice once at AAD and again at Shib)
Shib	Azure AD ADFS	AAD federated to ADFS which is federated to Shib	• Difference in token lifetimes for 3 linked IdPs is worse leading to higher rate of arcane issues • Requires expertise for 3 IdPs	• AAD Device join is not broken; can use Intune
Shib	Google	Google Federated to Shib	• All non-admin accounts are forced to use SSO. ◦ Non-person accounts (department shared inboxes, service accounts) require extra steps to set up and use. ◦ Granting temporary access to suspended Google accounts is challenging. • Inexperienced users sometimes also sign up for Google Two-Factor, adding confusion to their MFA experience	• No need to enforce Google Two-Factor to protect Gmail with MFA (provided via Shib) • "Login with Google" becomes a viable option, especially for very specialized services.

Login IdP	Linked IdP(s)	Method of Link/Trust	Significant issues	Significant benefits
			(and confusion to the help desk when they call about it). • We don't give out Google accounts to alumni, parents, and other affiliations that have SSO logins. Some campus "owners" of a service may want to use Login with Google, and are surprised when some of their intended users do not have Google accounts.	(Example: we had one department of 40 that wanted to use Asana. I don't recall if we needed a different kind of license to do SAML, or if it was just a matter of going with it until we had a reason not to.)
Shib	OAM CAS	CAS protected by OAM (via OAM webgate), OAM authentication delegated (via saml proxy) to Shib CAS was our original SSO system (password auth only, no MFA). When OAM was introduced we used an OAM webgate (apache module similar to ship sp reverse proxy, but oracle propriety protocol). The OAM webgate handles OAM auth processing and passes REMOTE_USER back to CAS. 2021-2022 Shib was introduced to take over SSO duties from both CAS and OAM. OAM was configured as an SP to shib which allowed us to delegate all OAM/CAS authentication to	• The logout process between OAM and CAS was very complex and was inconsistent across apps.	• Unified the login experience for CAS, and OAM, and ultimately Shib, ADFS, and AAD • Allowed for a transition state where Legacy linked sso systems (CAS and OAM) were linked up to the future state linked sso systems (AAD/ADFS/Shib). This allows us to gradually move apps at a pace that is reasonable to all involved (other sys admins, app admins, vendors, etc)

Login IdP	Linked IdP(s)	Method of Link/Trust	Significant issues	Significant benefits
		Shib to facilitate a temporary transition state until all apps are moved off of OAM & CAS.		
Shib	Azure AD ADFS	AAD to ADFS to Shib. Shib is the primary IDP for campus. • Shib login screen handles all password auth • Shib's MFA logic will trigger Azure MFA (saml proxy to AAD) for Shib integrated SPs that require MFA (conditional access at the shib layer instead of putting MFA in front of the entire IDP)	• Need to Administer, secure, and maintain ADFS - Not a huge issue as ADFS's only role is to sit between AAD and Shib. In our environment no other apps are integrated with ADFS so not a whole lot of time spent on ADFS other than patching, cert management, and failover testing. • Initially complex when defining the architecture and working through implementation, but this issue has been mitigated with training, documentation and experience. • The difference in Token/Session lifetime across azure AD and Shib was a theoretical pain point during implementation, but since go live so far (May 2022) it doesn't seem like it is generating a lot of calls or pain for users. Well know more as we get deeper into the fall 2022 semester. • Minor: When auth request is coming from AAD through ADFS we set Shib as the default CPT so ADFS does not prompt user (Set-AdfsRelyingPartyTrust) • Minor: Shib's MFA logic will select the SAML to AAD flow when MFA is required. We had to create an HRD with `"AccelerateToFederatedDomain`":true and attached it to the Shib Enterprise App in ADD so that the users are not prompted by AAD for email address during the MFA flow. • Needed some custom cookie handling code so the Shib logout velocity templates would know to display the button for the upstream AAD logout process if an MFA protected application was accessed during the SSO session that is being logged out of.	• A unified login experience for both Office 365 and all other SSO integrated applications - including resources in multilateral federations • MS Intune and Autopilot functionality operates as expected • Ability standardize on a single MFA technology (Azure MFA) for O365 email, Cisco VPN, and any SSO app that requires the additional security • No ADFS or AAD dependencies for SSO transactions that do not require MFA. If/When AAD has any type of outage only MFA protected apps will impacted • Our business office appreciates the "Exit Strategy" without needing to refactor all SSO integrations. If the university ever decides to move away from Azure AD/MFA we would only need to refactor the integration between shib and the MFA mechanism • Our UX folks appreciate the ability to

Login IdP	Linked IdP(s)	Method of Link/Trust	Significant issues	Significant benefits
				customize the login page to align with other aspects of university branding, and mitigate the poor user experience with the global microsoft sign-in screens since they include hardcoded links to Microsoft SSPR ("can't access your account?", "Forgot my password", etc) We don't have writeback and leverage our own account management portal.
Shib	Azure AD	Azure AD federated directly with Shib using WS-Fed/Trust code within the Shib IdP	• Customized code that is not part of Shib	• Unified sign in page. Limited concern over users having to enter their username potentially twice for Azure services since that's a common discovery pattern for other SPs as well. • AAD Device join works with InTune • Not tied to a specific cloud vendor. Centralizing in Shibboleth may be advantageous in the future when there may be other cloud integrations.
Shib	ADFS	ADFS federated with Shibboleth as a claims provider, with HRD override to avoid ADFS interrupting flow.	• Without home realm discovery override (which requires a small bit of custom shim code in C# for pre-ADFSv4 and in JS for ADFSv4), users first have to select "Shibboleth" as their claims provider and then authenticate.	• Consistent user login experience – both SAML and WS-Fed RPs "look" like they're using Shib to end users

Login IdP	Linked IdP(s)	Method of Link/Trust	Significant issues	Significant benefits
			• ADFS becomes the relying party for Shibboleth, and obscures all downstream RPs (so attribute release, etc., is universalized – whatever any ADFS RP needs must either be minted fresh from the AD by ADFS or released by Shib to ADFS regardless of the target RP beyond ADFS. • Cookie accumulation can become problematic for some use cases, due to ADFS operating as both a RP (to Shib) and an IDP (for its own RPs). • Session timeouts need to be carefully managed between Shib and ADFS to avoid confusing behaviors	• Shibboleth extensions for MFA, WebAuthN, etc. apply to both SAML and WS-Fed consumers • ADFS can add attribute information (including MS-specific attributes) in-line • ADFS fully supports WS-Trust as well as WS-Federation, enabling some non-web-based scenarios (with caveats) • We instituted this model well before AAD was available in order to support some 3rd party apps (from API, Ellucian, and other vendors) that could not do federated SSO via SAML, but did support WS-Fed (and in one case, WS-Trust). We moved away from it after shifting to model #2 in support of Office 365 and finding that ADFS was difficult to scale across multiple DCs in our environment. We migrated to model #1, then landed in our current model #10 arrangement after developing custom code to handle WS* protocols in the Shib IDP directly.
CAS	Shib	Shibboleth federated with CAS as a	• Unicon module is only partially updated to IDP 4.1+ module/plugin system, installation not yet seamless.	• Consistent login screen for our users. (With

Login IdP	Linked IdP(s)	Method of Link/Trust	Significant issues	Significant benefits
		registered application via the Unicon shib-cas-authn module.	• (We) Did not attempt to support MFA on CAS environment, limiting CAS use for sensitive applications. • There is some interoperability issues between various CAS and Shib versions, especially when forceauth / renew is in use which can present extra challenges during CAS and Shib upgrades, potentially including some custom code to avoid apparent incompatibilities between specific CAS and Shib / Unicon module versions. • Different session timeouts and triggers, only way to end SSO session is to close browser – risk of zombie / reanimating sessions. • For a variety of technical and non-technical reasons, did not attempt to leverage this SSO environment for Azure AD. *(Syncing password hashes for unified sign-on.)*	"frameless" Duo, end users don't even know that Shib is handling the MFA part of the process.) • CAS is a simpler protocol for some internal and external services. • Our web dev team is able to deploy apps requiring SSO with out IAM / SSO team support. (Via pre-agreed hostname patterns) *(Has made OIDC support less urgent in our environment.)*

At this point, the working group believes its work to be finished. We intend to keep the group's mailing list available as a point of contact should someone have questions about the table. We can also see a future group being convened to create HOW-TO documents for the different scenarios, which was determined to be out of scope for this working group.

No labels

7 Comments

Etan Weintraub (johnshopkins.edu)

1. Detailed feedback from CACTI to the WG
    a. The WG report should probably explicitly note that the scenarios included in the table within it do not constitute an exhaustive accounting of possible linking scenarios, but rather cover the scenarios with which the WG had collective experience.
    b. The WG report could use some introductory information about the rationale for operating multiple SSO systems. The introduction to the WG's charter has some text that can probably be repurposed in the report to fill that gap.
    c. The WG needs to consider consumption and automatic configuration of an IdP based on InCommon multilateral SAML metadata as a requirement for the IdP that gets published in InCommon. This is why, for example, Cirrus Bridge exists as a solution for Azure AD and Okta.
    d. Linking two IdPs together is the core value proposition for being able to federate an otherwise non-InCommon-compatible IdP with InCommon.
    e. Community may be confused by the distinction between proxying and linking. May need to make this clearer, what you can do with these options and why you would want to do them. Or, if the distinction isn't necessary for public understanding of the recommendations, remove the distinction.
    f. The report should call out the requirements of the Kantara SAML 2 Deployment Profile for Federation Interoperability, both as requirements for the "thing that gets published in InCommon" under these linking recommendations, and so that the reader community becomes reminded of / aware of this foundational documentation: https://kantarainitiative.github.io/SAMLprofiles/saml2int.html . It may be worth noting that the scenarios reviewed by the WG **can** all be deployed in compliance with the deployment profile (or noting any which cannot).
    g. InCommon Federation Library (documentation of InCommon Federation requirements): https://spaces.at.internet2.edu/display/federation

Etan Weintraub (johnshopkins.edu)

I will pull the table and information from Key Linked IdP Scenarios directly into this document to make one document.

Etan Weintraub (johnshopkins.edu)

Rob will take c/f/g - linking the Kantara stuff and InCommon Federation requirements...

Rob Carter (duke.edu)

I added a paragraph between the glossary and the actual table that **may** cover c/f/g – at least it starts in that direction (and it's relatively brief).

Etan Weintraub (johnshopkins.edu)

Majeed will take b/d/e and inviting others for more scenarios, and qualifying linking vs. proxying (e)

Majeed Abu-Qulbain

Finally got some updates in:

b: As suggested, I did repurpose some of the content from the working group charter to create an introduction section.

d: I added this in the last paragraph of the intro

Added a sentence at the end of the first paragraph after the glossary inviting folks to reach out to the linking_sso_working_group@incommon.org mailing list if they are interested in adding another linked scenario to the list that wasn't there before.

e: I added "**A note on Linking vs Proxying"** just before the table of linked scenarios. I'm not 100% sure the distinction is helpful - curious what others think - feel free to nuke it.

Also added a table of contents - it provides a minor outline with links which seems alright

Lastly, In the IdP glossary i did give "OAM" a "No" on supporting multilateral federations - We couldn't find any ootb documentation around it or any oracle support articles mentioning it. It might be possible with some metadata import scripting, but I wouldn't call that native.

Etan Weintraub (johnshopkins.edu)

Adjustment to (d) - A core value, not THE core value