building-incommon-ready-software

Jump to:

Before We Begin | General Considerations | Available SAML Implemenations | Writing Your Own SAML Implementation

This article offers implementation tips and advises to software makers who wish to produce software solution that integrate well with federated single sign-on (SSO) identity providers in the InCommon Federation.

Before We Begin

Research and education (R&E) identity federations worldwide, including InCommon in the United States, use SAML as its federated single sign-on protocol. Creating InCommon-ready software starts with supporting SAML. However, unless your software's primary purpose is to implement the SAML protocol stack, we strongly suggest adopting one of the existing InCommon-ready SAML implementations rather than building your own. These software reduce and/or eliminate the effort required to create and maintain customer code to handle the nuances of SAML protocol suite and to support the various federation integration profiles.

See: Available SAML Implementations.

General Considerations

Whether you build your own SAML software or not, an InCommon-ready software needs to allow its deployer to configure the software to properly interact with identity providers in the InCommon Federation using federation-endorsed integration profiles and practices. In particular, an InCommon-ready software should cover the following areas:

1. Support SAML 2 and Related Federation Profiles

Whenever possible, adopt a SAML 2.0 implementation that has proper support for requirements outlined in the Kantara SAML V2.0 Implementation Profile for Federation Interoperability. If you are writing your own SAML implementation, see Build Your Own SAML Software.

2. Consume SAML Metadata from the InCommon Metadata Service

A major benefit of participating in the InCommon Federation is that InCommon offers a trusted and scalable way for identity providers and service providers exchange service metadata and cryptographic keys. By consuming SAML metadata from the InCommon Metadata Service, service operator can automatically detect changes from any participating identity providers and dynamically update configurations. An InCommon-ready software should be able to consume an identity provider's metadata from the InCommon Metadata Service.

See:

- InCommon Metadata Service
- Understand SAML Metadata and Its Use in InCommon
- Trust Relationships for Access Management - The InCommon Model

3. Engineer Flexible Attribute Mapping Schemes

<basically, make sure your software or the SAML module you choose can be configured to map /consume federation attributes>

Topics

- Working with user data

4. Choose and Handle User Identifiers with Care

<provide SP-specific considerations when working with user identifier, including multi-lateral federation concerns (there might be multiple IdPs involved.>

5. Design the Right Federated SSO User Experience

Related Content

- Building InCommon-Ready Software
- Choosing the Right Federated User Identifier
- Determine How Your Offering Contributes to Providing Identity
- Make Your Identity Service Federation Ready
- Make Your Service Federation Ready
- Register a Resource in InCommon
- Determine How Your Offering Contributes to Providing a Service
- Step-by-Step Guide to Offering Services in the InCommon Community
- Make Your Platform Federation Ready
- Step-by-Step Guide to Providing Federated Identity Access and Management (IAM) Services

References

Kantara SAML V2.0 Implementation Profile for Federation Interoperability (fedinterop)

Kantara SAML V2.0 Deployment Profile for Federation Interoperability (saml2int)

OASIS Security Assertion Markup Language (SAML) V2.0

Shibboleth website

SimpleSAMLphp website

OpenSAML wiki

Get Help

Can't find what you are looking for?

help Ask the community

- Recognize that your software does not have complete control of user experience; the user's identity provider is your partner. This has a implications, such as:
    - Federated identifiers are often long and "ugly" in order to provide global uniqueness and long-term stability. If you need a friendly name for interacting with the current user, provide the ability to request the user's name as a separate attribute from the identifier.
    - After displaying an error message specific to your application, make use of the error URL advertised in federation metadata by the identity provider, for example when you do not receive required attributes.

- While federated identifiers look like electronic mail addresses, *they are not*. If you need an electronic mail address for the current user, provide the ability to request it as a separate attribute.
- There are multiple opportunities to distinguish yourself within the marketplace of federation-capable software to enhance Platform Deployers' and Services Operators' roles within the ecosystem. It is strongly suggested that you familiarize yourself with Step-by-Step Guide to Offering Services in the InCommon Community and the documents it references for such opportunities for your user community. The Cloud Services Cookbook also provides valuable tips.

Available SAML Implemenations

These are a few examples of available SAML implementations:

- **Shibboleth**, specifically the Shibboleth Service Provider module, is a free, open-source SAML software maintained by the Shibboleth Consortium. It installs as a web server plug-in and offloads all of the SAML processing from your software. Shibboleth is built by and for the R&E federated identity community. It is widely adopted and fully supports InCommon interoperability profiles and standards.
- **SimpleSAMLphp** is an open-source PHP implementation of the SAML protocol. It is a popular choice among the PHP software community.
- **OpenSAML** is a set of open source C++ & Java libraries developed to support Shibboleth Project's implementation of the Security Assertion Markup Language (SAML). It is licensed under the Apache 2.0 license.

Writing Your Own SAML Implementation

If you decide to write your own SAML implementation, to ensure your implementation will work well in the InCommon Federation, make sure your software conforms with the Common Requirements and Service Provider Requirements of the Kantara SAML V2.0 Implementation Profile for Federation Interoperability. Published by the Kantara Initiative, the SAML V2.0 Implementation Profile encompasses a set of software conformance requirements intended to facilitate interoperability within full mesh (multilateral) identity federations, such as those found in the research and education sector, including the InCommon Federation.

Further, check out Kantara SAML V2.0 Deployment Profile for Federation Interoperability. Where as the Implementation Profile is written for software makers, the Deployment Profile helps service operators deploy InCommon-Ready services. Your software should allow a service operator using your software to fully conform with the requirements in the Deployment Profile.