XSEDE Identity Management Use Cases

January 6, 2017
Version 1.3

These use cases describe how researchers, scientists, and other community members register themselves with the XSEDE system, manage their profile information, and authenticate their identities when using XSEDE services.

Use cases 1-10 were documented during XSEDE’s first five-year project period and served as the basis for a reimplementation activity. They have been reformatted to the newer XSEDE-2 use case format. More detail was provided in their original versions.¹ Use cases 11 and beyond were added during XSEDE-2.

IDM-1: Register with XSEDE
IDM-2: Login to XSEDE user portal with XSEDE username/password
IDM-3: Change an XSEDE user profile
IDM-4: Login to XSEDE user portal with a non-XSEDE identity
IDM-5: Link or unlink a non-XSEDE identity
IDM-6: Login to a science gateway with an XSEDE identity
IDM-7: Login to a locally installed application with XSEDE username/password
IDM-8: Login to a locally installed application with SSH/X.509 key
IDM-10: Authenticate to an XSEDE identity using WS-Trust Secure Token Service
IDM-11: Use an XSEDE identity for InCommon authentication
IDM-12: Single sign-on for XSEDE OpenStack resources
IDM-13: Authenticate to XSEDE OpenStack APIs

History

¹ XSEDE Identity Management Use Cases, version 1.2. July 29, 2015. (https://docs.google.com/document/d/1kgDoOpZOwaxbaBQpRvOB2Xef64uUgQuD1Cw8ly8e_k)
IDM-1: Register with XSEDE

A person needs to register his or her identity in order to use XSEDE services and resources.

In most cases, the person wants to experience it as follows.

1. The person opens a Web browser and visits www.xsede.org.
2. Then, the person clicks a “Create account” link or button and goes through the registration process.
3. Finally, the person completes the process, at which point he or she should have a username that represents him/her when using XSEDE services, and a confidential password that he/she can use to prove his/her identity. The XSEDE system should have an XSEDE user profile that contains the information provided by the person during the registration process.

We’ll accept any solution to this problem, as long as the following are true.

1. The registration process should be secure from third parties.
2. The person may choose his/her own username as long as no one else has registered with it.
3. At least 100 people can be using the registration system at the same time.
4. The system can register at least three new identities per minute.
5. The web interfaces should consistently respond to user actions in less than three seconds.
6. The registration interface should be available and working as described at least 99.9% of the time.

IDM-2: Login to XSEDE user portal with XSEDE username/password

An XSEDE user needs to login to the XSEDE User Portal (XUP) using his or her XSEDE username and password. We assume the user has already registered with XSEDE.

In most cases, the XSEDE user would like to experience it as follows.

1. The user opens a Web browser and visits www.xsede.org, which includes a form asking for a XSEDE username and password
2. Then, the user enters his/her XSEDE username and password.
3. Then, the user clicks “Sign In.”
4. If the username and password is incorrect, then the user is told the username or password is incorrect and they can try again.
5. If the username and password is correct, then the user is logged in.

We’ll accept any solution to this problem, as long as the following are true.

1. In Step 4, a delay should be built into the response to inhibit brute-force password guessing.
2. The interface should provide responses to user actions in less than three sec for all steps EXCEPT the one in which the user inputs an incorrect username and/or password. That step could take longer, but probably shouldn’t take less that 2 sec.
3. At least 25 people should be able to be in Step 3-5 at the same time, and at least 200 people
   should be able to be in Steps 1-2 simultaneously.
4. The interface must be available and working as described at least 99.95% of the time.

**IDM-3: Change an XSEDE user profile**

An XSEDE user wants to change his/her XSEDE user profile. We assume the user has already registered
with XSEDE.

In most cases, the XSEDE user wants to experience it as follows.

1. First, the user logs into the XSEDE User Portal (XUP).
2. Then, the user goes to their profile page.
3. Then, the user clicks “Edit profile.”
4. Then, the user changes some fields (e.g., email address, name, etc.).
5. Then, the user clicks “Save changes.”
6. Finally, the user is redirected back to the XUP profile page, which displays the updated profile.

We’ll take any solution, as long as the following are true.

1. If the user has two-factor authentication enabled, Step 3 will require the user to authenticate via
   a second factor before proceeding.
2. If the user changes his/her email address, it should not take longer than one minute for the
   email system to emit a validation email message after Step 5.
3. At least 25 users can be using the profile interface at the same time.
4. At least three users can submit profile changes per minute.
5. The interface must be available and working as described at least 99.9% of the time.

**IDM-4: Login to XSEDE user portal with a non-XSEDE identity**

An XSEDE user needs to login to the XSEDE user portal (XUP) using a non-XSEDE identity. We assume the
user has already registered with XSEDE. (See IDM-1.)

In most cases, the XSEDE user wants to experience it as follows.

1. First, the user clicks “Sign In” on the XUP home page.
2. Then, the user clicks “Other Sign In Options.”
3. Then, the user will be redirected to an XSEDE-branded page where they can pick their identity
   provider and optionally select “remember this selection” so that they do not see this page again.
4. Then, the user will be redirected to the selected identity provider, where they login. (If the login
   fails, the user may try again at the identity provider.)
5. Once the login succeeds, if the identity has not been previously linked to an XSEDE account, the
   user is asked if they want to create a new XSEDE account or link the identity with their existing
   XSEDE account.
a. If the user chooses to create a new XSEDE account, they get bounced to XUP to create an account. (See IDM-1.)

b. If the user chooses to link to an existing XSEDE account, XUP prompts the user to enter their XSEDE username and password.

6. Finally, the user is logged in to XUP and associated with his/her XSEDE username.

We’ll take any solution, as long as the following are true.

1. In Step 3, the interface should have familiar XSEDE branding (appearance and style) and should have a web address of the form *.xsede.org.
2. In Step 3, the XUP and XSEDE services may choose which identity providers they trust as linked identities.
3. The interface should respond to user actions in less than three seconds.
4. At least 25 users can be in Steps 1-2 and 4 at the same time.
5. At least 200 users can be in step 5 at the same time.
6. The interface must be available and working as described at least 99.95% of the time.

**IDM-5: Link or unlink a non-XSEDE identity**

An **XSEDE user** wants to link or unlink a non-XSEDE identity with their XSEDE identity. A linked identity can be used to authenticate to XSEDE instead of an XSEDE username and password. We assume the user has previously registered with XSEDE and has also previously registered with another organization.

In most cases, the **XSEDE user** wants to experience it as follows.

1. First, the user logs into the XSEDE user portal (XUP).
2. Then, the user navigates to his or her XUP profile page.
3. Then, the profile page displays his/her linked identities with a link to “Edit linked identities.”
4. Then, the user clicks on that link and is redirected to another XSEDE-branded web page that lists the linked identities, allows each to be removed, allows new linked identities to be added, and allows the default web-based login identity to be changed or cleared (i.e., this configures the “remember this selection” on the federated identity login page)
   a. Note: The XUP will be able to choose which identity providers it shows as linked identity providers. Users may have linked identities from other providers (e.g., if they linked through non-XSEDE branded sites), but XSEDE will only trust logins from those it accepts.
5. Finally, when the user is done with these activities, he/she clicks “Done” and is returned to the XSEDE profile page, which displays the updated information.

We’ll take any solution, as long as the following are true.

1. The solution should support linking identities from most web-based federated organizations, such as InCommon-participating campuses, Google, etc.
2. The solution should have future plans to allow linking X.509 identities and SSH public keys.
3. At least 25 users can use the interface at the same time.
4. At least three users can submit identity linking changes per minute.
5. The interface must be available and working as described at least 99.9% of the time.

**IDM-6: Login to a science gateway with an XSEDE identity**

An XSEDE user needs to be able to login to a science gateway (a non-XSEDE web application) using their XSEDE identity, such that the science gateway can securely interact with XSEDE services on behalf of the user. We assume that the user has previously registered with XSEDE, and that the science gateway has registered as a client of XSEDE’s Globus Auth service.

In most cases, the XSEDE user wants to experience it as follows.

1. First, the user points a Web browser at the science gateway web site.
2. Then, the user clicks “Login,” “Login with XSEDE,” or a similar link or button.
3. Then, the user is directed to an XSEDE-branded login page.
   a. The user may login using their XSEDE username and password or may instead use another identity that is linked to their XSEDE identity.
   b. If the User logs in with a linked identity that is not linked to an XSEDE account, they are given the option to enter and XSEDE username and password to link it, or to create a new XSEDE account.
4. Finally, the user is directed back to the science gateway, and the gateway is aware of the user’s XSEDE username and has a delegated credential that it can use to access XSEDE services on behalf of the user.

In most cases, the science gateway wants to experience it as follows.

1. When user clicks “Login” in Step 2...
   a. The gateway (SG) performs a standard OAuth2 Authorization Code Grant (OAuth2 spec section 4.1), extended with OpenID Connect, against Globus Auth.
      i. SG redirects browser to XSEDE-branded, Globus-provided OAuth2 login page.
         1. Globus takes care of everything in step #3
         2. When complete, Globus redirects back to the SG, with an OAuth2 authorization token.
      ii. SG exchanges the OAuth2 authorization token for an OAuth2 access token with the Globus Auth API (OAuth2 Access Token Request)
   b. SG can get the User’s identity by either:
      i. The OAuth2 access token returned in the previous step will include a standard OpenID Connect id_token, which includes the XSEDE identity (i.e., XSEDE Kerberos principal)
      ii. SG can call the Globus Auth API to request the XSEDE identity.
   c. SG can use the OAuth2 access token to get other User information, such as:
      i. SG can call Globus Auth API to get other linked identities, and validate email addresses.
      ii. SG can call Globus Groups API to get the list of groups that the user belongs to
      iii. SG can call the Globus Attributes API to query for attributes associated with the User’s XSEDE or linked identities.
   d. SG proceeds as normal.

We’ll accept any solution as long as the following are true.
1. XSEDE’s interfaces should provide responses to user actions in less than three seconds EXCEPT when the user enters an incorrect XSEDE username and/or password. That step could take longer, but probably shouldn’t take less than two seconds.

2. At least 25 users can be in step 3 at the same time.

3. At least 200 users can be in Step 4 at the same time.

4. XSEDE’s interfaces must be available and working as described at least 99.95% of the time.

**IDM-7: Login to a locally installed application with XSEDE username/password**

An **XSEDE user** needs to login to a locally installed **application** (a command line program, graphical desktop application, or mobile application) using his/her XSEDE username and password, such that the application can securely interact with XSEDE services on behalf of the user. We assume that the user has previously registered with XSEDE, and that the application is registered as a client of XSEDE’s Globus Auth service.

In most cases, the **XSEDE user** wants to experience it as follows.

1. First, the user starts the application.

2. Then, the user is prompted by the application for his or her XSEDE username and password. (Note: username could either be just “username” or “username@domain”. If no domain, there must be a defined default domain.)

3. Finally, the user’s login succeeds or fails.

In most cases, the **application** (TC) wants to experience it as follows.

1. When the user provides his/her username and password in step #2...
   a. TC calls Globus Auth API (OAuth2 Resource Owner Password Credentials Grant, section 4.3) with username and password
      i. Globus replies with an OAuth2 access token or a fault
   b. TC can get the User’s identity by either:
      i. The OAuth2 access token returned in the previous step will include a standard OpenID Connect id_token, which includes the XSEDE identity (i.e., XSEDE Kerberos principal)
      ii. TC can call the Globus Auth API (GET /v2/token_details) to request the XSEDE identity.

We’ll accept any solution as long as the following are true.

1. Fault codes must be well-specified and understood. (The application can programmatically interpret what went wrong from the fault code.)

2. The Globus Auth API should provide responses to requests in less than three seconds for all cases EXCEPT when the user provides an incorrect username and/or password. In that case, the response shouldn’t take less than two seconds.

3. At least 25 locally installed client applications can be authenticating at the same time.

4. The interface must be available and working as described at least 99.95% of the time.
**IDM-8: Login to a locally installed application with SSH/X.509 key**

Un XSEDE user needs to login to a locally installed application (a command line program, graphical desktop application, or mobile application) using a previously linked SSH key or X.509 certificate, such that the application can securely interact with XSEDE services via REST interfaces on behalf of the user. We assume that the user has previously registered with XSEDE and that the application is registered as a client of XSEDE’s Globus Auth service. We also assume that the user has previously associated an SSH/X.509 certificate with his/her XSEDE identity via use case IDM-5.

In most cases, the **XSEDE user** wants to experience it as follows.

1. First, the user starts the application.
2. Then, the user is prompted by the application for the password to unlock the user’s local SSH or X.509 private key.
3. Finally, the user login succeeds or fails.

In most cases, the **application** (TC) wants to experience it as follows.

1. After user provides password in step #2...
   
   a. TC calls Globus Auth API (Resource Owner SSH or X.509 Credentials Extension Grant, which leverages the standard OAuth2 Extension Grant) with an HMAC signed by the private key
   
   i. Globus replies with an OAuth2 access token or a fault.
   
   b. TC can get the User’s identity by either:
   
      i. The OAuth2 access token returned in the previous step will include a standard OpenID Connect id_token, which includes the XSEDE identity (i.e., XSEDE Kerberos principal)
      
      ii. TC can call the Globus Auth API (GET /v2/token_details) to request the XSEDE identity.

We’ll accept any solution as long as the following are true.

1. Fault codes must be well-specified and understood. (The thick client can programmatically interpret what went wrong from the fault code.)
2. The Globus Auth API should respond to requests in less than three seconds in all cases EXCEPT when the user provides an invalid key or certificate. In that case, the response shouldn’t take less than two seconds.
3. At least 25 locally installed client applications can be authenticating at the same time.
4. The interface must be available and working as described at least 99.95% of the time.

**IDM-10: Authenticate to an XSEDE identity using WS-Trust Secure Token Service**

An XSEDE user needs to authenticate to his or her XSEDE identity using an application that supports the WS-Trust Secure Token Service (WS-trust STS), such that the application can securely interact with XSEDE services on behalf of the user. We assume that the user has previously registered with XSEDE and that the WS-Trust STS is registered as a client of XSEDE’s Globus Auth service.

In most cases, the **XSEDE user** wants to experience it as follows.

1. The user starts the application.
2. The user is prompted for an XSEDE username and password by the application.
3. The user’s login succeeds or fails.

In most cases, the **application** (TC) wants to experience it as follows.

1. When user provides username and password in Step 2...
   a. TC calls STS with username and password.
   b. STS returns appropriate SAML token(s).

In most cases, the **WS-Trust STS** (STS) wants to experience it as follows.

1. When user provides username and password in step #2...
   a. STS calls Globus Auth API ([OAuth2 Resource Owner Password Credentials Grant](https://docs.globus.org/auth-api/oauth2)) with username and password, using the same interaction as Use Case UC IDM 7
      i. Based on the @domain of the username, Globus Auth will authenticate with the appropriate identity provider using the username and password (e.g., via XSEDE Kerberos, InCommon SAML ECP). Note that not all identity providers support non-browser-based authentication.
      ii. Globus replies with an OAuth2 access token and OpenIDConnect id_token, or fault.
   b. (future) STS calls Globus Groups API, passing in the received OAuth2 token, to get the groups (UUIDs) and roles (admin, manager, member, observer) the user has.
   c. STS turns appropriate SAML tokens

We’ll accept any solution as long as the following are true.

1. The Globus Groups API should take no longer than three seconds to return group membership/role information.
2. Fault codes must be well-specified and understood. (The WS-Trust STS and application can programmatically interpret what went wrong from the fault code.)
3. The WS-Trust STS should provide responses to requests in less than three seconds for all cases EXCEPT when the user provides an invalid username and password. That response shouldn’t take less than two seconds.
4. At least 25 applications can be using the WS-Trust STS simultaneously.
5. The interfaces must be available and working as described at least 99.95% of the time.

**IDM-11: Use an XSEDE identity for InCommon authentication**

A **researcher** wants to authenticate to a national CI service that accepts InCommon identities. We assume that the researcher has previously registered with XSEDE.

In most cases, the **researcher** wants to experience it as follows.

1. First, the researcher visits a national CI service that accepts InCommon identities.
2. Then, the researcher selects XSEDE as his/her InCommon identity provider, authenticates to XSEDE, and allows XSEDE to share his/her XSEDE identity information to the CI service.
3. Finally, the national CI service accepts his/her XSEDE identity and provides appropriate access to
the service per its own use policies.

We’ll take any solution, as long as...

1. The solution must be compatible with InCommon authentication protocols.
2. The solution should use the normal XSEDE XUP registration mechanism for signing up with
XSEDE in Step 1.
3. The solution should use the normal XSEDE authentication mechanism in Step 3.
4. The XSEDE authentication in Step 3 should release the researcher’s research & scholarship
attributes.
5. The solution should not require the researcher to have an active or prior XSEDE allocation.

IDM-12: Single sign-on for XSEDE OpenStack resources

An XSEDE-allocated researcher wants to be able to authenticate once using his/her XSEDE identity and
subsequently have authenticated access to all of the available XSEDE OpenStack resources. (These
services might include things such as a central object library (e.g. images, date, etc.), objects stored by
that user on other service provider’s resources, or cloud computing features such as elastic scheduling of
instances.) We assume the researcher has previously registered with XSEDE.

In most cases, the researcher wants to experience it as follows.

1. First, the user would connect to a service provider’s authentication web page.
2. Then, the user will be redirected to the Identity Provider’s (IdP) web page.
3. Then, the user will be prompted for authentication credentials.
4. Then, upon successful authentication, the user will be redirected to the SPs endpoint and an
unscoped token will be returned; included in this token are a list of Identity service groups that
the user belongs to.
5. Then, this unscoped token can be used to identify which groups and domains are accessible.
6. Then, the unscoped token, along with appropriate domain and project information, can be used
to request a scoped token.
7. Finally, with the scoped token, the user can access services and objects that are available to that
user on other XSEDE OpenStack service providers.

It’ll always be like that, except when the user is not using a web browser, and the Enhanced Client or
Proxy (ECP) is used as an alternative mechanism.

We’ll take any solution, as long as the central XSEDE authentication service accepts XSEDE usernames
and passwords and returns an unscoped Keystone token.

IDM-13: Authenticate to XSEDE OpenStack APIs

An XSEDE-allocated researcher wants to authenticate to the APIs offered by an XSEDE OpenStack
resource using his/her XSEDE identity. We assume the researcher has previously registered with XSEDE.

In most cases, the researcher wants to experience it like the following steps.
1. First, the researcher connects to a web page provided by the service provider and login using his/her XSEDE ID.

2. Then, the researcher navigates the service provider’s site to find OpenStack API credentials.

3. Finally, the researcher can configure his/her application to use the OpenStack API credentials to access OpenStack services that are available to that user on the resource.

We’ll take any solution, as long as Step 2 provides standard client_id+client_secret credentials that can be used with OpenStack APIs.
## History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Changes</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entire Document</td>
<td>1.2</td>
<td>7/29/2015</td>
<td>Use cases 1-10 documented in preparation for an XSEDE-1 re-implementation activity.</td>
</tr>
<tr>
<td>Entire Document</td>
<td>1.3</td>
<td>1/6/2017</td>
<td>Use cases 1-10 rewritten using XSEDE-2 format; revised use case 3; added use cases 11-13.</td>
</tr>
</tbody>
</table>