**Canonical Use Case 6 - Authenticate to one or more SP resources, SP services, and XSEDE central services**

**Definitions**

Please refer to the *Glossary of Terms for Canonical Use Cases* document.

The use cases are presented here using the following format, derived from the Malan and Bredemeyer white paper.

<table>
<thead>
<tr>
<th>Use Case UC CAN 6</th>
<th>Authenticate to one or more SP resources, SP services, and XSEDE central services</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Users, groups, and services may need to authenticate themselves in order to access both XSEDE operated services and services operated by others. Rather than manually authenticating against each service before use, it is desirable to authenticate once and use some proof of authentication for subsequent interactions with the same or other services. This approach is sometimes known as single sign-on.</td>
</tr>
</tbody>
</table>
| **References**    | InCommon: [https://incommon.org](https://incommon.org)  
OAuth 2.0: [http://oauth.net/2/](http://oauth.net/2/)  
XSEDE L3 Decomposition document, 5.1.5, 5.2.5, 6.1.2, 6.3.1, 6.3.2  
**The following documents are available via the XSEDE Use Case Registry, [https://software.xsede.org/registry-dev/index.php](https://software.xsede.org/registry-dev/index.php).**  
XSEDE User Identity and Access Management (Canonical Use Case 9). v1.0, 30 July 2013.  
XSEDE Campus Bridging Use Cases, v1.5, 16 March 2012.  
XSEDE Data Analytics Use Cases, v0.3, 14 June 2013.  
XSEDE Federation and Interoperation Use Cases, v0.3, 22 January 2013.  
XSEDE High Performance Computing Use Cases, v0.81, 17 June 2013.  
XSEDE High Throughput Computing Use Cases, v0.3, 31 May 2013.  
XSEDE Resource Allocation System 1.0 Use Cases, v1.0, 25 September 2013. |
### Actors

Client

### Prerequisites (Dependencies) & Assumptions

Necessary software is deployed at service providers and to user/principals.

Intermediate services and relying parties all support chosen authentication and delegation mechanism.

### Steps

**Repeat:**

1. Client selects (implicitly or explicitly) an XSEDE-approved identity provider (IDP).
2. Client provides the selected IDP with one or more credentials.
3. IDP returns either one or more credentials to the client, or a fault (and no credentials). The IDP MAY in turn interact with other IDPs on behalf of the client to acquire other credentials.

Until client has acquired all necessary credentials.

**Repeat:**

4. Client uses the credential(s) via appropriate mechanisms with a relying party, which may include intermediate services, XSEDE services, and SP services.
5. (Optional) The client delegates one or more credentials or restricted credentials to the relying party.
6. (Optional) The relying party uses the credential(s) to decide whether to perform the service requested by the client (authorization).
7. (Optional) The relying party makes requests to other services on behalf of the client using its own credentials and/or using delegated credentials from the client.

Until client is done.

### Variations (optional)

a) The client manages multiple security contexts.
b) The client persists (stores) its security context for later use.

### Quality Attributes

**There may be some redundancy in these quality attributes**

1. The availability of the XSEDE IDP is mission critical and must be 99.95%. Assumptions: Availability is defined as the percentage of time in a given <month> that a service responds correctly. (NB: Correctness can include timing.)
2. Delegation steps must be cryptographically secure and use accepted best-practices and techniques.
3. Communication of credentials must be via a commonly accepted private channel such as TLS/SSL.
4. Cryptographic algorithm and key lengths to conform to current FIPS standards and NIST recommendations.
5. Auditability. It must be possible to log for audit purposes all credential creation actions and to cryptographically verify all delegation steps.
6. Communication channels use cryptographic methods (such as SSL/TLS) that provide integrity and confidentiality protection, resistance to replay and man-in-the-middle attacks, and mutual authentication. Such protection could be provided at transport layer or message layer.
7. Useability. Information about failures either in acquiring or using credentials should be informative and useful to both users and administrators.
9. The client may be either a program or a web browser.
10. The authentication mechanism MUST be able to federate with other mechanisms such as those used by OSG, PRACE, EGI, and InCommon.
11. It must be possible to add and delete credentials from the security context after the context has been created.

<table>
<thead>
<tr>
<th>Non-functional (optional)</th>
</tr>
</thead>
</table>

Issues