# Home

This wiki space exists to develop a technical solution and project plan to add support within the Shibboleth System software for at least one variant of the so-called "proxy authentication" problem, wherein a service to which a user may have authenticated wishes to invoke another service on the user's behalf. To motivate the project, the specific case of a portlet living within the uPortal software as the proxying service has been chosen, but the solution should apply to non-portal use cases.

Solving this problem in the general case is very complex, which is one reason it hasn't been done within Shibboleth to date. Instead, we are starting with a more tightly constrained use case that limits the scope of the problem; we will design for generality where possible, but not at the expense of compromising our ability to solve at least one basic use case without an overwhelming scope of work.

For a discussion of the use case, see below.

Separate topics cover:

- Solution Proposal
- Portal Design Issues
- IdP Change Proposals

***For editing access in this wiki space, see the access instructions.***

## The Use Case

We have five actors:

- User Agent
- User's Identity Provider (IdP)
- Portal Instance (assumed to be uPortal)
- Portlet
- Web Service Provider (WSP)

In a non-portal use case, the Portal and Portlet can be treated as a single actor consisting of any kind of intermediary service. The rest of the use case is essentially unchanged.

The sequence of high-level application interactions is as follows:

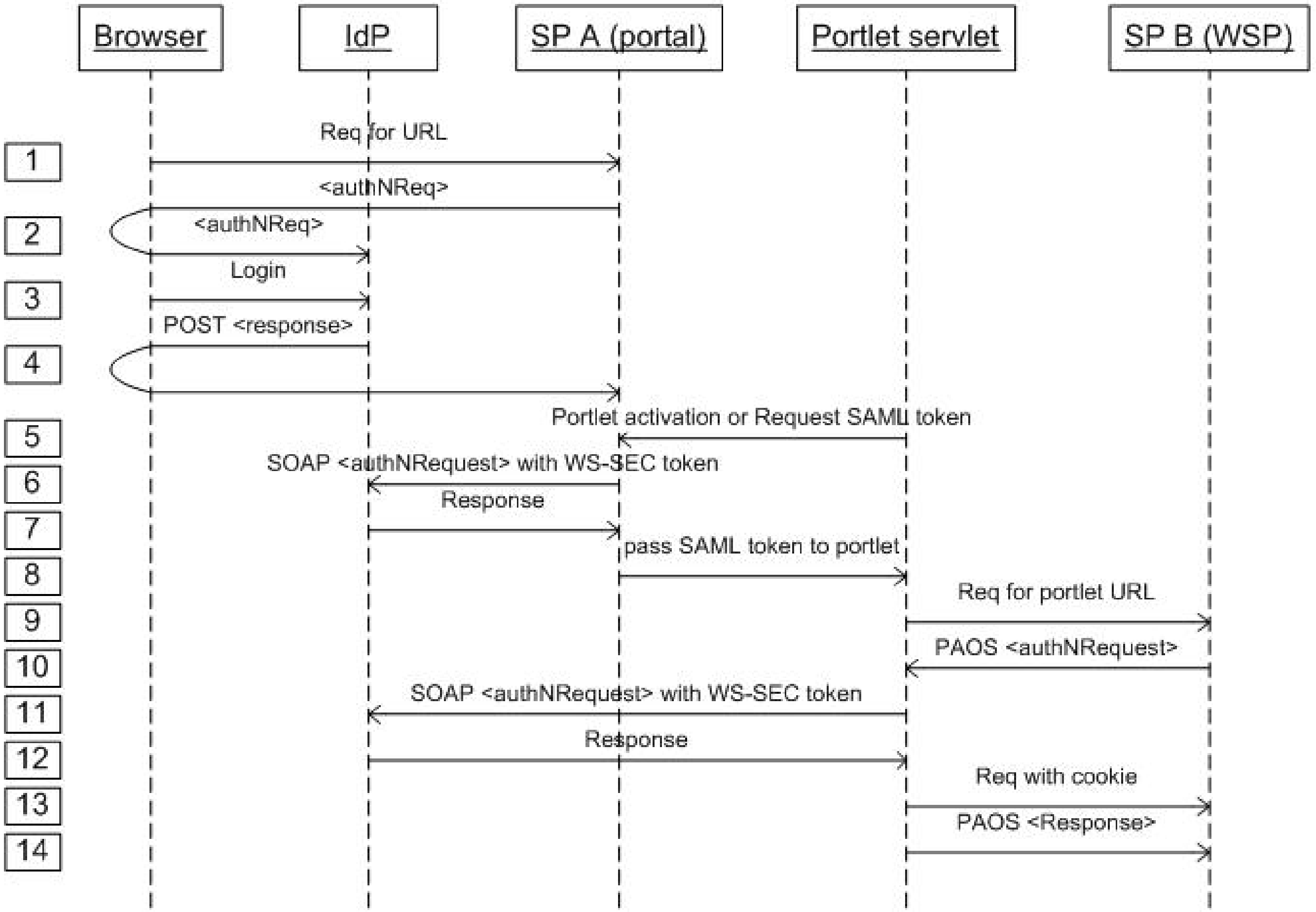
1. The User Agent logs into the Portal Instance with the mediation of the Identity Provider.
2. The User Agent accesses a Portlet (may simply be part of rendering the Portal to the User Agent).
3. The Portlet invokes the Web Service Provider in such a manner that the identities of both the User and the Portlet are securely communicated to the WSP.

An illustrative sequence diagram is further below, and a visio of the diagram is also available.

In proposing a solution, we hold to these assumptions and constraints:

- Access to and/or the behavior of the Portal/Portlet, as well as the WSP, depend on communicating one or more "attributes" about the user to them.
- Permission for the WSP to obtain the required user attributes is granted ahead of the User Agent's interaction with the Portal/Portlet.
- The user's unique identity may or may not be needed at both the Portal and WSP, and privacy is valued (i.e. the solution doesn't assume that providing a persistent identifier for the user is acceptable).
- The Portal, Portlet, and WSP may each be controlled by different organizations, making a federated solution a requirement.
- The Portal, Portlet, and WSP each have credentials known to the IdP, but need not be aware of each other's credentials.
- The protocol between the User Agent and Portal and between the Portlet and WSP is HTTP. We do not assume (but don't preclude) that it is SOAP over HTTP.
- The level of security of the User Agent's authentication and subsequent HTTP session with the Portal is also acceptable for the authentication and subsequent HTTP session between the Portlet and the WSP.
- The WSP need not (but may choose to) be aware of the distinction between a User Agent accessing it under direct control of a user, and a Portlet accessing it on behalf of a user.
- Within reason, requiring software changes to the Portlet to secure its interactions with the WSP is acceptable.
- Adoption of existing standards where possible is desirable.

# Portal/Portlet Proxies Browser-User



1-4. Browser-user authenticates to portal. SAML response includes list of entityIDs and URLs for portlets for which portal is authorized to delegate.

5-7. Portal follows ECP profile to obtain SAML token to be used by portlet. Occurs when portlet is “activated”, eg, when user switches to containing tab, or when an activated portlet requests a token.

8. Portlet API is used to pass SAML token to portlet, probably as a base64 encoded serialization of the token to ensure any signature remains valid.

9-14. Portlet follows ECP profile to obtain SAML token to access Web Service Provider (WSP) as browser-user.