

COIN Architecture - Adobe Connect

In an earlier SURFnet project, connecting Adobe Connect (AC) to the SURFederation has been researched and a proof-of-concept was performed. The document describing the installation and configuration can be found [here](#).

The AC connection to the SURFederation was performed in the context of the SURFgroepen platform. One of the properties of this platform is that users and groups in AC are already created before a user does an AC login. This provisioning is necessary as AC needs to have access to user and group data in its own database when a user performs a login. Within COIN, AC will be connected to the SURFederation outside of the context of SURFgroepen. This means that users and groups in the AC database need to be created when a user performs a login through the SURFederation. The preferred method for this is using the [AC Web Services API](#), which is described by Adobe as follows:

Acrobat Connect Pro Web Services provides an XML API, so your application must be able to communicate with Acrobat Connect Pro Server using XML over HTTP or XML over HTTPS. Your application calls the API by building a request URL and passing it one or more parameters, either as name/value pairs or as an XML document. Web Services returns an XML response, from which you can extract values.

A request to the XML API is formatted as an HTTP request URL that the API servlet handles. A request URL has an action name and parameters in name/value pairs, like this:

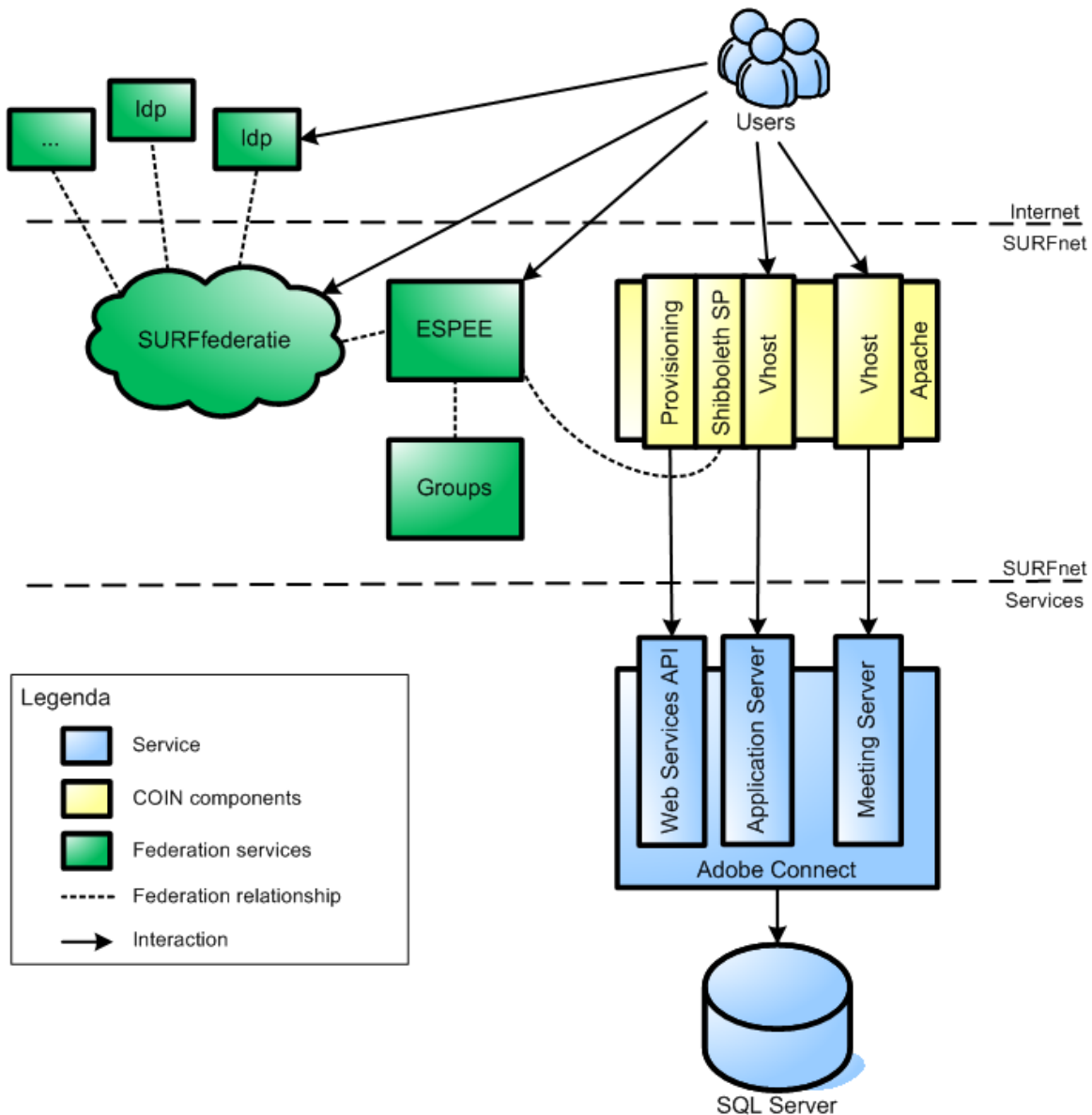
<https://example.com/api/xml?action=sco-info&sco-id=2006334909>

The call to the AC Web Services API that creates a user account (including its group relations) in the AC database, needs to be performed before the user arrives at AC. This call can be performed from a number of locations:

- From the SURFederation (ESPEE); when a user authenticates at the SURFederation for AC, the SURFederation could call the AC Web Services API and perform the user and group creation.
- From the Shibboleth SP installed inside the Apache Reverse Proxy; when a user performed a SURFederation authentication and returns to the Shibboleth SP, some piece of code (or script) could be called by a Shibboleth hook that performs the necessary calls to the AC Web Services API. This would mean extending the Shibboleth SP code and creating a AC specific Shibboleth SP distribution.
- From the Apache Reverse Proxy itself; whenever a user accesses the AC URL's through the Apache Reverse Proxy a script could be executed that performs the necessary AC Web Services API calls. As calling this API every time a user accesses AC would consume a lot of resources, this script could be optimized to keep track of the logged in users. As these users are already logged in and an AC user account has already been created, there would be no need to call the AC Web Services API.
- The Apache Reverse Proxy forwards all requests to a webpage that performs the AC Web Services API interactions. When done, this webpage redirects the user to the originally requested AC page.

The third option will be the one that will be implemented in COIN. Possibly, this method of performing user provisioning from a reverse proxy can be extended to other services besides AC.

The architecture for connecting AC to COIN is shown in the figure below:



In general, for implementing this scenario, the following steps need to be performed:

1. **Install Adobe Connect, Apache reverse proxy with Shibboleth and connect it to the SURFfederatie.** Manual is already present [here](#). Note that some issues were found with this setup that have been since been resolved with the help of other institutions. One of the most important issue was that the AC plugin did not function correctly. This can be resolved by following these instructions:

"We also used the SURFNETs "cookbook" for configuration apache. Surfnet complained that their ssl offloading of meetings didn't work with connect add in. The problem is solved when you use separate ssl certificates for each vhost, and each vhost has to have its own IP. This has to be done because ssl handshake is made before apache checks vhosts, and because flash can get only FQDN of a server out of certificate and not all the Cnames."

2. **Create provisioning script to create users in AC after SURFfederatie login.** This script should be executed from within Apache. To limit the performance impact from evaluating the existence of a user *every* time a user opens a link to AC, this script could implement some sort of caching (using memcache Apache module) that evaluates if the Shibboleth session already was used previously. If so, this means the user was already provisioned and no further action is needed from the provisioning script. If not, the user should be provisioned to AC.
3. **Modify provisioning script to include group relationships** (this is described here as a separate step as this depends on the addition of group attributes to the assertions coming from the ESPEE). The provisioning script will now also create groups (if the group does not already exist) and make the user a member of these groups

Hints and tips for developer:

- In AC, a group should be equal to a "kamer" (Chamber)
- Think about removing groups/kamers as the number of kamers can grow large. Groups assigned to a user will be created again when a

user does a new login

- Probably, only a groupid is enough for AC to be able to create a group
- Directive for calling script from Apache could be "Script" or "SetInputFilter"