AARC SA1.3 workplan

Michal Jankowski, Maciej Brzeźniak

The work to be done within SA1.3 task aims in improving access to research and education relevant, non-web resources located out of the home organization of the user. The main improvement is making use of existing AAI that provide user's credentials and authorization attributes instead of local user management. While this functionality have already several implementations and is widely used by web portals, the technologies for non-web scenarios is still in their early stage.

A number of pilots is going to be setup in order to investigate these technologies. The selection of software to be piloted is going to be discussed with JRA1 in order to focus on tools that would fit to the proposed by that activity models (JRA1.3 and JRA1.4) and blueprint architecture (JRA1.4). Also the requirements gathered by JRA1.1. will be used as a material for assessment of technologies used for the pilots. Finally, the experience gathered while running the pilots and the performed analyzes will be used as an input material for the final shaping of the blueprint architectures and best practices recommendations (NA3).

Compatibility between the technologies piloted within this task and technologies used for collecting attributes within task SA1.2 will be checked. Attribute requirements for non-web SSO, authorization and provisioning will be investigated and defined.

Usage of user credentials and attributes coming from different AAIs, including guest IdPs proposed by SA1.1 will be analyzed.

The following subsections describe pilots, that are the first choice and will test technologies, that seem to be matured enough and most promising at the moment.

LDAP Facade pilot

LDAP Facade description

LDAP Facade[1] is a single software component, that needs to be installed at SP. It makes use of the local accounts prepared during the registration step. The software is able to replace traditional LDAP server, as it provides the same interface. As the result, LDAP facade can be used as a local user manager as well as authentication and authorization component without any modification to servers (not only core code is unchanged, even implementing specialized plugin is not necessary). On the other side (collaboration with external IdP), the deployment is the same as for any other SAML-based SP.

[1] J. Köhler, M. Simon, M. Nussbaumer und H. Hartenstein: "Federating HPC access via SAML: Towards a plug-and-play solution" in *International Supercomputing Conference*, Leipzig, Juni 2013

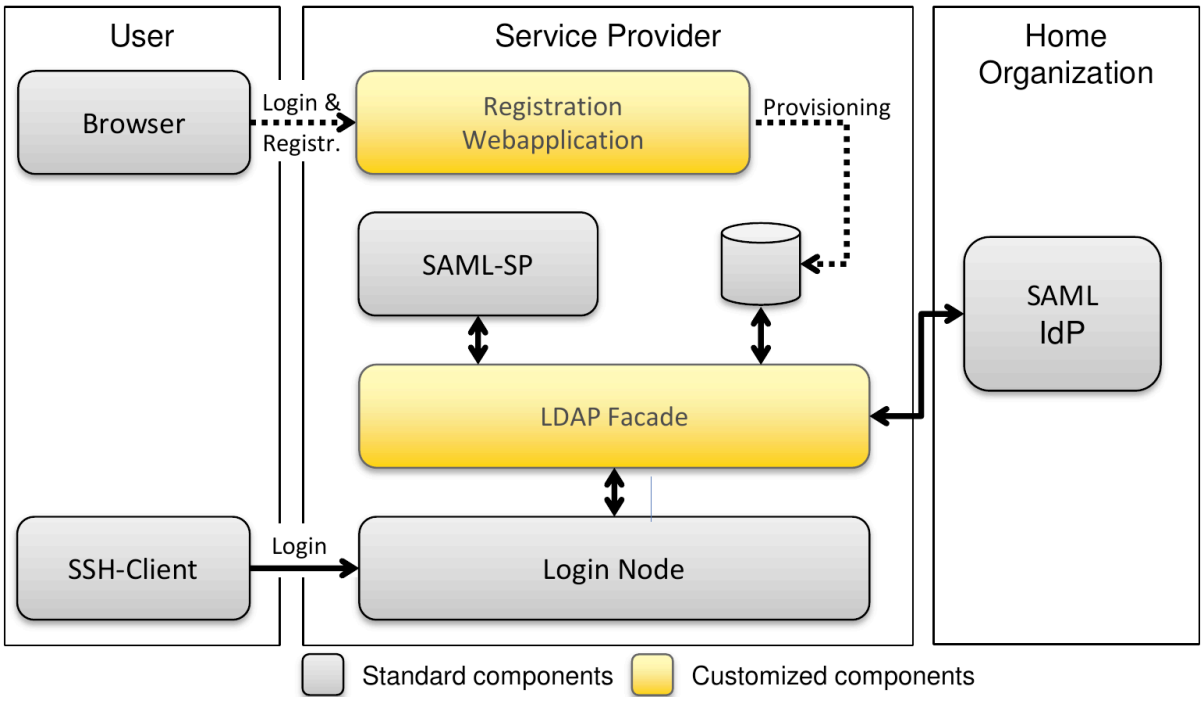

Figure 1. Architecture of LDAP Facade.

Source: S.Labitzke *Now SAML takes it all: Federation of non Web-based Services in the State of Baden-Württemberg,* European Identity & Cloud Conference 2013

Workflow

- The solution requires prior resource provisioning/registration which is done via web portal and **SAML WebSSO profile**. The local account is set up in this step.
- Access to the resource is possible in two models:
    - **Full trust** (the SP is able to observe user's password) and **unmodified client**: the client software is not modified, the user sends (IdP) password to SP, that authenticates the user and retrieves assertions from the IdP using **SAML ECP profile**.
    - **Limited trust** (the SP is NOT able to observe user's password) and **modified client**: the assertion signed by IdP is acquired on client side (using **SAML ECP profile)** and sent to SP as user's password. In this mode, the user have to run a program, that acquire the assertion prior running the client and pass to it the cooked up password. The acquisition and access may be also performed by a single but non-standard client.

Scope and goals

The pilot aims in providing access to non-web resources (e.g. sftp, ssh console) to non-grid users using existing AAIs, without need to obtain user certificates. The resources are considered stateful not only within single session, but also between sessions. The good example of stateful resource is data, that stored within one user's login session shall be also available when the user logs in next time. Thus, when the user is identified by some external identity provider, his (global) identity (and attributes) must be mapped to stable, local resources. This model requires an initial registration/provisioning step, which practically means setting up a local account.

The users shall have possibility to use the standard client software they used so far.

Another requirement is that the user need to be logged to the same account using different access and authentication methods (e.g. sftp/password, gridFTP/X.509).
The above requirements shall be fulfilled by LDAP Facade, which allows federated authentication by simulating standard LDAP authentication.
The goals of the pilot are:

- analyze additional effort of the users to access the resources (comparing to traditional access pattern)
- analyze administrative effort
- analyze security aspects of the solution
- propose possible enhancements investigate how the solution may be used for guest IdP
- investigate how the solution may be enhanced to work with many attribute authorities, different than the home IdP

Work to be done

1. Setup the registration portal.
2. Setup LDAP Facade and configure it with an existing, trusted IdP.
3. Test the solution.
4. Configure LDAP Facade with guest IdP and experiment with it. Use IdP suggested by JRA1.3 and SA1.1.
5. Investigate work with many attribute authorities, different than the home IdP. Consider AA suggested by JRA1.4 and SA1.2.

CILogon pilot

CILogon description

CILogon[2] software provides functionality to run X.509 certification authority (CA) relying on federated user authentication. In other words, CILogon allows users for obtaining certificates basing on their existing digital identities rather than manual authentication. This approach eliminates need for maintaining network of registration authorities (RA) and simplifies procedure of obtaining certificates for the users. This is especially valuable in large communities.
The CILogon system consists of a Web front-end, a user database and CA back-end. The front-end performs authentication (SAML, OpenID) and contains user interface for issuing certificates. It provides also OAuth interface for integration with external Web applications. The user database is used to manage sessions and user identities. The CA back-end issues the certificates and manages certificate revocation lists (CRL).
The CILogon provides several interfaces to retrieve certificates:

- GridShib Java Web Start -the keys are generated on client side, but requires maintaining compatible JVM on use side and performance might be an issue
- PKCS12/HTTPS -the credentials are generated on server side (including keys) and then available via randomized, short-lived HTTPS link
- ECP -allows users to authenticate using SAML ECP profile and download the certificate
- OAuth -enables delegating certificate to Web applications in order to act on user's behalf.

[2] J.Basney, T.Fleury, J.Gaynor, *CILogon: A federated X.509 certification authority for cyberinfrastructure logon*, Concurrency and Computation: Practice and Experience 26(13): 2225-2239 (2014).

The level of assurance (LOA) connected to a certificate issued by CILogon may depend on IdP, as IdP plays role of RA in the traditional PKI and different IdPs have different policies (e.g. Google account and account authenticated by eduGAIN). In order to comply with that, CILogon may operate a number of CAs, each one connected to certain LOA and having own policy.

Scope and goals

Work to be done

Unity pilot

Unity description

Unity claims to be a complete solution for identity, federation and inter-federation management[3]. The design follows cloud approach -Identity Management As a Service.
Access to Unity is available via different endpoints. Each endpoint has its binding low level access protocol which can be for instance *web* (designated for web-browsers) or *SOAP* (for web service clients). Each endpoint is associated with authenticator(s). Authenticator is a pair of credential retrieval and verificator. The retrieval collects the credential in a binding specific way. The verificator checks if it is correct.
Unity also maintain a database of local user identities. Each user must be registered and obtain local identity prior using the service, attributes of external user are imported.
Unity uses modular design approach, thus variety of endpoints may be combined with variety of authenticators. There are several implemented endpoints (Web admin UI, Web user profile UI, SAML 2 Web, SAML 2 WS, OpenId, OAuth1). There is also a number of implemented authenticators, both using local authentication (passwords, certificates, one time passwords, challenge-response, etc.) and external ones, that delegate authentication to an external service (LDAP/AD, SAML IdP, OpenId, OAuth, etc.).

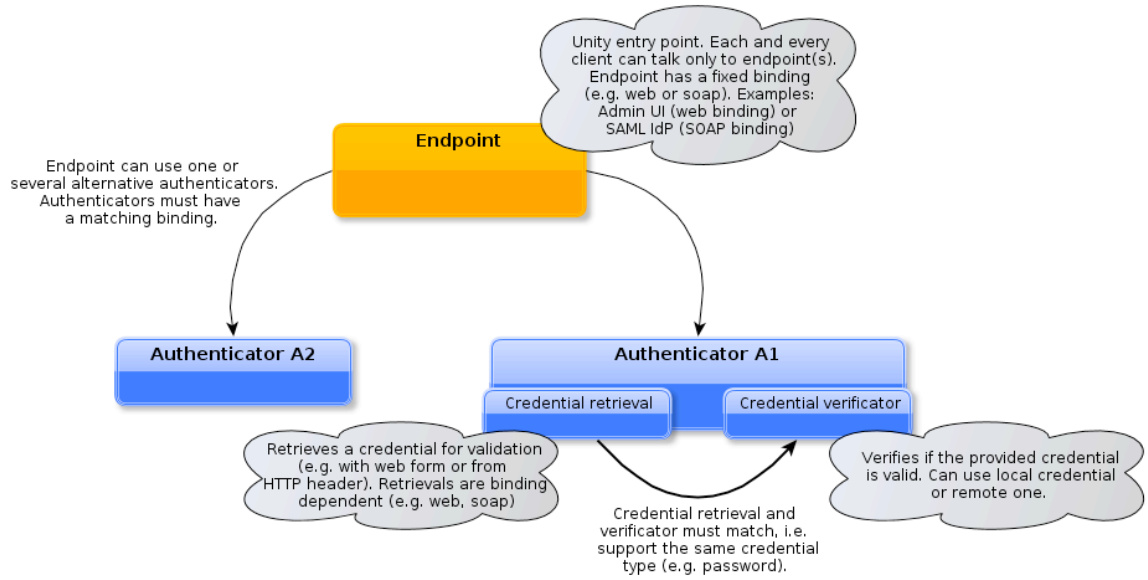

Figure 3. Endpoint-Authenticators Workflow in Unity.
Source: *Unity Manual*, v. 1.6.0, http://www.unity-idm.eu/documentation/unity-1.6.0/manual.html

[3] Unity Manual, http://www.unity-idm.eu/documentation/unity-1.6.0/manual.html.