



*Deliverable Number: D33.3*  
*Factsheet Number: #1*  
*Software prototype Name: "Federated SSO  
Utility Service – Federated SSO Server"*

Document Owner:	Maria Carla Mantuano (ENG)
Contributors:	Mauro Isaja (ENG), Antonio Scatoloni (ENG)
Dissemination:	Public
Contributing to:	WP33
Date:	29/03/2013
Revision:	1.0

## List of Abbreviations

<b>SSO</b>	Single Sign On
<b>CAS</b>	Central Authentication Service
<b>LSA</b>	Local Security Authority

## Table of Contents

<b>1</b>	<b>AVAILABILITY AND CONTACTS .....</b>	<b>4</b>
<b>2</b>	<b>ARCHITECTURE AND FUNCTIONALITIES .....</b>	<b>5</b>
2.1	ARCHITECTURE.....	5
2.2	SECURITY.....	6
2.3	ATTRIBUTES AND ROLES .....	7
<b>3</b>	<b>TECHNICAL INFORMATION.....</b>	<b>8</b>
3.1	TECHNICAL DETAILS .....	10
<b>4</b>	<b>LICENSING .....</b>	<b>11</b>
4.1	SERVICE LICENSE .....	11
4.2	THIRD PARTY LICENSES .....	11
<b>5</b>	<b>TECHNICAL MANUAL .....</b>	<b>12</b>
5.1	FEDERATED SSO SERVER .....	12
5.2	SSO CLIENT.....	12
5.3	SOURCE CODE .....	12
<b>6</b>	<b>USER MANUAL.....</b>	<b>13</b>
6.1	ACCESS .....	13
6.2	USER MANUAL .....	13
<b>7</b>	<b>FUTURE PLANS.....</b>	<b>14</b>
<b>8</b>	<b>REFERENCES.....</b>	<b>15</b>

## 1 Availability and Contacts

---

This table describes how to reach the prototype and the contact person.

<b>Version</b>	1.0
<b>Availability</b>	<a href="https://msee.eng.it/sso/login">https://msee.eng.it/sso/login</a>
<b>Accompanying specification and design document</b>	D33.3 FI Utility Services first prototype – M18
<b>Source control</b>	svn://repo.nimbus-ware.comMSEE/SP3/WP33/D33.3/trunk/MSEE_WP33_SsoSystem
<b>Contact person</b>	Refer to owners/contributors of this document

## 2 Architecture and Functionalities

Federated Single Sign On (SSO) Utility Service is a SSO System in the MSEE context, i.e. a central authentication point for the MSEE Federation of Ecosystems/Enterprises.

Federated SSO Utility Service allows to centralize the authentication mechanism of all the applications accessible in the Federation of Ecosystems/Enterprises; the users can access each application without repeat login every time.

The system is based on the Jasig's Central Authentication Service (CAS) open source SSO system.

CAS allows a user to access multiple applications providing credentials only once per browser session; in this way, central authentication avoids credential proliferation, and reduces the exposure of the user's credentials to applications.

Customizations to standard CAS, as we shall see below, are provided to enable the federation of multiple networked identity providers, and to implement the MSEE security model.

In the MSEE Federation context may exist more than one Local Security Authority (LSA), node of the Federation *responsible* for the authentication of a subset of users and *trusted* by all the other nodes.

Taking advantage of CAS extensibility, we provide a custom Authentication Handler component that, resolving the authority name from the user's credentials, is able to identify the relevant LSA and delegate authentication to the identified LSA itself.

The Architectural components of the Federated SSO Utility Service prototype will be presented in this chapter.

### 2.1 Architecture

The Federated SSO Utility Service solution is based on these components:

- **Federated SSO Server**
- **SSO Client**
- **LSA Service**

**Federated SSO Server** is a customized version of the Jasig's Central Authentication Service (CAS) open source product. For each MSEE Federation there is one single deployment of this server – some place of the network where it is visible to each client and has visibility of each LSA Server.

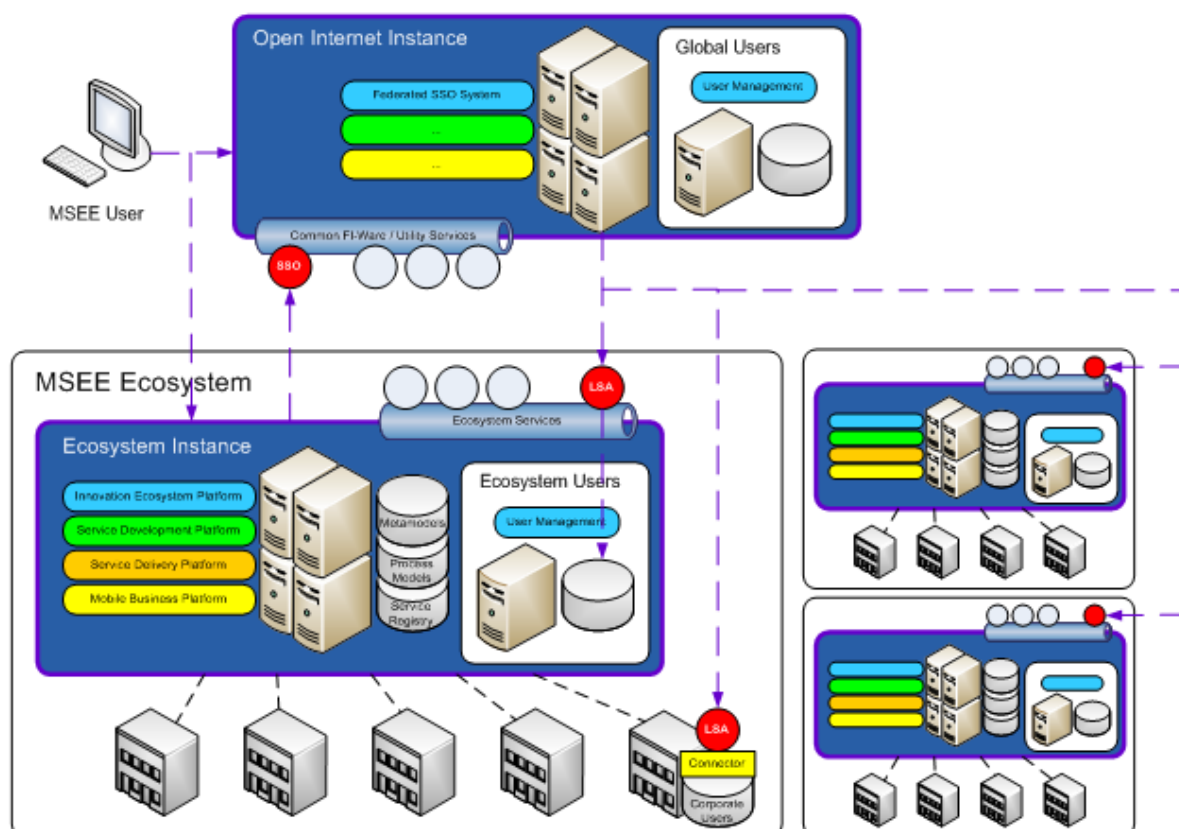
The customization consists of a new Authentication Handler component to resolve authority name and delegate user authentication.

**SSO Client** consists in the client modules that are part of the official CAS product. These modules are technology-specific, as a different distribution exists for each supported environment – that is, Java, PHP, .NET, and many others. As a rule, they are drop-in, configurable components capable of adding SSO support to any compatible web application. For further information, see [2].

**LSA Service** is a web service responsible for user credential validation and attribute release, deployable anywhere on the network provided its endpoint is visible to the Federated SSO Server.

The Local Security Authority Service is described in [3].

**Figure 1** shows relationships between Federated SSO Utility Service components and the other actors of an MSEE Ecosystem Federation.



**Figure 1 – Federated SSO Architecture**

## 2.2 Security

HTTP communications are secured by the use of SSL<sup>1</sup> sockets.

The relation of mutual trust between the centralized Federated SSO server and each LSA Service is protected by the use of secure communication channels and by digital certificates as a guarantee of the actual identity of each actor.

<sup>1</sup> Secure Sockets Layer is a cryptographic protocol that enable a safe communication *end-to-end* on TCP/IP network, providing authentication and data integrity

## 2.3 *Attributes and roles*

In the MSEE context, Federated SSO Utility Service does not involve only authentication, but also attributes release, i.e. the capability of assigning a set of attributes to an user as the result of a successful login.

The MSEE security model defines a set of user attributes and roles that applications in the MSEE Federation of Ecosystems need for their internal use; particularly, an application may use the roles to decide if a user is authorized to access resources and services.

MSEE security model defines the set of mandatory attributes that are relevant to every application, and must be provided by any LSA:

- ID of the user (login name)
- Full name of the user
- Name of the organization(s) the user belongs to

and a minimal set of optional attributes:

- Email address of the user
- List of roles of the user.

MSEE security model define also a list of user roles:

- **MSEE\_Administrator** – a super user enabled to system operations forbidden for users of other levels
- **MSEE\_Business\_Expert** – an user working in business field
- **MSEE\_IT\_Expert** – a technical user with tasks of analysis and development
- **MSEE\_Employee** - a base user who belongs to the Organization and doesn't have one of the above described authorization levels.

This list is work-in-progress; it will be possible to add new roles in the future if they seem fit.

Local roles and / or extended attributes may also be defined by each LSA, for internal use of some specific application; they will be ignored by unaware ones.

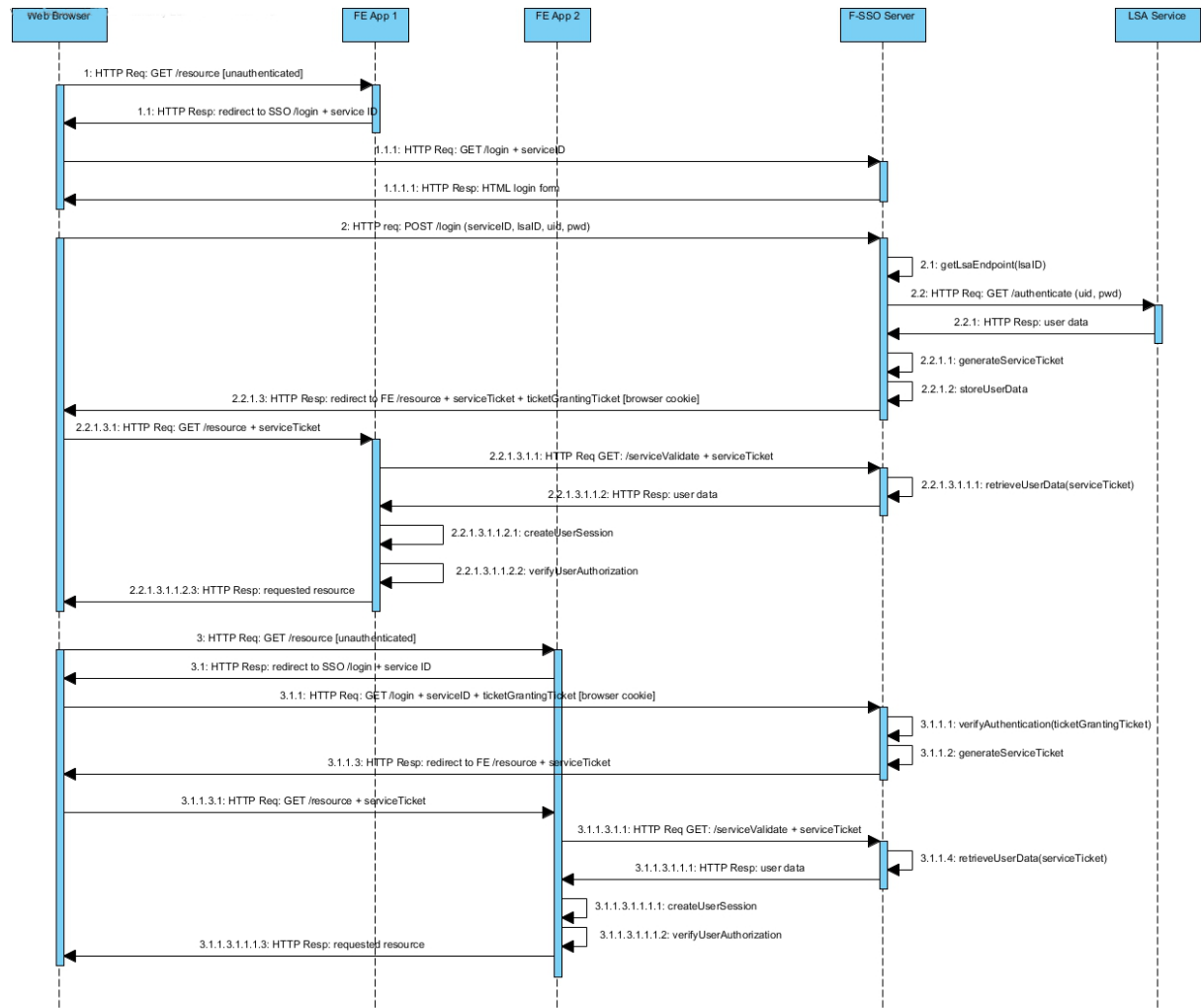
To summarize, this is the set of user attributes which any LSA must (mandatory attributes) or may (optional attributes) provide is:

Name	Description	Mandatory	Multiple Values
MSEE_ID	Login Name of the user	Yes	No
MSEE_Name	Name and last name of the user	Yes	No
MSEE_Organization	Organization Name of the user	Yes	Yes
MSEE_Email	Email address of the user	No	No
Role	List of roles of the user	No	Yes
.....	Further optional attribute	No	

### 3 Technical Information

**Federated SSO Server** customizes CAS solution by adding a new Authentication Handler component that identifies the authority which is actually responsible for the validation of credentials, and delegates user authentication to this authority.

**Figure 2** represents the message flow during authentication.



**Figure 2 – Federated SSO message flow**

When the user requires a web resource, the target Web Application handles the request. It determines that, by its own policies, authentication / authorization is required in order to access the requested resource, and that the incoming request does not belong to an authenticated user session: it then throws an exception.

The SSO Client, which is integrated within the Web application, intercepts the exception and redirects the user to the Federated SSO Server's login form. The redirection URL includes the "Service ID" parameter, which uniquely identifies the application requiring user authentication.



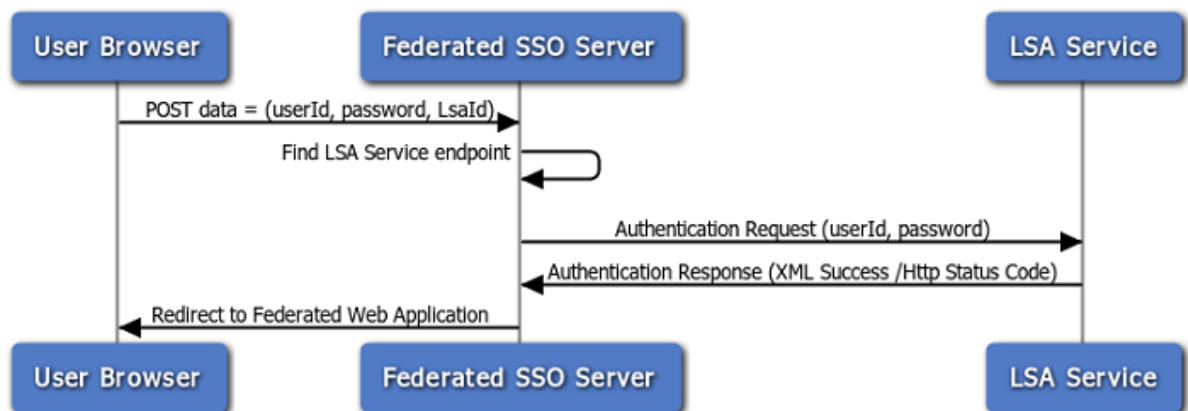
The user loads the login form into her browser, enters her credentials and submits the form to the Federated SSO Server. Credentials are in the form of a userid – password pair, where the userid element may optionally be composed as a two-part identifier <user@authority>. The Federated SSO Server identifies the authority which is actually responsible for the validation of credentials. This is done by matching the “authority” part of the userid with a list of names of registered LSA Services. If no authority name was explicitly provided by the user, it is assumed that authentication should be handled centrally – that is, the user is registered with a default authority at the Global level. When a match is found, the corresponding LSA Service endpoint is extracted from configuration.

The Federated SSO Server invokes the LSA Service endpoint, passing as arguments the user’s credentials. The LSA Service checks the credentials against its own private repository of user data, then replies with a message stating the “OK” status and declaring the name and value of all the attributes attached to the positively identified user. Attributes are those defined by MSEE security model (i. e. list of MSEE roles, organization, full name, etc.) plus any other which may be optionally defined by the local authority for “internal” use.

The Federated SSO Server initializes a local session for the newly authenticated user. The session is linked to a copy of user attributes released by the LSA Service.

The Federated SSO Server creates a “Service Ticket” for the Web Application: a unique authentication token bound to the local user session and to the service identified by “Service ID”. It also creates a “Ticket Granting Ticket” for the user: a unique authentication token bound to the local user session. Finally it redirects the user back to the Web Application, providing both the Service Ticket (as a URL parameter) and the Ticket Granting Ticket (as a browser cookie).

**Figure 3** focuses on the interaction between User Browser, Federated SSO Server e LSA Service:



**Figure 3 - Authentication Sequence**

The target Web Application handles the URL. Due to the presence of the Service Ticket parameter, the SSO Client module kicks in and intercepts the request before it is served by the application layer. Transparently to the user, the SSO Client calls a service on the Federated SSO Server to validate the Service Ticket.

The Federated SSO Server verifies that the Service Ticket is a valid token. It then recovers the local user session and builds a response which provides the userid and the full attribute set

of the user. Before returning the response message to the caller, the Service Ticket is invalidated (reuse of the same token is not allowed).

The SSO Client uses the Federated SSO Server response to properly initialize the user session on behalf of the Web Application.

Finally, it yields the request so that the application layer may serve it.

The target Web Application handles the request. As before, it determines that authentication / authorization is required in order to access the requested resource; this time the incoming request belongs to an authenticated user session, so it checks if the user has one of the roles required for access.

Now the user can interact with the Web Application; until the user session remains valid, neither the Federated SSO Server nor the SSO Client are further involved.

When the user tries to access a different Web Application, she is redirected to the Federated SSO Server's login form and it comes holding the Ticket Granting Ticket (which, as previously stated, is stored in a browser cookie issued by the Federated SSO Server). The Federated SSO Server verifies that this token is (still) valid, generates a new Service Ticket for the new Web Application and redirects back the user without showing any login page.

### 3.1 Technical details

<b>Nature</b>	Web Application
<b>Programming Language</b>	Java
<b>Development Framework</b>	Java SE Development Kit 1.6 or later
<b>Additional libraries</b>	OpenSAML 1.1b OpenSAML 2.5 JBoss RESTEasy 2.3.5 or later
<b>Application Server</b>	J2EE servlet container (tested on Apache Tomcat 7)
<b>Database</b>	n.a.

## 4 Licensing

---

### 4.1 Service license

All software developed is open-source and under the Apache License, Version 2.0.

### 4.2 Third party licenses

Single Sign On is based on Jasig's CAS (Central Authentication Service) authentication system, OpenSAML for Security Assertion Markup Language and JBoss RESTEasy. All third party software is open source.

Third party software	Licence
Jasig's CAS (Central Authentication Service)	Vedi <a href="http://www.jasig.org/cas/license">http://www.jasig.org/cas/license</a>
OpenSAML	Apache License, Version 2.0.
JBoss RESTEasy	Apache License, Version 2.0.

---

## 5 Technical Manual

The prerequisites for installing the **Federated SSO Utility Service** components are:

- A J2EE servlet container version 6 or later (the prototype has been tested on Apache Tomcat 7)
- The URL of at least one LSA Service endpoint, and the corresponding authority name, to identify and configure the related Authentication Handler. In login operation, the authority name is the finale part of the user identifier <user@authority>, identifying the Local Security Authority (LSA) of the user.

### 5.1 Federated SSO Server

The Federated SSO Utility Service package can be downloaded from the MSEE portal: <http://www.msee-ip.eu/intranet/sp3-workspace/sp3-wp33/d3.3.3-fi-utility-services-first-prototype-m18>

The package *D33.3 FI Utility Services first prototype - M18 - FederatedSSO.bin.zip* contains the file *sso.war* that must be deployed on the application server.

Edit the file *deployerConfigContext.xml* configure one or more specific *Authentication Handlers* that must be invoked for authentication.

Add the following bean definitions:

- In the list of *property name="credentialsToPrincipalResolvers"*  
*bean class* -> name of the specific *Authentication Handler class*  
 > *property name="URI"*: set the URL of the LSA Service endpoint
- In the map of *property name="authenticationHandlers"*  
*entry key* -> name of the specific authority  
 -> *bean class* = name of the specific *Authentication Handler class*  
 -> *property name="URI"*: set the URL of the LSA Service endpoint.

This operation must be repeated for each LSA.

### 5.2 SSO Client

The standard version of the CAS Client can be downloaded at the address: <http://downloads.jasig.org/cas-clients/cas-client-3.1-release.zip>.

It has to be configured according to Jasig's instructions by replacing the URL of the CAS Server with <https://bivolino-msee.eng.it/sso/>.

### 5.3 Source code

Module	Component	URL
Federated SSO Server	Customized version of the CAS Server	<a href="svn://repo.nimbus-ware.com/MSEE/SP3/WP33/D33.3/trunk/MSEE_WP33_SsoSystem">svn://repo.nimbus-ware.com/MSEE/SP3/WP33/D33.3/trunk/MSEE_WP33_SsoSystem</a>
SSO Client	Standard version of the CAS Client	<a href="http://downloads.jasig.org/cas-clients/cas-client-3.1-release.zip">http://downloads.jasig.org/cas-clients/cas-client-3.1-release.zip</a>

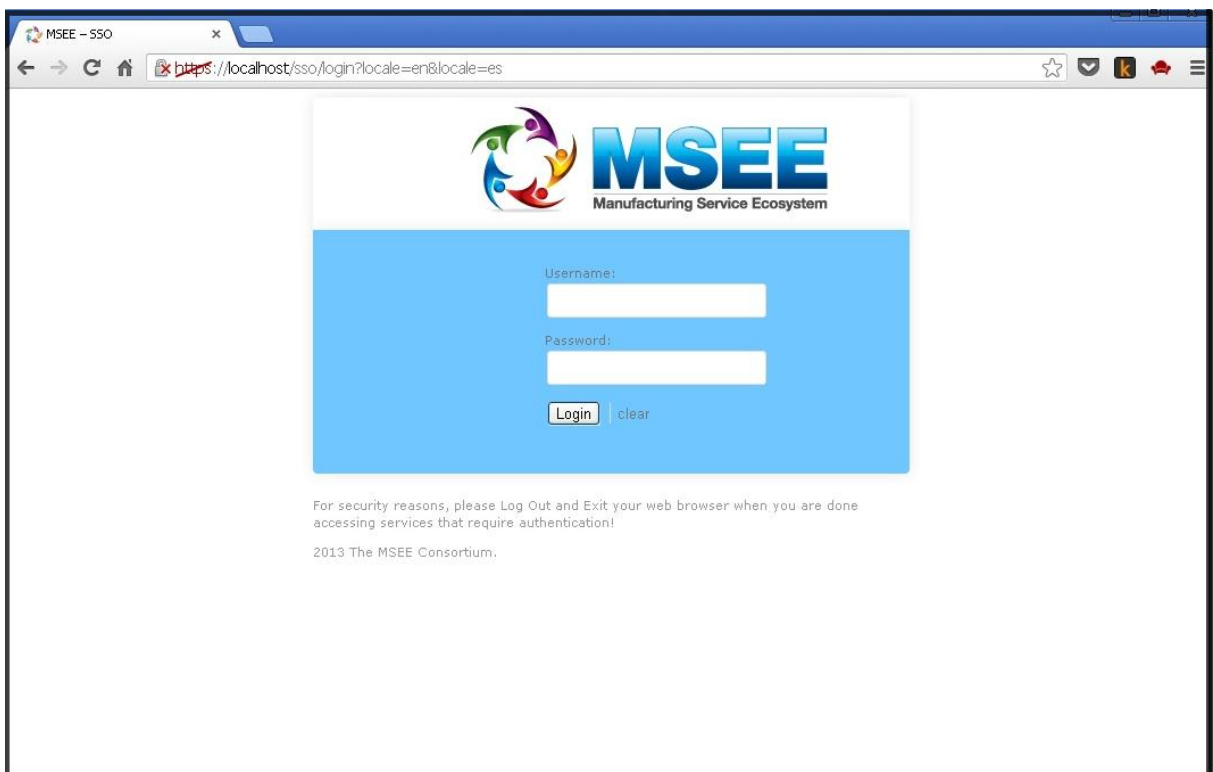
## 6 User Manual

### 6.1 Access

Federated SSO Utility Service is a subsidiary functionality of MSEE applications; it doesn't have its own entry-point. At the moment (M18) it is integrated only in the Consumer Marketplace application (see [5]), in the future it will be in every front-end application that needs central authentication in the MSEE context.

### 6.2 User Manual

The user requires a web resource by loading a URL into her web browser. The target Web Application determines that authentication / authorization is required in order to access the requested resource, and that the request does not belong to an authenticated user session. The user is then redirected to the login form (Figure 4).



**Figure 4 - Federated SSO login form**

To login, the user needs to have previously received her credentials and authority name from her LSA administrator.

The user enters credentials and submits the form. Credentials are in the form of a userid – password pair, where the userid element may optionally be composed as a two-part identifier <user@authority>, authority identifying the Local Security Authority (LSA) of the user.

If the user enters invalid credentials, the system returns the HTTP status code 404 (Not Found).

If the credentials entered by the user are valid, but the user is disabled, the system returns the HTTP status code 403 (Forbidden).

If user credentials are valid, the user is redirected to the Web Application; she can interact with the required Web Application and with other MSEE applications without further login, until the user session remains valid.

## 7 Future Plans

---

In future developments of MSEE Project, the Federated SSO Utility Service prototype delivered at Month 18 may be refined and extended.

## 8 References

---

- [1] <http://www.iasig.org/cas/cas2-architecture>
- [2] <http://www.iasig.org/cas/client-integration>
- [3] D33.3 Factsheet#2 – Federated SSO Utility Service - LSA Service
- [4] D3.3.3 FI Utility Services first prototype – M18
- [5] D3.2.2 FI Platform Federation first prototype – M18