

# EGEE-II

## SHIBBOLETH INTEROPERABILITY THROUGH A SHORT-LIVED CREDENTIAL SERVICE (SLCS)

M J R A 1 . 4

---

Document identifier: EGEE-II-MJRA1.4-770102-v0.96.doc

Date: **10/11/2006**

Activity: **JRA1:Middleware**

Document status: **DRAFT**

Document link: <https://edms.cern.ch/document/770102/1>

Abstract:

This document presents the motivation for and the software structure of a short-lived credential service that has been implemented as the first phase of the work item “Interoperability Shibboleth gLite” within the JRA1 activity. Different deployment options and operational modes of this service are also described.

Copyright notice:

Copyright © Members of the EGEE-II Collaboration, 2006.

See [www.eu-egee.org](http://www.eu-egee.org) for details on the copyright holders.

EGEE-II (“Enabling Grids for E-scienceE-II”) is a project co-funded by the European Commission as an Integrated Infrastructure Initiative within the 6th Framework Programme. EGEE-II began in April 2006 and will run for 2 years.

For more information on EGEE-II, its partners and contributors please see [www.eu-egee.org](http://www.eu-egee.org)

You are permitted to copy and distribute, for non-profit purposes, verbatim copies of this document containing this copyright notice. This includes the right to copy this document in whole or in part, but without modification, into other documents if you attach the following reference to the copied elements: “Copyright © Members of the EGEE-II Collaboration 2006. See [www.eu-egee.org](http://www.eu-egee.org) for details”.

Using this document in a way and/or for purposes not foreseen in the paragraph above, requires the prior written permission of the copyright holders.

The information contained in this document represents the views of the copyright holders as of the date such views are published.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED BY THE COPYRIGHT HOLDERS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE MEMBERS OF THE EGEE-II COLLABORATION, INCLUDING THE COPYRIGHT HOLDERS, OR THE EUROPEAN COMMISSION BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE INFORMATION CONTAINED IN THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Trademarks: EGEE and gLite are registered trademarks held by CERN on behalf of the EGEE collaboration. All rights reserved"

**Document Log**

Issue	Date	Comment	Author/Partner
0.1	19/10/2006	TOC	C.Witzig/SWITCH
0.9	23/10/2006	Integrated comments from SWITCH/Middleware	P.Flury, V.Tschopp, T.Lenggenhager
0.91	24/10/2006	Added comments	J.White, C.Grandi/CERN
0.95	9/11/2006	Integrated comments by reviewer D.Spence	C.Witzig
0.96	10/11/2006	Modifications according to comments by reviewer	C.Witzig

**Document Change Record**

Issue	Item	Reason for Change
0.1		

---

## TABLE OF CONTENTS

<b>1. INTRODUCTION .....</b>	<b>4</b>
1.1. PURPOSE.....	4
1.2. DOCUMENT ORGANISATION .....	4
1.3. APPLICATION AREA .....	4
1.4. REFERENCES .....	4
1.5. DOCUMENT AMENDMENT PROCEDURE .....	4
1.6. TERMINOLOGY .....	5
<b>2. MOTIVATION .....</b>	<b>7</b>
2.1. MOTIVATION FOR INTEROPERABILITY BETWEEN SHIBBOLETH AND GRID INFRASTRUCTURES .....	7
2.2. MOTIVATION FOR INTEROPERABILITY THROUGH A SHORT-LIVED CREDENTIAL SERVICE (SLCS) .....	7
<b>3. IMPLEMENTATION OF THE SLCS.....</b>	<b>8</b>
3.1. USER'S VIEW OF THE SLCS.....	8
3.2. SHIBBOLETH ADMINISTRATOR'S VIEW OF THE SLCS.....	9
3.3. SOFTWARE DESIGN .....	10
3.4. FUNCTIONAL DESCRIPTION.....	10
3.5. SOFTWARE IMPLEMENTATION.....	11
3.6. OPERATION IN A TEST-BED .....	13
<b>4. DEPLOYMENT ISSUES .....</b>	<b>13</b>
4.1. UI CLIENT COMPONENT .....	13
4.2. SLCS SERVER DEPLOYMENT .....	13
4.2.1. <i>Compliance of the SLCS with the Shibboleth Federation Policy</i> .....	14
4.2.2. <i>Accreditation with the IGTF</i> .....	14
4.2.3. <i>Operational Modes</i> .....	15
<b>5. SUMMARY AND OUTLOOK.....</b>	<b>16</b>

## 1. INTRODUCTION

### 1.1. PURPOSE

The purpose of this document is to describe the first phase of work aimed at providing interoperability between Shibboleth and gLite. This work has been done by SWITCH as part of the EGEE-II project. The goal of this phase is to achieve interoperability by issuing short-lived X.509 certificates to grid users, based on their authentication to a Shibboleth Identity Provider. The audience of this document are members of the EGEE-II JRA1 activity and interested grid users as well as Shibboleth and grid system administrators.

### 1.2. DOCUMENT ORGANISATION

Section 2 describes the motivation for achieving interoperability between Shibboleth and gLite in general and explains why the development of this short-lived credential service (SLCS) was chosen as the first of several phases. In section 3 we describe the software structure of the SLCS and in section 4 various deployment options, including the one chosen by SWITCH. In section 5 we look forward to the use of the SWITCH SLCS as a Shibboleth federation wide service.

### 1.3. APPLICATION AREA

This document applies to the implementation of the interoperability Shibboleth - gLite as achieved during the development of a short-lived credential service within the SWITCH Shibboleth federation.

### 1.4. REFERENCES

**Table 1: Table of references**

R 1	More information can be found on the Shibboleth project web page <a href="http://shibboleth.internet2.edu">http://shibboleth.internet2.edu</a>
R 2	A short overview of Shibboleth including an online demo can be found at <a href="http://www.switch.ch/aai/about">http://www.switch.ch/aai/about</a>
R 3	SLCS user manual: <a href="https://edms.cern.ch/document/788604/1">https://edms.cern.ch/document/788604/1</a>
R 4	SLCS administrator manual: to be submitted to EDMS
R 5	SLCS Profile: <a href="http://www.eugridpma.org/guidelines">http://www.eugridpma.org/guidelines</a>
R 6	Shibboleth architecture protocols: <a href="http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-arch-protocols-latest.pdf">http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-arch-protocols-latest.pdf</a>
R 7	gLite CVS repository, module org.glite.slcs. See <a href="https://edms.cern.ch/document/468700/0.7">https://edms.cern.ch/document/468700/0.7</a> for information how to access the source code.

### 1.5. DOCUMENT AMENDMENT PROCEDURE

Amendments, comments and suggestions should be sent to the authors. The procedures documented in the EGEE “Document Management Procedure” will be followed: <http://egee-jra2.web.cern.ch/EGEE-JRA2/Procedures/DocManagmtProcedure/DocMngmt.htm>.

## 1.6. TERMINOLOGY

A complete project glossary is provided in the EGEE glossary <http://egee-jra2.web.cern.ch/EGEE-JRA2/Glossary/Glossary.html>.

### Glossary

AAI	Authentication and Authorization Infrastructure
CA	Certificate Authority: An internal entity or trusted third party that issues, signs, revokes and manages digital certificates.
Certificate	Information issued by a trusted party. Used to identify an individual or system.
CMC	Specification for certificate management messages using CMS (Cryptographic message syntax). It is described in RFC 2797 ( <a href="http://www.ietf.org/rfc/rfc2797.txt">http://www.ietf.org/rfc/rfc2797.txt</a> )
CP/CPS	Certificate Policy / Certification Practice Statement: see CP and CPS
CP	Certificate Policy: Rules that a request must comply with for the RA to approve the request or a CA to issue a certificate
CPS	Certification Practise Statement: Document that regulates rights and responsibilities of all the parties involved (RA, CA, End Entity, Relying Party) within a PKI infrastructure.
Credentials	Evidence asserting the user's right to access certain systems (e.g. username, password, etc)
CSR	Certificate Signing Request: Document or digital information requesting the issuance of a certificate to an End Entity
EGEE	Enabling Grids for E-scienceE: EU funded grid project
End Entity	System (individual, host, service) that receives a certificate expressing its identity
EUGRIDPMA	International organization to coordinate the trust fabric for e-Science grid authentication in Europe. See <a href="http://www.eugridpma.org">http://www.eugridpma.org</a> for details.
Federation	Collection of organizations that agree to interoperate under a certain rule set.
Federated Identity	Management and use of identity information across security domains, e.g. between members of a federation. Federated identity inevitably deals with issues like liability, security, privacy and trust.
gLite	Middleware stack developed by the EGEE project
HSM	Hardware Security Module: Hardware-based security device that generates stores and protects cryptographic keys.
Identity Provider	Authority responsible for generating and asserting authentication, authorization and identity information about their users in a security domain
IGTF	International Grid Trust Federation: Body with the goal to harmonize and synchronize PMAs policies to establish and maintain global trust relationships in e-Science. See <a href="http://www.igtf.org">http://www.igtf.org</a> for details.
ITU	International Telecommunication Union: International organization established to standardize and regulate international radio and telecommunication.
ITU-T	International Telecommunication Standardization Sector

MICS	Member Integrated Credential Service: An IGTF profile for issuing (long-lived) X.509 certificates to End Entities based on an identity management system operated by an institution.
OASIS	A not-for-profit global consortium involved in the development, convergence and adoption of e-business standards. See <a href="http://www.oasis-open.org">http://www.oasis-open.org</a> for details.
PMA	Policy Management Authority: Body responsible for defining minimum standards for the CP/CPSs of a PKI infrastructures and accrediting against those standards
PKI	Public Key Infrastructure: Processes and technologies used to issue and manage digital certificates, enabling third parties to authenticate individual users, services and hosts.
RA	Registration Authority: An entity which asserts the identity of a certificate requester to the issuing Certificate Authority.
SAML	Security Assertion Markup Language: an XML framework for exchanging authentication and authorization information. SAML is a standard of OASIS and is the first standard for federated identity.
Service Provider	A collection of resources in the terminology of Shibboleth. Normally a Service Provider only contains one resource.
Shibboleth	Federated identity management solution from Internet2/MACE (Middleware Architecture Committee for Education). It is the name of the architecture as well as the name of the open source implementation.
Short-lived X.509 certificate	An X.509 certificate with a life time of less than 1 million seconds (approx. 11 days)
SLCS	Short-lived credential service: A service returning a short-lived X.509 certificate to a requester after successful authentication
SWITCHaai	Shibboleth Federation operated within the Swiss higher education and research sector. See <a href="http://www.switch.ch/aai">http://www.switch.ch/aai</a> for details.
UI	User Interface: host from where the user interacts with the grid software in the gLite middleware environment.
X.509	ITU-T standard for public key infrastructures. It defines among other things standard formats for certificates. See <a href="http://www.ietf.org/rfc/rfc2459.txt">http://www.ietf.org/rfc/rfc2459.txt</a> for details.
X.509 certificate	Certificate compliant with the format as specified in the X.509 standard.
XML	eXtensible Markup Language

## 2. MOTIVATION

### 2.1. MOTIVATION FOR INTEROPERABILITY BETWEEN SHIBBOLETH AND GRID INFRASTRUCTURES

The authentication of users in grid infrastructures is based on X.509 certificates, which are issued by certification authorities (CA). These CAs, mostly covering one country, form grid trust federations in order to allow users to access different grid infrastructures across different nations using only one X.509 certificate.

Recently novel mechanisms for inter-organisational authentication and authorization have been developed. These authentication and authorization infrastructures (AAI), unlike those used in conventional grids, do not require users to have certificates.

The implementation of AAI in different countries took place independent of grid infrastructures. Often these AAIs were driven by national research and education networks (NREN) and are based on campus user databases. Examples of such AAIs are Shibboleth (developed within the Internet2 project), A-Select (initiated by SURFnet), and PAPI (developed by RedIRIS). Shibboleth itself has attracted significant interest and has found the widest following with national installations in countries such as the USA, Switzerland, Finland, Australia and the UK. In addition, it is currently also under consideration in further European countries.

The following advantages can be gained by achieving interoperability between the grid infrastructures and these national AAIs:

- A significant increase of the potential grid user base. Currently, grid users have to request a user certificate from a certification authority, which most have never heard of previously. With the inclusion of the national AAIs, and thus the campus user databases, it becomes straightforward for a user to obtain grid credentials.
- X.509 certificates have turned out to be difficult for an average user to maintain. He/she has to understand the basics of asymmetric cryptography as well as various credential formats in order to use them in an efficient and secure manner.
- The user has one credential less to maintain.
- The user can access the grid using his/her campus credential, with which he/she already accesses many resources.

### 2.2. MOTIVATION FOR INTEROPERABILITY THROUGH A SHORT-LIVED<sup>1</sup> CREDENTIAL SERVICE (SLCS)

Shibboleth is based on the exchange of digitally signed assertions about users, expressed in a language called the “security assertion markup language [SAML]. These assertions are exchanged between an Identity Provider and a Service Provider. The role of the Identity Provider is to authenticate the user and administer the information about the user. The Service Provider is the resource that the user wants to access. A detailed description of Shibboleth can be found in [R1], [R2].

Because of the pervasive use of X.509 certificates in grid middleware a replacement of X.509 credentials by SAML statements is neither desirable nor practical. Therefore it was decided to issue a short-lived X.509 credential to a user based on his/her successful authentication to a Shibboleth

---

<sup>1</sup> Short-lived in this document means less than 1 million seconds (approx. 11 days).

Identity Provider. The user requests these certificates by executing a shell command on the machine from where he/she wants to submit jobs to the grid (see Figure 1).

The advantages of this approach are:

- The X.509 credential becomes less visible to the user as it is generated upon authentication to the Identity Provider (which the user already frequently uses for authentication purposes).
- The X.509 certificate “disappears” from the user’s point of view after it has expired. He/she thus does not have to retain it over a longer period of time.
- The addition of new software to the gLite middleware is limited to the host used by the user to submit jobs to the grid. In gLite this is the UI component.
- Only one additional service (per Shibboleth federation) has to be developed and deployed. This service is called SLCS and its implementation within the Swiss AAI federation is called SWITCHslcs.

As described above, the distinct advantage of the SLCS is to enable access to the grid infrastructure for Shibboleth users with a minimal amount of effort. However, several caveats should be mentioned as well. First this effort leaves the basic X.509 based grid security infrastructure in place and does not fully exploit the benefits of SAML. Second, the user’s attributes that are being maintained by the Identity Provider of the user are not available to any grid resources. Thus this work is a very useful first step for achieving interoperability between Shibboleth and gLite, but its scope is limited.

### 3. IMPLEMENTATION OF THE SLCS

In this section we present the implementation of the SLCS. We start by describing the SLCS from the user’s point of view, followed by the administrator’s point of view. We then proceed to discuss the principles of the software design and to give a functional description of the process used to obtain a certificate. Lastly we give an overview of the components that make up the service.

#### 3.1. USER’S VIEW OF THE SLCS

From the user’s point of view, he/she executes a shell command on the gLite UI and in return he/she obtains a short-lived certificate (in PEM format) in the \$HOME/.globus directory (see Figure 1).

The usage of the shell command is described in detail in [R3]. The user has to supply as command line arguments the name of the Identity Provider and optionally the user name (if different than username on the gLite UI). The slcs-init command then demands that the user enters his Shibboleth password and a passphrase for the private key that is being generated.

A command line command exists which lists all Identity Providers that are known to the UI.

The user does not have to do any configuration by himself, which is done by the gLite UI administrator.

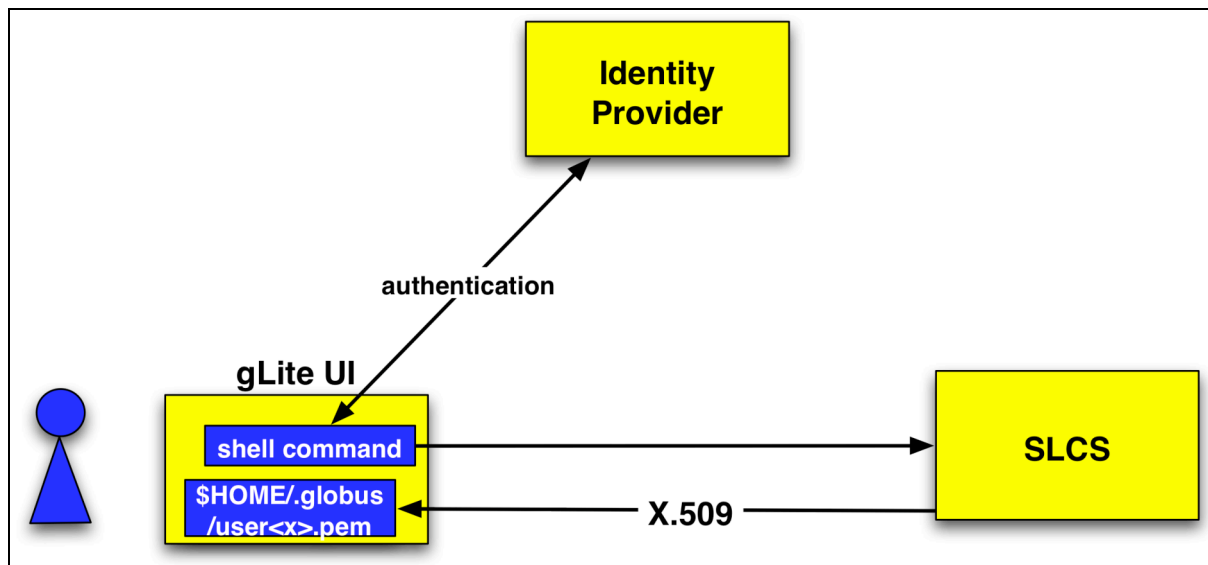


Figure 1 User's view of the SLCS

### 3.2. SHIBBOLETH ADMINISTRATOR'S VIEW OF THE SLCS

There are two tasks that the administrator has to do:

1. Installation and configuration SLCS client on the gLite UI
2. Installation of the SLCS server and (possibly) backend CA

The installation<sup>2</sup> and configuration of the SLCS client is done by the gLite UI administrator, who has to ensure that

1. a federation specific configuration file contains all Identity Providers of the federation and
2. the server certificates of the Shibboleth Identity Provider and SLCS server are known by the user agent if they are not certificates supported by default in generally available browsers.

The federation specific configuration file must be configured outside the gLite configuration mechanisms as a gLite Grid installation may cross several countries whereas Shibboleth federations are normally bound within national borders. However, several Identity Providers from different federations can be listed in the same configuration file but currently only one SLCS server.

The SLCS server is – from the Shibboleth administrator's point of view - a Shibboleth Service Provider that has to be set up on a dedicated host and has to be added to the federation's metadata. There is one SLCS per Shibboleth federation.

The setup and administration of the SLCS is described in detail in [R4].

<sup>2</sup> The SLCS client will be a part of the UI distribution for gLite version 3.1

### 3.3. SOFTWARE DESIGN

The software design of the SLCS is based on the following principles:

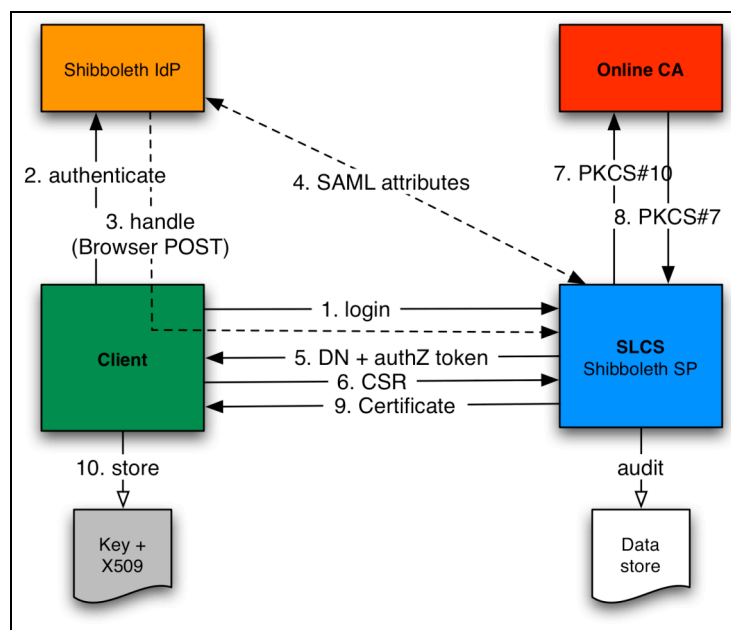
1. From the user's view the software should be a shell command.  
Comment: Shibboleth is, above all, a browser-based tool. However, it cannot be assumed that a browser is installed on the gLite UI.  
Consequence: All HTTP functionality is implemented in an HTTP agent running on the gLite UI.
2. The SLCS should issue short-lived X.509 certificates (and not long-lived or proxy certificates).  
Comment: This service should issue certificates consistent with the SLCS profile [R5] of the IGTF.
3. The private key associated with the certificate should never be transferred over the network.  
Consequence 1: This requires that the key pair is generated on the gLite UI.  
Consequence 2: This requirement excludes some existing certificate factories for the SLCS.
4. Only standard interfaces between software components should be used.  
Consequence: A commercial protocol should be used between the Shibboleth service provider part and the certification authority that issues the certificate.
5. The code should be very modular such that subcomponents can be easily exchanged.
6. The Java programming language should be used.
7. All error messages should be reported back in the original text to the calling party.  
Comment: As the software executes on various hosts, it was decided to report the original error messages as often as possible back to the caller in order to avoid searching in server log files for errors reported by the user executing on the UI. In particular this allows the user to report meaningful errors to Shibboleth or CA operators.

### 3.4. FUNCTIONAL DESCRIPTION

The following steps take place when a user requests a new short-lived certificate from the SLCS (see Figure 2):

1. The user executes the shell command to request a new short-lived certificate. This spawns a user agent, which tries to access the SLCS in order to request a certificate.
2. The SLCS redirects the user agent to the Shibboleth Identity Provider, where the user is authenticated based on his username and password.
3. The Identity Provider issues a handle to the SLCS Service Provider through the Shibboleth Browser POST profile [R6].
4. With the handle the SLCS Service Provider retrieves the SAML attributes of the user and generates a certificate subject derived from the user's attributes.
5. The user agent receives his certificate subject name and an authorizing token as an XML message. He generates locally a private key and a Certificate Signing Request (CSR).
6. The user agent sends the generated CSR and authorizing token back to the SLCS Service Provider.

7. The SLCS receives the CSR and verifies it. A PKCS#10 is sent to the Online CA for signing.
8. The Online CA automatically signs the PKCS#10 and sends a short-lived certificate (PKCS#7) back to the SLCS server.
9. The SLCS server sends the signed short-lived certificate back to the UI as an XML message. The audit log is updated.
10. The user's certificate is stored on the gLite UI.



**Figure 2 Functional description of the certificate generation**

### 3.5. SOFTWARE IMPLEMENTATION

The software implementation is structured in three functional blocks:

1. The SLCS front-end consisting of an Apache server and Shibboleth Service Provider.
2. The SLCS server, which is a Java servlet running under the Tomcat servlet engine.
3. An online CA, which signs the PKCS#10 requests.

Note that these three functional blocks are normally (but not necessarily) installed on separate hosts. Standard software interfaces are used between these components (mod\_jk between Apache and Tomcat and the CMC protocol between Tomcat and the online CA). Further details are given in the section 4 on deployment options.

Figure 3 shows an overview of the software components, on which we briefly comment on some key features.

- There are three servlets within the SLCS:
  - The login servlet controls the login of the user, builds the string Distinguished Name (DN) of the certificate and manages the user sessions.
  - The certificate servlet receives the CSR from the user agent, checks that it is consistent with the policy and communicates with the online CA through a CA client module.
  - The administrator servlet<sup>3</sup> allows an administrator to log in to the SLCS, view the log and add or remove accepted users from an access control list (see chapter 4.2.3).
- Java Filters control the access to these servlets (ACL filters in Figure 3)
- The communication between the servlets and the underlying codes takes place exclusively through standard Java interfaces. The classes implementing these interfaces are loaded at start-up based on entries in a configurable XML file. This mechanism allows easy replacement of existing components. For example if the DN needs to be constructed according to different rule. In this case a new class implementing the DN Builder interface must be written and its name added to the SLCS XML configuration file.
- The CA client interface is responsible for the connection to the online CA. The current distribution contains a class CMC Client, which implements the RFC 2797 protocol.

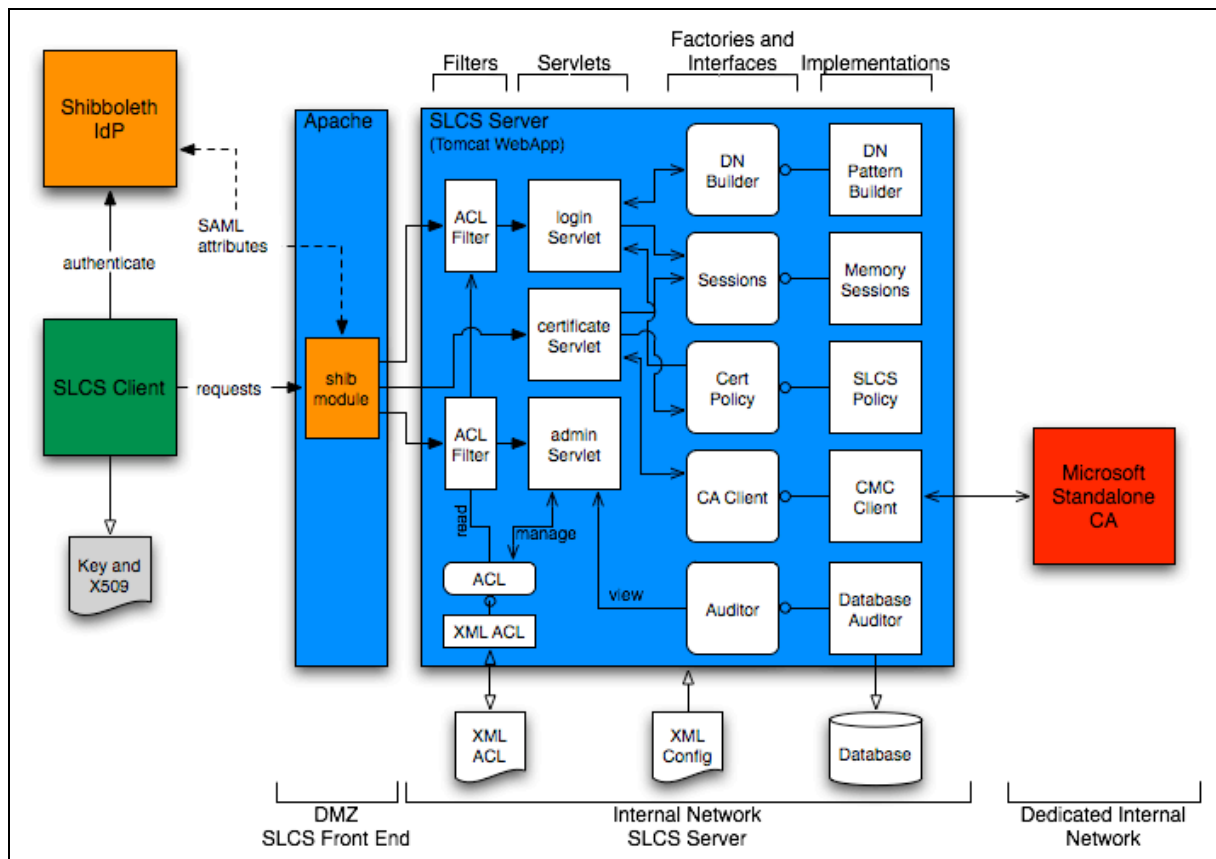


Figure 3 Software architecture of the SLCS

<sup>3</sup> This functionality will be added in the version 1.1.

### 3.6. OPERATION IN A TEST-BED

The SLCS software was tested in a SWITCH internal test-bed, where the following tests were performed:

- Over 3'000 certificates were issued. In three cases problems were encountered: two of them were network connection errors, the other a timeout by the Shibboleth attribute authority, which may or may not be network related.
- Two different CA's were used to issue certificates. One was a SWITCH internal CA, which was based on the Microsoft Credential Service, the other a test CA operated by SwissSign, a commercial company, with which SWITCH has a partner agreement.
- Jobs were submitted to the SWITCH internal gLite test-bed, which includes a computing and storage element as well as a resource broker and a UI. No problems were encountered using the SLCS certificates, whereas the SLCS CA certificate and policy files had to be added by hand to the IGTF distribution files by hand).

## 4. DEPLOYMENT ISSUES

Two components must be deployed in order to make the SLCS service available to grid users:

1. UI client component
2. SLCS server component

We describe in this section the different deployment scenarios for these two components.

### 4.1. UI CLIENT COMPONENT

The user interacts directly with the UI client component by invoking it through the command line. There are two possibilities for deploying the UI client component:

1. Through the gLite distribution: In this case the UI system administrator downloads an .rpm file from the official gLite repository and installs it through the standard gLite installation procedures. This is the recommended way for installing the UI client in a standard gLite installation.
2. By installing a .tar.gz file: In this case the UI system administrators installs a .tar.gz file containing all the Java classes and libraries of the UI client component into a well known directory and adjusts the classpath correspondingly. This is the recommended way to install in the UI client in a non-gLite or standalone environment.

### 4.2. SLCS SERVER DEPLOYMENT

Several fundamentally different options exist for the SLCS deployment and the following list of questions should be answered before deployment is started:

1. In which Shibboleth federation should the SLCS service be added and is its addition to the federation compliant with the policy of the federation?
2. Shall the service be accredited by the IGTF (resp. EUGRIDPMA) or not?
3. How shall the service be operated within the federation?

We now comment on these three questions and outline various deployment scenarios for each of the possibilities.

#### **4.2.1. Compliance of the SLCS with the Shibboleth Federation Policy**

The SLCS is ultimately tied to a Shibboleth federation and acts as a Shibboleth Service Provider. If the operation of such a service is not in agreement with the policy of the federation, then it does not make sense to install it in first place.

#### **4.2.2. Accreditation with the IGTF**

If the SLCS is not supposed to be accredited with the IGTF, then it is up to the system administrator to install and operate the SLCS within the framework of the federation policy. A minimal setup consists of installing the Apache web server and the Tomcat servlet engine on one host with the online CA on another (see Figure 3).

However, if the SLCS service is supposed to be accredited with the IGTF, then it must fulfil the requirements of one of the IGTF profiles, such as the SLCS profile or possibly the MICS profile (see the IGTF website for more details). These profiles put requirements on the SLCS service for:

- Network connectivity and security of the service: the service must operate in a highly secure network with tight network access control;
- Security of the private keys associated with the online CA: these keys should be stored in a hardware security module (HSM);
- Controlled physical access to the server hosting the service;
- Operational requirements: availability of the service, auditing capabilities etc.

These requirements have to be formulated in a CP/CPS, which must be reviewed by a PMA, such as the EUGRIDPMA.

SWITCH decided to accredit the SLCS with the EUGRIDPMA. The procedure is supposed to be finished in the first quarter of 2007. The proposed deployment of the SLCS service is shown in Figure 4. It corresponds to a high security installation where the Apache front end is located in a DMZ, the SLCS service and the online CA in a highly secure area with special network and access control. The private key of the online CA is stored in a HSM.

Note, that the accreditation also puts requirements on the quality of the Shibboleth Identity Providers. This is outside the scope of the SLCS service and is therefore not discussed in this document.

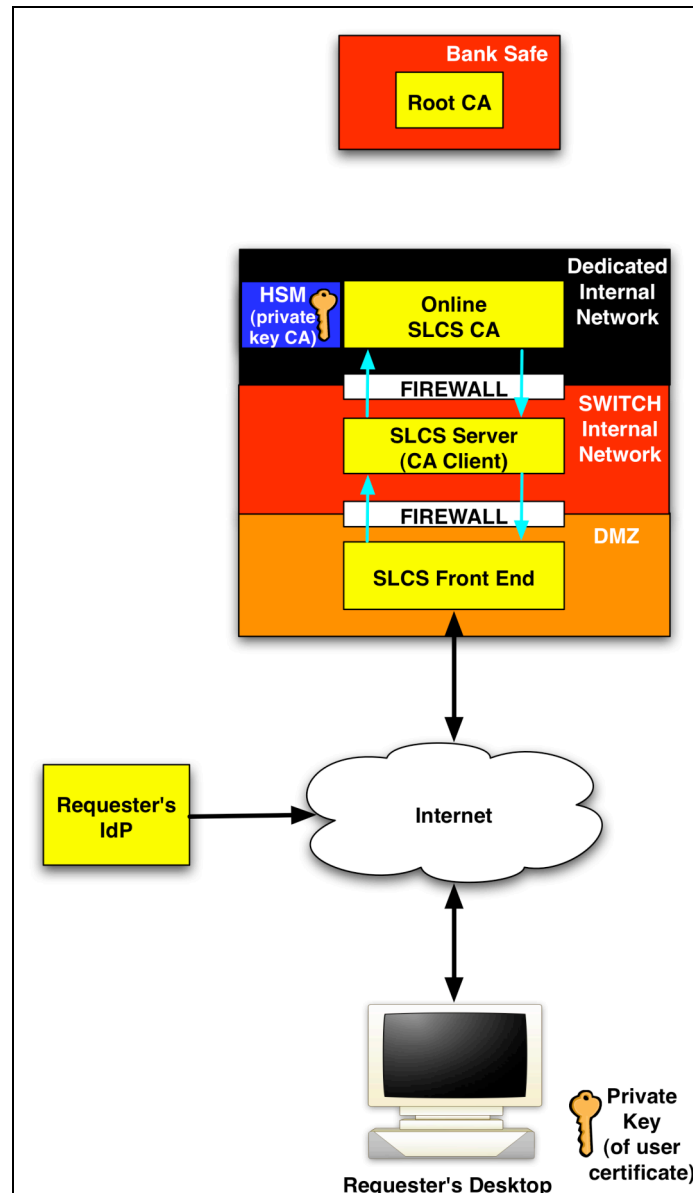


Figure 4 Example of SLCS high security setup

#### 4.2.3. Operational Modes

Besides setting up the SLCS server, it must also be decided how the service should be operated from an organizational point of view. The following possible modes can be envisaged:

1. Access to SLCS without any identity vetting of requesters: In this mode the SLCS server is set up within a federation and every member of the federation has access to the SLCS and can thus obtain a certificate. In this mode the ACL filters described in 3.5 are not configured.
2. Access to SLCS with identity vetting of requesters: In this mode every requester must go through a registration process at a registration authority (RA) that checks the identity of the requester and confirms the validity of his/her Shibboleth account. The RA structure can be set up in various ways:

- a. One RA per federation: This means that a requester must register with an authority which is not necessarily close to his work place and of which he probably has never heard before.
- b. One RA per Shibboleth Identity Provider: This means that a requester must register with a representative of his Identity Provider. This mode requires setup of an RA structure but is most convenient for the requester as he can register at his home institution.
- c. An RA organizational structure independent of the federation and Identity Provider: As this option does not rely on an already existing organizational framework, it most likely involves more work to set it up than the other possibilities.

For accreditation by the IGTF, the identity vetting process must be described in the CP/CPS and be approved by the corresponding PMA, e.g. the EUGRIDPMA.

SWITCH intends to operate the SLCS in the mode 2.b, whereby the system administrators of the Shibboleth Identity Providers perform initial identity vetting of requesters.

## 5. SUMMARY AND OUTLOOK

We described in this document the motivation for, and the software structure of, a short-lived credential service that issues X.509 certificates based upon successful authentication at a Shibboleth Identity Provider. We also reviewed various deployment options and possible operational modes for vetting the identity of prospective users (either with or without a registration authority).

The next release of the SLCS software will add the following functionalities:

- Administrator servlet allowing different RAs to manage the access of their users to the SLCS.
- User access to the SLCS through browsers enabling users to obtain a short-lived certificate that resides in his browser. This functionality is sometimes needed when accessing web-based resources or registration procedures.

SWITCH developed the code of this service as part of the EGEE-II project. The source code has been added to the gLite CVS repository as the modules `org.glite.slcs` [R7]. SWITCH operated the SLCS server in an internal gLite test-bed, where over 1'500 certificates were issued. In three cases problems were encountered, which are still under investigation (two of them seem to be network connection errors). It is planned to continue this test bed over the next few months and also include external users. The goal is to issue over 10'000 certificates during this test phase.

This test phase will end once the service is accredited by the EUGRIDPMA, the service will become a production service. The accreditation process has been started in early October 2006 and should be finished in early 2007. The CP/CPS, currently in draft form, specifies a high security setup of the online CA, for which the commercially available Microsoft Certificate Server (MSCS) will be used. The operational mode chosen is based on registration authorities, which perform an initial identity vetting in accordance with the CP/CPS of this service. The system administrators of the Shibboleth Identity Providers will thereby act as registration authority