EUROPEAN MIDDLEWARE INITIATIVE

EUROPEAN MIDDLEWARE INITIATIVE

GRIDSITE – FUNCTIONAL DESCRIPTION

Document version:	**2.0.0-1**
EMI Component Version:	**1.x, 2.x**
Date:	**February 6, 2013**

This work is co-funded by the European Commission as part of the EMI project under Grant Agreement INFSO-RI-261611.

CONTENTS

GridSite – Functional Description

EMI

February 6, 2013

1 MODULE ARCHITECTURE

The **Module Architecture** implemented by GridSite extends Apache by adding extra components at the access control and HTTP method execution layers, and supports CGI services supplied with GridSite or written by third parties.

mod_gridsite is a loadable module for the Apache web server which provides access control and page formatting for GridSite HTTP(S) Fileservers, Websites and Web Services hosts. mod_gridsite also intercepts some processing in the standard mod_ssl module to support GSI Proxies and VOMS attribute certificates, as well as the normal X.509 client certificates. The verification of these credentials is handled by functions within mod_gridsite and the main GridSite shared library, without the need to patch or rebuild mod_ssl.

EMI

2 DELEGATION PROTOCOL

2.1 INTRODUCTION

Delegation is a common requirement for a wide range of Grid applications. In line with the Web Services based Open Grid Services Architecture, and considering the fact that current in-place security mechanisms will continue to be used, defining a standard delegation mechanism is an essential effort.

By describing delegation as a standalone Web Services portType, and by providing ready-to-use library implementations of this portType, service implementers do not need to deal with the details of the delegation mechanisms, and can factor this functionality out of the internal application logic. Furthermore,

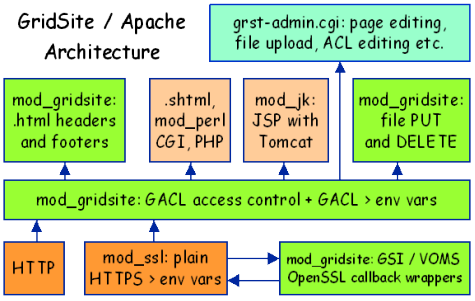

Figure 1: GridSite/Apache Architecture

delegation can be instantiated as a standalone service, front-ending a (trusted) credential repository, from which the target service can then extract the delegated credential in a controlled and secure manner.

The Delegation portType implementation included in GridSite illustrates how web services can be built using C and gSOAP (*http://www.cs.fsu.edu/~engelen/soap.html*) toolkit. The protocol was agreed within the EGEE project, and both C (GridSite) and Java implementation are available as part of EGEE's gLite framework. This document describes the GridSite implementation of the framework.

2.2 DEFINITIONS

- *Delegation* is the act of transferring rights and privileges to another party (the Delegatee).
- *The Delegatee* is the requestor of delegation. It is the entity that the Delegation Credential is delegated to.
- *The Delegator* is the entity that delegates the abilities and/or rights to the Delegatee.
- *A proxy certificate* is an X.509 certificates issued by the end-user (the Delegator) to a process acting on end-user's behalf (the Delegatee). The proxy certificate carries the identity of the Delegator. That is, it can be used to establish SSL connections and is interpreted using the GSI as authority to perform work on the Delegator's behalf.
- *gSOAP* is a set of tools for generating the WSDL description of a Web Service from the C header files of the functions that implement it or vice versa.

2.3 BRIEF DESCRIPTION OF THE GRIDSITE DELEGATION SERVICE

The GridSite Delegation Service (GridDeS) operates as a service with a single portType. It can be operated as a CGI program with a WSDL description and signing functions or standalone service. As a CGI program, the SOAP request received by Apache will be fed into the standard input of the CGI program and the SOAP response will be taken from the standard output. All the required authentication information by the CGI Web Service program is made available as environment variables by the GridSite.

The GridDeS is linked to the GridSite library to obtain access to its private and certificate signing functions because of the two levels of credential processing taking place i.e. authentication and authorization of the client attempting the delegation.

2.4 USER SCENARIOS

GSI delegation is basically exchange of messages between two partners. To perform delegation, the client sends a Get Proxy Request to the server, which causes the server to generate a public and private key and return an X.509 certificate signing request containing the public key.

The client then signs this request using its own private key and certificate, and sends a Put Proxy message back to the server, containing the signed certificate. Together the private key which the server generated and the new certificate form the proxy certificate (RFC3820 / GSI proxy).

Some of the use cases for delegation that are not well covered by X.509 public key certificate alone are:

- *Dynamic delegation*: It is often the case that a Grid user needs to delegate some subset of their privileges to another entity on relatively short notice and only for a brief amount of time.

- *Dynamic entities*: In addition to delegation to persistent services and entities, the requirement exists to support delegation of privileges to services that are created dynamically, often by the user them self, that do not hold any form of identity credential. A common scenario is that a user submits a job to a computational resource and wants to delegate privileges to the job to allow it to access other resources on the user's behalf, for example, to access data belonging to the user on other resources or start sub-jobs on other resources.

- *Repeated Authentication*: It is common practice to protect the private keys associated with X.509 public key certificates either by encrypting them with a pass phrase (if stored on disk) or by requiring a PIN for access (if on a smart card). This technique poses a burden on users who need to authenticate repeatedly in a short period of time, which occurs frequently in Grid scenarios when a user is coordinating a number of resources.

2.5 GETTING STARTED

The followings are required for building the GridDeS:

- the GridSite source code.

- a C or C++ compiler.

- the gSOAP 'stdsoap2' stub and skeleton compiler.

- the gSOAP 'wsdl2h' WSDL parser, which converst WSDL into a gSOAP specification header file for the 'stdsoap2' stub and skeleton compiler.

- the 'stdsoap2.c' and 'stdsoap2.h' files containing the runtime library to be linked with the GridDeS application.

The gSOAP can be downloaded from [4].

2.6 QUICK USER GUIDES

The user guide provides a quick way on how to get started with the GridDeS using gSOAP toolkit. You need a basic understanding of the SOAP protocol and some familiarity with C and/or C++. The GridDeS and its client are developed in C using the gSOAP tools. You don't need a detailed understanding of the SOAP protocol before using GridDeS. In this section, we describe the GridDeS and its client.

2.6.1 COMPILING THE DELEGATION SERVICE

The delegation service implementation is included with the GridSite download. After downloading the gSOAP, set up the directory location containing gSOAP soapcpp2 and stdsoap2.h in the Makefile located

in the source sub-directory (src) of the GridSite. To compile the service 'type make gridsite-delegation.cgi' in the 'src' directory. This will compile the delegation service as a CGI program.

The compilation of the CGI program starts with the invocation of the gSOAP stub and skeleton compiler on the delegation.h header file from within the Makefile. The compiler generates the skeleton routines for the getProxyReq and putProxy remote methods specified in the delegation.h header file, and also produces a WSDL that advertises the delegation service. The skeleton routines are respectively, soap_serve_ns__getProxyReq and soap_serve_ns__putProxy and saved in the file soapServer.c. The generated file soapC.c contains serializers and deserializers for the skeleton. The compiler also generates a service dispatcher-the soap_serve function. See [4] for further details of how to develop and compile a web service with gSOAP.

2.6.2 INSTALLATION AND CONFIGURATION OF THE DELEGATION SERVICE

The service is designed to run as run as a CGI application at GridSite location (https://grid7.hep.man.ac.uk/gridsite-delegation.cgi) for example. To install the service, copy the compiled gridsite-delegation.cgi to the 'cgi-bin' directory of your Web Server or set the installation directory in the http.conf of the Web Server using ScriptAlias.

You will also need to create a new directory called 'proxycache' in the www directory of your Web Server where the proxy certificates are going to be stored. See documentation for GridSite [6] on how to get and configure the host certificate and host key for your Web Server machine.

2.6.3 RUNNING THE DELEGATION SERVICE

After the installation and configuration, the GridDeS is ready to serve request from the clients. The soap_serve function as mentioned above acts as a service dispatcher. It listens to client requests on standard input stream and invokes the remote method implementation function via a skeleton routine to serve a SOAP client request. The response is encoded in SOAP and send to standard output after the request is served. Note that the function prototype of the remote method implementation function is specified in the header file that serves as input to the gSOAP compiler.

2.7 DELEGATION SERVICE CLIENTS

The GridDeS is designed to work with any type of clients that use SOAP protocol. There is an example client that comes with the GridDeS, which is command line based client. To compile this client, type 'make htproxyput' in the delegation.h header file directory location. To run or use the client, you must copy your valid certificate (usercert and userkey) into the '.globus' directory of your machine. The certificate is used to sign the proxy certificate. The GridSite Toolbar is a browser based client for the GridDeS in the form of a Mozilla Firefox (*http:// www.mozilla.com*) extension.

More types of clients possible?

- Browser based with Applet/Java Script/PHP?
- Small Java application agent that can be downloaded and installed through the Internet?
- other methods?

2.8 GRIDSITE DELEGATION SERVICE PROTOCOL SPECIFICATION

This section defines the message exchange required for the GridDeS implementation.

2.8.1 PROTOCOL OVERVIEW

GridDeS protocol is based on SOAP Request/Response message pair from a Delegator to a Delegatee or vice versa. The followings show the GridDeS Request/Response steps:

- The Delegator issues a getProxy request and send that to Delegatee, who verifies the validity of the request, generates a pair of key (public and private keys) and sends the response containing the public key back to the Delegator.

- The Delegator receives the response message, signs the public key with its certificate and issues a putProxy request to the Delegatee. The public key, which is signed using the Delegator's private key associated with its long time public key certificate constitutes the proxy certificate which is kept together with private key in a file with the Delegatee.

2.8.2 PROTOCOL OPERATIONS

The GridDeS presently supports two operations that can be requested by the Delegator to be performed by the Delegatee:

getProxyReq creates a new delegation request. The end result is a pair of key for the proxy certificate.

putProxy sends a new signed public key back to the Delegatee. The end result is a proxy certificate.

2.8.3 GRIDSITE DELEGATION SERVICE MESSAGE STRUCTURES

The GridDeS message model as described consists of two SOAP messages: Request and Response.

- *Request Message*: This message is issued by the Delegator either to ask the Delegatee for a public key to base the proxy certificate on or to push the proxy certificate to the Delegatee. A sample of the SOAP Request message from the Delegator to the Delegatee for getProxyReq is depicted below:

```
. . . .
<SOAP-ENV:Envelope xmlns:SOAP-ENV="\url{http://schemas.xmlsoap.org/soap/envelope/}"
xmlns:SOAP-ENC="\url{http://schemas.xmlsoap.org/soap/encoding/}"
xmlns:xsi="\url{http://www.w3.org/2001/XMLSchema-instance}"
xmlns:xsd="\url{http://www.w3.org/2001/XMLSchema}"
xmlns:ns="urn:delegation"
SOAP-ENV:encodingStyle="\url{http://schemas.xmlsoap.org/soap/encoding/}">
<SOAP-ENV:Body>
 <ns:getProxyReqRequest>
  <delegationID xsi:type="xsd:string">20050001</delegationID>
 </ns:getProxyReqRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- *Response Message*: This message contains a generated public key and it is sent from the Delegatee to the Delegator in response to a getProxyReq request. Sample of the above getProxyReq Response message is shown below:

```
. . . .
<SOAP-ENV:Envelope xmlns:SOAP-ENV="\url{http://schemas.xmlsoap.org/soap/envelope/}"
xmlns:SOAP-ENC="\url{http://schemas.xmlsoap.org/soap/encoding/}"
xmlns:xsi="\url{http://www.w3.org/2001/XMLSchema-instance}"
xmlns:xsd="\url{http://www.w3.org/2001/XMLSchema}"
xmlns:ns="urn:delegation"
SOAP-ENV:encodingStyle="\url{http://schemas.xmlsoap.org/soap/encoding/}">
<SOAP-ENV:Body>
 <ns:getProxyReqResponse>
  <request xsi:type="xsd:string">MIIuuBB33JjjjjO222MmEppoooddWwwwlll . . . </request>
  </ns:getProxyReqResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.8.4 GRIDSITE DELEGATION SERVICE WSDL

The WSDL description of the GridDeS reproduced below is automatically generated from the header file using gSOAP.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="delegation"
 targetNamespace="\url{http://www.gridsite.org/ns/delegation.wsdl}"
 xmlns:tns="\url{http://www.gridsite.org/ns/delegation.wsdl}"
 xmlns:SOAP-ENV="\url{http://schemas.xmlsoap.org/soap/envelope/}"
 xmlns:SOAP-ENC="\url{http://schemas.xmlsoap.org/soap/encoding/}"
 xmlns:xsi="\url{http://www.w3.org/2001/XMLSchema-instance}"
 xmlns:xsd="\url{http://www.w3.org/2001/XMLSchema}"
 xmlns:ns="urn:delegation"
 xmlns:SOAP="\url{http://schemas.xmlsoap.org/wsdl/soap/}"
 xmlns:MIME="\url{http://schemas.xmlsoap.org/wsdl/mime/}"
 xmlns:DIME="\url{http://schemas.xmlsoap.org/ws/2002/04/dime/wsdl/}"
 xmlns:WSDL="\url{http://schemas.xmlsoap.org/wsdl/}"
 xmlns="\url{http://schemas.xmlsoap.org/wsdl/}">
<types>
 <schema targetNamespace="urn:delegation"
  xmlns:SOAP-ENV="\url{http://schemas.xmlsoap.org/soap/envelope/}"
  xmlns:SOAP-ENC="\url{http://schemas.xmlsoap.org/soap/encoding/}"
  xmlns:xsi="\url{http://www.w3.org/2001/XMLSchema-instance}"
  xmlns:xsd="\url{http://www.w3.org/2001/XMLSchema}"
  xmlns:ns="urn:delegation"
  xmlns="\url{http://www.w3.org/2001/XMLSchema}"
  elementFormDefault="unqualified"
  attributeFormDefault="unqualified">
  <import namespace="\url{http://schemas.xmlsoap.org/soap/encoding/}"/>
```

```
 </schema>
</types>
<message name="getProxyReqRequest">
 <part name="delegationID" type="xsd:string"/>
</message>
<message "getProxyReqResponse">
 <part name="request" type="xsd:string"/>
</message>
</message>
<message name="putProxy">
 <part name="delegationID" type="xsd:string"/>
 <part name="proxy" type="xsd:string"/>
</message>
<message name="putProxyResponse">
</message>
<portType name="delegationPortType">
 <operation name="getProxyReq">
  <documentation>Service definition of function ns__getProxyReq</documentation>
  <input message="tns:getProxyReqRequest"/>
  <output message="tns:getProxyReqResponse"/>
 </operation>
 <operation name="putProxy">
  <documentation>Service definition of function ns__putProxy</documentation>
  <input message="tns:putProxy"/>
  <output message="tns:putProxyResponse"/>
 </operation>
</portType>
<binding name="delegation" type="tns:delegationPortType">
 <SOAP:binding style="rpc" transport="\url{http://schemas.xmlsoap.org/soap/http}"/>
 <operation name="getProxyReq">
  <SOAP:operation style="rpc" soapAction=""/>
  <input>
     <SOAP:body use="encoded" namespace="urn:delegation" encodingStyle="\url{http://schemas.xml
  </input>
  <output>
   <SOAP:body use="encoded" namespace="urn:delegation" encodingStyle="\url{http://schemas.xmlso
  </output>
 </operation>
 <operation name="putProxy">
  <SOAP:operation style="rpc" soapAction=""/>
  <input>
     <SOAP:body use="encoded" namespace="urn:delegation" encodingStyle="\url{http://schemas.xml
  </input>
  <output>
   <SOAP:body use="encoded" namespace="urn:delegation" encodingStyle="\url{http://schemas.xmlso
  </output>
 </operation>
</binding>
```

```
<service name="delegation">
 <documentation>gSOAP 2.7.4 generated service definition</documentation>
 <port name="delegation" binding="tns:delegation">
  <SOAP:address location="\url{http://localhost/delegserver.cgi}"/>
 </port>
</service>
</definitions>
```

2.9 DELEGATION SERVICE REMOTE METHODS SPECIFICATION

The GridDeS is developed by developing a header file that contains the C service remote methods and data types.The remote methods are based on the above GridDeS protocol specification. These methods must be implemented in the server application. The function prototype in the header file must be a valid prototype of the method implemented as a C/C++ function. See section on compiling the delegation service on how to compile the delegation service and its WSDL from the header file. The header file for the GridDeS is shown below:

```
//gsoap ns service name: delegation
//gsoap ns service style: rpc
//gsoap ns service encoding: encoded
//gsoap ns service namespace: \url{http://www.gridsite.org/ns/delegation.wsdl}
//gsoap ns service location: \url{http://localhost/delegserver.cgi}
struct ns__putProxyResponse { }~;
//gsoap ns schema namespace: urn:delegation
int ns__getProxyReq(char *delegationID, char **request);
int ns__putProxy(char *delegationID, char *proxy,  struct ns__putProxyResponse *unused);
```

The remote method 'ns_putProxy' has a response struct that is empty because it has no output parameters. The WSDL description of the delegation service generated from the header file is also located in the same directory with the Makefile (i.e. src directory).

2.9.1 IMPLEMENTATION OF THE REMOTE METHODS

As earlier stated, the application of gSOAP compiler on the header file produces C/C++ source files that are used to build the delegation service and client applications in C/C++. The gSOAP compiler acts as a preprocessor. The details of the files generated by the compiler are:

```
\textit{File Name}  \textit{Description}
soapStub.h  A modified and annotated header file produced from input header file
soapH.h  Main header file to be included by all client and service sources
soapC.c  Serializers and deserializers for the specified data structures
soapClient.c  Client stub routines for remote operations
soapServer.c  Service skeleton routines
soapClientLib.c  Client stubs combined with local static (de)serializers
soapServerLib.c  Service skeletons combined with local static (de)serializers
```

The implementation of the remote methods return a SOAP status code. The code SOAP_OK denotes success, while soap_sender_fault returns an exception.

2.9.2 GRIDSITE X.509 CERTIFICATE FUNCTIONS

The GridDeS remote methods implementation uses some function from the GridSite X.509 certificate functions:

The ns_getProxyReq(..) method uses GRSTx590GRSTx509MakeProxyRequest(...) function to make and store a X.509 request for GSI proxy. Its returns GRST_RET_OK on success, non-zero otherwise. The key is stored in proxydir as temporary file.

The ns_putProxy method uses the GRSTx509CacheProxy(...) function to store a GSI proxy chain in the proxy cache along with the private key. It returns GRST_RET_OK on success, non-zero otherwise. The existing private key with the same delegation ID and user DN is appended to make a valid proxy file.

2.10 REFERENCES

[1] McNab, A. and Kaushal, S. The GridSite Security Framework, Department of Physics and Astronomy, University of Manchester, Manchester, UK (Limited circulation).

[2] Welch, V., Foster, I., Kesselman, C., Mulmo, O., Pearlman, L., Teucke, S., Gawor, J., Meder, S. and Siebenlist, F. (2004). X.509 proxy certificate for dynamic delegation, Proceeding of the 3rd Annual PKI R&D Workshop.

[3] Ahsant, M., Basney, J. and Mulmo, O. (2004). Grid Delegation Protocol, UK Workshop on Grid Security Experiences, Oxford.

[4] gSOAP: Generator Tools for Coding SOAP/XML Web Services in C and C++, http://www.cs.fsu.edu/~engelen/soap.html

[5] Snelling, D. F., Van den Berghe, S. and Li, V. Q. (2004). Explicit Trust Delegation: Security for the Dynamic Grids, FUJITSU Sci. Tech. J.

[6] GridSite: Grid Security for the Web platforms for Grids, http://www.gridsite.org/