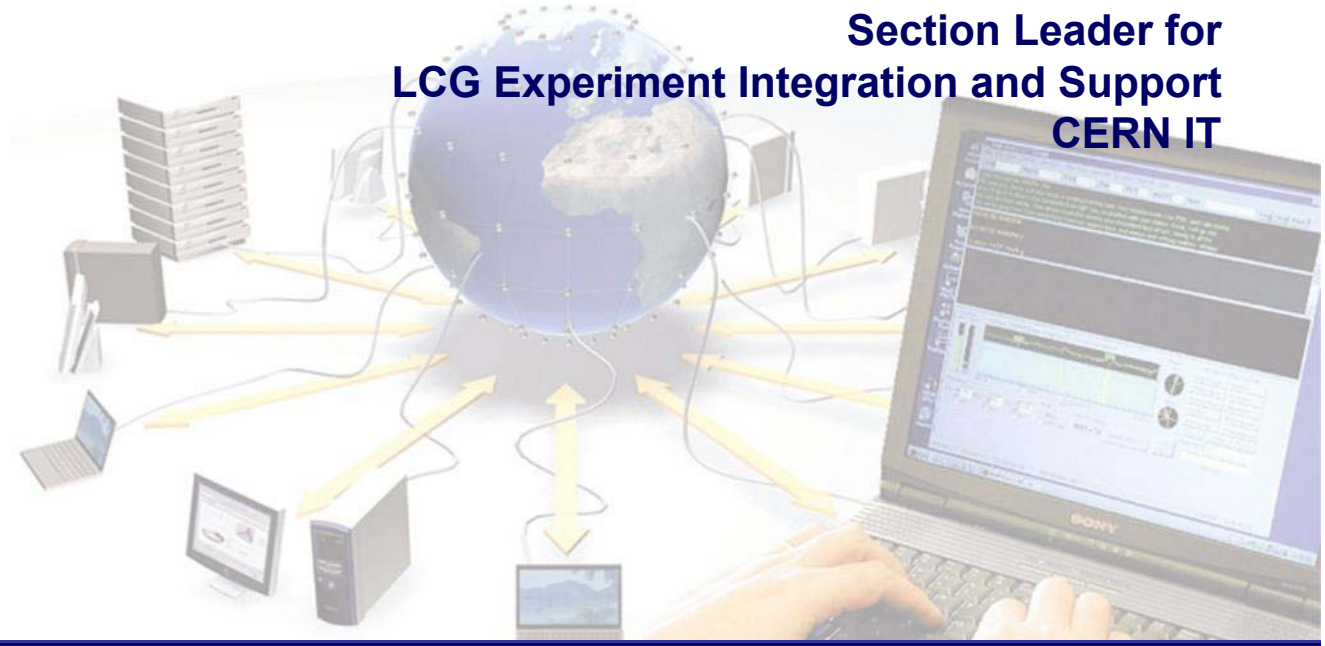
Enabling Grids for E-science in Europe

www.eu-egee.org

Biomed Application Developer's Course
6th October 2004

Authentication and Authorization in LCG-2

Flavia Donno
Section Leader for
LCG Experiment Integration and Support
CERN IT



EGEE is a project funded by the European Union under contract IST-2003-508833

Goals of this module and Overview

Enabling Grids for E-science in Europe

Describe …

- Security basics – Public/Private Keys in action
- Use of Certificates
- Importance of Certificate Authorities
- Virtual Organizations
- Main commands and Globus GSS-API

Introduction to Security

Enabling Grids for E-science in Europe

What aspects of security should we be concerned about?

- Authentication (Identification)
- Confidentiality (Privacy)
- Integrity (non-Tampering)
- Authorisation

Also

- Accounting
- Delegation
- Non-Repudiation

Tools of the trade

Enabling Grids for E-science in Europe

- Encryption
  - Secret “symmetric” key – both parties need to share the key (Kerberos)
    - DES, RC4
    - Comparatively efficient
  - Public/private key – “asymmetric” - 2 keys mathematically related
    - RSA, DSA
    - Slower
- Oneway hash / message digest
    - MD5, SHA-1
    - Fast

The Grid Security Infrastructure (GSI) uses public key cryptography (also known as asymetric cryptography) as the basis for its functionality. GSI uses SSL/TLS

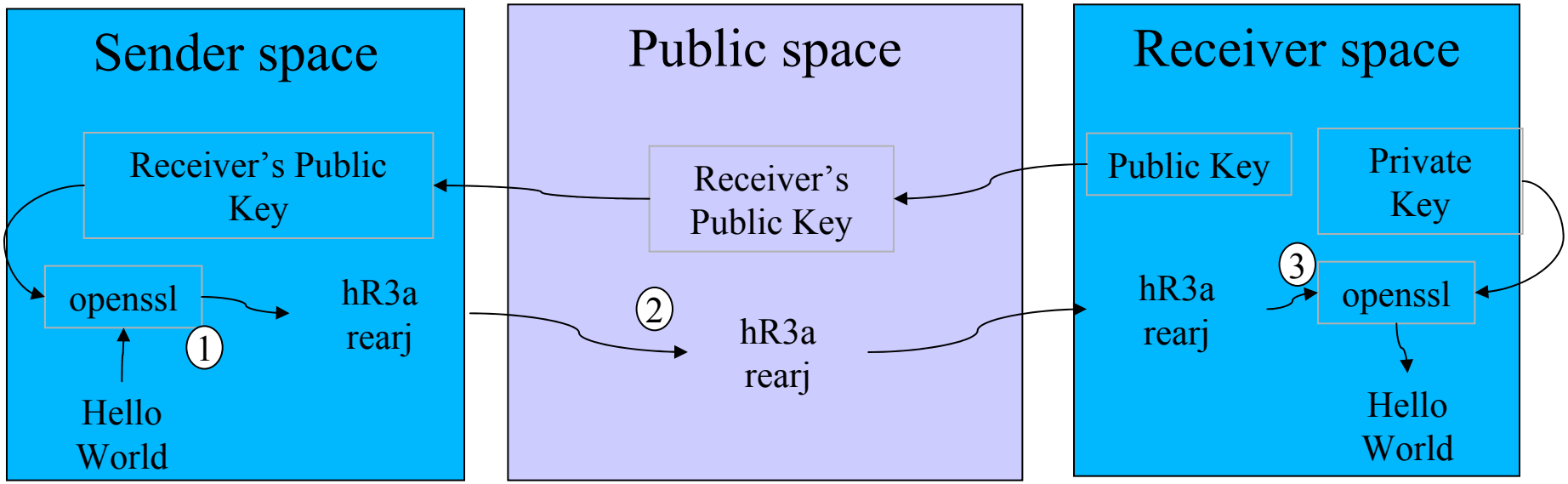
http://www-unix.globus.org/toolkit/docs/3.2/gsi/key/index.html

Encrypting for Confidentiality

Sending a message using asymmetric keys

1. Encrypt message using Receiver's public key
2. Send encrypted message
3. Receiver decrypts message using own private key

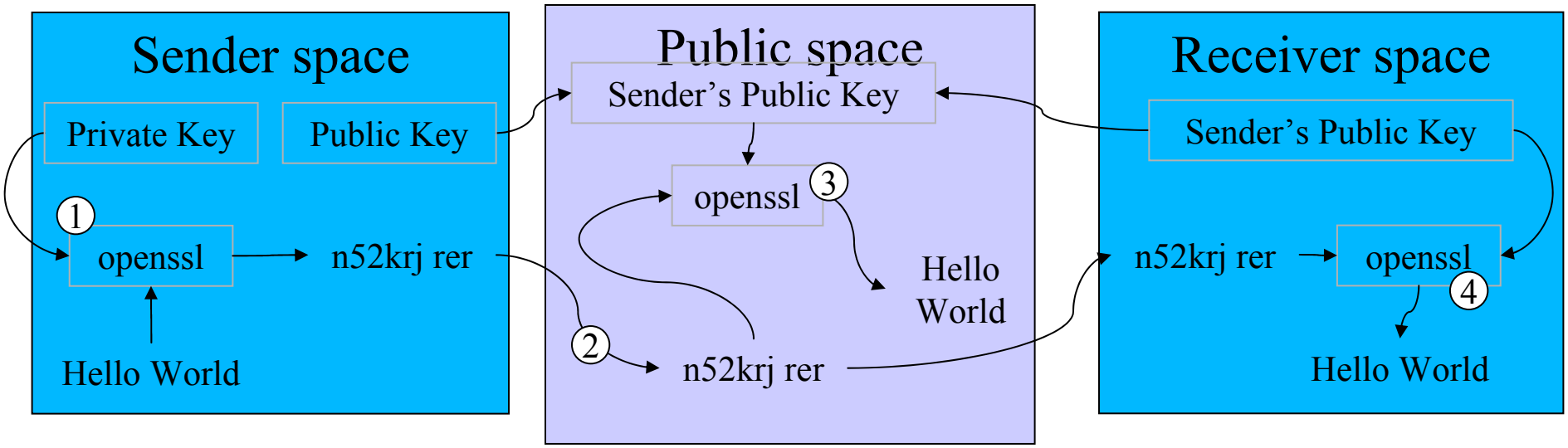
Only someone with Receiver's private key can decrypt message



Signing for Authentication

1. Encrypt message with Sender's private key.
2. Send encrypted message.
3. Message is readable by ANYONE with Sender's public key.
4. Receiver decrypts message with Sender's public key.

Receiver can be confident that only someone with Sender's private key could have sent the message.



Certificates

- A statement from someone else (the Certificate Authority), that your public key (and hence your private key) is associated with your identity
- A certificate can be checked if you have the public key of the party who signed it

CA's	CA's	CA's	CA's
Armenia	CESNET - CZ	NorduGrid	Spain
Belgium	Germany	Pakistan	Taiwan
Canada	Greece	Poland	United Kingdom
CERN	Ireland	Portugal	US DOE
CNRS - France (3)	Israel	Russia	US Esnet
Cyprus	NIKHEF	Slovakia	US FNAL

For a list of all EGEE CAs, please check:
https://lcg-registrar.cern.ch/pki_certificates.html

Certificate Authority

- A Certificate Authority (CA) issues you your certificates.
- By signing them it is able to vouch for you to third parties
- In return for this service, you must provide appropriate documentary evidence of identity when you apply for a certificate through a Registration Authority (RA)

Certificate contents

Enabling Grids for E-science in Europe

- The certificate that you present to others contains:
  - Your distinguished name (DN)
  - Your public key
  - The identity of the CA who issued the certificate
  - Its expiry date
  - Digital signature of the CA which issued it
- Grid Services and Hosts can have certificates

Certificate contents

You must have a valid certificate from a trusted CA!

- Certificate Info: **grid-cert-info**

- „login”: **grid-proxy-init**
  short lifetime certificate: 24 hours

```
Enter PEM pass phrase:
.........................+++++
.....................................+++++
```

  openssl x509 –in /tmp/x509up_u`id -u` -text

- checking the proxy: **grid-proxy-info -subject**

```
/O=Grid/O=CERN/OU=cern.ch/CN=Flavia Donno/CN=proxy
```

- „logout”: **grid-proxy-destroy**

-> use the grid services

Still on Certificates

Enabling Grids for E-science in Europe

- What kind of key does grid-cert-request generate ?
- What is the openssl command to get details about a request ?
- Check the following:

openssl rsa -in ~/.globus/userkey.pem –text

- What is X509 ? - Check:

openssl x509 –in ~/.globus/usercert.pem –text

- What is in:

/etc/grid-security/certificates ?

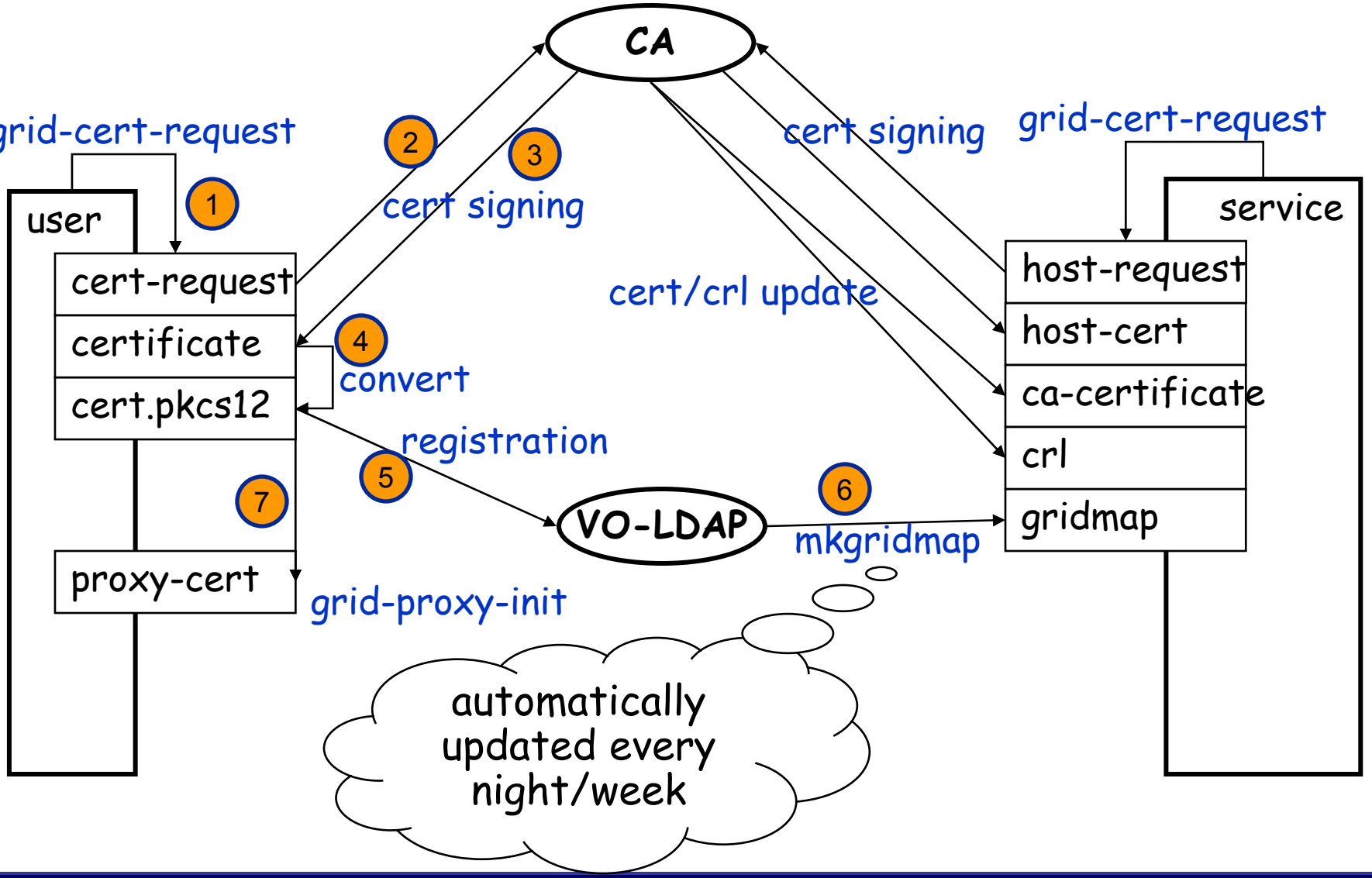
The role of a Virtual Organization

Enabling Grids for E-science in Europe

- A Virtual Organization identifies a group of people and/or resources that share the same working environment
- Access to certain resources is guaranteed only via registration to a specific VO. A list of supported VOs can be found here:

  https://lcg-registrar.cern.ch/virtual_organization.html
- Resources are configured to serve specific VOs

Authorization Information
The role of a Virtual Organization



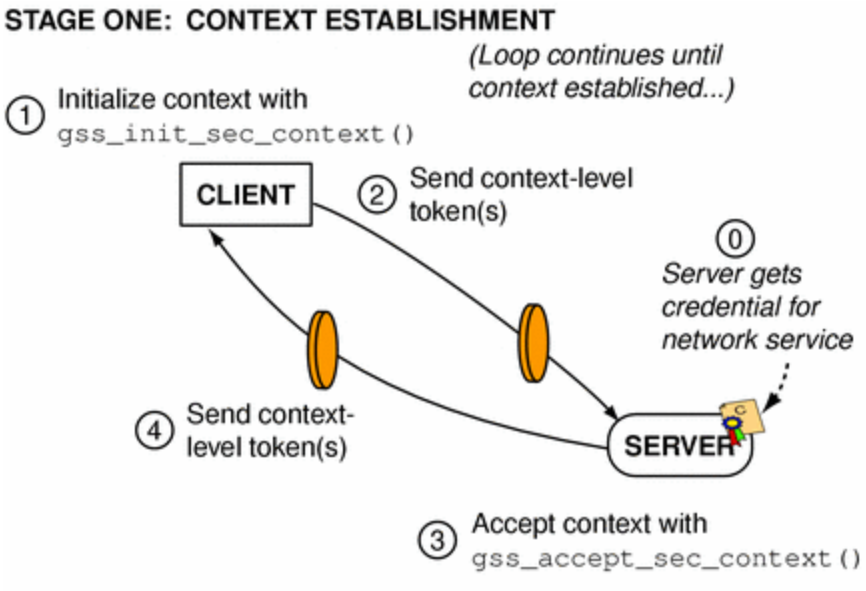
Security APIs

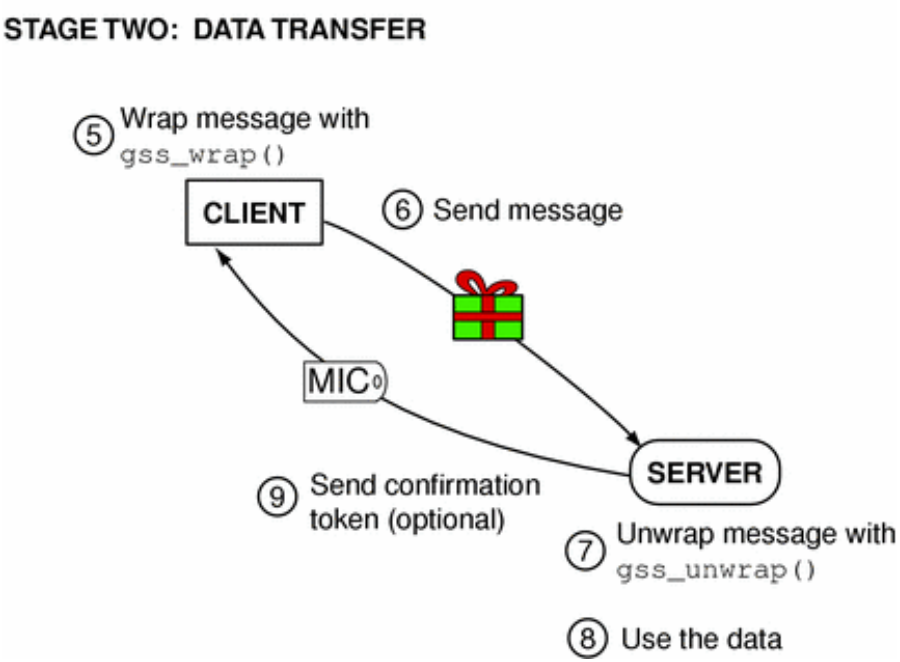
Enabling Grids for E-science in Europe

- For the moment no LCG/EGEE APIs
- Main Contribution comes from Globus.
  The GSS Assist code provides convenience functions for using the Globus GSS-API. – Poor documentation
- Technology dependent APIs: many GSS implementations
- Some development is under way. Check CHEP 2004:

  http://indico.cern.ch/contributionDisplay.py?contribId=78&sessionId=23&confId=0

GSS-API





- The client initiate a context and prepares a token for the server
- The token is sent to the server
- The server interprets the token and prepares a new one to be sent to the client
- The token is sent to the client
- Iterate process until authentication process successes or fails

- The client wraps a message for the server and sends it
- The server receives the message and unwraps it
- The server sends a confirmation message to the client (MIC)
- The client verifies the MIC

Client GSS-API

1) Client can use default credentials or obtain new ones with:

OM_uint32 ***gss_acquire_cred*** (

OM_uint32	***minor_status,**	
const gss_name_t	***desired_name,**	#name of the principal
OM_uint32	***time_req,**	#time validity of credential
const gss_OID_set	***desired_mechs,**	#...suggest: to use default
gss_cred_usage_t	***cred_usage,**	#how cred should be used
gss_cred_id_t	***output_cred_handle,**	#handler for generated cred.
gss_OID_set	***actual_mechs,**	#...suggest: use default
OM_uint32	***time_rec**)	

2) Import the name of the server into GSS-API internal format with:

OM_uint32 ***gss_import_name*** (

OM_uint32	***minor_status,**	
const gss_buffer_t	***input_name_buffer,**	#name to be imported
const gss_OID	***input_name_type,**	#format of input buffer
gss_name_t	***output_name**)	#imported name

Client GSS-API

3) Initiate a context:

```
do{

OM_uint32 maj_status = gss_init_sec_context (

OM_uint32                    *minor_status,
const gss_cred_id_t           initiator_cred_handle,     #could also be def. cred.
gss_ctx_id_t                 *context_handle,            #context handler returned
const gss_name_t              target_name,               #principal to connect to
const gss_OID                 mech_type,                 #mechanism (could be def.)
OM_uint32                     req_flags,
OM_uint32                     time_req,
Const gss_channel_bindings_t  input_chan_bindings,
const gss_buffer_t           input_token                 #token received
gss_OID                      *actual_mech_type,
gss_buffer_t                 output_token,               #token to be sent
OM_uint32                    *ret_flags,
OM_uint32                    *time_rec)
```

→ ***Send the token to the other party*** (how you do that is not a GSS-API issue)

} **while** (maj_stat == GSS_S_CONTINUE_NEEDED);

Client GSS-API

Enabling Grids for E-science in Europe

4) Once the context is established, encrypt the message to sent to the server :

```
OM_uint32 gss_wrap (
OM_uint32                 *minor_status,
const gss_ctx_id_t        context_handle,
int                       conf_req_flag,          #confidentiality & integrity
gss_qop_t                 qop_req                 #quality of protetion
const gss_buffer_t        input_mess_buffer,      #input message
int                       *conf_state,
gss_buffer_t              output_mess_buffer )    #encrypted message
```

→Send the buffer (message) to the other party
→Get the MIC from the other party

5) Verify the MIC:

```
OM_uint32 gss_verify_mic (
OM_uint32                 *minor_status,
const gss_ctx_id_t        context_handle,
const gss_buffer_t        message_buffer,         #message previously sent
const gss_buffer_t        token_buffer,           #received MIC
gss_qop_t                 qop_state)              # quality of protection
```

Server GSS-API

1) Server obtains credentials with **gss_acquire_cred** (can also use the default credentials)

2) Accept context:

→ ***Get token from the client.***

```
do{  OM_uint32 maj_status = gss_accept_sec_context (
OM_uint32                       *minor_status,
gss_ctx_id_t                    *context_handle,
const gss_cred_id_t             acceptor_cred_handle,
const gss_buffer_t              input_token_buffer,
const gss_channel_bindings_t    input_chan_bindings,
const gss_name_t                *src_name,
gss_OID                         *mech_type,
gss_buffer_t                    output_token,
OM_uint32                       *ret_flags,
OM_uint32                       *time_req,
gss_cred_id_t                   *delegated_cred_handle)
} while (maj_stat == GSS_S_CONTINUE_NEEDED);
```

Server GSS-API

3) Once the context is established, get message from the client and decrypt it:

```
OM_uint32 gss_unwrap (
OM_uint32                 *minor_status,
const gss_ctx_id_t        context_handle,
const gss_buffer_t        input_message_buffer,     #wrapped message
gss_buffer_t              output_mess_buffer,       #unwrapped message
int                       *conf_state,
gss_qop_t                 *qop_state)
```

4) Prepare a MIC to send back to the client:

```
OM_uint32 gss_get_mic (
OM_uint32                 *minor_status,
const gss_ctx_id_t        context_handle,
gss_qop_t                 qop_req,
const gss_buffer_t        message_buffer,           #message to be tagged
gss_buffer_t              msg_token)                #token with msg & its MIC
```

-> Send the MIC (token) to the client

An example C GSS-API usage

```
#include <gssapi.h>
#include "globus_gss_assist.h"
#include "gssapi_openssl.h"

void globus_print_error(
   globus_result_t                  error_result);

int main()
{
  [...]
   /* Initialize variables */

   token_ptr = GSS_C_NO_BUFFER;
   init_context = GSS_C_NO_CONTEXT;
   accept_context = GSS_C_NO_CONTEXT;
   del_init_context = GSS_C_NO_CONTEXT;
   del_accept_context = GSS_C_NO_CONTEXT;
   delegated_cred = GSS_C_NO_CREDENTIAL;
   accept_maj_stat = GSS_S_CONTINUE_NEEDED;
   ret_flags = 0;
[...]
 /* acquire the credential */

   maj_stat = gss_acquire_cred(&min_stat,
                               NULL,
                               GSS_C_INDEFINITE,
                               GSS_C_NO_OID_SET,
                               GSS_C_BOTH,
                               &cred_handle,
                               NULL,
                               NULL);
```

```
OM_uint32                 init_maj_stat;
OM_uint32                 accept_maj_stat;
OM_uint32                 maj_stat;
OM_uint32                 min_stat;
OM_uint32                 ret_flags;
OM_uint32                 time_rec;
gss_buffer_desc           send_tok;
gss_buffer_desc           recv_tok;
gss_buffer_desc *         token_ptr;
gss_buffer_desc           oid_buffer;
gss_buffer_set_desc       oid_buffers;
gss_buffer_set_t          inquire_buffers;
gss_OID                   mech_type;
gss_OID_set_desc          oid_set;
gss_name_t                target_name;
gss_ctx_id_t              init_context;
gss_ctx_id_t              accept_context;
gss_ctx_id_desc *         init_context_handle;
gss_ctx_id_t              del_init_context;
gss_ctx_id_t              del_accept_context;
gss_cred_id_t             delegated_cred;
```

An example C GSS-API usage

Enabling Grids for E-science in Europe

```
if(maj_stat != GSS_S_COMPLETE)
  {
    globus_gss_assist_display_status_str(&error_str,
                                         NULL,
                                         maj_stat,
                                         min_stat,
                                         0);
 printf("\nLINE %d ERROR: %s\n\n", __LINE__, error_str);
    globus_print_error((globus_result_t) min_stat);
    exit(1);
  }


  /* get the subject name */

  maj_stat = gss_inquire_cred(&min_stat,
                              cred_handle,
                              &target_name,
                              NULL,
                              NULL,
                              NULL);

  if(maj_stat != GSS_S_COMPLETE)
  {
    globus_gss_assist_display_status_str(&error_str,
                                         NULL,
                                         maj_stat,
                                         min_stat,
                                         0);
    printf("\nLINE %d ERROR: %s\n\n", __LINE__, error_str);
    globus_print_error((globus_result_t) min_stat);
    exit(1);
  }
```

Globus GSS Assist

An example C GSS-API usage

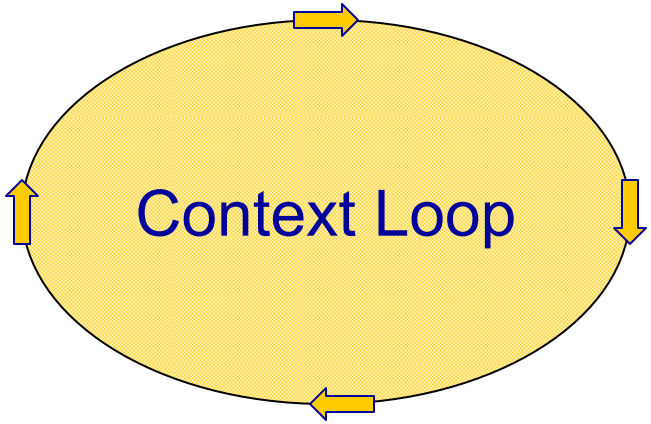
```
/* set up the first security context */

   init_maj_stat = gss_init_sec_context(&min_stat,
                                        cred_handle,
                                        &init_context,
                                        target_name,
                                        GSS_C_NULL_OID,
                                        0,
                                        0,
                                       GSS_C_NO_CHANNEL_BINDINGS,
                                        token_ptr,
                                        NULL,
                                        &send_tok,
                                        NULL,
                                        NULL);


   if(init_maj_stat != GSS_S_CONTINUE_NEEDED)
   {
      globus_gss_assist_display_status_str(&error_str,
                                           NULL,
                                           init_maj_stat,
                                           min_stat,
                                           0);
      printf("\nLINE %d ERROR: %s\n\n", __LINE__, error_str);
      globus_print_error((globus_result_t) min_stat);
      exit(1);
   }
```

Pointer to token to be received

Pointer to token to be sent

An example C GSS-API usage

```
while(1)
  {

     accept_maj_stat=gss_accept_sec_context(&min_stat,
                                  &accept_context,
                                  GSS_C_NO_CREDENTIAL,
                                  &send_tok,
                                  GSS_C_NO_CHANNEL_BINDINGS,
                                  NULL,
                                  &mech_type,
                                  &recv_tok,
                                  &ret_flags,
                                  /* ignore time_rec */
                                  NULL,
                                  GSS_C_NO_CREDENTIAL);

     if(accept_maj_stat != GSS_S_COMPLETE &&
        accept_maj_stat != GSS_S_CONTINUE_NEEDED)
     {
        globus_gss_assist_display_status_str(&error_str,
                                     NULL,
                                     init_maj_stat,
                                     min_stat,
0);
        printf("\nLINE %d ERROR: %s\n\n", __LINE__, error_str);
        globus_print_error((globus_result_t) min_stat);
        exit(1);
     }
     else if(accept_maj_stat == GSS_S_COMPLETE)
     {
        break;
     }
```



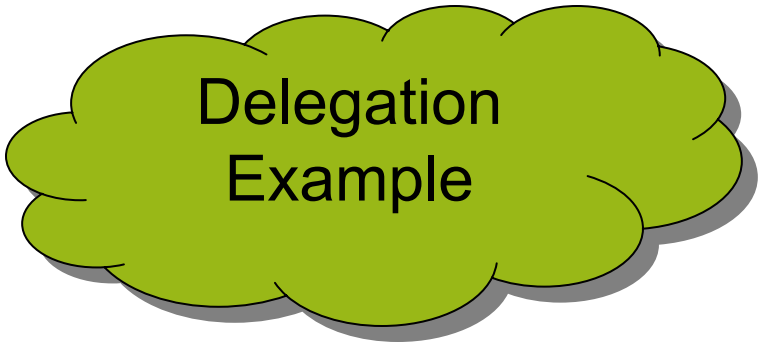
An example C GSS-API usage

```
init_maj_stat = gss_init_sec_context(&min_stat,
                                     GSS_C_NO_CREDENTIAL,
                                     &init_context,
                                     target_name,
                                     GSS_C_NULL_OID,
                                     0,
                                     0,
                                     GSS_C_NO_CHANNEL_BINDINGS,
                                     &recv_tok,
                                     NULL,
                                     &send_tok,
                                     NULL,
                                     NULL);


    if(init_maj_stat != GSS_S_COMPLETE &&
      init_maj_stat != GSS_S_CONTINUE_NEEDED)
    {
       globus_gss_assist_display_status_str(&error_str,
                                            NULL,
                                            init_maj_stat,
                                            min_stat,
                                            0);
       printf("\nLINE %d ERROR: %s\n\n", __LINE__, error_str);
       globus_print_error((globus_result_t) min_stat);
       exit(1);
    }
}

printf("%s:%d: Successfully established initial security context\n",
       __FILE__,
       __LINE__);
```

Context established

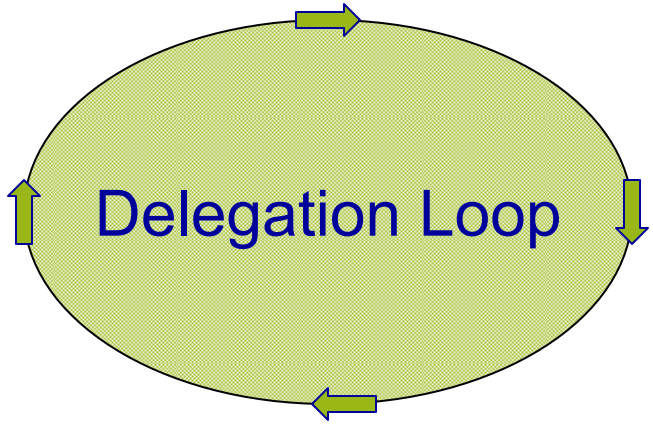
An example C GSS-API usage

```
 init_maj_stat = gss_init_delegation(&min_stat,
                                     init_context,
                                     cred_handle,
                                     GSS_C_NO_OID,
                                     GSS_C_NO_OID_SET,
                                     GSS_C_NO_BUFFER_SET,
/*                                     &oid_set, */
/*                                     &oid_buffers, */
                                     token_ptr,
                                     0,
                                     0,
                                     &send_tok);


  if(init_maj_stat != GSS_S_COMPLETE &&
    init_maj_stat != GSS_S_CONTINUE_NEEDED)
  {
     globus_gss_assist_display_status_str(&error_str,
                                          NULL,
                                          init_maj_stat,
                                          min_stat,
                                          0);
printf("\nLINE %d ERROR: %s\n\n", __LINE__, error_str);
     globus_print_error((globus_result_t) min_stat);
     exit(1);
  }
```



An example C GSS-API usage

```
while(1)
 {
    accept_maj_stat=gss_accept_delegation(&min_stat,
                                accept_context,
                                GSS_C_NO_OID_SET,
                                GSS_C_NO_BUFFER_SET,
                                &send_tok,
                                0,
                                0,
                                &time_rec,
                                &delegated_cred,
                                &mech_type,
                                &recv_tok);

    if(accept_maj_stat != GSS_S_COMPLETE &&
      accept_maj_stat != GSS_S_CONTINUE_NEEDED)
    {
       globus_gss_assist_display_status_str(&error_str,
                                NULL,
                                init_maj_stat,
                                min_stat,
                                0);
       printf("\nLINE %d ERROR: %s\n\n", __LINE__, error_str);
globus_print_error((globus_result_t) min_stat);
       exit(1);
    }
    else if(accept_maj_stat == GSS_S_COMPLETE)
    {
       break;
    }
```



An example C GSS-API usage

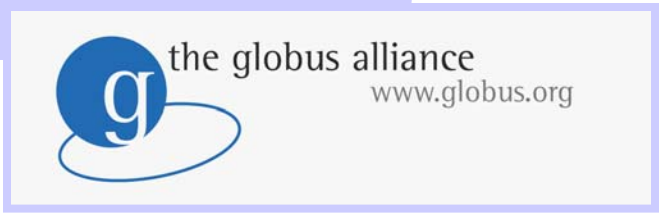
```
init_maj_stat = gss_init_delegation(&min_stat,
                                    init_context,
                                    cred_handle,
                                    GSS_C_NO_OID,
                                    GSS_C_NO_OID_SET,
                                    GSS_C_NO_BUFFER_SET,
                                    &recv_tok,
                                    0,
                                    0,
                                    &send_tok);
if(init_maj_stat != GSS_S_COMPLETE &&
       init_maj_stat != GSS_S_CONTINUE_NEEDED)
    {
       globus_gss_assist_display_status_str(&error_str,
                                    NULL,
                                    init_maj_stat,
                                    min_stat,
                                    0);
       printf("\nLINE %d ERROR: %s\n\n", __LINE__, error_str);
       globus_print_error((globus_result_t) min_stat);
       exit(1);
    }
}

printf("%s:%d: Successfully delegated credential\n",
      __FILE__,
      __LINE__);
```

Credentials delegated

An example C GSS-API usage

```
/* export and import the delegated credential */
/* this can be done both to a buffer and to a file */

maj_stat = gss_export_cred(&min_stat,
                           delegated_cred,
                           GSS_C_NO_OID,
                           0,
                           &send_tok);

if(maj_stat != GSS_S_COMPLETE)
{
   globus_gss_assist_display_status_str(&error_str,
                                        NULL,
                                        init_maj_stat,
                                        min_stat,
                                        0);
   printf("\nLINE %d ERROR: %s\n\n", __LINE__, error_str);
   globus_print_error((globus_result_t) min_stat);
   exit(1);
}
```

To pass delegated credentials from one process to another



An example C GSS-API usage

```
maj_stat = gss_import_cred(&min_stat,
                           &imported_cred,
                           GSS_C_NO_OID,
                           0,
                           &send_tok,
                           0,
                           &time_rec);


if(maj_stat != GSS_S_COMPLETE)
{
   globus_gss_assist_display_status_str(&error_str,
                                        NULL,
                                        init_maj_stat,
                                        min_stat,
                                        0);
   printf("\nLINE %d ERROR: %s\n\n", __LINE__, error_str);
   globus_print_error((globus_result_t) min_stat);
   exit(1);
}

printf("%s:%d: Successfully exported/imported the delegated
credential\n",
       __FILE__,
       __LINE__);
```

To pass delegated credentials from one process to another

the globus alliance
www.globus.org

http://www.gridforum.org/documents/GWD-I-E/GFD-E.024.pdf

An example C GSS-API usage

```
/* set up another security context using the delegated credential */

init_maj_stat = gss_init_sec_context(&min_stat,
                                     imported_cred,
                                     &del_init_context,
                                     target_name,
                                     GSS_C_NULL_OID,
                                     0,
                                     0,
                                     GSS_C_NO_CHANNEL_BINDINGS,
                                     token_ptr,
                                     NULL,
                                     &send_tok,
                                     NULL,
                                     NULL);


if(init_maj_stat != GSS_S_COMPLETE &&
  init_maj_stat != GSS_S_CONTINUE_NEEDED)
{
   globus_gss_assist_display_status_str(&error_str,
                                        NULL,
                                        init_maj_stat,
                                        min_stat,
                                        0);
   printf("\nLINE %d ERROR: %s\n\n", __LINE__, error_str);
   globus_print_error((globus_result_t) min_stat);
   exit(1);
}
```

Establish a second
Security context
Using delegated credentials

An example C GSS-API usage

```
 while(1)
 {
    accept_maj_stat=gss_accept_sec_context(&min_stat,
                                &del_accept_context,
                                imported_cred,
                                &send_tok,
                                GSS_C_NO_CHANNEL_BINDINGS,
                                &target_name,
                                &mech_type,
                                &recv_tok,
&ret_flags,
                                /* ignore time_rec */
                                NULL,
                                GSS_C_NO_CREDENTIAL);

    if(accept_maj_stat != GSS_S_COMPLETE &&
      accept_maj_stat != GSS_S_CONTINUE_NEEDED)
    {
       globus_gss_assist_display_status_str(&error_str,
                                NULL,
                                init_maj_stat,
                                min_stat,
                                0);
       printf("\nLINE %d ERROR: %s\n\n", __LINE__, error_str);
       globus_print_error((globus_result_t) min_stat);
       exit(1);
    }
    else if(accept_maj_stat == GSS_S_COMPLETE)
    {
       break;
    }
```

Establish a second Security context Using delegated credentials

An example C GSS-API usage

```
    init_maj_stat = gss_init_sec_context(&min_stat,
                                         imported_cred,
                                         &del_init_context,
                                         target_name,
                                         GSS_C_NULL_OID,
                                         0,
                                         0,
                                         GSS_C_NO_CHANNEL_BINDINGS,
                                         &recv_tok,
                                         NULL,
                                         &send_tok,
                                         NULL,
                                         NULL);


      if(init_maj_stat != GSS_S_COMPLETE &&
        init_maj_stat != GSS_S_CONTINUE_NEEDED)
      {
         globus_gss_assist_display_status_str(&error_str,
                                              NULL,
                                              init_maj_stat,
                                              min_stat,
                                              0);
printf("\nLINE %d ERROR: %s\n\n", __LINE__, error_str);
         globus_print_error((globus_result_t) min_stat);
         exit(1);
      }
   }

   /* got sec context based on delegated cred now */

   printf("%s:%d: Successfully established security context with delegated
credential\n",
          __FILE__,
          __LINE__);
```

Establish a second
Security context
Using delegated credentials

An example C GSS-API usage

```
% cat compile_globus_sec

#!/bin/sh
CC= /opt/gcc-3.2.2/bin/gcc
GLOBUS_LOCATION=/opt/globus
GLOBUS_FLAVOR=gcc32dbg
$CC –I${GLOBUS_LOCATION}/include/${GLOBUS_FLAVOR} ${1}.c \
                                    -L${GLOBUS_LOCATION}/lib \
                                    -lglobus_gssapi_gsi_${GLOBUS_FLAVOR} \
                                    -lglobus_gss_assist_${GLOBUS_FLAVOR} -o ${1}

% ./compile_globus_sec acquire_credential

OR


% cat compile_globus_sec

#!/bin/sh
GLOBUS_LOCATION=/opt/globus
GLOBUS_FLAVOR=gcc32dbg
CC= /opt/gcc-3.2.2/bin/gcc
#
$GLOBUS_LOCATION/bin/globus-makefile-header -flavor $GLOBUS_FLAVOR
globus_gss_assist > globus_make_header
#
. ./globus_make_header
$CC  ${GLOBUS_INCLUDES} ${1}.c ${GLOBUS_LDFLAGS} \
                                    ${GLOBUS_PKG_LIBS} -o ${1}
```

Compiling and Linking

Further Information

Enabling Grids for E-science in Europe

Grid

- LCG Security: http://proj-lcg-security.web.cern.ch/proj-lcg-security/
- Globus Security: http://www.globus.org/security/

Background

- GGF Security: http://www.gridforum.org/security/
- GSS-API: http://www.faqs.org/faqs/kerberos-faq/general/section-84.html
- IETF PKIX charter: http://www.ietf.org/html.charters/pkix-charter.html
- PKCS: http://www.rsasecurity.com/rsalabs/pkcs/index.html