Authorisation and Authentication in gLite

Mike Mineter
National e-Science Centre, Edinburgh

CERN -- EGEE tutorial
27-28 February 2006

Information Society

Acknowledgements

- **Emidio Giorgio, INFN/University of Catania**
- **Additional material from**
  - Richard Sinnott, University of Glasgow http://csperkins.org/teaching/grid/lecture09.pdf

Note – additional information is in hidden slides of this presentation

How does EGEE…

- **How does EGEE build dynamic distributed systems?**
  - For many international collaborations (“virtual organisations”)
  - With n,000 processors and m,000 users in hundreds of independent sites (“administrative domains”)
  - With no prior direct relationship between users and resource providers
  - In a world where public networks are abused by hackers, etc.

1. **Authentication - communication of identity**

   Basis for

   - Message integrity - so tampering is recognised
   - Message confidentiality, if needed - so sender and receiver only can understand the message
   - Non-repudiation: knowing who did what when – can’t deny it
2. **Authorisation - once identity is known, what can a user do?**
3. **Delegation- A allows B to act on behalf of A**

Overview

- **Encryption**
  - Symmetric algorithms
  - Asymmetric algorithms
- **Certificates**
  - Digital Signatures
  - X509 certificates
- **Grid Security**
  - Grid Security Infrastructure
  - Proxy certificates
  - “MyProxy”
- **Virtual Organisations and Authorisation**
  - Concepts
  - VOMS – “2nd generation” approach to authorization

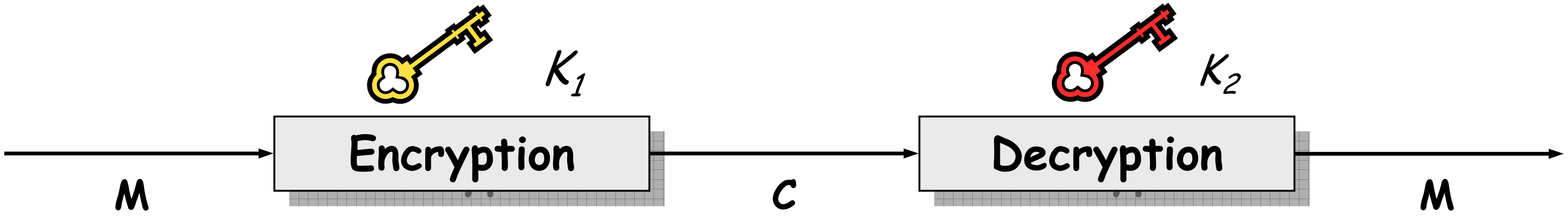
Glossary

- **Principal**
  - An entity: a user, a program, or a machine
- **Credentials**
  - Some data providing a proof of identity
- **Authentication**
  - Verify the identity of a principal
- **Authorization**
  - Map an entity to some set of privileges
- **Confidentiality**
  - Encrypt the message so that only the recipient can understand it
- **Integrity**
  - Ensure that the message has not been altered in the transmission
- **Non-repudiation**
  - Impossibility of denying the authenticity of a digital signature

Overview

- **Encryption**
  - Symmetric algorithms
  - Asymmetric algorithms

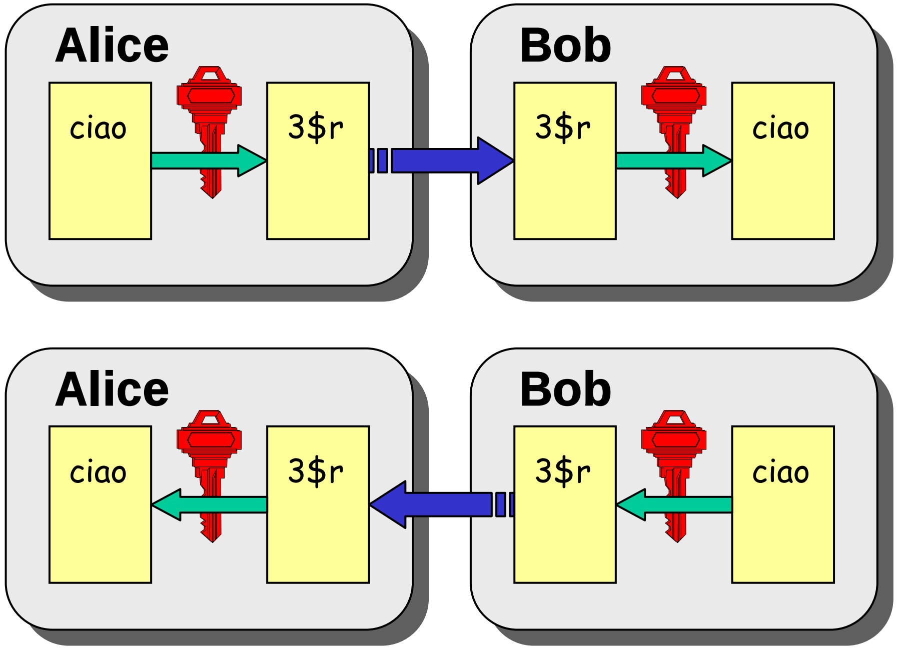
Cryptography: “hidden writing”



- **Mathematical algorithms that provide important building blocks for the implementation of a security infrastructure**
- **Symbology**
  - Plain text: *M*
  - Encrypted text: *C*
  - Encryption with key K_1 : $E_{K_1}(M) = C$
  - Decryption with key K_2 : $D_{K_2}(C) = M$
- **Algorithms**
  - **Symmetric**: $K_1 = K_2$
  - **Asymmetric**: $K_1 \neq K_2$

Symmetric Algorithms

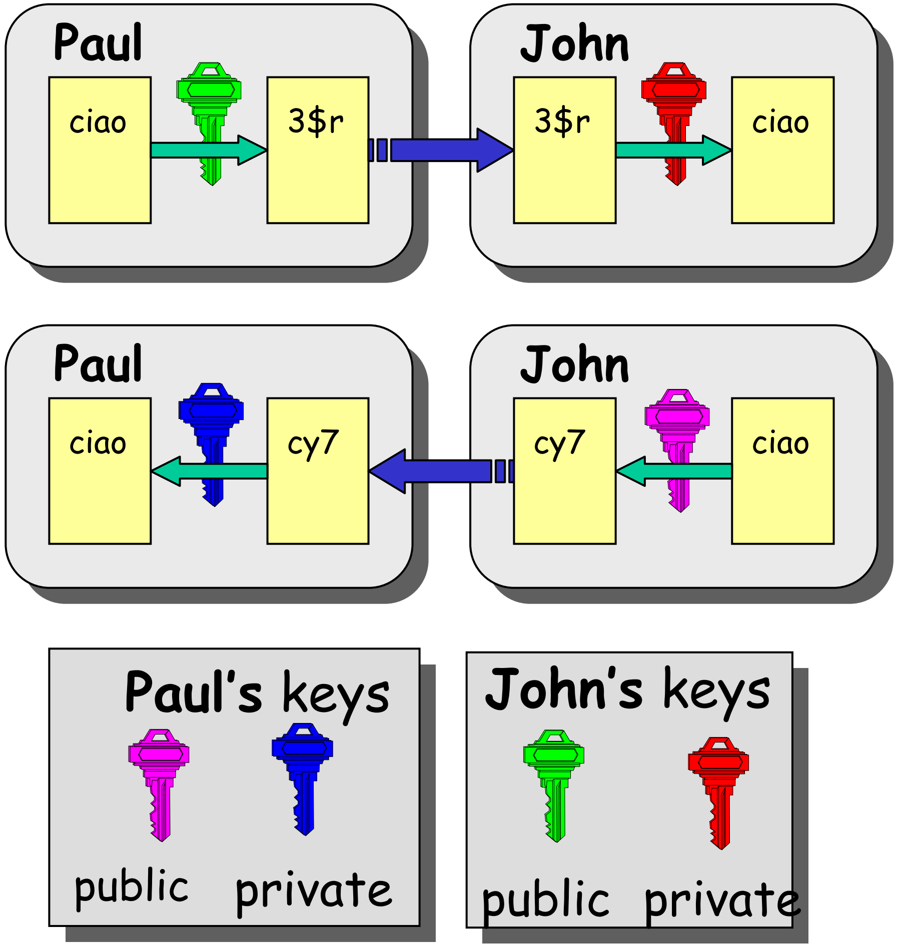
- **The** same **key is used for encryption and decryption**



- **Disadvantages:**
  - how to distribute the keys?
  - the number of keys is O(n^2)
  - n: number of people

Public Key Algorithms

- **Every user has two keys: one *private* and one *public*:**
  - it is *impossible* to derive the private key from the public one;
  - a message encrypted by one key can be decrypted **only** by the other one.
- **Public keys are exchanged**
- **The sender encrypts using the *public* key of the receiver**
- **The receiver decrypts using his *private* key;**
- **The number of keys is O(n)**



Overview

- **Encryption**
  - Symmetric algorithms
  - Asymmetric algorithms: PKI
- **Certificates**
  - Digital Signatures
  - X509 certificates

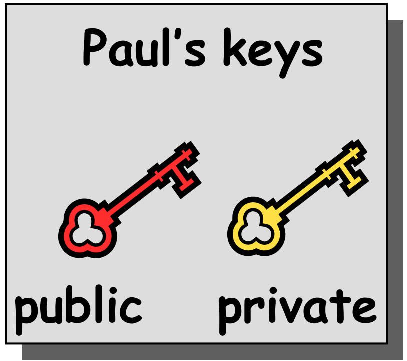
One-Way Hash Functions

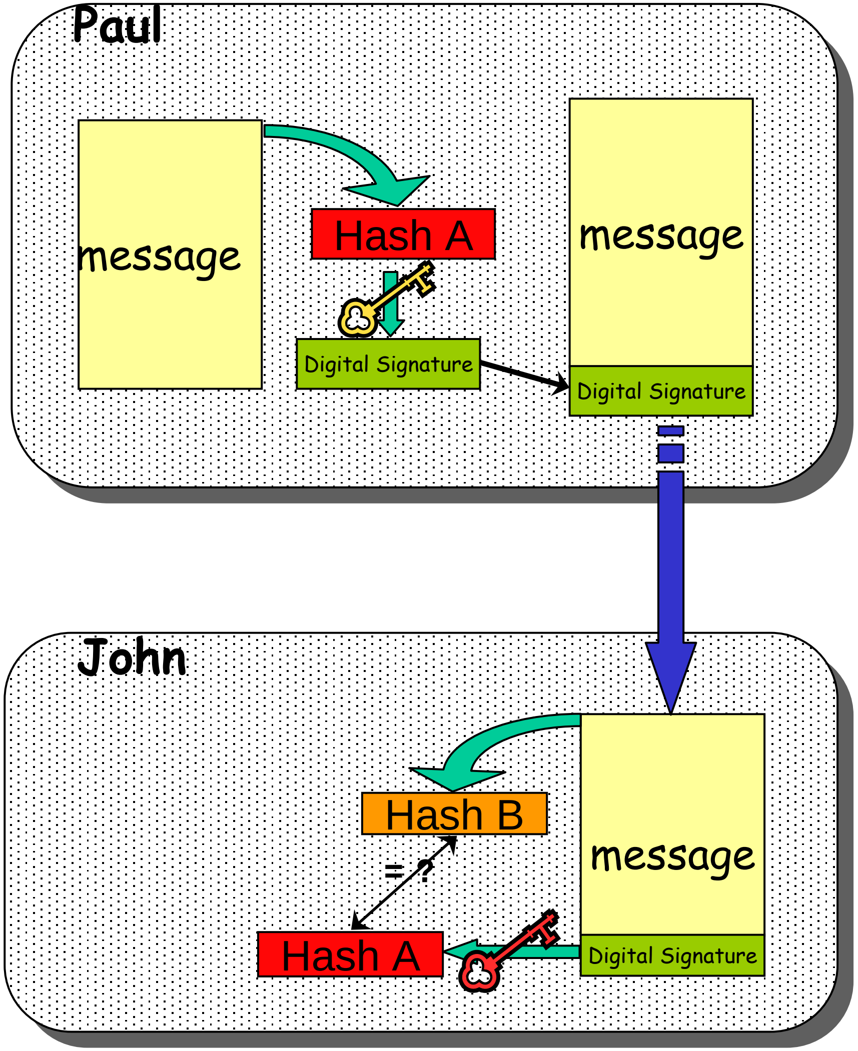
- **Functions (*H*) that given as input a variable-length message (*M*) produce as output a string of fixed length (*h*)**

1. given *M*, it **must be easy** to calculate $h = H(M)$
2. given *h*, it **must be difficult** to calculate $M = H^{-1}(h)$
3. given *M*, it **must be difficult** to find *M'* such that $H(M) = H(M')$
   i.e. hash is unlikely to be identical for different messages

Digital Signature

- Paul **calculates the *hash* of the message**
- Paul **encrypts the hash using his *private* key: the encrypted hash is the *digital signature*.**
- Paul **sends the signed message to** John.
- John **calculates the hash of the message**
- **Decrypts A with Paul's *public* key.**
- **If hashes equal:**
  1. message wasn't modified;
  2. hash B is from Paul's private key





Digital Certificates

- **Paul's digital signature is useful to John if:**
  1. Paul's private key is not compromised – keep these safe!!!
  2. John has Paul's public key

- **How can John be sure that Paul's public key is really Paul's public key and not someone else's?**
  - A *third party* establishes the correspondence between public key and owner's identity.
  - Both John and Paul trust this third party

The "third party" is called a *Certification Authority* (CA).

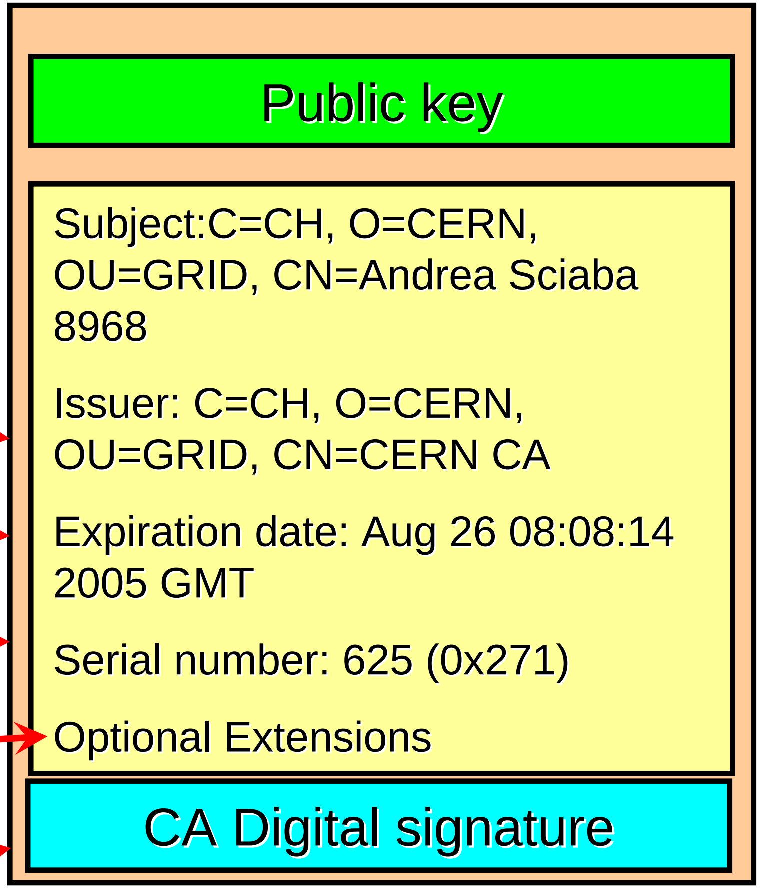
Certificate Authority

- **Issues Digital Certificates for users, programs and machines**
  - Combines public key + owner information
  - Signed by CA using its private certificate
  - Can use the CA’s public certificate to check integrity of certificates
- **CA’s check the identity and the personal data of the requestor of a certificate**
  - Registration Authorities (RAs) do the actual validation
- **CA’s periodically publish a list of compromised certificates**
  - **Certificate Revocation Lists** (CRL): contain all the revoked certificates yet to expire
- **CA’s own certificates are self-signed**

X.509 Certificates

- **An X.509 Certificate contains:**
  - owner's public key;
  - identity of the owner;
  - info on the CA;
  - time of validity;
  - Serial number;
  - Optional extensions
  - digital signature of the CA

Structure of a X.509 certificate



Public key

Subject:C=CH, O=CERN, OU=GRID, CN=Andrea Sciaba 8968

Issuer: C=CH, O=CERN, OU=GRID, CN=CERN CA

Expiration date: Aug 26 08:08:14 2005 GMT

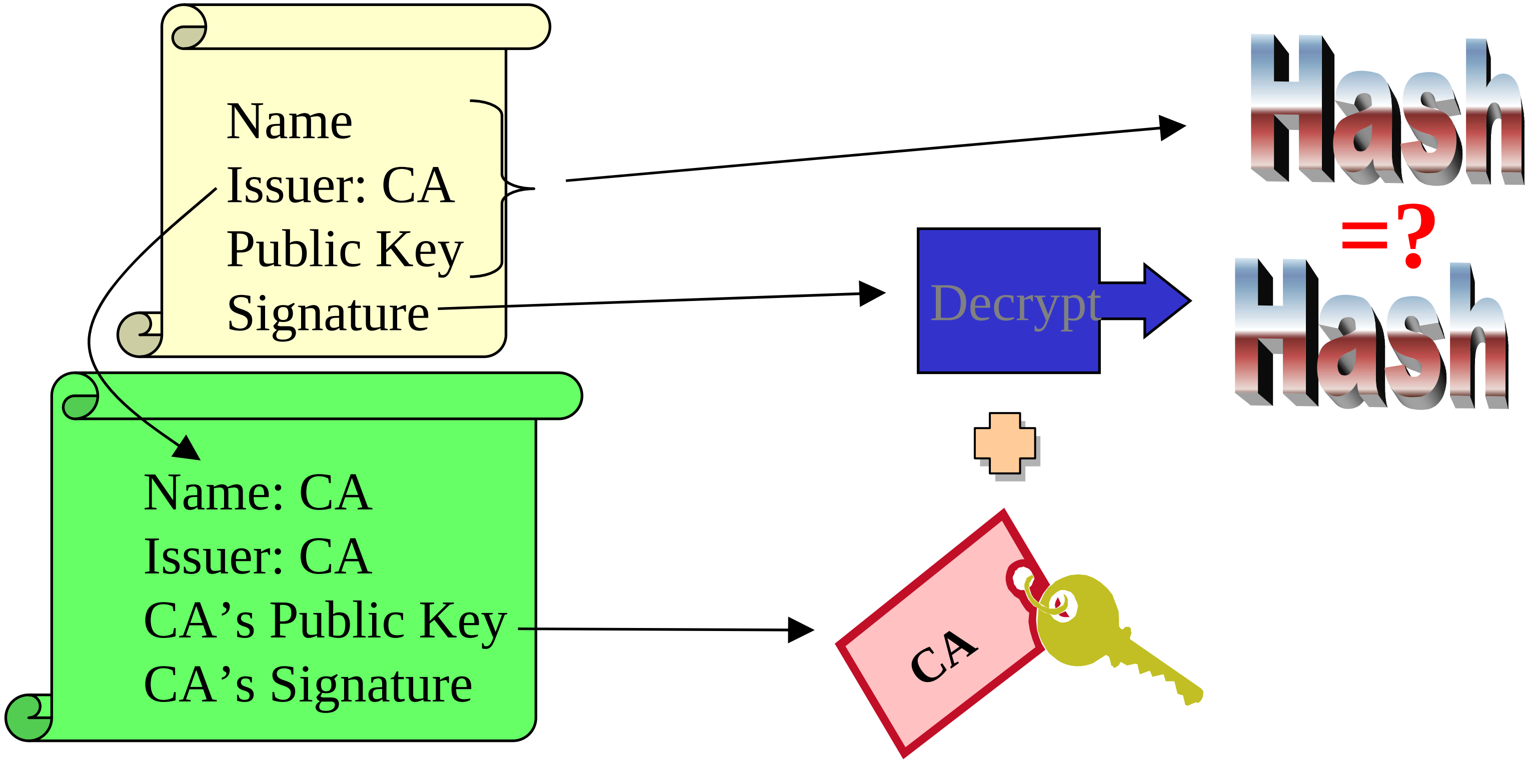
Serial number: 625 (0x271)

Optional Extensions

CA Digital signature

Certificate Validity

- The public key from the CA certificate can then be used to verify the certificate.



Name
Issuer: CA
Public Key
Signature

Name: CA
Issuer: CA
CA's Public Key
CA's Signature

slide based on presentation given by Carl Kesselman at GGF Summer School 2004

VERY IMPORTANT

Private keys must be stored only:

in ***protected*** places

AND

in ***encrypted*** form

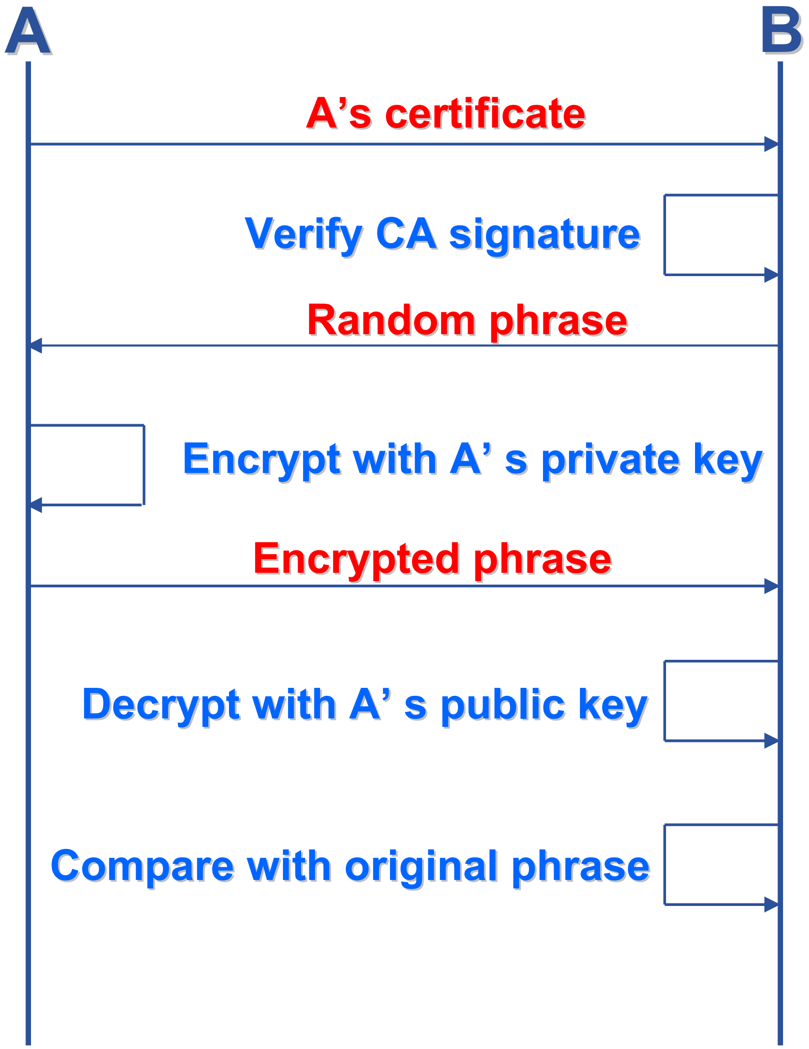
Overview

- **Encryption**
  - Symmetric algorithms
  - Asymmetric algorithms: PKI
- **Certificates**
  - Digital Signatures
  - X509 certificates
- **Grid Security Infrastructure**

The Grid Security Infrastructure (GSI)

Based on X.509 PKI:

- **every user/host/service has an X.509 certificate;**
- **certificates are signed by trusted (by the local sites) CA's;**
- **every Grid transaction is mutually authenticated:**
  1. A sends his certificate;
  2. B verifies signature in A's certificate using CA public certificate;
  3. B sends to A a challenge string;
  4. A encrypts the challenge string with his private key;
  5. A sends encrypted challenge to B
  6. B uses A's public key to decrypt the challenge.
  7. B compares the decrypted string with the original challenge
  8. If they match, B verified A's identity and A can not repudiate it.


A
B
A's certificate
Verify CA signature
Random phrase
Encrypt with A' s private key
Encrypted phrase
Decrypt with A' s public key
Compare with original phrase

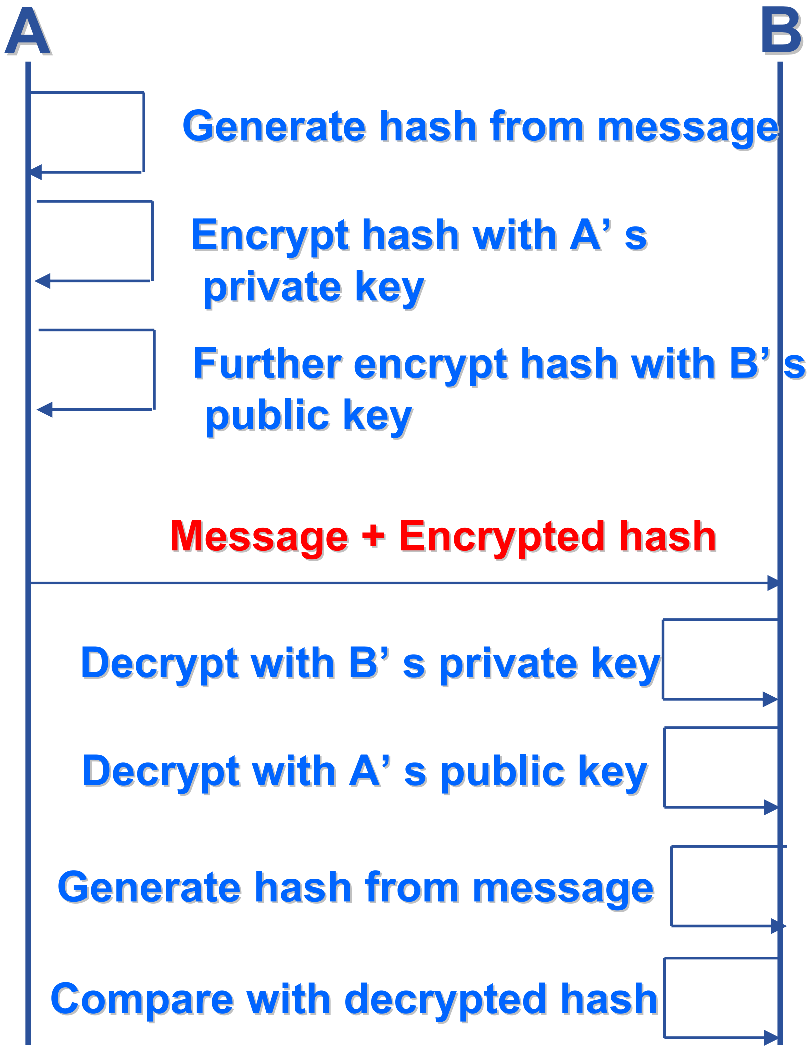
The Grid Security Infrastructure (GSI) - continued

After A and B authenticated each other,
for A to send a message to B:

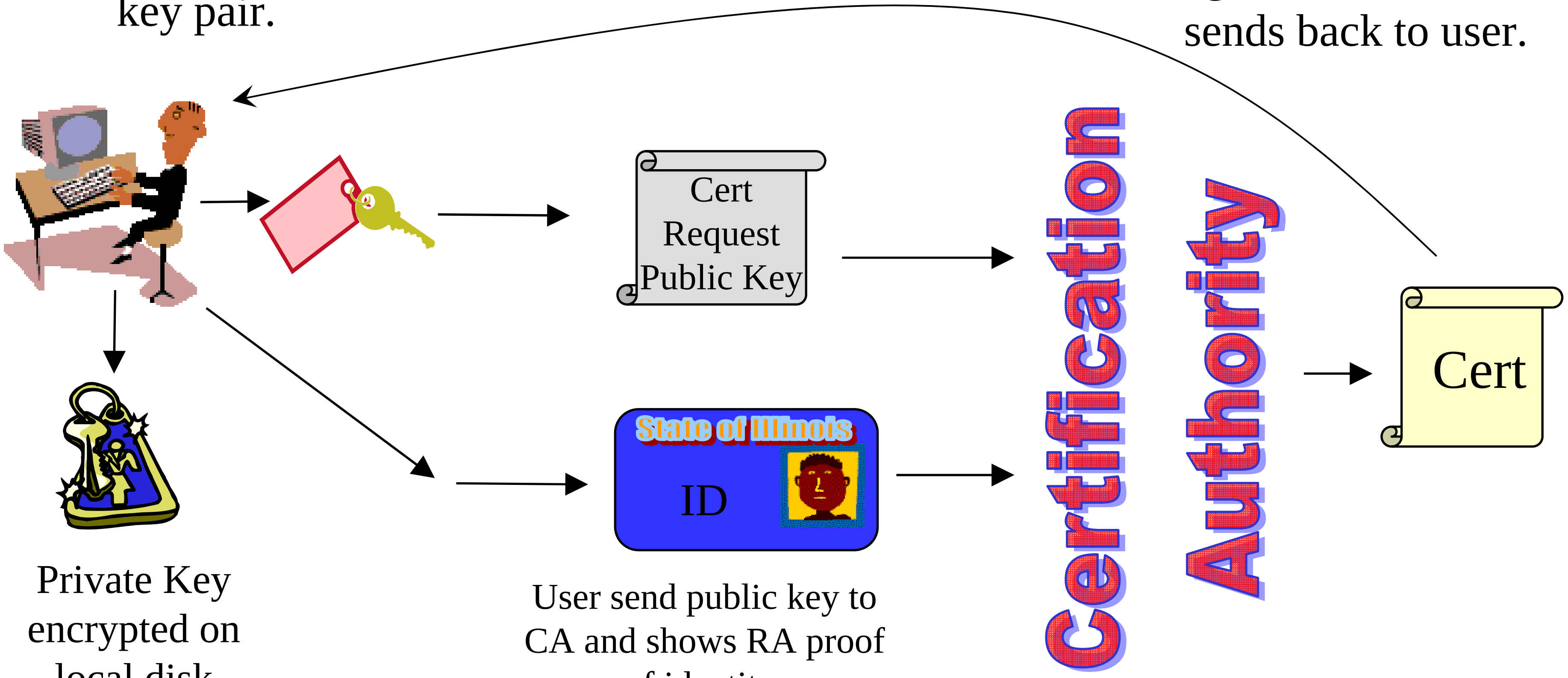
- **Default: message integrity checking**
  - Not private – a test for tampering

.

- **For private communication:**
  - Encrypt all the message (not just hash) - Slower


A
B
Generate hash from message
Encrypt hash with A’ s private key
Further encrypt hash with B’ s public key
Message + Encrypted hash
Decrypt with B’ s private key
Decrypt with A’ s public key
Generate hash from message
Compare with decrypted hash

Certificate Request



User generates public/private key pair.

CA confirms identity, signs certificate and sends back to user.

Private Key encrypted on local disk

User send public key to CA and shows RA proof of identity.

International agreement

- **X 509 Digital certificate is the basis of Authentication in EGEE**
- ***Certification Authorities* (CAs)**
  - ~one per country
  - each builds network of “Registration Authorities” who issue certificates
- ***CAs are mutually recognized* – to enable international collaboration**
- **International Grid Trust Federation http://www.gridpma.org/**

Certificate request… more details

- **EGEE/LCG recognizes a given set of CAs**
  - https://lcg-registrar.cern.ch/pki_certificates.html
- **How you request a certificate depends on your CA**

- **For GILDA, have a look at the Video Tutorials:**
  - https://gilda.ct.infn.it/video/Certification/Allproxy.html (Flash)
  - https://gilda.ct.infn.it/video/Certification/AllCertproxy.ram (Real)

To use the EGEE grid

- **Get an internationally recognised certificate**
  - From a local RA – you will need to see them personally, bringing passport or other identification
- **Contact the VO manager**
- **Accept the VO and the EGEE conditions of use to register with both EGEE and the VO**
- **Upload your certificate to a "User Interface" machine – a machine that can run the gLite commands**

- **We will be continuing the practical from this stage**
- **We have GILDA certificates on the GILDA testbed**

On the side: certificate management

- **Import your certificate in your browser**
  - If you received a .pem certificate you need to convert it to PKCS12
  - Use *openssl* command line (available in each egee/LCG UI)
    - `openssl pkcs12 –export –in usercert.pem –inkey userkey.pem –out my_cert.p12 –name ’My Name’`

- **GILDA (and other VOs):**
  - You receive already a PKCS12 certificate (can import it directly into the web browser)
  - For future use, you will need *usercert.pem* and *userkey.pem* in a directory ~/.globus on your UI
  - Export the PKCS12 cert to a local dir on UI and use again *openssl:*
    - `openssl pkcs12 -nocerts -in my_cert.p12 -out userkey.pem`
    - `openssl pkcs12 -clcerts -nokeys -in my_cert.p12 -out usercert.pem`

X.509 Proxy Certificate

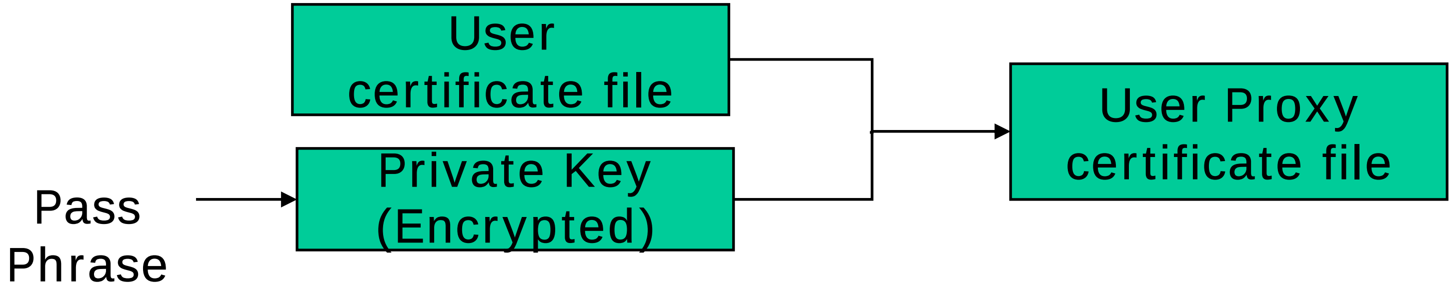
- **GSI extension to X.509 Identity Certificates**
  - signed by the normal end entity cert (or by another proxy).
- **Enables single sign-on**
- **Support some important features**
  - Delegation
  - Mutual authentication
- **Has a limited lifetime (minimized risk of “compromised credentials”)**
- **It is created by the** grid-proxy-init **command:**

```
% grid-proxy-init
Enter PEM pass phrase: ******
```

  - Options for grid-proxy-init:
    - -hours <lifetime of credential>
    - -bits <length of key>
    - -help

grid-proxy-init

- **User enters pass phrase, which is used to decrypt private key.**
- **Private key is used to sign a proxy certificate with <u>its own</u>, new public/private key pair.**
  - User's private key not exposed after proxy has been signed



- **Proxy placed in /tmp**
  - the private key of the Proxy is *not* encrypted:
  - stored in local file: must be readable **only** by the owner;
  - proxy lifetime is short (typically 12 h) to minimize security risks.
- **NOTE: *No* network traffic!**

Proxy again …

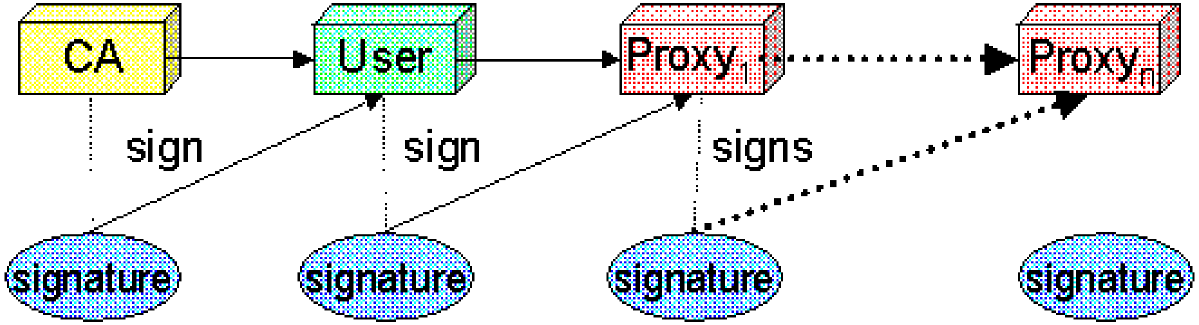
- **grid-proxy-init ≡ "login to the Grid"**
- **To "logout" you have to destroy your proxy:**
  - `grid-proxy-destroy`
  - This does *NOT* destroy any proxies that were delegated from this proxy.
  - You cannot revoke a remote proxy
  - Usually create proxies with short lifetimes
- **To gather information about your proxy:**
  - `grid-proxy-info`
  - Options for printing proxy information

-subject	-issuer
-type	-timeleft
-strength	-help

Grid Security Infrastructure - proxies

- **To support….**
  - Single sign-on: to a machine on which your certificate is held
  - Delegation: a service can act on behalf of a person

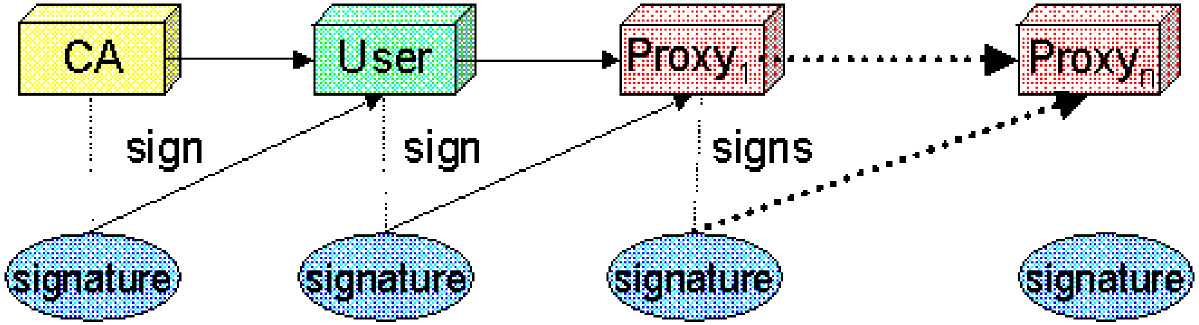
- **….GSI introduces proxy certificates**
  - Short-lived certificates signed with the user’s certificate or a proxy
  - Reduces security risk, enables delegation
- **New key pair generated remotely on server**
  - Client signs proxy cert and returns it



- Each service decides whether it will allow authentication with a limited proxy
- Job manager service requires a full proxy
- GridFTP server allows either full or limited proxy to be used

Delegation and limited proxy

- **Delegation - allows remote process to authenticate on behalf of the user**
  - Remote process "impersonates" the user
- **Achieve by creation of next-level proxy from a proxy**
  - New key pair generated remotely on server
  - Client signs proxy cert and returns it
- **The client can elect to delegate a "limited proxy"**
  - Each service decides whether it will allow authentication with a limited proxy
  - Job manager service requires a full proxy
  - GridFTP server allows either full or limited proxy to be used



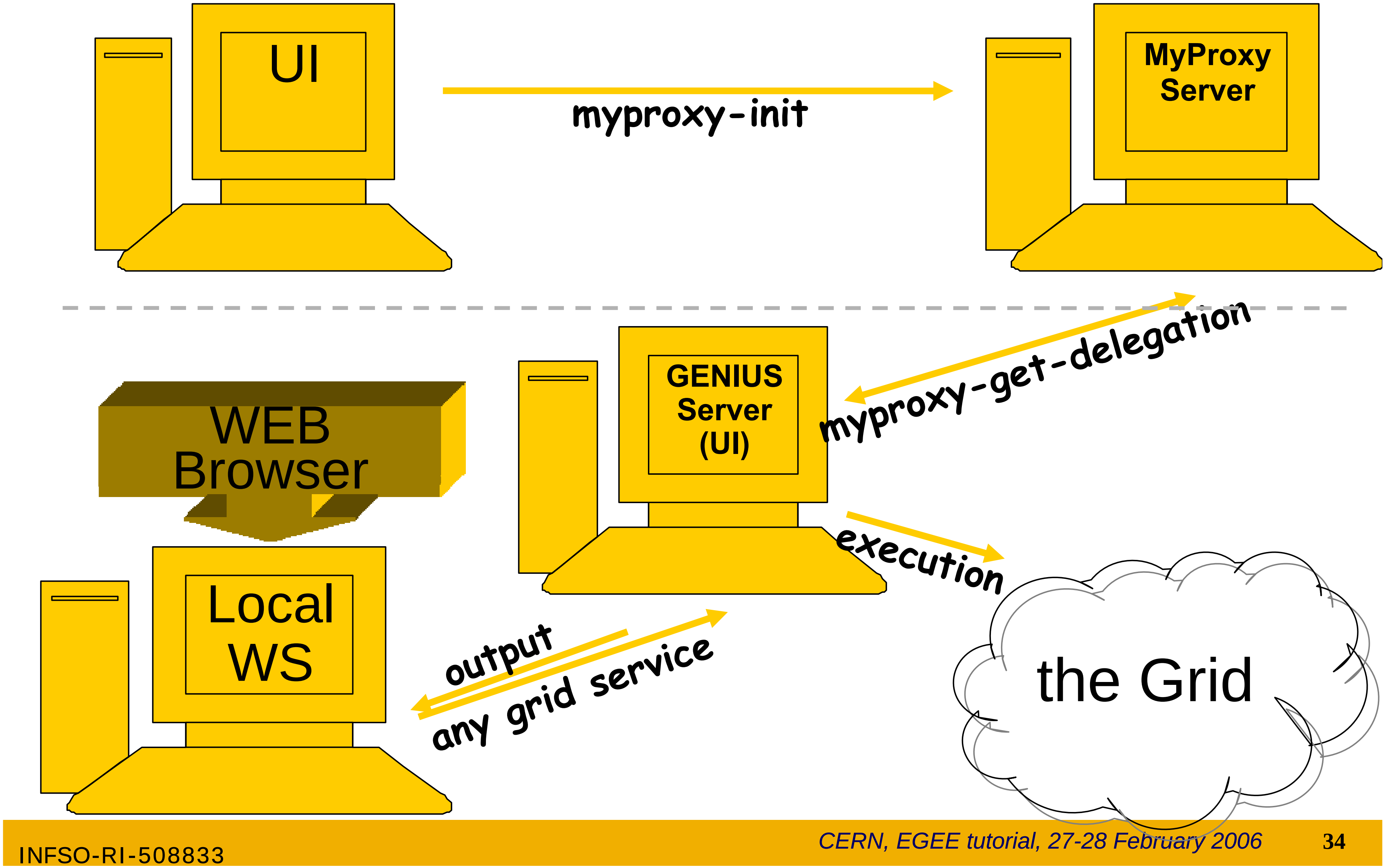
"MyProxy"

- **You may need:**
  - To interact with a grid from many machines
    - And you realise that you must NOT, EVER leave your certificate where anyone can find and use it….
  - To use a portal, and delegate to the portal the right to act on your behalf (First step is for the portal to make a proxy certificate for you)
  - To run jobs that might last longer than the lifetime of a short-lived proxy
- **Solution: you can store a long-lived proxy in a "MyProxy repository" and derive a proxy certificate when needed.**

Long term proxy

- **Proxy has limited lifetime (default is 12 h)**
  - Bad idea to have longer proxy
- **However, a grid task might need to use a proxy for a much longer time**
  - Grid jobs in HEP Data Challenges on LCG last up to 2 days
- **myproxy server:**
  - Allows to create and store a long term proxy certificate:
  - myproxy-init -s <host_name>
    - -s: <host_name> specifies the hostname of the myproxy server
  - myproxy-info
    - Get information about stored long living proxy
  - myproxy-get-delegation
    - Get a new proxy from the MyProxy server
  - myproxy-destroy
  - Check out the myproxy-xxx - - help option
- **A dedicated service on the RB can renew automatically the proxy**
- **File transfer services in gLite validate user request and eventually renew proxies**
  - contacting myproxy server

Grid authentication with MyProxy



Overview

- **Encryption**
  - Symmetric algorithms
  - Asymmetric algorithms
- **Certificates**
  - Digital Signatures
  - X509 certificates
- **Grid Security**
  - Grid Security Infrastructure
  - Proxy certificates
  - MyProxy
- **Virtual Organisations and Authorisation**

Building on GSI

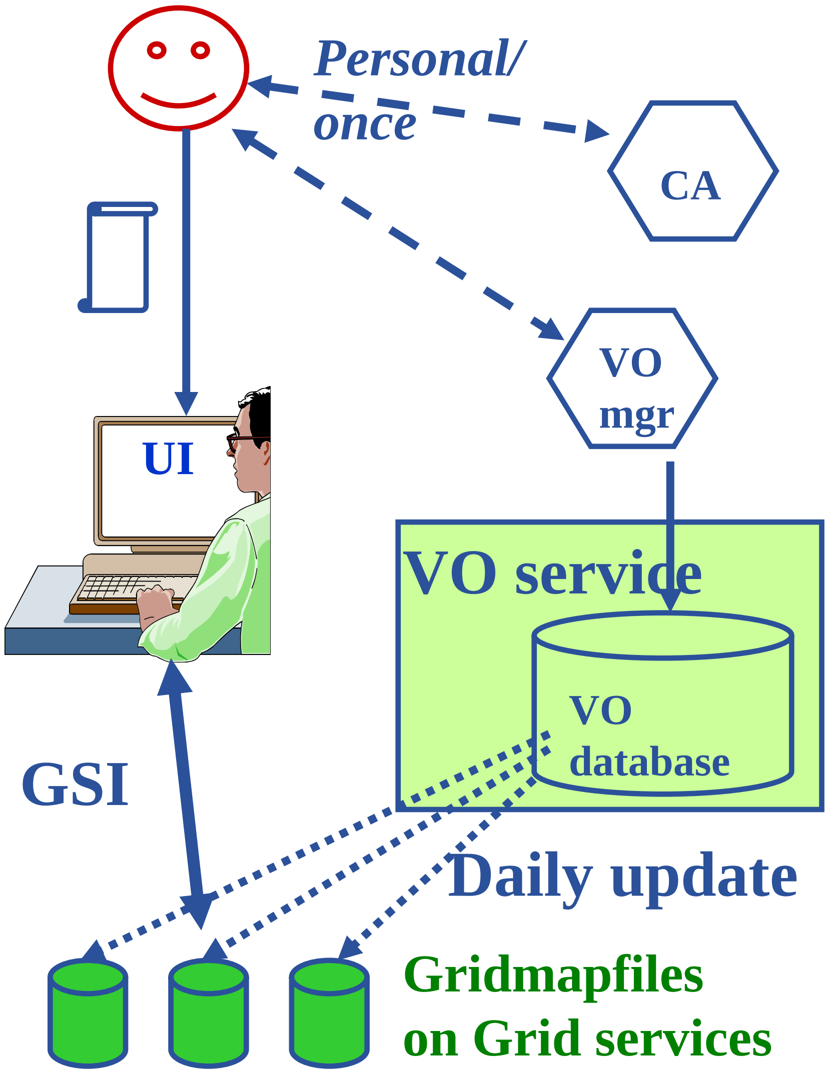
- **Grid users MUST belong to virtual organizations**
  - Sets of users belonging to a collaboration
  - User must sign the usage guidelines for the VO

- **Authorisation**
  - What are you allowed to do?
  - … and how is this controlled??

- **In EGEE the answer is VOMS**
  - Virtual Organisation Management System
  - “second generation” of VO management

Authentication, Authorisation: pre-VOMS

- **Authentication**
  - User receives certificate signed by CA
  - Connects to “UI” by ssh
  - Downloads certificate
  - **Single logon to Grid** – create proxy - then **Grid Security Infrastructure identifies user to other machines**
- **Authorisation**
  - User joins Virtual Organisation
  - VO negotiates access to Grid nodes and resources
  - Authorisation tested by CE
  - gridmapfile maps user to local account



Virtual Organizations and authorization

- **Grid users MUST belong to virtual organizations**
  - Sets of users belonging to a collaboration
  - User must sign the usage guidelines for the VO
  - You will be registered in the VO-LDAP server (wait for notification)
  - List of supported vos:
    - https://lcg-registrar.cern.ch/virtual_organization.html
- **Vos maintained a list of their members on a LDAP Server**
  - The list is downloaded by grid machines to map user certificate subjects to local “pool” accounts
  -

```
...
"/C=CH/O=CERN/OU=GRID/CN=Simone Campana 7461" .dteam
"/C=CH/O=CERN/OU=GRID/CN=Andrea Sciaba 8968" .cms
"/C=CH/O=CERN/OU=GRID/CN=Patricia Mendez Lorenzo-ALICE" .alice
...
```

`/etc/grid-security/grid-mapfile`

Evolution of VO management

Before VOMS

- **User is authorised as a member of a single VO**
- **All VO members have same rights**
- **Gridmapfiles are updated by VO management software: map the user's DN to a local account**
- **grid-proxy-init – derives proxy from certificate – the "sign-on to the grid"**

VOMS

- **User can be in multiple VOs**
  - Aggregate rights
- **VO can have groups**
  - Different rights for each
    - Different groups of experimentalists
    - …
  - Nested groups
- **VO has roles**
  - Assigned to specific purposes
    - E,g. system admin
    - When assume this role
- **Proxy certificate carries the additional attributes**
- **voms-proxy-init**

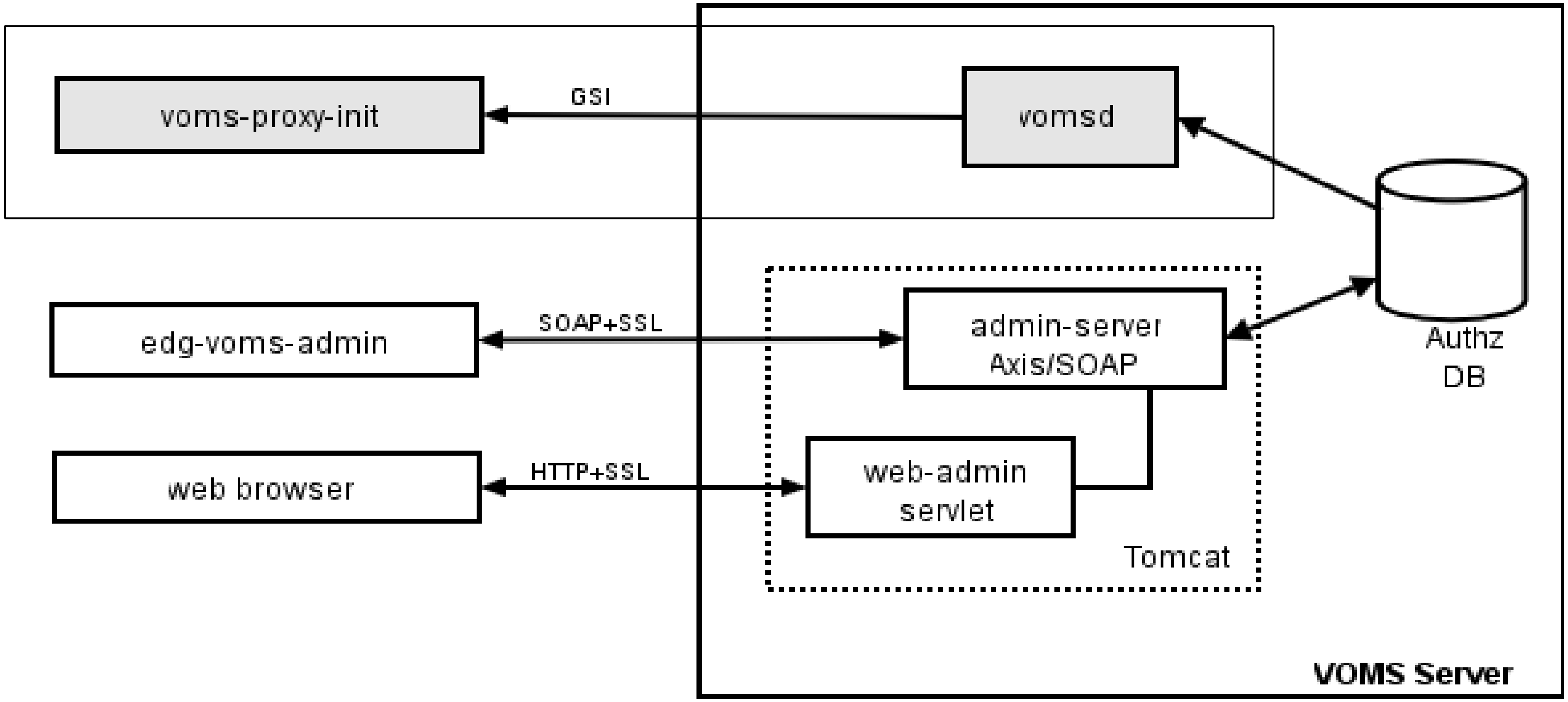
VOMS – now in both the production (LCG) and pre-production (gLite) middleware

VOMS : concepts

- **Virtual Organization Membership Service**
  - Extends the proxy with info on VO membership, group, roles
  - Fully compatible with Globus Toolkit
  - Each VO has a database containing group membership, roles and capabilities informations for each user
  - User contacts voms server requesting his authorization info
  - Server send authorization info to the client, which includes them in a proxy certificate

```
[glite-tutor] /home/giorgio > voms-proxy-init --voms gilda
Cannot find file or dir: /home/giorgio/.glite/vomses
Your identity: /C=IT/O=GILDA/OU=Personal
Certificate/L=INFN/CN=Emidio
Giorgio/Email=emidio.giorgio@ct.infn.it
Enter GRID pass phrase:
Your proxy is valid until Mon Jan 30 23:35:51 2006
Creating temporary
proxy.................................Done
Contacting  voms.ct.infn.it:15001
[/C=IT/O=GILDA/OU=Host/L=INFN
Catania/CN=voms.ct.infn.it/Email=emidio.giorgio@ct.infn.it
] "gilda"
Creating proxy ............................... Done
Your proxy is valid until Mon Jan 30 23:35:51 2006
```

VOMS - components



- Authz DB is a RDBMS (currently MySQL and Oracle are supported).

FQAN and AC

- short for Fully Qualified Attribute Name, used by VOMS to express membership and other authorization info
- Groups membership, roles and capabilities may be expressed in a format that bounds them together
  <group>/Role=[<role>][/Capability=<capability>]

```
[glite-tutor] /home/giorgio > voms-proxy-info -fqan
/gilda/Role=NULL/Capability=NULL
/gilda/tutors/Role=NULL/Capability=NULL
```

- FQAN are included in an Attribute Certificate
- Attribute Certificates are used to bind a set of attributes (like membership, roles, authorization info etc) with an identity
- AC are digitally signed
- VOMS uses AC to include the attributes of a user in a proxy certificate

VOMS and AC

- Server creates and sign an AC containing the FQAN requested by the user, if applicable
- **AC is included by the client in a well-defined, non critical, extension assuring compatibility with GT-based mechanism**
- At resources level, authorization info are extracted from the proxy and processed by LCAS and LCMAPS

```
/home/giorgio > voms-proxy-info -all
subject    : /C=IT/O=GILDA/OU=Personal Certificate/L=INFN/CN=Emidio
Giorgio/Email=emidio.giorgio@ct.infn.it/CN=proxy
issuer     : /C=IT/O=GILDA/OU=Personal Certificate/L=INFN/CN=Emidio
Giorgio/Email=emidio.giorgio@ct.infn.it
identity   : /C=IT/O=GILDA/OU=Personal Certificate/L=INFN/CN=Emidio
Giorgio/Email=emidio.giorgio@ct.infn.it
type       : proxy
strength   : 512 bits
path       : /tmp/x509up_u513
timeleft   : 11:59:52
=== VO gilda extension information ===
VO         : gilda
subject    : /C=IT/O=GILDA/OU=Personal Certificate/L=INFN/CN=Emidio
Giorgio/Email=emidio.giorgio@ct.infn.it
issuer     : /C=IT/O=GILDA/OU=Host/L=INFN
Catania/CN=voms.ct.infn.it/Email=emidio.giorgio@ct.infn.it
attribute  : /gilda/tutors/Role=NULL/Capability=NULL
attribute  : /gilda/Role=NULL/Capability=NULL
timeleft   : 11:59:45
```

Groups

- The number of users of a VO can be very high:
  - E.g. the experiment ATLAS has 2000 member

- Make VO manageable by organizing users in groups:

  Examples:
  - VO GILDA
    - Group Catania
      - *INFN*
        - Group Barbera
      - *University*
    - Group Padua
  - VO GILDA
    - /GILDA/TUTORS `can write to normal storage`
    - /GILDA/STUDENT `only write to volatile space`

- Groups can have a hierarchical structure, indefinitely deep

Roles

- **Roles are specific roles a user has and that distinguishes him from others in his group:**
  - Software manager
  - VO-Administrator

- **Difference between roles and groups:**
  - Roles have no hierarchical structure – there is no sub-role
  - Roles are not used in ‘normal operation’
    - They are not added to the proxy by default when running *voms-proxy-init*
    - But they can be added to the proxy for special purposes when running *voms-proxy-init*

- **Example:**
  - User Emidio has the following membership
    - VO=gilda, Group=tutors, Role=SoftwareManager
  - During normal operation the role is not taken into account, e.g. Emidio can work as a normal user
  - For special things he can obtain the role “Software Manager”

LCAS & LCMAPS

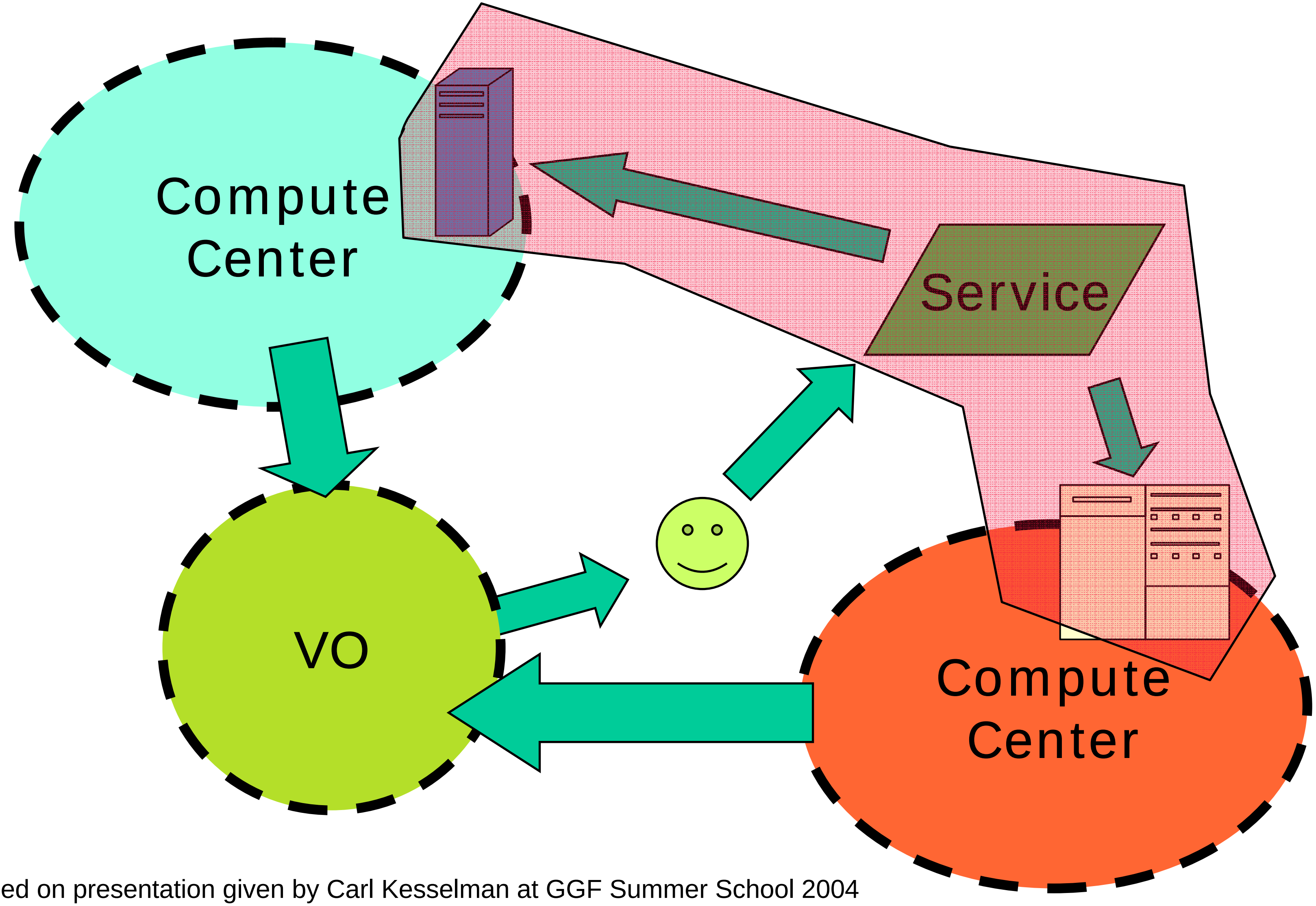
- **Local Centre Authorization Service (LCAS)**
  - Checks if the user is authorized (currently using the grid-mapfile)
  - Checks if the user is banned at the site
  - Checks if at that time the site accepts jobs
- **Local Credential Mapping Service (LCMAPS)**
  - Maps grid credentials to local credentials (eg. UNIX uid/gid, AFS tokens, etc.)
  - Map also VOMS group and roles (full support of FQAN)

```
"/VO=cms/GROUP=/cms"                        .cms
"/VO=cms/GROUP=/cms/prod"                   .cmsprod
"/VO=cms/GROUP=/cms/prod/ROLE=manager"   .cmsprodman
```

GSI environment variables

- **User certificate files:**
  - Certificate: X509_USER_CERT (default: `$HOME/.globus/usercert.pem`)
  - Private key: X509_USER_KEY (default: `$HOME/.globus/userkey.pem`)
  - Proxy: X509_USER_PROXY (default: `/tmp/x509up_u<id>`)
- **Host certificate files:**
  - Certificate X509_HOST_CERT (default: `/etc/grid-security/hostcert.pem`)
  - Private key X509_HOST_KEY (default: `/etc/grid-security/hostkey.pem`)
- **Trusted certification authority certificates:**
  - X509_CERT_DIR (default: `/etc/grid-security/certificates`)
- **Voms server public keys**
  - X509_VOMS_DIR (default: `/etc/grid-security/vomsdir`)

Summary -1



slide based on presentation given by Carl Kesselman at GGF Summer School 2004

Summary -2

1. **Authentication - communication of identity**
   - **X.509 certificate issued by Certificate Authority**
   - **proxy extensions**
   - **long-lived proxies can be held in MyProxy server**

   Basis for
   - Message integrity and confidentiality
   - Building trust – users, sites, services trust CA’s
   - Non-repudiation: knowing who did what when – can’t deny it
2. **Authorisation - once identity is known, what can a user do?**
   - **Determined by their group and roles in Virtual Organisation**
   - **VOMS: Virtual Organisation Management System**
3. **Delegation- A allows B to act on behalf of A**
   - **Proxies**
   - **VOMS: determines rights of users**

Further Information

Grid

- LCG Security: http://proj-lcg-security.web.cern.ch/proj-lcg-security/
- LCG Registration: http://lcg-registrar.cern.ch/
- Globus Security: http://www.globus.org/security/
- VOMS: http://infnforge.cnaf.infn.it/projects/voms

Background

- GGF Security: http://www.gridforum.org/security/
- IETF PKIX charter: http://www.ietf.org/html.charters/pkix-charter.html
- PKCS: http://www.rsasecurity.com/rsalabs/pkcs/index.html