

# Security on Grid:

Emidio Giorgio

INFN – Catania

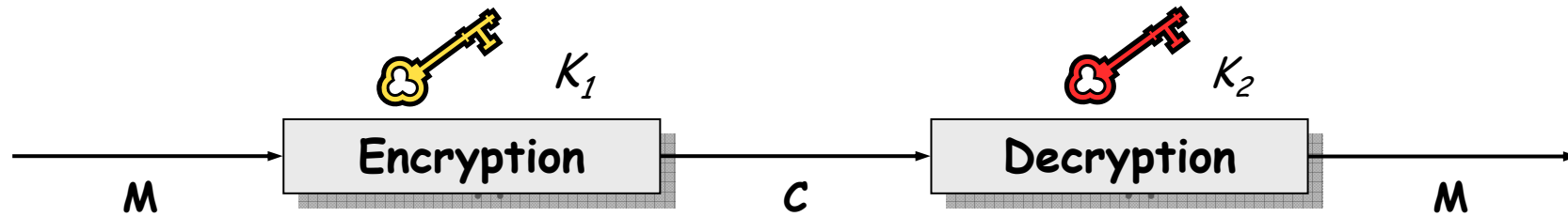
*EGEE Tutorial, Rome, 2 November 2005*

- **Glossary**
- **Encryption**
  - Symmetric algorithms
  - Asymmetric algorithms: PKI
- **Certificates**
  - Digital Signatures
  - X509 certificates
- **Grid Security**
  - Basic concepts
  - Grid Security Infrastructure
  - Proxy certificates
  - Command line interfaces
- **Virtual Organisation**
  - Concept of VO and authorization
  - VOMS, LCAS, LCMAPS

- **Glossary**
- **Encryption**
  - Symmetric algorithms
  - Asymmetric algorithms: PKI
- **Certificates**
  - Digital Signatures
  - X509 certificates
- **Grid Security**
  - Basic concepts
  - Grid Security Infrastructure
  - Proxy certificates
  - Command line interfaces
- **Virtual Organisation**
  - Concept of VO and authorization
  - VOMS, LCAS, LCMAPS

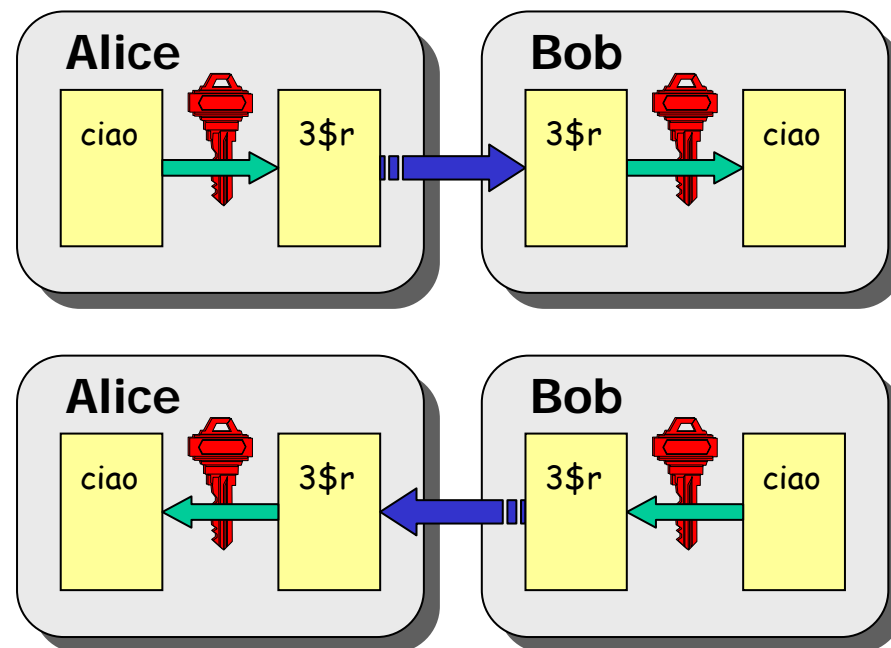
- **Principal**
  - An entity: a user, a program, or a machine
- **Credentials**
  - Some data providing a proof of identity
- **Authentication**
  - Verify the identity of a principal
- **Authorization**
  - Map an entity to some set of privileges
- **Confidentiality**
  - Encrypt the message so that 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

- **Glosary**
- **Encryption**
  - Symmetric algorithms
  - Asymmetric algorithms: PKI
- **Certificates**
  - Digital Signatures
  - X509 certificates
- **Grid Security**
  - Basic concepts
  - Grid Security Infrastructure
  - Proxy certificates
  - Command line interfaces
- **Virtual Organisation**
  - Concept of VO and authorization
  - VOMS, LCAS, LCMAPS

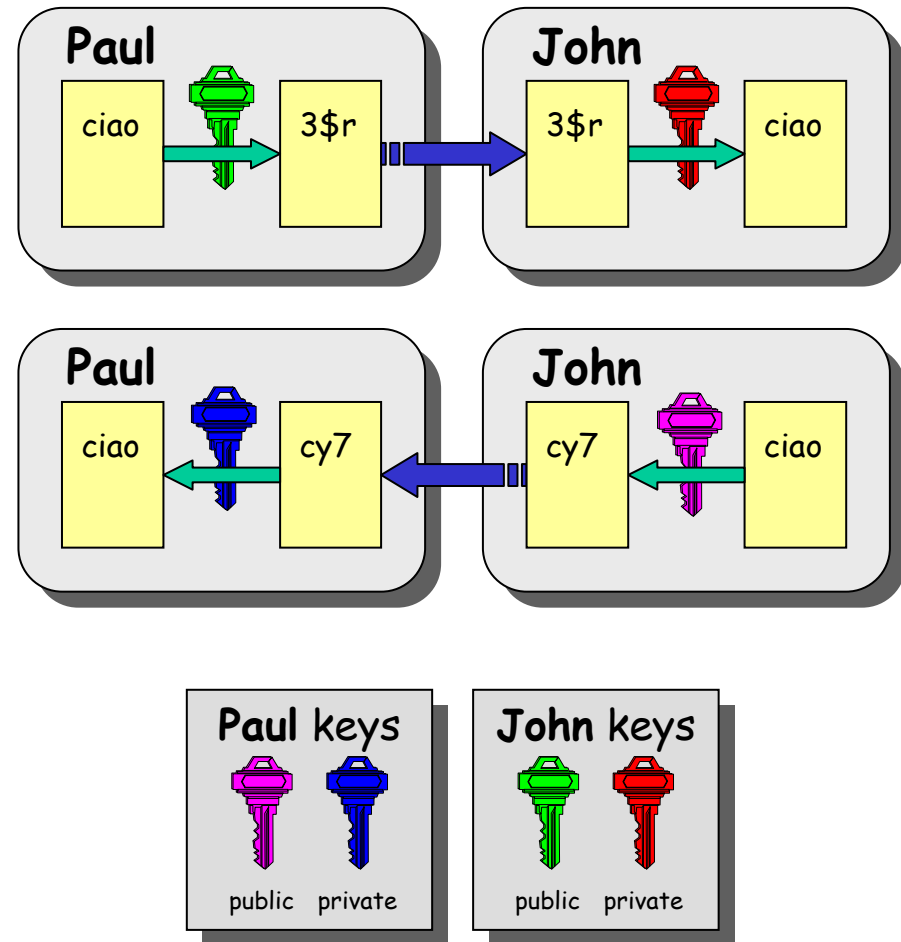


- **Mathematical algorithms that provide important building blocks for the implementation of a security infrastructure**
- **Symbology**
  - Plaintext:  $M$
  - Cyphertext:  $C$
  - Encryption with key  $K_1$ :  $E_{K_1}(M) = C$
  - Decryption with key  $K_2$ :  $D_{K_2}(C) = M$
- **Algorithms**
  -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

- The same key is used for encryption and decryption
- Advantages:
  - Fast
- Disadvantages:
  - how to distribute the keys?
  - the number of keys is  $O(n^2)$
- Examples:
  - DES
  - 3DES
  - Rijndael (AES)
  - Blowfish
  - Kerberos



- **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.
- **No exchange of secrets is necessary**
  - the sender cyphers using the *public* key of the receiver;
  - the receiver decrypts using his *private* key;
  - the number of keys is  $O(n)$ .
- **Examples:**
  - **Diffie-Hellmann** (1977)
  - **RSA** (1978)

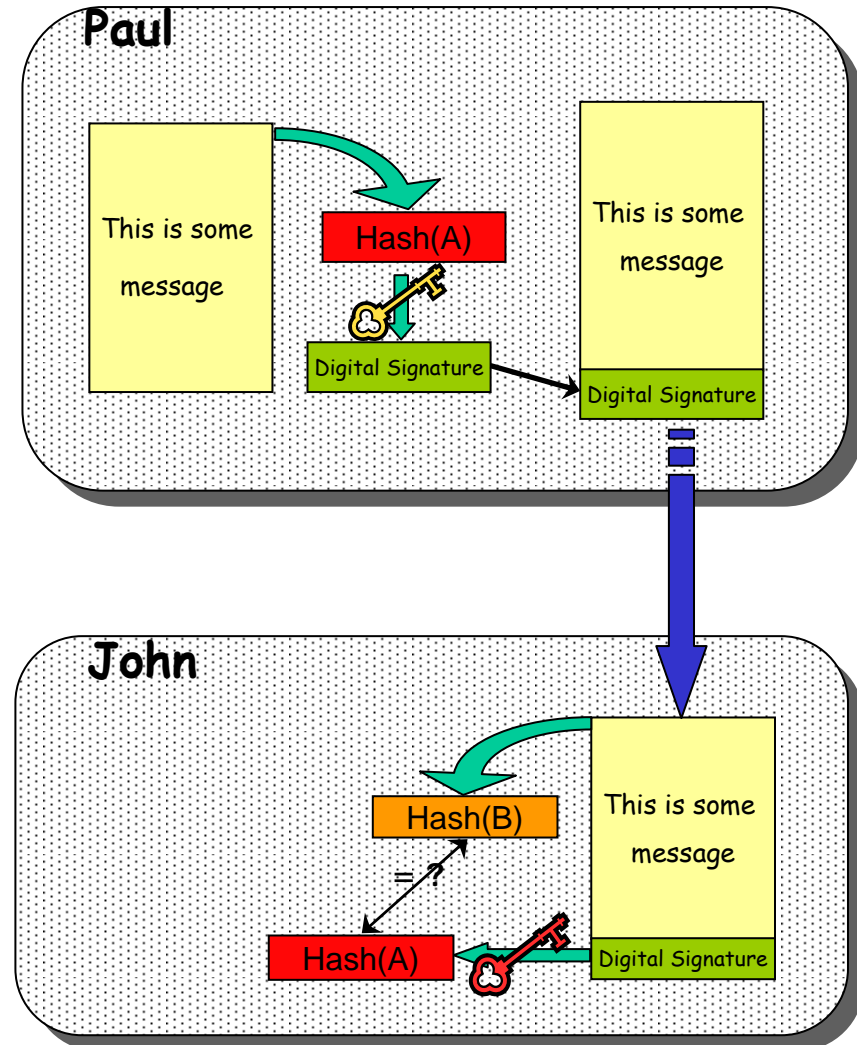
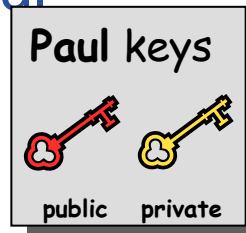




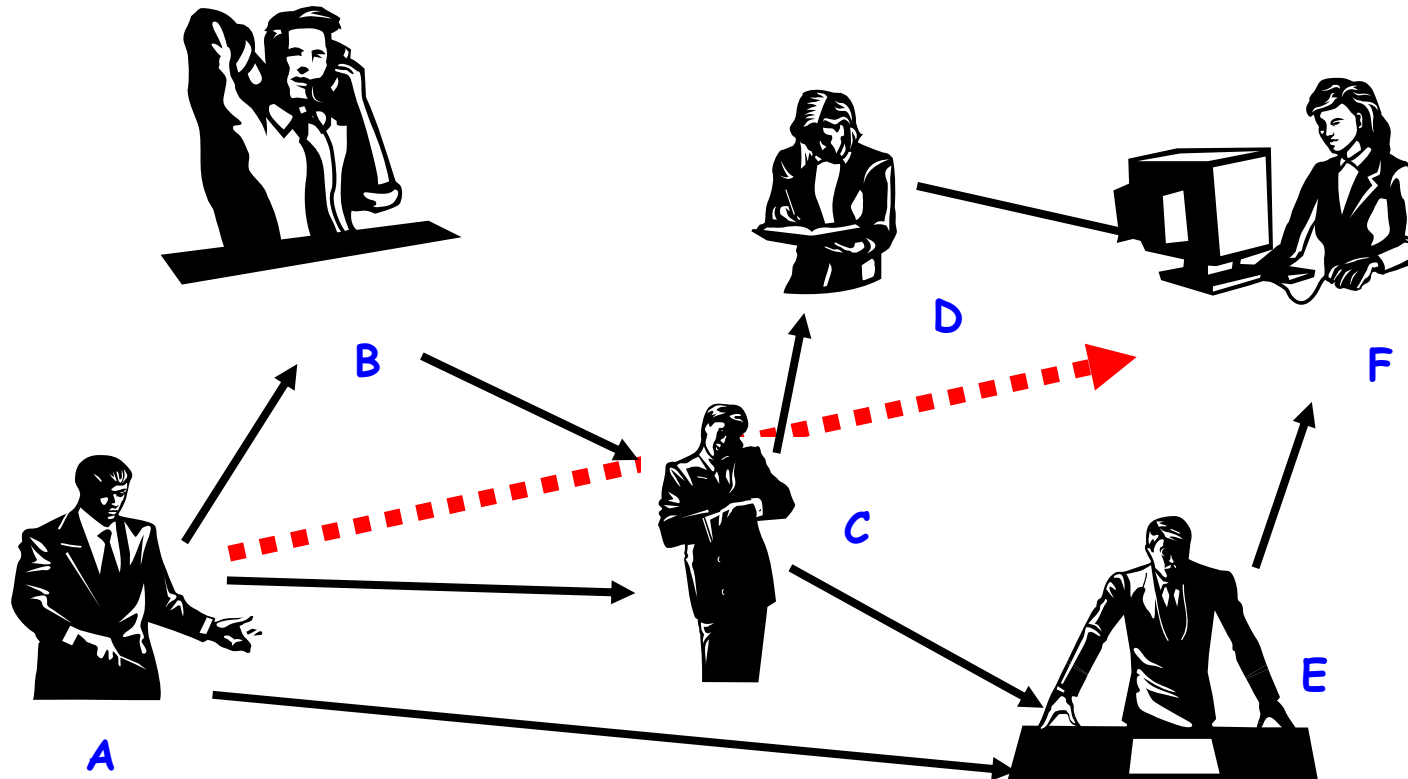
- **Glossary**
- **Encryption**
  - Symmetric algorithms
  - Asymmetric algorithms: PKI
- **Certificates**
  - Digital Signatures
  - X509 certificates
- **Grid Security**
  - Basic concepts
  - Grid Security Infrastructure
  - Proxy certificates
  - Command line interfaces
- **Virtual Organisation**
  - Concept of VO and authorization
  - VOMS, LCAS, LCMAPS

- **Functions ( $H$ ) that given as input a variable-length message ( $M$ ) produce as output a string of fixed length ( $h$ )**
  - the length of  $h$  must be at least 128 bits (to avoid *birthday attacks*)
  - 1. given  $M$ , it **must be easy** to calculate  $H(M) = h$
  - 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  $H(M) = H(M')$
- **Examples:**
  - **SNEFRU**: hash of 128 or 256 bits;
  - **MD4/MD5**: hash of 128 bits;
  - **SHA** (Standard FIPS): hash of 160 bits.

- 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 and verifies it with A, decyphered with Paul's *public* key.
- If hashes equal: message wasn't modified; Paul cannot repudiate it.



- **Paul's digital signature is safe if:**
  1. Paul's private key is not compromised
  2. John knows 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* guarantees the correspondence between public key and owner's identity.
  - Both A and B must trust this third party
- **Two models:**
  - X.509: hierarchical organization;
  - PGP: "web of trust".



- F knows D and E, who knows A and C, who knows A and B.
- F is reasonably sure that the key from A is really from A.

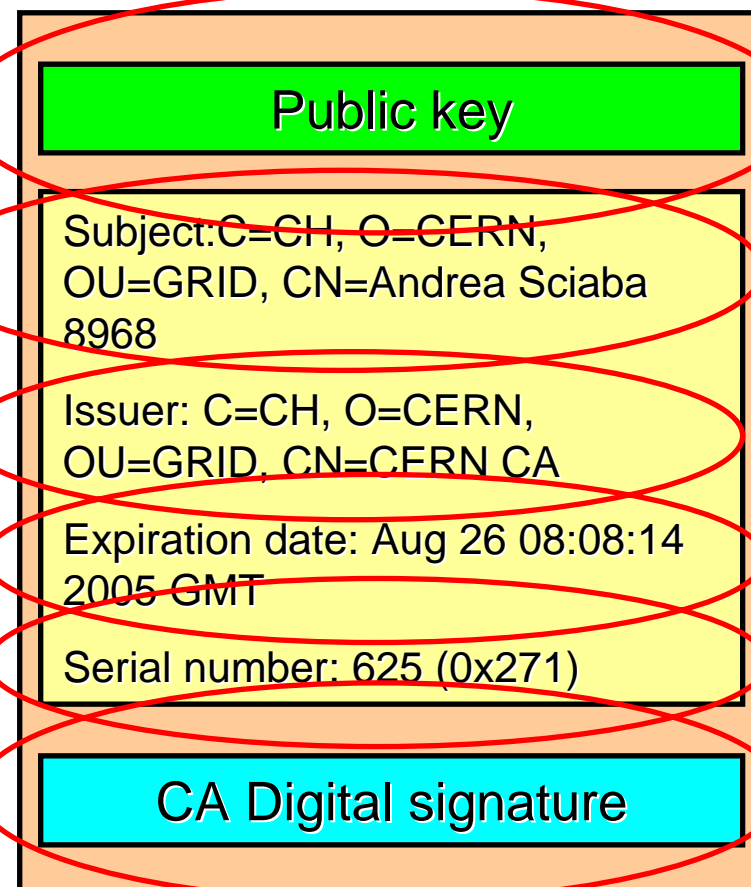
The “third party” is called Certification Authority (CA).

- Issue Digital Certificates for users, programs and machines
- Check the identity and the personal data of the requestor
  - 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 certificates are self-signed

- **An X.509 Certificate contains:**

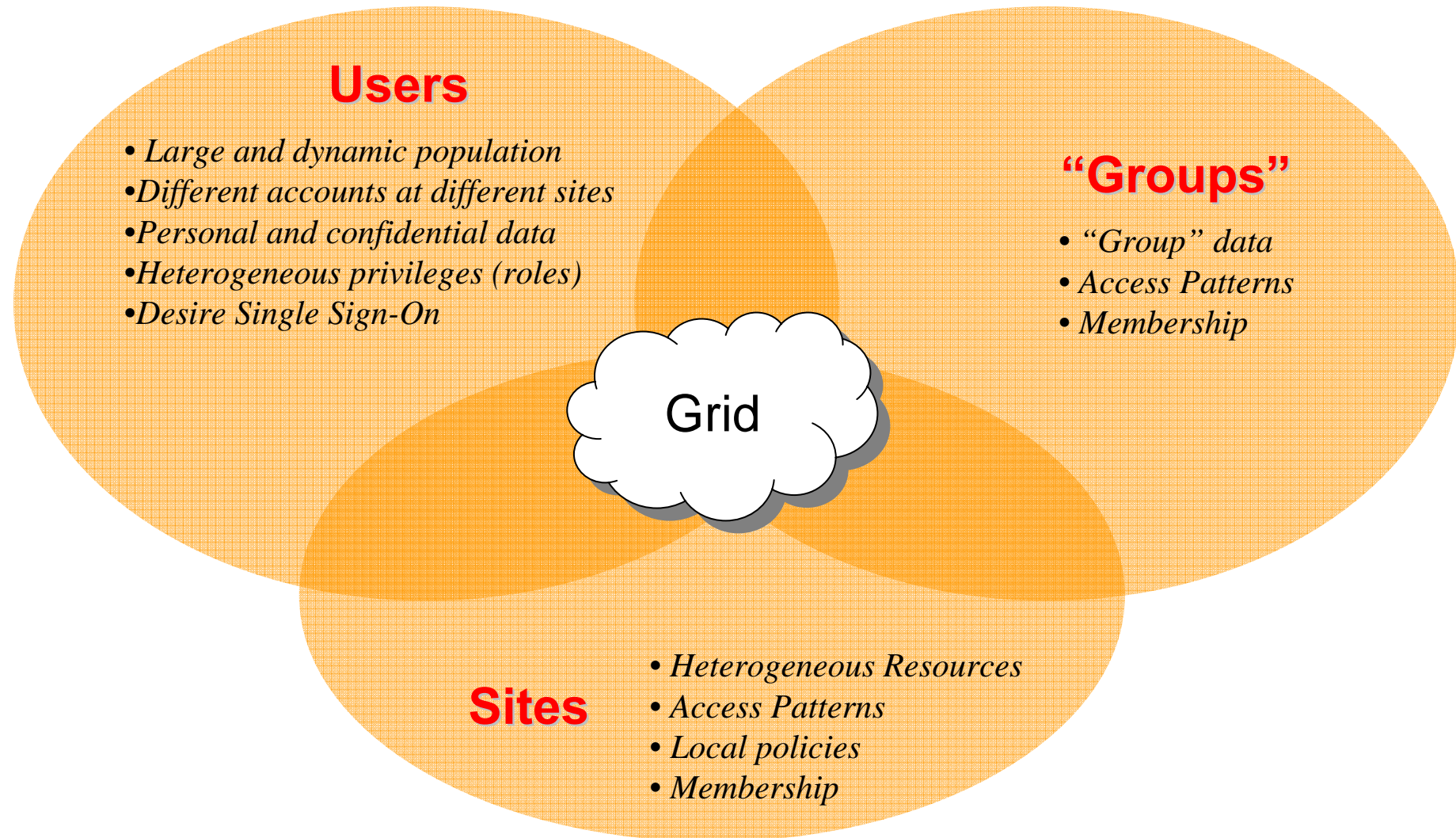
- owner's public key;
- identity of the owner;
- info on the CA;
- time of validity;
- Serial number;
- digital signature of the CA

**Structure of a X.509 certificate**



- **Glossary**
- **Encryption**
  - Symmetric algorithms
  - Asymmetric algorithms: PKI
- **Certificates**
  - Digital Signatures
  - X509 certificates
- **Grid Security**
  - Basic concepts
  - Grid Security Infrastructure
  - Proxy certificates
  - Command line interfaces
- **Virtual Organisation**
  - Concept of VO and authorization
  - VOMS, LCAS, LCMAPS

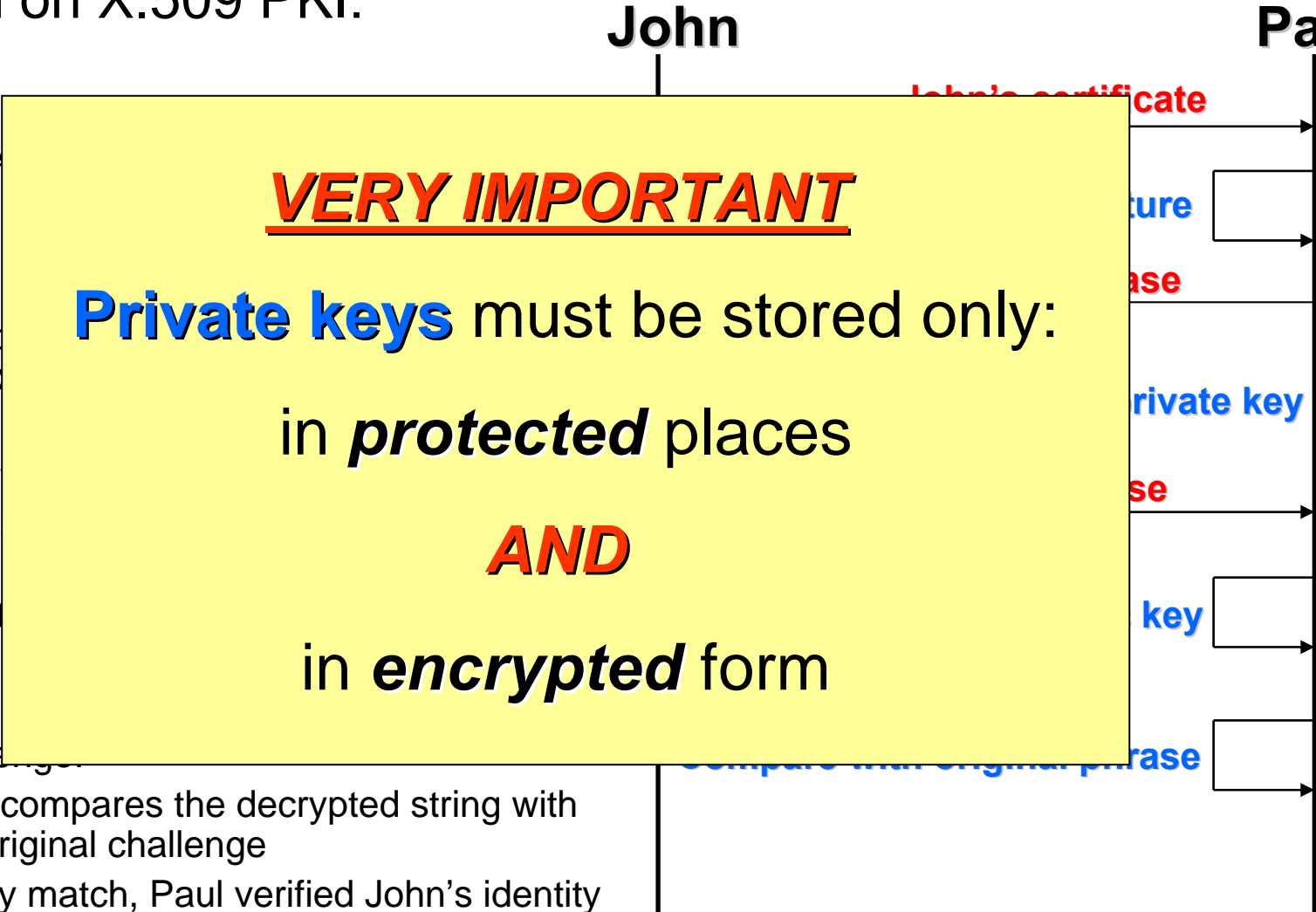




Based on X.509 PKI:

- every user has a certificate
- certificate authorities (CAs)
- every Grid user authenticates

1. John sends his certificate to Paul
2. Paul checks the certificate
3. Paul sends a challenge to John
4. John signs the challenge with his private key
5. John sends the signed challenge to Paul
6. Paul decrypts the challenge
7. Paul compares the decrypted string with the original challenge
8. If they match, Paul verified John's identity and John can not repudiate it.



**VERY IMPORTANT**

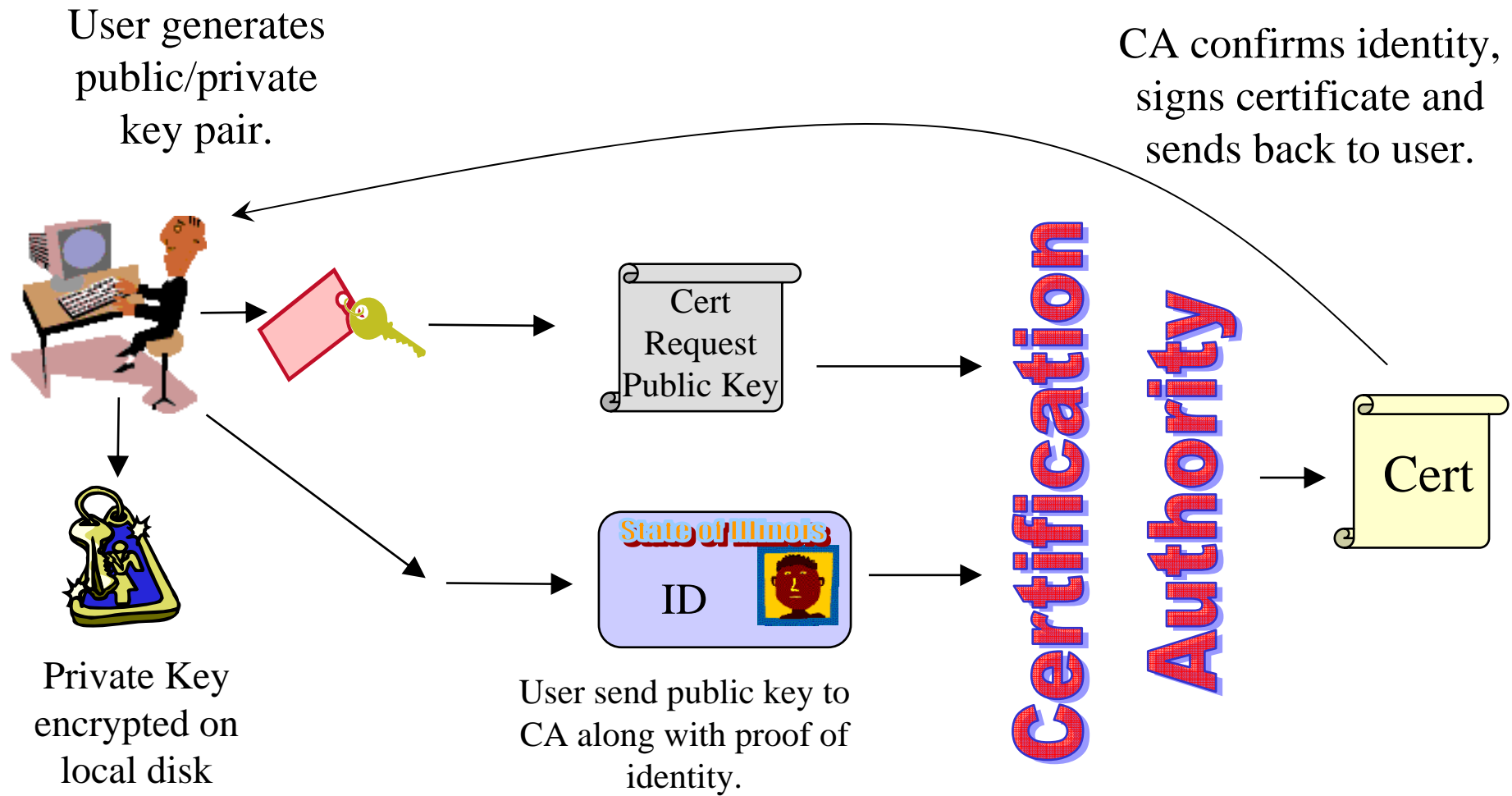
**Private keys** must be stored only:

in *protected* places

**AND**

in *encrypted* form

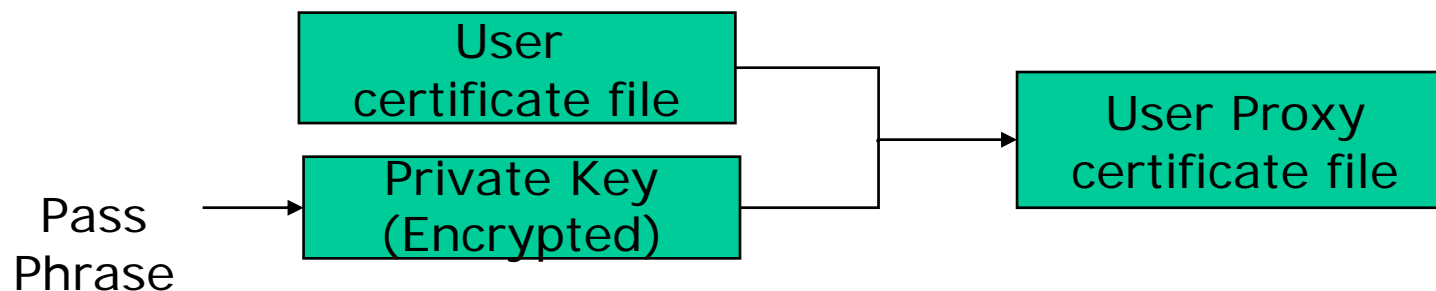
- **Egee/LCG recognizes a given set of CAs**
  - [https://lcg-registrar.cern.ch/pki\\_certificates.html](https://lcg-registrar.cern.ch/pki_certificates.html)
- **How do you request a certificate depends on your CA**
- **For GILDA, have a look at the Video Tutorials:**
  - <https://gilda.ct.infn.it/video/Certification/Allproxy.html> (Flash)
  - <https://gilda.ct.infn.it/video/Certification/AllCertproxy.ram> (Real)



- **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`

- **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

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

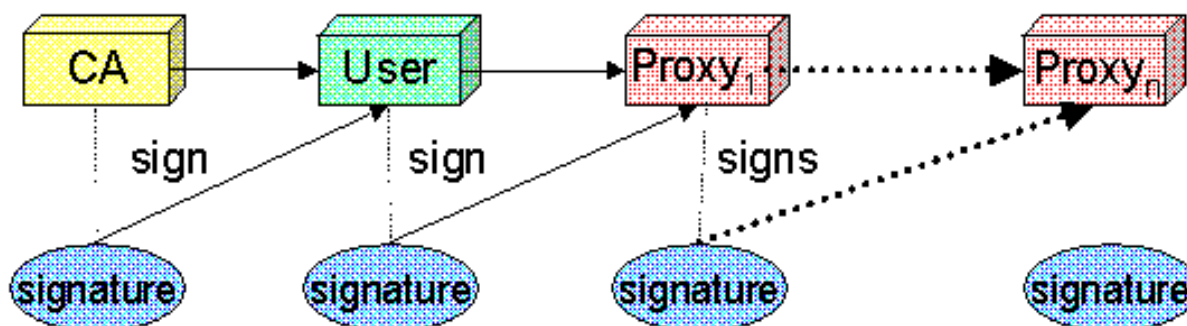


- **Proxy placed in /tmp**
  - the private key of the Proxy is *not* encrypted:
  - stored in local file: must be readable **only** by the owner;
  - proxy lifetime is short (typically 12 h) to minimize security risks.
- **NOTE: No network traffic!**

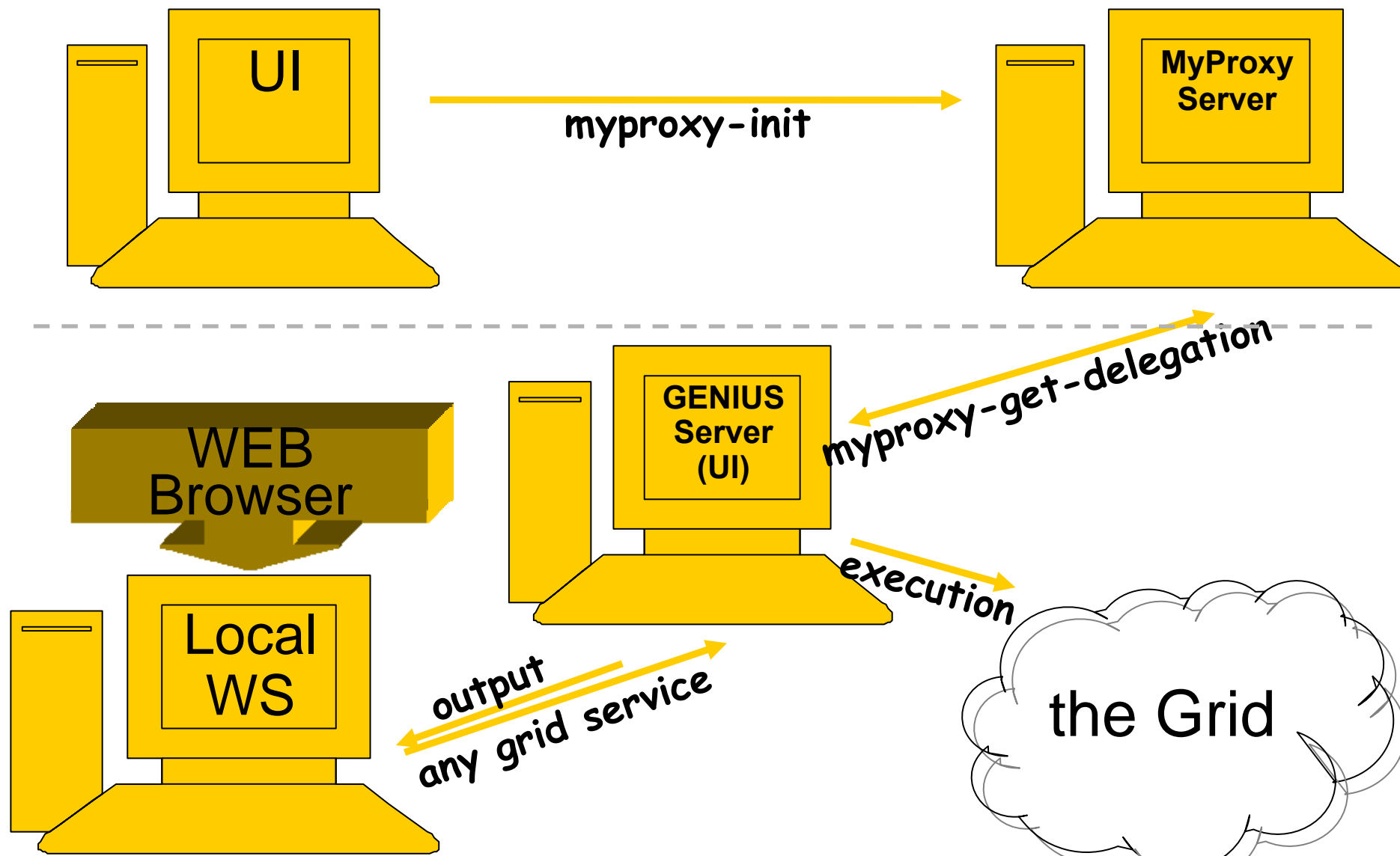
- **grid-proxy-init**  $\equiv$  “login to the Grid”
- **To “logout” you have to destroy your proxy:**
  - grid-proxy-destroy
  - This does *NOT* destroy any proxies that were delegated from this proxy.
  - You cannot revoke a remote proxy
  - Usually create proxies with short lifetimes
- **To gather information about your proxy:**
  - grid-proxy-info
  - Options for printing proxy information
    - subject                      -issuer
    - type                         -timeleft
    - strength                    -help



- **Delegation = remote creation of a (second level) proxy credential**
  - New key pair generated remotely on server
  - Client signs proxy cert and returns it
- **Allows remote process to authenticate on behalf of the user**
  - Remote process “impersonates” the user
- **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



- **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 validates user request and eventually renew proxies**
  - contacting myproxy server



- **Glossary**
- **Encryption**
  - Symmetric algorithms
  - Asymmetric algorithms: PKI
- **Certificates**
  - Digital Signatures
  - X509 certificates
- **Grid Security**
  - Basic concepts
  - Grid Security Infrastructure
  - Proxy certificates
  - Command line interfaces
- **Virtual Organisation**
  - Concept of VO and authorization
  - VOMS, LCAS, LCMAPS

- **Grid users MUST belong to virtual organizations**
  - What we previously called “groups”
  - 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](https://lcg-registrar.cern.ch/virtual_organization.html)

- **Vos maintain 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
...

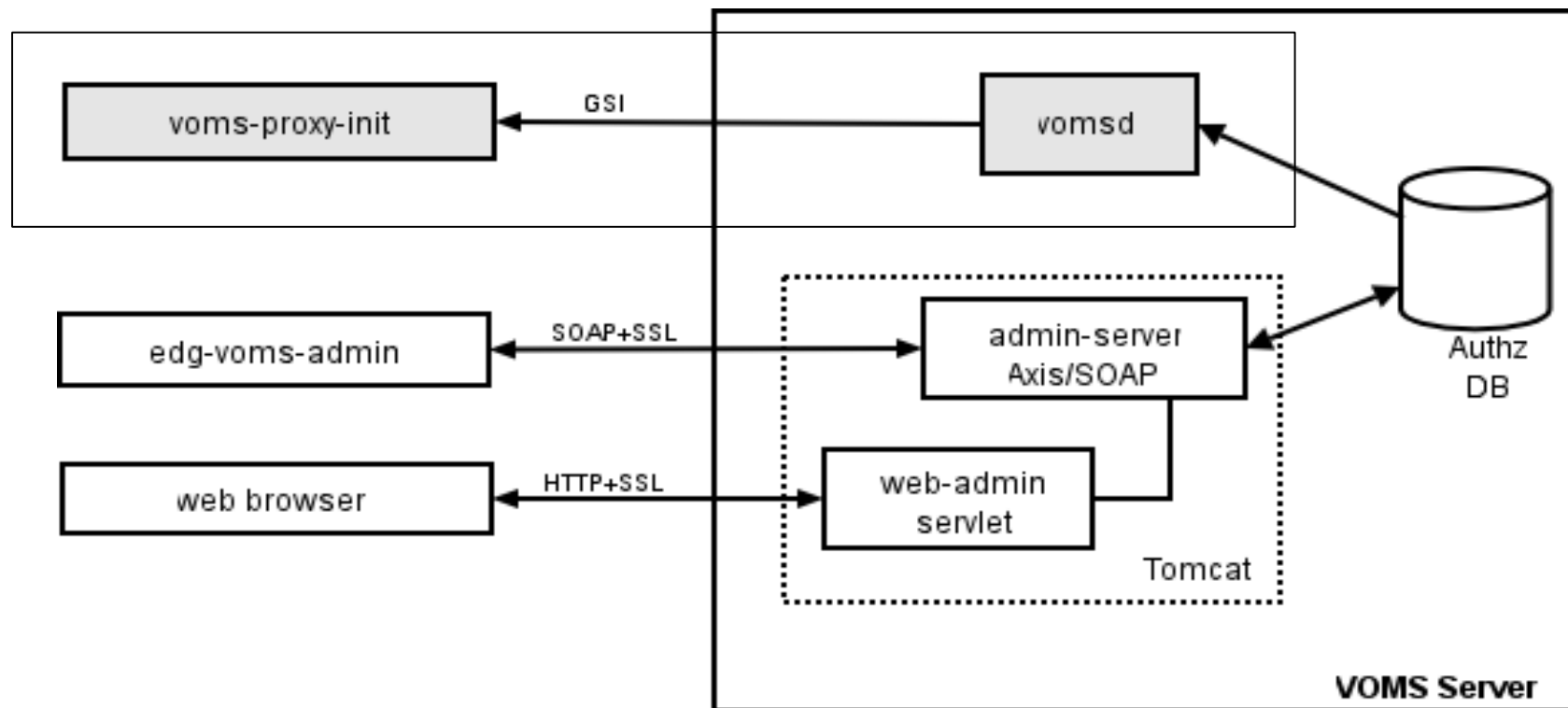
```

- Sites decide which vos to accept

/etc/grid-security/grid-mapfile

- **Virtual Organization Membership Service**
  - Extends the proxy with info on VO membership, group, role and capabilities
  - 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
gildav
Your identity: /C=IT/O=GILDA/OU=Personal
Certificate/L=INFN/CN=Emidio
Giorgio/Email=emidio.giorgio@ct.infn.it
Enter GRID pass phrase:
Creating temporary proxy ..... Done
Contacting cert-voms-01.cnaf.infn.it:15008
[/C=IT/O=INFN/OU=Host/L=CNAF/CN=cert-voms-01.cnaf.infn.it]
"gildav" Done
Creating proxy ..... Done
Your proxy is valid until Thu Oct 20 05:37:22 2005
```



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

- short for Fully Qualified Attribute Name, is what VOMS uses 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
/gildav/Role=NULL/Capability=NULL
/gildav/staff/Role=NULL/Capability=NULL
/gildav/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



- 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

```
[glite-tutor] /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  : 11:28:55
VO        : gildav
subject   : /C=IT/O=GILDA/OU=Personal Certificate/L=INFN/CN=Emidio
Giorgio/Email=emidio.giorgio@ct.infn.it
issuer    : /C=IT/O=INFN/OU=Host/L=CNAF/CN=cert-voms-01.cnaf.infn.it
attribute : /gildav/Role=NULL/Capability=NULL
attribute : /gildav/staff/Role=NULL/Capability=NULL
attribute : /gildav/tutors/Role=NULL/Capability=NULL
timeleft  : 11:28:55
```

- **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
```

- **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`)

## 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>