

Security in gLite

Gergely Sipos

MTA SZTAKI

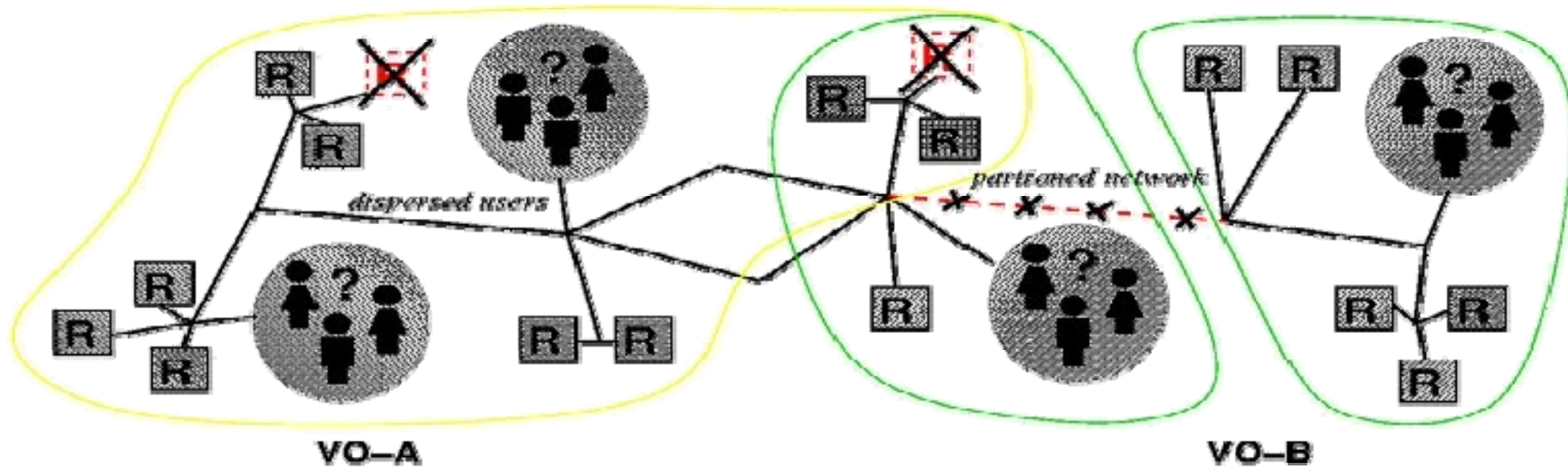
sipos@sztaki.hu

With thanks for some slides to EGEE and Globus colleagues

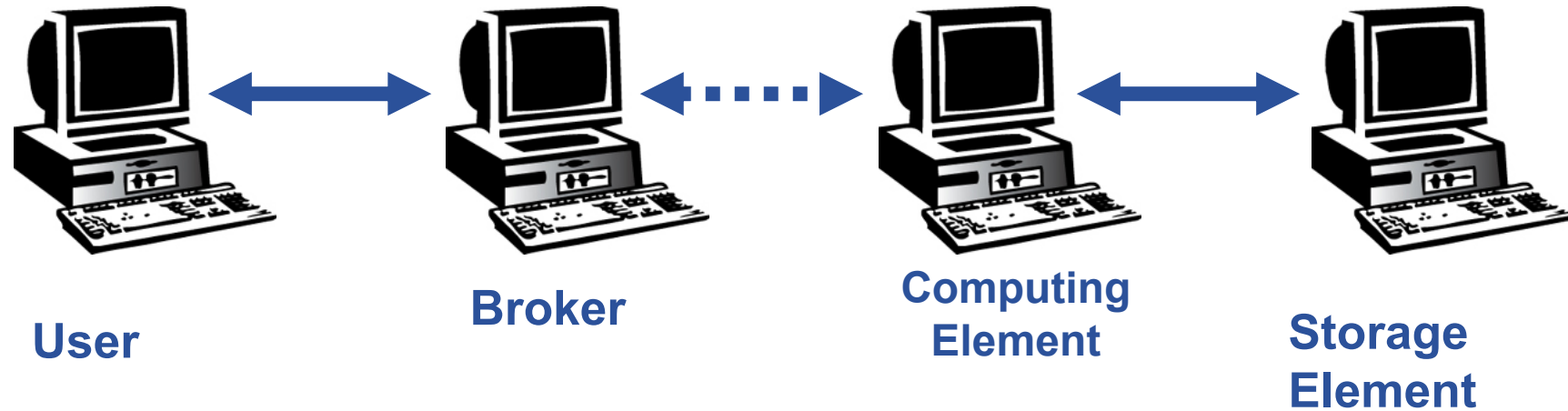
The Grid problem is to enable “coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations.”

From “The Anatomy of the Grid” by Ian Foster et al.

- So Grid Security is security to enable VOs
- What is needed in terms of security for a VO?



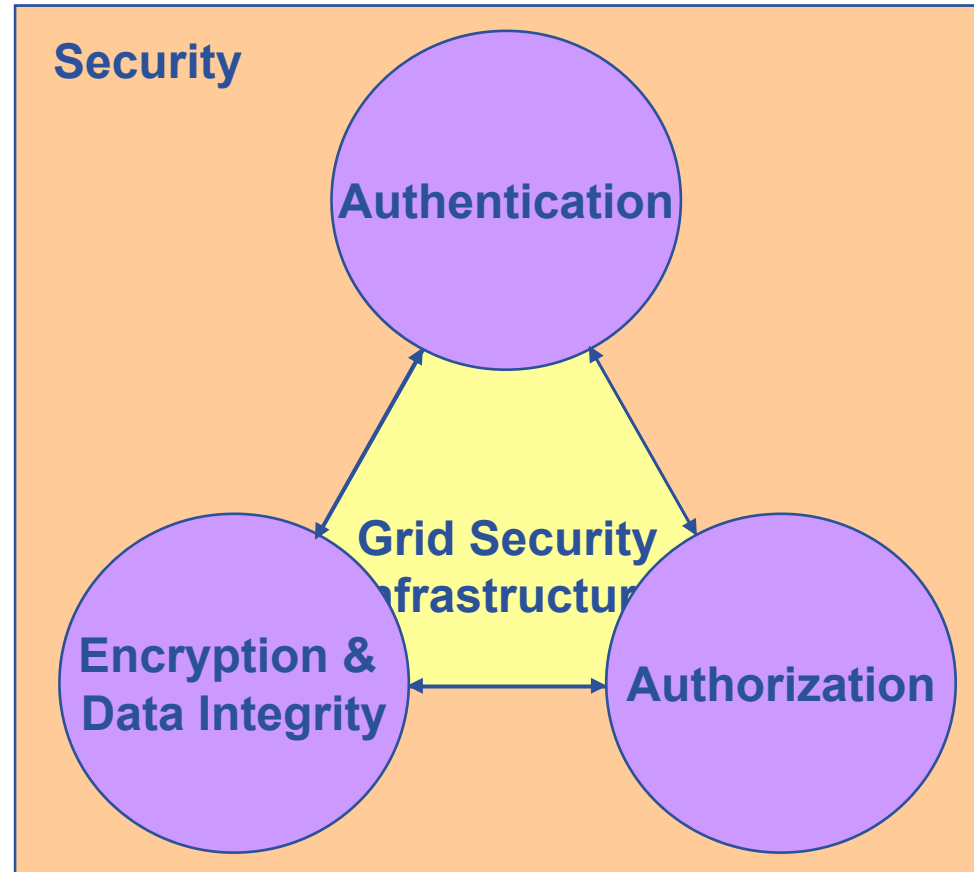
- VO for each application or workload
- Carve out and configure resources for a particular use and set of users
- The more dynamic the better...



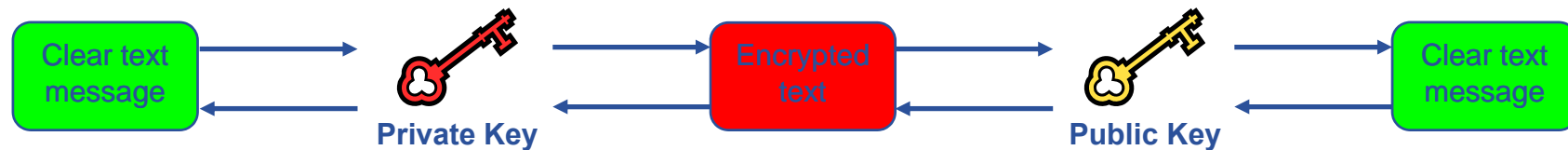
- How can the members of the VO identified?
- Who does belong to a VO? Who does not?
- How does a machine identifies its client?
- How are access rights controlled?
- How does a user access a VO resource without having an user account on the machines in between or even on the resource?

- **Launch attacks to other sites**
 - Large distributed farms of machines, perfect for launching a Distributed Denial of Service attack.
- **Illegal or inappropriate data distribution and access sensitive information**
 - Massive distributed storage capacity ideal for example, for swapping movies.
 - Growing number of users have data that must be private – biomedical imaging for example
- **Damage caused by viruses, worms etc.**
 - Highly connected infrastructure means worms could spread faster than on the internet in general.

- **Authentication:** how is identity of user/site communicated?
- **Authorisation:** what can a user do?
- **Encryption:** encrypted messages
- **Integrity:** unchanged messages



- **Asymmetric encryption...**



- **.... and Digital signatures ...**

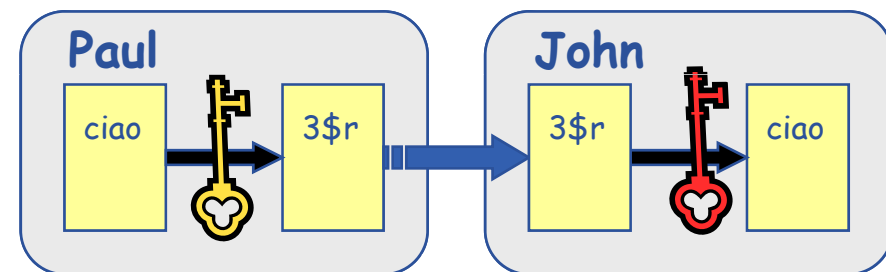
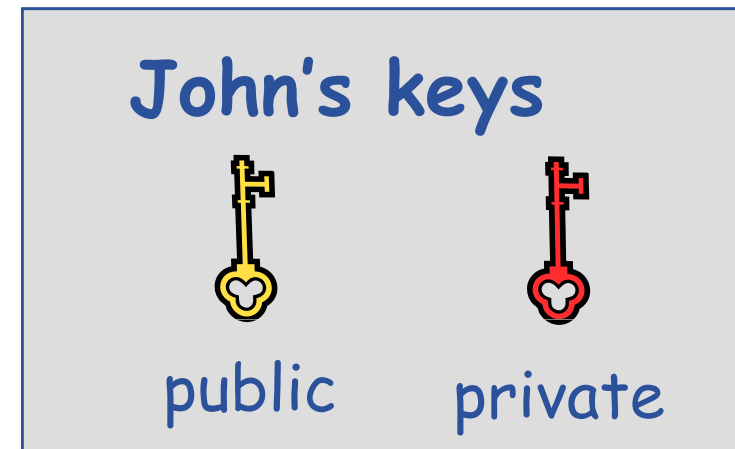
- A hash derived from the message and encrypted with the signer's private key
- Signature is checked by decrypting with the signer's public key

- **Are used to build trust**

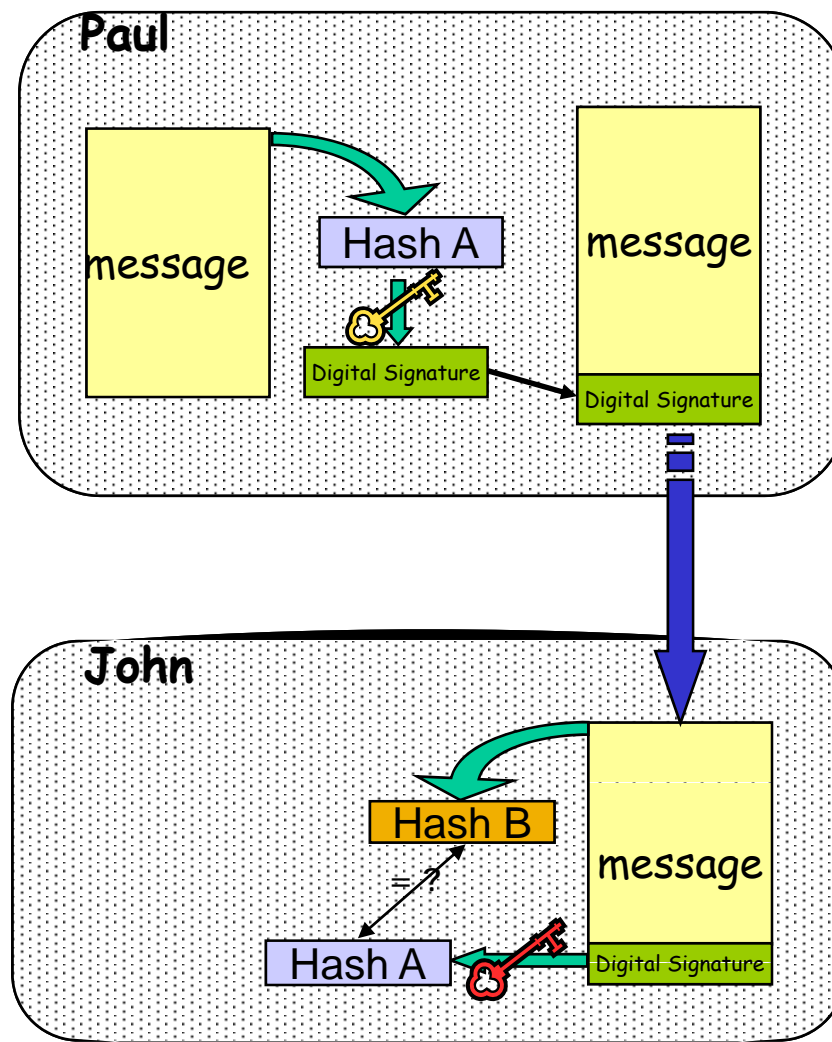
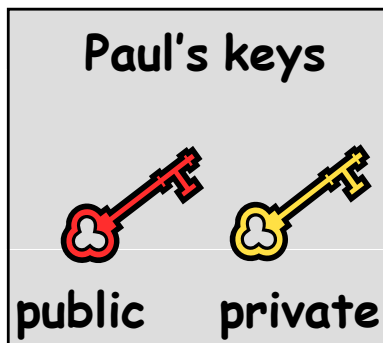
- That a user / site is who they say they are
- And can be trusted to act in accord with agreed policies

- Every entity that wants to join a VO (user/machine/software) 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.

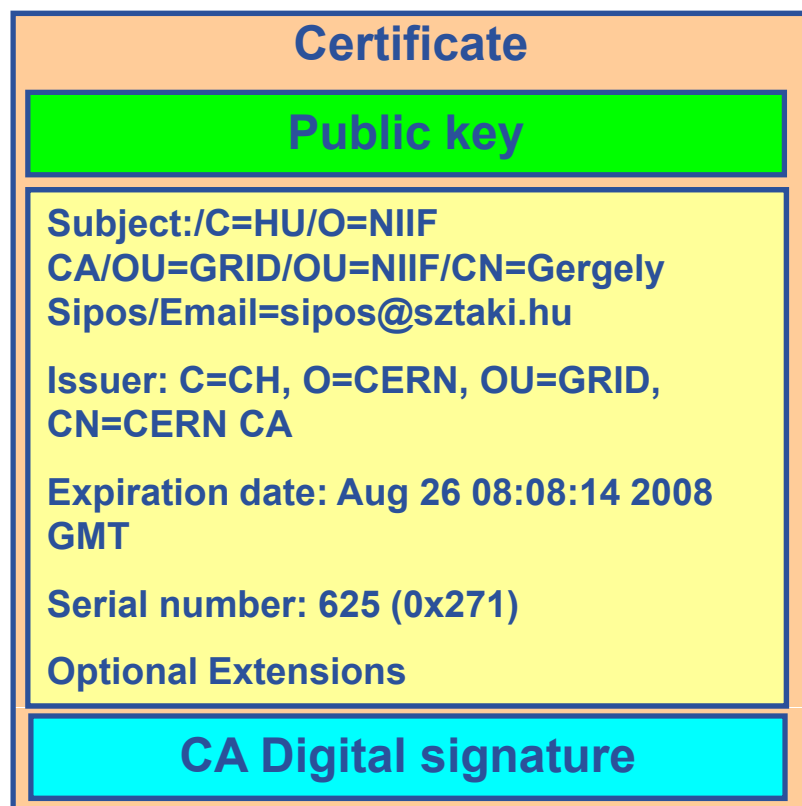
- **Concept (simplified version):**
 - Public keys are exchanged
 - The sender encrypts using receiver's public key
 - The receiver decrypts using their private key;



- Paul calculates the *hash* of the message: a 128 bit value based on the content 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 → Hash B
- Decrypts A with Paul's *public* key → Hash A
- If hashes equal:
 1. hash B is from Paul's private key;
 2. message wasn't modified;



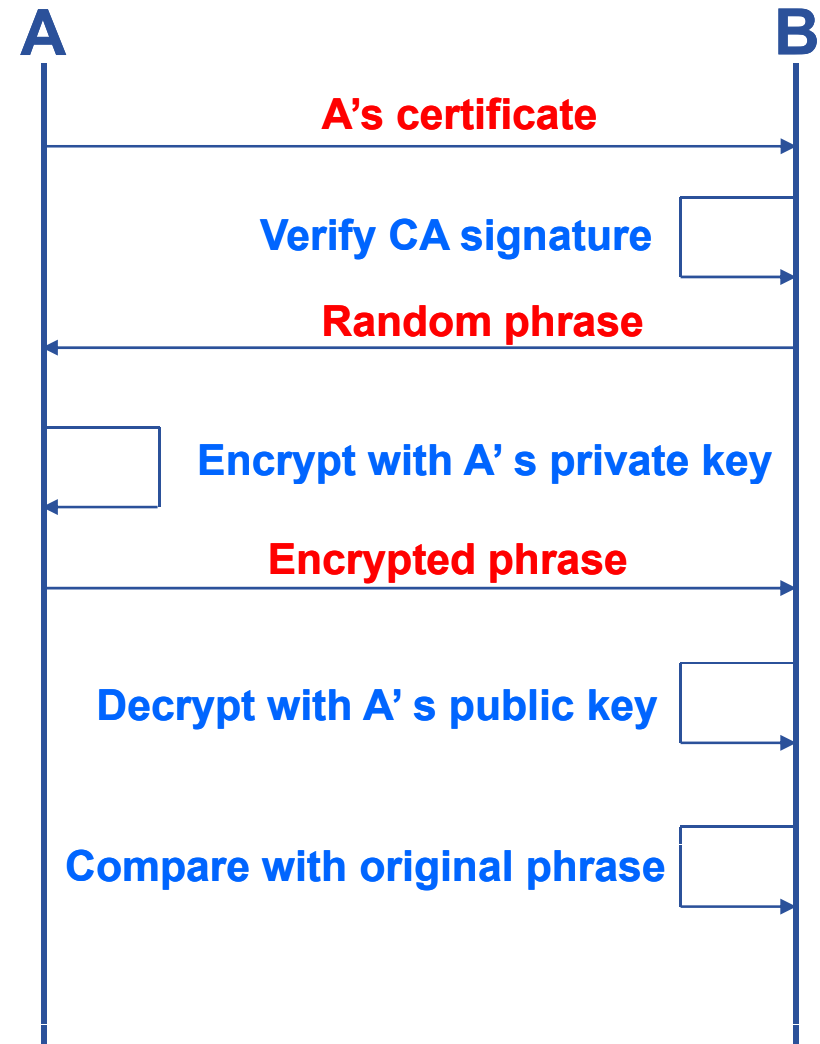
- Public key is wrapped into a “certificate file”
- Certificate files are created by trusted third parties: Grid Certification Authorities (CA)
- Private key is stored in encrypted file – protected by a passphrase
- Private key is created by the grid user



- **User's identity has to be certified by one of the national *Certification Authorities (CAs)***
- **Resources are also certified by CAs**
- **CAs recognized by EGEE VOs:**
<http://www.gridpma.org/>
- **CAs can establish a number of people “registration authorities” RAs**
 - Personal visit to the nearest RA instead of the national CA

Based on X.509 PKI:

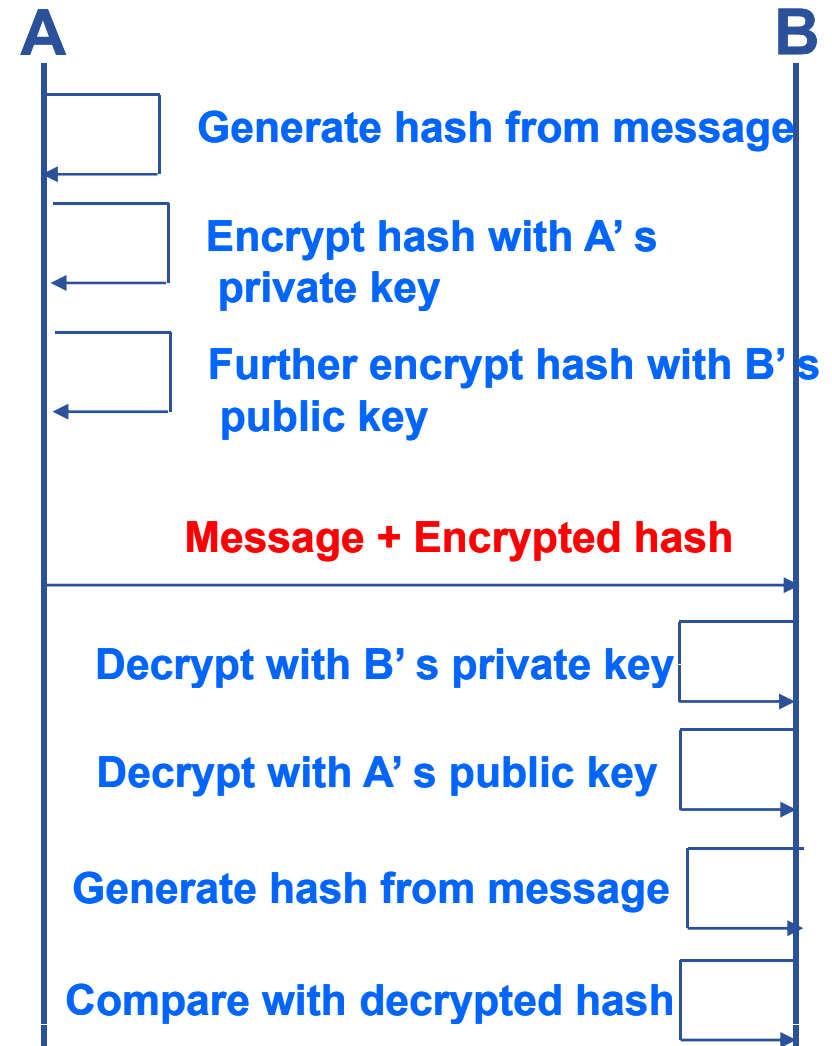
- every Grid transaction is mutually authenticated:
 1. A sends his certificate;
 2. B verifies signature in A's certificate using CA public certificate;
 3. B sends to A a challenge string;
 4. A encrypts the challenge string with his private key;
 5. A sends encrypted challenge to B
 6. B uses A's 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.
 9. Repeat for A to verify B's identity



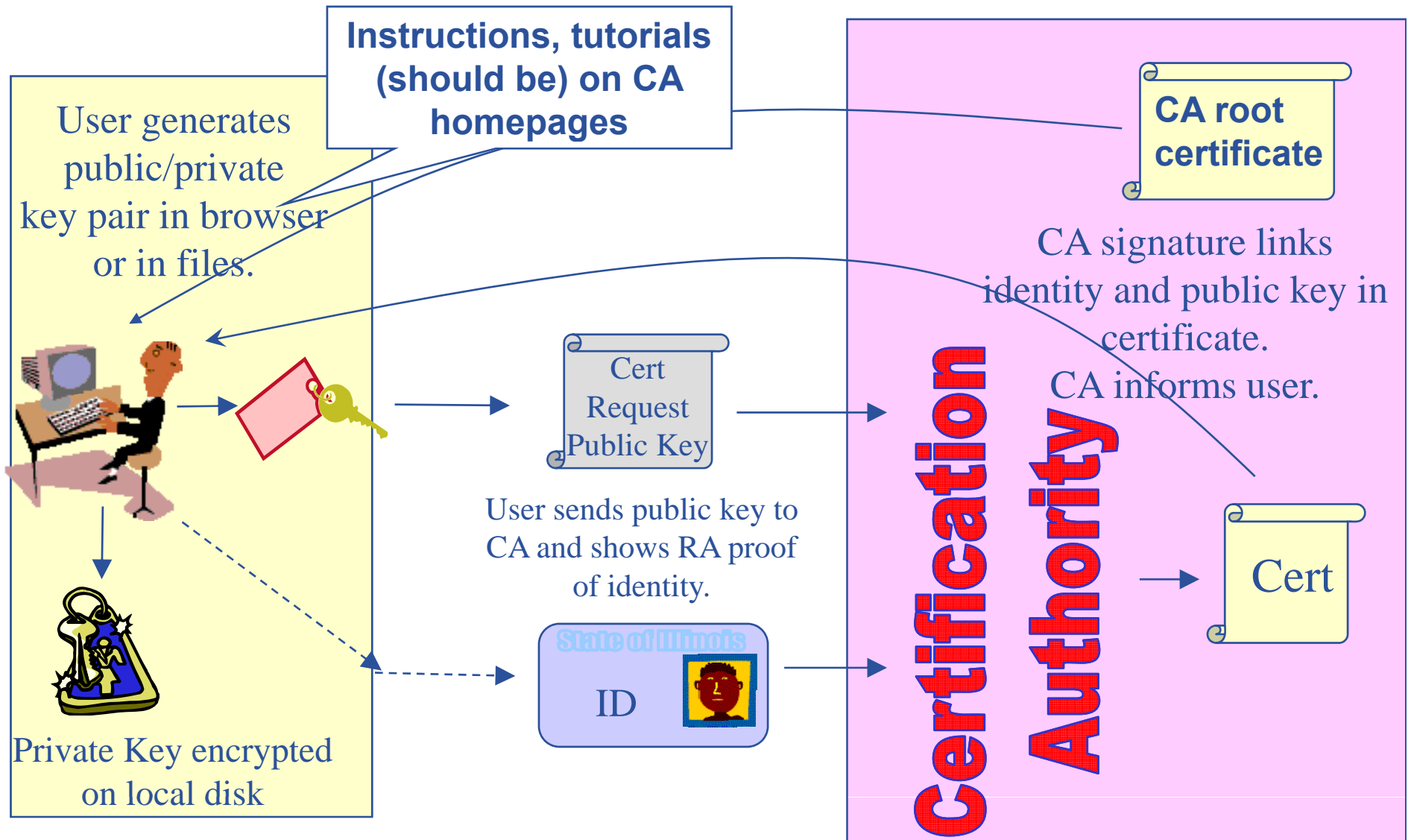
After A and B authenticated each other, for A to send a message to B:

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

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



Issuing a grid certificate



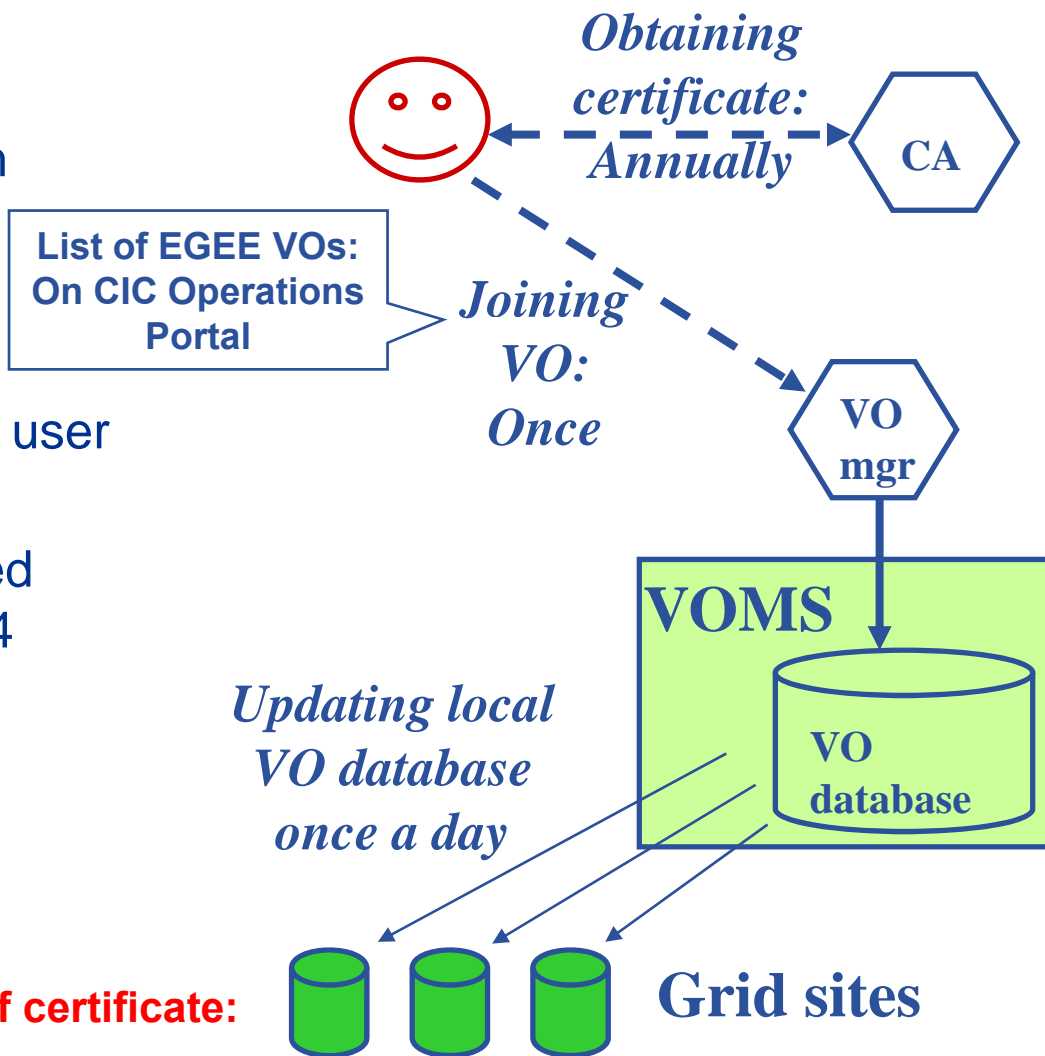
- **Keep your private key secure**
 - if possible *on a USB drive only*
- **Do not loan your certificate to anyone**
- **Report to your local/regional contact if your certificate has been compromised.**
- **Note file access rights:**

```
[sipos@glite-tutor sipos]$ ls -l .globus/
total 8
-rw-r--r--    1 sipos    users    1761 Oct 25  2006 usercert.pem
-r-----    1 sipos    users    951  Oct 24  2006 userkey.pem
```

If your certificate is used by someone other than you, it cannot be proven that it was not you.

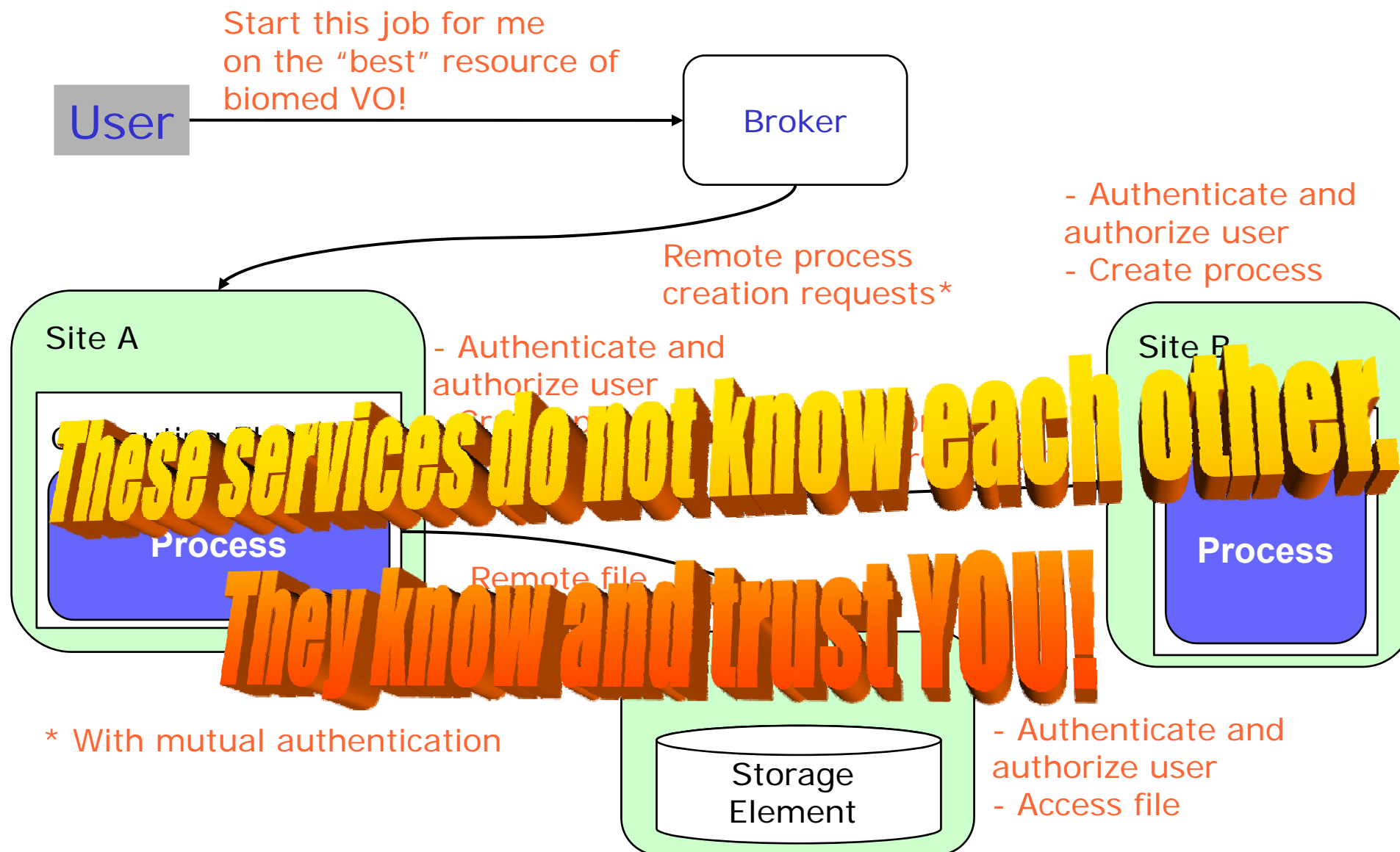
- **Steps**

- User obtains certificate from Certification Authority
- User registers at the VO
 - usually via a web form
- VO manager authorizes the user
 - VO DB updated
- User information is replicated onto VO resources within 24 hours



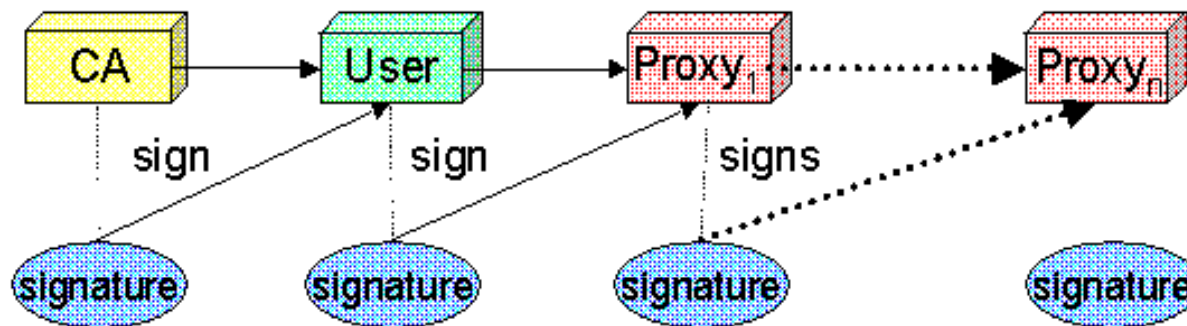
User's identity in the Grid = Subject of certificate:

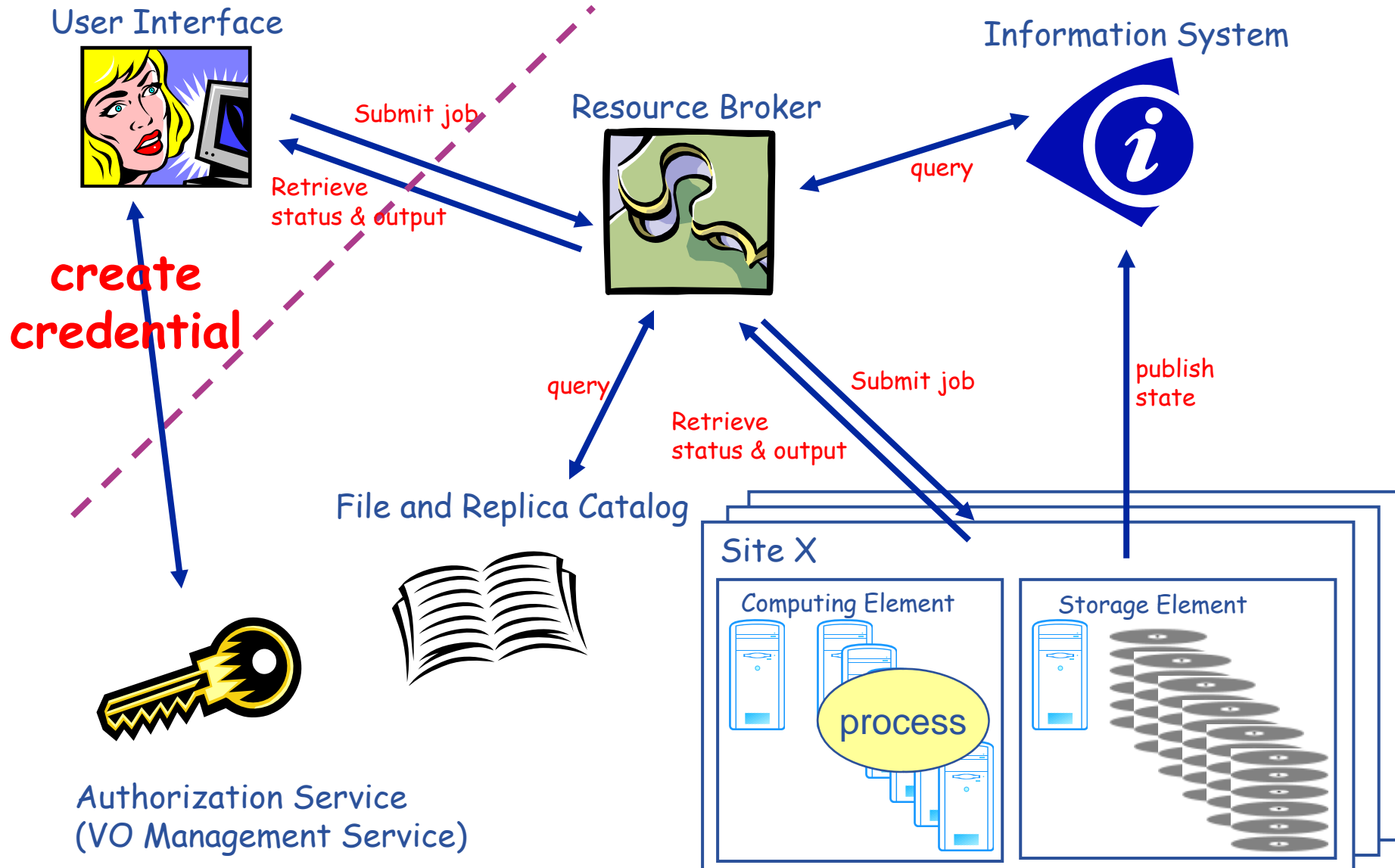
/C=HU/O=NIIF CA/OU=GRID/OU=NIIF/CN=Gergely Sipos/Email=sipos@sztaki.hu



* With mutual authentication

- **Delegation** - allows remote process and services to authenticate **on behalf of the user**
 - Remote process/service “**impersonates**” the user
- **Achieved by creation of next-level key-pair from the user’s key-pair.**
 - New key-pair is a single file: **Proxy credential**
 - Proxy has limited lifetime
 - Proxy may be valid for limited operations
- **The client can delegate the proxy to processes**
 - Each service decides whether it accepts proxies for authentication





```
[sipos@glite-tutor sipos]$ voms-proxy-init --voms gilda
Enter GRID pass phrase: *****
Your identity: /C=HU/O=NIIF CA/OU=GRID/OU=NIIF/CN=Gergely
Sipos/Email=sipos@sztaki.hu
Creating temporary proxy ..... Done
Contacting voms.ct.infn.it:15001 [/C=IT/O=INFN/OU=Host/L=Catania/CN=voms.ct.infn.it]
"gilda" Done
Creating proxy ..... Done
Your proxy is valid until Sat Jun 23 04:55:19 2007
```

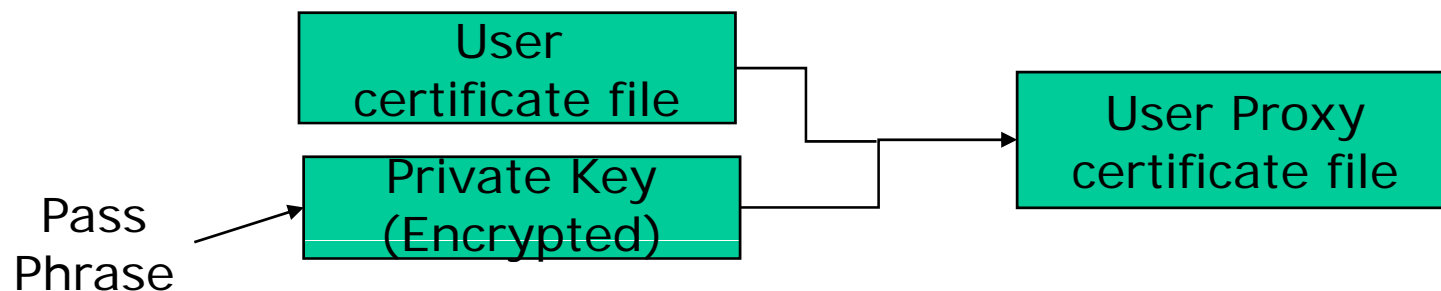
% voms-proxy-init → login to the Grid

Enter PEM pass phrase: ***** → private key is protected by a password

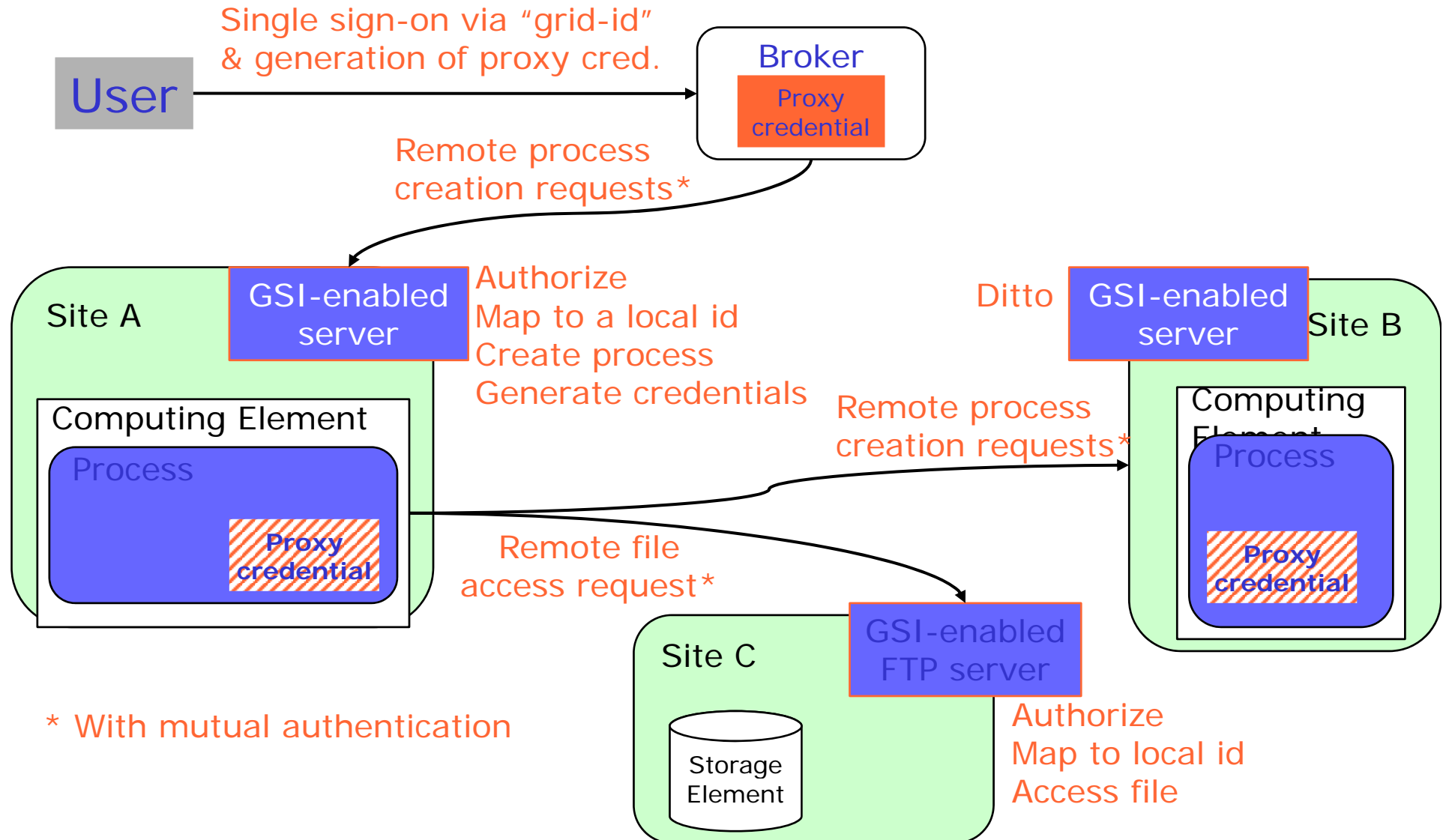
– Options for voms-proxy-init:

- VO name
- -hours <lifetime of new credential>
- -bits <length of key>
- -help

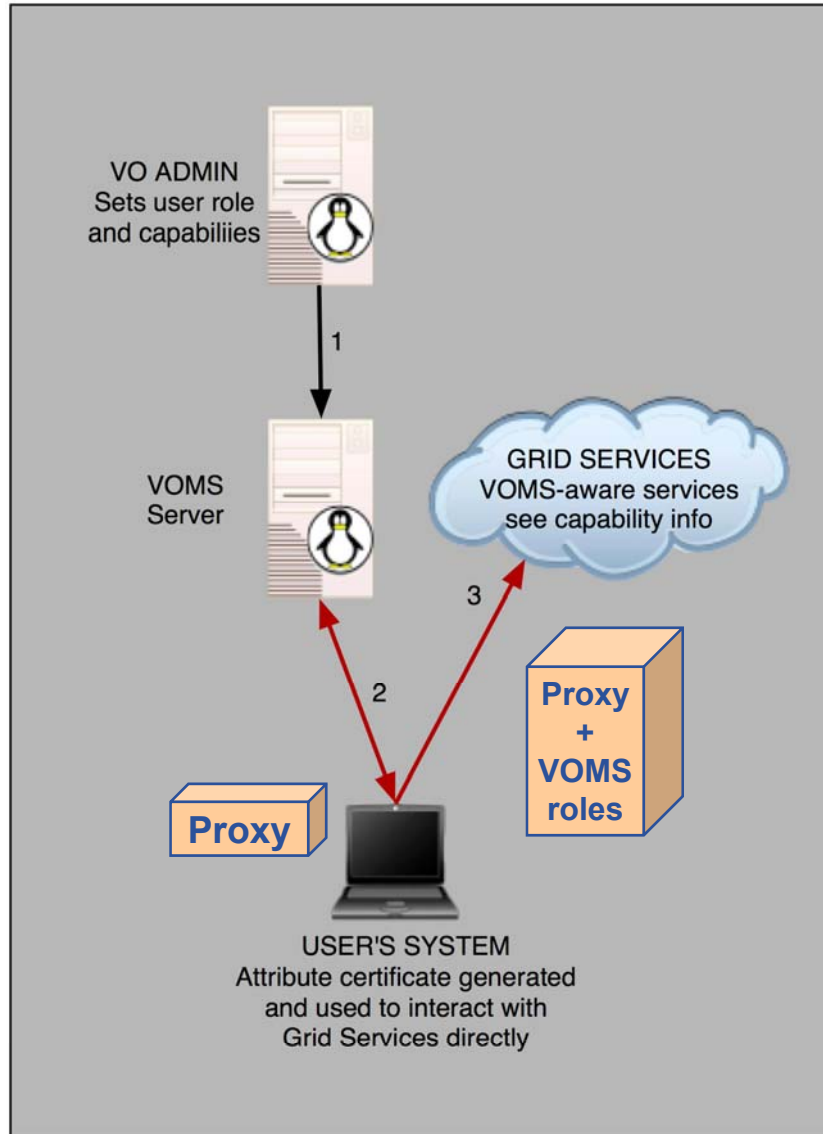
- User enters pass phrase, which is used to decrypt private key.
- New private and new public key-pair generated and saved into proxy file
- Original private key is used to sign the proxy file
 - User's private key not exposed after proxy has been signed



- Proxy file saved in `/tmp`
 - the private key part of the Proxy is *not* encrypted:
 - proxy lifetime is short (typically 12 h) to minimize security risks.
- NOTE: No network traffic during proxy creation!



- **voms-proxy-init** \equiv “login to the Grid”
- **To “logout” you have to destroy your proxy:**
 - `voms-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:**
 - `voms-proxy-info`
 - Options for printing proxy information
 - subject -issuer
 - type -timeleft
 - strength -help



- **VOMS: VO Management Service**

- VO level service
- Database of user roles

- **voms-proxy-init**

- Creates a proxy locally
- Contacts the VOMS server and extends the proxy with a role

`voms-proxy-init -voms gilda`

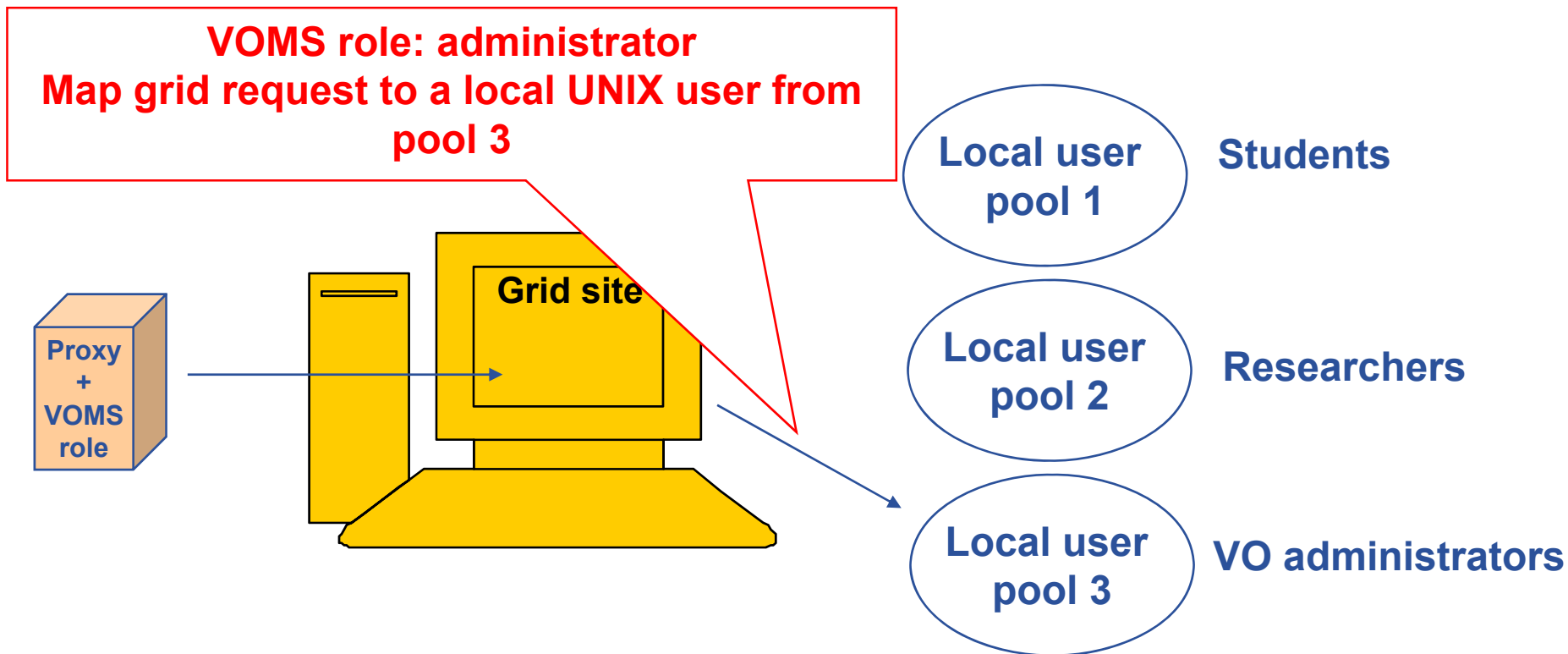
- **Allows VOs to centrally manage user roles**

Before VOMS

- All VO members have same rights
- Grid user identities are mapped onto local user accounts statically
- User is authorised as a member of a single VO (no aggregation of roles)
- `grid-proxy-init`

VOMS

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



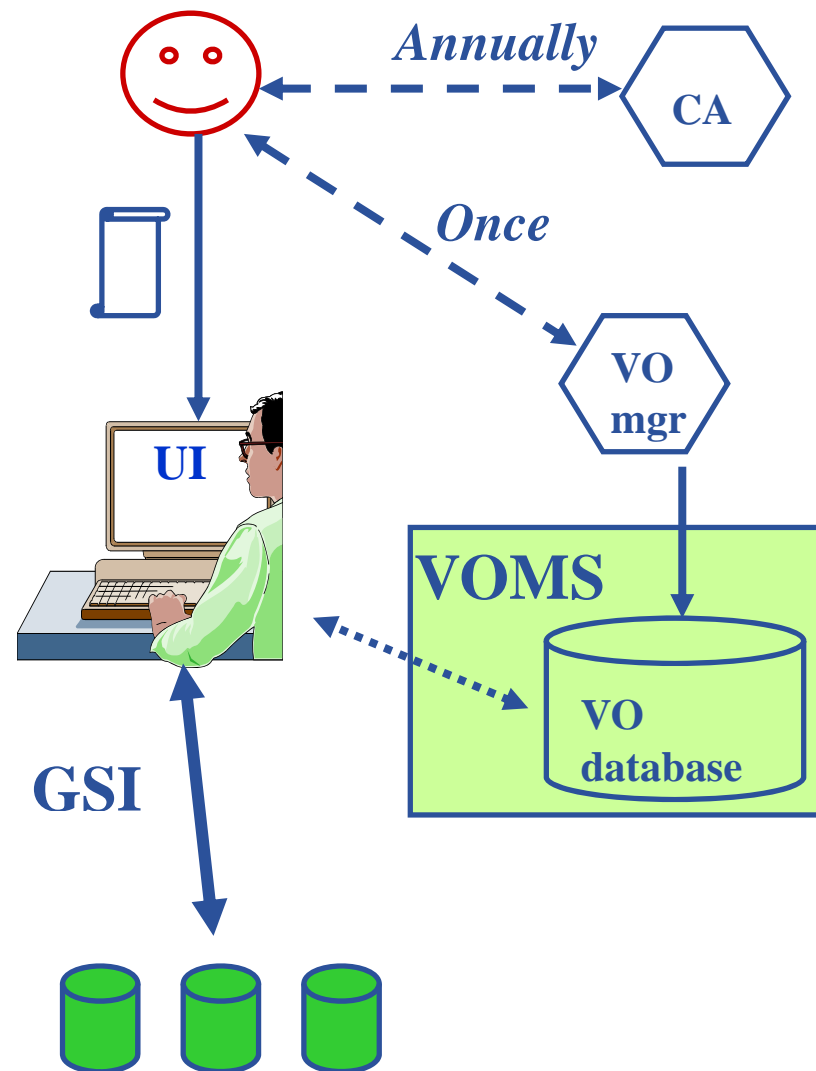
The grid user can perform those actions on the site that any user account from pool 3 is allowed to

- **Authentication**

- User obtains certificate from Certificate Authority
- Connects to UI by ssh and uploads certificate to UI
- or
- Login to a portal and use MyProxy
- Single logon to the Grid - create proxy
- then **Grid Security Infrastructure uses proxies**

- **Authorisation**

- User joins Virtual Organisation
- VO manager updates VOMS DB
- Capabilities added to proxy by VOMS

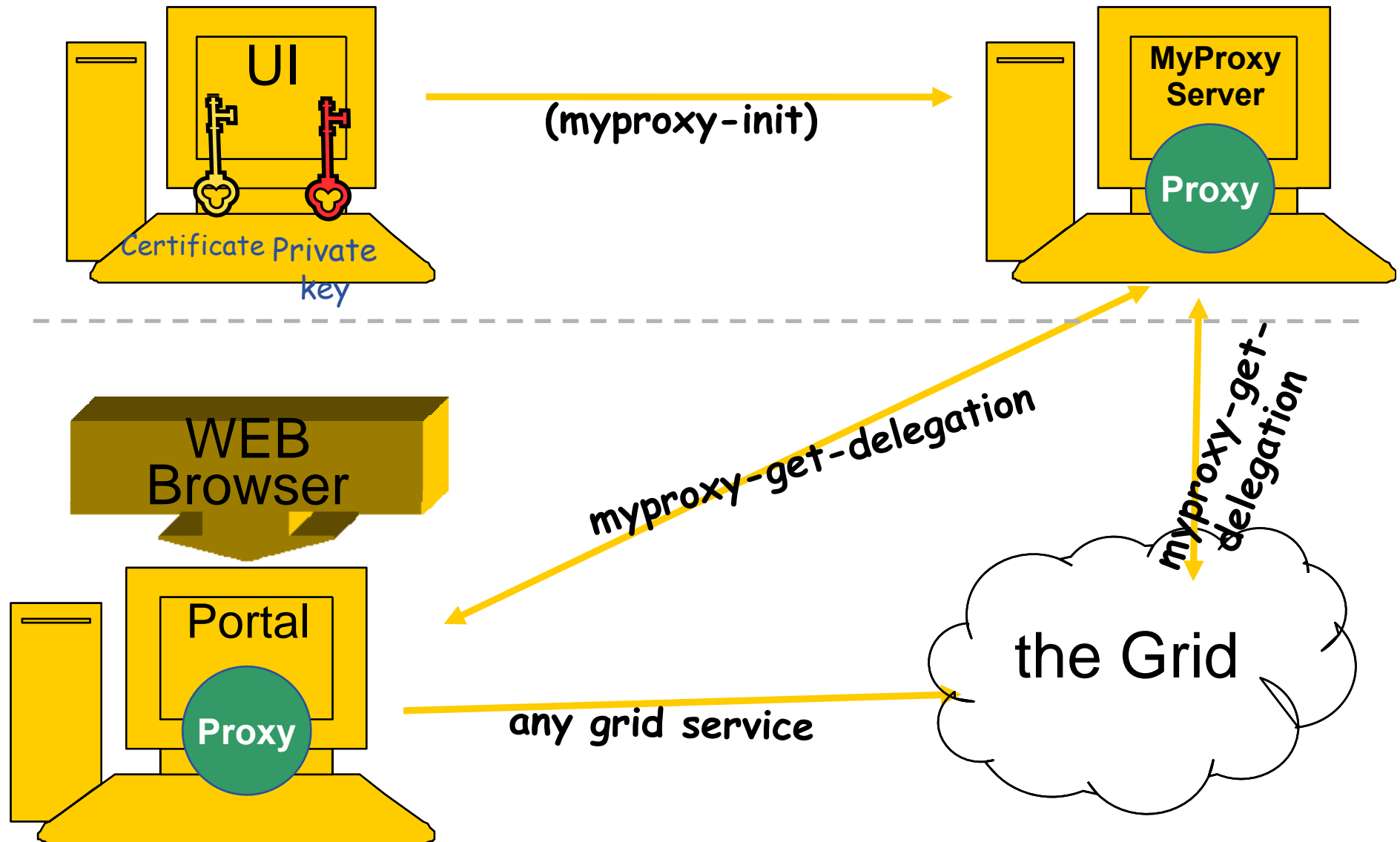


- Do not launch a delegation service for longer than your current task needs.

If your certificate *or delegated service* is used by someone other than you, it cannot be proven that it was not you.

- **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....
- **Solution: you can store a proxy in a “MyProxy server” and derive a proxy certificate when needed**
- **MyProxy ~ storage server for proxy files**

MyProxy example



- **Obtain a certificate from a recognized CA:**
 - www.gridpma.org → 1 year long, renewable certificates, accepted in every EGEE VO
- **Find and register at a VO**
 - EGEE NA4 - CIC Operations portal: <http://cic.gridops.org/>
- **Use the grid:**
 - **command line clients installed on the User Interface** (UI is maintained by the VO / your institute / you)
 - **voms-proxy-init –voms VONAME**
 - **voms-proxy-destroy**
 - **Use third party clients**
 - Might be satisfied with voms-proxy-init or require MyProxy
 - **Use programming APIs to interact with gLite services**
 - E.g. gfal for data management → later today



Enabling Grids for E-scienceE

Thank you!

Questions?

www.eu-egee.org



INFSO-RI-508833