

Security in gLite

Gergely Sipos

MTA SZTAKI

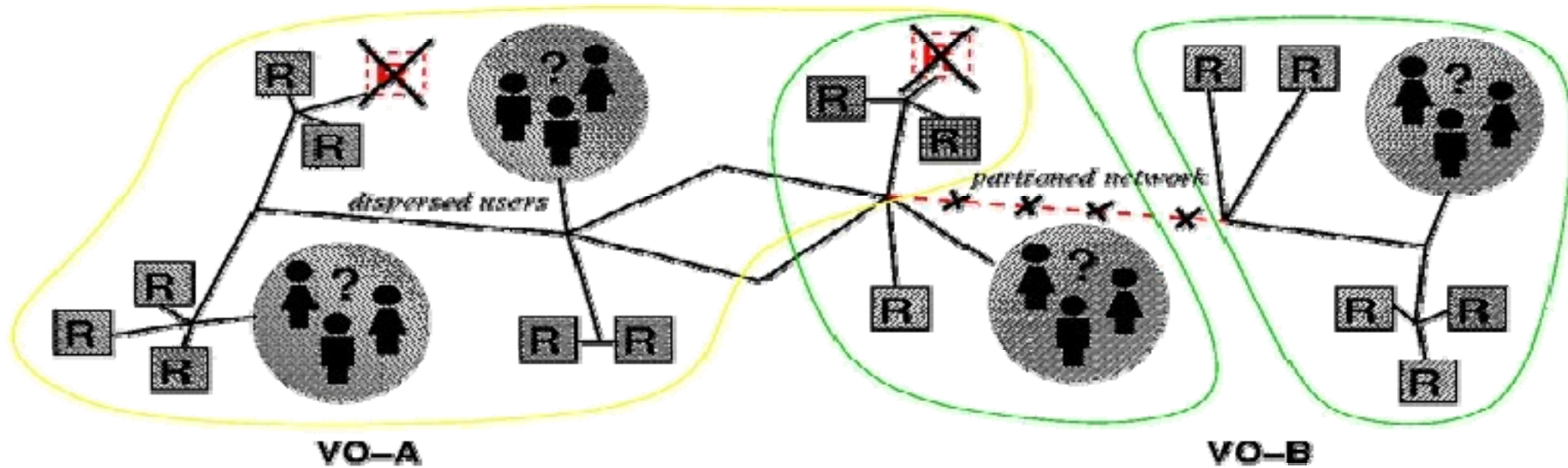
sipos@sztaki.hu

With thanks for some slides to EGEE and Globus colleagues

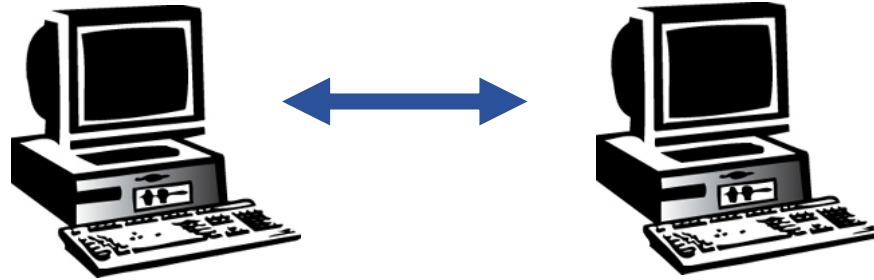
The Grid problem is to enable “coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations.”

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

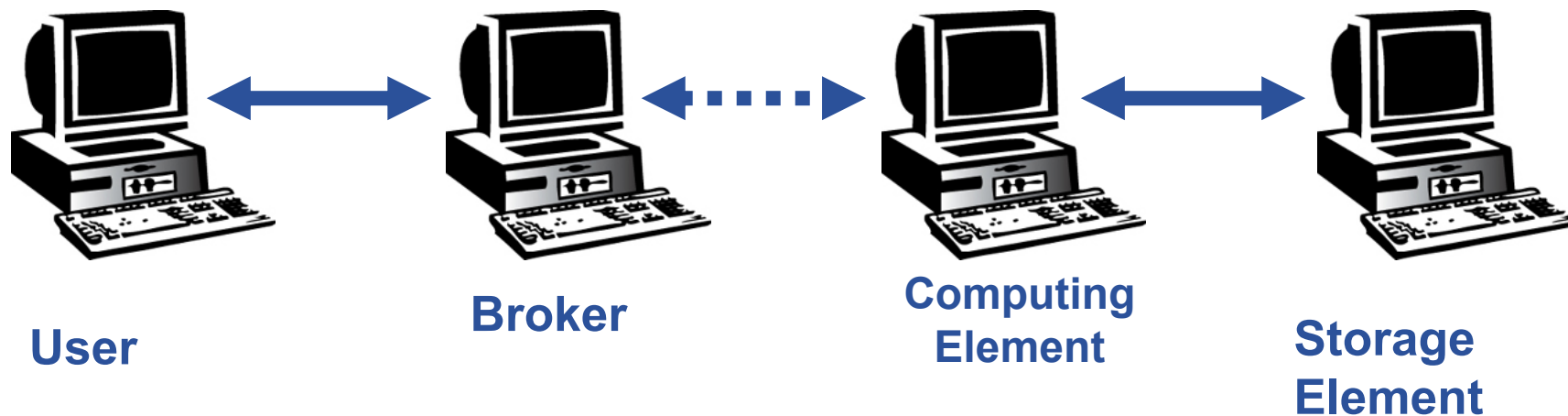


User

Grid service

VO members communicate over the Internet

- **How can communication endpoints be identified?**
 - Authentication
- **How can a secure channel established between two partners?**
 - Encryption
 - Non-repudiation
 - Integrity



- **Who can belong to a VO? Who cannot not?**
- **What are VO members allowed to do?**
 - Authorization
- **How can services act on behalf of a user?**
 - How can a broker access the user's sites?
 - How can a job started by the broker access the user's private data?

- **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.



Enabling Grids for E-scienceE

Grid Security Infrastructure

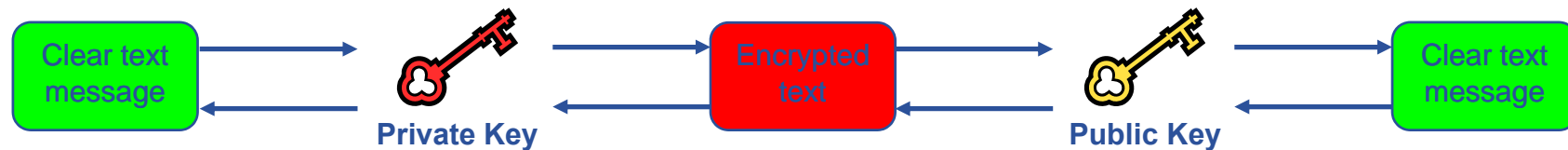
www.eu-egee.org



Grid Security Infrastructure

Security at network level:
Public key infrastructure (PKI)

- **Asymmetric encryption...**



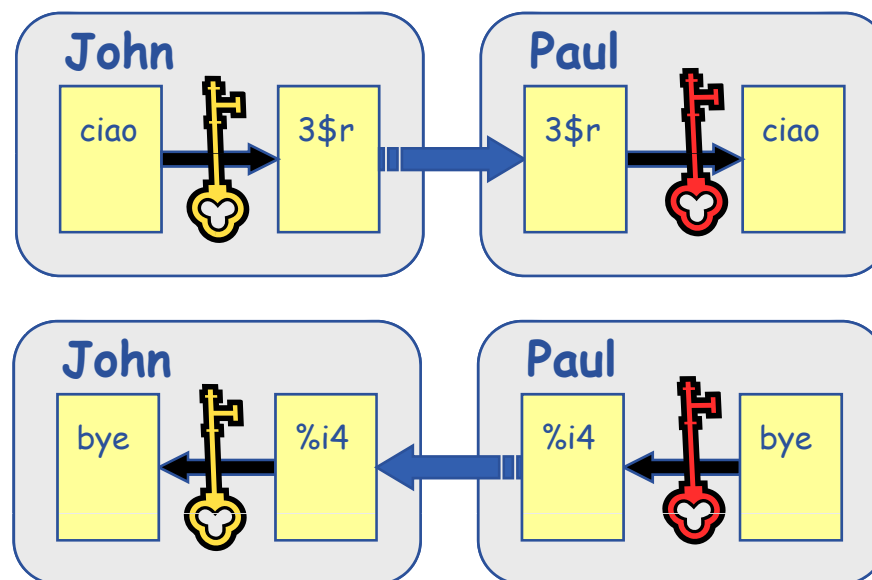
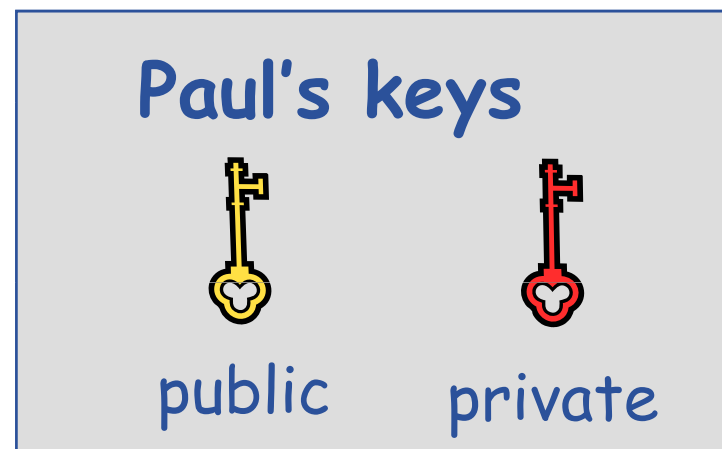
- **.... and Digital signatures ...**

- 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**

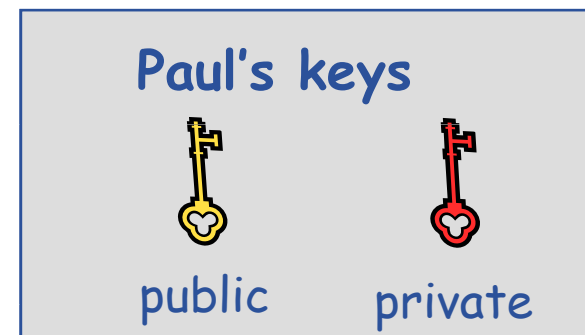
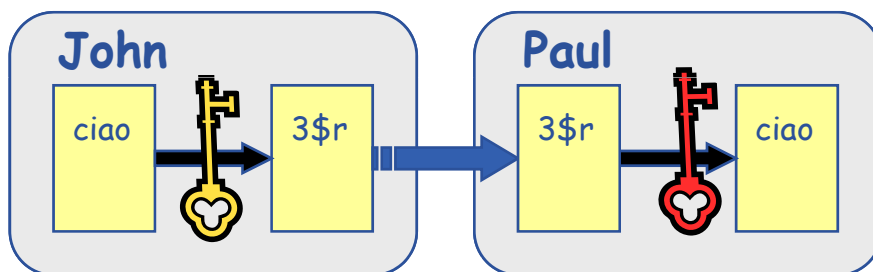
- That a user / site is who they say they are
- And can be trusted to act in accord with agreed policies

- Every networked entity (user/machine/software) is assigned with two keys: one **private key** and one **public key**
 - 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;



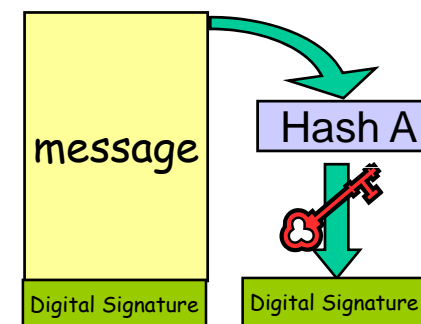
- **Encryption**

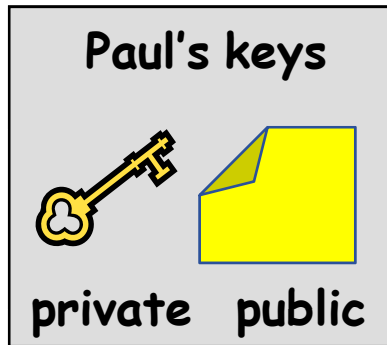
- Encryption with recipient's public key
- Only recipient can decrypt the message



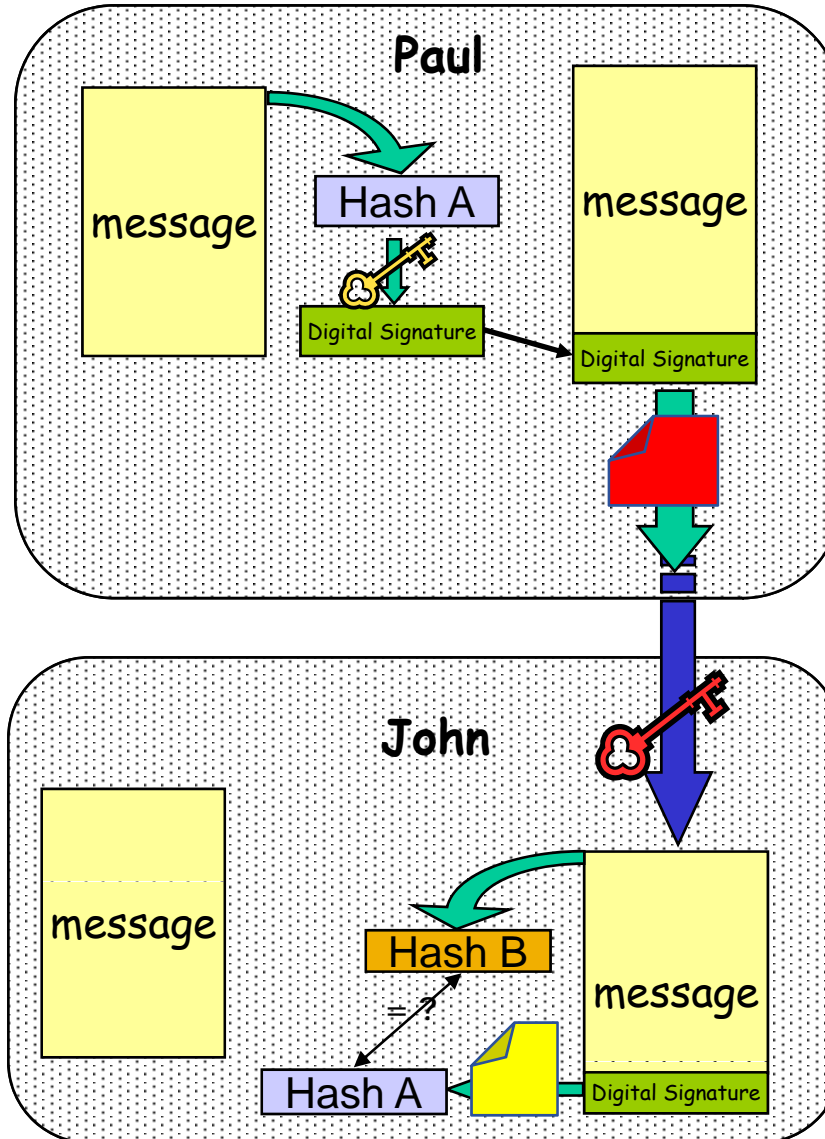
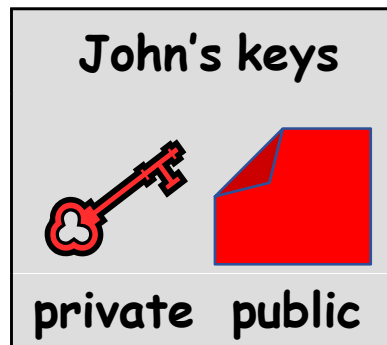
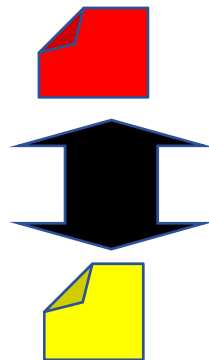
- **Non-repudiation**

- **Naiv approach:** encrypt message with sender's private key
 - Too costly for long messages
- **Solution:**
 - generate hash of the message
 - Encrypt hash with sender's private key
 - Attach encrypted hash to message → **Digital signature**
- Additional benefit: Integrity (hash is constant)

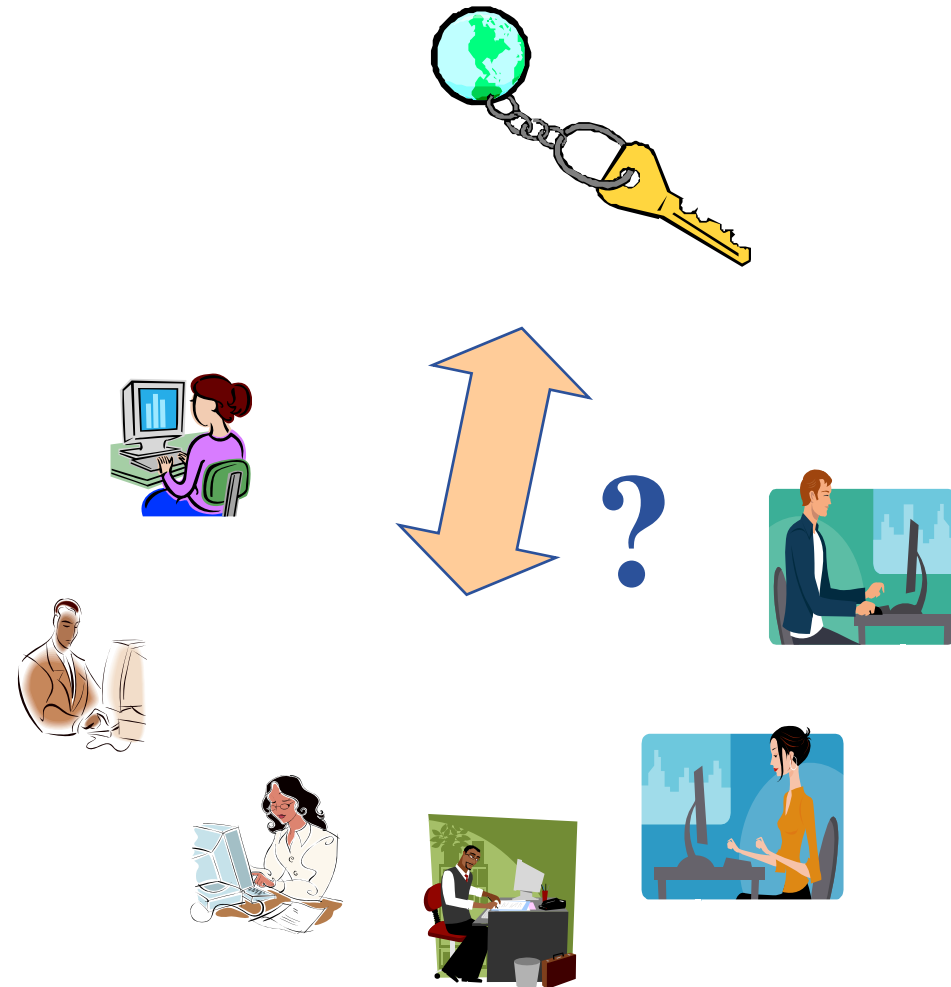




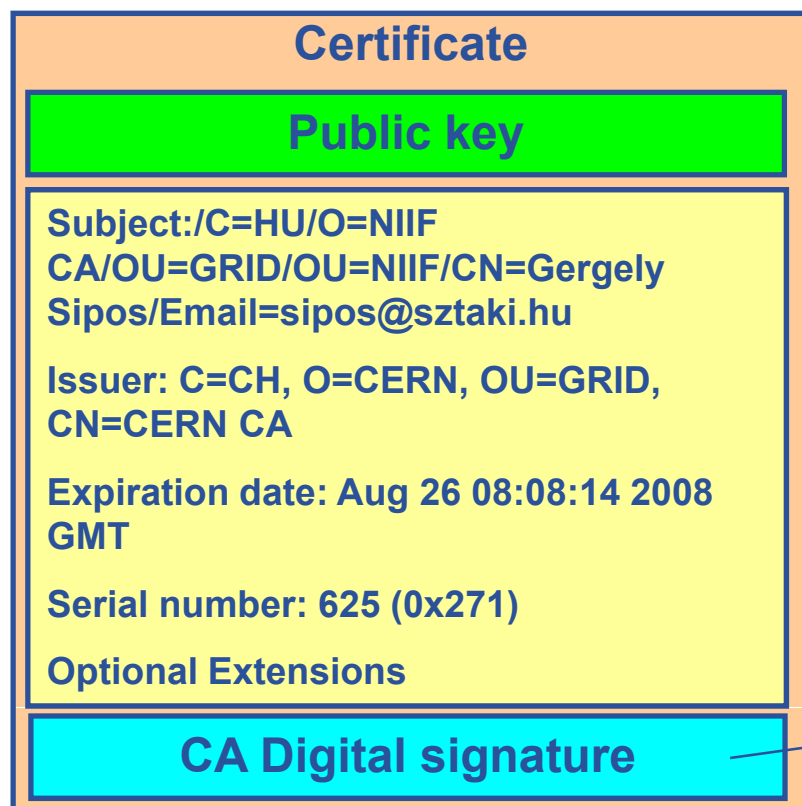
Mutual authentication and exchanging public keys: SSL protocol



- Since I'm the only one with access to my private key, you know I signed the data associated with it
- But, how do you know that you have my correct public key?



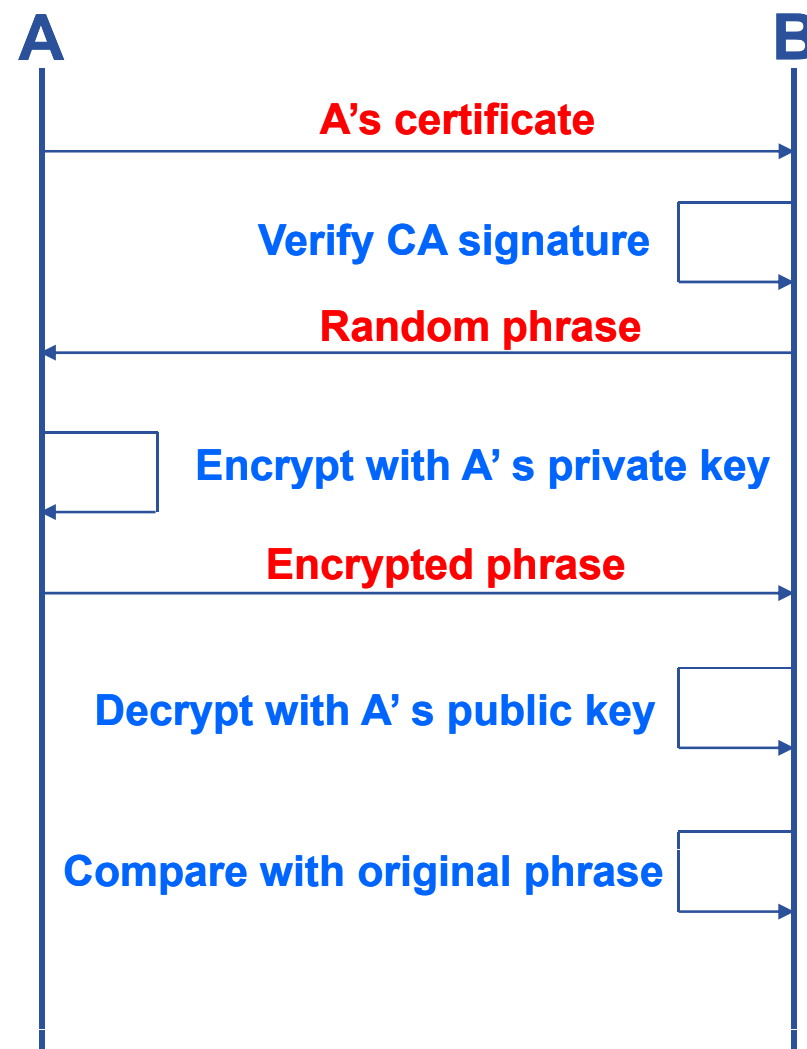
- Public key is wrapped into a “certificate file”
- Certificate files are created by trusted third parties: Grid Certification Authorities (CA)
- Private key is stored in encrypted file – protected by a passphrase
- Private key is created by the grid user



1. Hash of Public key & metadata,
2. Encrypt hash with CA's private key

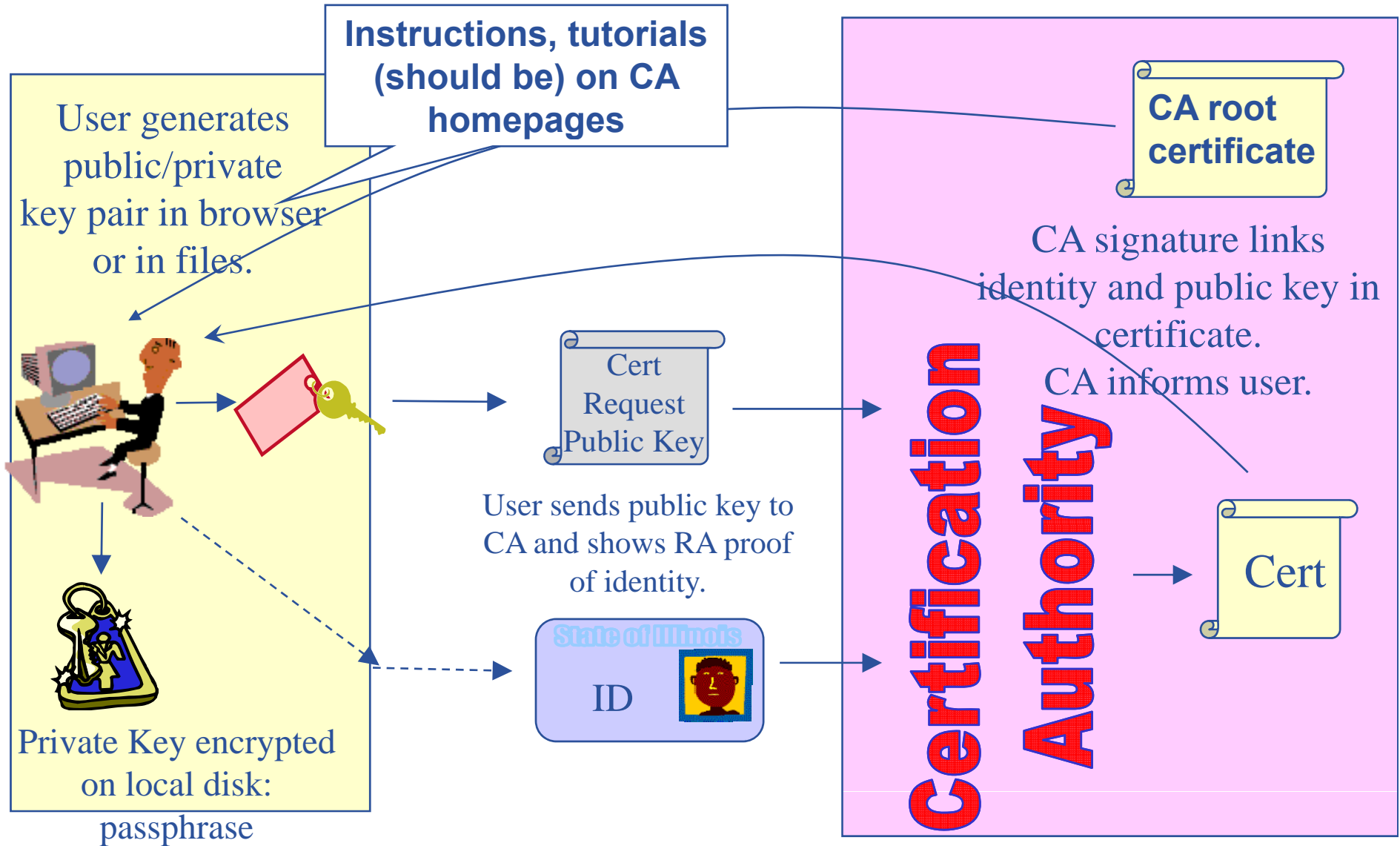
Based on X.509 PKI:

- 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



- **Grid users' must generate private and public key**
- **Public key must be signed by a recognized CA**
 - CAs can establish a number of people “registration authorities” RAs: Personal visit to the nearest RA instead of the national CA
- **CAs recognized by EGEE: <http://www.gridpma.org/>**
 - Per continent
 - Per country
 - *Per region*

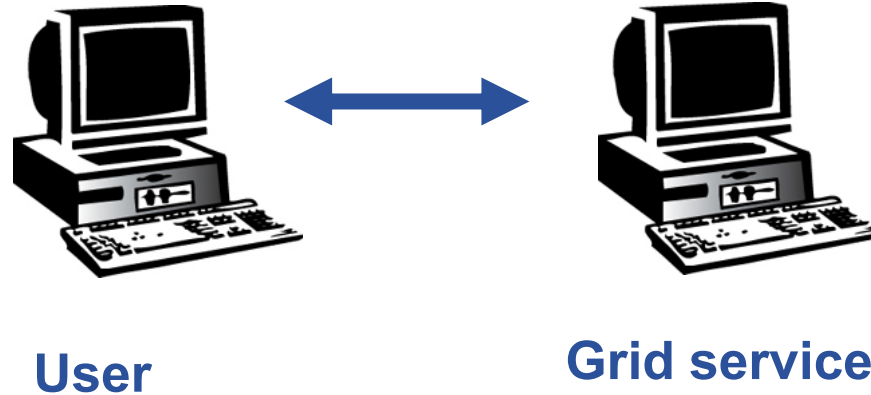
Issuing a grid certificate



- **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.**
- **Private key and certificate can:**
 - Stored in your browser
 - Stored in files using different file format (PEM, P12, ...)
- **Typical situation on Globus, gLite, ARC middleware based grids:**

```
[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.

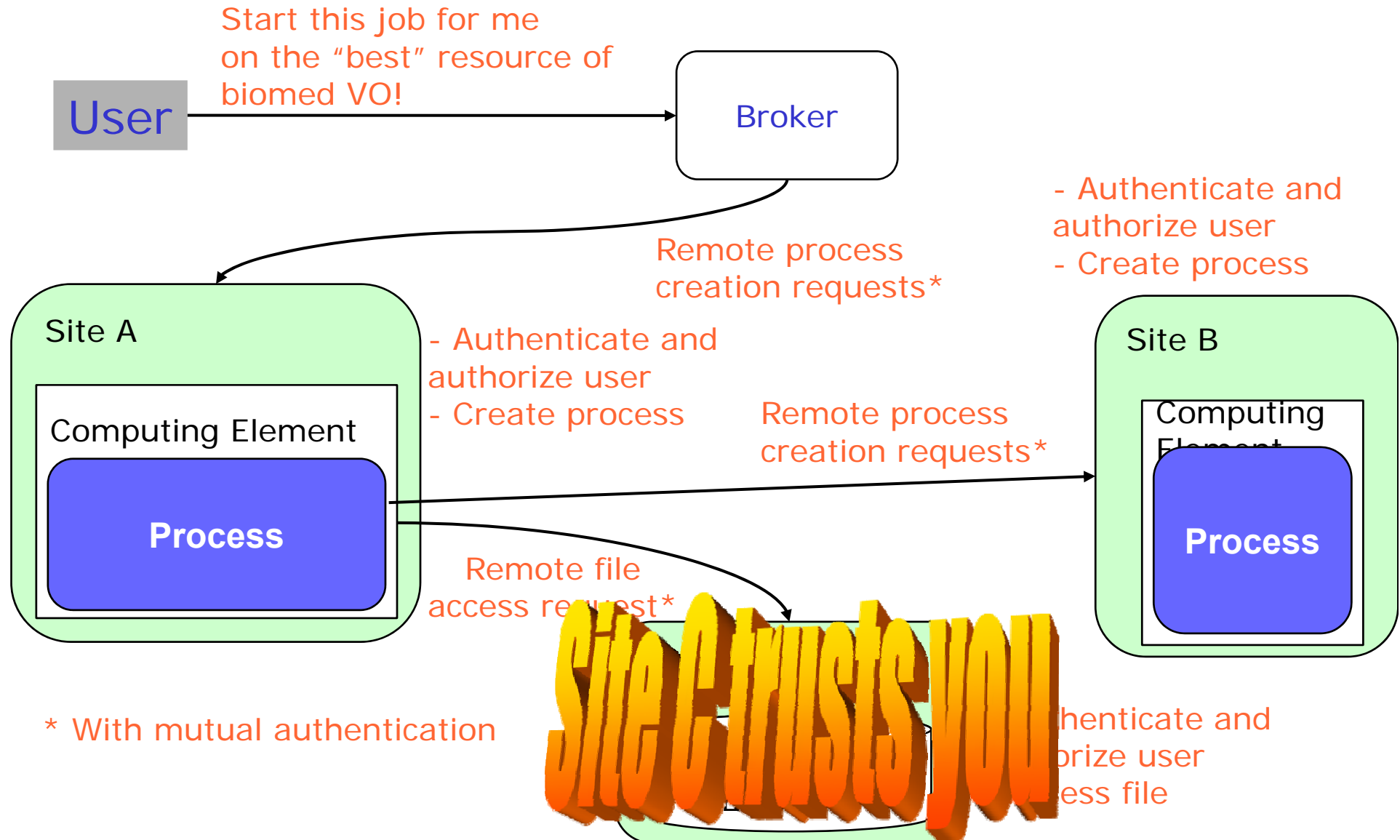


VO members communicate over the Internet

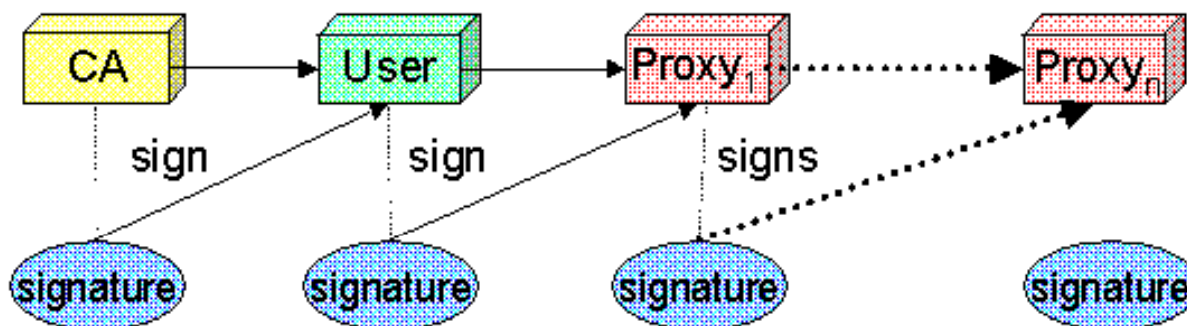
- **How can communication endpoints be identified?** ✓
 - Authentication
- **How can a secure channel established between two partners?**
 - Encryption ✓
 - Non-repudiation ✓
 - Integrity ✓

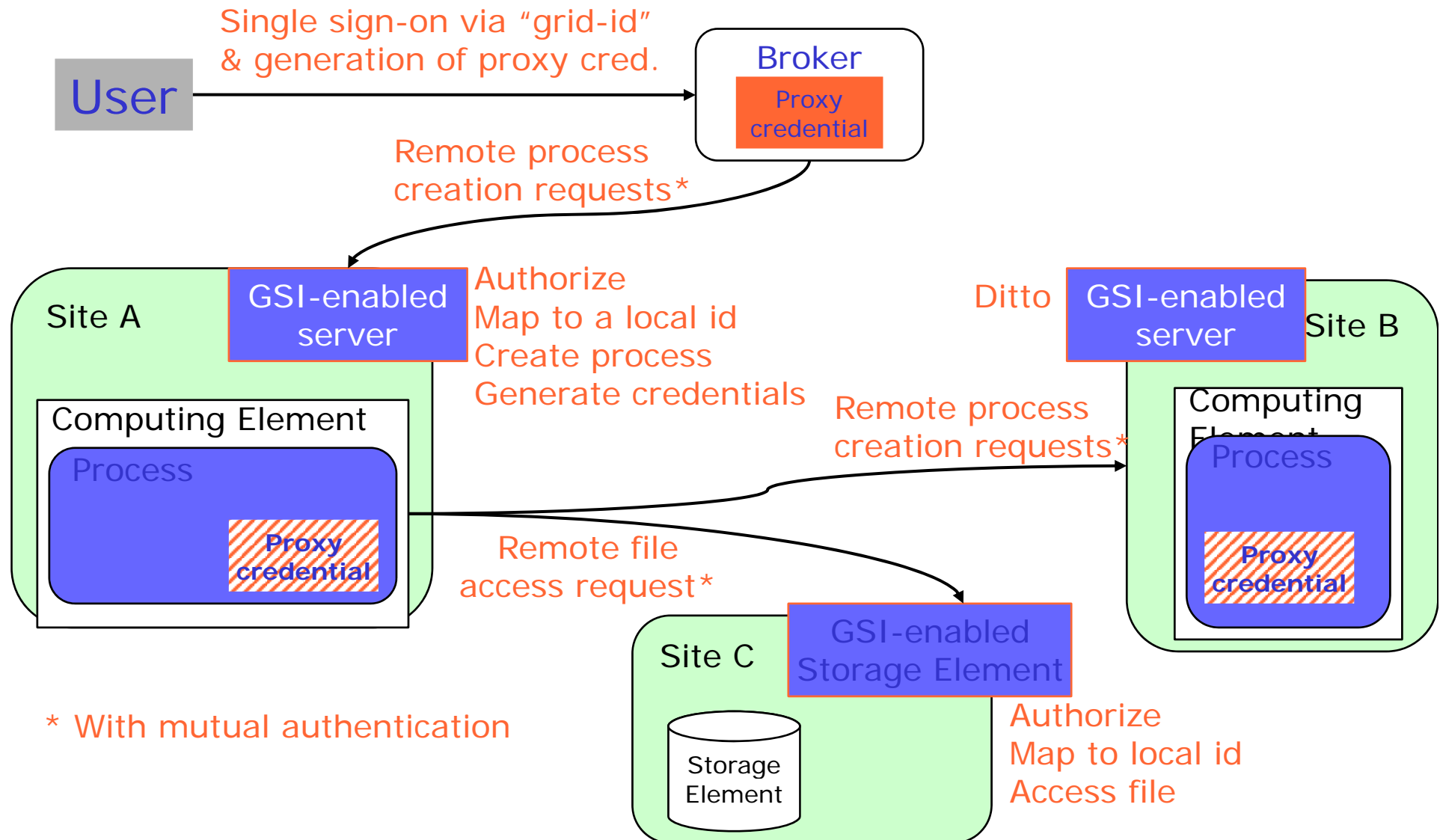
Grid Security Infrastructure

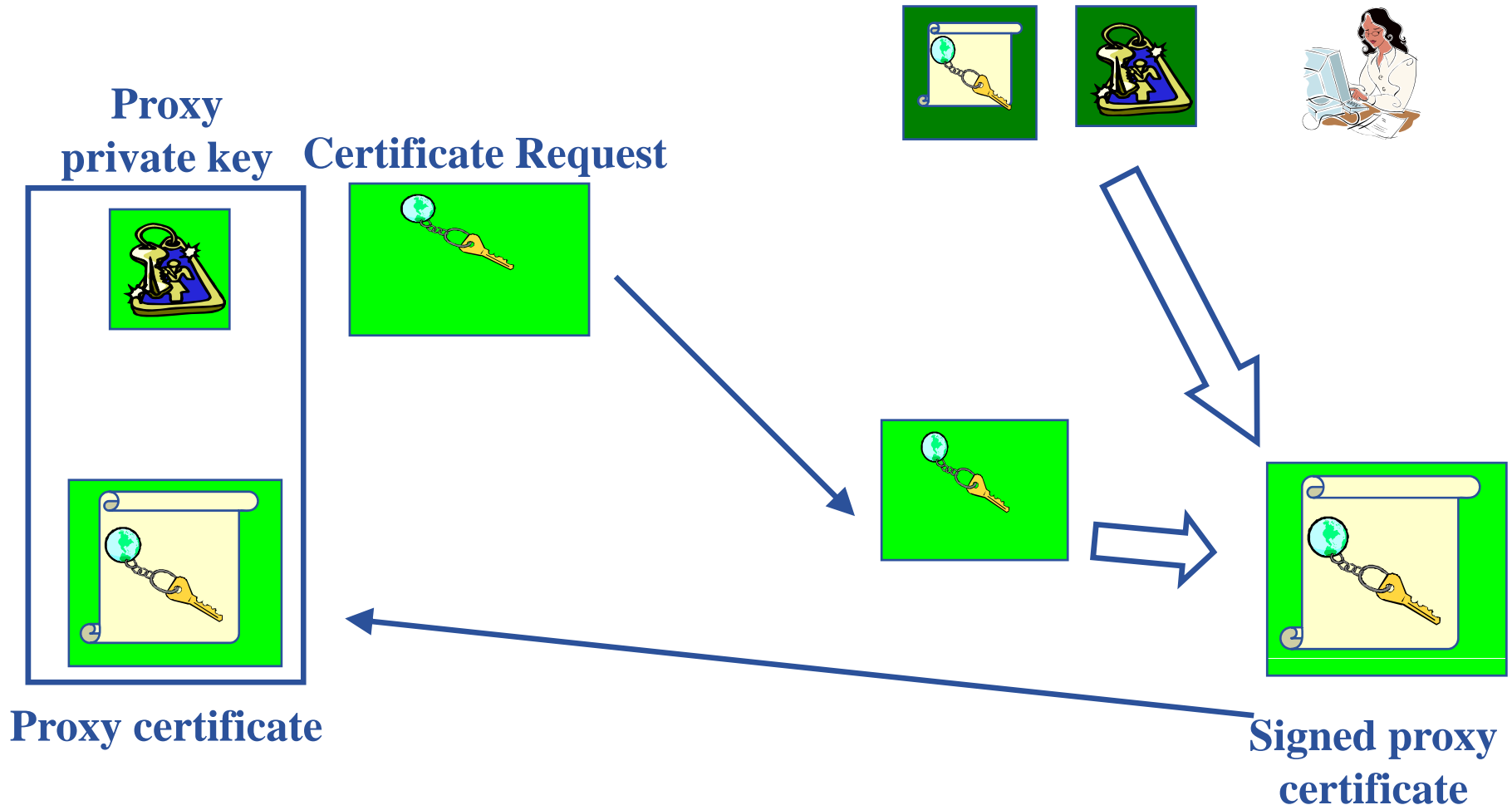
Security at VO level



- **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 private key–certificate pair from the user’s private key–certificate.**
 - New key-pair is a single file: **Proxy credential**
 - Proxy private key is not protected by password
 - Proxy may be valid for limited operations
 - Proxy has limited lifetime
- **The client can delegate proxies to services, 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

% voms-proxy-destroy → logout from the grid

Delegated credentials will not be revoked

User Interface



create proxy



Authorization Service
(VO Management Service)

Submit job
Retrieve status & output

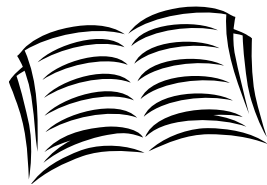
Resource Broker



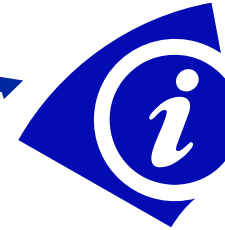
query

Retrieve status & output

File and Replica Catalog



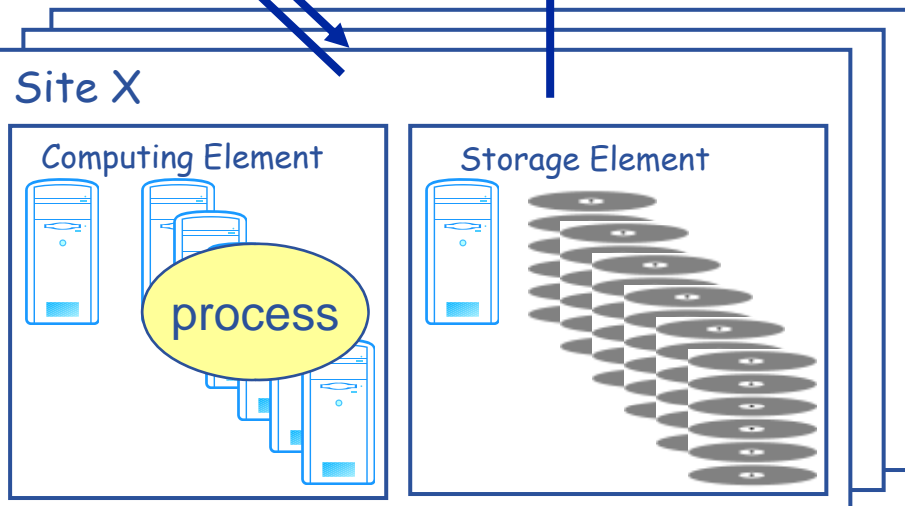
Information System



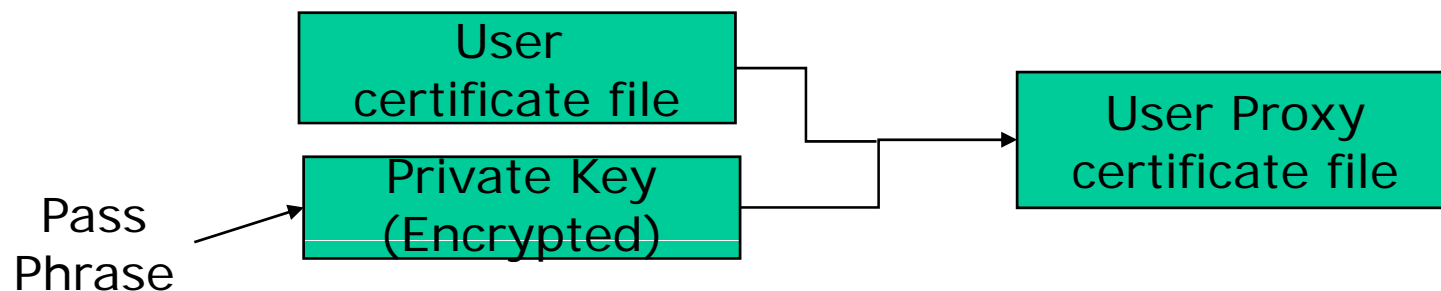
query

Submit job

publish state



- 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



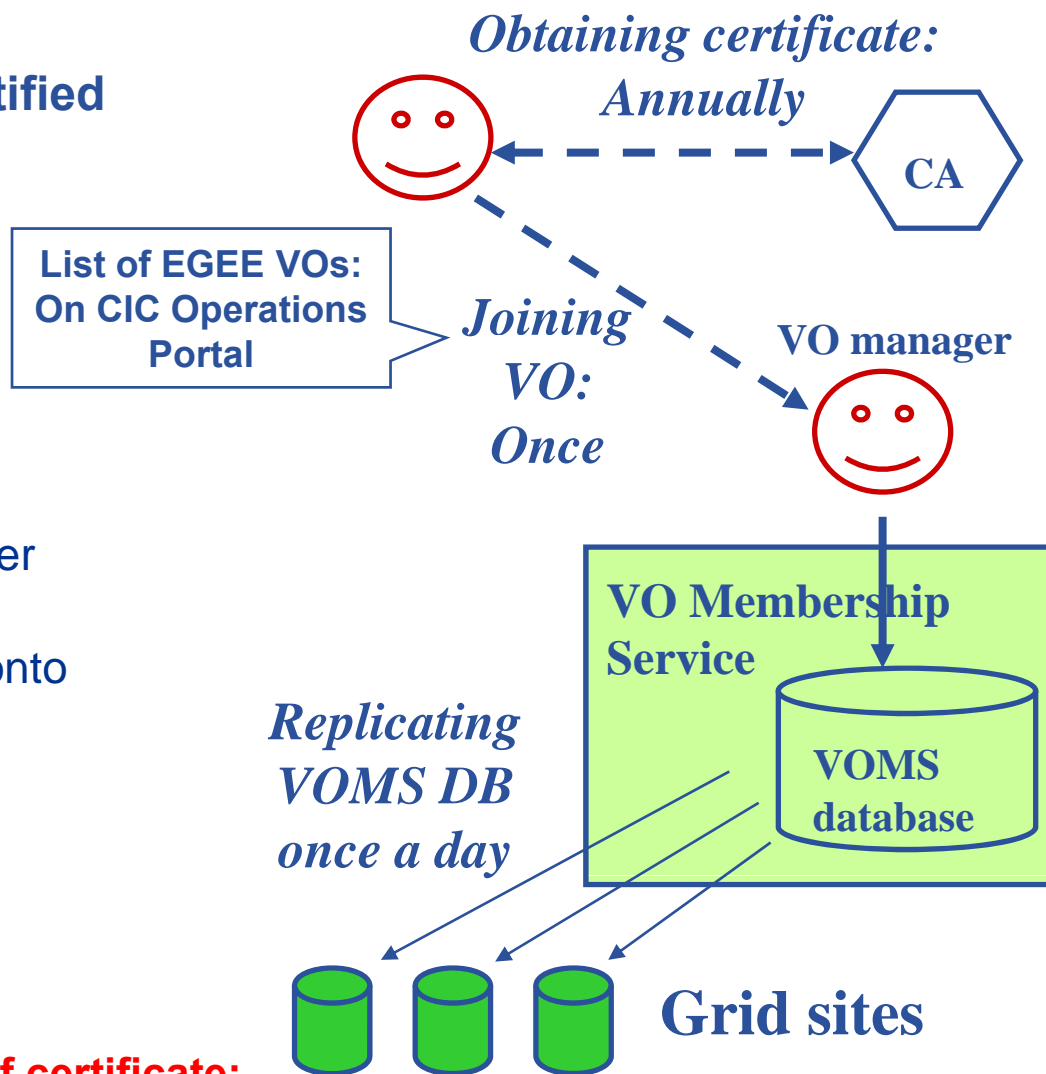
- Proxy file saved in `/tmp`
 - the private key part of the Proxy is *not* encrypted:
 - proxy lifetime is short (typically 12 h) to minimize security risks.
- NOTE: No network traffic during proxy creation!

- **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

- Users (and machines) are identified by certificates.

Steps

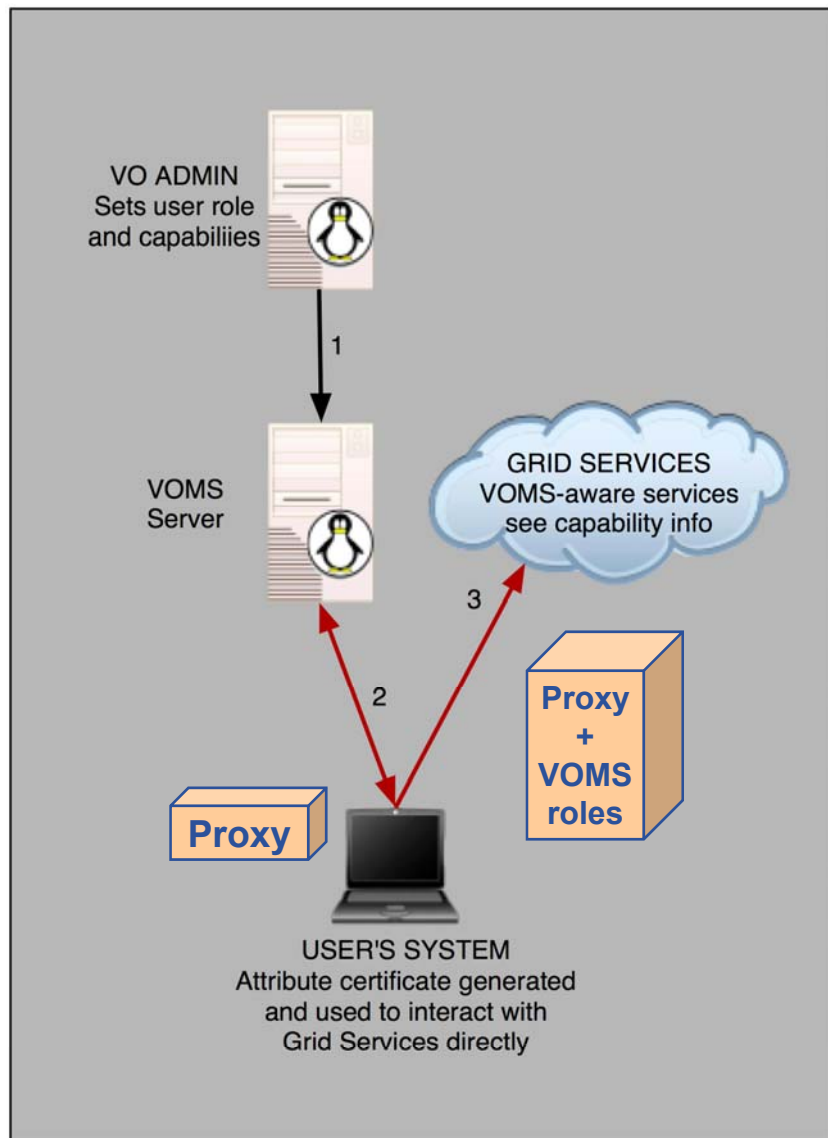
- User obtains certificate from Certification Authority
- User registers at the VO
 - usually via a web form
- VO manager authorizes the user
 - VO DB updated
- User information is replicated onto VO resources within 24 hours



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

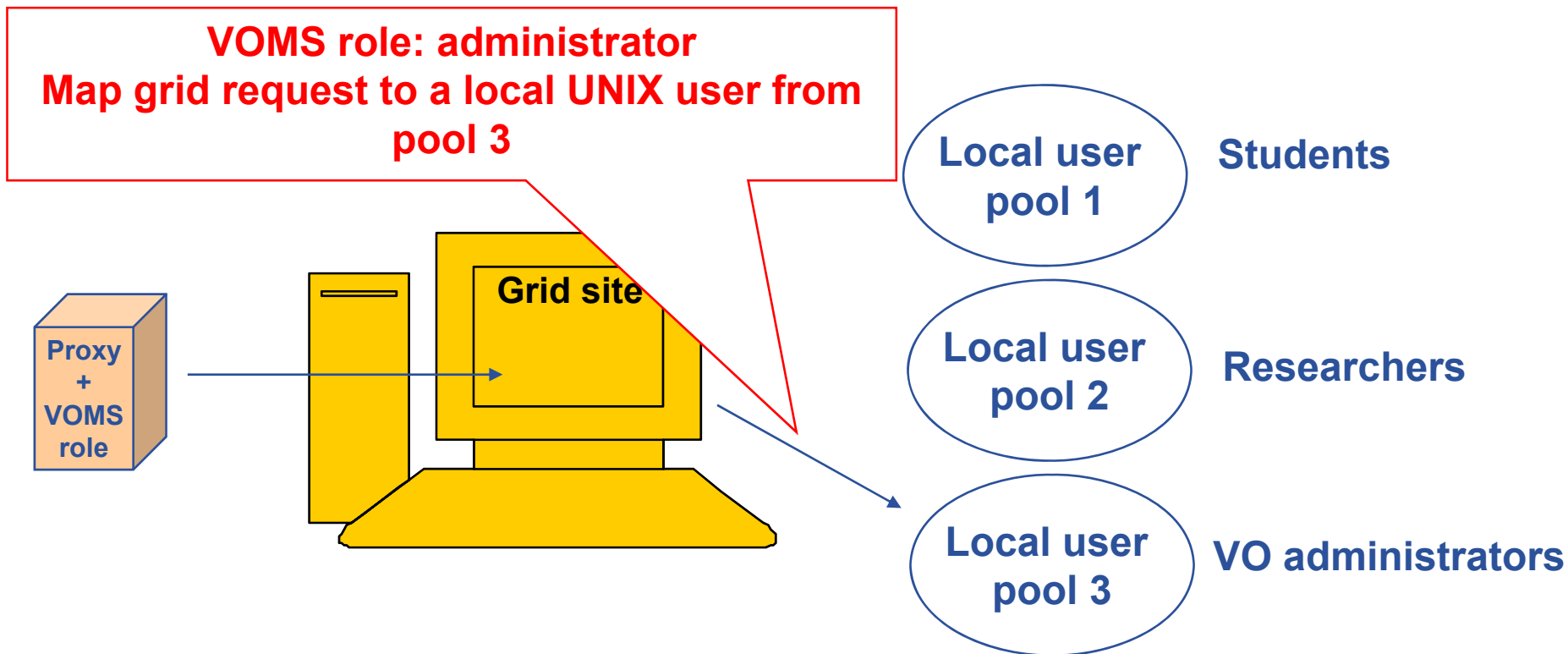
- **VO can have groups**
 - Different rights for each
 - Different groups of experimentalists
 - ...
 - Nested groups
- **VOMS has roles**
 - Assigned to specific purposes
 - E,g. system admin
 - When assume this role
- **VO members belong to one/more groups and can have extra roles**



- **voms-proxy-init**
 - Creates a proxy locally
 - Contacts the VOMS server and extends the proxy with a role
 - VOMS server signs the proxy

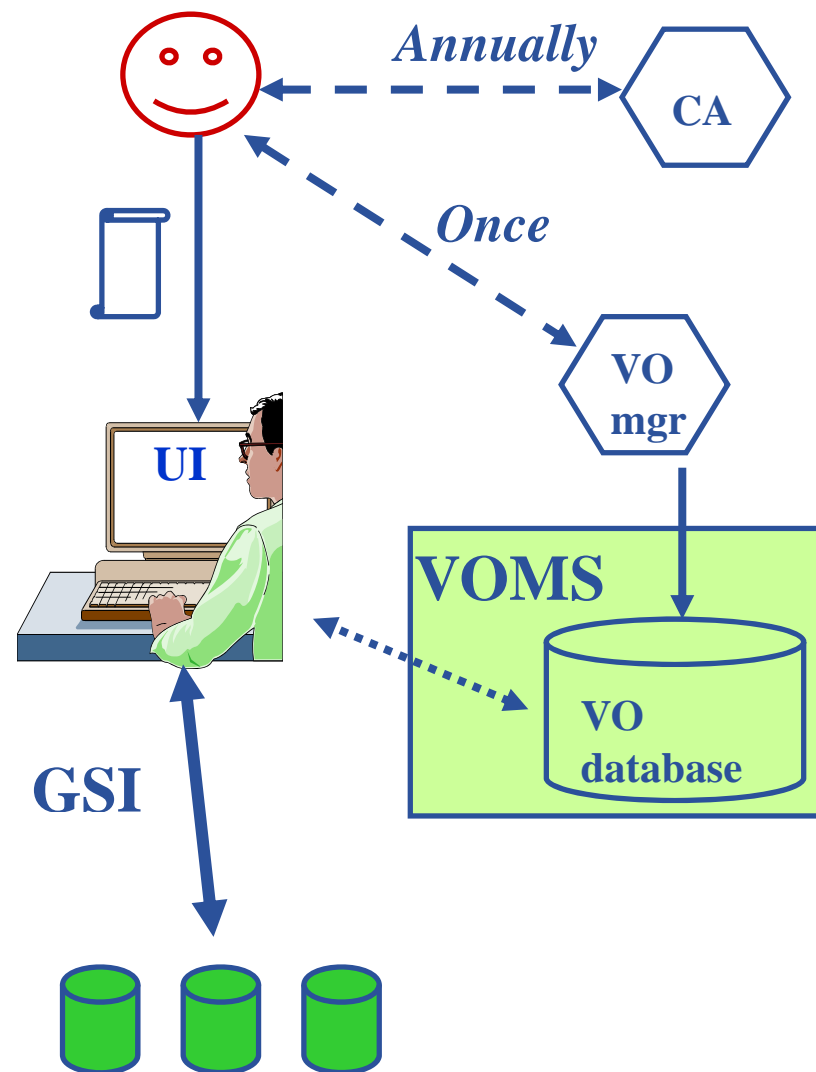
`voms-proxy-init -voms gilda`

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



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

- **User obtains certificate from Certificate Authority**
 - Import it into your browser
 - Have it on a User Interface
- **User selects and joins VO**
- **User connects to UI by ssh**
 - Create proxy
 - Submit jobs, manage files, ...

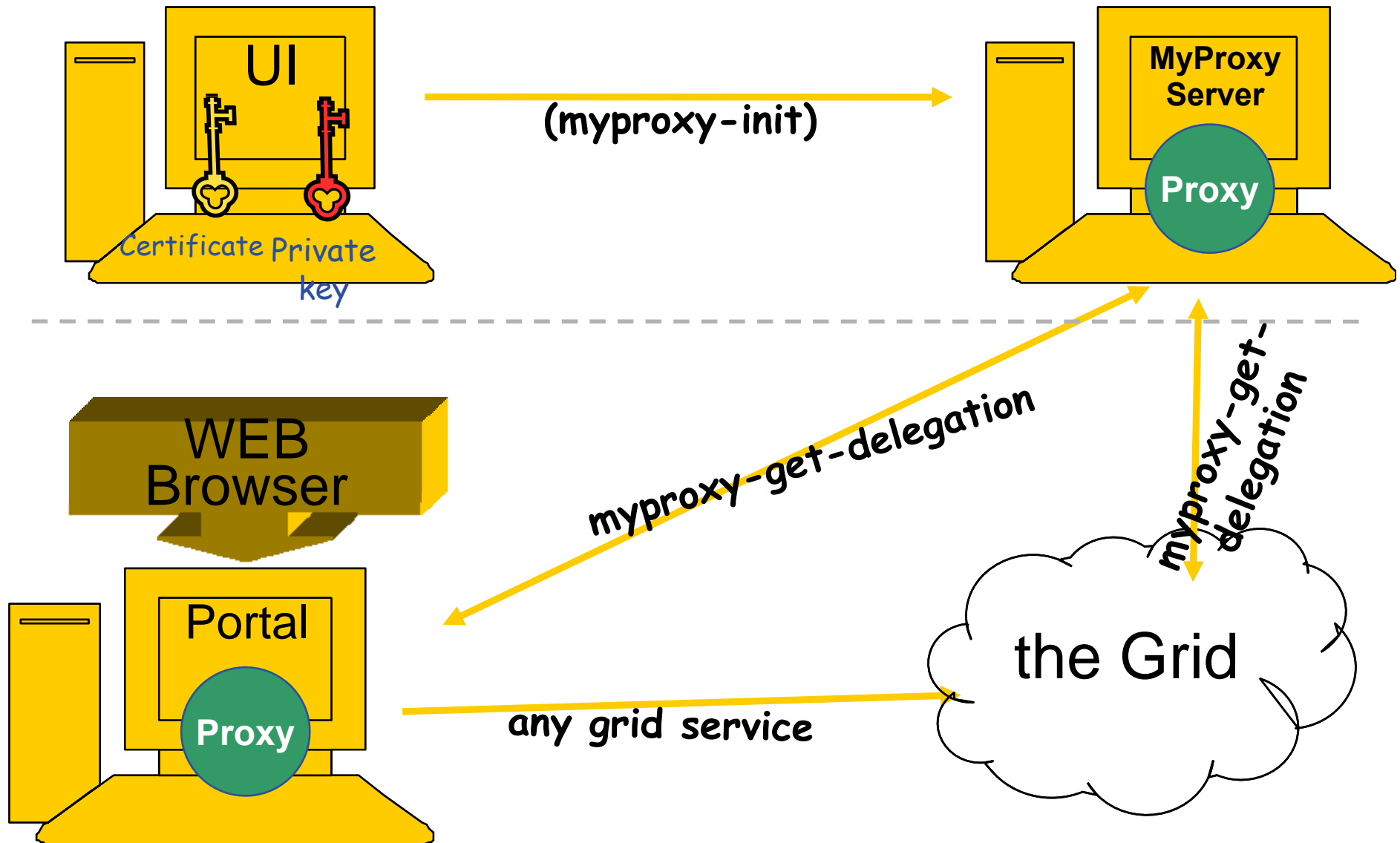


- Do not launch a delegation service for longer than your current task needs.

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....
 - Your job may need a long proxy
 - And you should keep the lifetime of delegated proxy short
- **Solution: you can store a long term proxy in a “MyProxy server” and derive a short living 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 and APIs installed on the User Interface**
(UI is maintained by the VO / your institute / you)
 - **voms-proxy-init –voms <voName>**
 - **voms-proxy-destroy**
 - **Portals**
 - Might require MyProxy

1. Basics:

- Investigate your certificate
- Create proxy with voms extension
- Investigate your proxy
- Destroy your proxy

2. Certificates and the web:

- Import your certificate into your web browser
- Access a protected web site: SAM Grid monitor

3. MyProxy:

- Upload a proxy into MyProxy server
- Delegate a proxy from MyProxy server



Enabling Grids for E-scienceE

Thank you!

Questions?

www.eu-egee.org

