



Enabling Grids for E-scienceE

Authorisation and Authentication in gLite

Mike Mineter

National e-Science Centre, Edinburgh

Taipei, 22 August 2005

www.eu-egee.org



Information Society

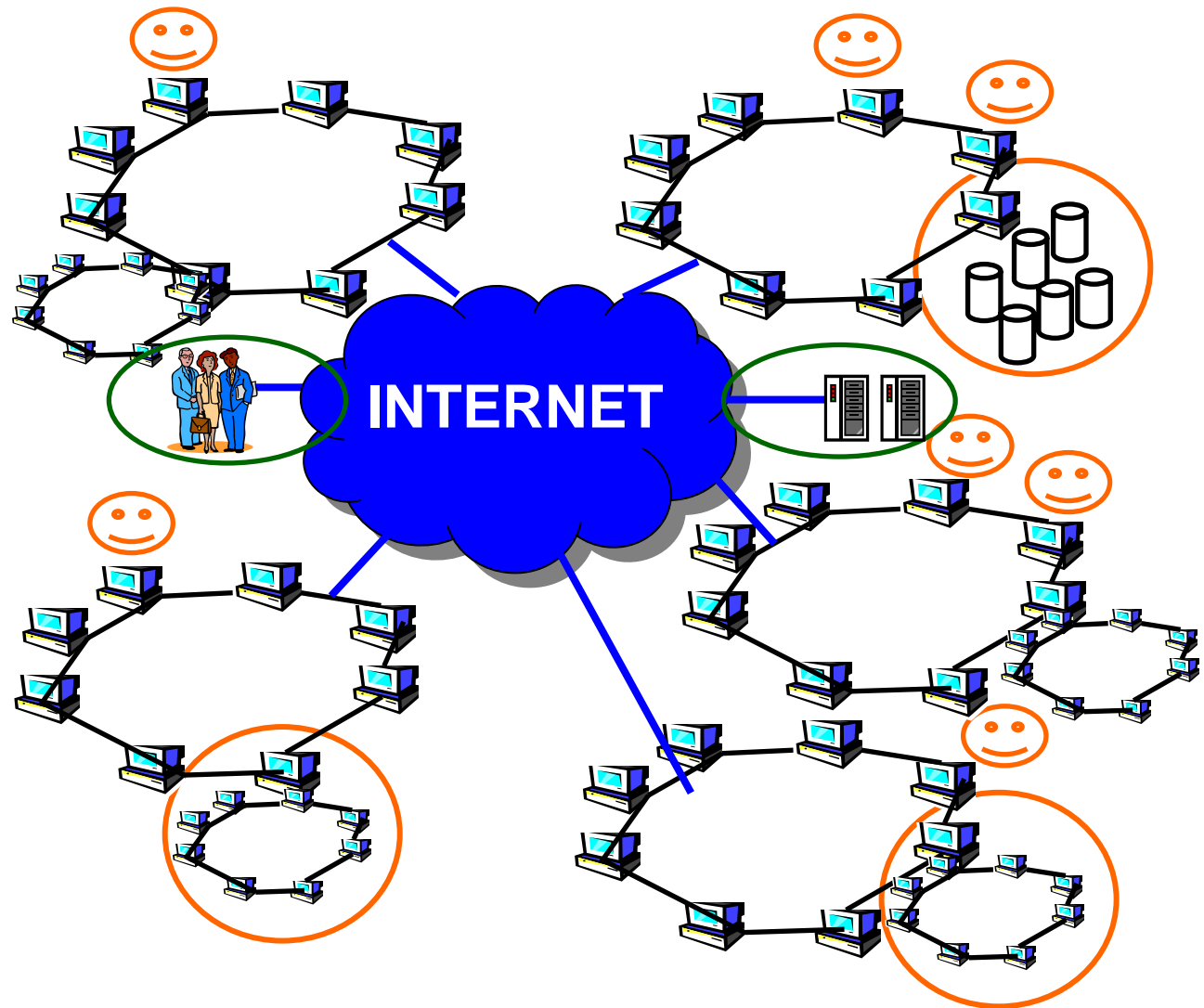


- **This presentation includes slides taken from the following talks:**
 - Roberto Barbera at ISSGC05, Vico Equense, July 2005
<http://www.dma.unina.it/~murli/GridSummerSchool2005/index.htm>
 - Richard Sinnott at ISSGC05, Vico Equense, July 2005
 - Carl Kesselman, at ISSGC04, Vico Equense, July 2004
<http://www.dma.unina.it/~murli/GridSummerSchool2004/index.htm>
 - David Fergusson at EMBRACE/EGEE Tutorial, Clermont Ferrand, July 2005 <http://agenda.cern.ch/fullAgenda.php?ida=a053765>
 - Joachim Flammer at EMBRACE Tutorial, Clermont-Ferrand, July 2005
- **Also information from:**
 - Globus Alliance: GT4 Security: Key concepts
 - <http://www.globus.org/toolkit/docs/4.0/security/key>

- **Why Authorisation and Authentication (AA) are the basis of grids**
- **Authentication: “AuthN”**
 - “Are you who you claim to be?”
- **Authorisation: “AuthZ”**
 - “What are you allowed to do?”
- **MyProxy: management of certificates**

Importance of AuthN and AuthZ

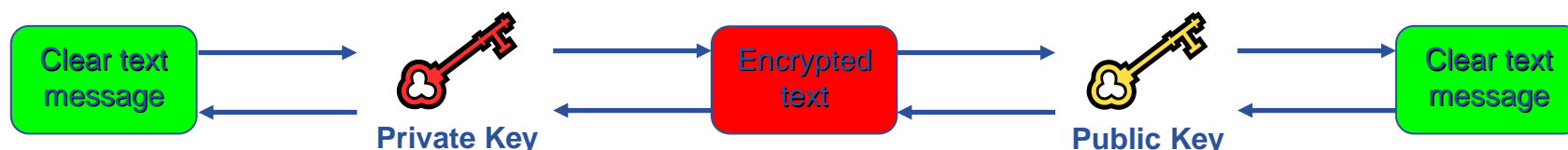
- **Collaboration across**
 - Administrative domains
 - National borders
 - Via unprotected Internet
- **Single sign-on**
 - Multiple services
- **Scalability:**
 - N,000 users
 - M,000 CPUs
 - (N*M) million usernames / passwords ???!
- **Security**
- **Delegation**
- **Non-repudiation**



Authentication and X.509 certificates

- **X 509 Digital certificate is basis of AA in EGEE**
- **national *Certification Authorities* (CAs)**
 - *mutually recognized* – to enable international collaboration
 - International Grid Trust Federation <http://www.gridpma.org/>
 - Taiwan: Academia Sinica Grid Computing Certification Authority <http://ca.grid.sinica.edu.tw/>
- **CA certificates:**
 - User's identity: you get a Certificate and use it to access a grid
 - Resources have a site certificate
- **Uses Public Key Infrastructure**
 - Private key – known only to you
 - Public key included in your certificate

- Provides authentication, integrity, confidentiality, non-repudiation
- Asymmetric encryption

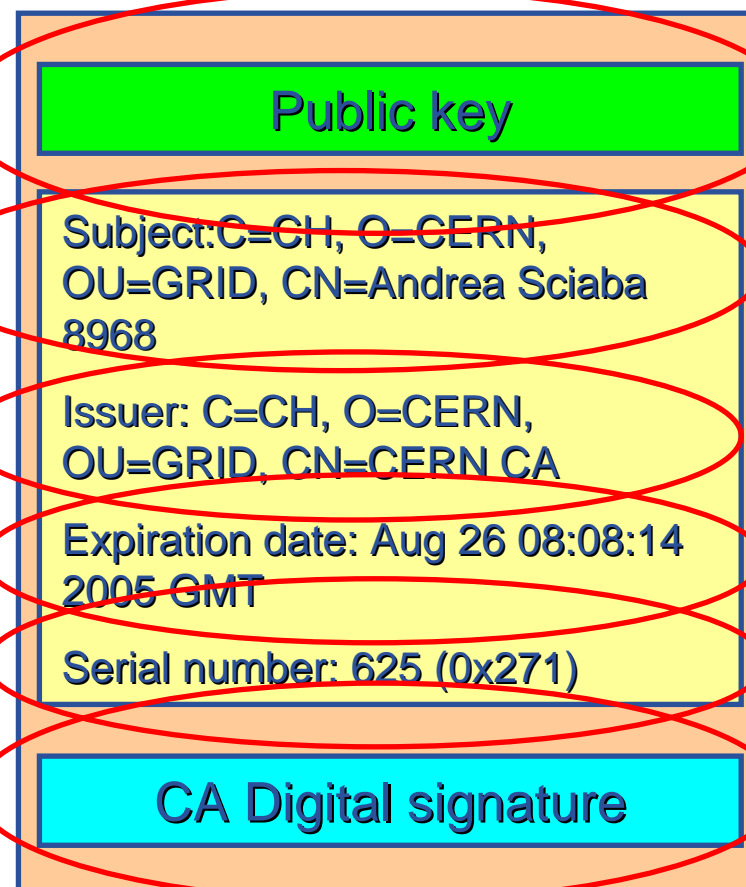


- **Digital signatures**
 - A hash derived from the message and encrypted with the signer's private key
 - Signature checked decrypting with the signer's public key
- **Allows key exchange in an insecure medium using a trust mode**
 - Keys trusted only if signed by a trusted third party (Certification Authority)
 - A CA certifies that a key belongs to a given principal
- **Certificate**
 - Public key + principal information + CA signature
 - X.509 format most used

- **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

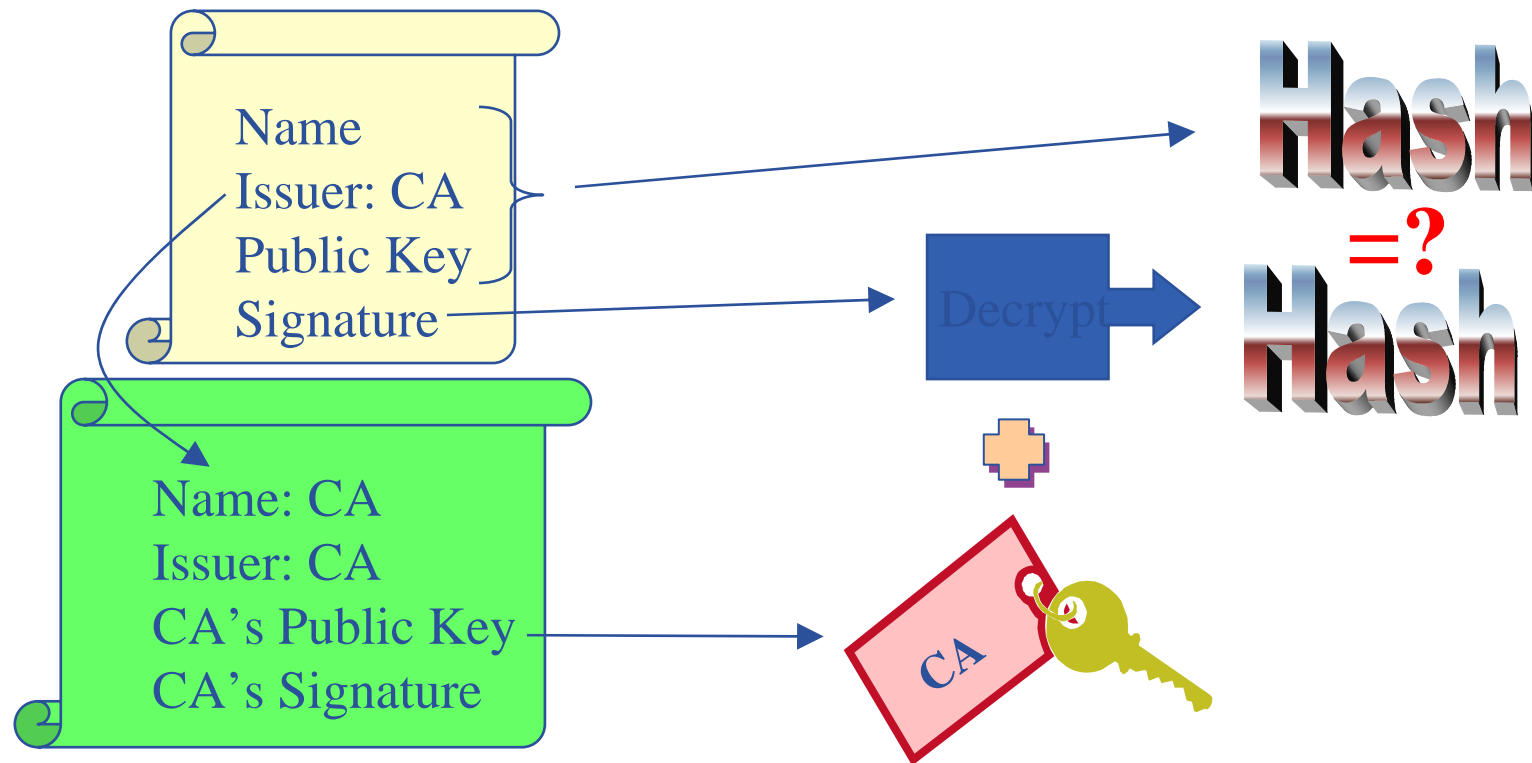


- **Keep your private key secure.**
- **Do not loan your certificate to anyone.**
- **Report to your local/regional contact if your certificate has been compromised.**
- **Do not launch a delegation service for longer than your current task needs.**

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

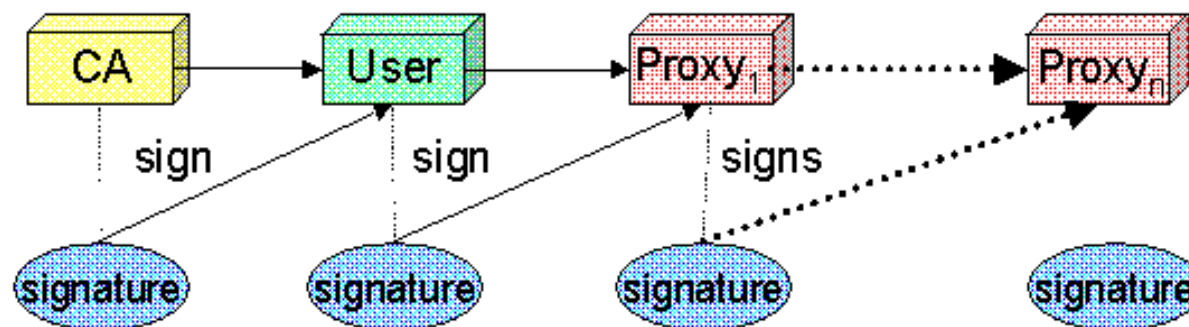
IT IS YOUR PASSPORT AND CREDIT CARD

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



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

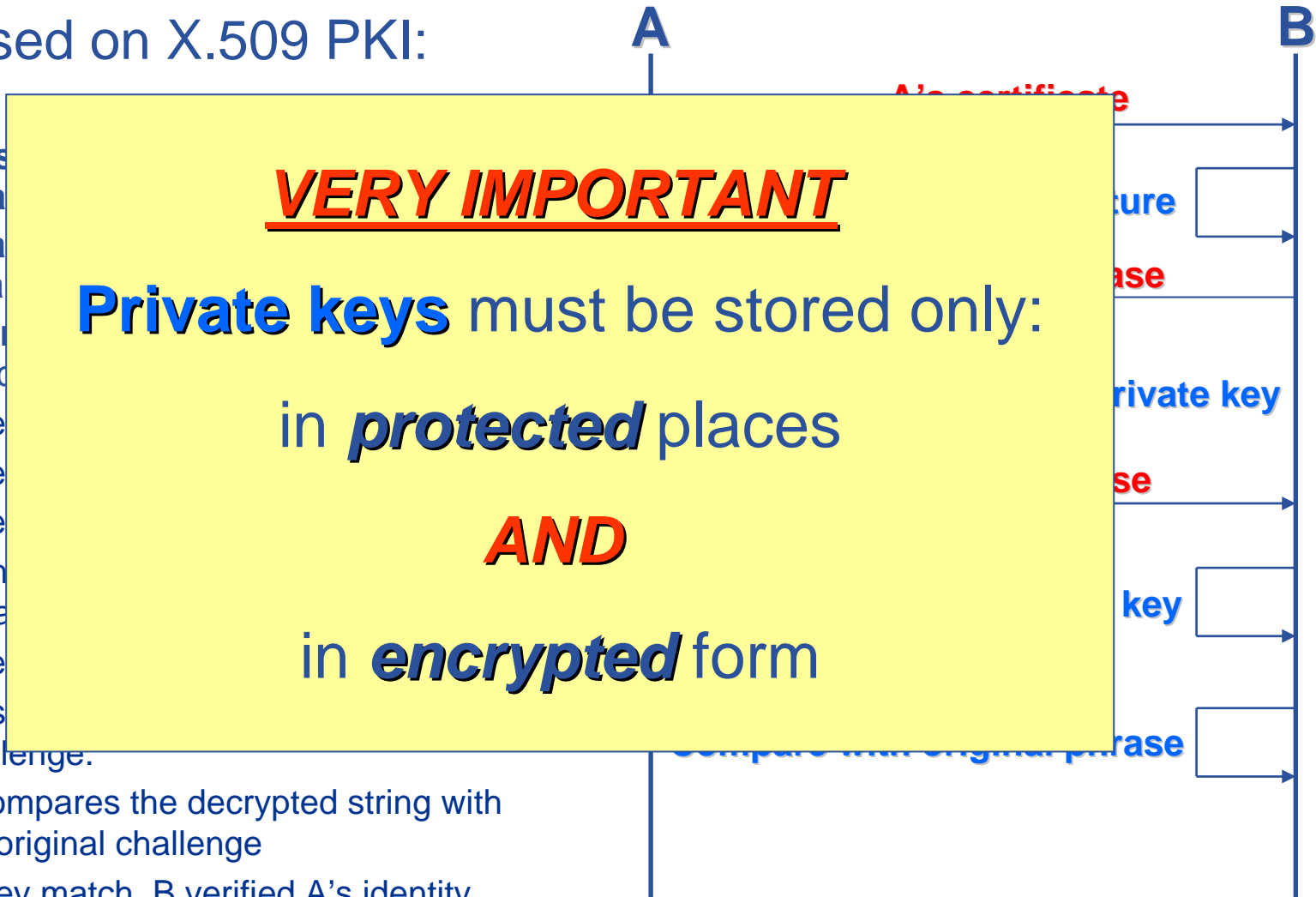
- *de facto* standard for Grid middleware
- Based on PKI
- Implements some important features
 - Single sign-on: no need to give one's password every time
 - Delegation: a service can act on behalf of a person
 - Mutual authentication: both sides must authenticate to the other
- Introduces **proxy certificates**
 - Short-lived certificates signed with the user's certificate or a proxy
 - Reduces security risk, enables delegation



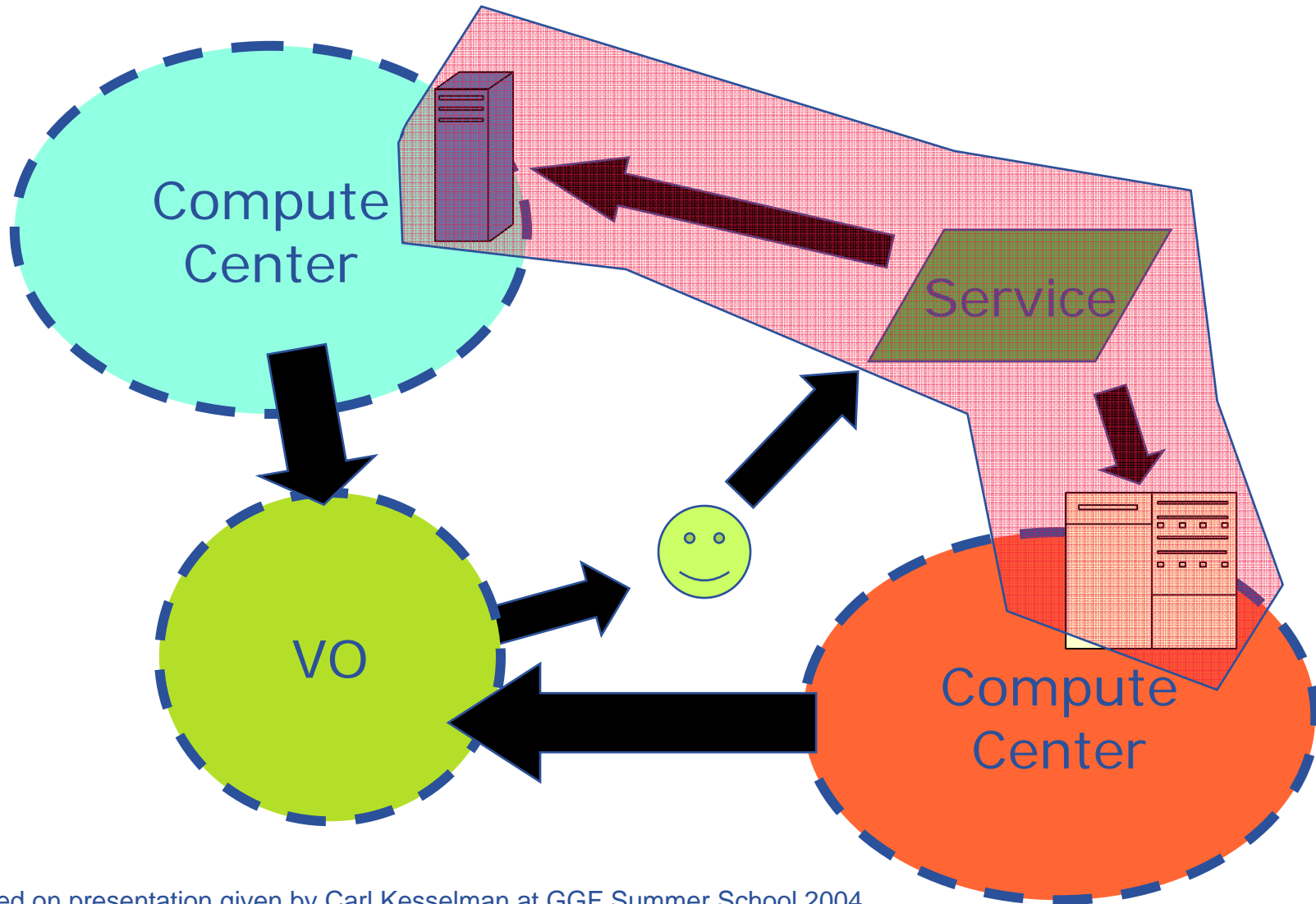
Based on X.509 PKI:

- every user has a certificate
- certificates are stored in the local file system
- every Grid user authenticates himself

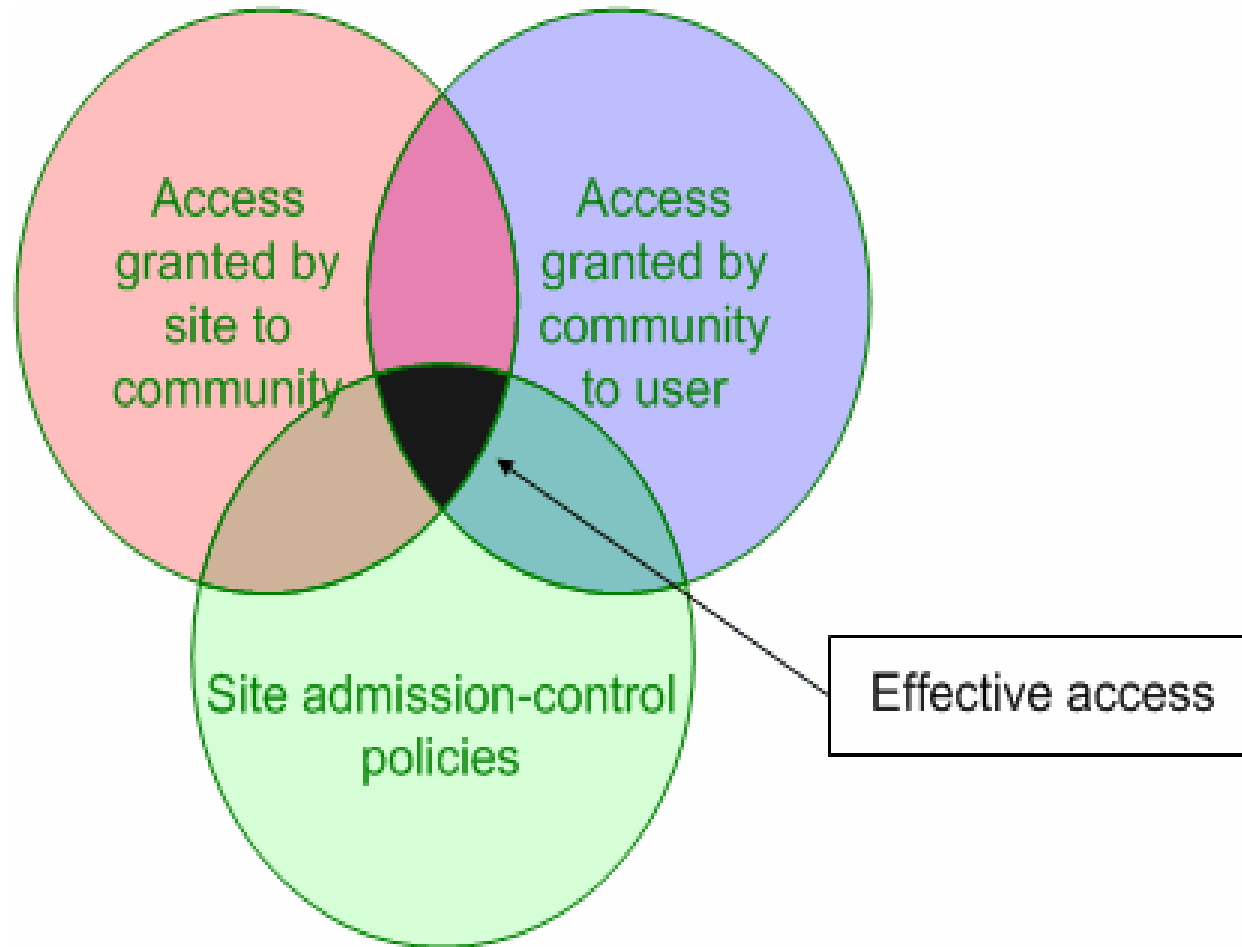
1. A sends a challenge to B
2. B verifies the challenge
3. B sends a response to A
4. A encrypts the response with his private key
5. A sends the encrypted response to B
6. B uses A's public key to decrypt the response
7. B compares the decrypted string with the original challenge
8. If they match, B verified A's identity and A can not repudiate it.



- **Delegation is the act of giving an organisation, person or service the right to act on your behalf.**
 - A Site delegates responsibility for the users that may access its resources to the VO
 - A Virtual Organisation delegates its rights to a user.
 - A user delegates rights to a service to allow programs to run on remote sites.



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



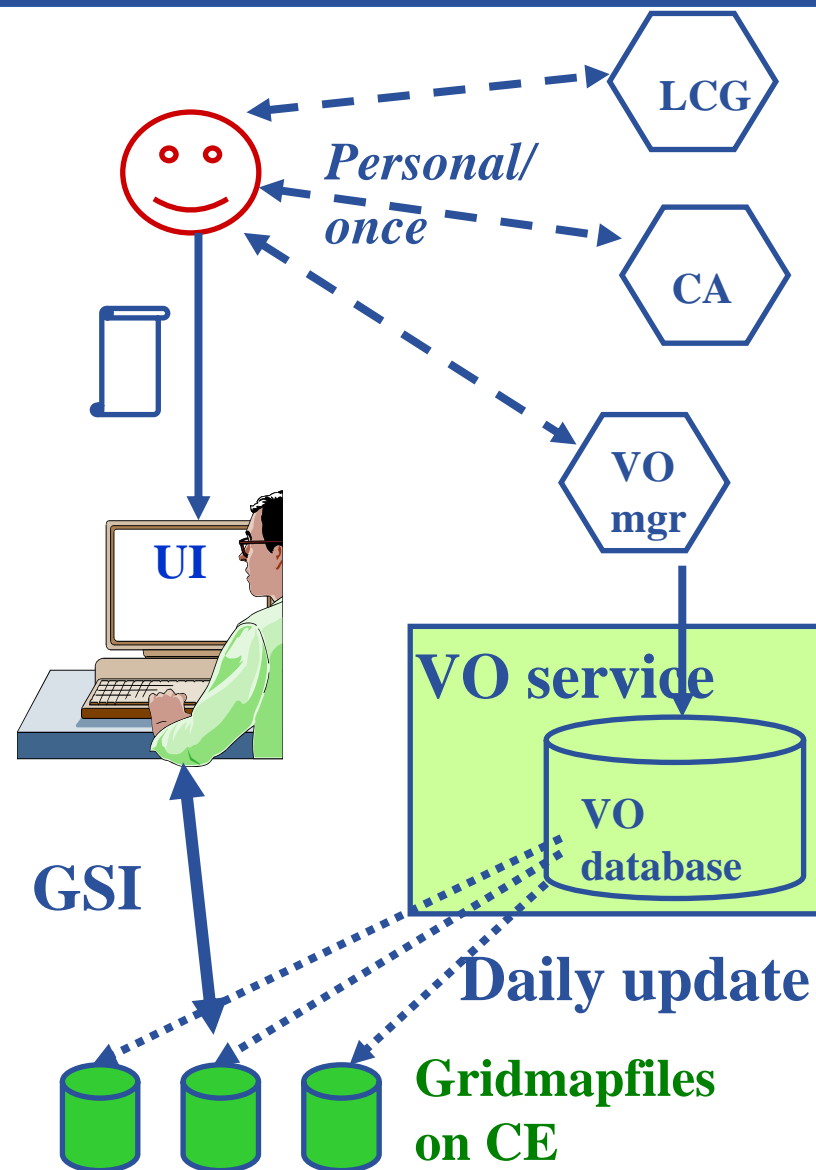
slide based on presentation given by Carl Kesselman at GGF Summer School 2004

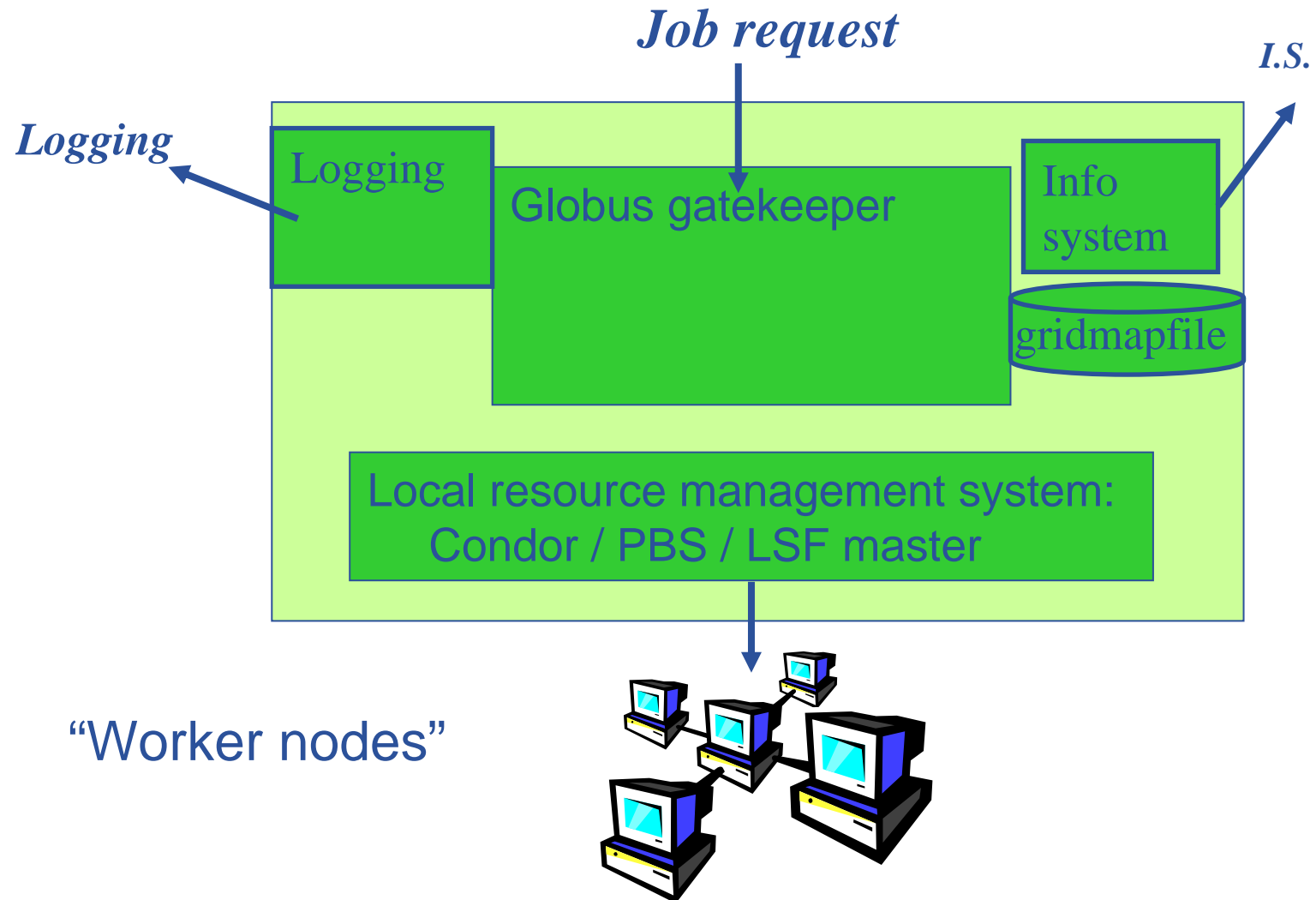
- **Authentication**

- User certificate signed by CA
- Connects to UI by ssh
- Downloads certificate
- Invokes Proxy server
- **Single logon** – to UI - 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**





- **Authorisation...**
- **What are you allowed to do?**
- **... and how is this controlled??**
- **In gLite the answer will be VOMS**
- **Virtual Organisation Management System**

LCG

- User is authorised as a member of a single VO
- All VO members have same rights
- Gridmapfiles are updated by VO management software

gLite, using VOMS In future...

- User can be in multiple VOs
 - Aggregate rights
- VO has groups
 - Different rights for each
 - Students
 - Experimentalists
 - ...
- VO has roles
 - Assigned to specific purposes
 - E,g. system admin
 - When assume this role
- Proxy certificate carries the additional attributes

- **single login using voms-proxy-init only at the beginning of the session (was grid-proxy-init)**
- **backward compatibility: the extra VO related information is in the user's proxy certificate, which can be still used with non VOMS-aware services**
- **multiple VOs: the user may "log-in" into multiple VOs and create an aggregate proxy certificate, which enables her to access resources in any of them**

- **VOMS Features**

- Single login using (proxy-init) only at the beginning of a session
 - Attaches VOMS attributes to user proxy
- Expiration time
 - The authorization information is only valid for a limited period of the time as the proxy certificate itself
- Multiple VO
 - User may log-in into multiple VOs and create an aggregate proxy certificate, which enables him/her to access resources in any one of them
- Backward compatibility
 - The extra VO related information is in the user's proxy certificate
 - User's proxy certificate can be still used with non VOMS-aware service
- Security
 - All client-server communications are secured and authenticated

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

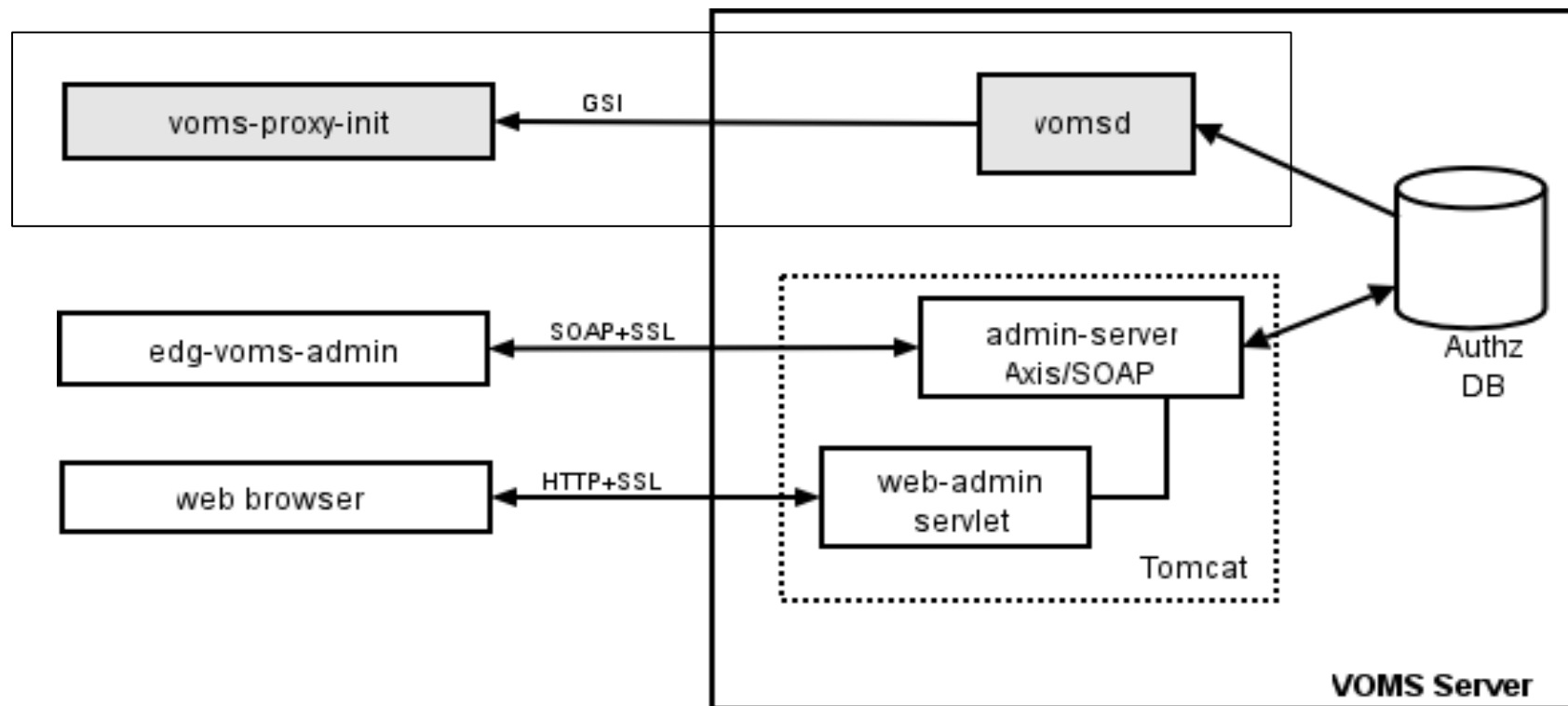
 - **Make VO manageable by organizing users in groups:**
 - VO BIOMED-FRANCE
 - Group Paris
 - *Sorbonne University*
 - Group Prof. de Gaulle
 - *Central University*
 - Group Lyon
 - Group Marseille
-
- **Groups can have a hierarchical structure**
- **Group membership is added automatically to your proxy when doing a *voms-proxy-init***

- **Assign rights to certain members of the groups**
 - using Access Control Lists (ACL) like in a file system
 - Allow / Deny
 - *Create user*
 - *Delete user*
 - *Get ACL*
 - *Set ACL*
 - *List user*
 - *Remove ACL*
 - Specifying unit for entry:
 - The local database administrator
 - A specific user (not necessarily a member of this VO)
 - Anyone who has a specific VOMS attribute FQAN
 - Anyone who presents a certificate issued by a known CA (Including host and service certificates)
 - Absolutely anyone, even unauthenticated clients

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

- **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*

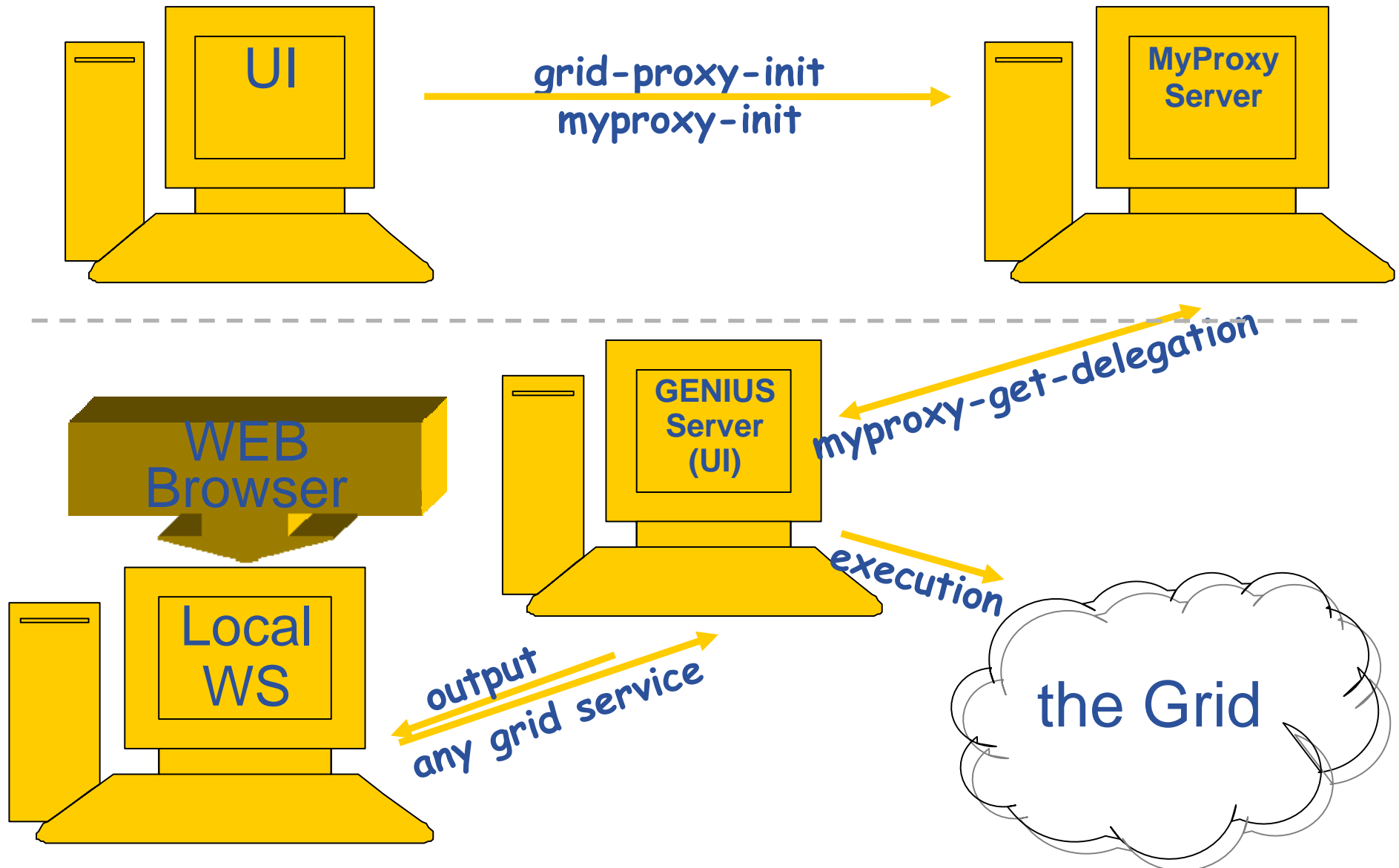
- **Example:**
 - User Yannick has the following membership
 - VO=BIOMED-FRANCE, Group=Paris, Role=SoftwareManager
 - During normal operation the role is not taken into account, e.g. Yannick can work as a normal user
 - For special things he can obtain the role “Software Manager”



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

- **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.... Its on a USB drive only.
 - To use a portal, and delegate to the portal the right to act on your behalf (by logging in to an account that can 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.**

Grid authentication with MyProxy



Consists of a server and a set of client tools that can be used to delegate and retrieve credentials to and from a server.

MyProxy Client commands:

- *myproxy-init*
- *myproxy-info* // `myproxy-info -s <host name> -d`
- *myproxy-destroy*
- *myproxy-get-delegation* // `myproxy-get-delegation -s <host name> -d
-t <hours> -o <output file> -a <user proxy>`
- *myproxy-change-pass-phrase*

The ***myproxy-init*** command allows you to create and send a delegated proxy to a MyProxy server for later retrieval; in order to launch it you have to assure you're able to execute the `grid-proxy-init` or `vomsproxy-init` command.

```
myproxy-init -s <host name> -t <hours> -d -n
```

The `myproxy-init` command stores a user proxy in the repository specified by `<host name>` (the `-s` option). Default lifetime of proxies retrieved from the repository will be set to `<hours>` (see `-t`) and no password authorization is permitted when fetching the proxy from the repository (the `-n` option). The proxy is stored under the same user-name as is your subject in your certificate (`-d`).

- **Digital credentials and the “Grid Security Infrastructure” (GSI) are the basis of AA**
- **Need to establish trust, so**
 - Resource can trust user
 - User can trust the resource
- *Achieve this if*
 - ***both users and sites trust Certificate Authorities***
 - ***CA’s trust each other***
 - ***CA’s sign user and site certificates***
 - ***Protect your certificate it is your grid identity***

- **The EGEE multi-VO grid is built on**
 - Authentication based on X.509 digital certificates
 - These are internationally recognised (hence we can have international collaboration)
 - Authorisation
 - Using VOMS
- **VOMS supports**
 - multiple groups, roles within a VO
 - a user being a member of several VOs
- **MyProxy supports**
 - Secure storage of certificates and proxies
 - Delegation to services: these can create and use a proxy on your behalf
 - E.g. portals

- VOMS
 - Available at <http://infnforge.cnaf.infn.it/voms/>
 - Alfieri, Cecchini, Ciaschini, Spataro, dell'Agnello, Fronher, Lorentey, From gridmap-file to VOMS: managing Authorization in a Grid environment
 - Vincenzo Ciaschini, A VOMS Attribute Certificate Profile for Authorization
- GSI
 - Available at www.globus.org
 - A Security Architecture for Computational Grids. I. Foster, C. Kesselman, G. Tsudik, S. Tuecke. *Proc. 5th ACM Conference on Computer and Communications Security Conference*, pp. 83-92, 1998.
 - A National-Scale Authentication Infrastructure. R. Butler, D. Engert, I. Foster, C. Kesselman, S. Tuecke, J. Volmer, V. Welch. *IEEE Computer*, 33(12):60-66, 2000.
- RFC
 - S.Farrell, R.Housley, An internet Attribute Certificate Profile for Authorization, RFC 3281