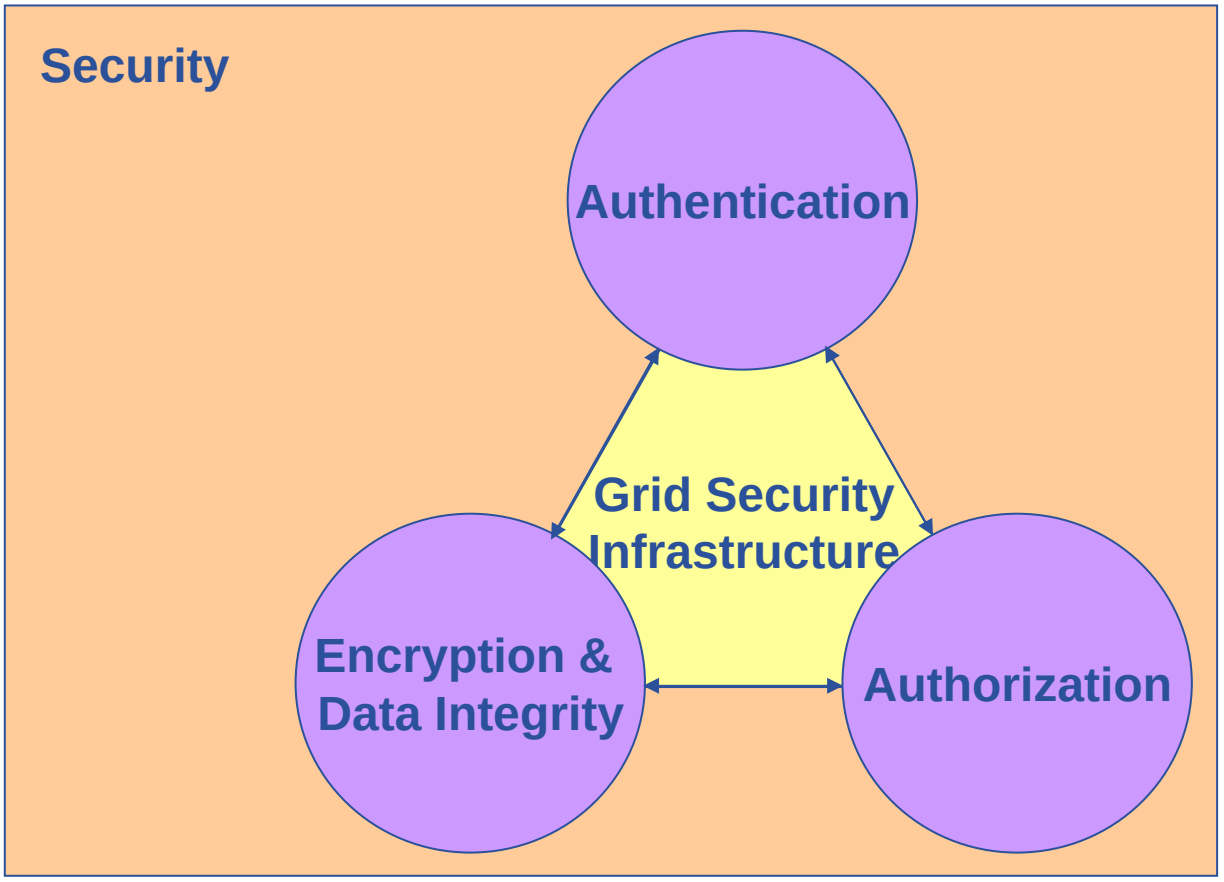
Security, Authorisation and Authentication

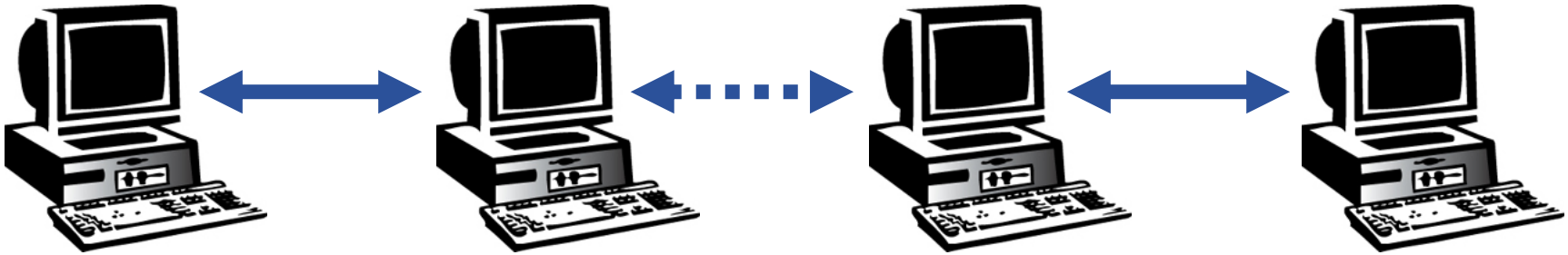
Mike Mineter
Training, Outreach and Education
National e-Science Centre
mjm@nesc.ac.uk

With thanks for some slides to EGEE and Globus colleagues

Security Overview



The Problems - 1



User **Resource**

- **How does a user securely access the Resource without having an account with username and password on the machines in between or even on the Resource?**

- **How does the Resource know who a user is?**
- **How are rights controlled?**

Authentication: how is identity of user/site communicated?

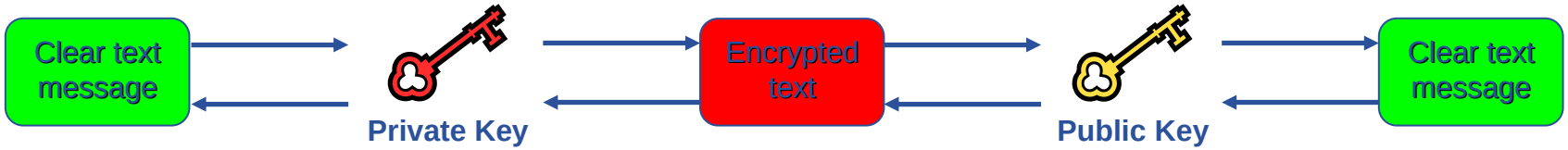
Authorisation: what can a user do?

The Problems -2: reducing vulnerability

- **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.

Basis of security & authentication

- **Asymmetric encryption…**

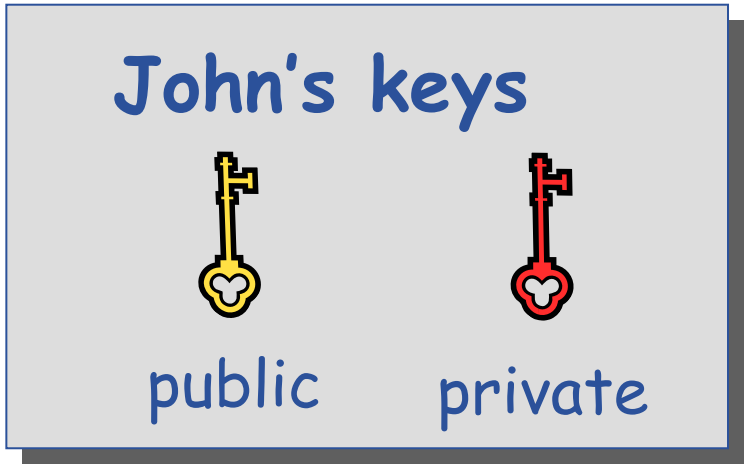


- **…. and Digital signatures …**
  - A hash derived from the message and encrypted with the signer's private key
  - Signature is checked by decrypting with the signer's public key

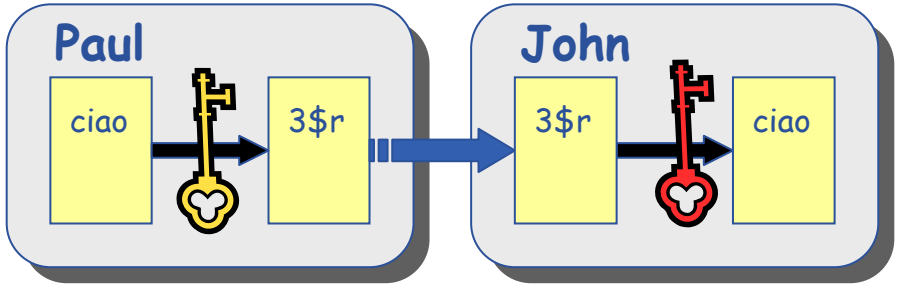
- **Are used to build trust**
  - That a user / site is who they say they are
  - And can be trusted to act in accord with agreed policies

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.

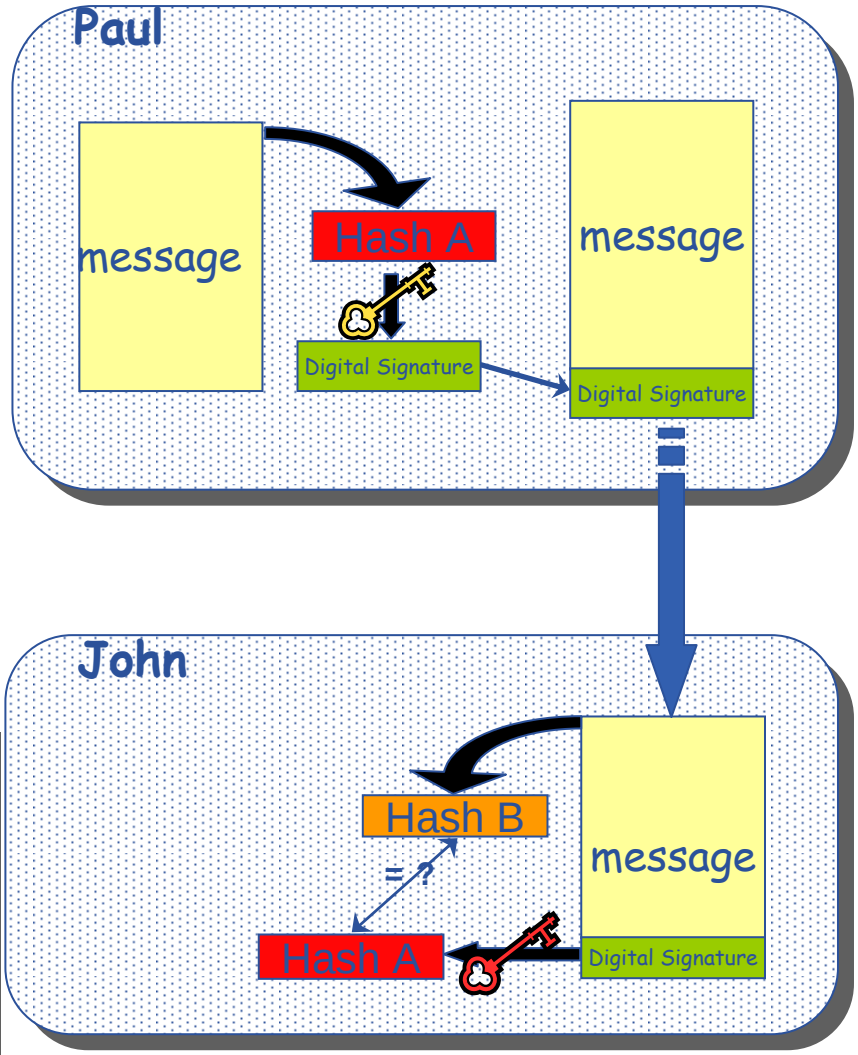


- **Concept - simplified version:**
  - Public keys are exchanged
  - The sender encrypts using receiver's public key
  - The reciever decrypts using their private key;



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 signature, to get A, using Paul's *public* key.**
- **If hashes equal:**
  1. message wasn't modified;
  2. hash A is from Paul's private key



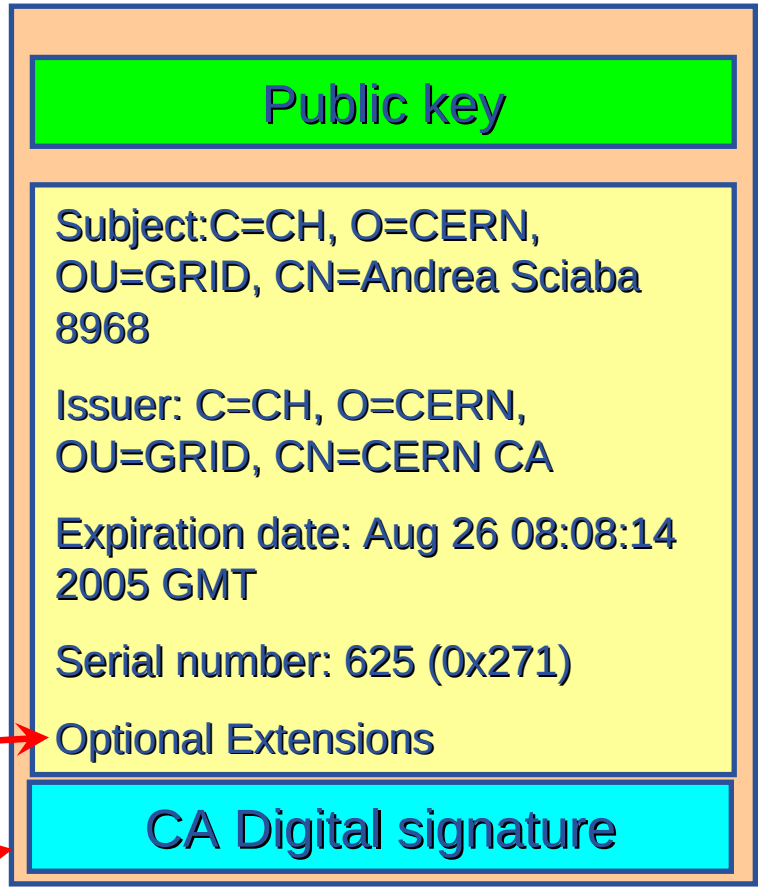
Digital Certificates

- **How can John be sure that Paul's public key is really Paul's public key and not someone else's?**
  - A *third party* signs a certificate that binds the public key and Paul's identity.
  - Both John and Paul trust this third party

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

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



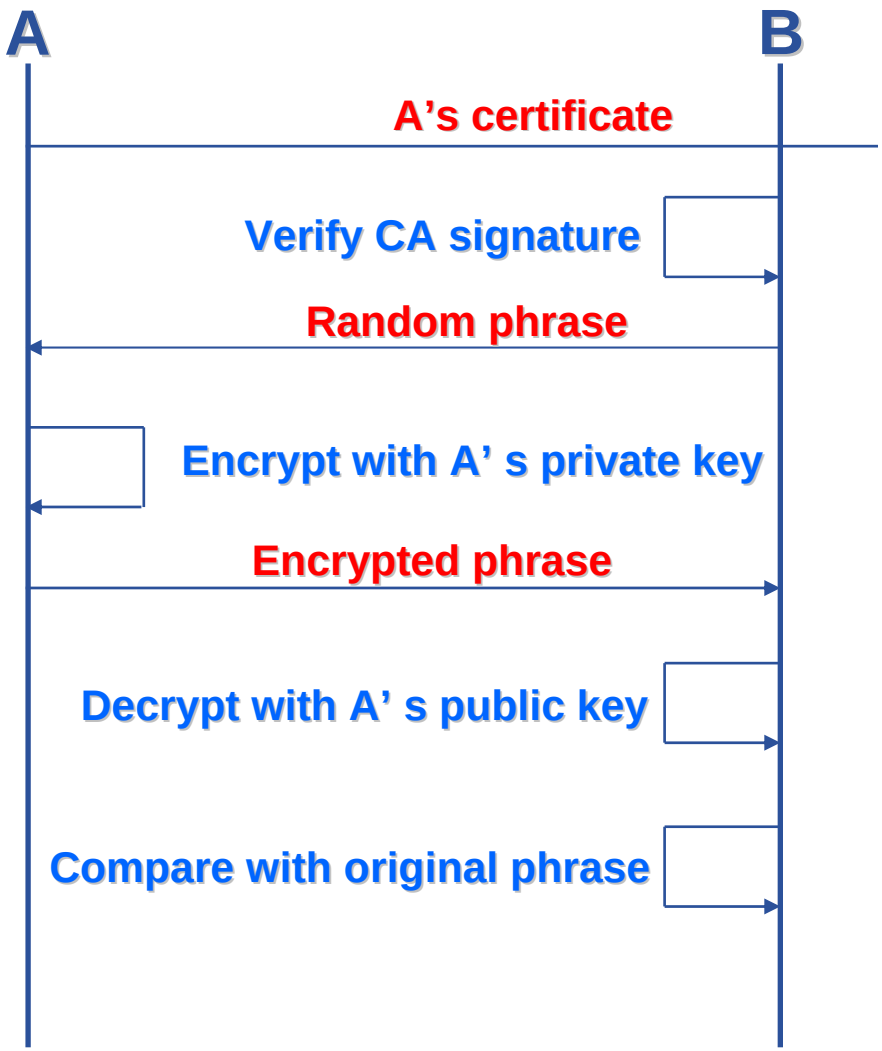
Certification Authorities

- **User's identity has to be certified by one of the national *Certification Authorities* (CAs)**
- **Resources are also certified by CAs**
- **CAs are mutually recognized http://www.gridpma.org/,**
- **CAs each establish a number of people "registration authorities" RAs**

The Grid Security Infrastructure (GSI)

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



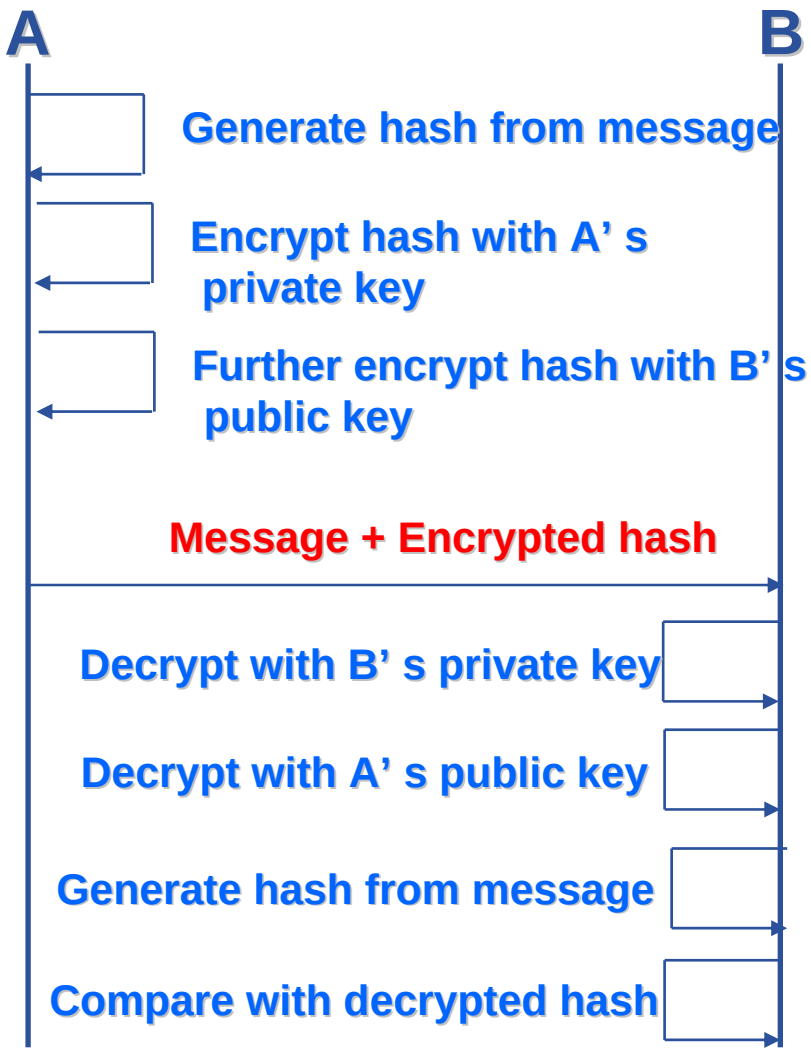
The Grid Security Infrastructure (GSI) - continued

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

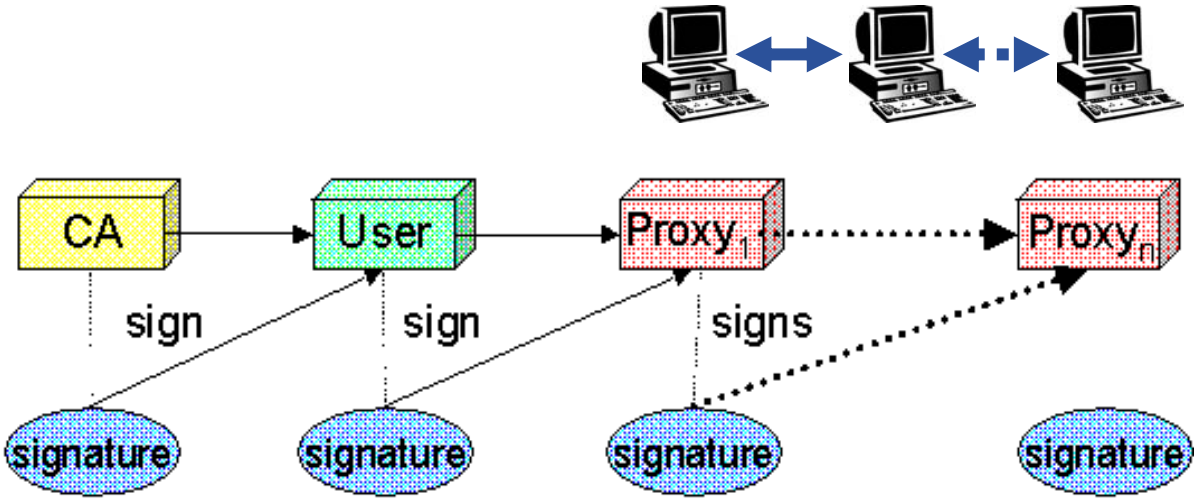
.

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

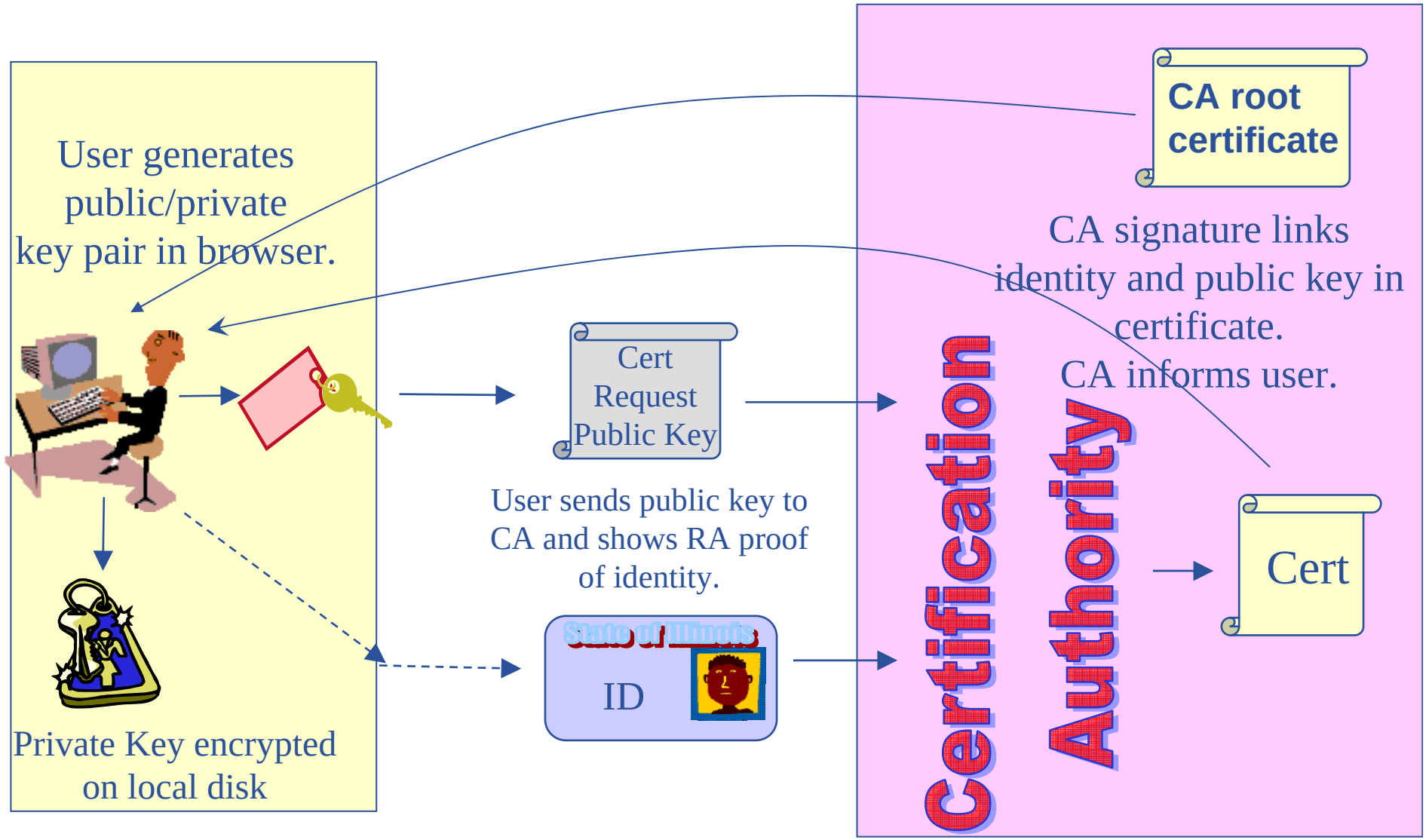


Grid Security Infrastructure - proxies

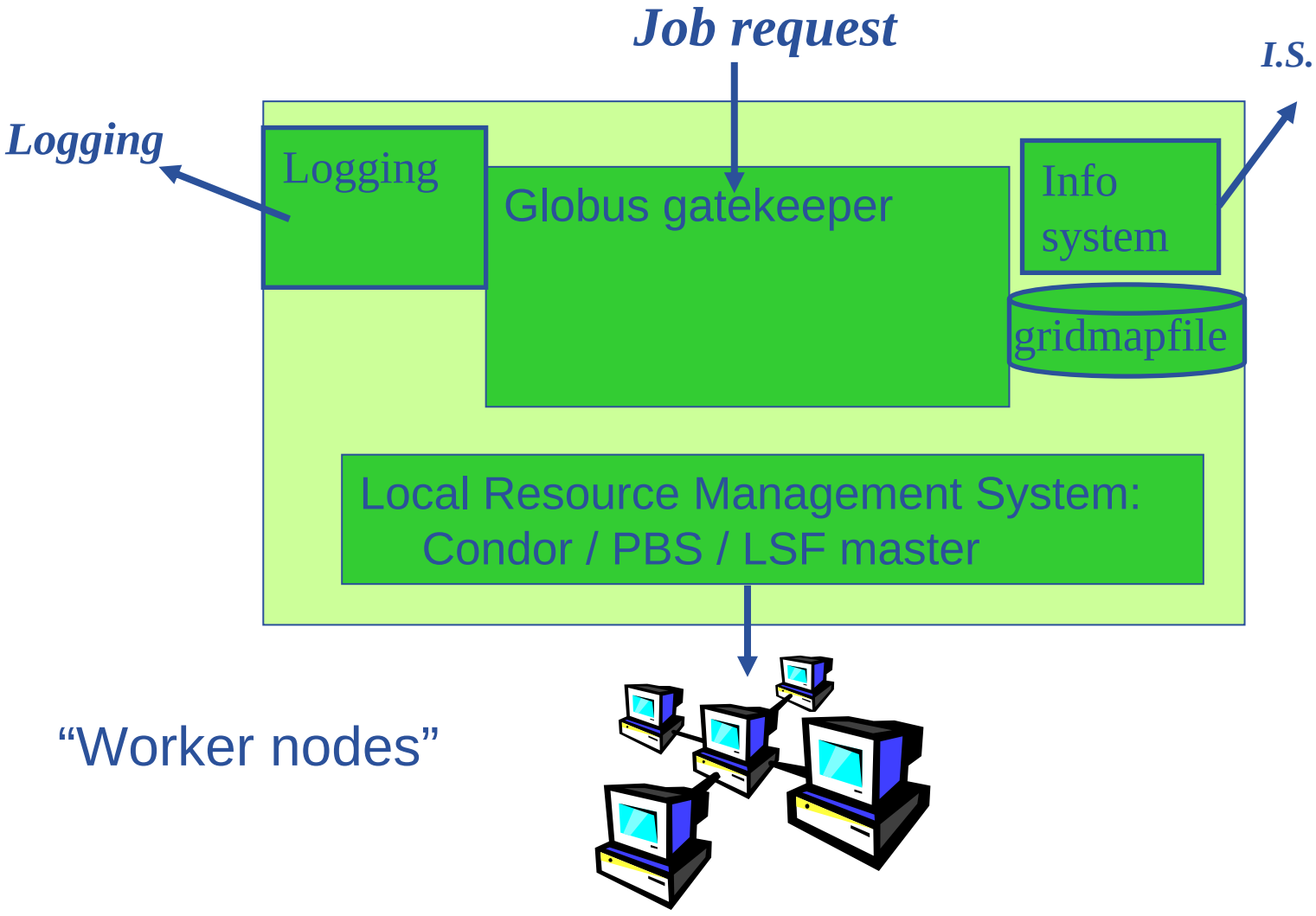
- **To support delegation: A delegates to B the right to act on behalf of A**
- **proxy certificates *extend X.509 certificates***
  - Short-lived certificates signed by the user's certificate or a proxy
  - Reduces security risk, enables delegation



Certificate Request



"Compute element": a batch job queue



VOMS: Virtual Organization Membership Service

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**

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**

User Responsibilities

- Keep your private key secure – *on USB drive only*
- 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.

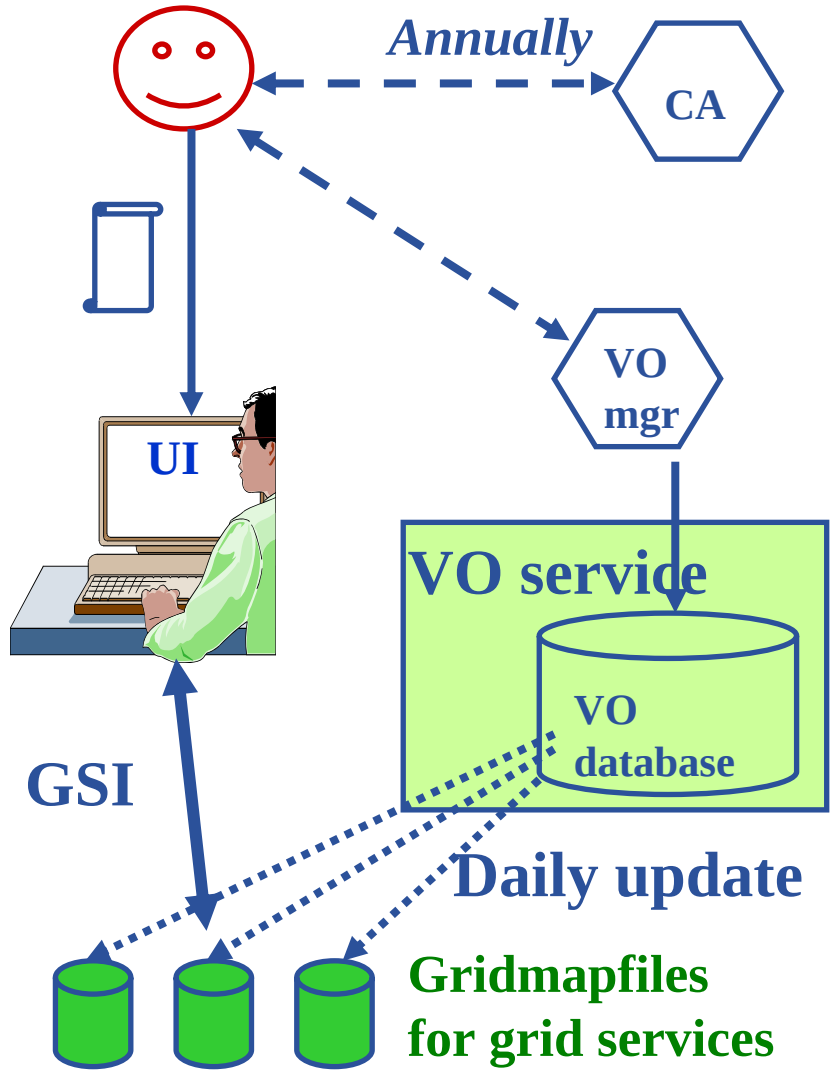
If your certificate or delegated service is used by someone other than you, it cannot be proven that it was not you.

AA Summary

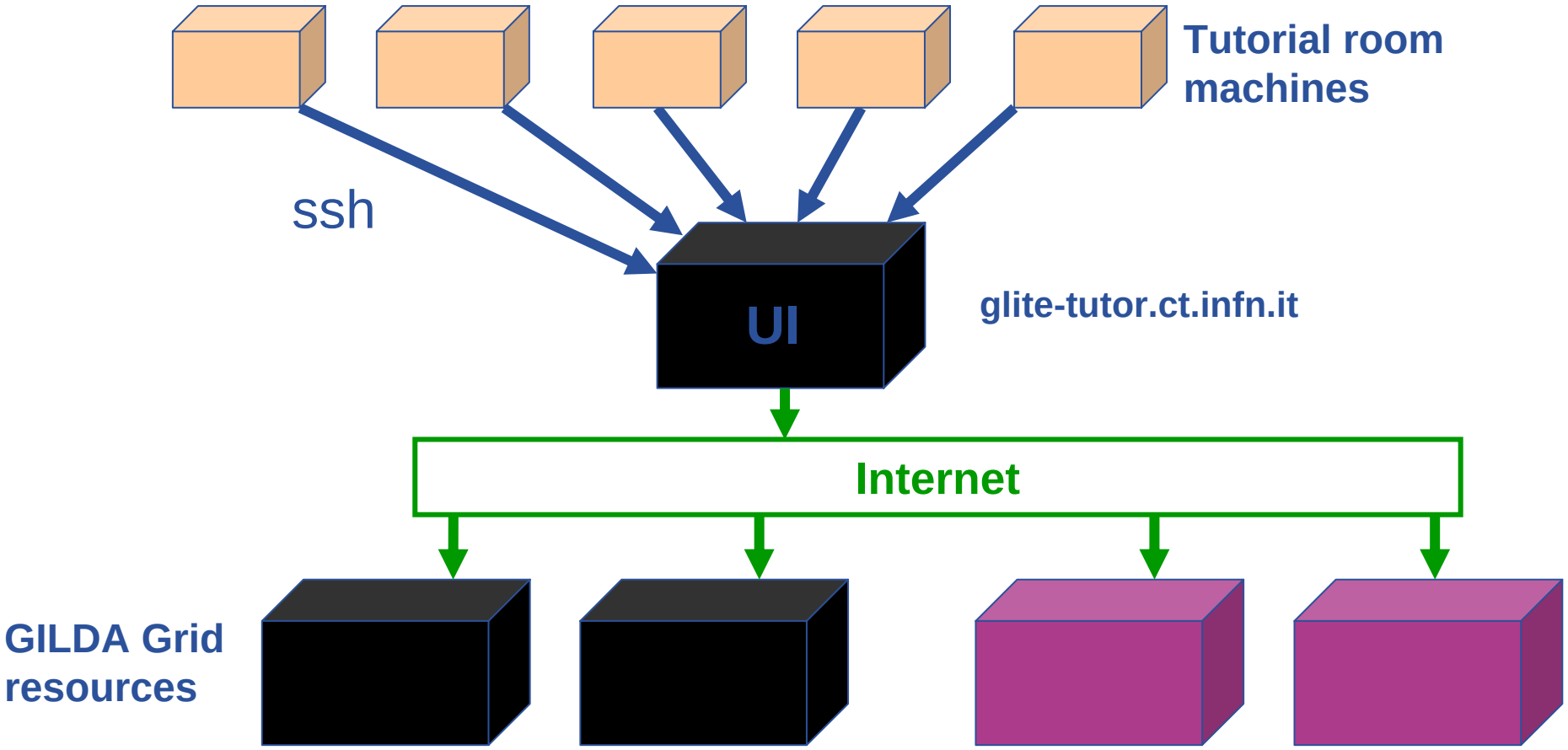
- **Authentication**
  - User obtains certificate from Certificate Authority
  - Connects to UI by ssh
    UI is the user's interface to Grid
  - Uploads certificate to UI
  - Single logon – to UI - create proxy
  - then **Grid Security Infrastructure uses proxies**

- **Authorisation**
  - User joins Virtual Organisation
  - VO negotiates access to Grid nodes and resources
  - Authorisation tested by resource:

Gridmapfile (or similar) maps user to local account



Our setup


Tutorial room machines
ssh
UI
glite-tutor.ct.infn.it
Internet
GILDA Grid resources

How to start putty to enable x11

1. Run exceed
2. Run putty
3. Set X11 before opening session
4. (kwrite editor available)