Grid Certificates and Security

Yiannis Ioannou

Certificates

- Access to the EGEE Grid is controlled by the use of digital certificates
- Each EGEE user is associated with a digital certificate
- The digital certificate is your identity card to the Grid World

What is a certificate???

Certificate - Example

Signature Algorithm: sha1WithRSAEncryption

Issuer: C=CY, O=CyGrid, O=HPCL, CN=CyGridCA

Validity

Not Before: Sep 15 09:41:13 2006 GMT

Not After : Sep 15 09:41:13 2007 GMT

Subject: **C=CY, O=CyGrid, O=HPCL, CN=John Smith**

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

00:9e:33:db:90:bc:5f:0b:71:d7:83:24:20:7b:08:14:dc:a3:b0:e2:16:07:e0:45:79:1e:a8:09:40:90:
4f:d7:72:ba:35:12:af:0b:07:11:e5:7a:6a:82:2f:0f:8c:51:ed:fe:e4:df:5d:ac:2e:39:a6:53:20:80:
78:e6:16:0a:58:f3:95:06:15:9a:94:98:08:cd:e4: d1:d8:cb:45:e0:b4:38:61:3c:bb:fb:77:25:a7:ab:
98:ae:1c:d0:df:bc:cc:49:4a:ab:ad:7b:27:53:ae:4f:fb:bb:81:79:c3:c7:e6:7a:24:bd:c4:80:c9:94:
39:b2:00:fd:e2:e9:43:21:0b

Certificate - Example

X509v3 Subject Alternative Name:

email:johnsmith@helloworldcom

Signature Algorithm: sha1WithRSAEncryption

39:fc:9e:4f:70:35:33:20:8f:09:f0:1b:80:8a:7e:8b:b9:a8: 93:77:35:3b:13:8a:e3:47:10:78:4d:64:e6:63:7b:94:05:4c:
5c:8d:de:03:06:e4:c0:bf:9d:90:ef:ca:51:34:9b:c4:94:2a: a5:60:f1:59:17:2c:60:47:d0:f6:69:3c:74:21:27:79:d6:60:
b5:1d:f7:15:ae:3b:bd:70:32:04:51:83:3a:3a:08:0e:3e:f6: c3:fb:18:ca:8f:06:cc:c0:49:08:15:05:ef:3d:d4:3e:69:fd:
6c:14:c5:56:77:92:22:62:89:24:58:36:32:d3:50:bc:bd:64: 74:82:92:f3:56:f7:43:c6:73:c2:ff:cd:6f:ec:86:88:91:21:
20:f6:6a:d7:ff:50:82:25:8d:1a:63:99:e2:f7:13:38:c5:ff: 34:14:db:69:38:1d:22:3d:4f:8f:da:5b:1c:ca:c3:ea:ed:1e:
32:2f:7e:8d:1e:97:16:98:53:67:89:6d:e7:12:1d:a4:b9:95:11:37:e8:7d:b6:87:37:db:53:52:d6:9a:bf:78:df:a7:89:a6:
cd:5c:8e:fc:73:96:24:8f:09:ba:9f:ea:75:69:5e:b7:ed:11:c9:3c:a8:95:06:0a:a7:0c:6b:c5:bb:d4:3a:17:21:e4:6c:11:
23:fa:63:0a

Certificates - PKC

Certificates are based on Public Key Cryptography

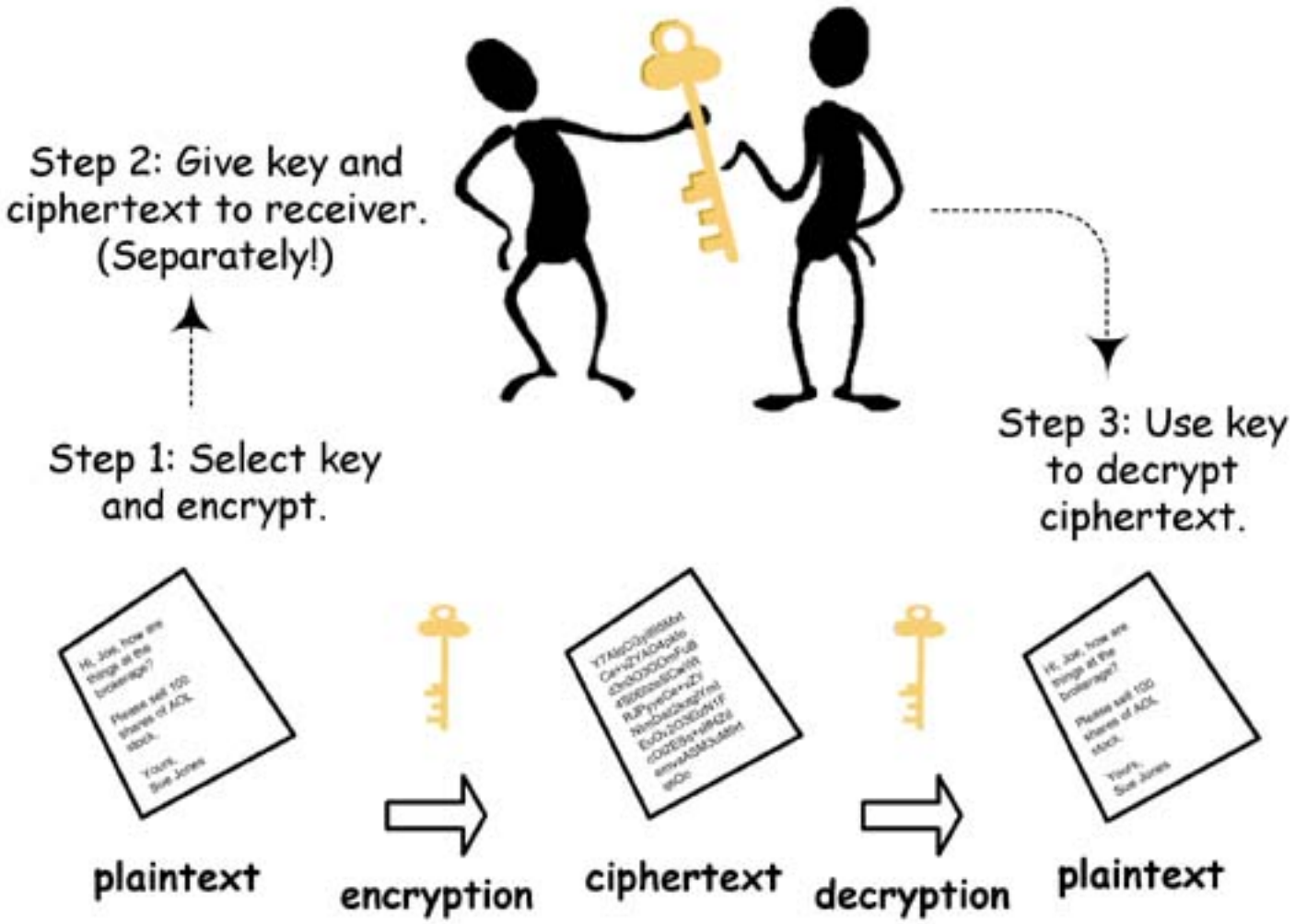
Certificates - Cryptography

- **Symmetric cryptography**
  - There is only one key and is used both for encryption and decryption
- **Public key cryptography**
  - The decryption key is not the same with the encryption key
  - One key is secret/private while the other one is public
  - By knowing the public key, it is computationally infeasible to reveal the secret key

Certificates - Cryptography

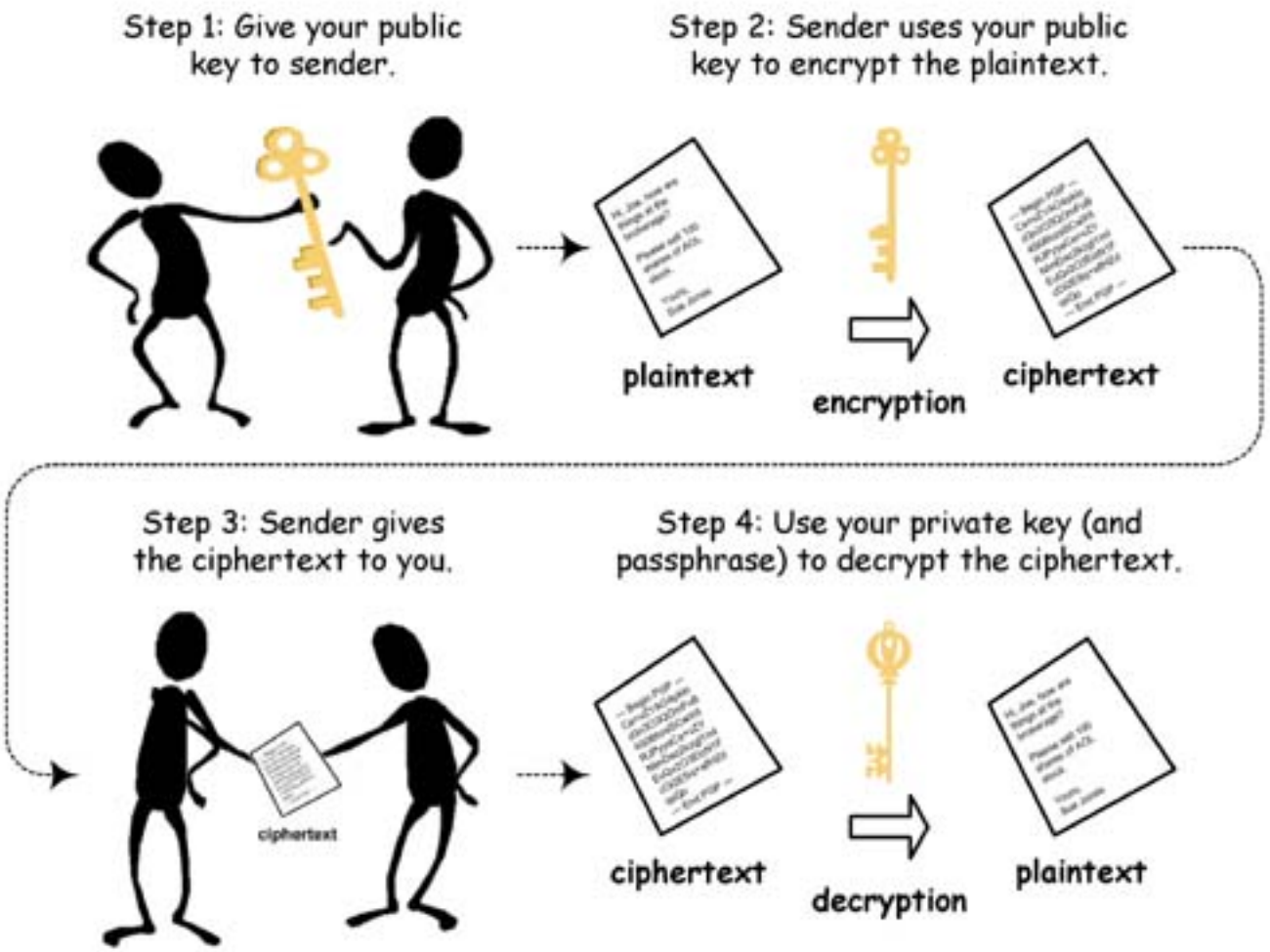
- **Symmetric cryptography**
  - 3DES, AES (rijndael variation) , blowhish, twofish
- **Public key cryptography**
  - RSA and DSA

Certificates – Symmetric Crypt.



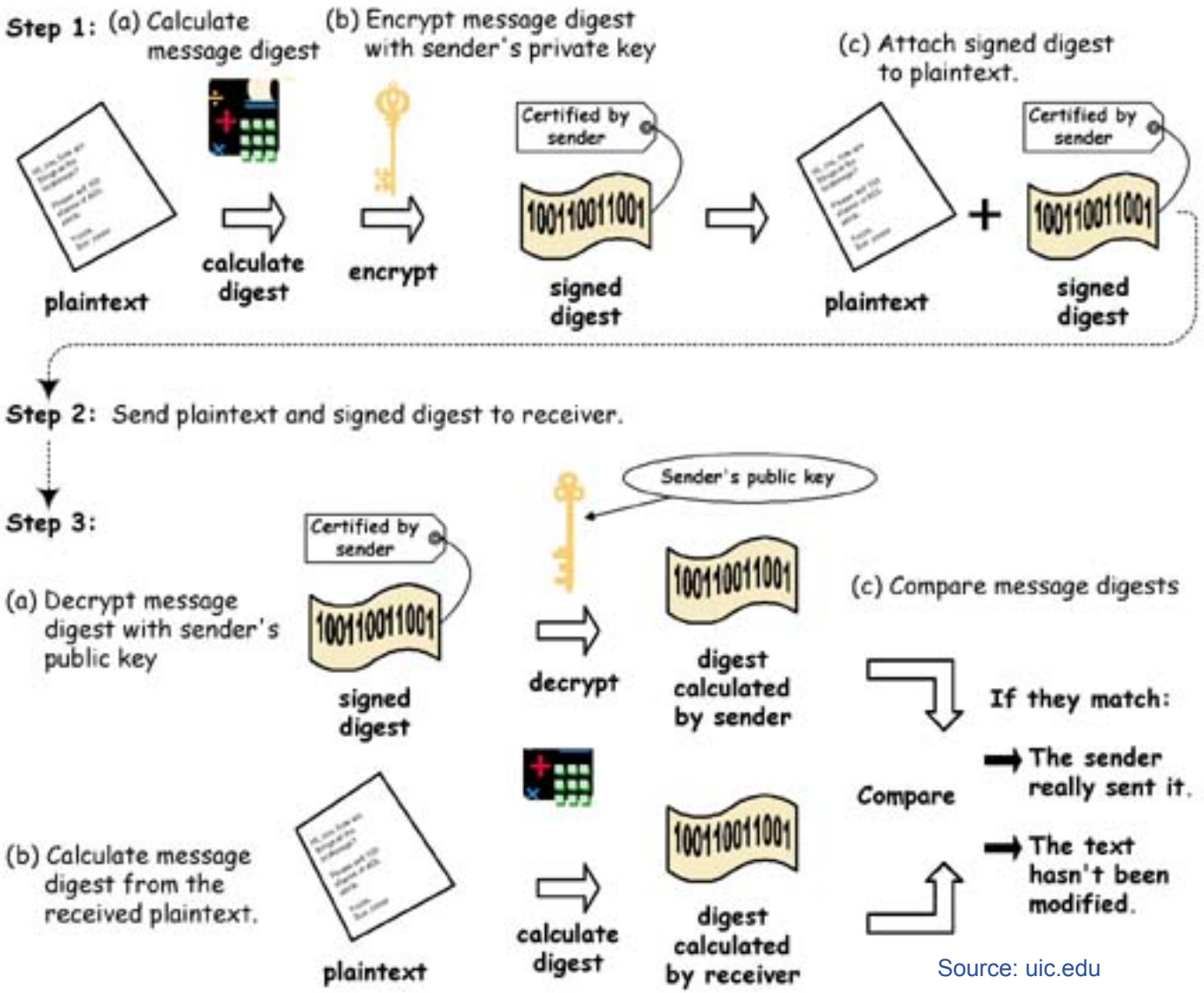
Source: uic.edu

Certificates - PKC



Source: uic.edu

Certificates - Digital Signatures



Source: uic.edu

Certificates - Digital Signatures

- **Why do we sign fingerprints?**
  - Performance
    - It is faster to encrypt a short string of characters
  - Security
    - If we don't use fingerprints, then the encryption and signing of information would take place in blocks
      - *This means that each block can be presented independently*
      - *A malice could for example delete a block while the signature is still valid.*

Certificates - PKC

- **The private key can be used either**
  - to digital sign information or
  - decrypt encrypted data
- **The corresponding public key can be used either to**
  - to verify a digital signature
  - encrypt data
- **Note that anyone knowing the corresponding public key can verify the authenticity of a signature**

Certificates - EGEE

- **A user is associated with a digital certificate**
- **The digital certificate contains user's information and public key**
  - The secret key is not (must never be) revealed
- **A user can prove that he/she is the owner of a certificate by demonstrating his/her ability to sign data**
  - Note that anyone can verify the digital signature since the public key is actually public
- **The relation of the user with the digital certificate is signed by a recognized Certification Authority**

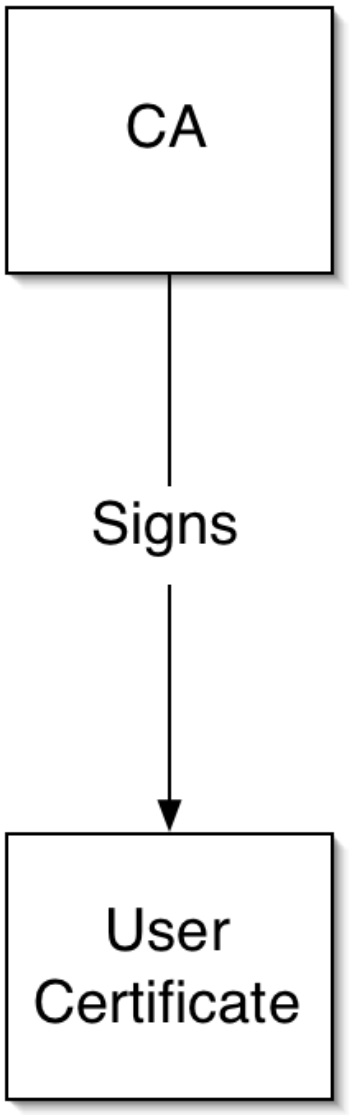
Certificates - CA

- Anyone can generate a private and public key => This means that anyone can generate a certificate
- A valid certificate is only a certificate you can trust
- The assurance that a certificate can be trusted is provided by the Certification Authority

Certificates - CA

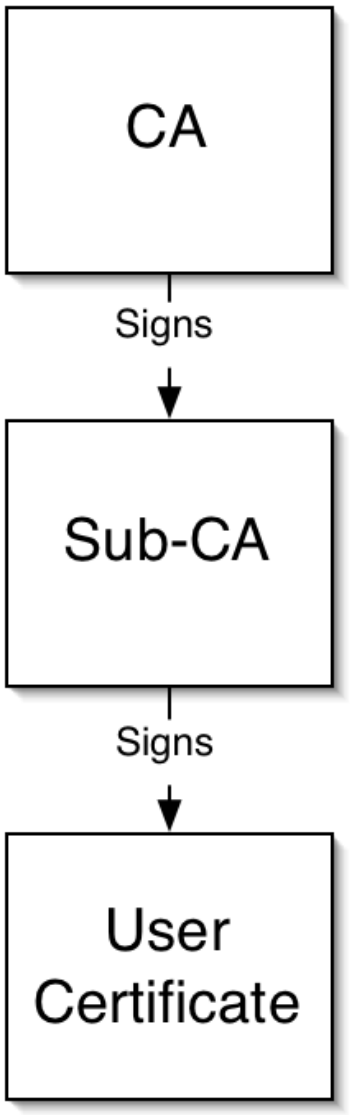
- CA has its own certificate that corresponds to a private key
- The Certification Authority signs users' certificates
- The Certification Authority signs only these certificates whose contents have been verified
- A site (or a user) can check the authenticity of a user's certificate by checking the authenticity of the CA's signature

Certificates – User Certificate Validation



- **Check a user's certificate details**
  - Is the user's certificate known? No!
  - Can we trust the information on the certificate? No!
- **Check the issuer of the certificate**
  - Is the issuer known? Yes!
  - Do we trust the issuer? Yes!
- **Use the CA's public key to verify the CA's signature on the User Certificate**
  - A valid signature provides a kind of trust about the certificate's details

Certificates – User Certificate Validation



- Verify user's certificate details using Sub-CA's certificate
- Verify Sub-CA's certificate using CA's certificate
- Verify that a user is the owner of the certificate by asking the user to encrypt some random data

Certificates – CA Security

- **CA's private key is of critical importance**
- **The machine hosting the private key is always offline**
- **Sites with many users use alternative methods**
  - The CA machines are on a highly monitored network.
  - The key is stored on tamper resistant devices, rather than on hard disks
  - ...

Certificates – Security Services

- **Authentication**
  - Fresh signatures provides evidence about the real identities of the communication parties
- **Authorization**
  - Since we know with whom we are communicating we can restrict the access to certain services
- **Data Integrity**
  - Any message alterations can be detected by verifying the digital signature
- **Data Confidentiality**
  - Public key cryptography can be used for building confidential channels

Grid Infrastructure

Main Logical Middleware Services

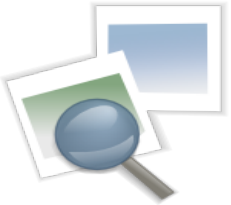
- **User Interface**

- **Computing Element**

- **Worker Nodes**

- **Information Service (BDII)**

- **Resource Broker**



- **VOMS Server**

- **Storage Element**

Main Logical Middleware Services

- **User Interface**
  - Used by the end-users to exploit the grid infrastructure
- **Resource Broker**
  - handles job submission requests
  - locates resources that meet the requested requirements
  - stores the output of the jobs
- **Information Service**
  - maintains fresh information on the grid infrastructure
    - testbed sites, computing elements, storage elements
- **Computing Element**
  - is the “master” of a number of worker nodes

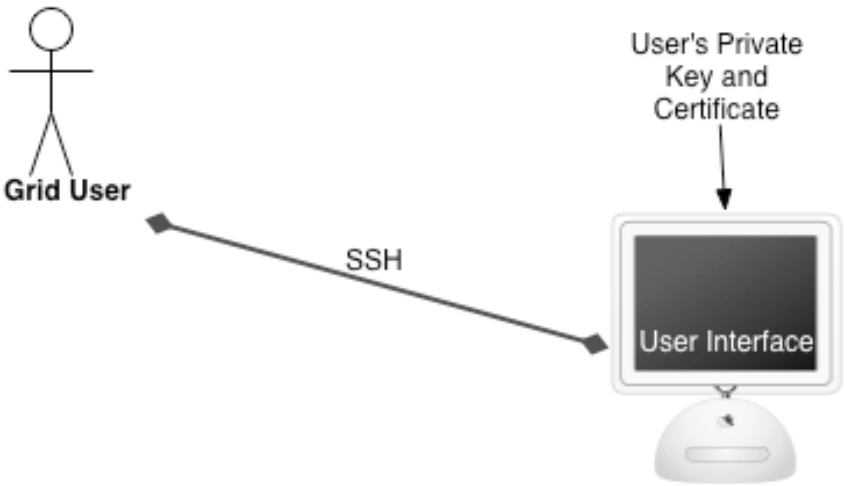
Main Logical Middleware Services

- **Storage Element**
  - Used to provide storage facilities to the grid users
- **Worker Nodes**
  - execute users' jobs
- **VOMS Server**
  - Keeps a list of the users registered to a certain VO
  - Queried by other services in order to determine whether a user is registered to a VO
- **Myproxy Server**

Virtual Organizations

- **A lot of Grid members support the Grid Infrastructure and provide Grid Resources**
- **Grid members may provide different resources and support different goals**
- **These resources and goals are grouped in "Virtual Organizations"**
  - Examples: Atlas, CMS, SEE
- **A user may be a member of one or more Virtual Organizations**
- **Access to resources is controlled via the concept of the Virtual Organizations**
  - Users not registered to a VO can not use the resources supporting this VO

How it works???(1)

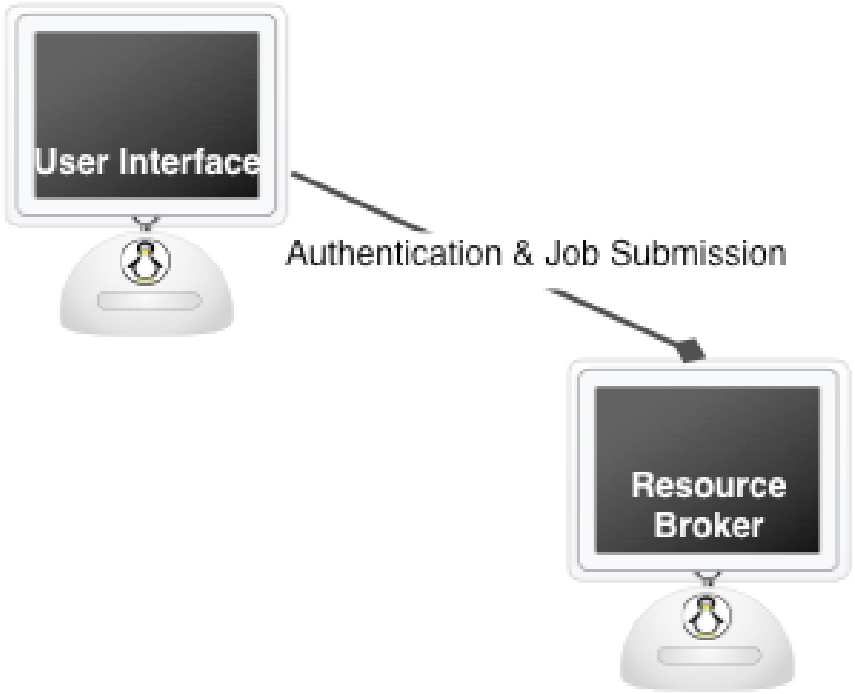


- A user ssh to a UI machine
- The user gets a shell screen
- The UI machine provides all the tools required for using the grid
- A user's certificate and private key must be located on this machine

How it works???(2)

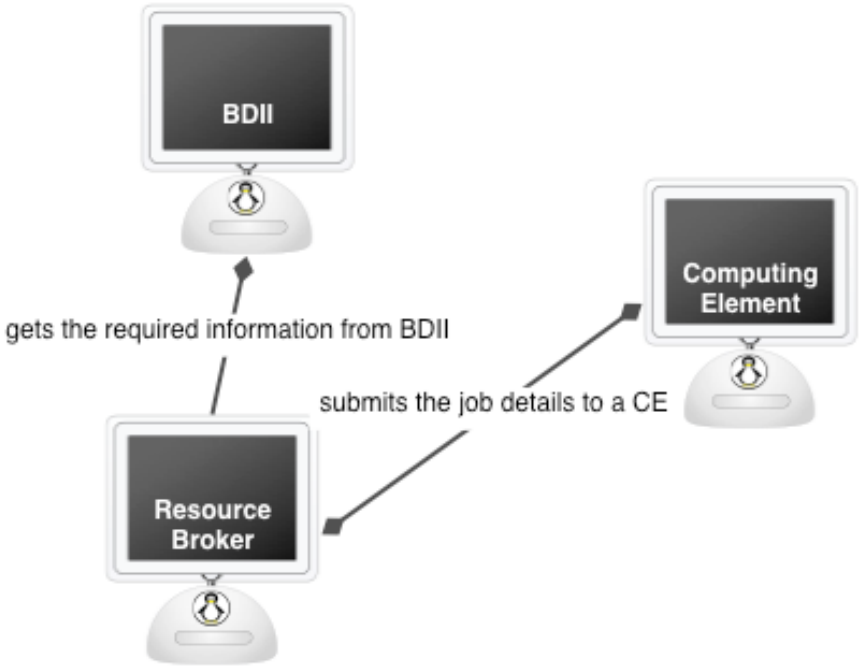
- Using the tools provided by the UI, a user can initialize a proxy. A proxy is the user's temporary ticket to the grid.
- A proxy is a certificate with limited time of validity and self-signed by the user
- A valid proxy allows a user to use the grid

How it works???(3)



- The UI requests to submit a job
- A mutual authentication takes place
- The RB generates a proxy and it is signed by the UI
- The job input files and executable are stored in the RB
- The RB responds with a url corresponding to the job

How it works???(4)

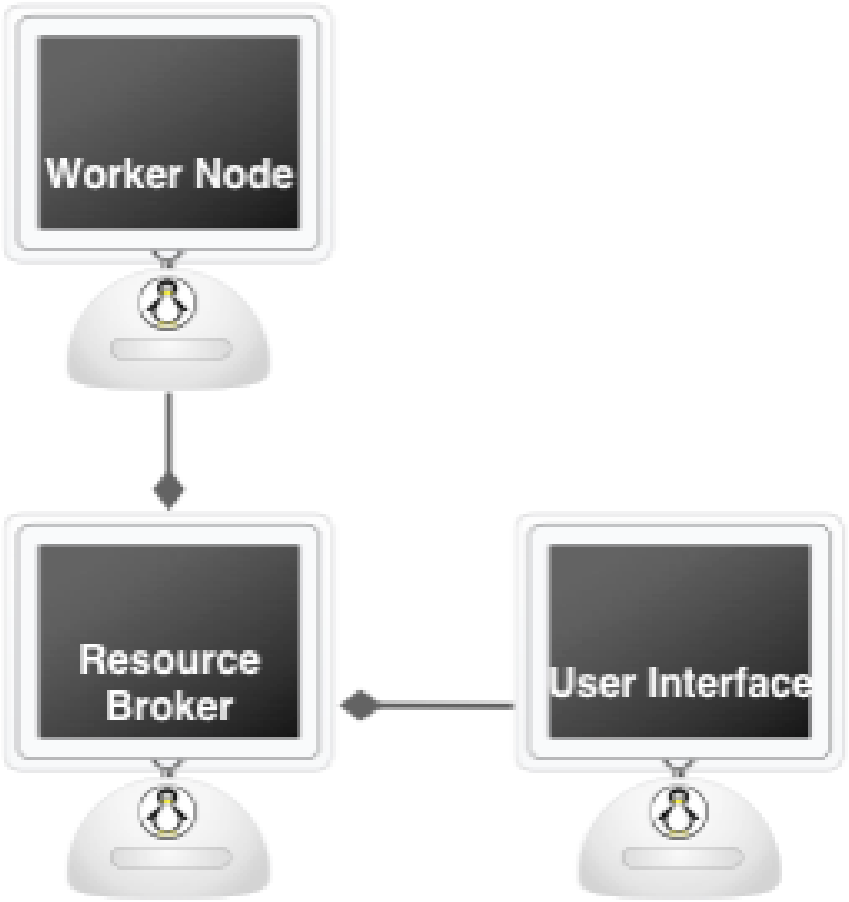


- The Resource Broker gathers the required information from the BDII
- It locates an appropriate CE and submits a job description to a CE
- Authentication between grid service (i.e. RB-CE) takes place on every step using the host and proxy certificates

How it works???(5)

- **The Computing Element locates an available worker node and forwards the job details**

How it works???(6)



- The worker node uses the job details to download the actual job files from the RB
- The job is executed and the results are stored back to the RB
- The owner of the job can retrieve the job output from the RB using the job's url

Grid Security Services

- Single Sign-on
-
- Authentication
- Delegation
- Proxy Autorenewal
- Authorization

Grid Security Services – Single Sign-ON

- Private Keys at the UI are protected by a pass phrase
- A user initializes a proxy by giving his/her pass phrase
- Once a proxy has been initialized, it is not necessary to give the pass phrase again until the proxy has expired

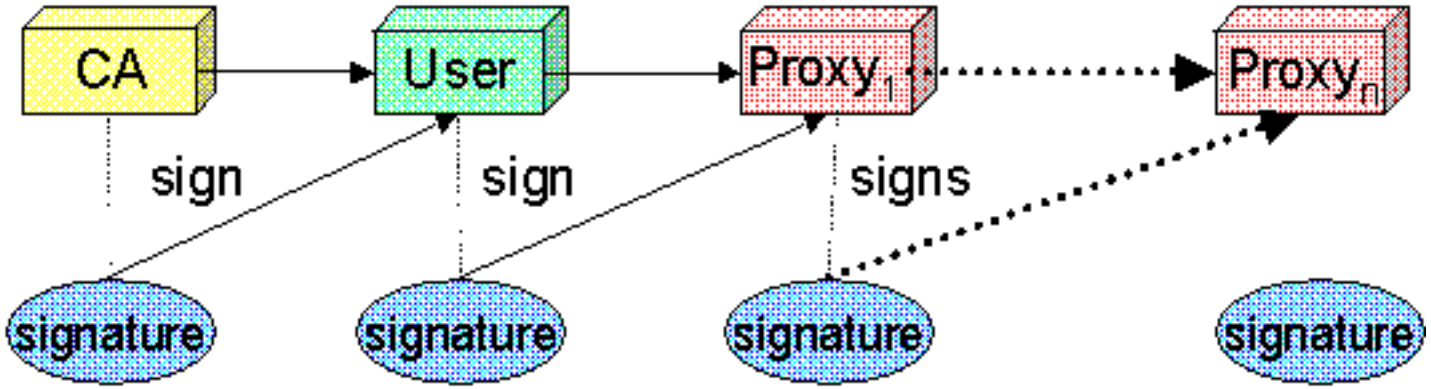
Grid Security Services – Authentication

- **Authentication takes place before a job submission and delegation**

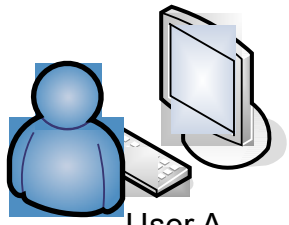
1)	Client to Server:	ClientHello
2)	Server to Client:	ServerHello ServerCertificateChain, ServerHelloDone
3)	Client to Server:	ClientCertificate, ClientKeyExchange, ClientCertificateVerify Finished
4)	Server to Client	Finished

Grid Security Services – Delegation

- **User initializes a proxy to use the grid**
- **A proxy certificate can be used by other middleware services to access resources on behalf of the user**
- **Each time a service needs access to resources a proxy is created and signed by a current valid proxy**
  - Note that the service defines the characteristics for the proxy certificate. This means that it may have very specific requirements in order to limit the effect in the case its security has been compromised



Grid Security Services – Delegation



1. Authentication using TLS

3. Transmit signed request and proxy public key

5. Return proxy certificate

4. Sign proxy certificate

2. Generate proxy key pair

Proxy Certificates - Example

- **A user creates a proxy certificate in the UI, that is proxy1.**
- **A user submits the job to the grid. The proxy1 is used to sign a new proxy2. Now the resource broker has access to the proxy2.**
- **The Resource broker submits the job to Computing Element. The proxy2 may be used to sign a new proxy3**
- **...**

- **Note that**
  - the validity period of each proxy can not be longer than that of its parent.
  - The whole procedure is transparent to the user
  - A new proxy may have special characteristics that may further limit the access

Proxy Certificate - Example

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 92 (0x5c)

Signature Algorithm: md5WithRSAEncryption

Issuer: C=CY, O=CyGrid, O=HPCL, CN=John Smith, CN=proxy

Validity

Not Before: Apr 16 07:43:23 2007 GMT

Not After : Apr 16 19:23:35 2007 GMT

Subject: C=CY, O=CyGrid, O=HPCL, CN=John Smith, CN=proxy, CN=proxy

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (512 bit)

Modulus (512 bit):

00:b2:41:50:49:59:bb:64:84:b4:b6:e1:51:62:ae:

f8:5a:97:fb:17:e8:3d:d9:96:91:e1:d3:70:45:91:

b3:1b:7d:93:53:9e:4b:bb:82:3b:b4:2e:1f:1e:53:

27:a5:8e:e7:d9:f6:62:98:8d:96:0a:8e:6f:3f:44:

24:35:17:d5:21

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Key Usage: critical

Digital Signature, Key Encipherment, Data Encipherment

Signature Algorithm: md5WithRSAEncryption

39:06:12:87:21:05:97:bb:63:8a:0e:53:07:2c:d4:98:35:f6:

12:7d:07:6c:46:0b:b9:36:e6:d5:47:90:06:7a:4b:9c:dc:95:

86:58:d0:6b:a0:4c:74:75:70:d7:e0:bc:e6:f1:5f:82:8e:f8:

29:c3:24:19:f6:fc:b3:e3:df:bf

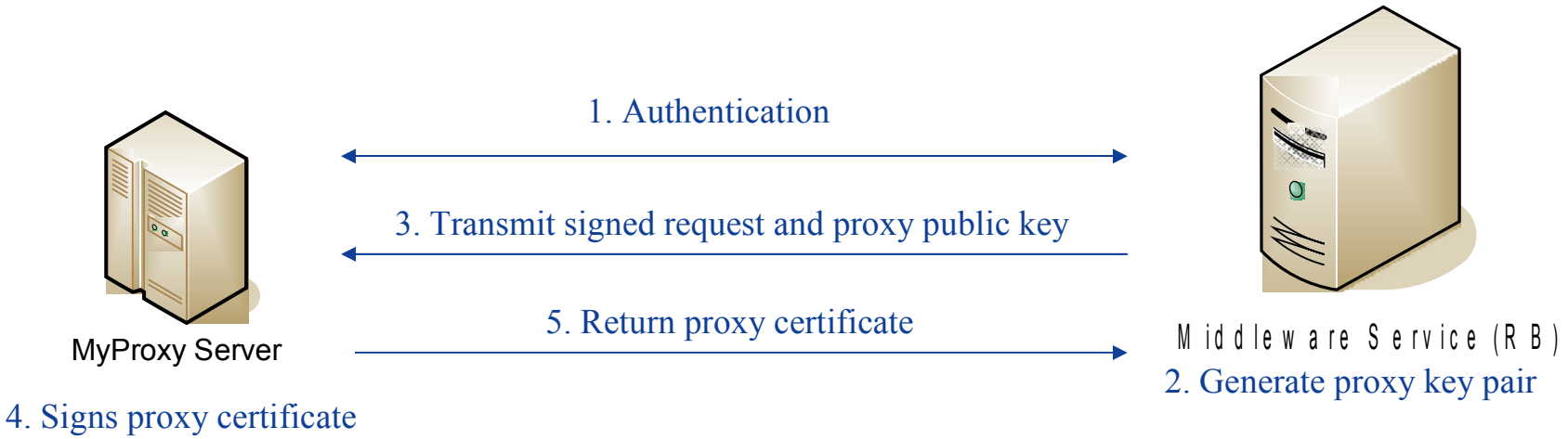
Grid Security Services – Proxy Renewal

- Proxy certificates have limited validity period
- A job that has surpassed the proxy validity period finishes immediately
- Solution -> Myproxy, a credential repository system

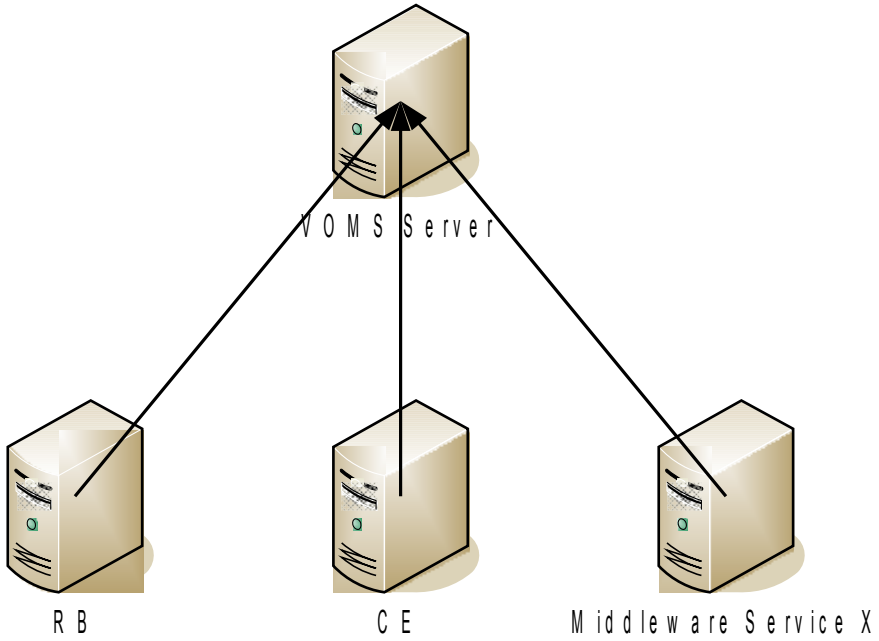
Grid Security Services – Proxy Renewal

- **A user can store a kind of long-term proxy to the myproxy server**

- **A middleware service (e.g. RB) can ask for a new proxy certificate**



Grid Security Services - Authorization



- Users are able to use the resources that support the VO to which they belong
- Middleware services maintain local user accounts that are used to support the VO
- Middleware services create a file that maps a user to a VO
- The required information is downloaded from the VOMS servers

Grid Security Services - Authorization

- **Gridmap file maps certificate DNs to local accounts**

  "/C=CY/O=CyGrid/O=HPCL/CN=John Smith" .see
  "/O=dutchgrid/O=users/O=sara/CN=Another User" .dteam

- **For each VO a set of local accounts is set up during the configuration of the middleware**

Problems

- **Communication channels are not confidential**
  - UI-RB
  - RB-BDII, RB-CE
  - WN-RB
- **Proxies generated on each step causes long certificate chains**
- **Each generated proxy requires the generation of long public cryptographic keys**
- **Trust is provided by the top level Cas**
  - The protection of the CAs public keys repository is critical

References

- **GT 4.0 Security: Key Concepts**
  - http://www.globus.org/toolkit/docs/4.0/security/key-index.html
- **gLite User Manual**
  - https://edms.cern.ch/file/722398//gLite-3-UserGuide.html
- **Cryptography: A Very Short Introduction (Fred C. Piper, Sean Murphy)**
-

Questions

Thank you!