gLite Security

Antonio Juan Rubio Montero

antonio.rubio -at- ciemat.es

CIEMAT, Madrid (Spain)

Joint CHAIN/GISELA/EPIKH School for Application Porting
Nov 29- Dic 9, 2010. Valparaíso(Chile)

Based in previous EPIKH, EGEE-III and EELA-2 presentations

Índice

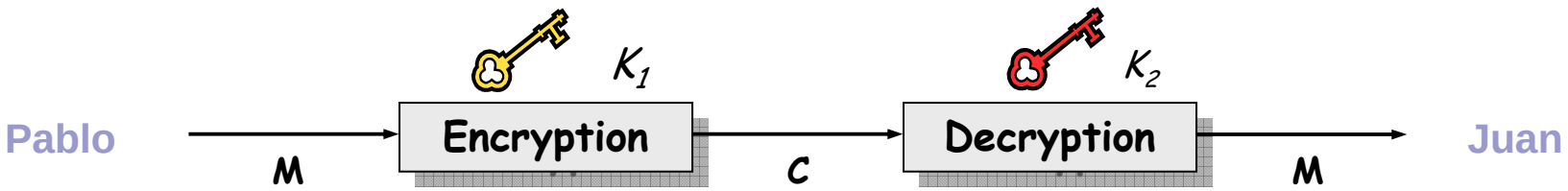
- Glosario
- Encriptación
  - Algoritmos simétricos
  - Algoritmos asimétricos: PKI
- Certificados
  - Firmas Digitales
  - Certificados X509
- Seguridad en la Grid
  - Dónde se obtienen y almacenan los Certificados
  - Comandos útiles
  - Creación de proxies de usuario
- Organizaciones Virtuales
  - Concepto de VO y autorización
  - Delegación de certificados proxy de larga duración

Glosario

- **Entidad**
  - Un usuario, un programa o una máquina
- **Credenciales**
  - Datos que proporcionan prueba de identidad
- **Autenticación**
  - Verificación de la identidad de la entidad
- **Autorización**
  - Concesión de una serie de privilegios a la entidad
- **Confidencialidad**
  - Encriptación de información de tal forma que únicamente el receptor de ésta pueda entenderla
- **Integridad**
  - Seguridad en que la información no ha sido alterada en la transacción
- **No repudiación**
  - Imposibilidad de negar al autenticidad de la firma digital

Criptografía

La disciplina matemática que estudia la seguridad de la información y de las acciones relacionadas, en particular encriptación, autenticación y control de acceso.



Simbología:

- **Texto plano: *M***
- **Texto cifrado: *C***
- **Encriptamos con clave K_1 : $E_{K_1}(M) = C$**
- **Desencriptamos con clave K_2 : $D_{K_2}(C) = M$**

Algoritmos

- **Simétrico: $K_1 = K_2$**
- **Asimétrico: $K_1 \neq K_2$**

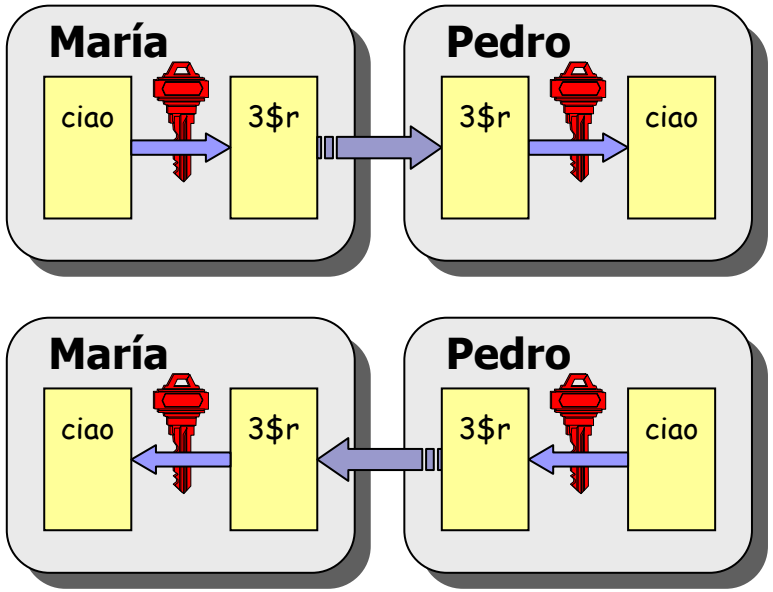
Algoritmos Simétricos

- **La misma clave es empleada para encriptar y desencriptar la infomación**

- **Ventajas**:
  - Velocidad

- **Desventajas**:
  - ¿cómo distribuir las claves?

- **Ejemplos**:
  - **DES**
  - **3DES**
  - **Rijndael** (AES)
  - **Blowfish**
  - **Kerberos**

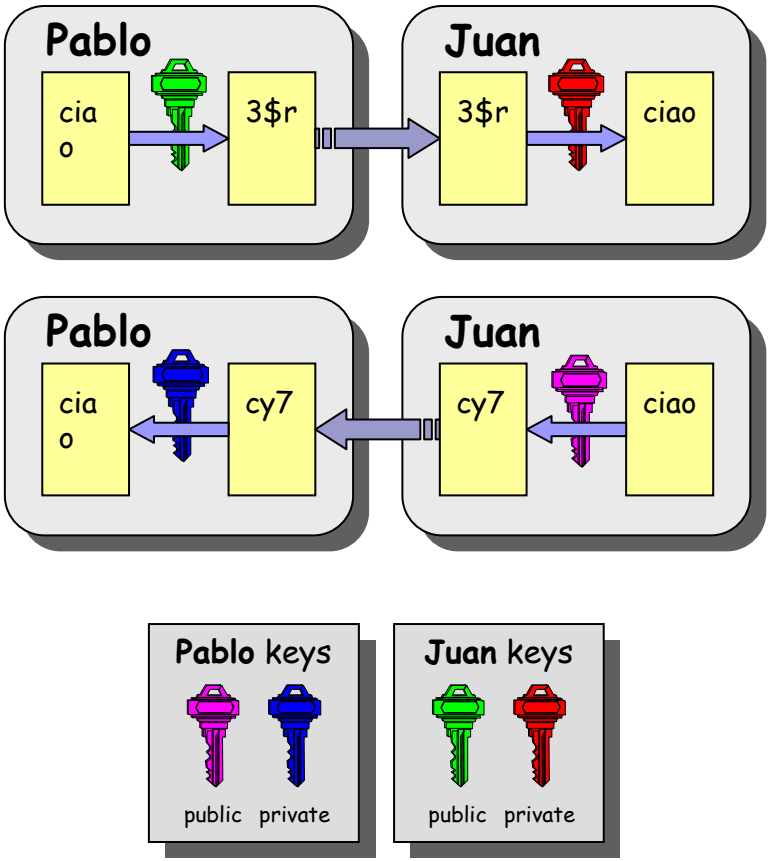


Algoritmos de clave pública

- **Todo usuario tiene dos claves: una clave pública y una privada**
  - Esto hace "imposible" obtener la clave privada a partir de la pública.
  - Un mensaje encriptado por una sólo puede ser desencriptado con la otra.

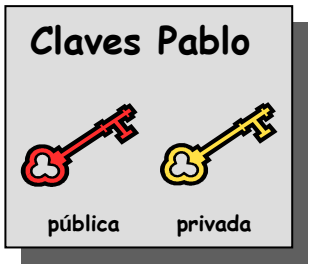
- **No es necesario un intercambio de secretos**
  - Se cifra la información usando la clave pública del recptor;
  - El receptor desencripta usando la su clave privada;

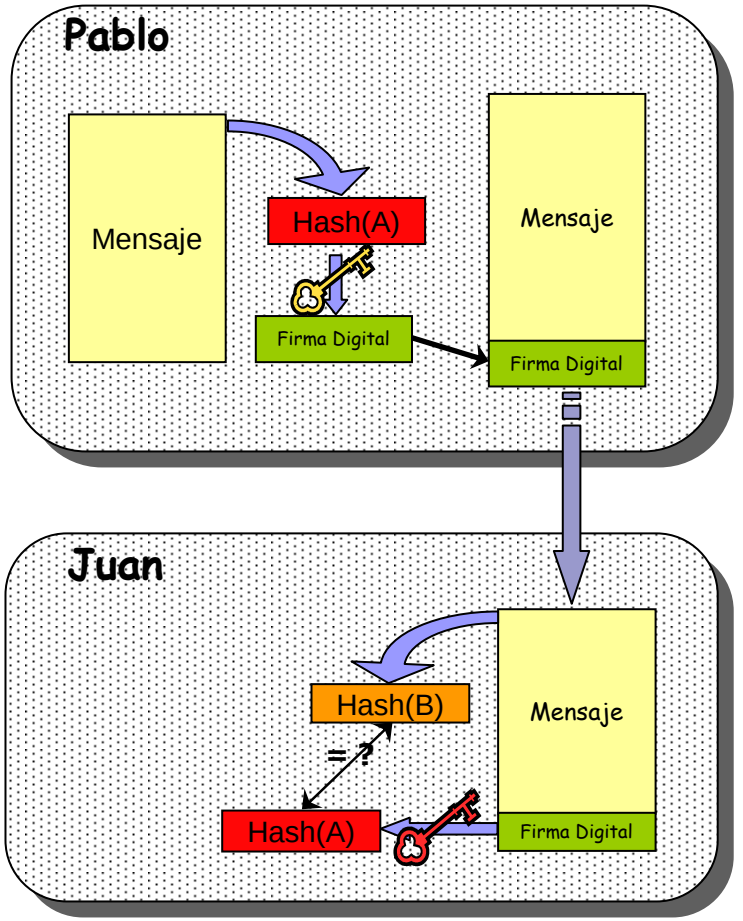
- Ejemplos:
  - **Diffie-Helmann** (1977)
  - **RSA** (1978)



Firma digital

- Pablo calcula el hash del mensaje, y a continuación encripta este hash empleando su clave privada, el hash encriptado es la firma digital.
- Pablo envia el mensaje firmado a Juan.
- Juan calcula el hash del mensaje sin la firma digital, desencripta con la clave pública de Pablo la firma.
- Si los dos hash son iguales, se garantiza la integridad del mensaje, y el principio de no repudio.





Certificados y Autoridad Certificadora (CA)

- **La firma digital de Pablo es segura si:**
  - La clave privada de pablo no está comprometida
  - Juan conoce la clave pública de Pablo

- **¿ Como puede Juan estar seguro de que la clave pública Pablo es realmente suya y no de otro ?**
  - Una tercera parte garantiza la correspondencia entre clave pública y la identidad del propietario.
  - Ambos, Pablo y Juan, deben confiar en esta tercera parte.

- **Esta tercera parte es conocida como *Autoridad Certificadora* (CA).**

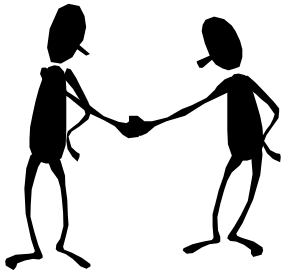
- **Emitir Certificados Digitales (contienen la identidad del propietario y su clave pública) para usuarios, servicios y máquinas (firmados por la CA)**

- **Comprobar la identidad personal del solicitante**
  - La ***Autoridades de Registro (RAs)*** realizan esta validación

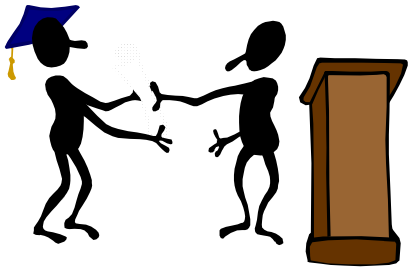
Procedimiento con la CA

Como obtener un certificado:

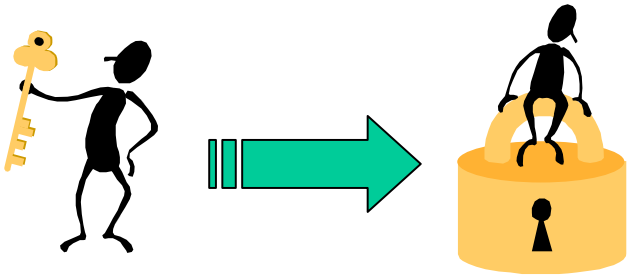
Se realiza la petición de certificado



La identidad del usuario es verificada por la RA



El certificado es emitido por la CA



El certificado es usado como llave de acceso a la grid

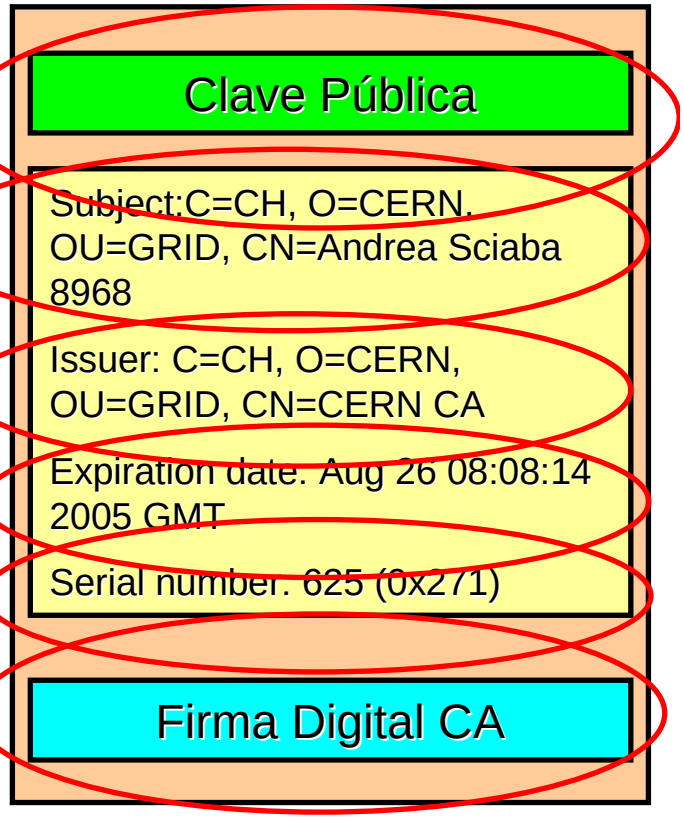
Países Latinoaméricanos y su correspondiente RA/CA

C(Country)	O(Organization)	RA	CA
AR (Argentina)	UNLP	CeSPI RA	PKIGrid CA UNLP https://www.pkiunlpgrid.unlp.edu.ar
BR (Brasil)	--	IC UFF	UFF BRGrid CA https://brgridca.ic.uff.br/
CU (Cuba)	CUBAENERGIA		
	Softel		
CO (Colombia)	UAN		
	UIS		
	UNIANDES		
	UPB		
	UT		
EC (Ecuador)	EPN		
	UTPL		
PA (Panama)	UTP		
UY (Uruguay)	UdelaR		
PE(Perú)	CIP		
	SENAMHI	SENAMHI	
CL (Chile)	--	REUNA	REUNA CA https://reuna-ca.reuna.cl
		UDEC	
		UACH	
		UTA	
		UNAP	
		UCHILE	
		UFRO	
		UCN	
		UBIOBIO	
MX (México)	UNAM	UNAMGrid	UNAM Grid https://ca.unamgrid.unam.mx/grid/
VE (Venezuela)	ULA	CeCalCULA	ULAgrid http://ra.cecalc.ula.ve

Certificados X.509

Un certificado X.509 contiene:

1. Clave pública propietario;
2. Identidad propietario;
3. Información de la CA;
4. Tiempo de validez;
5. Número de serie;
6. Firma digital de la CA:



Gestión del certificado X.509

- Una vez firmado el certificado por la CA, desde la web de IrisGrid el usuario puede descargarlo en formato PKCS12 (se puede instalar y exportar directamente desde el navegador Web)

- Para poder utilizarlo en la Grid hay que
  - separar la clave pública y privada en dos ficheros: hostcer.pem y hostkey.pem

```
openssl pkcs12 -clcerts -in <CERT.pfx/.p12> -out userkey.pem
openssl pkcs12 -clcerts -nokeys -in <CERT.pfx/.p12> -out usercert.pem
```

  - Copiar estos ficheros a $HOME/.globus
  - Cambiar los permisos, para asegurar que la clave privada sólo sea leída por nosotros mismos

```
chmod 400 userkey.pem
chmod 444 usercert.pem
```

- Se puede ver ver la información del certificado con comandos “openssl”

```
UserInterf# openssl x509 -noout -in usercert.pem -dates
notBefore=Apr 24 14:16:36 2007 GMT
notAfter=Apr 23 14:16:36 2008 GMT

UserInterf# openssl x509 -noout -in usercert.pem -subject
/DC=es/DC=irisgrid/O=ciemat/CN=juanito
```

Renovación de certificados

- El máximo tiempo de vida de un certificado es de un año
- La idea es que cuando el año esta llegando a su fin un nuevo certificado sea emitido
- Los usuarios deben ser advertidos de la proximidad de la expiración del plazo y de la necesidad de ser renovado
- No es necesaria la revocación del certificado para emitir uno nuevo a no ser que éste sea comprometido o el usuario cese la actividad por la cuál se solicitó el certificado

Renovación de certificados

- Durante el proceso de renovación del certificado no es necesario repetir el proceso de identificación:
  - Esto es una gran ventaja tanto para usuarios como para RA's
  - Pero es aconsejable el establecer un número máximo de renovaciones sin identificación (por ejemplo: hacer pasar a las entidades por el proceso de identificación cada dos años)

- Si es necesario sin embargo que la petición de renovación sea realizada con la firma del certificado de usuario, ejemplos:
  - E-mail firmado con el certificado de usuario
  - Una interfaz Web CA/RA que pudiera identificar el certificado de usuario

- Crear hostcert.pem con un .cer de una RA. (Ya que el hostkey se mantiene).

```
openssl x509 -inform der -in a12b12c2.cer -out usercert.pem
```

- Si el certificado expira antes de la renovación deberá seguirse el proceso como si fuese una nueva solicitud

Certificados Proxy X.509

- **Puede se comprometido el transferir tu certificado personal a través de la Grid**

- **Certificados Proxy**
  - Puede ser firmado por un certificado final de entidad (o por otro proxy).
  - Soporta algunas características importantes
    - Delegación
  - Tiene un tiempo de vida limitado (minimiza el riesgo de que las credenciales sean comprometidas)

- **El certificado Proxy se crea con el comando grid-proxy-init:**

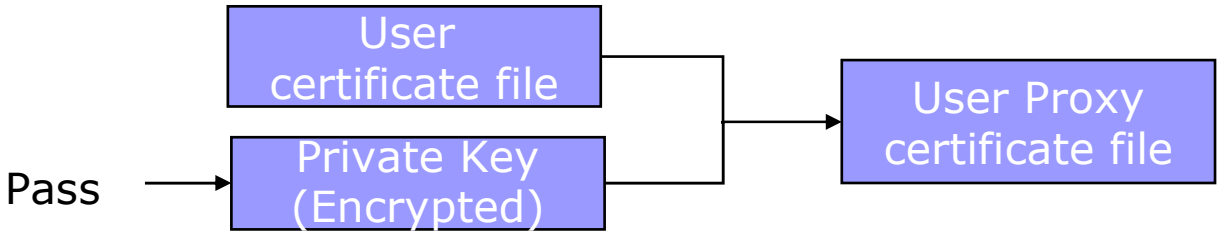
```
grid-proxy-init
  Enter PEM pass phrase: ******
```

  - Opciones de grid-proxy-init:
    - -hours <tiempo de vida de las credenciales>
    - -bits <tamaño de la clave>
    - -help

Generar un proxy: grid-proxy-init

- **El usuario introduce el pass, que es empleado para desencriptar la calve privada**

- **La clave privada se emplea para firmar el certificado proxy con el certificado de usuario. Se generan en el proxy un nuevo par de claves pública/privada**
  - La clave privada de usuario no es expuesta tras la firma del proxy



Proxy

Se almacena en un fichero local: debe ser válido durante un periodo corto de tiempo (tipicamente 12 h) para minimizar los riesgos de seguridad.

Generar un proxy: grid-proxy-init

- **grid-proxy-init ≡ “login en el Grid”**
- **“logout” en el grid debes destruir el proxy:**

```
grid-proxy-destroy
```

- **Para obtener información de nuestro proxy:**

```
grid-proxy-info
```

  - Opciones

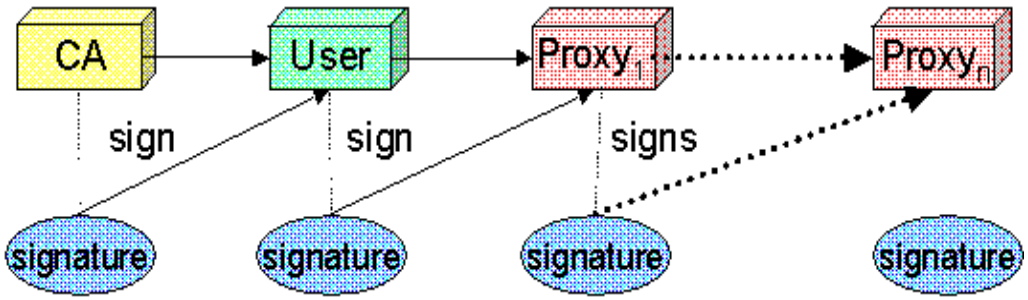
-subject	-issuer
-type	-timeleft
-strength	-help

- **Simplemente traducen los comandos “openssl” anteriores al aplicarse sobre el fichero de proxy**

Delegación

- **Delegación = creación remota de (segundo nivel) credenciales proxy**
  - El nuevo par de claves se generan remotamente en el servidor
  - El cliente firma el certificado proxy

- **Permite a procesos remotos autenticarse en nombre del usuario**



Prácticas con certificados: Acceso a la UI del

- UIs configuradas:
  - “cluster29.fis.utfsm.cl” → gilda VO
  - “cluster30.fis.utfsm.cl” → prod.vo.eu-eela.eu VO

- Nombres de usuario --> valparaiso01....20
- Contraseñas --> xxxxxx01....20
- Contraseña de los certificados de la VO gilda → “VALPARAISO”

- Modo de acceso: SSH
  - Desde Windows: usar Putty
  - Desde Linux: ssh -l <valparaisoXX> clusterXX.fis.utfsm.cl

Prácticas con el certificado personal

1. Crea los archivos usercert.pem y userkey.com desde el certificado .p12
2. Mueve los .pem al directorio correcto y comprueba los permisos.
3. Inspecciona la información del certificado con comandos “openssl”:
   1. La fecha de caducidad
   2. El DN del propietario
   3. La CA que ha firmado el certificado
4. Crea un proxy e inspecciónalo con grid-proxy-info su contenido (atención al “timeleft” y “path”).
5. Busca el fichero del proxy y haz un “diff” de su contenido contra la clave publica de tu certificado .globus/usercert.pem
6. Inspecciona ahora el fichero del proxy con comandos “openssl”. Destruye el certificado y comprueba el fichero se ha borrado

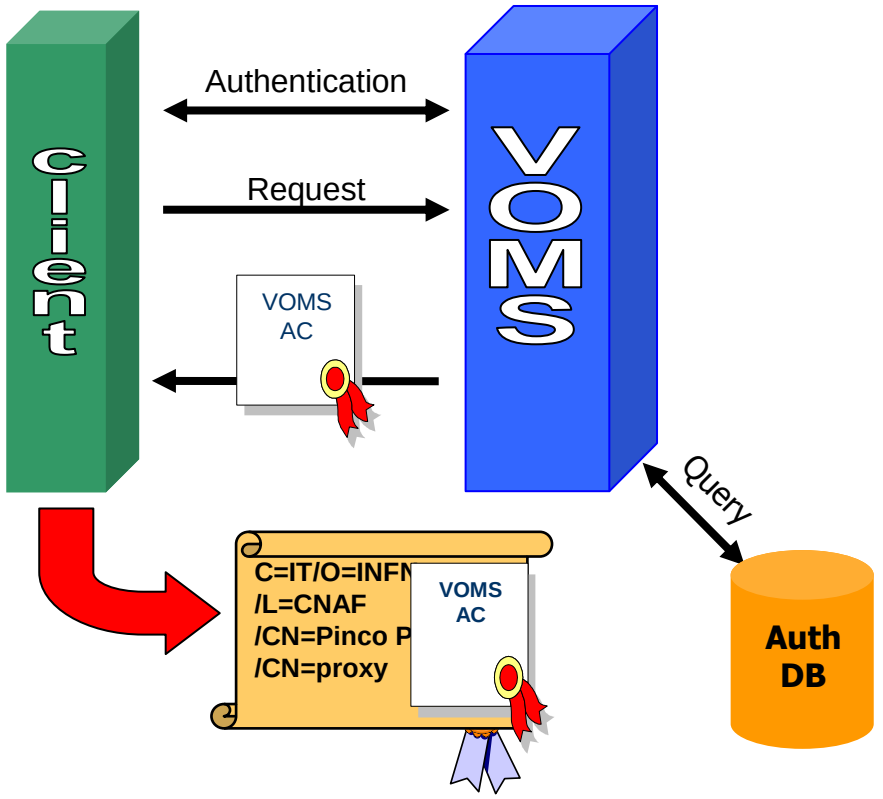
Organizaciones Virtuales (VOs)

- Conceptos:
  - Las VOs **agrupan usuarios** y recursos cedidos a un proyecto por las instituciones en el mismo dominio de administración virtual.
  - En EGI las VOs **corresponden** a organizaciones y reales o **proyectos científicos**: experimentos LHC, comunidad de investigadores biomédicos, comunidad de astrónomos,…
- Aceptar una aplicación en una VO:
  - Área científica de la VO
  - Tenga interés
  - Pase una fase de pre-producción
- Admitir a un usuario de una aplicación aceptada
  - Tener un certificado válido de usuario Grid
  - Ser autorizado por un administrador de la VO (comprueba que pertenece a la comunidad)

VOMS: creando un proxy firmado por una VO

- Extiende la información del certificado proxy con la VO, grupo, roles en la VO de ese usuario.
- Se crea con el comando: voms-proxy-init --voms <VO>
- Se utiliza y se delega como un certificado normal.
- Habilita a usar los recursos asignados para una VO en cada recurso Grid.

(Virtual Organization Membership Service)



VOs y autorización

- **Los usuarios Grid deben pertenecer a organizaciones virtuales**
- **VOs mantienen una lista de sus miembros en un servidor LDAP**
  - Esta lista es descargadas por las máquinas que ofrecen servicios o recursos Grid (No la UI) para mapear los certificados de usuarios con “pools” de cuentas locales.
- Ejemplo de grid-map en un CE:

```
...
"/C=CH/O=CERN/OU=GRID/CN=Simone Campana 7461" .dteam
"/C=CH/O=CERN/OU=GRID/CN=Andrea Sciaba 8968" .cms
"/C=CH/O=CERN/OU=GRID/CN=Patricia Mendez Lorenzo-ALICE" .alice
...
```

Creación de un proxy con extensión VOMS

- **Similar al grid-proxy-init, pero ahora el VOMS va a firmar y añadir información al certificado referente a la VO y del rol de usuario que éste ocupa en la VO**.

```
voms-proxy-init  --voms <VO_name>
```

- **Para obtener información de un proxy**

```
voms-proxy-info -all
```

  - **Se muestran dos tiempos de vida diferentes**:
    - El primero relacionado con el proxy en sí mismo
    - El segundo es parte la la información añadida por el servidor VOMS

- **El Proxy tiene un tiempo de vida limitado (por defecto 12 h)**

- **Una tarea en el grid puede necesitar un tiempo de ejecución más largo**
  - Hay jobs que pueden necestar tiempos de ejecución de hasta 2 días (el máximo tiempo por defecto de ejecución en una cola PBS)

- Se podría hacer un grid proxy init de miles de horas, pero los servicios Grid no se creen que ese proxy sea válido pasadas 24:

```
voms-proxy-init -hours 99999 -debug -verify
```

Proxies de larga duración

- **MyProxy server:** Permite crear y guardar certificados proxy de larga duración

```
myproxy-init --voms <VO_name> -s <myproxy_server>
```

  - –d permite crear un proxy de larga duración con nombre el DN del certificado, si no se utiliza el nombre será el mismo que el usuario local de la máquina

```
myproxy-init --voms gilda –s <myproxy_server> -d
```

  - –l permite nombrar al proxy con cualquier nombre

```
myproxy-init --voms gilda –s <myproxy_server> –l <random_name>
```

- Cada usuario puede crear y almacenar varios proxies en un servidor MyProxy,
- Se puede obtener información de cada proxy según su nombre:

```
myproxy-info –s <myproxy_server>
myproxy-info –s <myproxy_server> -d
myproxy-info –s <myproxy_server> –l <random_name>
```

- Si en la UI no hay un proxy local, es imposible autenticarse con el servidor MyProxy.
- En este caso es necesario conseguir un proxy delegado desde el servidor MyProxy o bien crear un nuevo proxy local con:

```
voms-proxy-init
```

Proxies delegados desde MyProxy

- Si se pretende obtener un proxy delegado hay que asegurarse de destruir los proxies locales que existan:

```
voms-proxy-destroy
voms-proxy-info
                couldn´t find a valid proxy
```

- Para obtener el proxy delegado previamente se le llama según su nombre

```
myproxy-get-delegation –s <myproxy_server>
myproxy-get-delegation –s <myproxy_server> –d
myproxy-info  –s <myproxy_server> –l <random_name>
```

- Se verifica después que tenemos de nuevo el proxy anterior:

```
voms-proxy-info
```

- Si se destruye el proxy remoto

```
myproxy-destroy –s <myproxy_server>
myproxy-destroy –s <myproxy_server> -d
myproxy-destroy –s <myproxy_server> –l <random_name>
```

- Se puede comprobar que ya no existe

```
myproxy-info –s <myproxy_server>
myproxy-info –s <myproxy_server> –d
myproxy-info –s <myproxy_server> –l <random_name>
```

Prácticas con proxies firmados por un VOMS

1. Crea un proxy firmado por la VO gilda o prod.vo.eu-eela.eu, con un voms-proxy-init

2. Crea uno reomoto de larga duración usando como servidor MyProxy a “myproxy.ct.infn.it” y con un nombre aleatorio

3. Inspecciona la información de tu proxy , destrúyelo y obtén un proxy delegado servidor MyProxy y vuelve a inspeccionar la información.

4. Intenta enviar un trabajo con un proxy delegado (con las explicaciones de temas posteriores), con un comando como:

```
glite-wms-job-submit -d <del_proxy_name> test.jdl
```