Open Science Grid

Security in OSG

Tuesday afternoon, 3:15pm

Igor Sfiligoi <isfiligoi@ucsd.edu>
Member of the OSG Security team
University of California San Diego

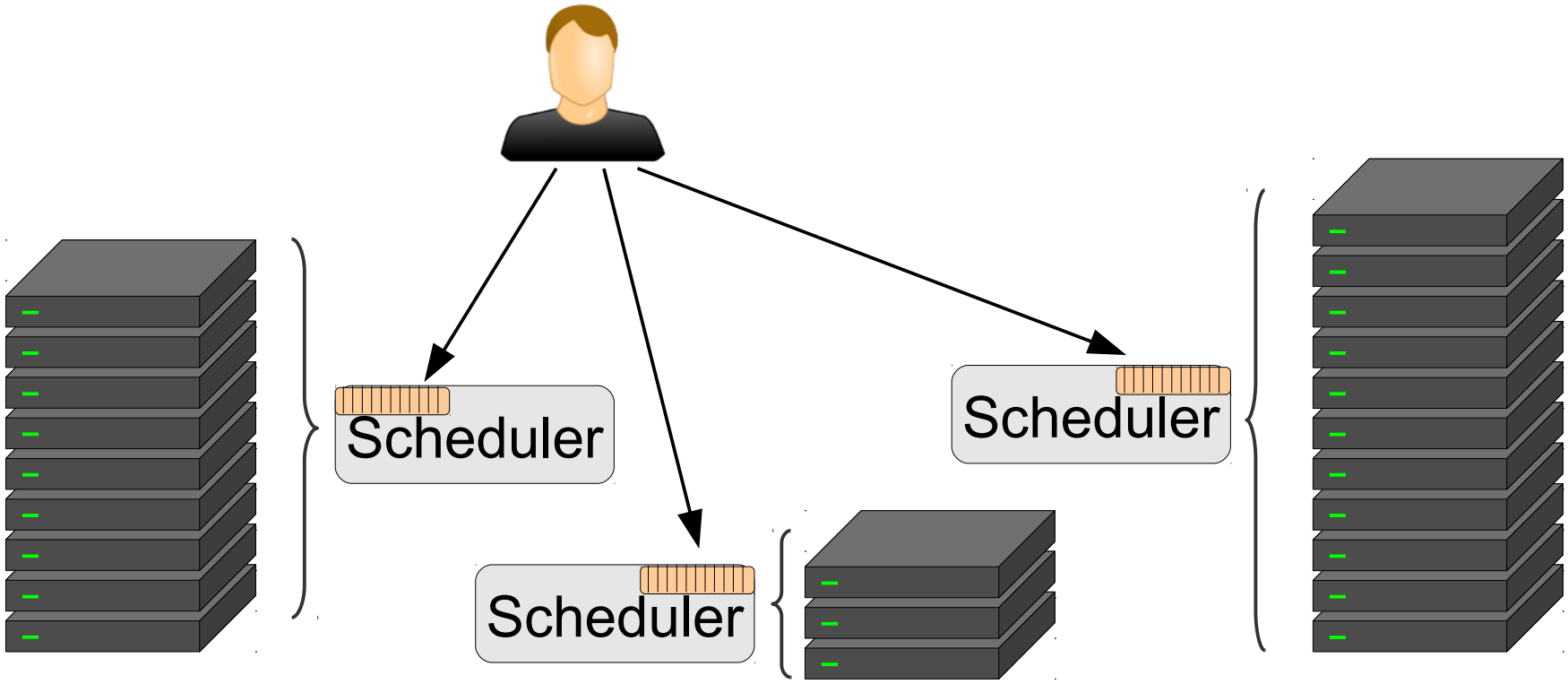
University of California
San Diego

Logistical reminder

- It is OK to ask questions
  - During the lecture
  - During the demos
  - During the exercises
  - During the breaks
- If I don't know the answer,
  I will find someone who likely does

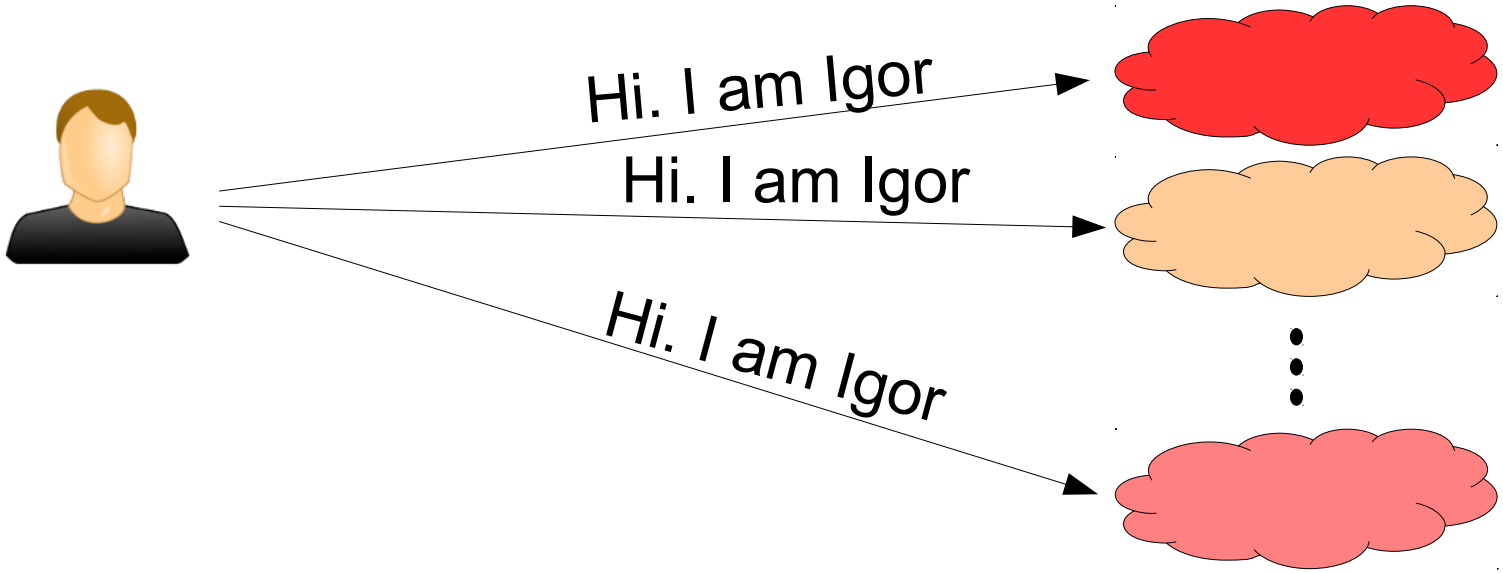
Grid and DHTC

- Grid, as a form of DHTC, is about computing on more than one cluster



Single sign-on

- The user should use the same mechanism to submit jobs to any site
  - And there are 100s of them in OSG

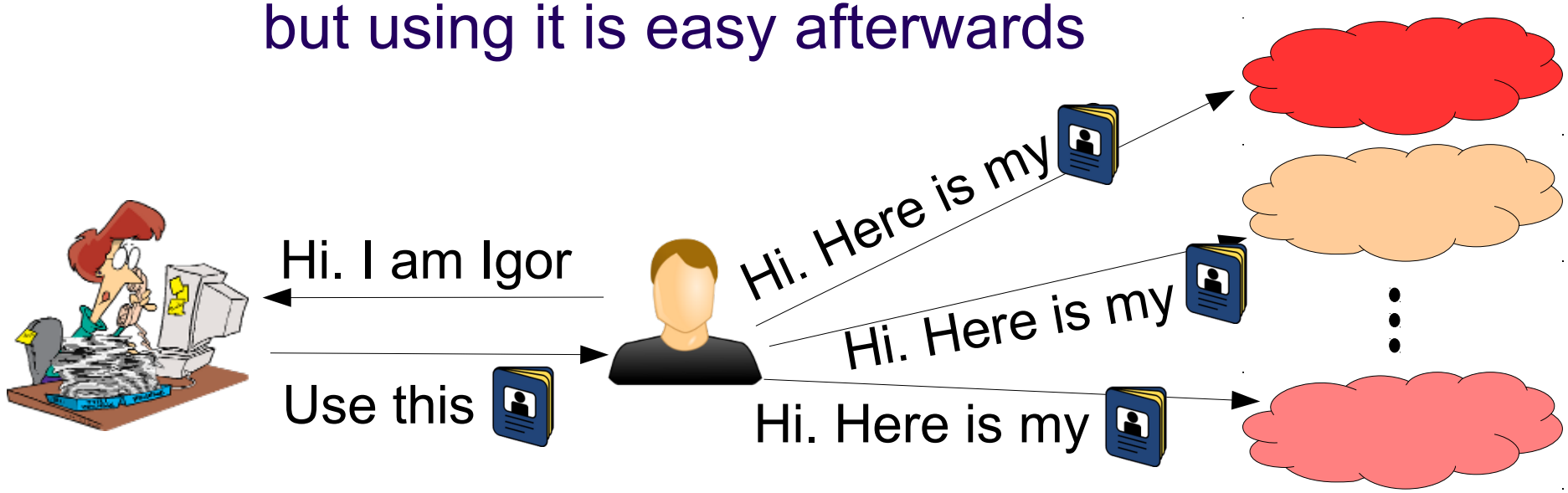


OSG Auth not password based

- OSG not using password authentication
  - For several reasons
- For starters, it is effectively a **shared secret** between **the user and the service provider**
  - And secrets stay secret only if few entities know it
  - Imagine sharing passwords between 100s of site operators

Adding an intermediary

- An alternative approach is to introduce a highly trusted intermediary
- Have been used in real life for ages
  - e.g. States as issuers of IDs/Passports
  - Getting the ID can be a lengthy process, but using it is easy afterwards



Adding an intermediary

- An alternative approach is to introduce a highly trusted intermediary
- Have been [illegible] for ages
  - e.g. [illegible]
  - [illegible]

Chain of trust.

You are trusted because the site trusts the issuer, and the issuer trusted you.

Hi. I [illegible]

Use this

Hi. [illegible] my

Hi. Here is my

Technical implementations

- Many technical solutions
  - x.509 PKI
  - Kerberos
  - OpenID
  - many more...
- All based on the same basic principle
  - Each has strengths and weaknesses
  - OSG standardized on x.509

Will not argue
if it is the best one.

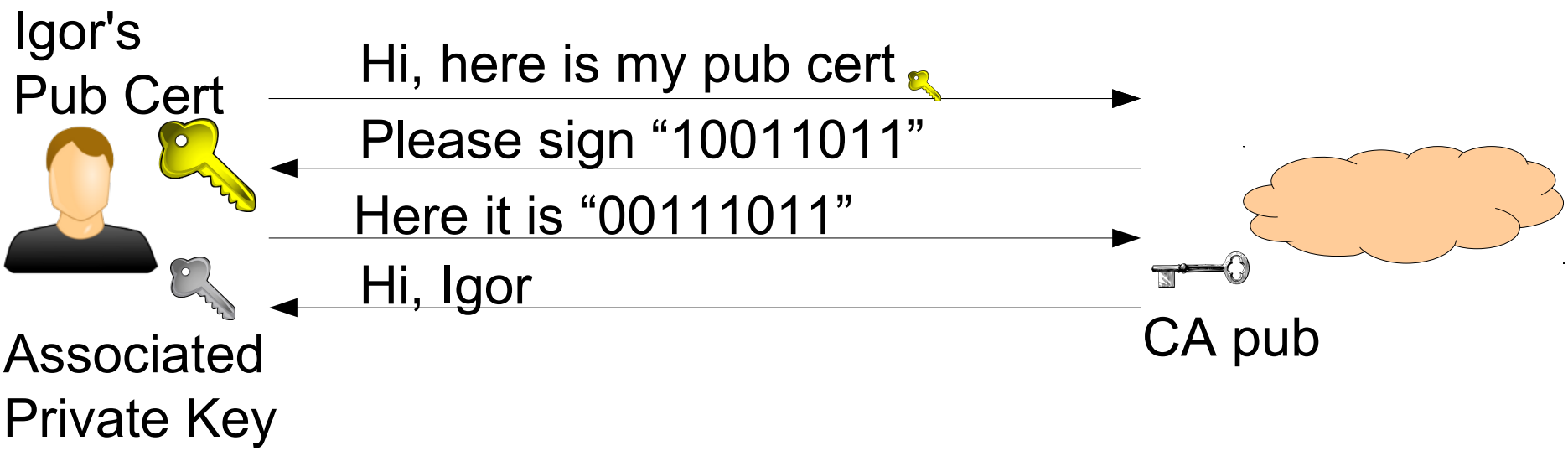
x.509 PKI

- Based on public key cryptography
  - A user has a (private,public) key pair
    - One signs, the other verifies
- The highly trusted entity is called a **Certification Authority** (CA)
  - The user is given a **certificate**
  - Cert. has user name in it
  - Cert. also contains the (priv,pub) key pair
  - Cert. has a limited lifetime
  - Cert. is signed by the CA private key

You should have one by now.

x.509 authentication

- Sites have CA public key pre-installed
- User authenticates by signing a site provided string and providing the public part of the cert

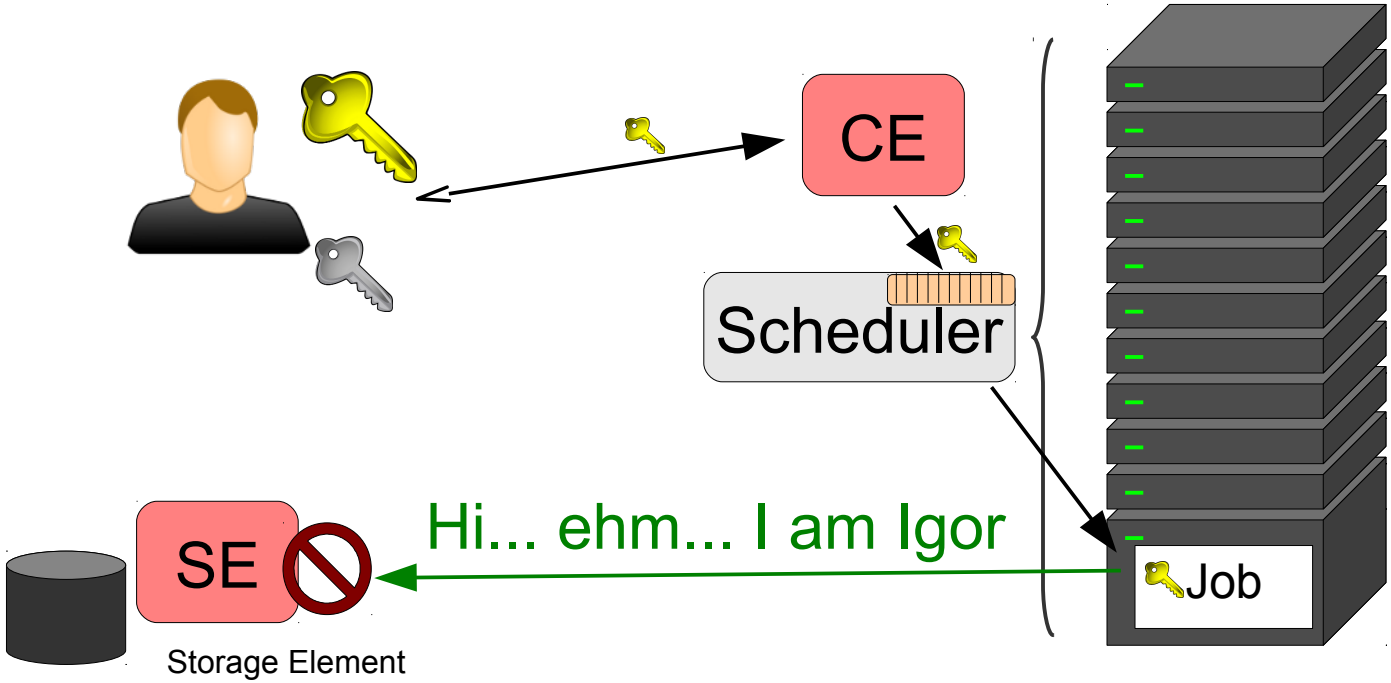


Mutual authentication

- The OSG clients also require servers to authenticate
  - Same principle as before
  - The site's server owns a x.509 certificate
  - User client must have the CA pre-installed
- So we have mutual authentication

Impersonation

- Sometimes your jobs need to impersonate you
  - For example to access remote data

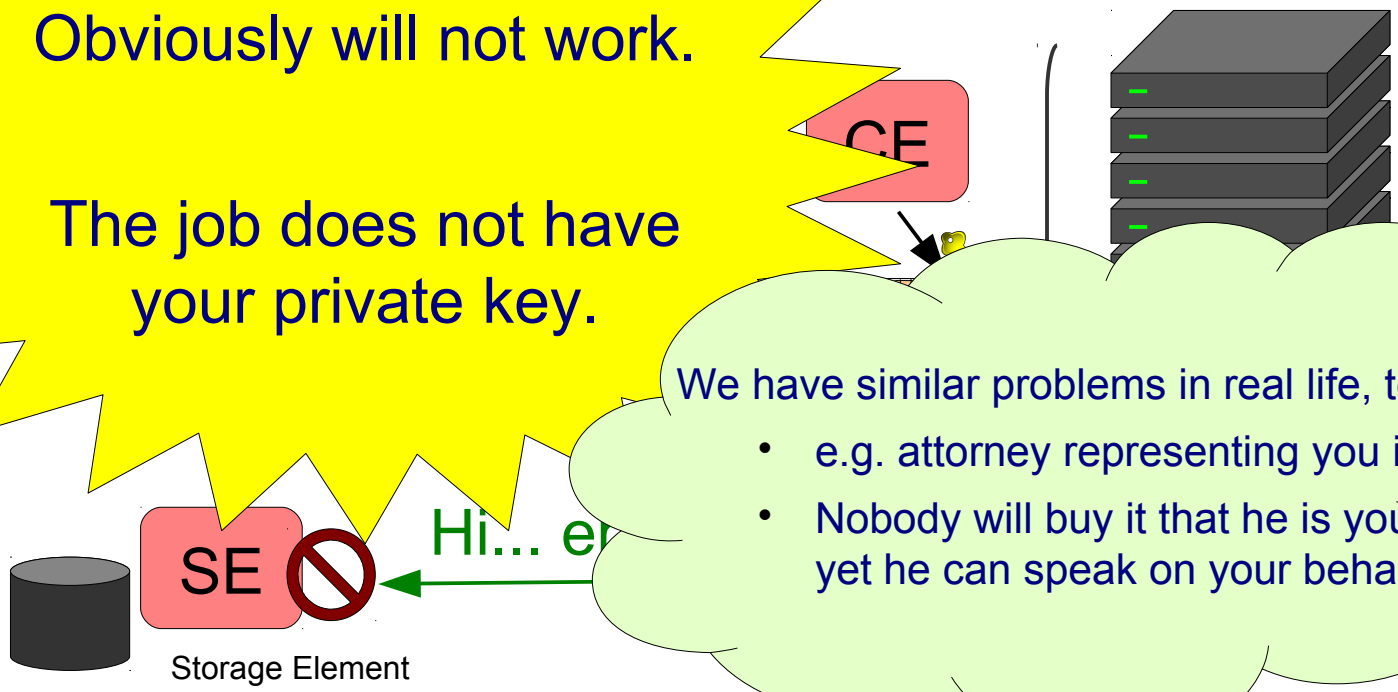


Impersonation

- Sometimes your jobs need to impersonate you
  - H access remote data

Obviously will not work.

The job does not have your private key.



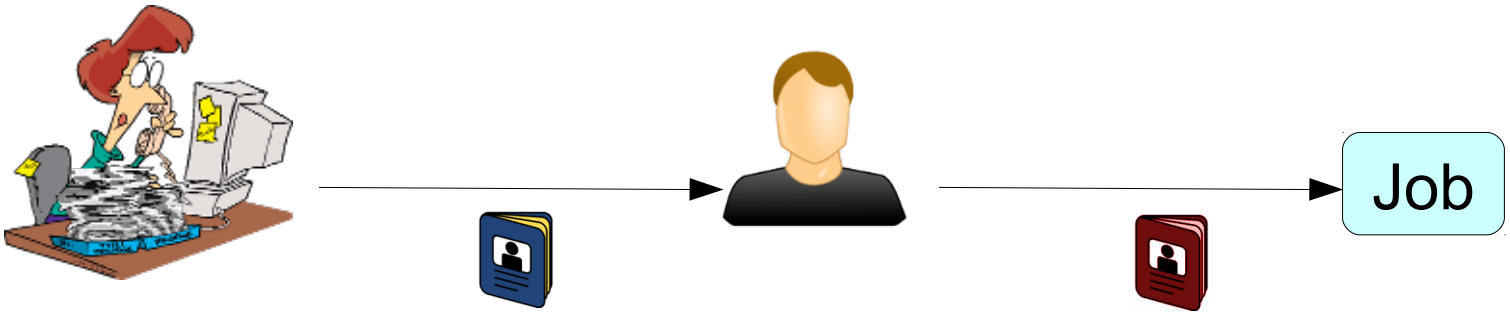
We have similar problems in real life, too

- e.g. attorney representing you in court
- Nobody will buy it that he is you, yet he can speak on your behalf

Proxy delegation

One more reason why we do not use passwords

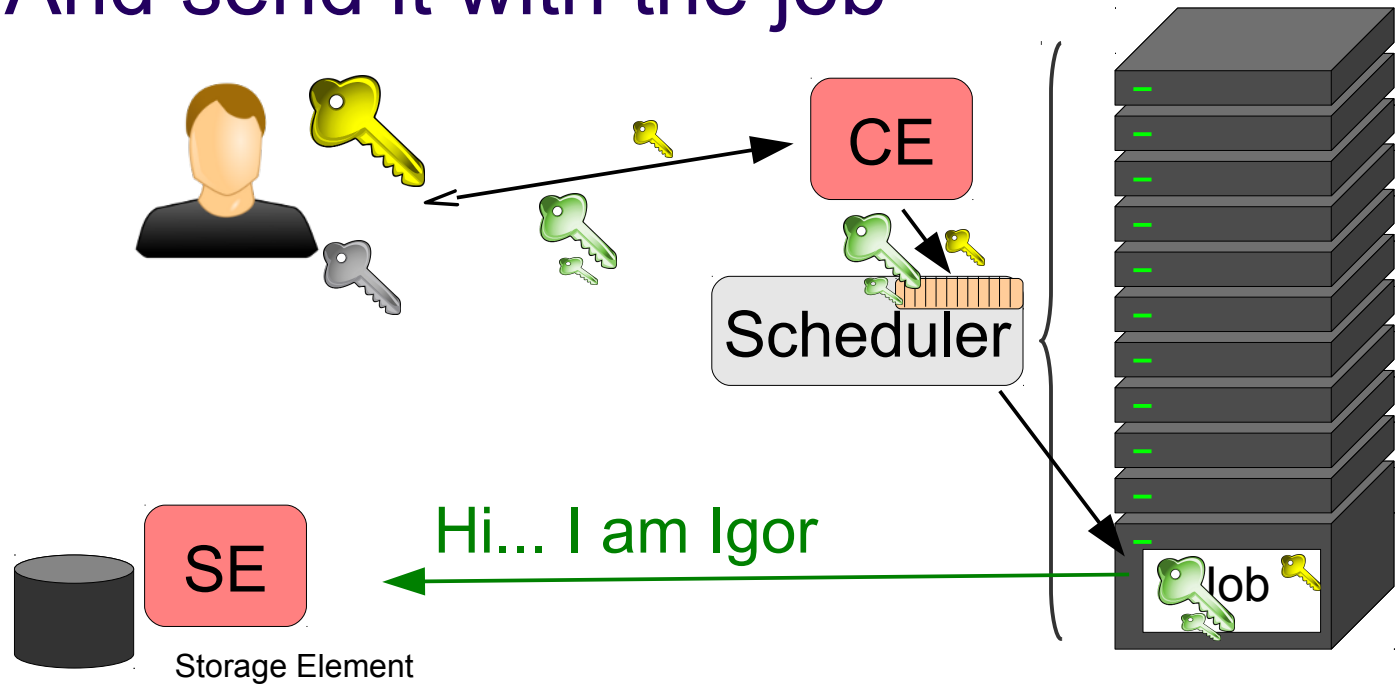
- The **job is** indeed **not you**
- Create a proxy certificate for the job
- A proxy is a new certificate issued by your (old) existing certificate
  - Just add another level of trust delegation



Proxy delegation

One more reason why we do not use passwords

- The **job is** indeed **not you**
- Create a proxy certificate for the job
  - Add another level of trust delegation
- And send it with the job



Proxy delegation

One more reason why we do not use passwords

- The **job is** indeed **not you**
- Create a proxy certificate for the job
  - Add an [illegible] delegation
- And [illegible]

YES!

You are sending the **proxy's private key** to the WN.

Risk mitigation

- Proxy delegation is risky
  - Your proxy could be stolen
- In OSG, we mitigate by limiting lifetime
  - At most few hours recommended
  - After the proxy expires, the proxy is useless
- Can be annoying
  - Must keep renewing, if long running job!

But we don't have anything better.

Risk mitigation

- Proxy delegation is risky
  - Y[illegible] stolen
  - [illegible] limiting lifetime
    [illegible] nded
    [illegible] proxy is useless
  - C[illegible]
    - Must keep renewing, if long running job!

If using HTCondor,
it will automatically
create a short lived proxy
and keep re-delegating it.

But we do [illegible]
anything [illegible]

Completely
transparent
to you.

Authentication vs. Authorization

- Just because you can authenticate yourself, it does not mean you are authorized, too
  - e.g. your passport tells who you are, but does not allow you to drive a car
- x.509 PKI only covers **authentication**
  - Tells the site who you are

Need a different mechanism for authorization

Per-user authorization not an option

- The naive approach is using a list
  - Since we do not want let just anyone in!
- However, the problem is scale
  - OSG has ~10,000 users!
  - Sites do not want to decide on a user-by-user basis!

Server authorization is easy.

Just require host name in the certificate name; CA will enforce this.

The client decides which host to talk to.

Adding roles

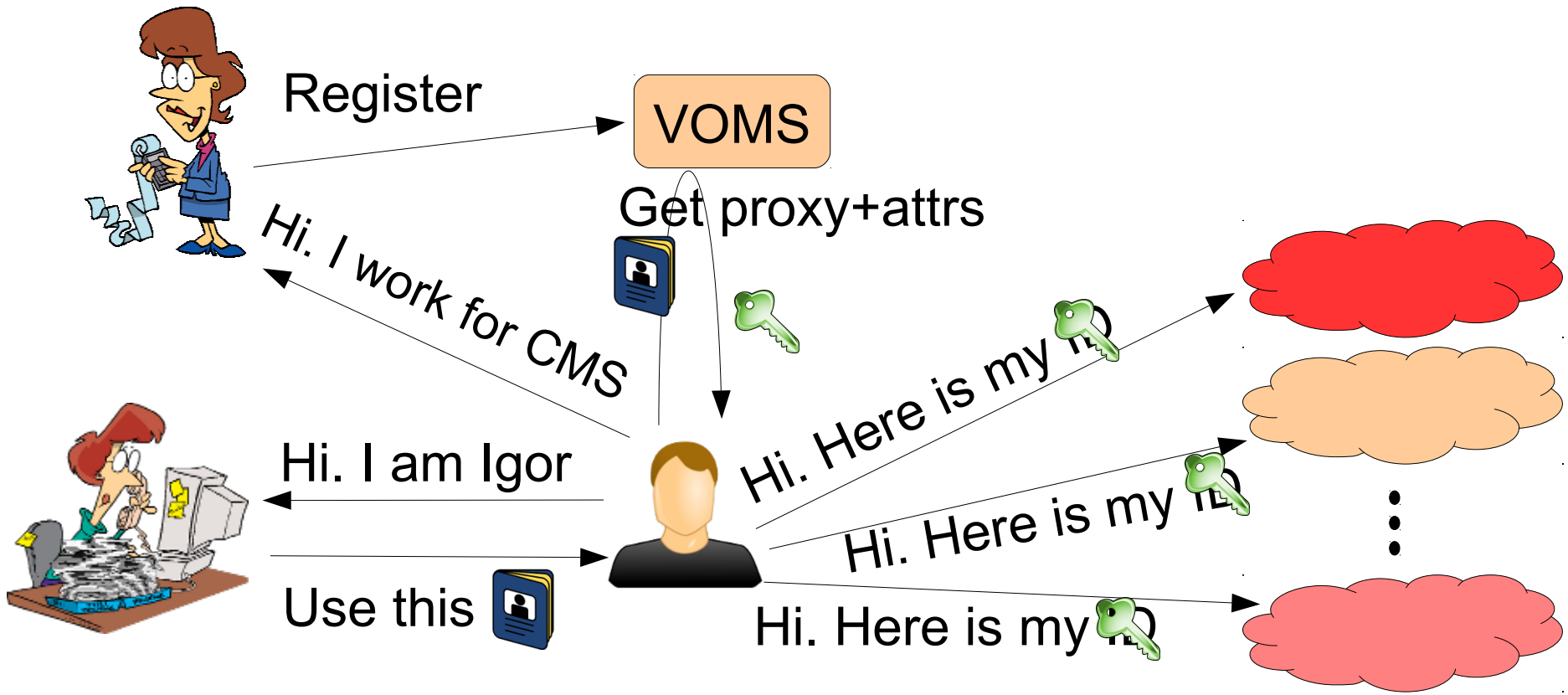
- Sites want to operate on higher level concepts
  - Some kind of attribute
- Like in real life
  - Think about passport vs driver's license
  - Both tell a cop who you are
    (and to 1st approx. are issued by the same entity)
  - But the driver's license tells him
    you are allowed to use a car, too
    - “Class:C”

Need for an attribute authority

- Users can have many roles
  - But don't want to have multiple certs
  - e.g. I may be running HEP jobs or School jobs
- So the attributes cannot come from the CA
  - And you would not just trust the user
- In OSG, we use VOMS
  - Virtual Organization Management System
  - OSG expects well organized VOs (e.g. CMS)

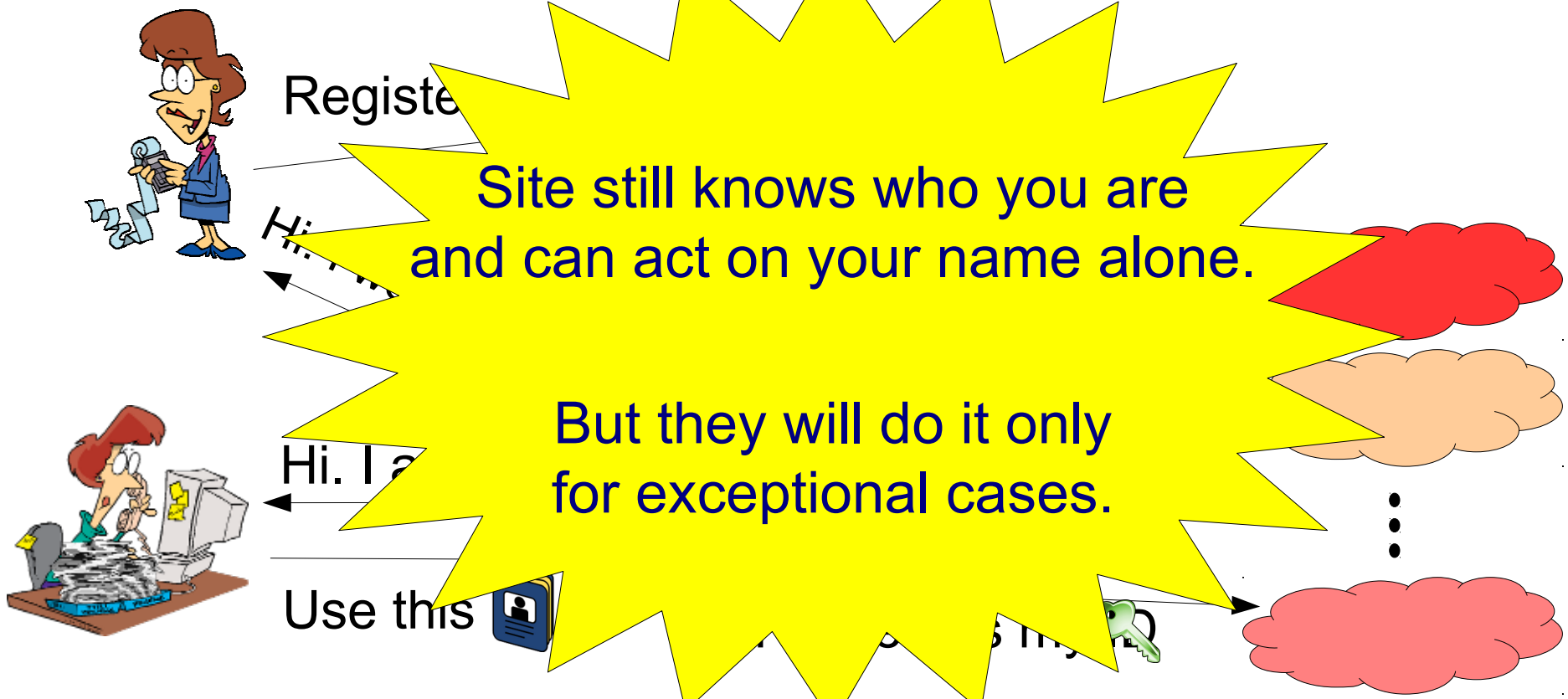
VO and VOMS

- VO decides who is worthy of an attribute
  - Site decides based on that attribute



VO and VOMS

- VO decides who is worthy of an attribute
  - Site decides based on that attribute



More security considerations

- There is much more than authentication and authorization to security
  - But we don't have the time to cover everything
- Just briefly
  - Sharing of resources
  - Privacy
  - Acceptable conduct

Sharing of resources

- Modern CPUs are many-core, so
  - Very likely your job will be sharing the node with other jobs
- Sites will map your Grid name into UID
  - Hopefully unique... be sure to ask
- Standard *NIX protections
  - Act accordingly
  - e.g. no file should be world writable

Privacy

- By default, no privacy in OSG
  - Assume all your files are publicly readable
  - Apart from your proxy
- If you need privacy, you will have to take explicit measures
  - Both during network transfers, and
  - For files on disk
- x.509 can be used for encryption
  - But remember, proxy has new keys

Acceptable conduct

- Each OSG user is bound by its AUP (Acceptable User Policy)
  - And sites are allowed to have additional rules in place
- In a nutshell
  - Use only for the declared science purpose
  - Do not overload the system
  - Do not attempt to circumvent security

Else,
you will be banned!

Questions?

- Questions? Comments?
  - Feel free to ask me questions later:
    Igor Sfiligoi <isfiligoi@ucsd.edu>
- Upcoming sessions
  - Now – 5:00pm
    - Hands-on exercises
  - 5:00pm - 7:00pm
    - On your own
  - 7:00pm – 9:00pm
    - Evening work session (optional but recommended)

Security is serious business