GlobusWORLD 2012 Tutorials

Admin Track

GlobusWORLD 2012

April 10, 2012

GlobusWORLD 2012 Admin Tutorials

The goal of the Admin Track is to show you how to take an existing HPC resource and turn it into a GO endpoint.

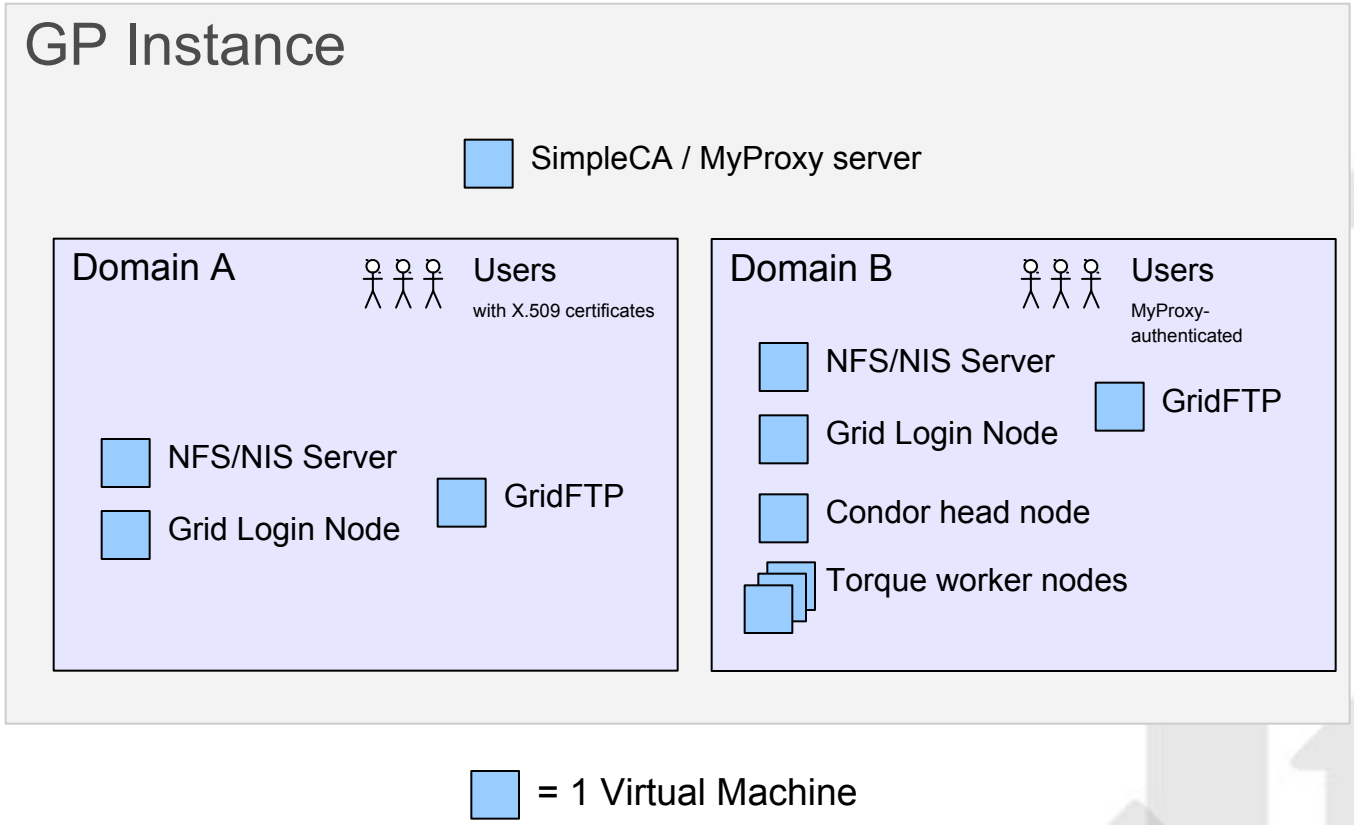
But what HPC resource will we use for this tutorial?

Globus Provision

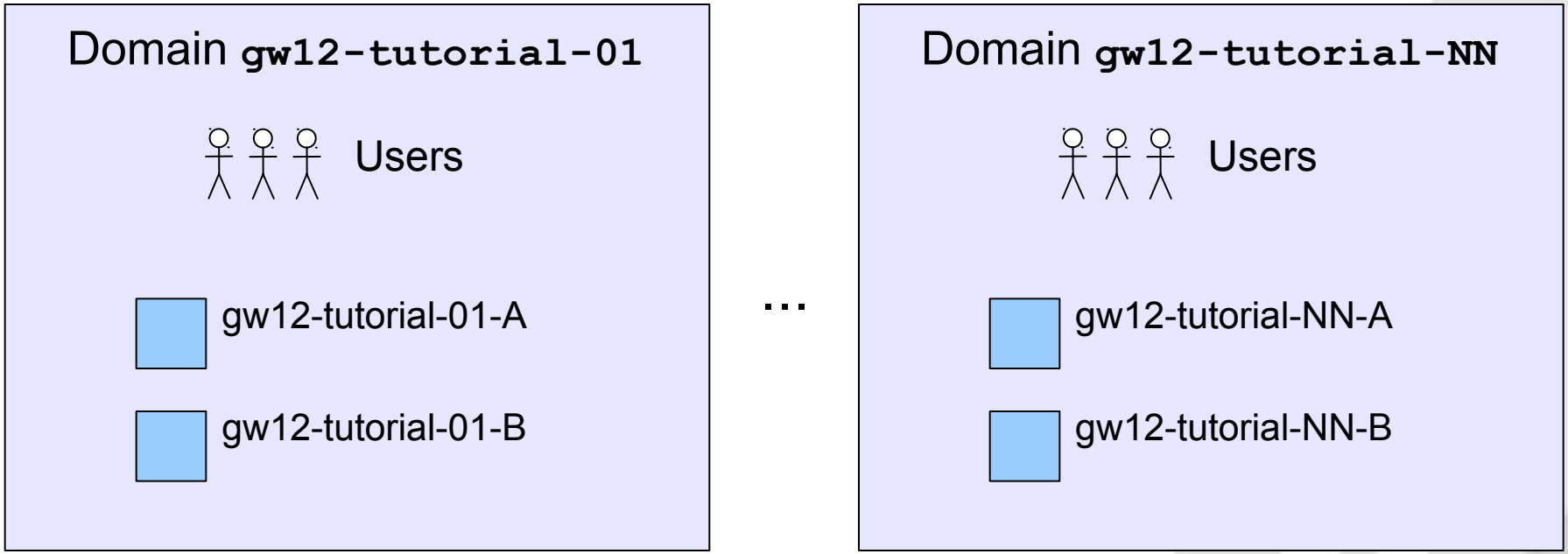
A tool for deploying fully-configured Globus systems on Amazon EC2.

http://globus.org/provision/

Example GP Instance



GP instance for this tutorial



Log into your A host

Your slip of paper has two hosts: one for the GCMU portion of this tutorial, and one for the Globus Toolkit portion of this tutorial.

Log in as user "gw12":

ssh gw12@ec2-xx-xx-xx-xx.compute-1.amazonaws.com

Use the password on the slip of paper.

`gw12` has passwordless sudo privileges.

Log into your A host

Upload gw12's SSH key to Globus Online (/home/gw12/.ssh/id_rsa.pub).

Each host also has five users: joe, bob, sue, ann, and sam.

Password for joe, bob, sue, ann: the same password as the "gw12" user.

Password for sam is “gw12tutorial” for all hosts.

Globus Online Endpoint Setup with Globus Connect Multi-User

GlobusWORLD 2012

April 10, 2012

Globus Connect Multi-User

- **What is GCMU?**
  - Globus Connect version for easily creating shared endpoints
  - Packages a GridFTP server and MyProxy CA authentication server, pre-configured for use with Globus Online
- **Why GCMU?**
  - Create transfer endpoints in minutes
  - Avoid complex GridFTP install
- **To download: https://www.globusonline.org/gcmu/**

University of Michigan

"We used GCMU to form a campus-wide GSI authentication service spanning multiple servers. Now my users have a fast, easy way to get their data wherever it needs to go, and the setup process was trivial."
--University of Michigan

OAK RIDGE National Laboratory

"As a resource admin, I've found GCMU an exceedingly useful tool.... With GCMU, setting up a GridFTP server and handling authentication for multiple users is easy."
--Oak Ridge National Lab

From zero to GO in three steps...

Let's say we want to turn our resource into a GO endpoint. We will need to:

1. Provide our local users a way of authenticating themselves with GO and other endpoints.
2. Set up a GridFTP server that will be in charge of actually serving the files.
3. Create a GO endpoint.

… or just one step

Let's say we want to turn our resource into a GO endpoint. We will need to:

1. Install Globus Connect Multi-User

Globus Connect Multi User

cd /opt

sudo wget http://connect.globusonline.org/linux/stable/globusconnect-multiuser-latest.tgz

sudo tar xzf globusconnect-multiuser-latest.tgz

cd gcmu*

sudo ./install

Try doing the following

Create a file called tutorial.txt in /home/joe

Go to the GO Web UI -> Start Transfer

Select endpoint *username*#gw12

Activate the endpoint as user “joe” (not gw12). You should see joe's home directory.
(Remember: joe's password is the same as gw12's)

Transfer to/from the endpoint of the person sitting next to you (activate their endpoint as user “sam”).

Does tutorial.txt show up in /home/sam in the host of the person sitting next to you?

Endpoint Configuration

What happens under the covers?

GlobusWORLD 2012

April 10, 2012

What happens under the covers?

GCMU automatically did the following three steps for us:

1. Provide our local users a way of authenticating themselves with GO and other endpoints.
2. Set up a GridFTP server that will be in charge of actually serving the files.
3. Create a GO endpoint.

Why would I need to do this manually?

- GCMU is very simple to use, but there is a lot you can tweak under the covers to:
  - Improve performance of the transfers
  - Use alternative authentication mechanisms
  - Use your own Certificate Authority
- Although you could install GCMU and then modify its configuration files, this is easier to do if you understand what's happening under the covers.
  - And, at that point, you might as well install the Globus Toolkit components from scratch, which is actually really simple (configuring them is the tricky part)

Step 1: User Authentication

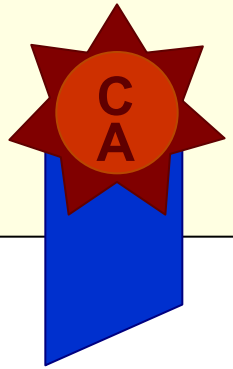
Problem: GO and other endpoints don't trust your users.

We need to give our users a “global identity” that they can use when accessing other endpoints.

This global identity takes the form of an X509 certificate.

Quick certificate refresher

I, *Certificate Authority FOO*, do hereby **certify** that *Borja Sotomayor* is who he/she claims to be and that his/her public key is *49E51A3EF1C*



Certificate Authority FOO

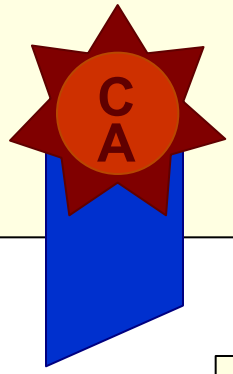
CA's Signature

Distinguished name

"O=Grid, OU=Globus Provision, CN=Borja Sotomayor"

Quick certificate refresher

I, *Certificate Authority FOO* , do hereby **certify** that *Borja Sotomayor* is who he/she claims to be and that his/her public key is *49E51A3EF1C*

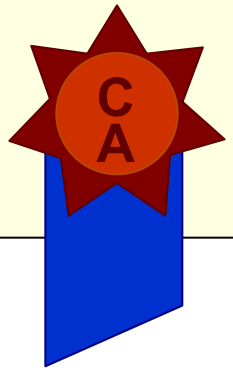


Certificate Authority FOO

CA's Signature

CA FOO signs my certficate

I, *Certificate Authority FOO* , do hereby **certify** that *CA FOO* is who he/she claims to be and that his/her public key is *7192BE61DCA*

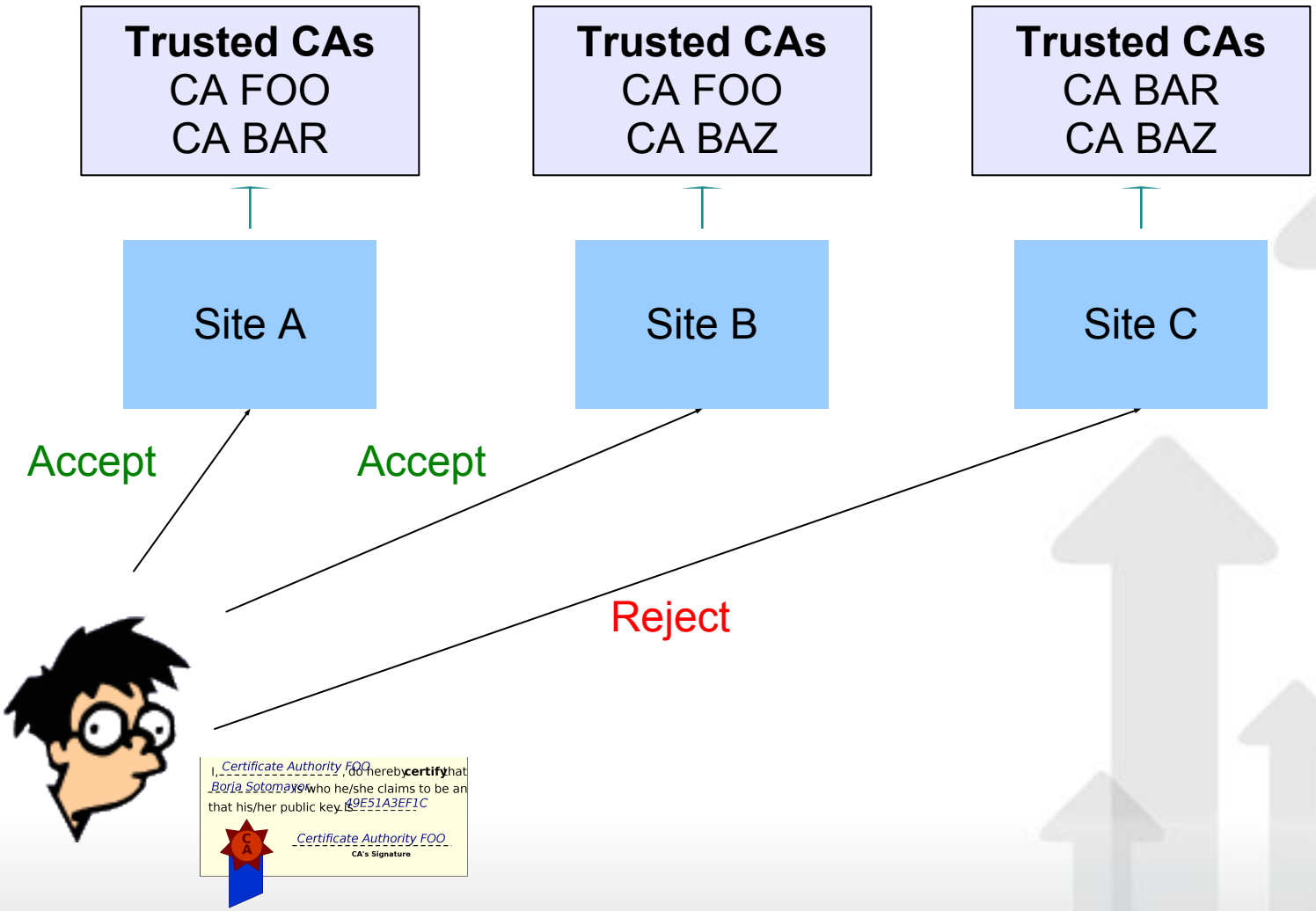


Certificate Authority FOO

CA's Signature

CA FOO signs its own certificate

Quick certificate refresher



Quick certificate refresher

If you want a more in-depth explanation, check out the *Security Primer* chapter of the *Resource Provider's Guide to Globus Online*

http://bit.ly/go-resource-provider-guide

Step 1: User Authentication

Solution: Give certificates to your users

Option #1: Go to a CA trusted by Globus Online, and request certificates from it.
Option #2: Set up your own CA, and ask Globus Online to trust it.

MyProxy CA will allow you to implement Option #2. It will also automatically generate certificates based on your local authentication domain.

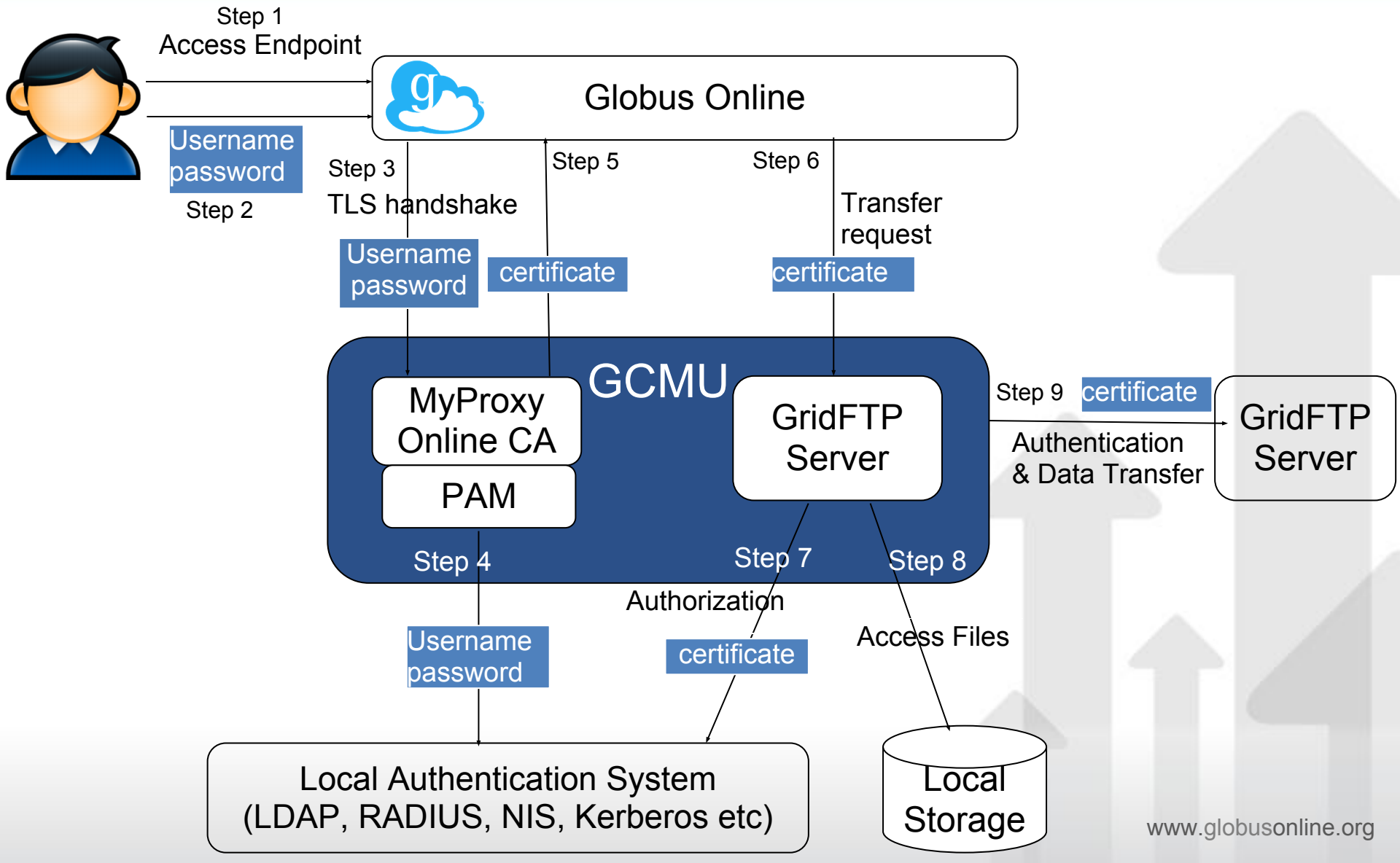
No need for your users to explicitly request a certificate, and keep track of it.

Step 2: GridFTP

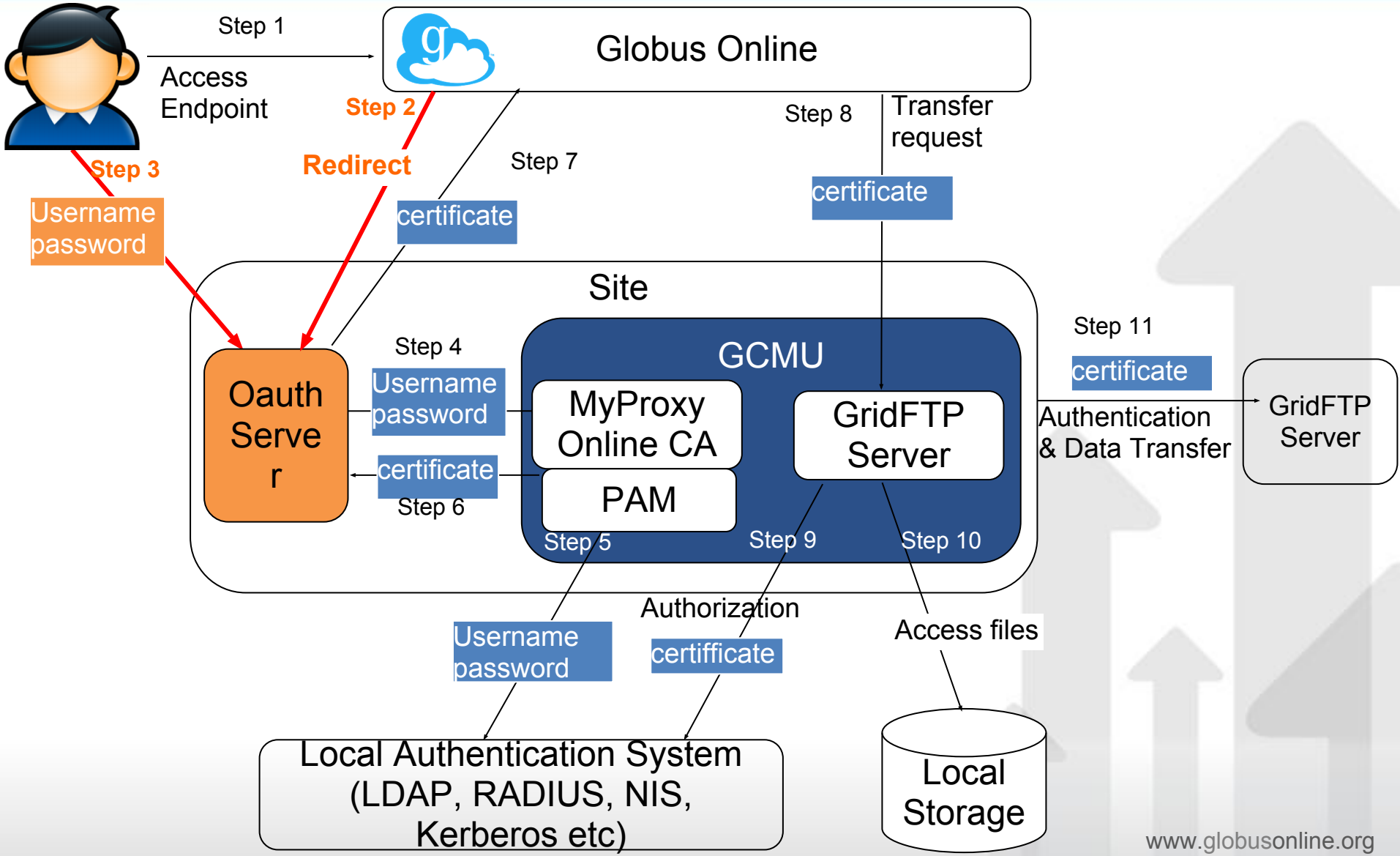
GridFTP is a high-performance and secure version of FTP.

In most cases, authentication is done with certificates, not with traditional user/pass.

What happens under the covers?



… and using MyProxy w/ OAuth



Details

- **GCMU may require firewall configuration**
  - Inbound ports: 2811, 7512, and 50,000-51,000
  - Outbound ports: 50,000-51,000
- **GCMU GridFTP cannot be (easily) used with other GridFTP clients besides GO**
  - GCMU uses a GO-issued cert, not a host cert
- **GridFTP has many configuration options**
  - E.g., Gridmap file can be used to allow other certs
  - Limit access to particular directories (soon)
  - You must re-apply these manually after an update
- **MyProxy CA default proxy lifetime is 12 hours**

Advanced Endpoint Configuration with Globus Toolkit

GlobusWORLD 2012

April 10, 2012

GlobusWORLD 2012 Admin Track

The goal of the Admin Track is to show you how to take an existing HPC resource and turn it into a GO endpoint.

In the previous tutorial, we saw how to do this using Globus Connect for Multiple Users.

If you did not attend the previous tutorial, please make yourself known, and we will get you set up so you can follow the examples.

If you attended the previous tutorial, log into your "Globus Toolkit" host.

Installing Globus Toolkit 5

As user “gw12” on your "Globus Toolkit" host:

wget http://www.globus.org/ftppub/gt5/5.2/5.2.0/installers/repo/globus-repository-natty_0.0.2_all.deb

sudo dpkg -i globus-repository-natty_0.0.2_all.deb

sudo apt-get update

sudo tasksel install globus-gridftp

sudo apt-get install myproxy myproxy-server

Step 1: User Authentication

We need to set up a CA and give certificates to our users.

However, we already have an authentication domain (UNIX accounts). Can we reuse that?

Yes! We can use MyProxy CA to generate certificates on-the-fly from our local accounts.

SimpleCA

MyProxy uses a simple CA (aptly named SimpleCA) that will generate a self-signed certificate.

We need to send the CA certificate to Globus Online so it will trust certificates signed by this CA.

For this tutorial, we've simply pre-installed a SimpleCA on all your hosts with a certificate that is trusted by Globus Online.

However, if you do need to set up SimpleCA, just run:

sudo grid-ca-create

MyProxy CA

Once SimpleCA is installed, installing MyProxy involves:

- Writing a MyProxy configuration file
- Adding a xinetd entry for MyProxy

MyProxy CA configuration file

```
certificate_issuer_cert /var/lib/globus/simple_ca/cacert.pem
certificate_issuer_key /var/lib/globus/simple_ca/private/cakey.pem
certificate_issuer_key_passphrase "cagrid"
certificate_issuer_subca_certfile /var/lib/globus/simple_ca/cacert.pem
certificate_serialfile /var/lib/globus/simple_ca/serial
certificate_out_dir /var/lib/globus/simple_ca/newcerts
pam  "sufficient"
certificate_mapapp /var/lib/myproxy/myproxy-certificate-mapapp
authorized_retrievers "*"
```

Mapping local accounts to DNs

```
#!/bin/sh
username=$1
if [ X"$username" = X ]; then
    # no username given
    exit 1
fi
echo "/O=Grid/OU=Globus Provision/CN=${username}"
exit 0
```

xinetd entry for MyProxy

```
service myproxy-server
{
 socket_type   = stream
 protocol      = tcp
 wait          = no
 user          = root
 server        = /usr/sbin/myproxy-server
 disable       = no
}
```

Add to /etc/services:

myproxy-server 7512/tcp

Create files and restart xinetd

sudo chown -R root.root /var/lib/myproxy/

sudo cp myproxy-certificate-mapapp /var/lib/myproxy

sudo cp myproxy-server.config /etc

sudo cp myproxy /etc/xinetd.d/

echo myproxy-server 7512/tcp | sudo tee -a /etc/services

sudo service xinetd reload

Give it a try!

As user joe, run this:

export MYPROXY_SERVER=`hostname --fqdn`

myproxy-logon

You should see:

A credential has been received for user joe in /tmp/x509up_u2002.

Give it a try!

More details about the certificate:

grid-proxy-info

You should see:

```
subject  : /O=Grid/OU=Globus Provision/CN=joe
issuer   : /O=Grid/OU=Globus Provision/CN=Globus Provision CA
identity : /O=Grid/OU=Globus Provision/CN=joe
type     : end entity credential
strength : 2048 bits
path     : /tmp/x509up_u2002
timeleft : 11:58:22
```

From zero to GO in three steps

Let's say we want to turn this into a GO endpoint. We will need to:

1. ~~Provide our local users a way of authenticating themselves with GO and other endpoints.~~
2. Set up a GridFTP server that will be in charge of actually serving the files.
3. Create a GO endpoint.

Step 2: GridFTP

Since GT5 is installed, starting a basic GridFTP server is actually very simple.

- But lots of knobs to turn if you want to customize it for maximum performance.
- http://fasterdata.es.net/fasterdata/host-tuning/

Configuring GridFTP involves:

- Writing a GridFTP configuration file
  Default file is actually enough (/etc/gridftp.conf)
- Adding a xinetd entry for GridFTP

xinetd entry for GridFTP

```
service gsiftp
{
        instances       = 50
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/globus-gridftp-server
        server_args     = -i
        log_on_success += DURATION
        nice            = 10
        disable         = no
}
```

No need to add anything to /etc/services.

Create files and restart xinetd

Run as "gw12":

sudo cp gsiftp /etc/xinetd.d/

sudo service xinetd reload

Not quite done yet...

If we try to do a simple transfer (as "joe"):

```
echo Hello > /tmp/tutorial.txt
globus-url-copy gsiftp://`hostname --fqdn`/tmp/tutorial.txt ./
```

We'll run into this:

```
error: globus_ftp_client: the server responded with an error
530 530-Login incorrect. : globus_gss_assist: Gridmap
lookup failure: Could not map /O=Grid/OU=Globus
Provision/CN=joe
530-
530 End.
```

Authorization

"joe" is not authorized to use the GridFTP server.

The default authorization method is the “gridmap” file, a simple file listing the DNs that are authorized to use a service, and what local user account they map to.

We need to add:

"/O=Grid/OU=Globus Provision/CN=joe" joe

To /etc/grid-security/grid-mapfile. Run this as "gw12":

echo \"/O=Grid/OU=Globus Provision/CN=joe\" joe | sudo tee -a /etc/grid-security/grid-mapfile

There are more sophisticated ways of doing this.

From zero to GO in three steps

Let's say we want to turn this into a GO endpoint. We will need to:

1. ~~Provide our local users a way of authenticating themselves with GO and other endpoints.~~
2. ~~Set up a GridFTP server that will be in charge of actually serving the files.~~
3. Create a GO endpoint.

Step 3: Creating the GO endpoint

We can create the endpoint using either the GO website or the GO CLI.

We will start with the GO CLI. All we need to do is:

- Use endpoint-add to create the endpoint.
- Use endpoint-modify to specify the MyProxy server and to make it a public endpoint.
- Use endpoint-activate to authenticate our GO user with the endpoint.

endpoint-add

endpoint-add
gw12tutorial
-p ec2-xx-xx-xx-xx.compute-1.amazonaws.com
-s "/O=Grid/OU=Globus Provision/CN=host/ec2-xx-xx-xx-xx.compute-1.amazonaws.com"

The name of the endpoint.

endpoint-add

```
endpoint-add
    gw12tutorial
    -p ec2-xx-xx-xx-xx.compute-1.amazonaws.com
    -s "/O=Grid/OU=Globus Provision/CN=host/ec2-xx-xx-xx-xx.
compute-1.amazonaws.com"
```

The hostname of the GridFTP server (host B)

endpoint-add

```
endpoint-add
    gw12tutorial
    -p ec2-xx-xx-xx-xx.compute-1.amazonaws.com
    -s "/O=Grid/OU=Globus Provision/CN=host/ec2-xx-xx-xx-xx.
compute-1.amazonaws.com"
```

The DN that GO should expect to be presented with when it contacts that GridFTP server.

endpoint-modify

endpoint-modify --myproxy-server=ec2-xx-xx-xx-xx.compute-1.amazonaws.com gw12tutorial

endpoint-modify --myproxy-dn="/O=Grid/OU=Globus Provision/CN=host/ec2-xx-xx-xx-xx.compute-1.amazonaws.com" gw12tutorial

endpoint-modify --public gw12tutorial

Specify what MyProxy server will be used to obtain a proxy certificate to authenticate with the GridFTP server.

endpoint-modify

endpoint-modify --myproxy-server=ec2-xx-xx-xx-xx.compute-1.amazonaws.com gw12tutorial

endpoint-modify --myproxy-dn="/O=Grid/OU=Globus Provision/CN=host/ec2-xx-xx-xx-xx.compute-1.amazonaws.com" gw12tutorial

endpoint-modify --public gw12tutorial

Specify the DN of the MyProxy server.

endpoint-modify

endpoint-modify --myproxy-server=ec2-xx-xx-xx-xx.compute-1.amazonaws.com gw12tutorial

endpoint-modify --myproxy-dn="/O=Grid/OU=Globus Provision/CN=host/ec2-xx-xx-xx-xx.compute-1.amazonaws.com" gw12tutorial

endpoint-modify --public gw12tutorial

Make the endpoint public.

The public name will be "*username*#gw12tutorial"

endpoint-activate

endpoint-activate -U joe gw12tutorial

Activate the endpoint as user “joe”.

Give it a try!

You can now transfer files to/from the *username*#gw12tutorial endpoint!

From zero to GO in three steps

Let's say we want to turn this into a GO endpoint. We will need to:

1. ~~Provide our local users a way of authenticating themselves with GO and other endpoints.~~
2. ~~Set up a GridFTP server that will be in charge of actually serving the files.~~
3. ~~Create a GO endpoint.~~