

# Leiden Grid Infrastructure

## the future of computational chemistry

Hugo Meiland (LIC-ICT)  
Mark Somers (LIC-THEOR)

## **Battle plan of this presentation:**

- Some awarenesses.
- The true problems and challenges developers will be facing.
- Are computer GRIDs a possible solution?
- Grid middle ware.
- The 3 layer structure of LGI.
- Communication within LGI:
  - XML
  - HTTPS
- Where are we now.
- What's next?

## Some awarenesses

- Research focus turns to results rather than methodology.
- Computers are more and more used as disposable objects.
- (PhD) Students are encouraged to spend more time on computing/research.
- The number of programs used is relatively low.

In contrast to:

- Computers are becoming more and more complex and diverse.
- Theories and algorithms are getting more advanced.
- Installing most of the (chemistry) software is not for the fainthearted.
- Using computers efficiently becomes harder to do.
- Doing both chemistry and computer science is no longer possible.

## Some awarenesses

- Research focus turns to results rather than methodology.
- Computers are more and more used as disposable objects.
- (PhD) Students are encouraged to spend more time on computing/research.
- The number of programs used is relatively low.

**More users...**

In contrast to:

- Computers are becoming more and more complex and diverse.
- Theories and algorithms are getting more advanced.
- Installing most of the (chemistry) software is not for the fainthearted.
- Using computers efficiently becomes harder to do.
- Doing both chemistry and computer science is no longer possible.

**Less developers !**

## The problems developers will be facing

The program serves a **purpose** and **users** determine program's existence:

- Need to make programs more **intuitively** and **accessible** for the most common tasks performed by the program so that they are easier to use.
- Need to disentangle program interface from hardware details so user needn't worry about queues, nodes, security and all the complexity of setting up the program on each type of hardware.
- Need to provide **scalability** and **continuity** for the programs to support more users in future and to match the demand for more hardware again *without interfering the users.*

For example...

The usual way of starting a calculation...

```
Shell - Konsole
Session Edit View Bookmarks Settings Help

mark::/home/mark>ssh -X mark@woodcrest.wks.gorlaeus.net
mark@woodcrest.wks.gorlaeus.net's password:
Last login: Fri Jun 22 10:04:22 2007 from fwnc7003.leidenuniv.nl
mark::/home/mark>cd SA-DVR_2.03.dir/INPUTFILES.dir/
mark::/home/mark/SA-DVR_2.03.dir/INPUTFILES.dir>ls
PES4_H2ONCU100.dir  v0j4m0p5_01.inp  v0j4m2p5_01.inp  v0j4m4p5_01.inp
PES5_H2ONCU100.dir  v0j4m1p4_01.inp  v0j4m3p4_01.inp  v1j0m0p5_highE_01.inp
v0j0m0p5_01.inp    v0j4m1p5_01.inp  v0j4m3p5_01.inp  v1j0m0p5_lowE_01.inp
v0j4m0p4_01.inp    v0j4m2p4_01.inp  v0j4m4p4_01.inp
mark::/home/mark/SA-DVR_2.03.dir/INPUTFILES.dir>vi v0j0m0p5_01.inp
mark::/home/mark/SA-DVR_2.03.dir/INPUTFILES.dir>cd ..
mark::/home/mark/SA-DVR_2.03.dir>cd SCRIPTS.dir/
mark::/home/mark/SA-DVR_2.03.dir/SCRIPTS.dir>ls
BACKUP_GOOD_JOB_FILE_FOR_ALTIX_MPI  BACKUP_GOOD_JOB_FILE_FOR_WOODCREST_MPI
BACKUP_GOOD_JOB_FILE_FOR_ALTIX_OMP  BACKUP_GOOD_JOB_FILE_FOR_WOODCREST_OMP
BACKUP_GOOD_JOB_FILE_FOR_LISA_MPI    DoBuild
BACKUP_GOOD_JOB_FILE_FOR_NOCONA_MPI  DoClean
BACKUP_GOOD_JOB_FILE_FOR_NOCONA_OMP  DoMake
BACKUP_GOOD_JOB_FILE_FOR_SGI_MPI      PreProcess
BACKUP_GOOD_JOB_FILE_FOR_SGI_OMP      v0j0m0p5_01.job
BACKUP_GOOD_JOB_FILE_FOR_SP2_MPI      v0j0m0p5_01.job_mpi
BACKUP_GOOD_JOB_FILE_FOR_SP2_OMP
mark::/home/mark/SA-DVR_2.03.dir/SCRIPTS.dir>vi v0j0m0p5_01.job_mpi
mark::/home/mark/SA-DVR_2.03.dir/SCRIPTS.dir>qsub v0j0m0p5_01.job_mpi
8492.master.woodcrest.gorlaeus.net
mark::/home/mark/SA-DVR_2.03.dir/SCRIPTS.dir>qstat
Job id      Name                User                Time Use S Queue
-----
7994.master  aneb.ti_a1          chen                03:35:25 R long
8184.master  aneb.ti_a1          chen                03:05:51 R long
8198.master  ....pure_a1.test    chen                04:22:47 R long
8306.master  W03.bulk.4k         alvaro              107:52:4 R long
8312.master  trabajoa10          gustavo              71:19:15 R long
8313.master  trabajoa11          gustavo              70:58:22 R long
8333.master  trabajoa12          gustavo              68:03:08 R long
8334.master  trabajoa13          gustavo              67:12:41 R long
8335.master  trabajoa14          gustavo              66:43:58 R long
8336.master  trabajoa15          gustavo              66:35:24 R long
8386.master  W03.bulk.a.4k       alvaro              60:49:00 R long
8399.master  trabajoa16          gustavo              44:58:38 R long
8403.master  trabajoa17          gustavo              42:34:00 R long
8443.master  Ru0001-slab         irene               18:21:01 R long
8444.master  Ru0001-slab         irene               18:15:50 R long
8445.master  Ru0001-slab         irene               17:36:30 R long
8447.master  Ru0001-slab         irene               20:41:16 R long
8448.master  Ru0001-slab         irene               20:38:26 R long
8449.master  Ru0001-slab         irene               11:18:46 R long
8450.master  Ru0001-slab         irene               11:36:22 R long
8452.master  TiO2.OOH.5          alvaro              08:35:45 R long
8459.master  Ru0001-slab         irene               28:08:15 R long
8460.master  Ru0001-slab         irene               14:30:49 R long
8461.master  Ru0001-slab         irene               19:58:21 R long
```

- 1) Start with logging into cluster
- 2) Setup program
- 3) Edit input file
- 4) Edit PBS script file
- 5) Submit calculation
- 6) Regularly check job

**Only input file is independent of cluster and the hardware used!**

# For example... wouldn't it be really cool if...

New job for m.somers at 22 Mar 2007 9:16:47 UTC

**New job specifics**

Application: **Classical**

Name of job (no spaces, quotes or slashes): **test\_sommeke**

Input:

```
interaction bending f=0.175 deg=111.2 ( 3 4 400 )
interaction bending f=0.175 deg=111.2 ( 3 4 401 )
interaction bending f=0.175 deg=111.2 ( 3 4 402 )

# the bonding interactions of the first CH3
interaction harmonic f=1.195 r0=3.76 ( 4 400 )
interaction harmonic f=1.195 r0=3.76 ( 4 401 )
interaction harmonic f=1.195 r0=3.76 ( 4 402 )

# -----
# find stable situation first
conformation n=10000 error=1.0E-4 maxstep=2.5

# -----
# then setup a constant temperature
temperature k=3.166829379841521e-06 constant

# and finally the dynamics statement, run 100
dynamics dt=4.13411 tend=4134110.5461 t=0.0
```

Estimated time to completion: **3 h.**

Estimated memory usage: **4 Mb.**

Estimated disk usage: **4 Mb.**

Submit Job

**Commands**

[Run Classical-Builder](#)

[Go back to your queue](#)

[Log out](#)

... you could do that with:

- ADF
- Gaussian
- VASP
- ...

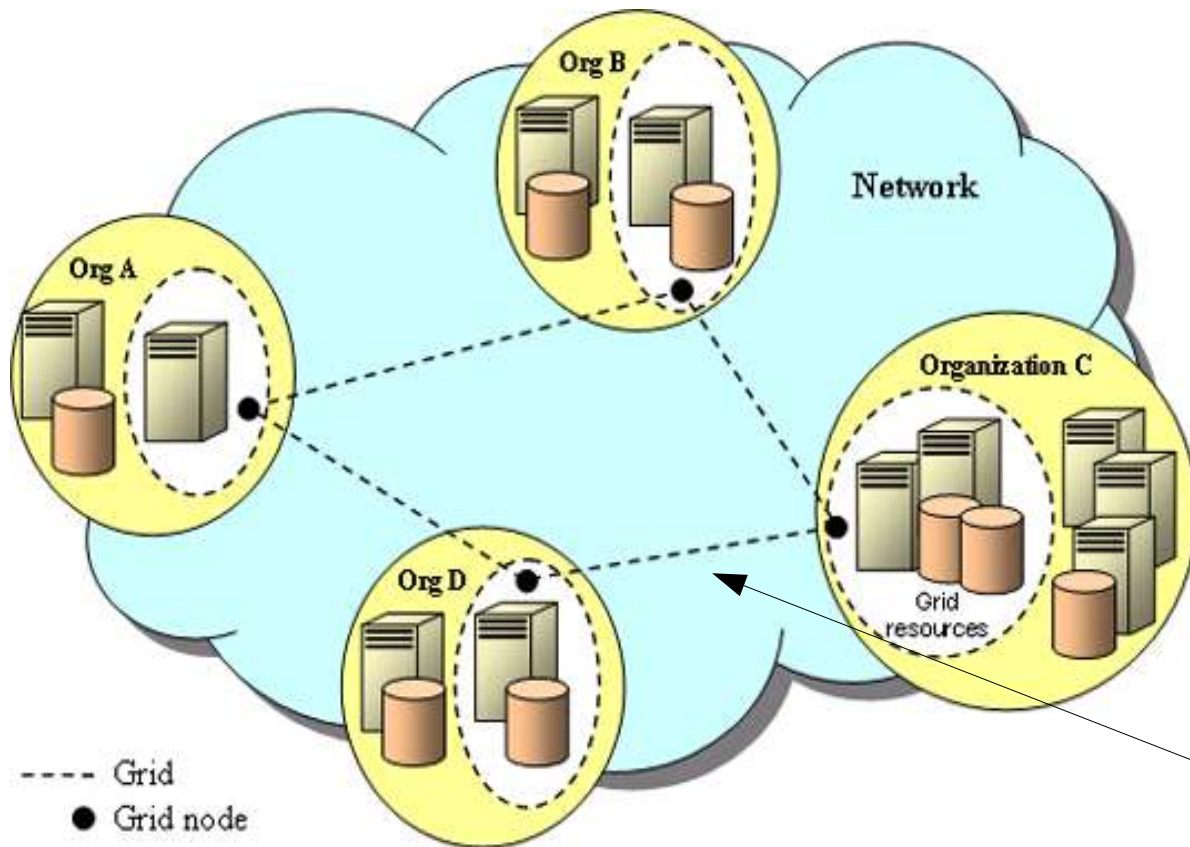
Personal job list of m.somers at 22 Mar 2007 9:08:45 UTC

You have 6 jobs listed !				
Job #	Job submit time	Job status	Job name	Job ID
1	16 Mar 2006 12:22:58 UTC	finished	<a href="#">test_trajtou_pd110paw</a>	<a href="#">88811</a>
2	16 Mar 2006 12:22:01 UTC	finished	<a href="#">test_trajtou_pt111</a>	<a href="#">88804</a>
3	23 Oct 2006 13:05:57 UTC	finished	<a href="#">test_ABC</a>	<a href="#">1534582</a>
4	20 Feb 2007 7:27:46 UTC	finished	<a href="#">test_butaan</a>	<a href="#">1854209</a>
5	9 Feb 2007 14:21:34 UTC	finished	<a href="#">test_pd_again</a>	<a href="#">1824217</a>
6	26 Feb 2007 13:23:15 UTC	finished	<a href="#">test_pd_again</a>	<a href="#">1868383</a>
Commands				
You can submit 84 more jobs: <a href="#">Submit a job</a>				
<a href="#">Your account</a>				
<a href="#">Your computers</a>				
<a href="#">Log out</a>				

[Return to Leiden Classical main page](#)

# Are computer grids the answer?

A GRID is a collection of different types of computers, all connected to each other, that are owned and administrated by several people, that are capable of performing tasks and can be regarded as a whole.



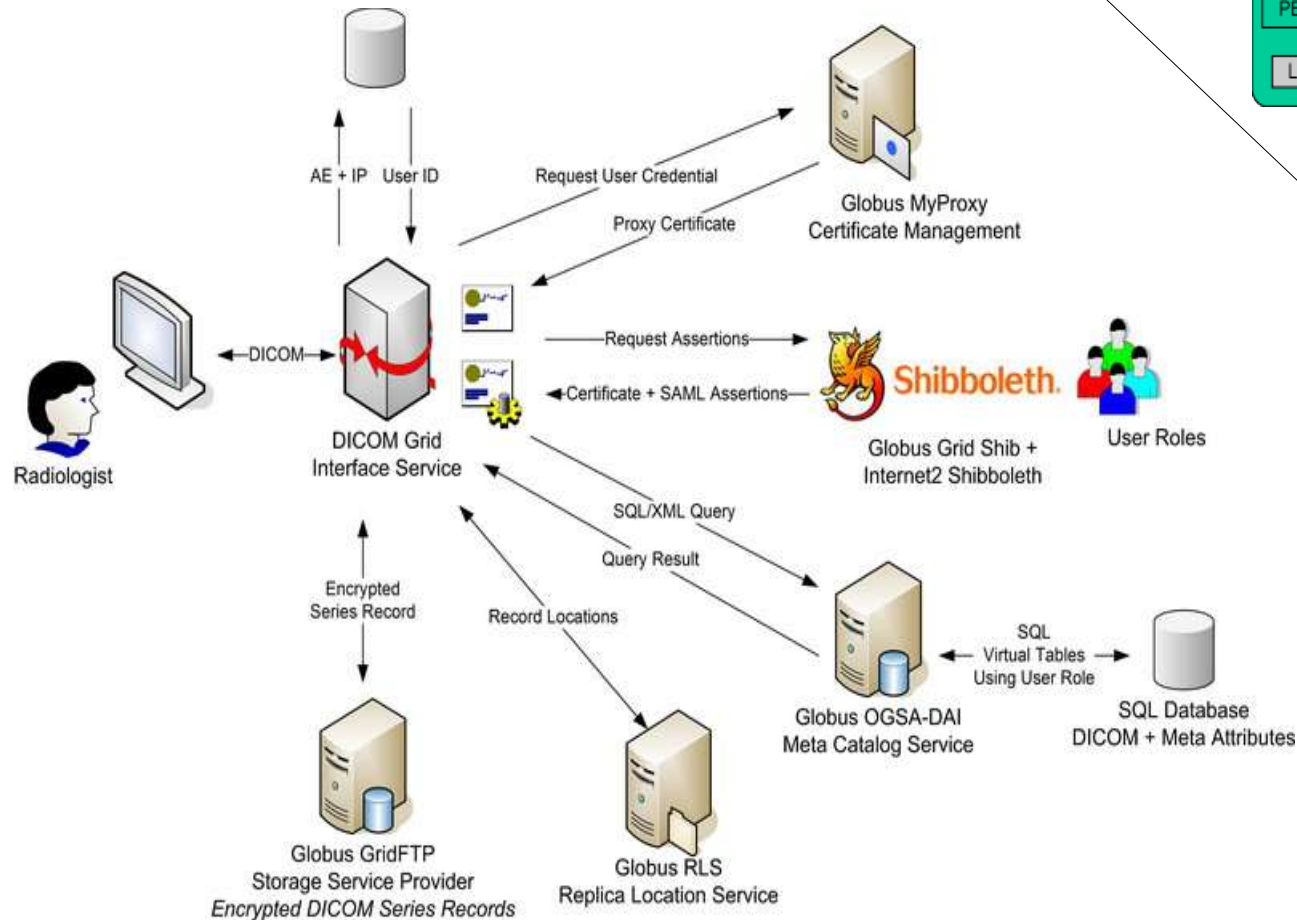
Grids do allow for **scalability** and **continuity** but need “**middle ware**”...



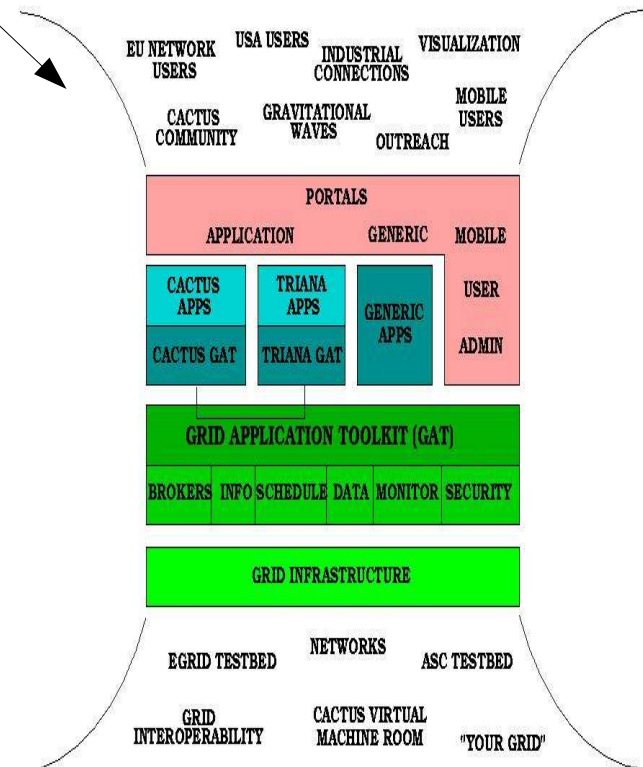
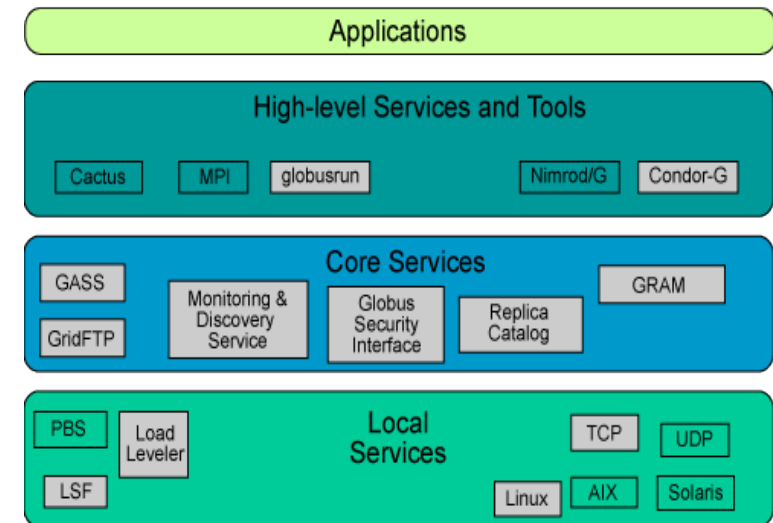
# Grid middle warez...

like GLOBUS and GAT...

allow you to do setup something like this...



## Globus Architecture



Grid middle warez...

like GLOBUS and GAT...

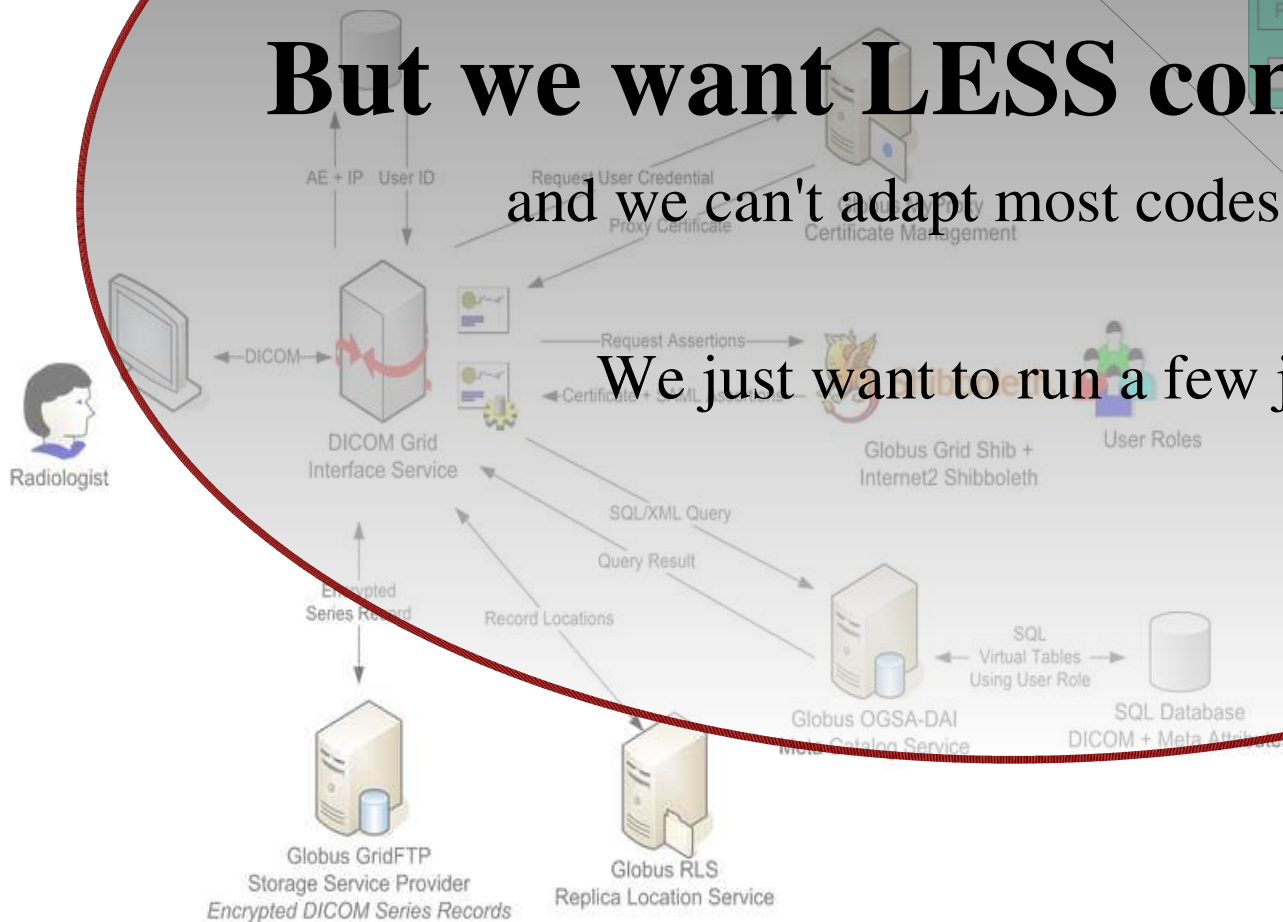
allow you to do setup something like this...

**But we want LESS complexity ?!**

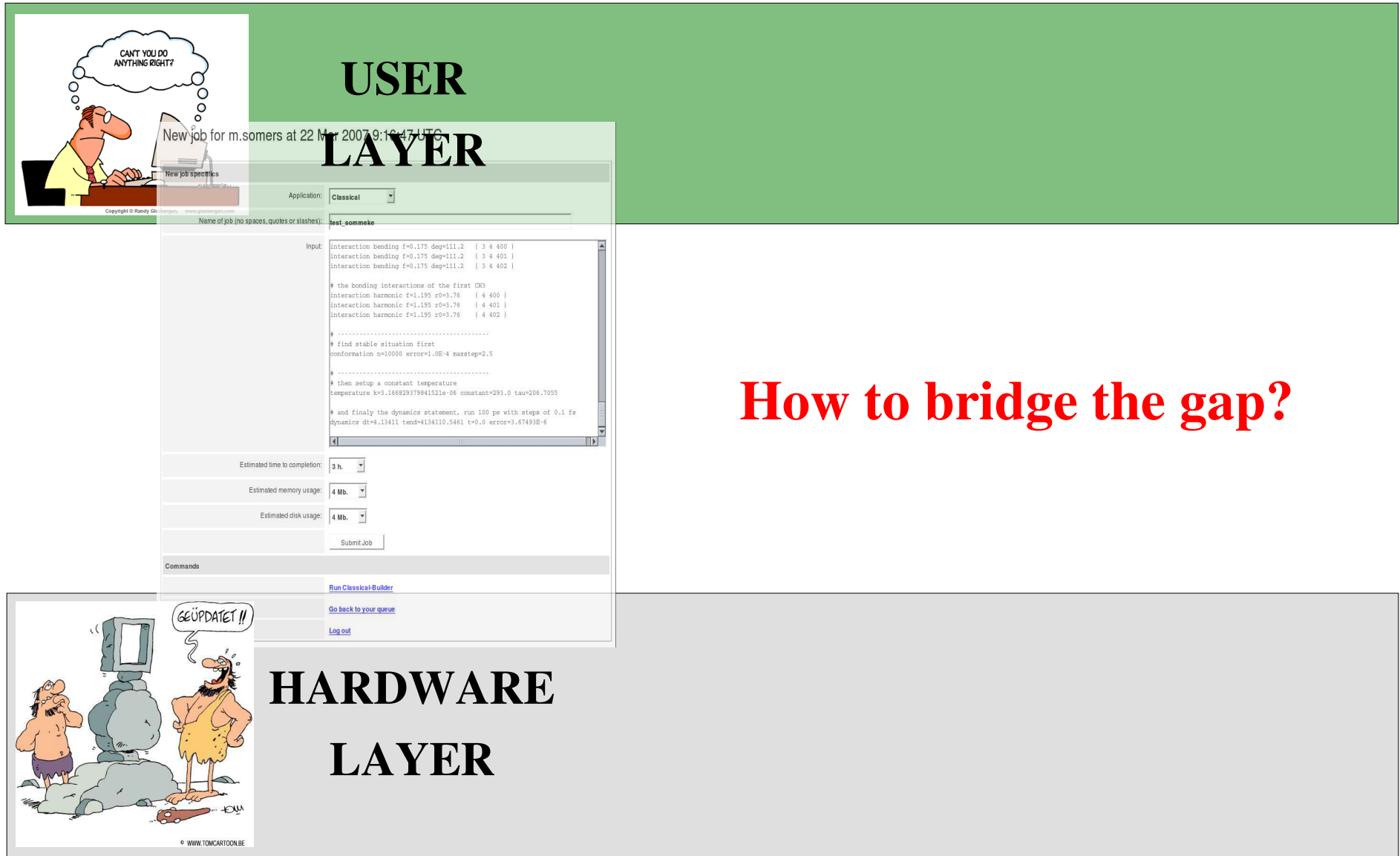
and we can't adapt most codes we use...

**We just want to run a few jobs...**

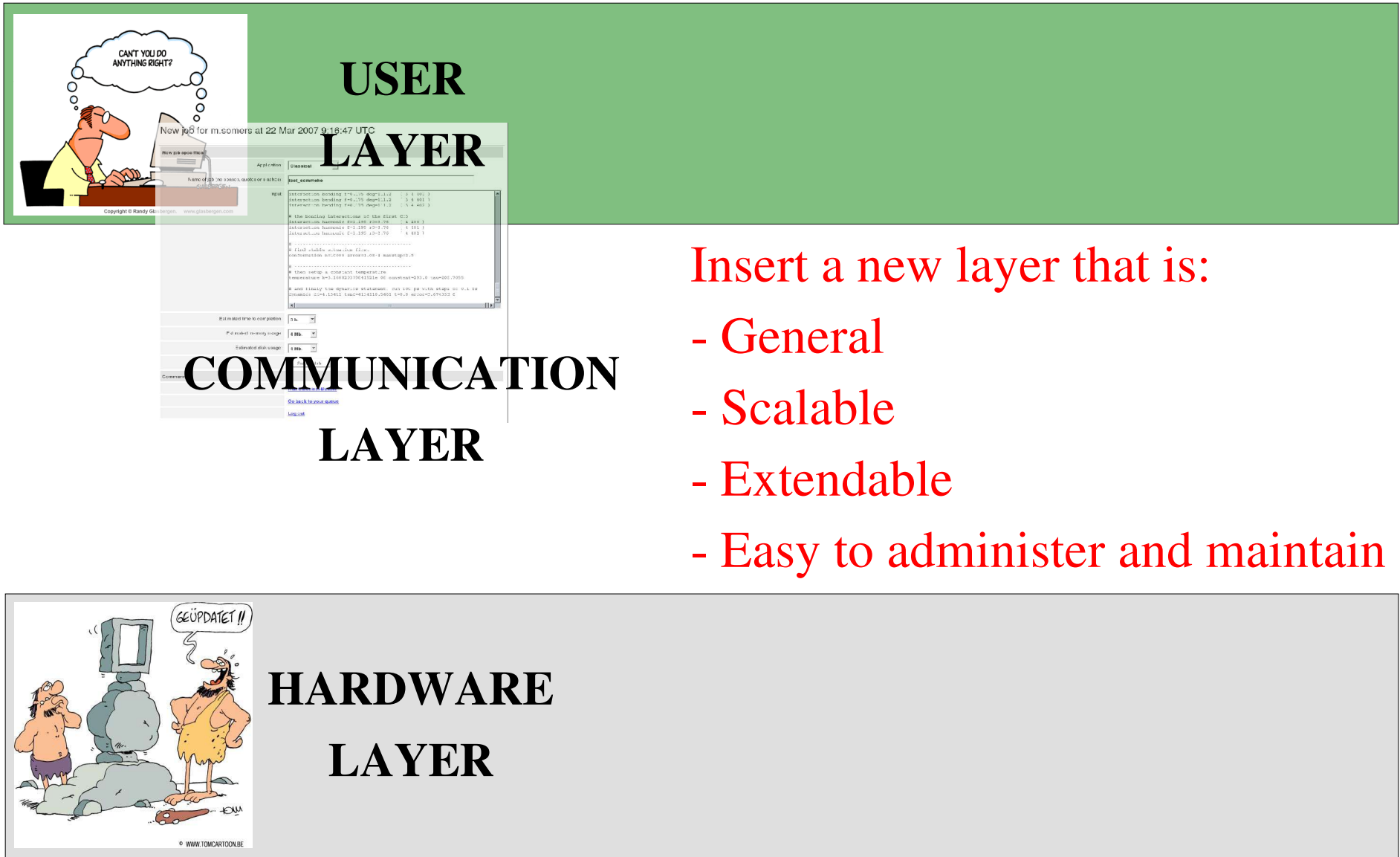
## Globus Architecture



# Enter LGI... The (lightweight) grid middle ware we need...



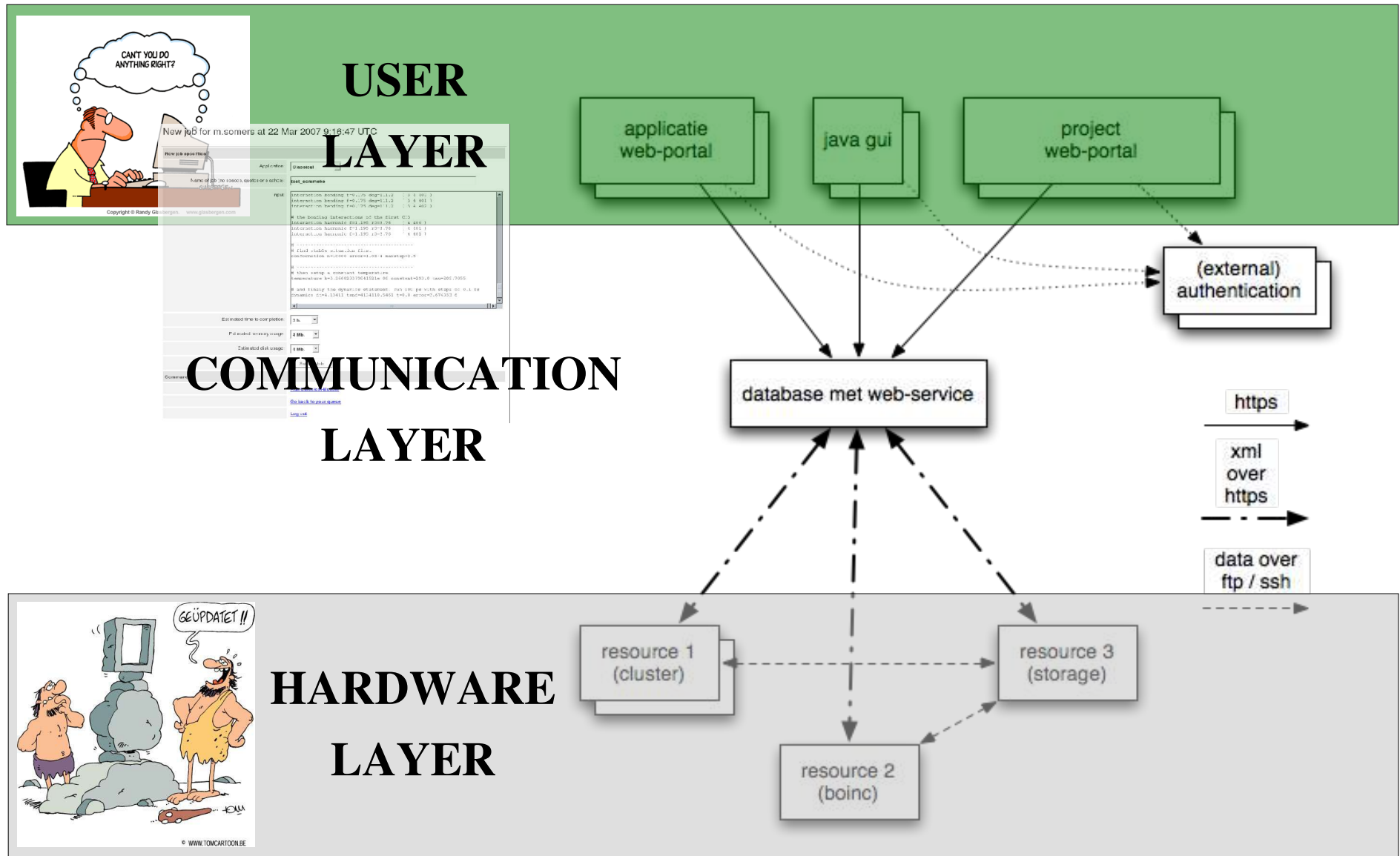
# Enter LGI... The (lightweight) grid middle ware we need...



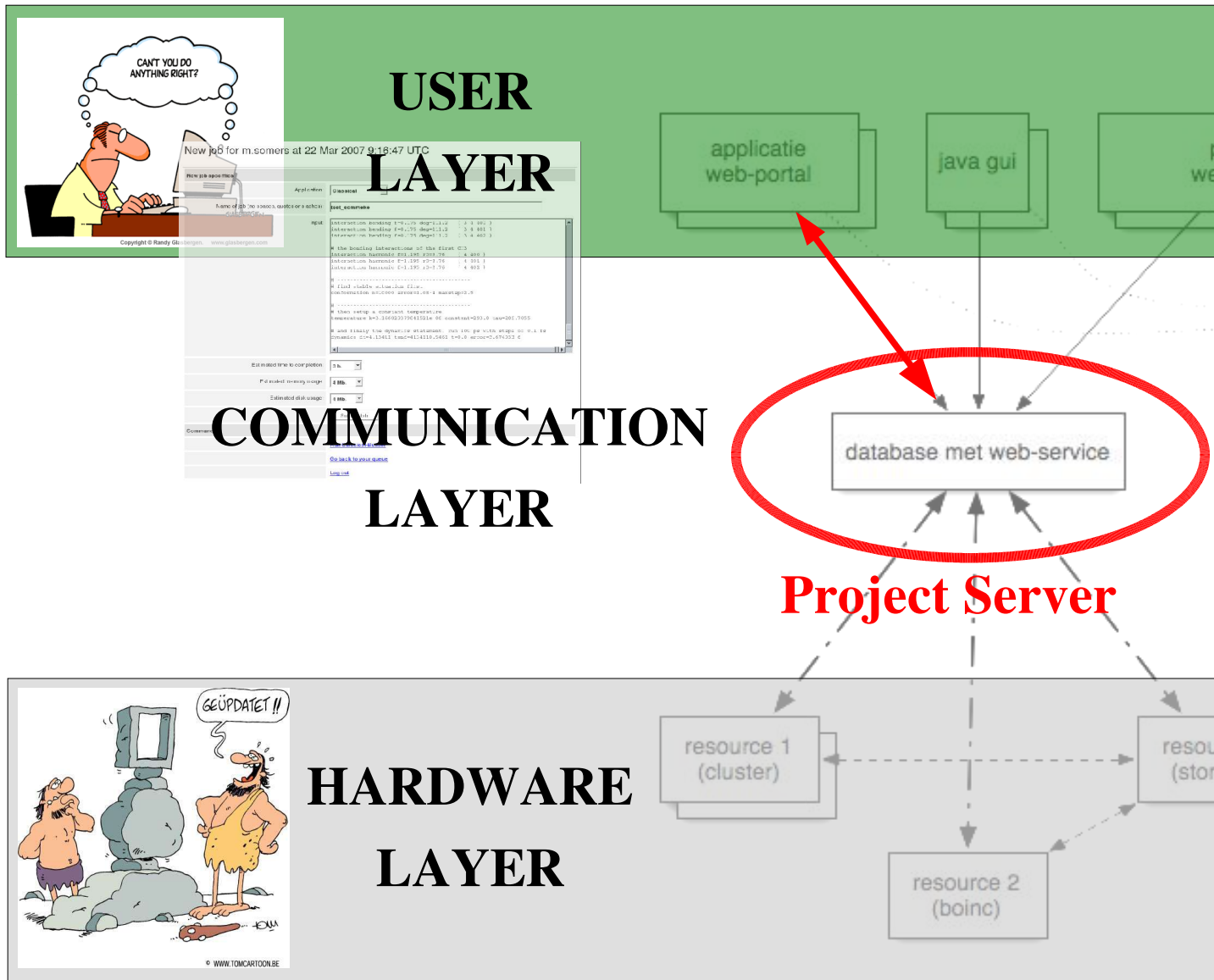
Insert a new layer that is:

- General
- Scalable
- Extendable
- Easy to administer and maintain

# Enter LGI... The (lightweight) grid middle ware we need...



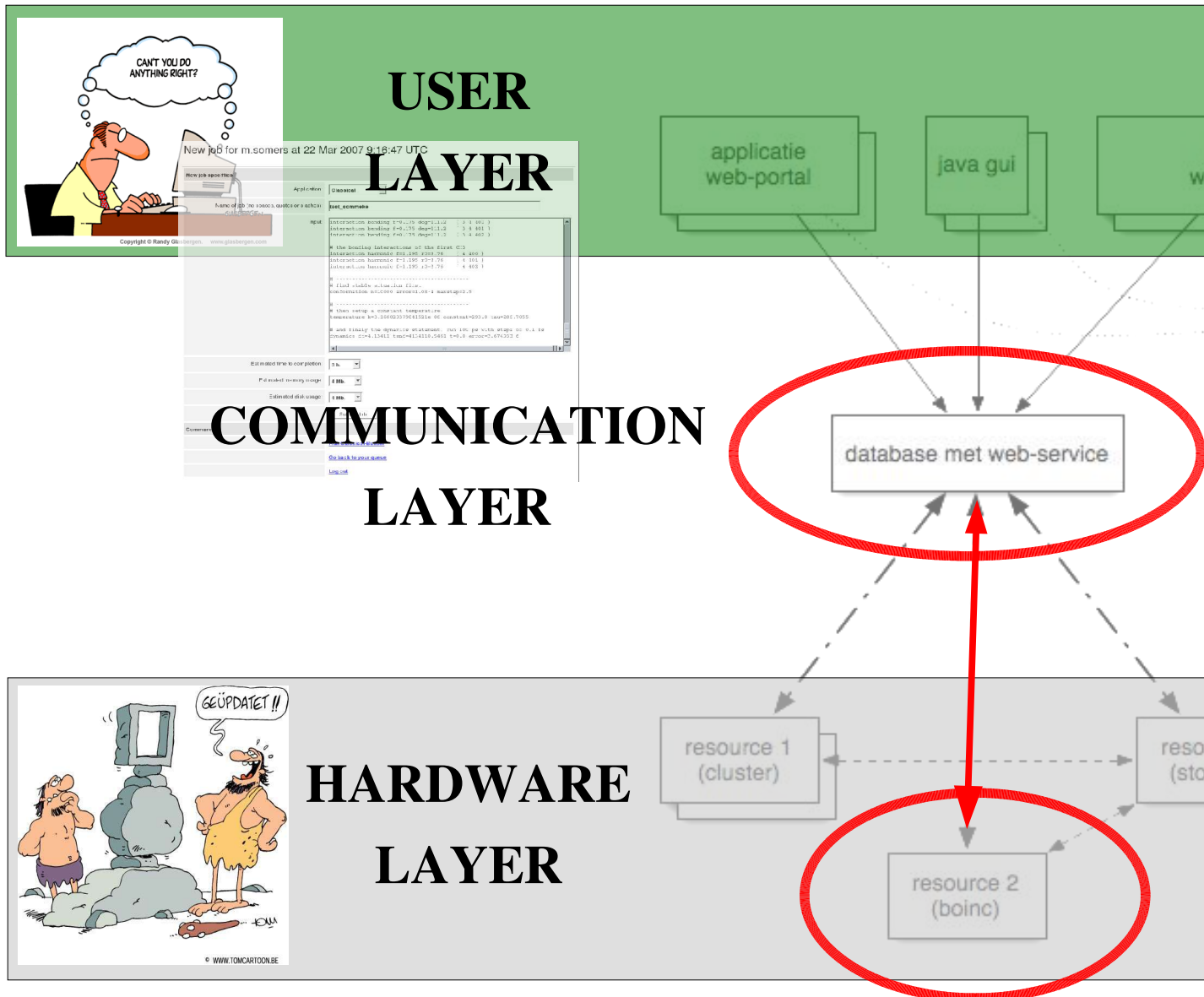
# Enter LGI... The (lightweight) grid middle ware we need...



Users submit jobs to database on project server and check for results by using a browser...



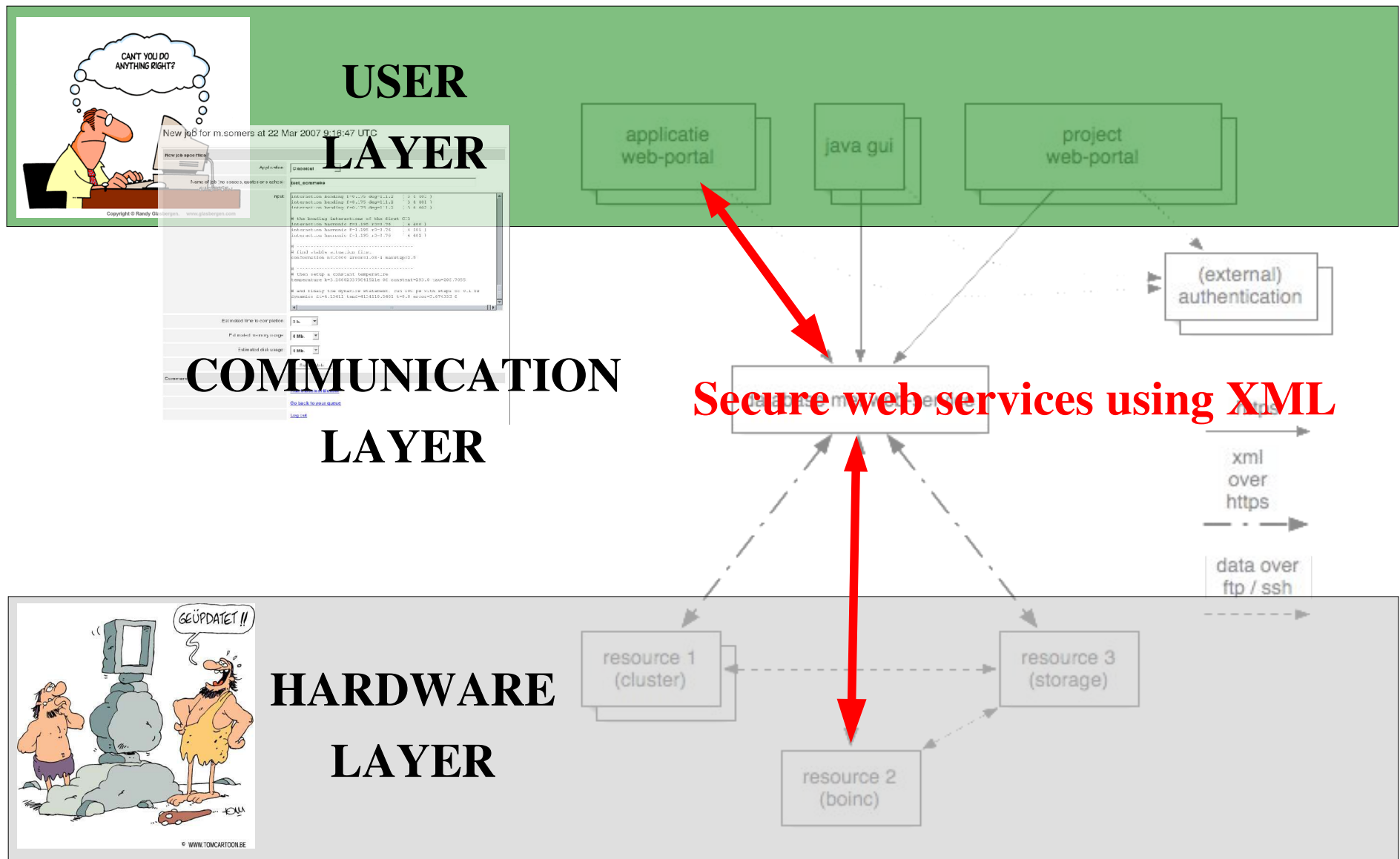
# Enter LGI... The (lightweight) grid middle ware we need...



Resources  
request work  
they can handle  
from database on  
project server  
and report back  
the results...

**Resource Daemon**

# Communication within the LGI...





# Communication within the LGI...

What is eXtensible Markup Language?

A format that allows to markup or encapsulate data suitable for transferring and extending...

It uses tags:

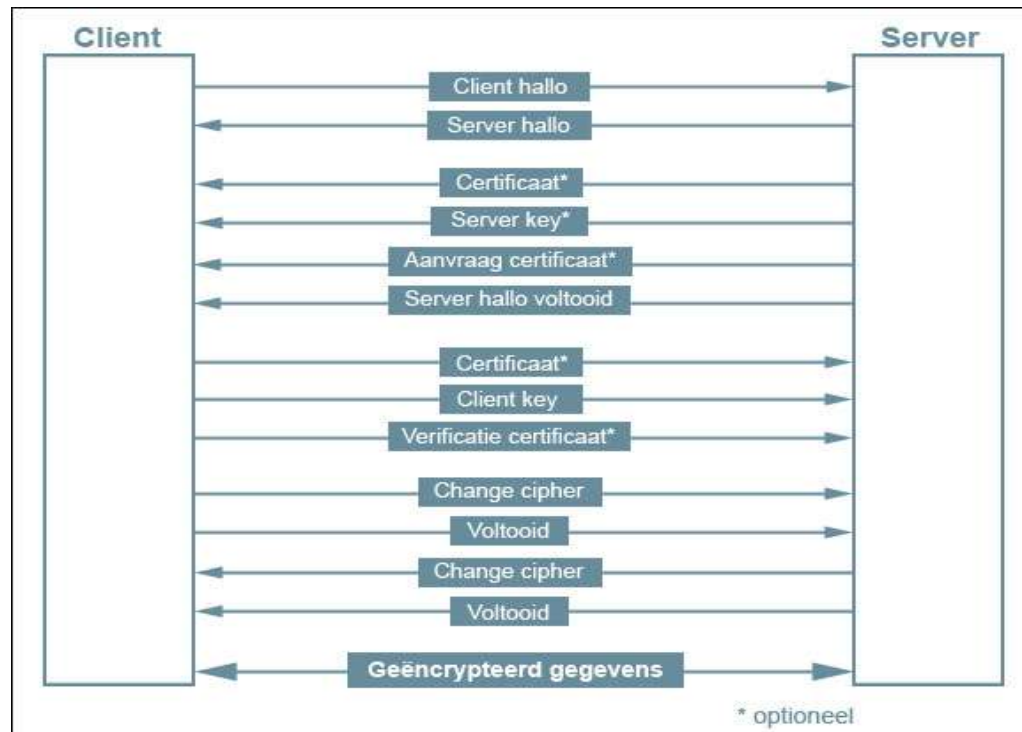
```
<LGI>
  <CA_certificate> http://www.LGI.org/LGI-CA.crt </CA_certificate>
  <response>
    <project> testproject </project>
    <project_master_sever> https://fwnc7003.leidenuniv.nl/testmasterserver </project_master_server>
    <project_sever> https://fwnc7003.leidenuniv.nl/testslaveserver </project_server>
    <user> mark </user>
    <groups> teras, cytttron </groups>
    <job>
      <job_id> 147 </job_id>
      <application> testapp </application>
      <state> queued </state>
      <target_resources> any </target_resources>
      <owners> sjoerd, cytttron </owners>
      <read_access> any, sjoerd, cytttron </read_access>
      <state_time_stamp> 1259936661 </state_time_stamp>
      <job_specifics> </job_specifics>
      <input> CDEF9021569C8787E </input>
    </job>
  </response>
</LGI>
```

# Communication within the LGI...

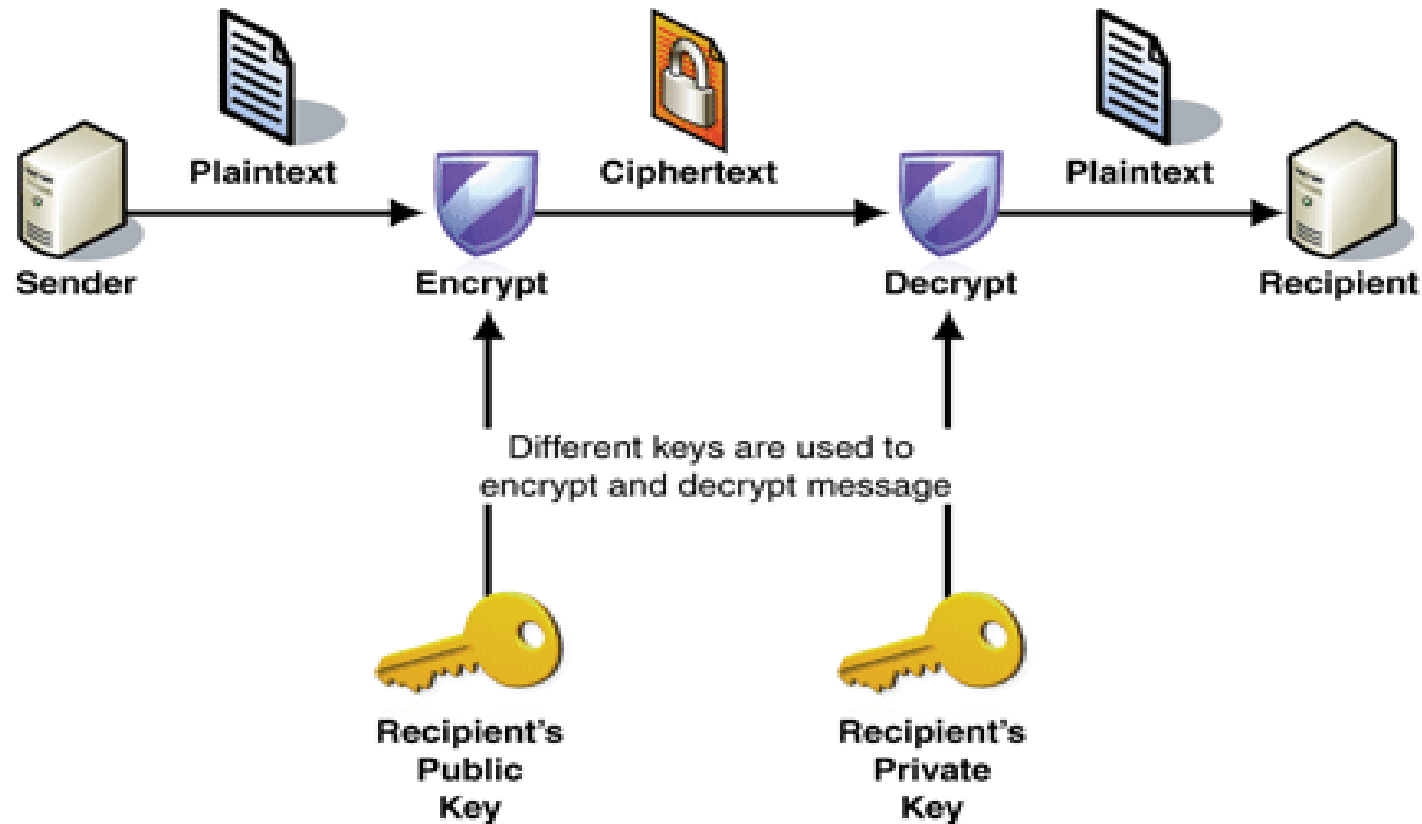
What is **H**yper **T**ext **T**ransport **P**rotocol **S**ecure?

A protocol to transfer data between computers securely using **Secure Socket Layer** technology...

It uses **certificates** and **asymmetric** encryption:



## Asymmetric encryption (RSA)...

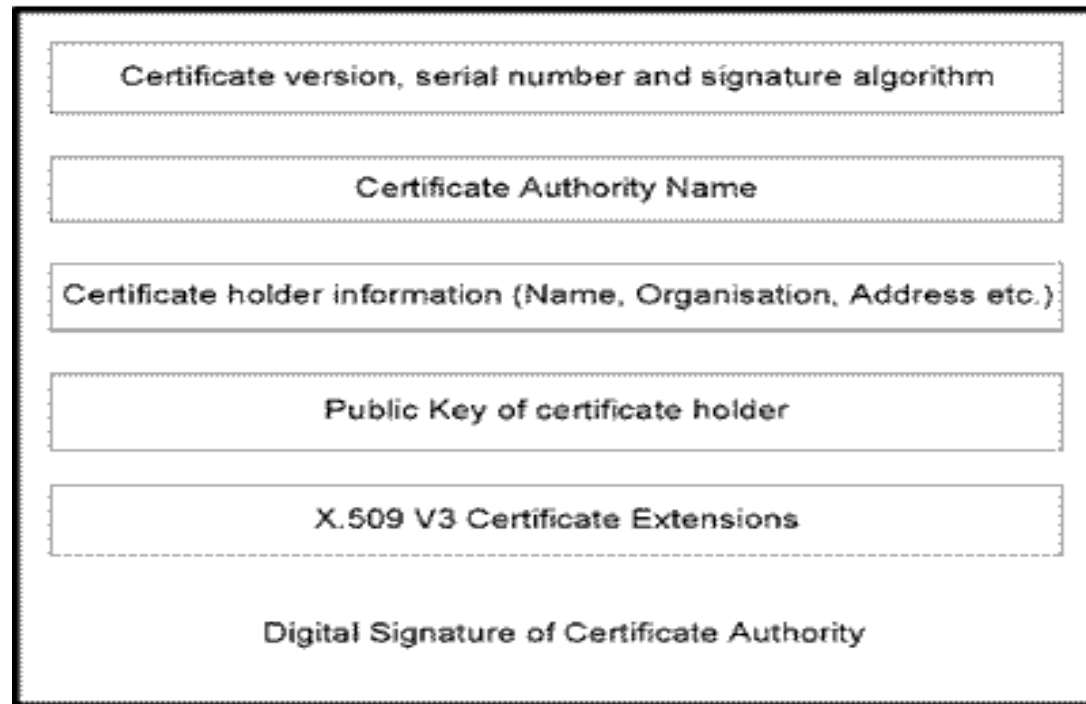


**If we are sure that we use the recipients public key, the data can only be read by that recipient!**

**Assurance is guaranteed by using certificates!**

## x509 Certificates...

**Piece of data that stores the public key and the identity of a recipient:**



**Certificate has been 'signed' by a trusted third party!**

**'Signing' by encrypting the fingerprint with the private key of trusted party.**

## Where are we now...

Project server software and communication has been implemented using:

- Apache web server using SSL (https) with client- and server-certificates.
- (PHP) API for 'Interface' - 'Project server' and 'Project server' – 'Resource' communication using XML:

*interface\_submit\_job*  
*interface\_job\_state*  
*interface\_delete\_job*  
*interface\_project\_server\_list*

*resource\_signup\_resource*  
*resource\_request\_work*  
*resource\_job\_details*  
*resource\_lock\_job*  
*resource\_unlock\_job*  
*resource\_update\_job*  
*resource\_submit\_job*  
*resource\_job\_state*

- MySQL database as back-end on 'Project server' with user management tables.
- Two example 'Project servers' have been setup:

<https://lgi.cyttron.gorlaeus.net>  
<https://fwnc7003.leidenuniv.nl/LGI>

## **What's next?**

- 1) Start with implementing the 'Resource Daemon'.
- 2) Implementing the secure 'file transfer programs'.
- 3) Implementing a very basic and general program interface.
- 4) Testing phase.

**Thank you for your attention... any further questions?**

**<http://fwnc7003.leidenuniv.nl/LGI/docs/LGI.pdf>**