

XWHEP 5.7.5 :

XTREMWEB

BY

HIGH ENERGY PHYSICS



# XWHEP

- Introduction
- Architecture
- Rights
- Objects management
- Compilation, installation
- Coordinator service
- Worker service
- Client service
- Benchmark
- Pilot Jobs
- Perspective



# INTRODUCTION



XWHEP is developed by IN2P3.

It is based on XtremWeb 1.8.0. by INRIA.



# INTRODUCTION

XWHEP is a generic multi purposes desktop grid platform (*DG*) enabling eSciences computations over volatile nodes.

Main features are :

- three tiers architecture
- multi platforms (win32, linux, mac os x)
- virtual stable cluster over volatile volunteers individual PCs
- multi applications
- multi users
- firewall bypassing
- automatic load balancing
- fault tolerance



# GOALS

XWHEP main goals:

- full production platform
- inter grids connexions (especially focusing on EGEE).

To achieve this goal, XWHEP proposes a secured DG:

- certified server;
- X509 user proxy usage;
- access rights;
- usage levels including two major ones : “public” and “private”:
  - ➔ “public”, intrinsically secured, enabling inter grid sharings;
  - ➔ “private”, intrinsically secured.

# XWHEP VS XTREMWEB

1/2

	XWHEP	XtremWeb 1.8
Inter-grids connexions	+	-
User rights	++	+
Data management	+	-
Access rights	+	-
Multi transport protocols	UDP, TCP	
Multi communication layers	XW, HTTP	-
User application management	+	<b>admin only</b>
User worker management	+	-
SSL / certificates	+	-
Proxy	+	-
ACL	+	-

enabling inter grid sharings
implemented & tested
not fully implemented



# XWHEP VS XTREMWEB

2/2

	XWHEP	XtremWeb 1.8
Dynamically linked applications	+	-
Avg. ping	+	-
Avg. bandwidth usage	+	-
Custom scheduler	+	-
Worker launcher	+	-
Input files / job	+	+
Input files / app	+	-
Match making	OS, CPU, RAM, DISK	OS, CPU
CPU/RAM requirements	+	+
CPU/RAM requirements	+	-

implemented & tested
not fully tested
not fully implemented

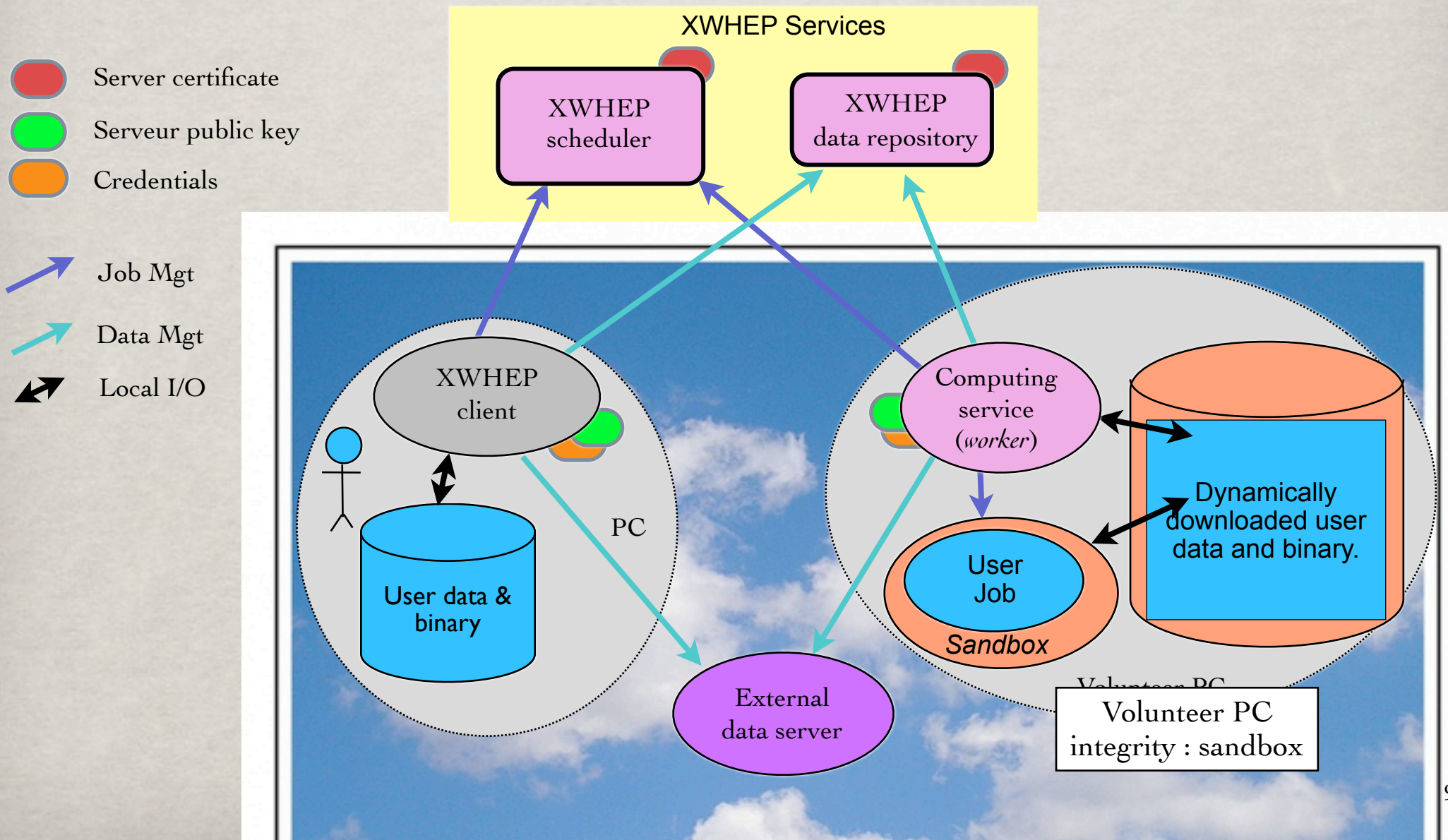
# XWHEP

- Introduction
- Architecture
- Rights
- Objects management
- Compilation, installation
- Coordinator service
- Worker service
- Client service
- Benchmark
- Pilot Jobs
- Perspective



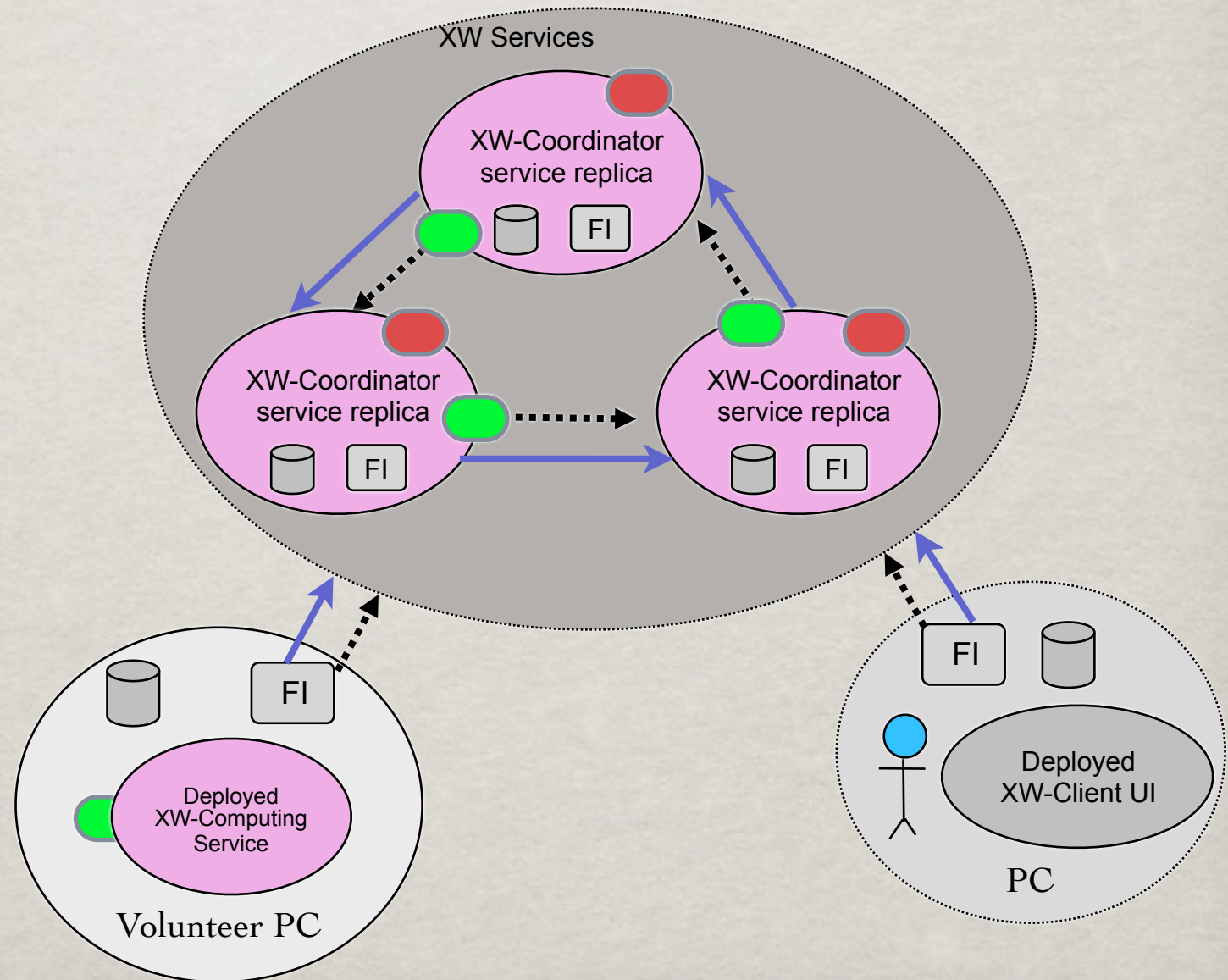
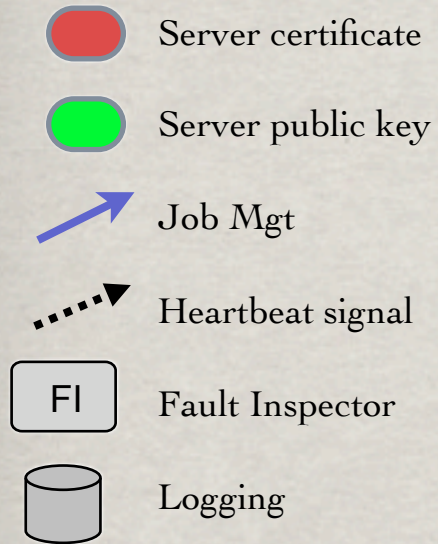
# XWHEP : ARCHITECTURE

Services are signed; communications are encrypted.  
 Distributed parts (clients, workers) must present valid credentials.





# FAULT TOLERANT MODEL



Management of  
**stateless**  
application



# XWHEP

- Introduction
- Architecture
- Rights
- Objects management
- Compilation, installation
- Coordinator service
- Worker service
- Client service
- Benchmark
- Pilot Jobs
- Perspective



# ACCESS RIGHTS

Any object in XWHEP is associated with an access rights.

Access rights are linux fs like : they are defined for the user (owner), the group and others :

- 0400 Allow read by owner.
- 0200 Allow write by owner.
- 0100 For applications, allow execution by owner.
- 0040 Allow read by group members.
- 0020 Allow write by group members.
- 0010 For applications, allow execution by group members.
- 0004 Allow read by others.
- 0002 Allow write by others.
- 0001 For applications, allow execution by others.

Default access rights is 0x755

The `xwchmod` command helps to change access rights.



# ACCESS RIGHTS

Access rights help to define access types

Access Types	Default Access Rights
Private	700
Group	750
Public	755



# ACCESS RIGHTS

Some sensitive datas are **private** with no way to change their access rights.

This is typically the case of X509 proxy which may be temporary stocked on XWHEP data repository.

This ensures access to data owner only.



# ACCESS RIGHTS



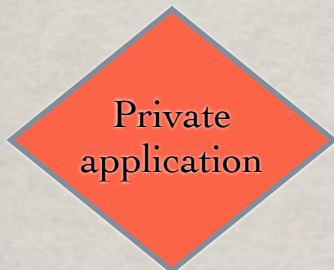
## Public applications:

- can only be inserted with administrator user rights
- all users can submit jobs for such applications
- referring jobs are public jobs



## Group applications:

- can only be inserted with administrator user rights
- only group users can submit jobs for such applications
- referring jobs are group jobs



## Private applications:

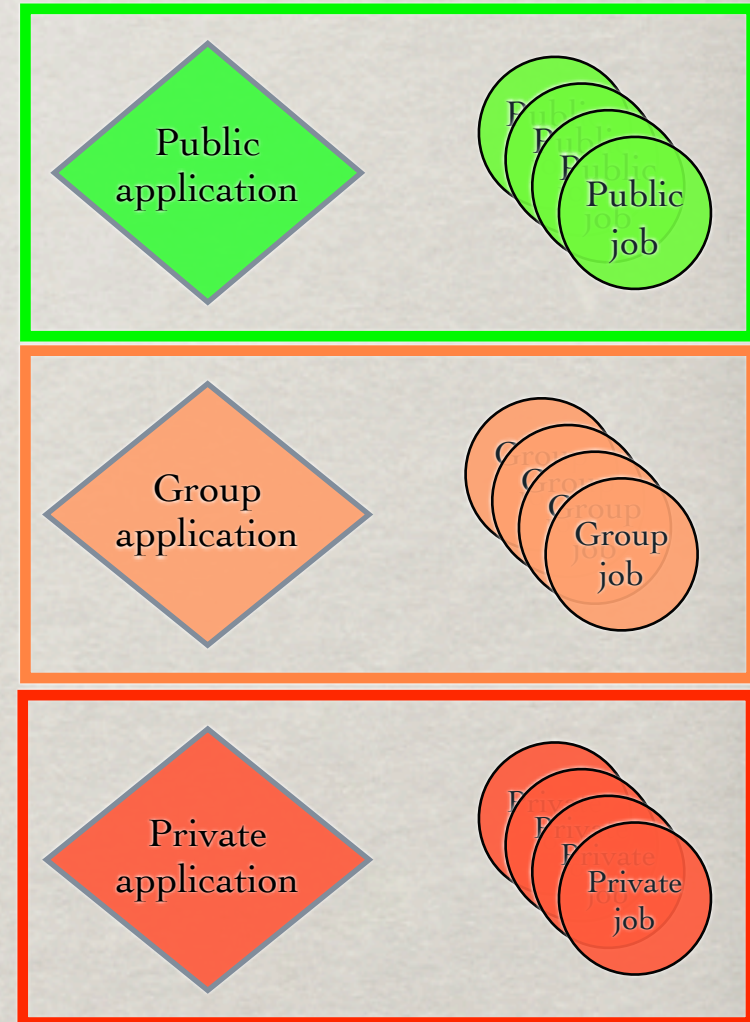
- any user can insert private applications
- only application owner can submit jobs for such applications
- referring jobs are private jobs



# ACCESS RIGHTS

Jobs access rights depend of the level of the referenced application.

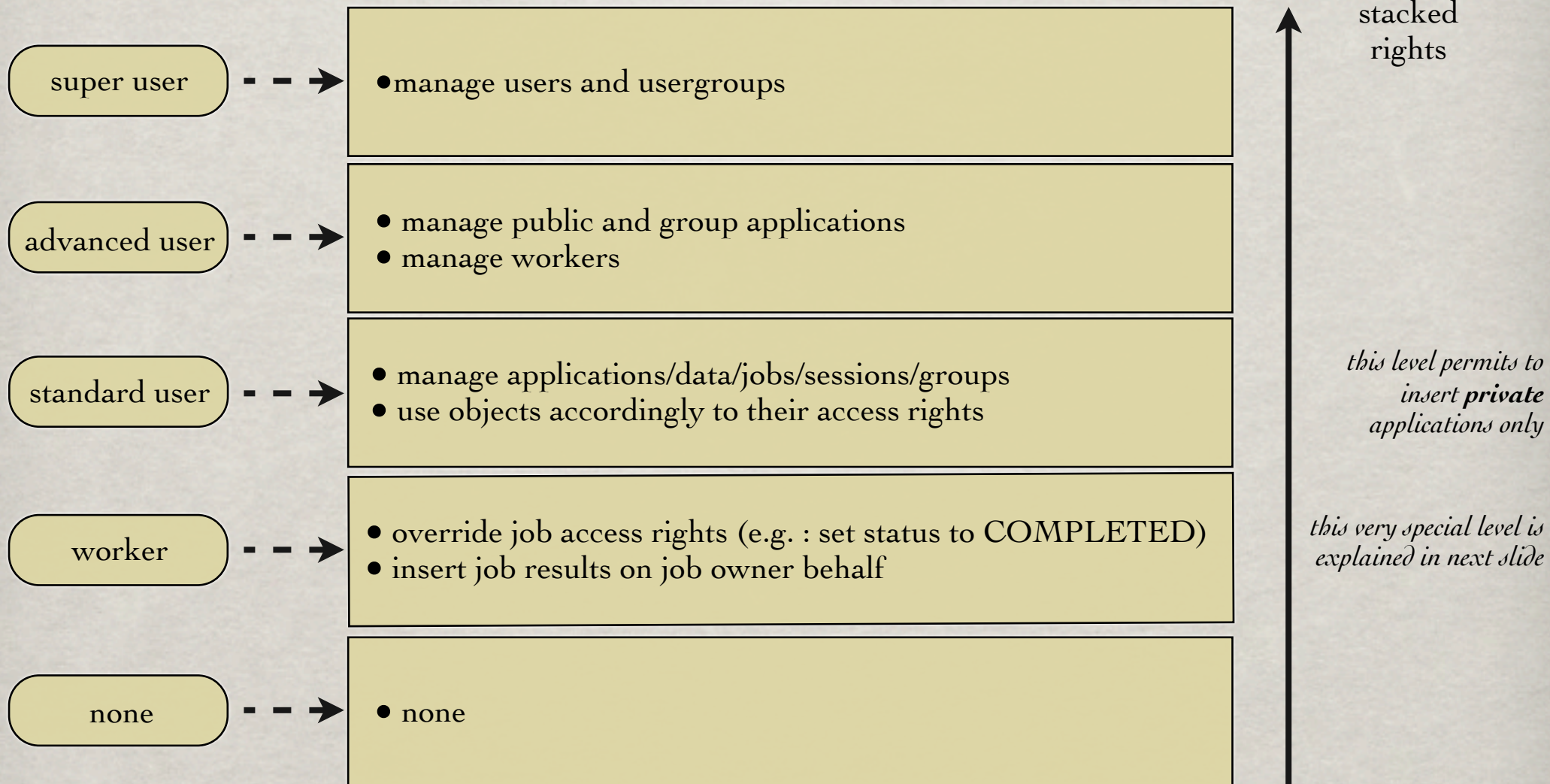
There is no way to extend job access rights





# AUTHORIZATION

Credentials define usage level





# AUTHORIZATION

Public and group workers have WORKER\_USER credentials.

This make workers able to compute jobs.

No other action is allowed with such credentials: it is not permit to insert application or submit jobs.

This is due to the fact that worker (with their credentials) are widely distributed to untrusted volunteer PCs and it would be too easy to hack worker credentials.



# CONFIDENTIALITY

User rights associated to access rights permit to confine deployment and executions with three levels:

- **public**
- **group**
- **private**



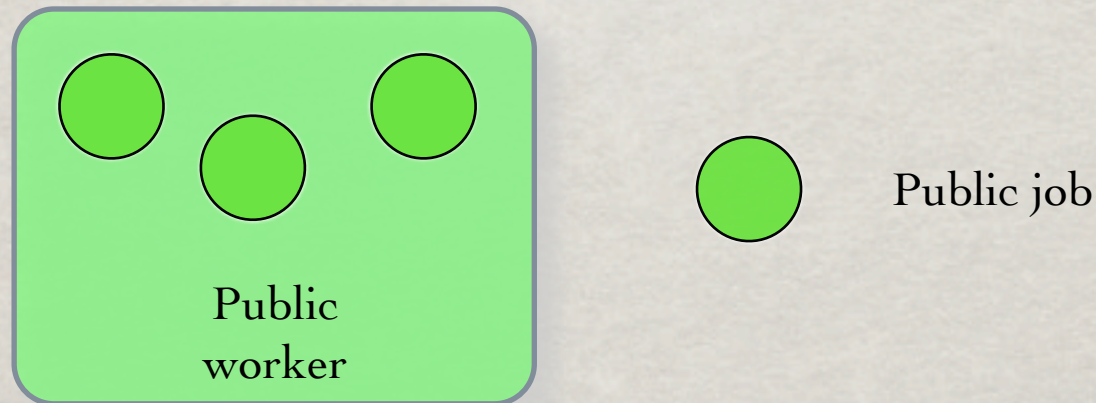
# CONFIDENTIALITY

Deployment confinement:

**public worker** has `WORKER_USER` credentials.

Execution confinement:

**public worker** can execute any public job, and public jobs only.





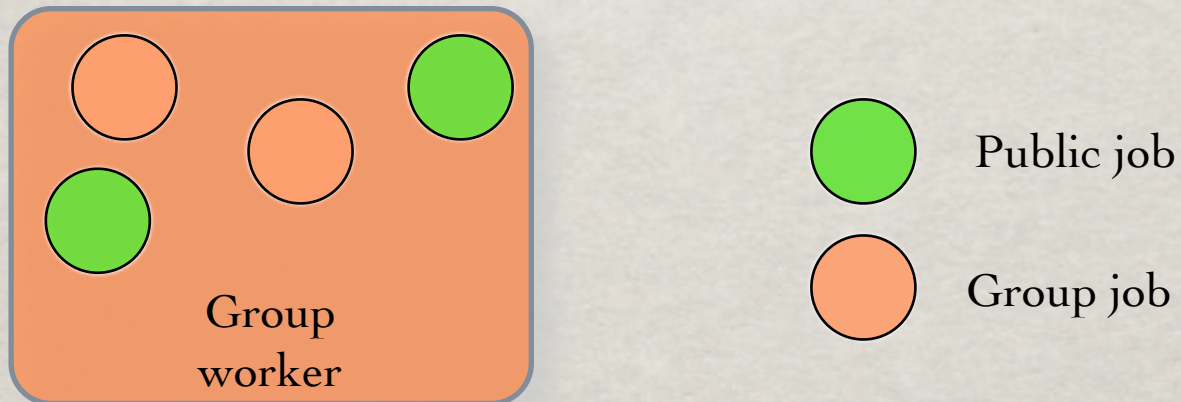
# CONFIDENTIALITY

## Deployment confinement:

**group worker** has WORKER\_USER credentials.

## Execution confinement:

**group worker** can execute any public job, any jobs of its group, and its group only.

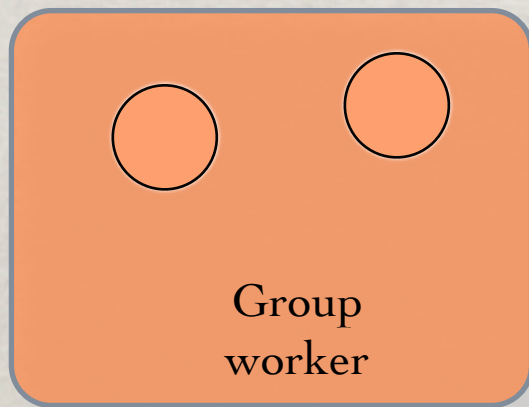




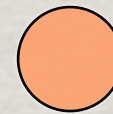
# CONFIDENTIALITY

Execution confinement:

**group worker** can also be strictly confined to its group.



Public job



Group job



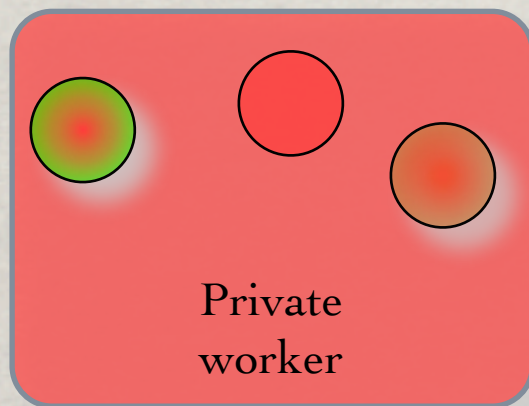
# CONFIDENTIALITY

Deployment confinement:

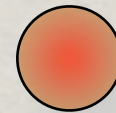
**private worker** has STANDARD\_USER credentials.

Execution confinement:

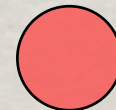
**private worker** can execute any job of its owner, and its owner only.



Public job of the worker owner.



Group job of the worker owner.



Private job of the worker owner.



# XWHEP

- Introduction
- Architecture
- Rights
- Objects management
- Compilation, installation
- Coordinator service
- Worker service
- Client service
- Benchmark
- Pilot Jobs
- Perspective



# OBJECTS MANAGEMENT

XWHEP defines a set of different objects.

Here we detail :

- users and user groups
- datas
- applications
- jobs
- workers

All objects are identified by an UID composed of five hexadecimal values.

Example :

81c6e97a-9d85-4aeb-ae07-593980fb611f

Null value:

00000000-0000-0000-0000-000000000000



# USERS AND USER GROUPS

<i>calculated</i>
<i>mandatory</i>
<i>optional</i>

Partial view of the internal user structure.

<b>uid</b>	
<b>login</b>	<i>string</i>
<b>password</b>	<i>string</i>
<b>rights</b>	e.g : STANDARD USER
<b>usergroupuid</b>	

Partial view of the internal user group structure.

<b>uid</b>	
<b>label</b>	<i>string</i>





# DATAS

1/3

Datas are **write once** only.  
Datas are referenced by URI.

XWHEP coordinator service may serve datas.

But data can be served by any data server as soon as they  
are described by an URI.

Data security, availability and consistency is the data server  
responsibility.



# DATAS

2/3

XWHEP introduces a new URI schema : “xw:”.  
Hence, data managed by XWHEP have URI like:  
`xw://yourServer/UID`

XWHEP can manage XW schema and HTTP schema.  
Any new schema needs to implement Client API.  
*(src/xtremweb/communications/ClientAPI.java)*

XWHEP uses data to manage :

- application binaries/libraries
- application/job input files
- job results



# DATAS

3/3

Partial view of the internal data structure

uid	
size	content size
md5	md5sum
status	available or not
links	how many objects use this data
insertionDate	the insertion date
accessDate	the last access date
owneruid	the uid of the user who owns the data
name	the name of the file
uri	the content URI
accessrights	e.g. : 0x755
type	X509 cert, raw, binary, text, zip
cpu	ppc, intel
os	linux, mac, win32

<i>calculated</i>
<i>mandatory</i>
<i>optional</i>



# APPLICATIONS

Partial view of the internal application structure

<i>calculated</i>
<i>mandatory</i>
<i>optional</i>

uid	
owneruid	the uid of the user who owns the data
accessrights	e.g. 0x755
name	the name of the file
binaryURI	the URI of the binary
mincpuspeed	used by scheduler
minmemory	used by scheduler
defaultStdinURI	the URI of the default stdin
baseDirinURI	the URI of the dirin provided to all jobs
defaultDirinURI	the URI of the default dirin

if set, this is provided to jobs by default.  
Jobs may override this.

if set, this is always expanded  
on worker FS

if set, this is provided to jobs by default.  
Jobs may override this.



# JOBS

Partial view of the internal job structure

uid	
accessrights	e.g. 0x755
appid	the UID of the application to run
userid	the UID of the owner
X509 userproxy	the URI of the user X509 proxy
result	the URI to store the result
cmdLine	the command line
stdin	the URI of the stdin
dirin	the URI of the dirin provided to all jobs
expectedHost	the UID of the worker this job <b>MUST</b> run on

calculated
mandatory
optional

If set, this allows Pilot job usage.  
Jobs can only be executed by workers with the same user proxy

If not set, XWHEP automatically a new data

If not set, use app default, if any.  
Set NULLURI if app default is not expected.



# WORKERS

Partial view of the internal host structure

*calculated*

uid	
ownerUID	
natedIPAddress	local IP address
IPAddress	public IP address
X509 user proxy	the URI if the X509 user proxy
avg. ping	
avg. upload bandwidth	
OS	linux, win32, mac
CPU	intel, ppc
CPU speed	
mem/swap	
alive	still connected ?
available	according to local policy
active	the platform may use this worker

If set, this allows Pilot job usage.  
Worker can only execute jobs with  
the same X509 user proxy



# XWHEP

- Introduction
- Architecture
- Rights
- Objects management
- Compilation, installation
- Coordinator service
- Worker service
- Client service
- Benchmark
- Pilot Jobs
- Perspective



# REQUIREMENTS

To compile, install and run XWHEP, one needs :

- java SDK 1.5 or above

Optional requirements :

mysql	the package includes embedded hsqldb engine, if mysql not desired
apache	for dissemination and monitoring only



# SOURCE TREES

The distribution source tree contains :

build/	configuration, ant and make files
classes/	third party libraries
doc/	XWHEP documentation
misc/	runtime configuration files
php/	web pages
src/	source tree



# CONFIGURATION

## Build.conf

```
xtremweb.admin.login=administrator  
xtremweb.admin.password=xwpassword
```

```
xtremweb.worker.login=worker  
xtremweb.worker.password=aWorkerPassword
```

```
dispatcher.servers=localhost  
# Default : ${dispatcher.servers}  
#data.servers=localhost
```

```
launcher.url=http://localhost
```

```
db.system=mysql  
db.host=localhost  
#db.engine=MEMORY  
db.su.login=root  
#db.su.password=  
db.name=xtremweb
```

```
install.dir=/opt  
install.www.dir=/Users/oleg/Sites/XWHEP  
ganglia.www.dir=/Users/oleg/Sites/ganglia
```

```
xw.passwordPass=some chars to generate keys
```

```
# By default, the xtremweb.admin.login  
# and xtremweb.admin.password are used  
#db.user.login=xtremweb  
#db.user.password=
```

```
debug=on  
logger.level=error
```



# COMPILE & INSTALL

## Compile and install

It is not mandatory to compile for each platform; one successful compilation generates a single jar file for all platforms.

As soon as the build.conf is correct

```
bash $> export JAVA_HOME="..."  
csh $> setenv JAVA_HOME "..."
```

```
$> make installDB  
$> make  
$> make createKeys  
$> make install
```

## What is installed

- The distribution
  - ✓ `${install.dir}`
- The win32 client
  - ✓ `build/installers/win32/xtremwebclient-1.0.28`
- The Mac OS X worker
  - ✓ `build/installers/macosx/installer`
    - ➔ use `xtremwebworker.pproj` to generate Mac OS X package

There is no automatic way to generate the Win32 MSI



# XWHEP

- Introduction
- Architecture
- Rights
- Objects management
- Compilation, installation
- Coordinator service
- Worker service
- Client service
- Benchmark
- Pilot Jobs
- Perspective



# SERVER CONFIGURATION

## xtremweb.server.conf

- src/misc/xtremweb.server.conf.in
- /opt/XWHEP-1.0.29/conf/xtremweb.server.conf

```
xtremweb.role=server
```

### *Database*

```
# mysql
XWdbVendor: mysql
XWdbHost: @DBHOST@
XWdbName: @DBNAME@
XWdbUser: @DBUSER@
XWdbPass: @DBPASSWORD@
# hsqldb on disk
# XWdbVendor: hsqldb
# XWdbHost: @DBHOST@
# XWdbName: @DBNAME@
# XWdbUser: @DBUSER@
# XWdbPass: @DBPASSWORD@
# hsqldb in memory
# XWdbVendor: hsqldb:mem
# XWdbHost: @DBHOST@
# XWdbName: @DBNAME@
# XWdbUser: @DBUSER@
# XWdbPass: @DBPASSWORD@
```

### *HTTP*

```
#server.http=false
```

### *Misc*

```
HomeDir: @HOMEDIR@
```

### *Security*

```
XWkeyStore: @KEYDIR@/server.keys
XWpassPhrase: @PASSWORDPASS@
```

### *ACL*

```
# server.comm.acl=.*
# server.stat.acl=+*.lal.in2p3.fr,-168.192.*.*
```

### *Logging*

```
#mileStones=xtremweb
logger.level=@LOGGERLEVEL@
```



# Server control

## Control the server

### The server

➡ /etc/init.d/xtremweb.server

➡ /opt/XWHEP-1.0.29/bin/xtremweb.server [start|stop|console]



# XWHEP

- Introduction
- Architecture
- Rights
- Objects management
- Compilation, installation
- Coordinator service
- Worker service
- Client service
- Benchmark
- Pilot Jobs
- Perspective



# Worker configuration

## xtremweb.worker.conf

```
xtremweb.role=worker
```

```
launcher.url=@LAUNCHERURL@  
dispatcher.servers=@DISPATCHERS@  
#data.servers=@DATASERVERS@
```

```
login=@DEFAULTUSER@  
password=@DEFAULTPASSWORD@
```

*Servers*

*SG-DG Bridging*

```
#computing.jobs=-1  
#noopTimeout=-1
```

- src/misc/xtremweb.worker.conf.in
- /opt/XWHEP-1.0.29/conf/xtremweb.worker.conf

*HTTP*

```
#server.http=false
```

*Misc*

```
workpool.size=1  
#project=  
# path.tmpdir=/tmp/XW.tmp  
# acceptBin=true  
## activator.class=xtremweb.worker.AlwaysActive  
#activator.class=xtremweb.worker.DateActivator  
activator.date=* 20-7  
#commHandlers= xw:xtremweb.communications.TCPClient,http:xtremweb.communications.HTTPClient
```

*Security*

```
XWkeyStore=@KEYDIR@/worker.keys  
cert.uri=URI to X509 user proxy (file://, xw://srv/uid etc. )
```

*ACL*

```
# server.comm.acl=.*  
# server.stat.acl=+*.lal.in2p3.fr,-168.192.*.*
```

*Logging*

```
#mileStones=xtremweb  
logger.level=@LOGGERLEVEL@
```



# Worker control

## Control the worker : linux like

### Linux

- ➡ /etc/init.d/xtremweb.worker
- ➡ /opt/XWHEP-1.0.29/bin/xtremweb.server [start|stop|restart|console]

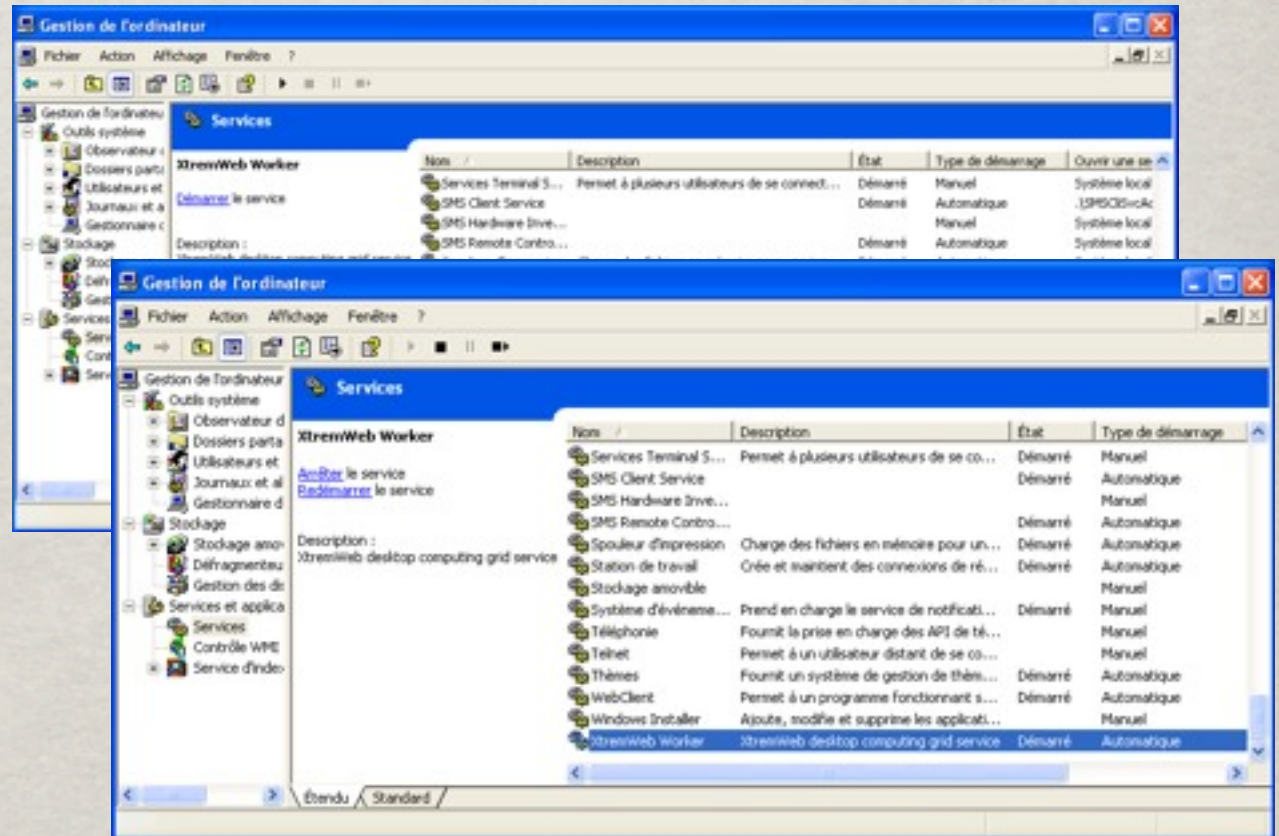
### Mac OS X

- ➡ /Library/StartupItem/xtremweb.worker/xtremweb.worker [start|stop|restart]
- ➡ /private/etc/xtremweb.worker/
- ➡ /usr/local/bin/xtremweb.worker



# Worker control

Control the worker : win32





# XWHEP

- Introduction
- Architecture
- Rights
- Objects management
- Compilation, installation
- Coordinator service
- Worker service
- Client service
- Benchmark
- Pilot Jobs
- Perspective



# Client configuration

## xtremweb.client.conf

- src/misc/xtremweb.client.conf.in
- /opt/XWHEP-1.0.29/conf/xtremweb.client.conf

```
xtremweb.role=client
```

### *Servers*

```
launcher.url=@LAUNCHERURL@  
dispatcher.servers=@DISPATCHERS@  
#data.servers=@DATASERVERS@
```

```
login=@DEFAULTUSER@  
password=@DEFAULTPASSWORD@
```

### *Security*

```
XWkeyStore=@KEYDIR@/worker.keys
```

### *Misc*

```
#commHandlers= xw:xtremweb.communications.TCPClient,http:xtremweb.communications.HTTPClient
```

### *Logging*

```
#mileStones=xtremweb  
logger.level=@LOGGERLEVEL@
```



# Client control

## Control the client : linux like

### *Send objects*

- xwsendwork
- xwsubmit
  
- xwsendapp
- xwsenddata
- xwsendgroup
- xwsendsession
- xwsenduser
- xwsendusergroup

### *Get objects*

- xwapps [UID|URI ...]
- xwdatas [UID|URI ...]
- xwgroups [UID|URI ...]
- xwsessions [UID|URI ...]
- xwtasks [UID|URI ...]
- xwusers [UID|URI ...]
- xwusergroups [UID|URI ...]
- xwworkers [UID|URI ...]

### *Manage Objects*

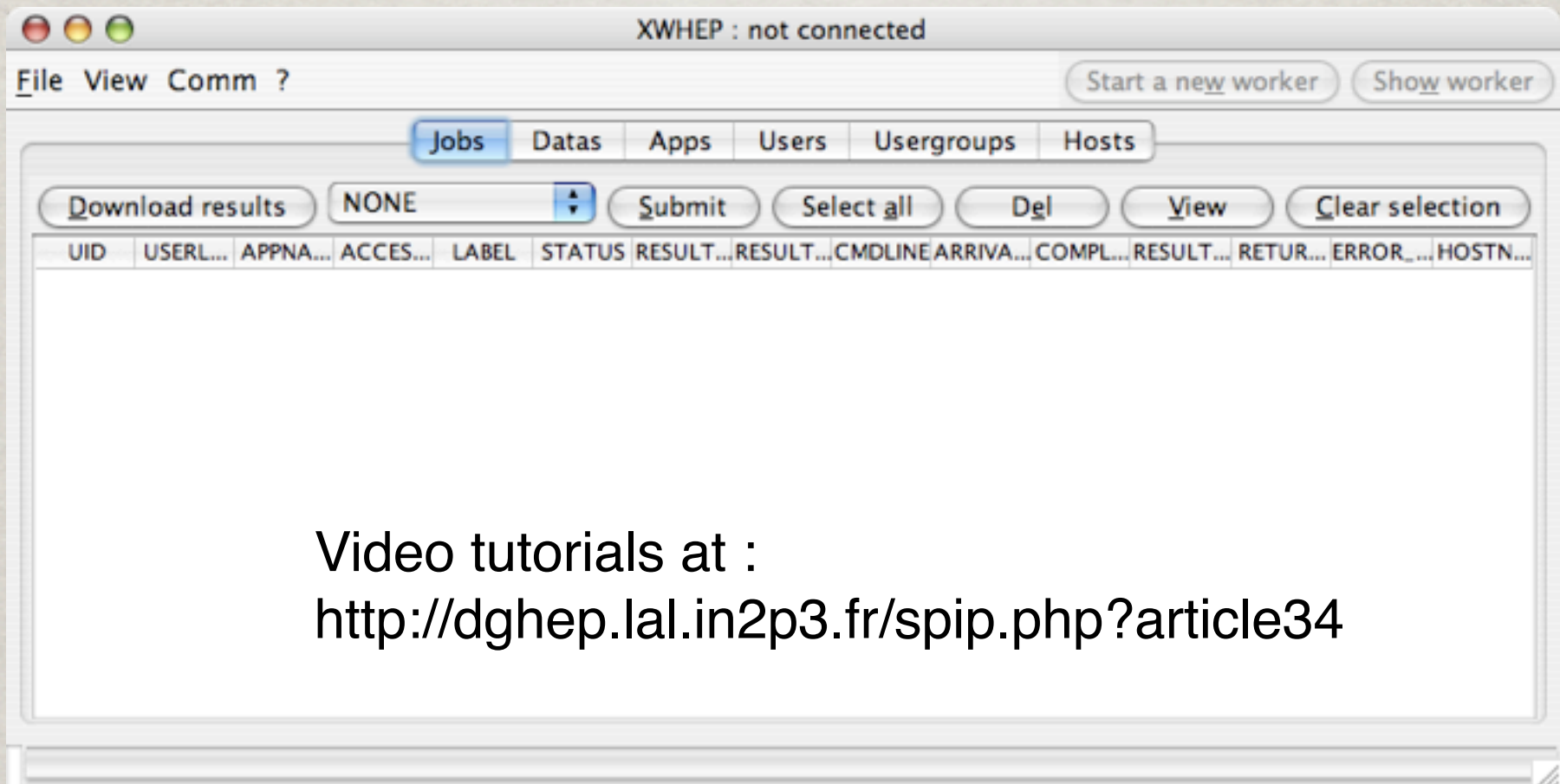
- xwchmod
- xwrm

### *Misc*

- xwgui



# Client GUI



Video tutorials at :  
<http://dghep.lal.in2p3.fr/spip.php?article34>



# XWHEP

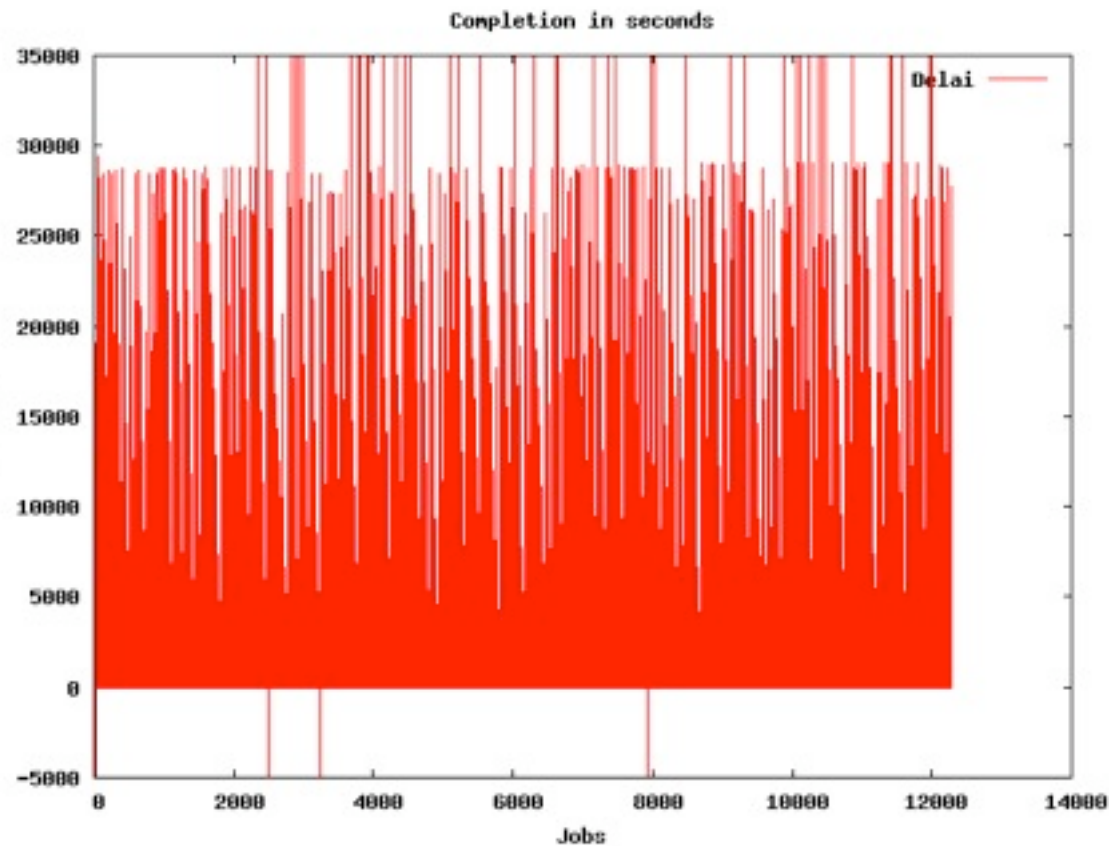
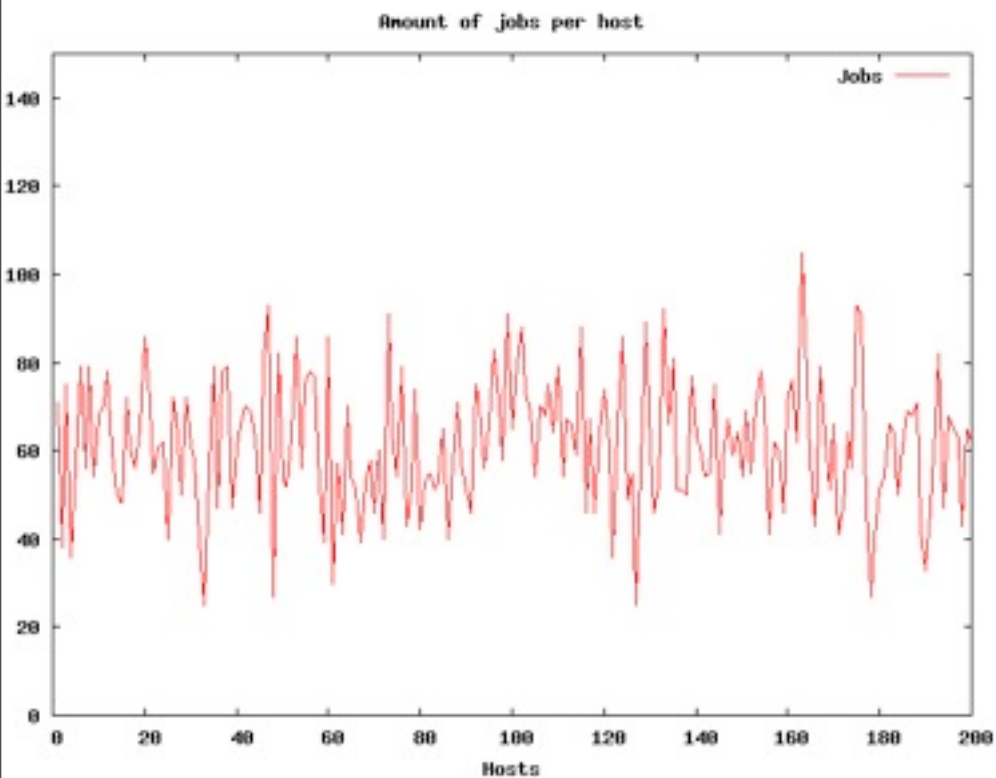
- Introduction
- Architecture
- Rights
- Objects management
- Compilation, installation
- Coordinator service
- Worker service
- Client service
- Benchmark
- Pilot Jobs
- Perspective



# BENCHMARK

	Hosts	Status	count(*)
997MHz	1	COMPLETED	12283
2GHz	104		
2.4GHz	95		

*Run on Grid5000  
thanks to  
Haiwu He*





# XWHEP

- Introduction
- Architecture
- Rights
- Objects management
- Compilation, installation
- Coordinator service
- Worker service
- Client service
- Benchmark
- Pilot Jobs
- Perspective



# PILOT JOBS INTRODUCTION

Pilot Jobs is a way to use a Grid infrastructure to deploy end user jobs with an external scheduler (i.e. a scheduler which is not part of the infrastructure itself).

XtremWeb and Condor teams have introduced this as “Glide-in” in  
“XtremWeb & Condor : sharing resources between Internet connected Condor pools.”

*O. Lodygensky, G. Fedak, F. Cappello, V. Neri, M. Livny, D. Thain*  
CCGRID 2003, Tokyo, JAPAN; May 12-15, 2003.

## EGEE experiments use Pilot Jobs

*LHCb Dirac*

*CMS Glide-in*

*ATLAS Panda*

ALICE



# PILOT JOBS MONITORING

Security, monitoring and logging are the main issues in Pilot Jobs. (<http://edms.cern.ch/document/855383>)

XWHEP solves these issues thanks to its innovative features:

- user rights management
- user rights delegation
- user groups
- user group applications

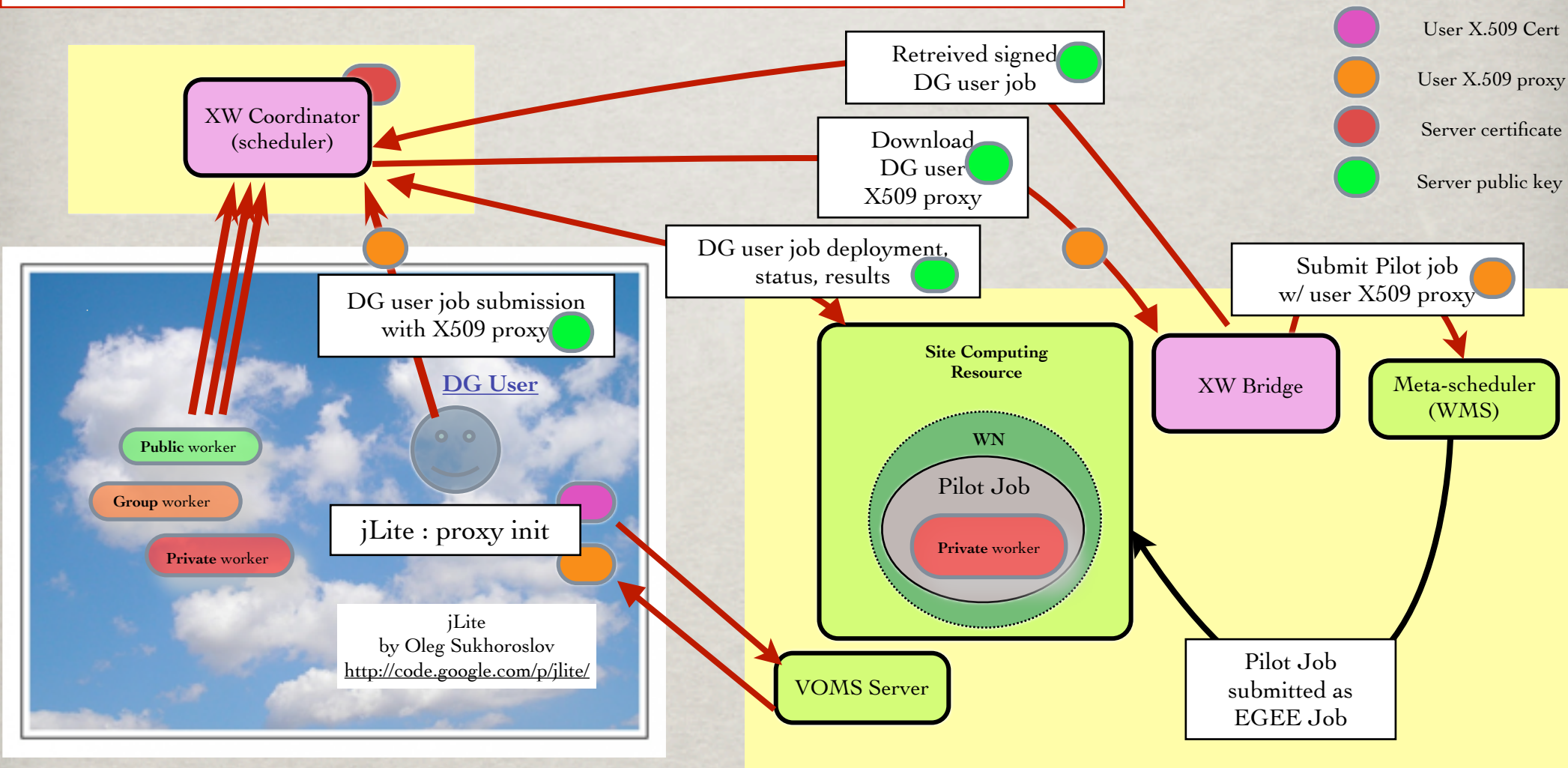
Security is ensured at three levels:

- 1.computing node.
  - a) XWHEP includes a sandbox to isolate end user job computation
  - b) only validated applications from repository are candidate to run on SG nodes
2. Application and data integrity.
  - a) application repository and data servers (including XWHEP) ensure integrity
3. User authentication
  - a) only X.509 certified users can use SG nodes
  - b) users provide proxy certificate to submit a job to XWHEP scheduler
  - c) this proxy is used to submit Pilot Jobs to SG



# Pilot Jobs

Security, monitoring and logging are the main issues in Pilot Jobs. (<http://edms.cern.ch/document/855383>)





# WEB SITE

<http://dghep.lal.in2p3.fr/?lang=en>



# XWHEP

- Introduction
- Architecture
- Rights
- Objects management
- Compilation, installation
- Coordinator service
- Worker service
- Client service
- Benchmark
- Pilot Jobs
- Perspective



# PILOT JOBS

## jLite

by Oleg Sukhoroslov

<http://code.google.com/p/jlite/>

jLite provides a high-level Java API with basic functionality similar to gLite shell commands.  
This API hides the complexity of underlying middleware and its configuration

XWHEP next version will use jLite API to easily manage X509 certificates with VOMS extensions.



# PILOT JOBS MONITORING

Security, monitoring and logging are the main issues in Pilot Jobs. (<http://edms.cern.ch/document/855383>)

XWHEP solves these issues thanks to its innovative features:



- user rights management
- user rights delegation
- user groups
- user group applications

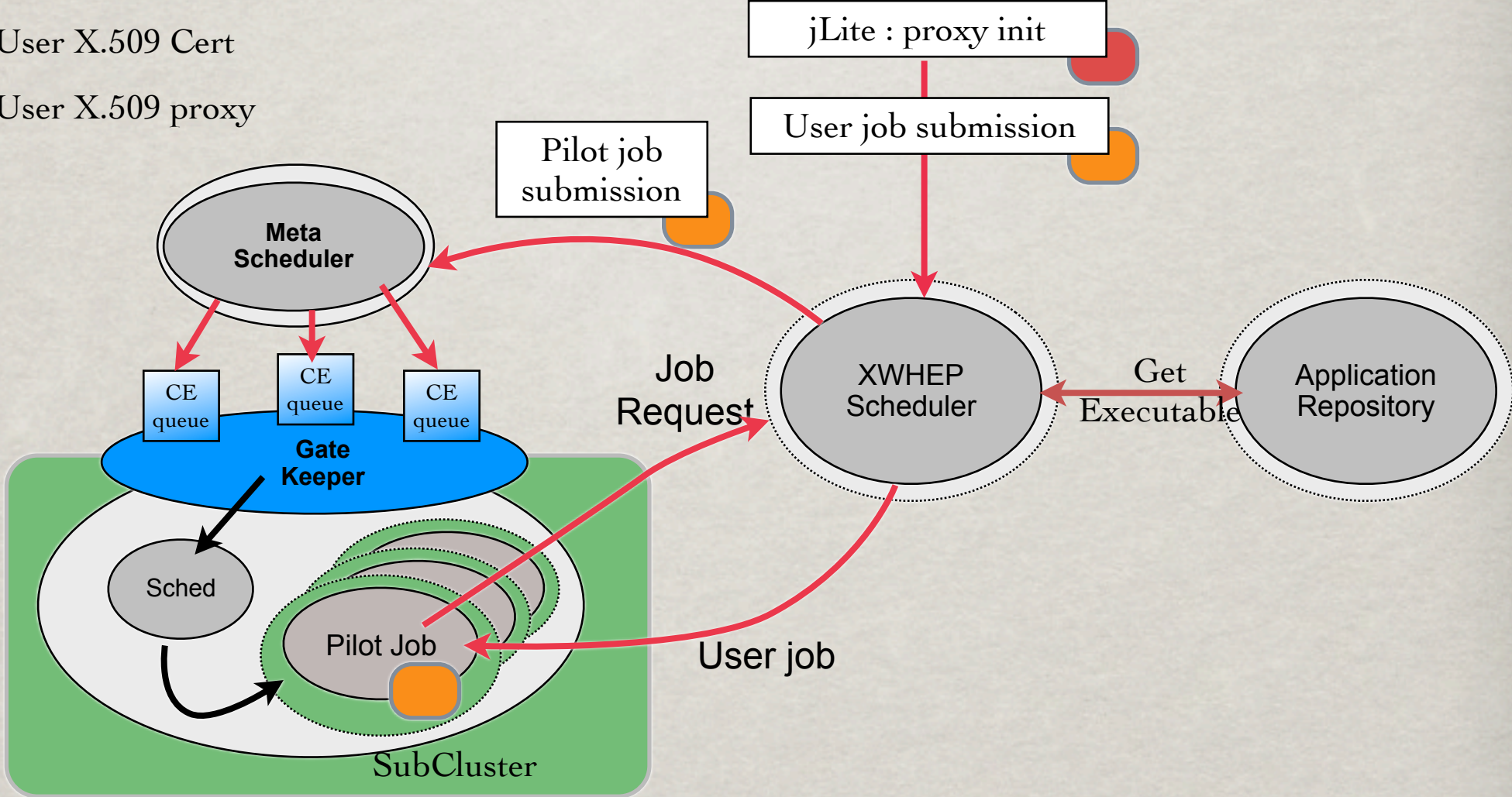
Security is ensured at three levels:

- 1.computing node.
  - a) XWHEP includes a sandbox to isolate end user job computation
  - b) only validated applications from repository are candidate to run on SG nodes
2. Application and data integrity.
  - a) application repository and data servers (including XWHEP) ensure integrity
3. User authentication
  - a) only X.509 certified users can use SG nodes
  - b) users provide proxy certificate to submit a job to XWHEP scheduler
  - c) this proxy is used to submit Pilot Jobs to SG



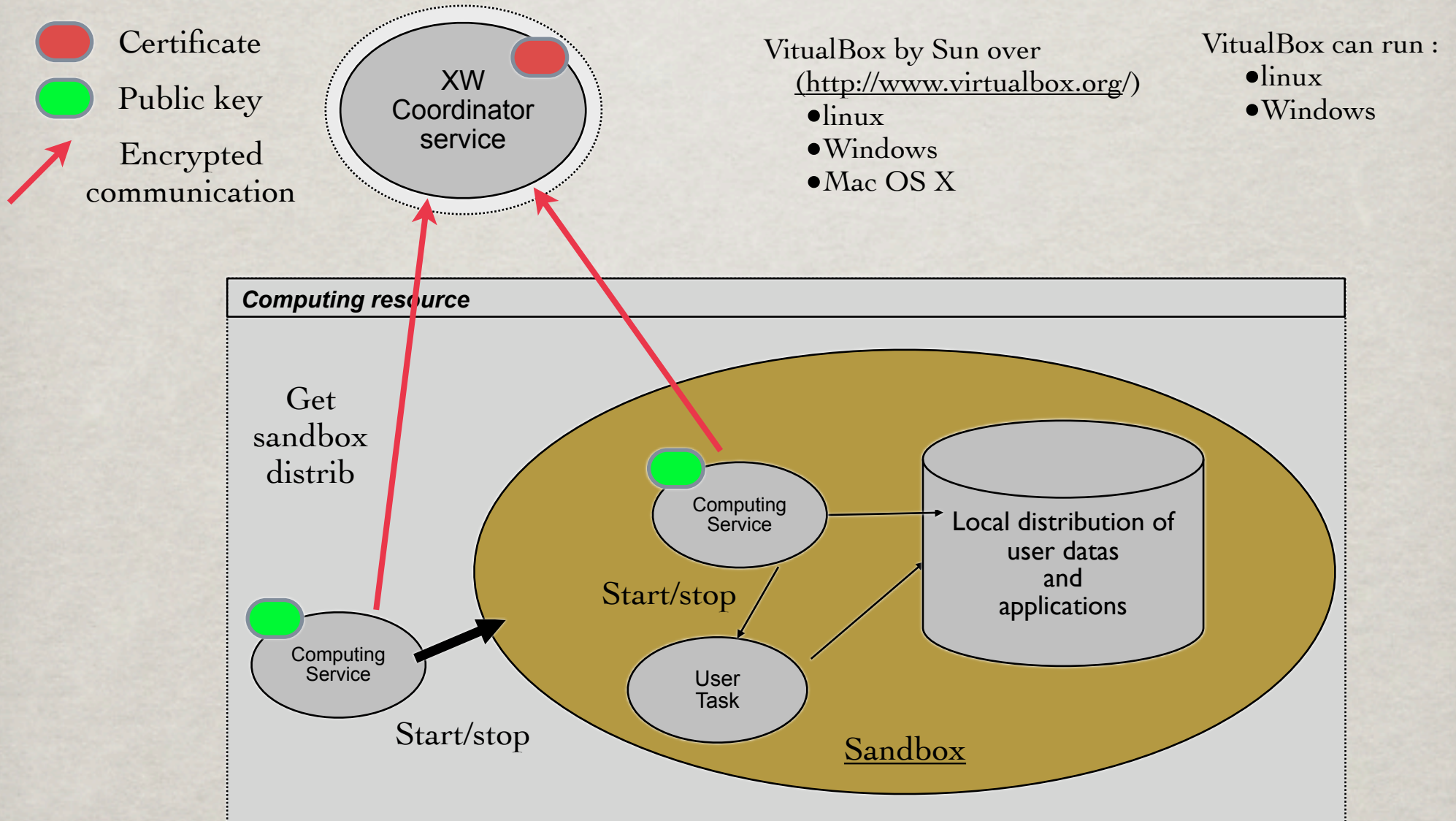
# PILOT JOBS

-  User X.509 Cert
-  User X.509 proxy





# PERSPECTIVE : SANDBOXING








# PERSPECTIVE : HOLE PUNCHING

 Certificate

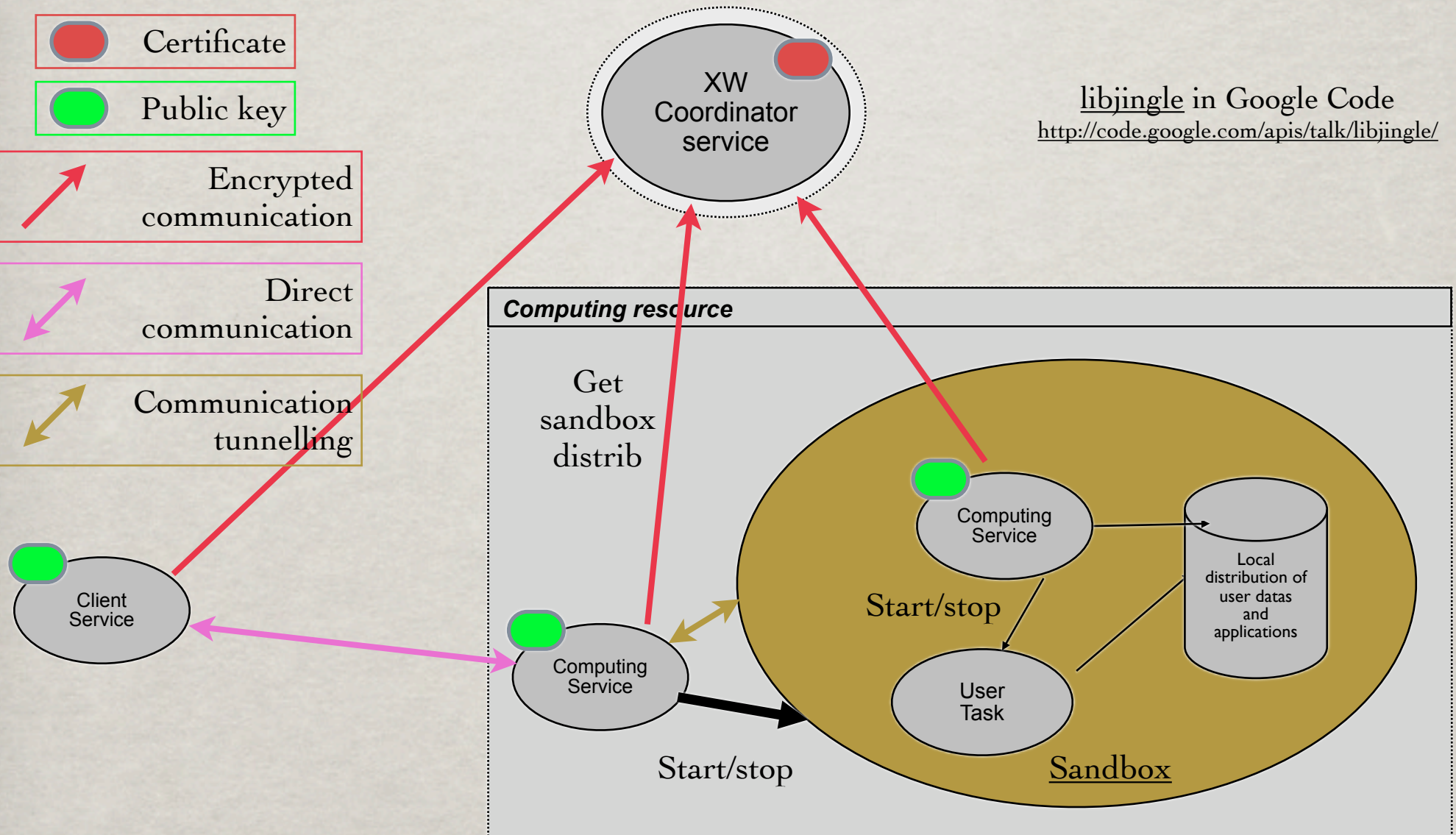
 Public key

 Encrypted communication

 Direct communication

 Communication tunnelling

libjingle in Google Code  
<http://code.google.com/apis/talk/libjingle/>





# PERSPECTIVE : CLOUD COMPUTING

Application sandboxing  
+ OS deployment on the fly  
= Cloud Computing