XWHEP : XtremWeb for High Energy Physics

XtremWeb 2.0

XWHEP

- Introduction
- Architecture
- Rigths
- Objects management
- Coordinator service
- Client service
- Worker Service

Presentation

XWHEP is a generic multi purposes desktop grid platform (*DG*) enabling eSciences applications deployment.

XWHEP (*"XtremWeb 2.0"*) is based on XtremWeb 1.8.0.
Main features are :

- three tiers architecture
- virtual stable cluster over volatile volonteers individual PCs
- multi applications
- multi users
- firewall bypassing
- automatic laod balancing

Goal

XWHEP aims to propose a global computing platform by proposing its own ressources and sharing others with EGEE.
To achieve this goal, XWHEP will :

- propose a secured DG, enabling certificate (*Cert*) usage, focusing on EGEE Cert.
- define different usage levels including two major ones : "public" and "private", which may be divided in sublevels :
  - ➡ "public", intrinsically non secured, will accept public ressources; such ressources will not be shared with EGEE.
  - ➡ "private", intrinsically secured, will not accept public ressources (insecured by essence). Such ressources may be shared with EGEE.
  - ➡ levels may be divided in projects, subprojects etc.

XtremWeb 2.0 vs 1.8

	XtremWeb 2.0 "XWHEP" (Oct 2007)	XtremWeb 1.8 (Oct 2006)
Data management	+	-
Multi protocols (transport)	+ **(UDP, TCP)**	+ **(UDP, TCP)**
Multi protocols (communication)	+ **(XW, HTTP)**	+ **(XW only)**
SSL / certificates	*Scheduled (HTTPS)*	-
User application management	+	- **(admin onlv)**
User worker management	+	-

XtremWeb 2.0 vs 1.8 2/2

	XtremWeb 2.0 "XWHEP" (Oct 2007)	XtremWeb 1.8 (Oct 2006)
Avg ping	+	-
Avg bandwidth usage	+	-
Custom scheduler	+	-
Worker launcher	+	-
Input files / job	+	+
Input files / app	+	-
CPU/RAM requirements per job	+	+
CPU/RAM requirements per app	+	-

XWHEP

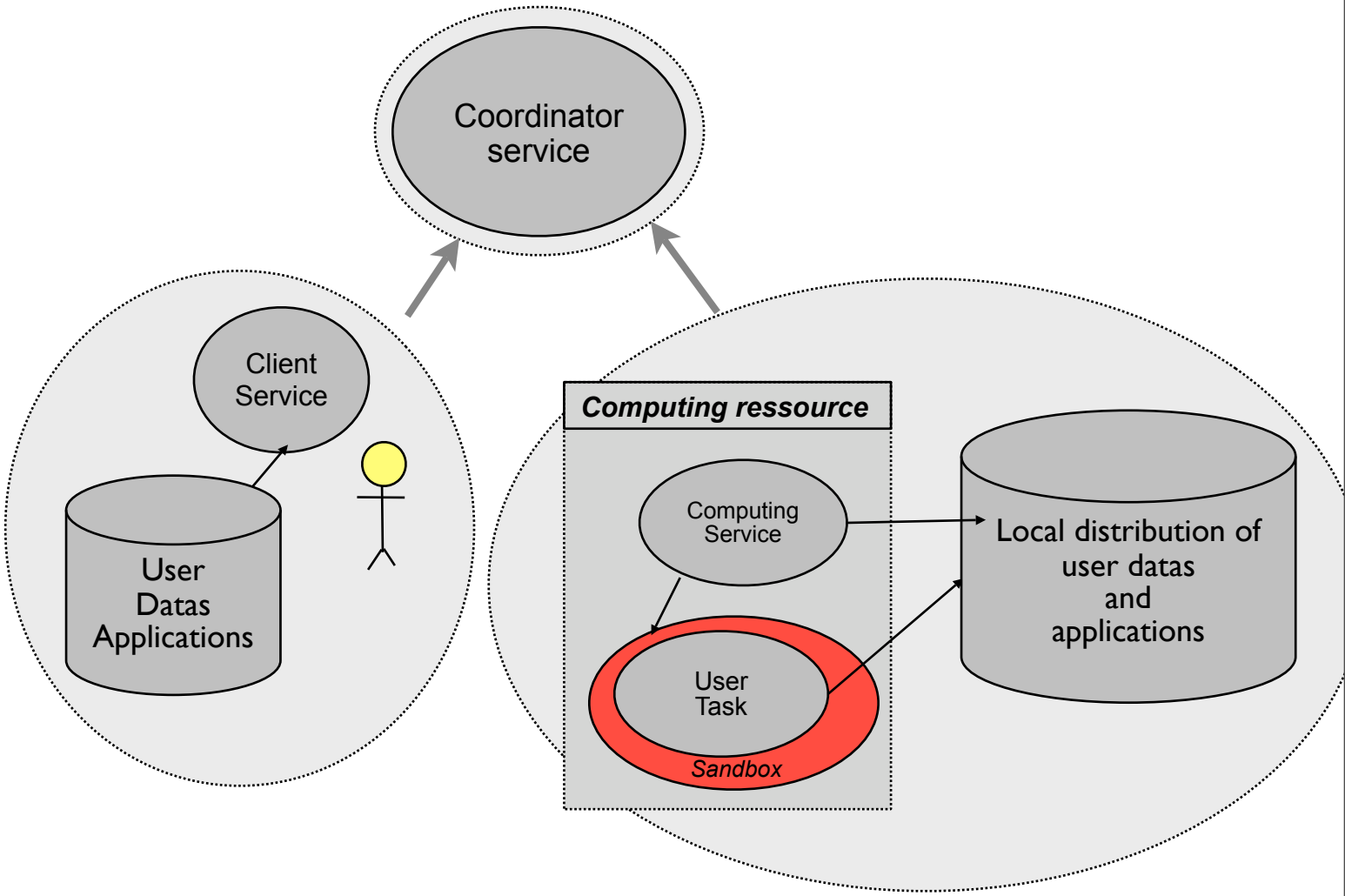
- Introduction
- Architecture
- Rigths
- Objects management
- Coordinator service
- Client service
- Worker Service

Three tiers architecture

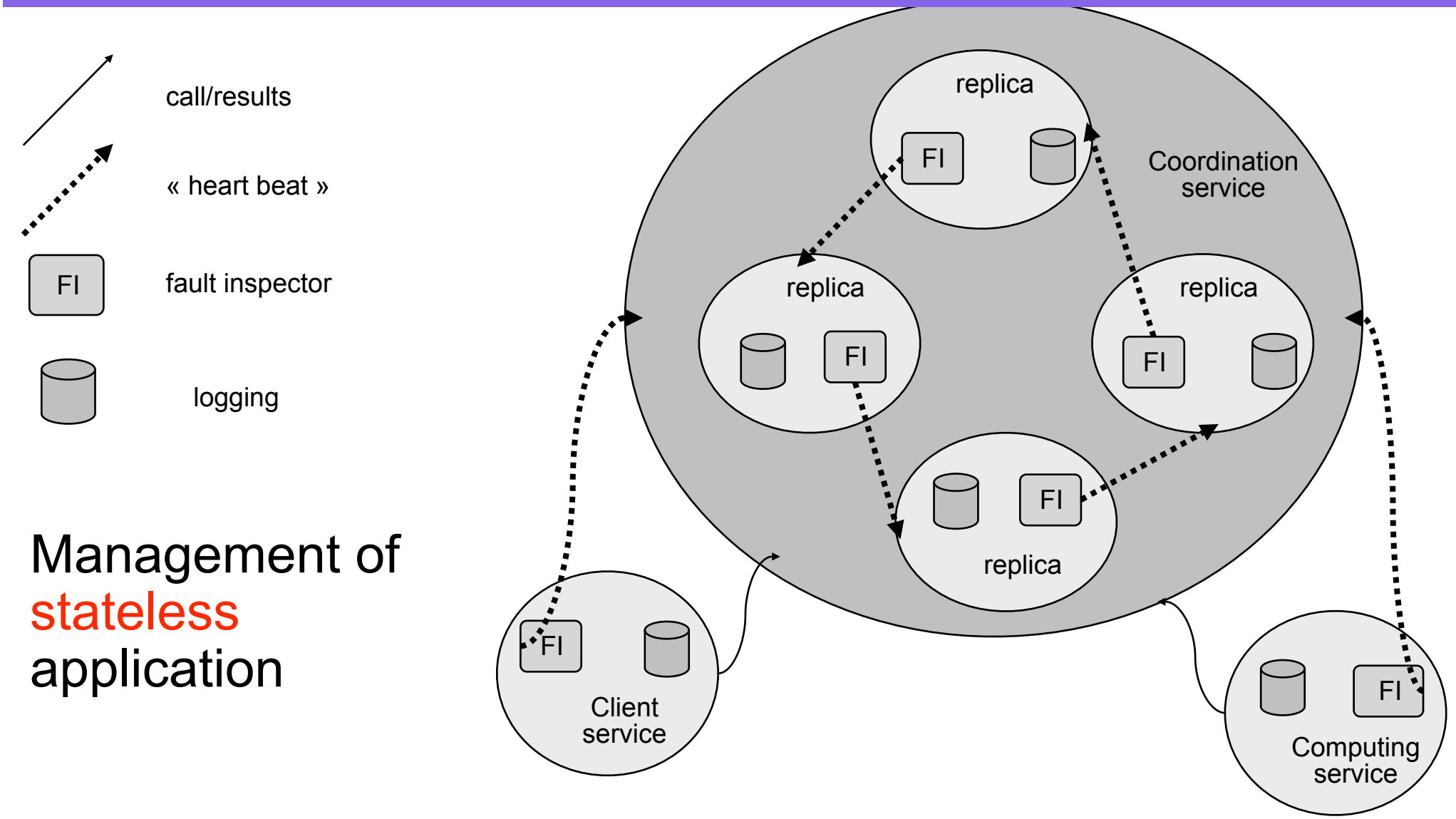
Main characteristics :

- Firewalls bypassing
- three tiers architecture
- mode « pull »
- multi-protocoles
- multi-applications
- security
- multi-OS
  - Mac OS X
  - Windows
  - Linux
- cycle stealing
- volunteers PC

- User client
- User task distribution



Fault tolerant model



Management of stateless application

XWHEP

- Introduction
- Architecture
- Rigths
- Objects management
- Coordinator service
- Client service
- Worker Service

XWHEP : access rights

Any object in XWHEP is associated with an access rights definition.
Access rights are linux fs like : they are defined for the user (owner), the group and others :

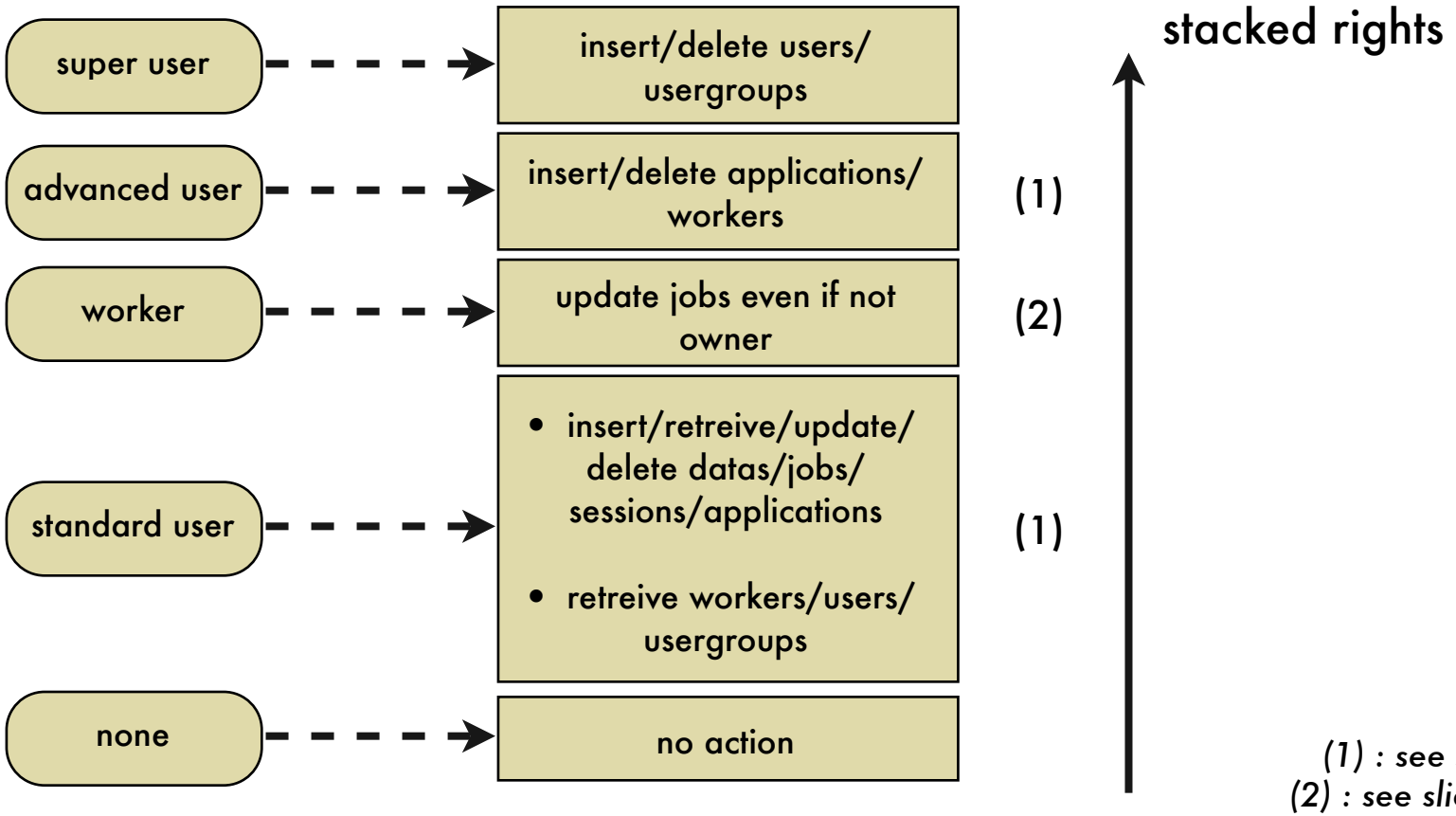
0400	Allow read by owner.
0200	Allow write by owner.
0100	For applications, allow execution by owner.
0040	Allow read by group members.
0020	Allow write by group members.
0010	For applications, allow execution by group members.
0004	Allow read by others.
0002	Allow write by others.
0001	For applications,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

Default access rights is 0x755

The xwchmod command helps to change access rights.

XWHEP : user rights

User rights define interaction level for each user.
XWHEP extends user rights as defined since XtremWeb 1.8.0.



(1) : see slide 14
(2) : see slides 15/16

XWHEP : user rights

The user rights are used accordingly to access rights.

Users can get an object only if this is readable.

Users can modify/delete an object only if it is writable.

XWHEP : user rights

Inserting a new application is a special case which leads to two differents scenarii.

1. the user has advanced rights. The application can be used by any user, accordingly to the application access rights.
2. the user does not have advanced rights
   2.1. the application can only be used by user itself
   2.2. jobs will only be run by workers of the user itself

Application access rights modifications need advanced user rights.

XWHEP : user rights

Workers are all identified by a user definition (login/password, certificate etc.). These worker identities follow the general user rights/access rights rules.
Any object (app/data/job) willing to be managed must be readable.

We may hence define:

- private workers may manage objects of the user itself.
- group workers may manage objects of the same group.
- public workers may manage any objects.

XWHEP : user rights

1. Private workers use a private user identity.
2. Group workers use a user identity included in the user group.

3. Public workers are a very special case :
   - they need to be able to overide write access because workers need to be able to udpate objects (e.g. set the job status to "completed")
   - they need a specific user rights : the worker rights allows to overide write access.

XWHEP

- Introduction
- Architecture
- Rigths
- Objects management
- Coordinator service
- Client service
- Worker Service

XWHEP : objects management

XWHEP defines a set of differents 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

XWHEP : users and groups

Partial vue of the internal user structure.

uid	
usergroupuid	
login	
password	
password	
rights	the user rights

Partial vue of the internal user group structure.

uid	
label	

XWHEP : datas

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 responsability.

XWHEP : datas

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

Where UID is a 5 hex digits value as described earlier.

XWHEP uses data to manage :

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

XWHEP : datas

Partial vue of the internal data structure

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

mandatory
calculated
optionnal
calculated if not set

If not set, XWHEP automatically manage the data.

XWHEP : applications

Partial vue of the internal application structure

uid,	
owneruid	the uid of the user who owns the data
accessrights	e.g. 0x755
name	the name of the file
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
libraryURI	the URI of the library, if any
binaryURI	the URI of the binary

mandatory
optionnal

this is provided to jobs by default. Jobs may overide this.

this is always provided to all jobs

this is provided to jobs by default. Jobs may overide this.

XWHEP : jobs

Partial vue of the internal job structure

uid,	
accessrights	e.g. 0x755
appuid	the UID of the application to run
useruid	the UID of the owner
expectedHost	the UID of the worker this job MUST run on
cmdLine	the command line
stdin	the URI of the stdin
dirin	the URI of the dirin provided to all jobs
result	the URI to store the result

mandatory
calculated
optionnal
calculated if not set

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

If not set, XWHEP automatically manage the result

XWHEP : workers