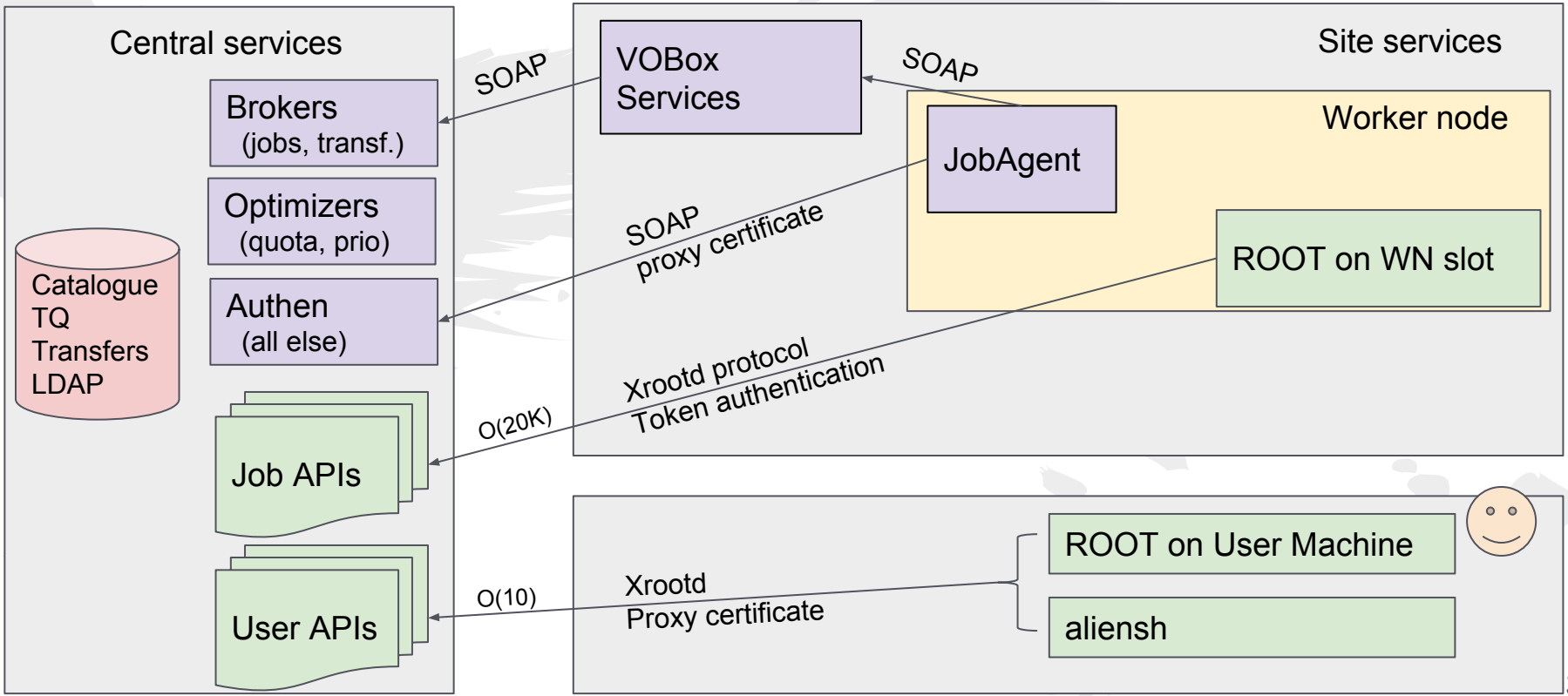
Migration steps

costin.grigoras@cern.ch

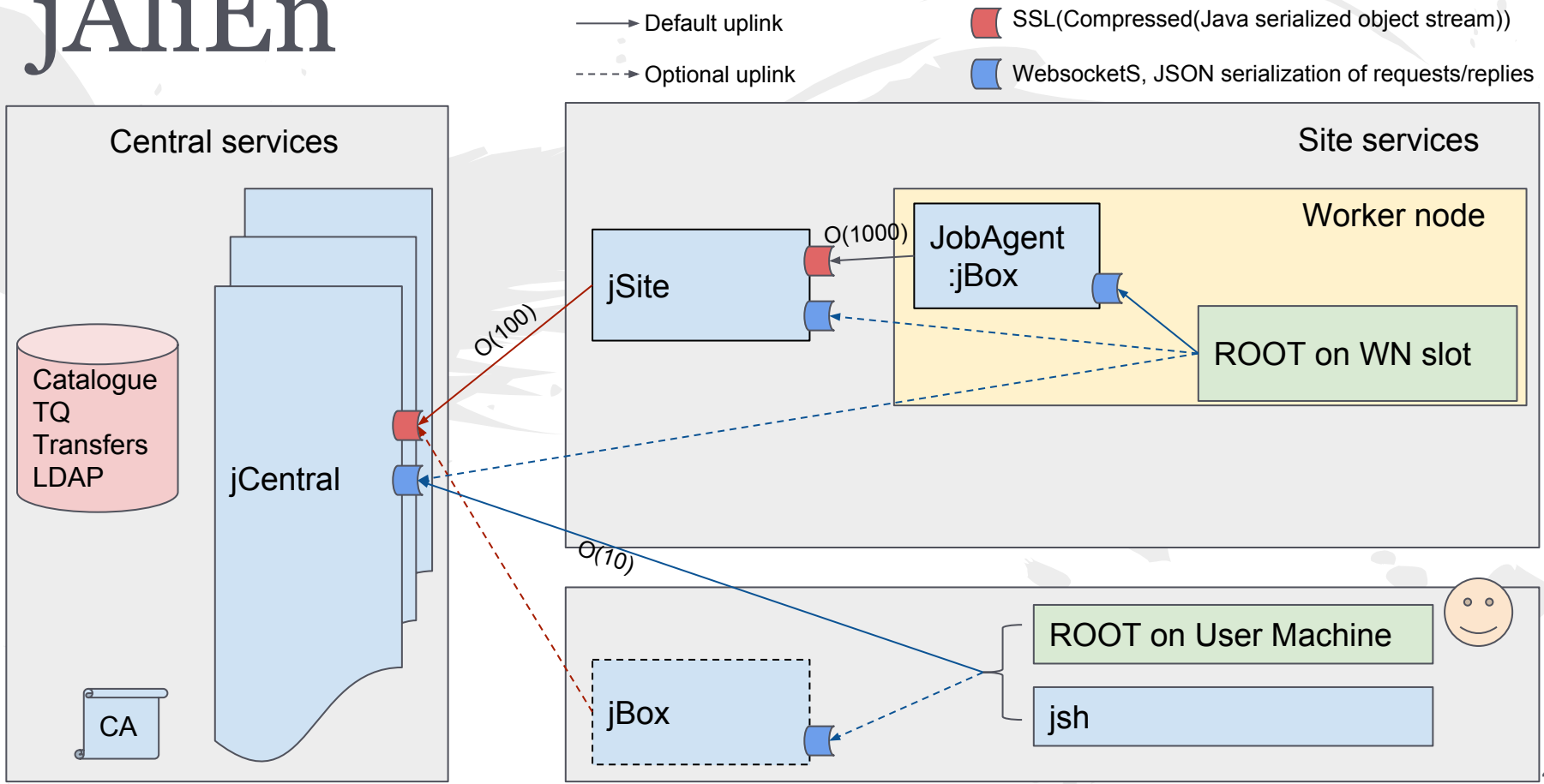
Migration steps

costin.grigoras@cern.ch

AliEn communication



jAliEn


Default uplink
Optional uplink
SSL(Compressed(Java serialized object stream))
WebsocketS, JSON serialization of requests/replies
Central services
Catalogue
TQ
Transfers
LDAP
jCentral
CA
O(100)
O(10)
Site services
jSite
O(1000)
Worker node
JobAgent
:jBox
ROOT on WN slot
jBox
ROOT on User Machine
jsh

Parallel central services

AliEn and jAliEn are running in parallel on the same set of central service nodes

alice-jcentral.cern.ch:8097

SSL WebSocket for everybody (jobs and users)

alice-jcentral.cern.ch:8098

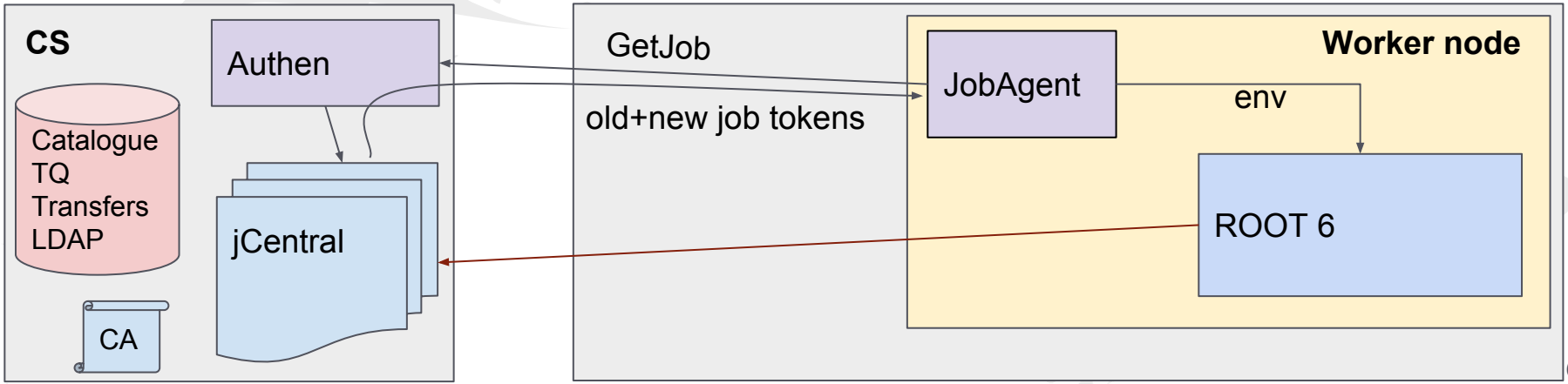
Java object channel + SSL

Please make sure your firewall allows outgoing connections to these ports

1st step: deprecate API srv.

Intermediate step on the existing infrastructure

Replacing the API services that are the weakest link at the moment



1st step: deprecate API srv.

Done:

Tokens are issued by jCentral

Existing brokers and JobAgent code modified to publish the new tokens in the job environment

In testing:

New plugin to be fully transparent to existing code

To do:

Packaging for ROOT 6 + TJAliEn

Simpler dependency chain

AliROOT (ROOT, TJAliEn)
ROOT (Xrootd 4+, OpenSSL 1+)
TJAliEn (ROOT, websockets, json)

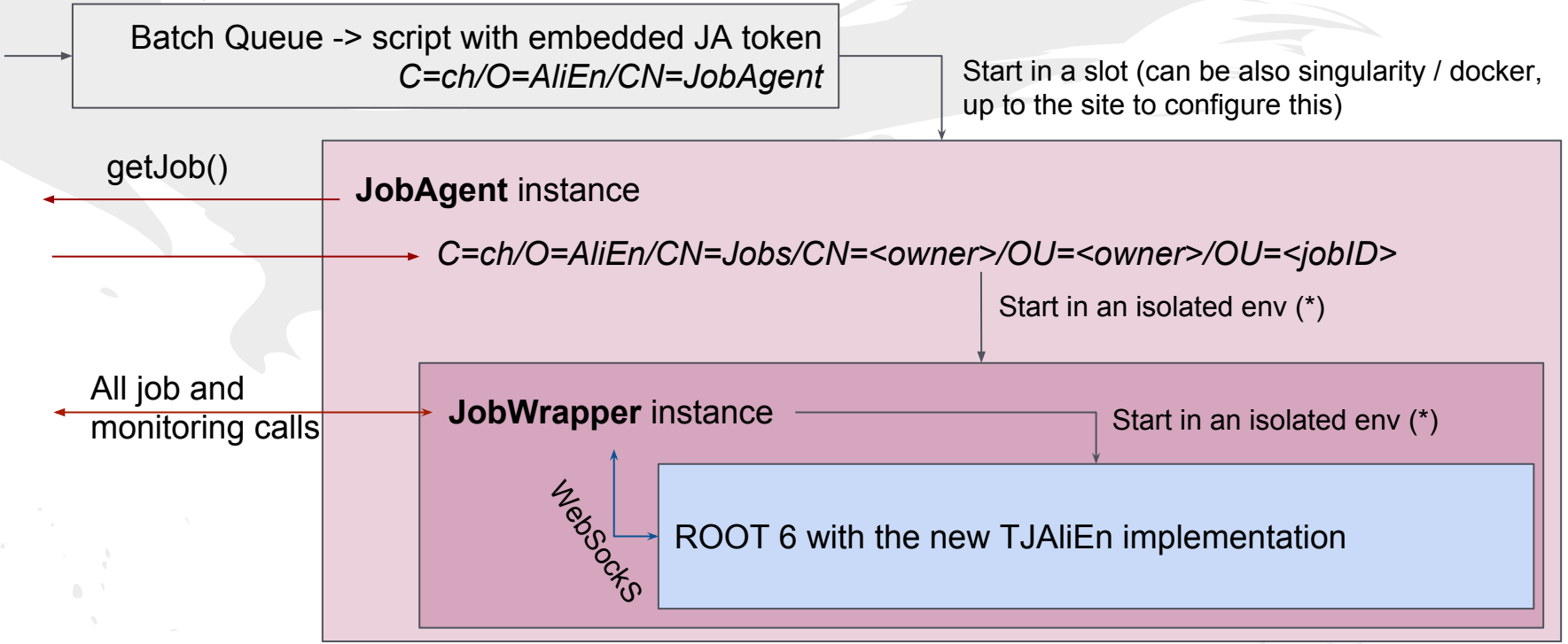
2nd step : JobAgent

Keep the VoBox services (CE in particular) in place

Modify just the submitted script to start the new JobAgent instead

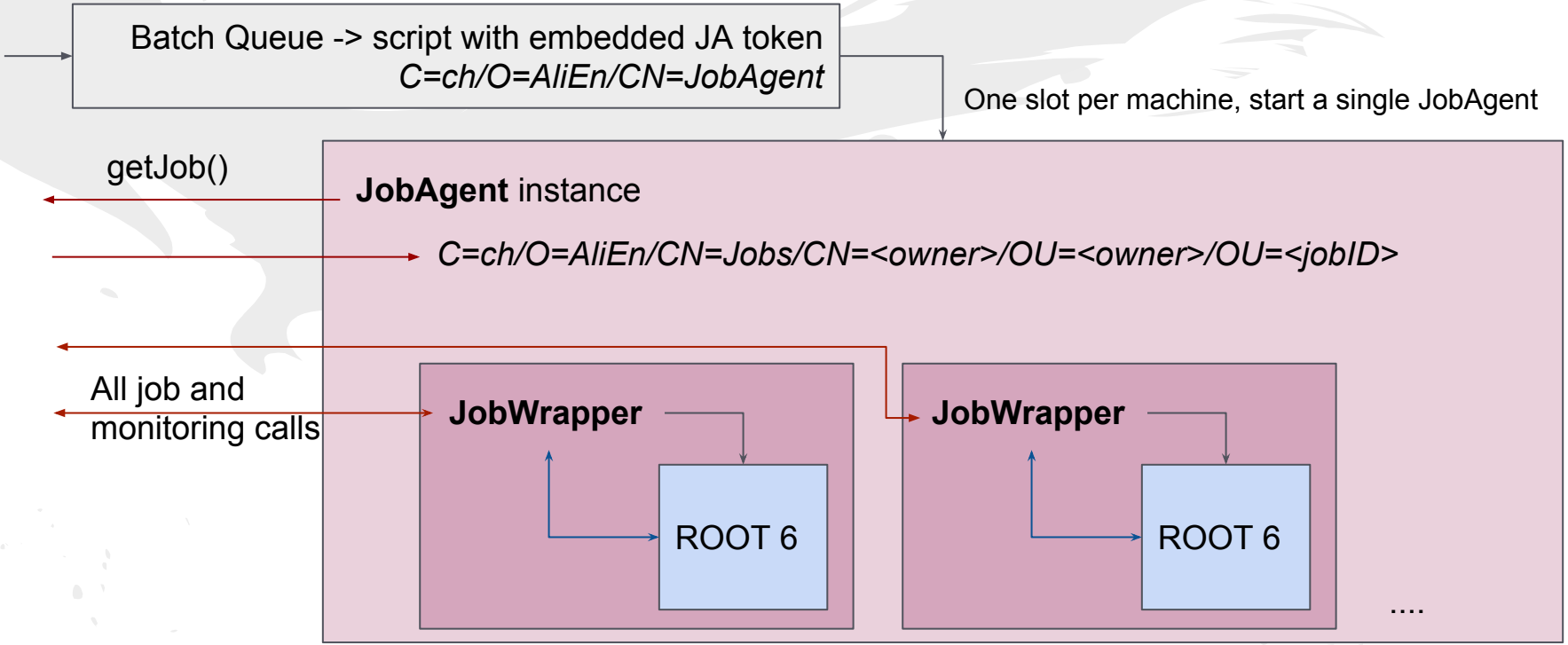
It should be transparent for everybody but we'll do it site by site and with advance notice

JobAgent split



(*) Can be a simple wrapper script or container or singularity… Depends on what the site supports

Whole node submission



(*) Can be a simple wrapper script or container or singularity… Depends on what the site supports

Whole node submission

Steered by the MJF configuration

$JOBFEATURES/allocated_CPU

Will advertise this value and match jobs with at most this many cores

Time limited by the JA token expiration and *$JOBFEATURES/shutdowntime_job*

MJF screenshots

Variable	Contents	Comments
MACHINEFEATURES	Path to a directory	Execution specific information
JOBFEATURES	Path to a directory	Job specific information
JOBSTATUS	Path to a directory	Internal job (pilot) information

The *$MACHINEFEATURES* and *$JOBFEATURES* directories contain files created by the resource provider.

The *$JOBSTATUS* directory is initially empty, and will be populated by the pilot job itself.

$JOBFEATURES

File Name (key)	Value	(Optional) Comments
cpufactor_lrms	Normalization factor as used by the batch system.	Can be site specific
cpu_limit_secs_lrms	CPU limit in seconds, normalized	Divide by **cpufactor_lrms** to retrieve the real time seconds. For multi-core jobs it's the total.
cpu_limit_secs	CPU limit in seconds, real time (not normalized)	For multi-core jobs it's the total.
wall_limit_secs_lrms	Run time limit in seconds, normalized	Divide by **cpufactor_lrms** to retrieve the real time seconds
wall_limit_secs	Run time limit in seconds, real time (not normalized)	
disk_limit_GB	Scratch space limit in GB (if any)	If no quotas are used on a shared system, this corresponds to the full scratch space available to all jobs which run on the host. Counting is 1GB = 1000MB = 1000^2kB
jobstart_secs	Unix time stamp (in seconds) of the time when the job started in the batch farm.	This is what the batch system sees, not when the user payload started to work.
mem_limit_MB	Memory limit (if any) in MB.	Total memory. Count with 1000 not 1024, that is 4GB corresponds to 4000
allocated_CPU	number of allocated cores to the current job	Allocated cores can be physical or logical
shutdowntime_job	dynamic value, shutdown time as a UNIX time stamp (in seconds)	optional, if the file is missing no job shutdown is foreseen. The job needs to have finished all its processing when the shutdowntime has arrived

$MACHINEFEATURES

File Name (key)	Value	(Optional) Comments
hs06	HS06 rating of the full machine in it's current setup	Static value. HS06 is measured following the HEPiX recommendations. If Hyperthreading is enabled, the additional cores are treats as if they were full cores
shutdowntime	dynamic value, shutdowntime as a UNIX time stamp (in seconds)	Dynamic. If the file is missing, no shutdown is foreseen. The value is in real time, and must be in the future. Must be removed if the shutdowntime has arrived
jobslots	Number of job slots for the host	dynamic value, can change with batch reconfigurations
phys_cores	number of physical cores	-
log_cores	number of logical cores	can be zero if hyperthreading is off
shutdown_command	path to a command on the machine	optional, only relevant for virtual machines. A command provided by the site which provides a hook for the user to properly destroy the virtual machine and unregister it

$JOBSTATUS

File Name (key)	Value	(Optional) Comments
used_CPU	Number of used cores by the job.	Must be locked before any of the other files in this section are either read or written to. Must be less or equal than allocated_CPU.
last_job_start	UNIX time (integer)	
first_exp_job_end	UNIX time (integer)	Good faith estimate
last_exp_job_end	UNIX time (integer)	Good faith estimate
last_max_job_end	UNIX time (integer)	Enforced limit
add_uncom_time	CPU seconds (integer)	
add_final_exp_waste	CPU seconds (integer)	Good faith estimate
can_postpone_last_job	string, either "True" or "False"	If the job decides to revert from "False" to "True", it should not update any of the other values for a significant amount of time.
priority_factor	Integer, higher is better	The semantics is user specific, and should not be used to compare jobs of different users.

What else would you like reported from the job agent? Log files? Job IDs?

3rd step : VoBox

Finally, address the VoBox as well

- Deploy the new connection multiplexing service
- Batch queueing interfaces

Packaging

Single *.jar* file of jAliEn + all deps (CS)

Externally we need (alienv / cvmfs):

- JRE (8+)
- Xrootd (cmd line interface)

Alternatively it can be packaged for

- Library only, to embed in other projects
- User = jAliEn + minimal deps to run the shell / JA

Old services retirement

Most central services can be retired after the upgrade

Apart from the job API services that are needed by the old sw packages

We will have to keep a few instances around (or start them on demand) if we want to run older binaries

To be continued …