gLite

Lightweight Middleware for Grid Computing

Job Description Language
JDL

Outline

- *Job Description Language (JDL)*
  - is the language used to describe a job. User has to describe his/her jobs and their requirements, and to retrieve the output when the jobs are finished.

Job Description Language

- The Job Description Language (JDL) is a high-level language based on the Classified Advertisement (ClassAd) language, used to describe jobs and aggregates of jobs with arbitrary dependency relations.
  - A job description is a file (called **JDL file**) consisting of lines having the format:

```
attribute = expression;
```

  - Expressions can consist of several lines, but only the last one must be terminated by a **semicolon**. Literal strings are enclosed in **double quotes**. If a string itself contains double quotes, they must be escaped with a backslash
    (e.g.: `Arguments = "\"hello\" 10";`).

Job Description Language

- The character “ ‘ ” cannot be used in a JDL file.
- Comments must be preceded by a sharp character (#) or a double slash (//) at the beginning of each line.
- Multi-line comments must be enclosed between “/*” and “*/” .

Attention! JDL is sensitive to blank characters and tabs. No blank characters or tabs should follow the semicolon at the end of a line.

Simple example

```
Type = "Job";
JobType = "Normal";
Executable = "myexe";
StdInput = "myinput.txt";
StdOutput = "message.txt";
StdError = "error.txt";
InputSandbox = {"myinput.txt",
  "/home/user/example/myexe"};
OutputSandbox = {"message.txt", "error.txt"};
```

Executable

```
Executable = "/bin/hostname";
```

The Executable attribute specifies the command to be run by the job. If the command is already present on the WN, it must be expressed as an absolute path; if it has to be copied from the UI, only the file name must be specified, and the path of the command on the UI should be given in the InputSandbox attribute.

```
Executable = "test.sh";
InputSandbox = {"/home/doe/test.sh"};
```

Arguments

- The Arguments attribute can contain a string value, which is taken as argument list for the executable:

```
Executable = "/bin/hostname";
Arguments = “-f";
```

- In the `Executable` and in the `Arguments` attributes it may be necessary to use special characters, such as &, \, |, >, <. These characters should be preceded by triple \ in the JDL, or specified inside quoted strings e.g.:

```
Arguments = "-f file1\\\&file2";
```

StdOutput and StdError

- The attributes StdOutput and StdError define the names of the files containing the standard output and standard error of the executable, once the job output is retrieved.

```
StdOutput = "std.out";
StdError = "std.err";
```

InputSandbox and OutSandbox

- If files have to be copied from the UI to the execution node, they must be listed in the InputSandbox attribute:

  ```
  InputSandbox = {"test.sh", .. , "fileN"};
  ```

- The files to be transferred back to the UI after the job is finished can be specified using the `OutputSandbox` attribute:

  ```
  OutputSandbox = {"std.out","std.err"};
  ```

- The InputSandbox cannot contain two files with the same name, even if they have a different absolute path, as when transferred they would overwrite each other.

Enviroment and Virtual Organisation

- The shell environment of the job can be modified using the `Environment` attribute.

```
Environment = {"CMS_PATH=$HOME/cms",
"CMS_DB=$CMS_PATH/cmdb"};
```

- The `VirtualOrganisation` attribute can be used to explicitly specify the VO of the user:

```
VirtualOrganisation = “gilda";
```

JobType

`JobType`

- *Normal* (simple, sequential job), *Interactive*, *MPI*, *Checkpointable*, *Partitionable*, *Parametric*

- Or combination of them
  - Checkpointable, Interactive
  - Checkpointable, MPI

```
JobType = “Interactive”;
JobType =
{“Interactive”,”Checkpointable”};
```

“Interactive” + “MPI” not yet permitted

Requirements

- The `Requirements` attribute can be used to express constraints on the resources where the job should run.
  - **Its value is a Boolean expression that must evaluate to true for a job to run on that specific CE.**

- Note: *Only one Requirements attribute can be specified* (if there is more than one, only the last one is considered). If several conditions must be applied to the job, then they all must be combined in a single Requirements attribute.

- For example, let us suppose that the user wants to run on a CE using PBS as batch system, and whose WNs have at least two CPUs:

```
Requirements = other.GlueCEInfoLRMSType == "PBS" &&
   other.GlueCEInfoTotalCPUs > 1;
```

- The WMS can be also asked to send a job to a particular queue in a CE with the following expression:

```
Requirements = other.GlueCEUniqueID ==
   "lxshare0286.cern.ch:2119/jobmanager-pbs-short";
```

- It is also possible to use regular expressions when expressing a requirement.
  - **Let us suppose for example that the user wants all his jobs to run on any CE in the domain cern.ch. This can be achieved putting in the JDL file the following expression:**

```
Requirements = RegExp("cern.ch",other.GlueCEUniqueID);
```

- The opposite can be required by using:

```
Requirements = (!RegExp("cern.ch", other.GlueCEUniqueID));
```

RetryCount

- If the job duration is significant, it is strongly advised to put a requirement on the maximum CPU time, or the wallclock time (expressed in minutes), needed for the job to complete.
  - **For example, to express the fact that the job needs at least 8 CPU hours and 20 wallclock hours:**

```
Requirements = other.GlueCEPolicyMaxCPUTime > 480 &&
   other.GlueCEPolicyMaxWallClockTime > 720;
```

- It is possible to have the WMS automatically resubmitting jobs which, for some reason, are aborted by the Grid. The user can limit the number of times the WMS should resubmit a job by using the JDL attribute `RetryCount`.

```
RetryCount = 7; or RetryCount = 0;
```

MyProxyServer

- The `proxy renewal feature` of the WMS is automatically enabled, as long as the user has stored a long term proxy in the default MyProxy server (usually defined in the MYPROXY SERVER) environment variable. However it is possible to indicate to the WMS a different MyProxy server in the JDL file:

```
MyProxyServer = “grid001.ct.infn.it";
```

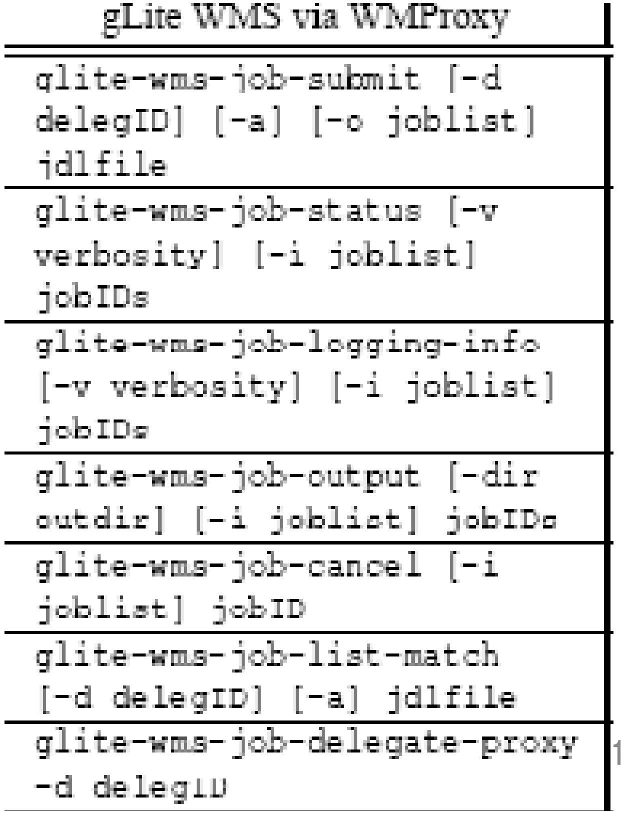
Rank

- The choice of the CE where to execute the job, among all the ones satisfying the requirements, is based on the *Rank* of the CE, a quantity expressed as a floating-point number. The CE with the highest rank is the one selected.
  - **By default, the rank is equal to `other.GlueCEStateEstimatedResponseTime`, where the estimated response time is an estimation of the time interval between the job submission and the beginning of the job execution.**
  - `Rank = other.GlueCEStateFreeCPUs;` **which will rank best the CE with the most free CPUs.**

The Command Line Interface

- The gLite WMS implements two different services to manage jobs: the *Network Server* and the *WMProxy*.
  - The recommended method to manage jobs is through the gLite WMS via WMProxy, because it gives the best performance and allows to use the most advanced functionalities

- The WMProxy implements several functionalities, among which:
  - **submission of job collections;**
  - **faster authentication;**
  - **faster matchmaking;**
  - **faster response time for users;**
  - **higher job throughput.**



gLite WMS via WMProxy
glite-wms-job-submit [-d delegID] [-a] [-o joblist] jdlfile
glite-wms-job-status [-v verbosity] [-i joblist] jobIDs
glite-wms-job-logging-info [-v verbosity] [-i joblist] jobIDs
glite-wms-job-output [-dir outdir] [-i joblist] jobIDs
glite-wms-job-cancel [-i joblist] jobID
glite-wms-job-list-match [-d delegID] [-a] jdlfile
glite-wms-job-delegate-proxy -d delegID

Delegating a proxy to WMProxy

- Each job submitted to WMProxy must be associated to a proxy credential previously delegated by the owner of the job to the WMProxy server.
  - **This proxy is then used any time the WMProxy needs to interact with other services for job related operations (e.g. submission to the CE, a GridFTP file transfer etc.)**
  - **There are two possible mechanisms to ask for a delegation of the user credentails:**
    - **asking the "automatic" delegation of the credentials during the submission operation**
    - **asking for an "explicit" delegation**

Delegating a proxy to WMProxy

- To explicitly delegate a user proxy to WMProxy, the command to use is:
  `glite-wms-job-delegate-proxy -d <delegID>`
  where `<delegID>` is a string chosen by the user.

For example, to delegate a proxy:

```
$ glite-wms-job-delegate-proxy -d mydelegID
Connecting to the service
https://rb102.cern.ch:7443/glite_wms_wmproxy_server
======= glite-wms-job-delegate-proxy Success ========
Your proxy has been successfully delegated to the
  WMProxy:
https://rb102.cern.ch:7443/glite_wms_wmproxy_server
with the delegation identifier: mydelegID
====================================================
```

Submitting a simple job

- Starting from a simpl it it via WMProxy by doir

glite-wms-job-submit –a test.jdl

For the automatic delegation

```
$ glite-wms-job-submit -d mydelegID test.jdl
Connecting to the service
https://rb102.cern.ch:7443/glite_wms_wmproxy_serve
  r


======== glite-wms-job-submit Success ========
The job has been successfully submitted to the
  WMProxy
Your job identifier is:
https://rb102.cern.ch:9000/vZKKk3gdBla6RySximq_vQ
==============================================
```

Submitting a simple job

- The command returns to the user the `job identifier` (*`jobID`*), which uniquely defines the job and can be used to perform further operations on the job, like interrogating the system about its status, or canceling it.

- The format of the jobID is:

`https://<LB_hostname>[:<port>]/<unique_string>`

- where `<unique string>` is guaranteed to be unique and `<LB hostname>` is the host name of the Logging and Bookkeeping (LB) server for the job, which usually sits on the WMS used to submit the job.

Troubleshooting

- To submit jobs via WMProxy, it is required to have a valid VOMS proxy, otherwise the submission will fail with an error like the following:

```
Error - Operation failed
Unable to delegate the credential to the
  endpoint:
https://rb102.cern.ch:7443/glite_wms_wmproxy_server
User not authorized:
unable to check credential permission
   (/opt/glite/etc/glite_wms_wmproxy.gacl)
   (credential entry not found)
  credential type: person
  input dn: /C=CH/O=CERN/OU=GRID/CN=John Doe
```

glite-wms-job-submit Options

- The `-o <file path>` option allows users to specify a file to which the jobID of the submitted job will be appended. This file can be given to other job management commands to perform operations on more than one job with a single command, and it is a convenient way to keep trace of one's jobs.

```
$ glite-wms-job-submit -d mydelegID -o jobid test.jdl
```

- The `-r <CEId>` option is used to directly send a job to a particular CE. If used, the match making will not be carried out.
  - **The drawback is that the *BrokerInfo* file, which provides information about the evolution of the job, will not be created, and therefore the use of this option is discouraged.**

```
$ glite-wms-job-submit -d mydelegID -r <CEId> test.jdl
```

Computing Element Id (match-making)

- A CE is identified by <CEId>, which is a string with the following format:

```
<CE hostname>:<port>/jobmanager-<service>-<queue>
<CE hostname>:<port>/blah-<service>-<queue>
```

- where `<CE hostname>` and `<port>` are the host name of the machine and the port where the Grid Gate is running
  - **(the Globus Gatekeeper for the LCG CE and CondorC+BLAH for the gLite CE)**
- `<queue>` is the name of one of the corresponding LRMS queue
- `<service>` is the LRMS type, such as lsf, pbs, condor.

- E.g.:

```
adc0015.cern.ch:2119/jobmanager-lcgpbs-infinite
prep-ce-01.pd.infn.it:2119/blah-lsf-atlas
```

Listing CE(s) that matching a job

- It is possible to see which CEs are useful to run a job described by a given JDL using:

```
$ glite-wms-job-list-match -d mydelegID --rank
  test.jdl
Connecting to the service
https://rb102.cern.ch:7443/glite_wms_wmproxy_server
====================================================
COMPUTING ELEMENT IDs LIST
The following CE(s) matching your job requirements
  have been found:

*CEId*
      *Rank*
- CE.pakgrid.org.pk:2119/jobmanager-lcgpbs-cms 0
- grid-ce0.desy.de:2119/jobmanager-lcgpbs-cms -10
- gw-2.ccc.ucl.ac.uk:2119/jobmanager-sge-default -56
- grid-ce2.desy.de:2119/jobmanager-lcgpbs-cms -107
====================================================
```

Retrieving the status of a job

```
$ glite-wms-job-status
   https://rb102.cern.ch:9000/fNdD4FW_Xxkt2s2aZJeoeg
************************************************************
   *
BOOKKEEPING INFORMATION:
Status info for the Job :
   https://rb102.cern.ch:9000/fNdD4FW_Xxkt2s2aZJeoeg
Current Status: Done (Success)
Exit code: 0
Status Reason: Job terminated successfully
Destination: ce1.inrne.bas.bg:2119/jobmanager-lcgpbs-cms
Submitted: Mon Dec 4 15:05:43 2006 CET
***********************************************************
```

- The *verbosity* level controls the amount of information provided. The value of the -v option ranges from **0** to **3**.

- The commands to get the job status can have several jobIDs as arguments, or you can use the `–i <file path>` option:

```
                glite-wms-job-status –i jobid
```

Cancelling a job

```
glite-wms-job-cancel
   https://rb102.cern.ch:9000/P1c60RFsrIZ9mnBALa7yZA
Are you sure you want to remove specified job(s)
   [y/n]y : y
Connecting to the service
https://128.142.160.93:7443/glite_wms_wmproxy_server
========== glite-wms-job-cancel Success ============
The cancellation request has been successfully
   submitted for the following job(s):
- https://rb102.cern.ch:9000/P1c60RFsrIZ9mnBALa7yZA
====================================================
```

If the cancellation is successful, the job will terminate in status CANCELLED

Retrieving the output(s)

```
$ glite-wms-job-output
   https://rb102.cern.ch:9000/yabp72aERhofLA6W2-LrJw
Connecting to the service
https://128.142.160.93:7443/glite_wms_wmproxy_server
=====================================================
JOB GET OUTPUT OUTCOME
Output sandbox files for the job:
https://rb102.cern.ch:9000/yabp72aERhofLA6W2-LrJw
have been successfully retrieved and stored in the
   directory:
/tmp/doe_yabp72aERhofLA6W2-LrJw
====================================================
```

- The default location for storing the outputs (normally `/tmp`) is defined in the UI configuration, but it is possible to specify in which directory to save the output using the `--dir <path name>` option.

```
glite-wms-job-output -i jobId -dir /path
```

‘Scattered’ Input Sandboxes

- A new feature introduced by the gLite WMS is the possibility to indicate input sandbox files stored not on the UI,but on a GridFTP server, and, similarly, to specify that output files should be transferred to a GridFTP server when the job finishes.

```
InputSandbox = {"gsiftp://lxb0707.cern.ch/cms/fileA",
   "fileB"};
```

- It is also possible to specify a base GridFTP URI with the attribute `InputSandboxBaseURI`
  - **files expressed as simple file names or as relative paths will be looked for under that base URI.**

```
InputSandbox = {"fileA", "data/fileB",
   "file:///home/doe/fileC"};

InputSandboxBaseURI =
   "gsiftp://lxb0707.cern.ch/cms/doe";
```

Storing output files in a GridFTP Server

- In order to store the output sandbox files to a GridFTP server, the `OutputSandboxDestURI` attribute must be used together with the usual OutputSandbox attribute.
  - **The latter is used to list the output files created by the job in the WN to be transferred.**
  - **The former is used to express where the output files are to be transferred.**

```
OutputSandbox = {"fileA", "data/fileB", "fileC"};

OutputSandboxDestURI =
   {"gsiftp://lxb0707.cern.ch/cms/doe/fileA",

   "gsiftp://lxb0707.cern.ch/cms/doe/fileB","fileC
   "};
```

  - **where the first two files have to be copied to a GridFTP server, while the third file will be copied back to the WMS with the usual mechanism. Clearly, glite-wms-job-output will retrieve only the third file.**

- Another possibility is to use the OutputSandboxBaseDestURI attribute to specify a base URI on a GridFTP server where the files listed in OutputSandbox will be copied.

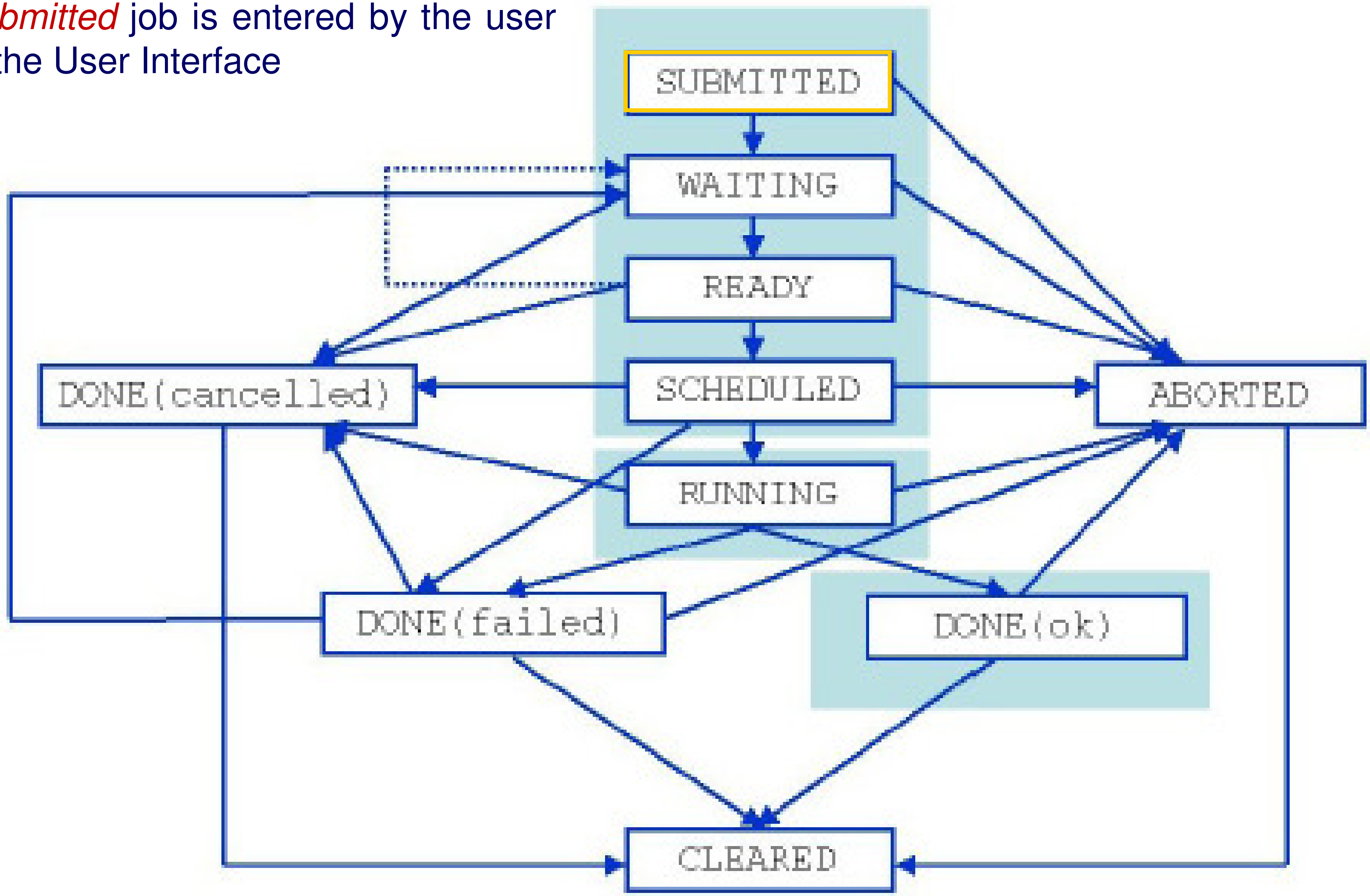
```
OutputSandbox = {"fileA", "fileB"};

OutputSandboxBaseDestURI =
  "gsiftp://lxb0707.cern.ch/cms/doe/";
```

will copy both files under the specified GridFTP URI.

- **Note:** the directory on the GridFTP where the files have to be copied must already exist.

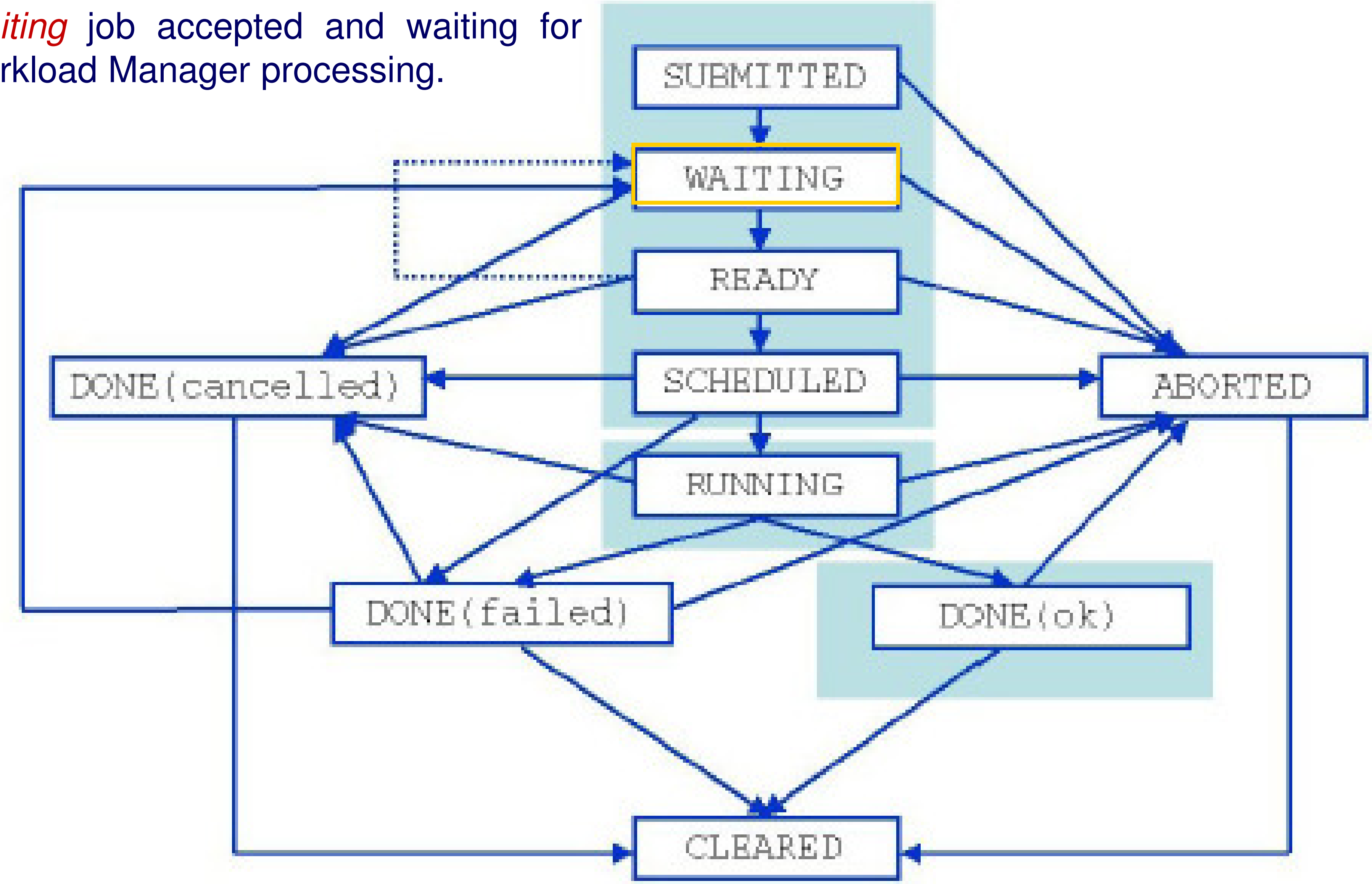
Jobs State Machine (1/9)

Submitted job is entered by the user to the User Interface

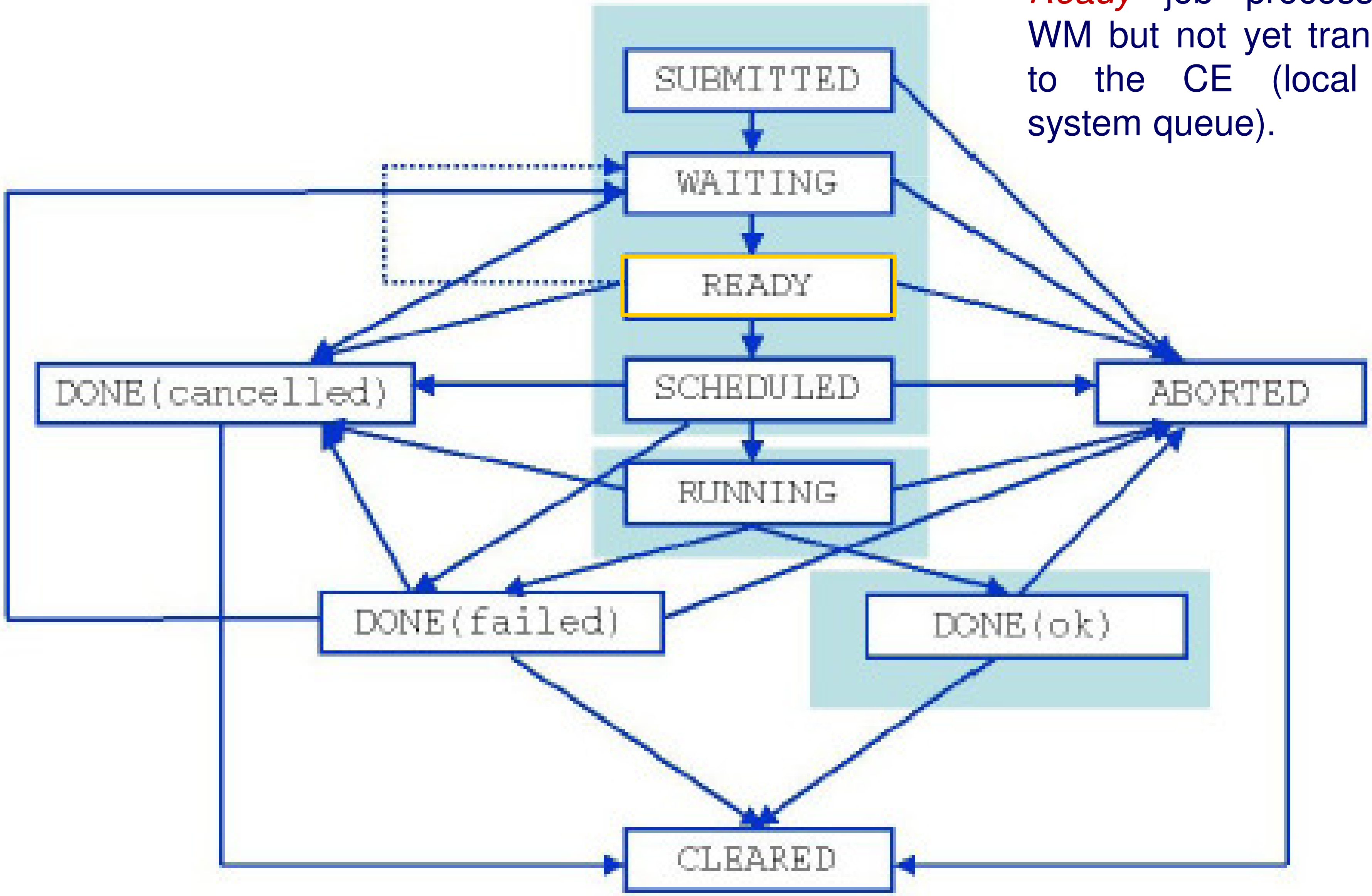


Jobs State Machine (2/9)

Waiting job accepted and waiting for Workload Manager processing.

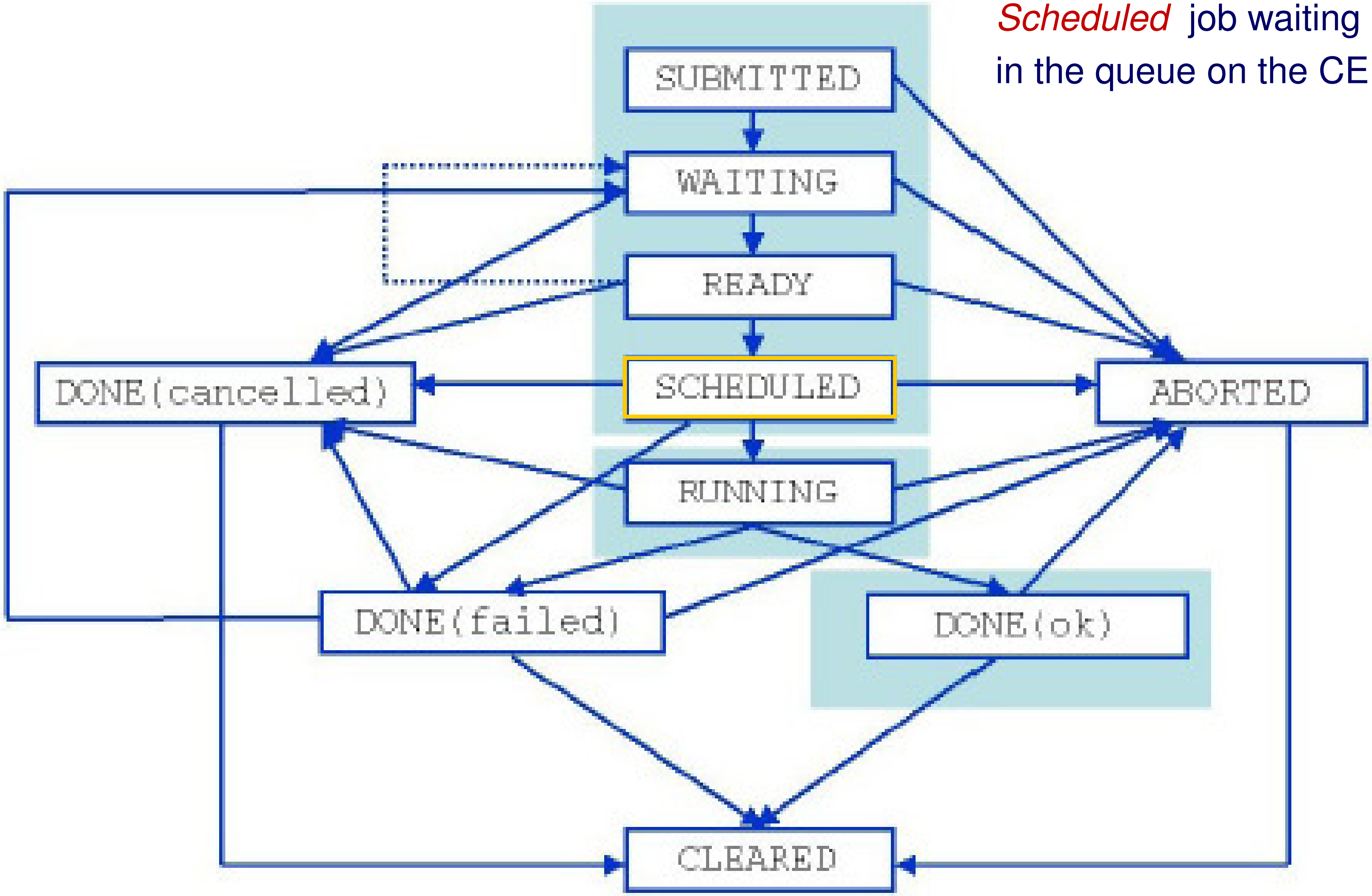


Jobs State Machine (3/9)



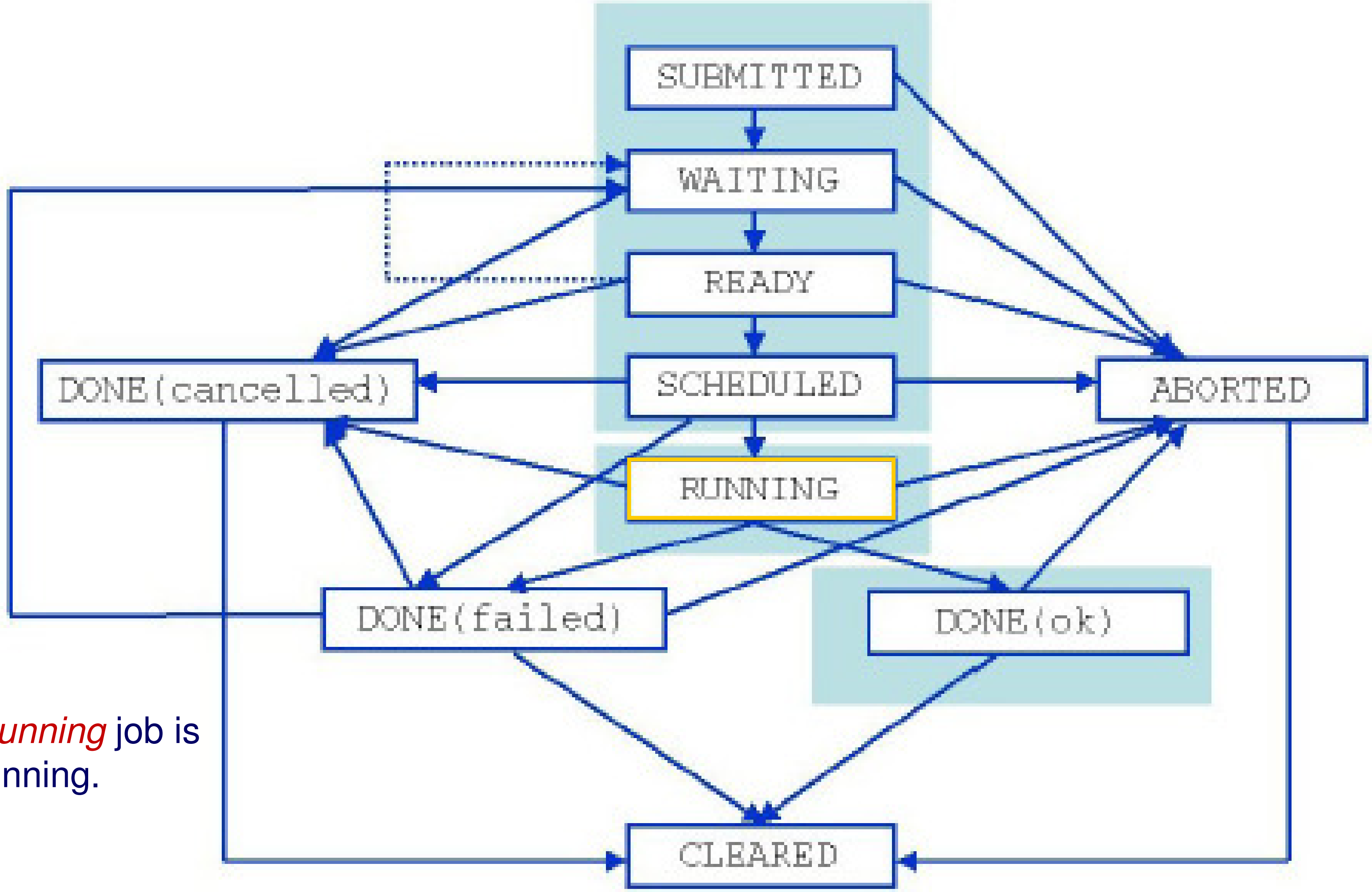
Ready job processed by WM but not yet transferred to the CE (local batch system queue).

Jobs State Machine (4/9)



Scheduled job waiting in the queue on the CE.

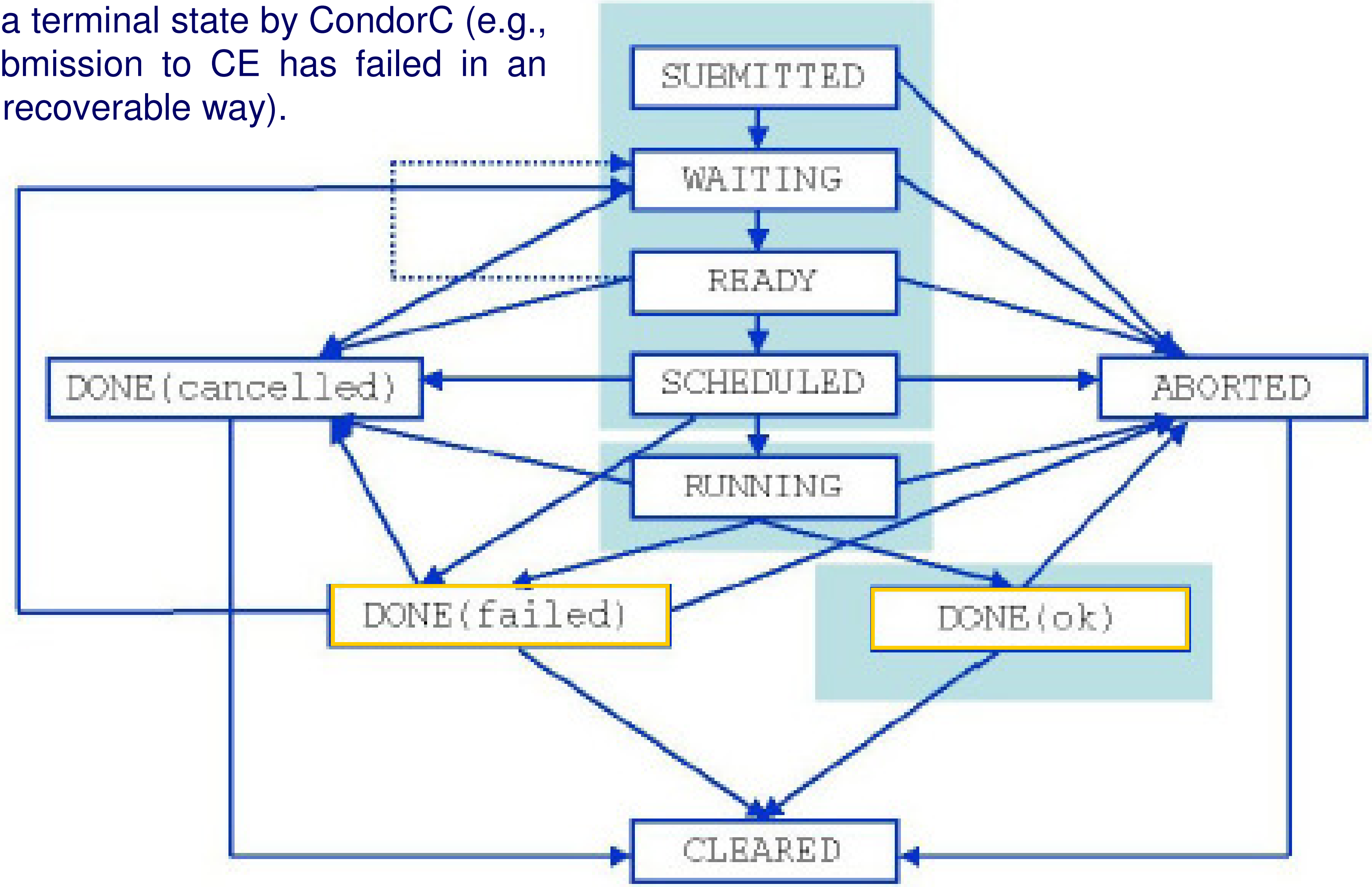
Jobs State Machine (5/9)



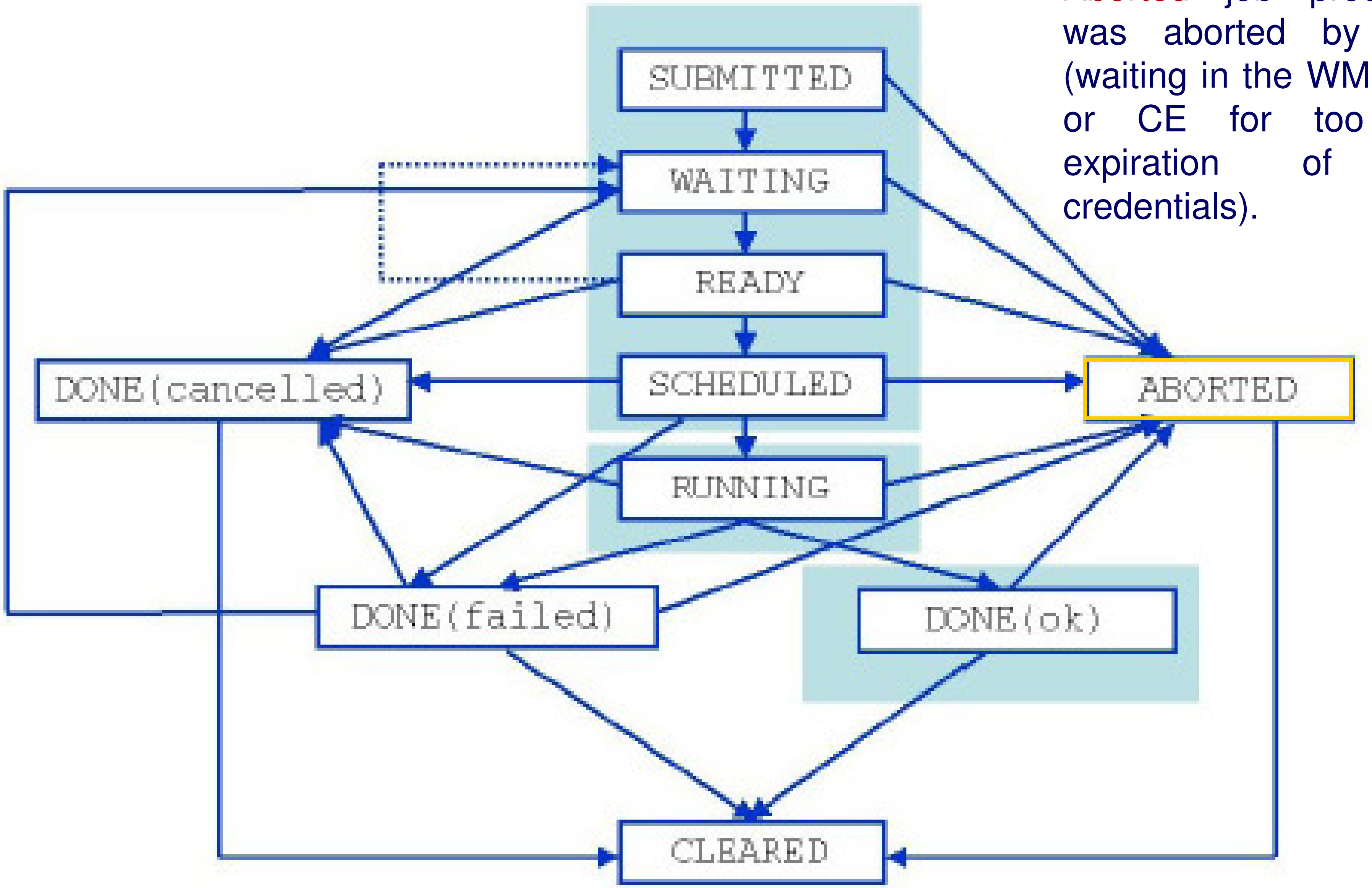
Running job is running.

Jobs State Machine (6/9)

Done job exited or considered to be in a terminal state by CondorC (e.g., submission to CE has failed in an unrecoverable way).



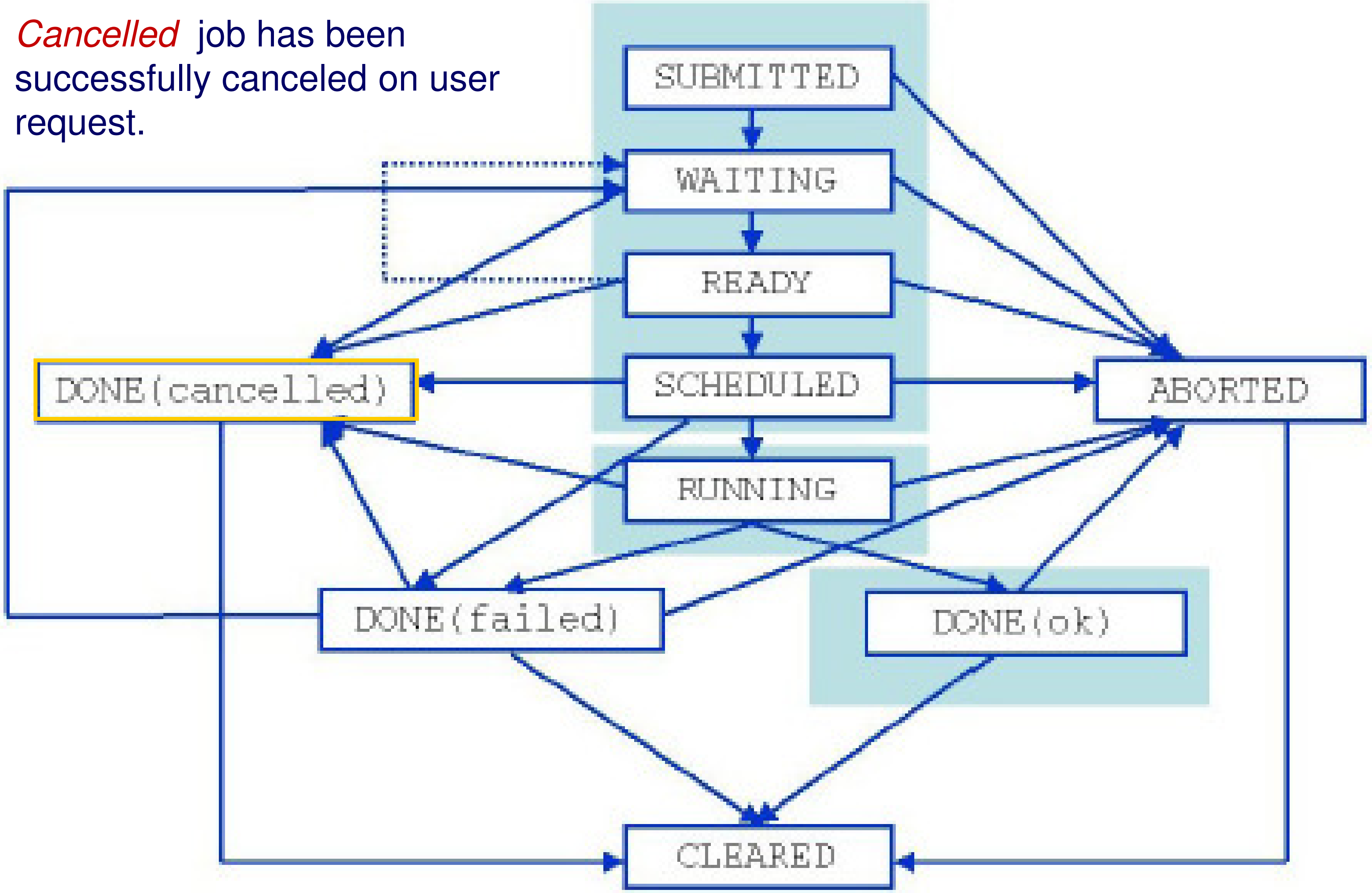
Jobs State Machine (7/9)



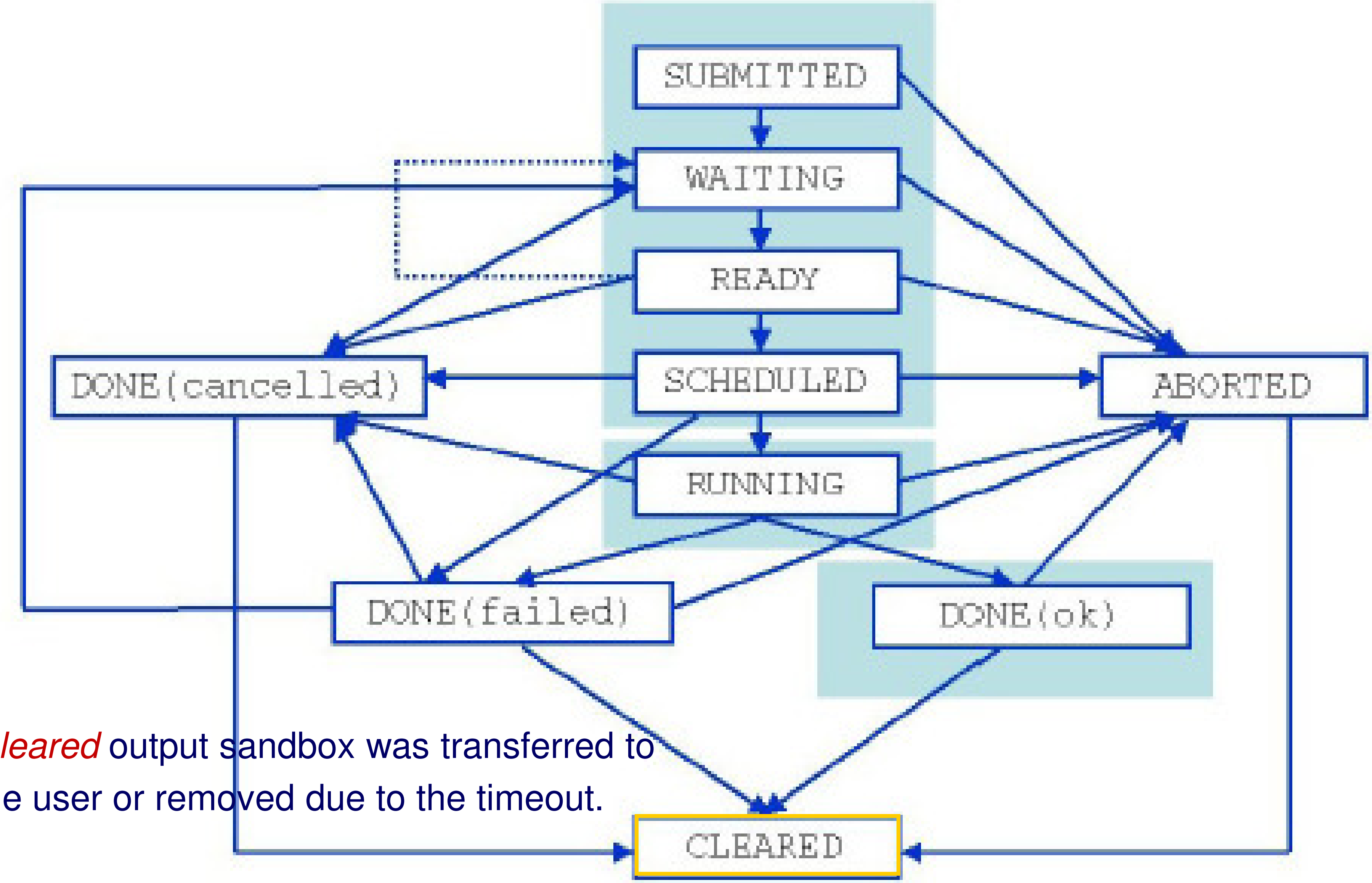
Aborted job processing was aborted by WMS (waiting in the WM queue or CE for too long, expiration of user credentials).

Jobs State Machine (8/9)

Cancelled job has been successfully canceled on user request.

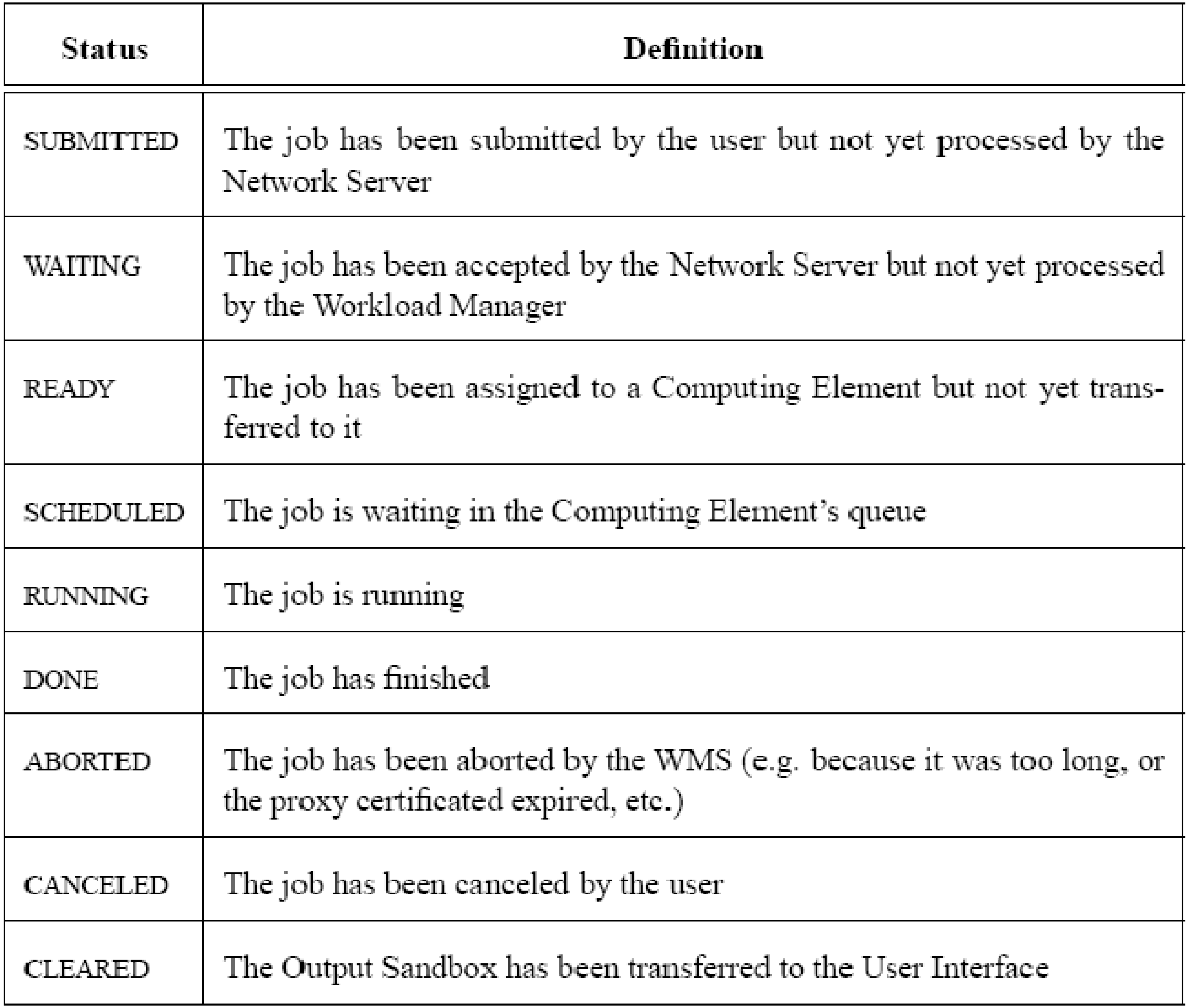


Jobs State Machine (9/9)



Cleared output sandbox was transferred to the user or removed due to the timeout.

..an useful reminder



Status	Definition
SUBMITTED	The job has been submitted by the user but not yet processed by the Network Server
WAITING	The job has been accepted by the Network Server but not yet processed by the Workload Manager
READY	The job has been assigned to a Computing Element but not yet transferred to it
SCHEDULED	The job is waiting in the Computing Element's queue
RUNNING	The job is running
DONE	The job has finished
ABORTED	The job has been aborted by the WMS (e.g. because it was too long, or the proxy certificated expired, etc.)
CANCELED	The job has been canceled by the user
CLEARED	The Output Sandbox has been transferred to the User Interface

References

- https://edms.cern.ch/file/674643/1/EGEE-JRA1-TEC-674643-WMPROXY-guide-v0-3.pdf
- https://edms.cern.ch/file/590869/1/EGEE-JRA1-TEC-590869-JDL-Attributes-v0-9.pdf
- https://edms.cern.ch/file/722398/1.2/gLite-3-UserGuide.pdf

Hands-on

- https://grid.ct.infn.it/twiki/bin/view/GILDA/SimpleJobSubmission
- https://grid.ct.infn.it/twiki/bin/view/GILDA/MoreOnJDL
- https://grid.ct.infn.it/twiki/bin/view/GILDA/WmProxyUse