



gLite Job Management (Básico)

Prof. Yudith Cardinale <yudith@ldc.usb.ve>

Prof. Jesus De Oliveira <jdeoliveira@ldc.usb.ve>



- EL **Workload Management System (WMS)** es el componente gLite que permite a los usuarios enviar **jobs** y realiza todas las tareas necesarias para ejecutarlos, ocultando la complejidad del grid al usuario
 - Es responsabilidad del usuario **describir** sus jobs y sus requerimientos, y de obtener sus resultados una vez sean completados
- En esta presentación se describirán los conceptos básicos del lenguaje empleado para describir un job, y las interfaces de línea de comandos básicas para enviar y gestionar jobs simples



Workload Management System

Workload Management System (WMS) comprende un conjunto de componentes del middleware de grid, responsables de la distribución y gestión de tareas a través de los recursos del grid.

Su propósito es **aceptar** y **satisfacer** requerimientos para gestión de trabajos que vienen desde clientes

El WMS enviará el job a un CE apropiado para su ejecución, considerando los requerimientos y preferencias del usuario expresados en la **descripción del job**

La decisión sobre que recurso debe ser utilizado, es el resultado del proceso de **matchmaking**



Job Description Language

- El **Job Description Language** (JDL) es un lenguaje de alto nivel basado en el lenguaje Classified Advertisement (ClassAd), usado para describir jobs a conjuntos de jobs con relaciones de dependencias arbitrarias
 - En gLite, el lenguaje JDL es usado para especificar las características y limitaciones de los jobs, que son tomadas en cuenta por el WMS para seleccionar el mejor recurso para ejecutar el job
 - Un archivo JDL consiste en líneas con el siguiente formato:
atributo = expresion;
 - Las expresiones pueden tener varias líneas, pero la última siempre debe ser terminada con un **punto y coma (;)**. Cadenas literales deben ser encerradas entre **comillas dobles (")**. Si dentro de una cadena deben incluirse comillas dobles, estas deberán ser escapadas con un **símbolo backslash (\)**.
Ejemplo: **Arguments = "\hello\" 10";**



Job Description Language

- El caracter “ ‘ ” no puede ser usado en el JDL
- Los comentarios se preceden con un caracter numeral (#) o doble barra (//) al principio de cada línea
- Comentarios de múltiples líneas deben ser encerrados entre símbolos “/*” y “*/” .

Importante! El JDL es sensible a espacios en blanco y tabuladores. No debe existir ningún espacio en blanco o tabuladores al final de una línea



Ejemplo Simple

```
Executable = "/bin/hostname";  
StdOutput = "std.out";  
StdError = "std.err";
```

El atributo **Executable** especifica el comando a ser ejecutado por el job. Si el comando existe previamente en el WN, debe ser especificado con el **path absoluto**; si ha de ser copiado desde el UI, únicamente el nombre del archivo debe ser especificado, y el path al ejecutable debe ser indicado en el atributo **InputSandbox**.

```
Executable = "test.sh";  
InputSandbox = {"/home/doe/test.sh"};  
StdOutput = "std.out";  
StdError = "std.err";
```



- EL atributo **Arguments** puede contener una cadena, que es pasada como una lista de argumentos al ejecutable:
Arguments = "fileA 10";
- En los atributos **Executable** y **Arguments** puede ser necesario utilizar caracteres especiales, como **&**, ****, **|**, **>**, **<**. Esos caracteres deben ser precedidos por backslash-triples (****): **Arguments = "-f file1\\\&file2";**
- Los atributos **StdOutput** y **StdError** definen el nombre de archivos que contendrán la salida estándar y la salida de error del ejecutable, una vez que los resultados del job sean obtenidos.



- Si se requiere copiar archivos desde el UI al nodo de ejecución, deben listarse en el atributo **InputSandbox**:
InputSandbox = {"test.sh", .. , "fileN"};
- Los archivos que se deseen copiar "de vuelta" al UI una vez que el job finalice, pueden especificarse en el atributo **OutputSandbox**:
OutputSandbox = {"std.out", "std.err"};
- El InputSandbox *no puede* contener dos archivos con el mismo nombre, *aun cuando tengan paths absolutos distintos*, dado que al transferirlos, uno sobrescribiría al otro.



- El ambiente del shell para el job puede ser modificado con el atributo **Environment**

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

- El atributo **VirtualOrganisation** puede utilizarse para especificar explícitamente la VO del usuario:

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



- **JobType**

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

- Permite especificar combinaciones:

- Checkpointable, Interactive
- Checkpointable, MPI

JobType = "Interactive";

JobType =

{ "Interactive", "Checkpointable" };

"Interactive" + "MPI" not yet permitted



- El atributo **Requirements** puede emplearse para expresar restricciones sobre los recursos en los que puede ejecutarse el job
 - Su valor es una expresión Boolean que debe evaluarse a "verdadero" si el job puede ejecutarse en un CE en particular

Nota: Solo debe especificarse un (1) atributo Requirements (Si hay más de uno, solo el último es considerado). Si se deben aplicar varias condiciones al job, todas deben combinarse en un único atributo Requirements.

- Por ejemplo, supongamos que el usuario desea ejecutar en un CE que tenga PBS como manejador de colas, y cuyos WNs tengan al menos 2 CPUs:

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



- También se puede obligar al WMS a que envíe un job a una cola en determinada en un CE con la siguiente expresión:

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

- También es posible utilizar *expresiones regulares* para especificar un requerimiento:
 - Supongamos que el usuario desea que sus jobs ejecuten en cualquier CE en el dominio cern.ch Esto puede lograrse con la siguiente expresión en el JDL:

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

- Lo opuesto puede expresarse con lo siguiente:

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



- Si la duración del job es significativa, es importante incluir un requerimiento que indique el tiempo máximo de CPU, o tiempo máximo de wallclock (expresado en minutos) requeridos para que el job pueda terminar:
 - Para expresar que el job necesesita **al menos 8 horas de CPU hours y 20 horas de wallclock**:

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

- Es posible solicitar al WMS que reenvie jobs automáticamente cuando sean abortados por alguna razón. El usuario puede limitar el número de veces que el WMS reenviara un job con el atributo **RetryCount**.

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



- La **renovación de proxies** del WMS esta activa automáticamente, siempre que el usuario haya generado y almacenado un certificado proxy de larga duración en el servidor MyProxy predeterminado (normalmente definido en la variable de ambiente **MYPROXY SERVER**). Tambien es posible solicitar el uso de un servicio MyProxy diferente, con el atributo MyProxyServer:

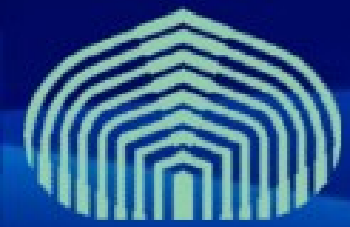
```
MyProxyServer = "grid001.ct.infn.it";
```



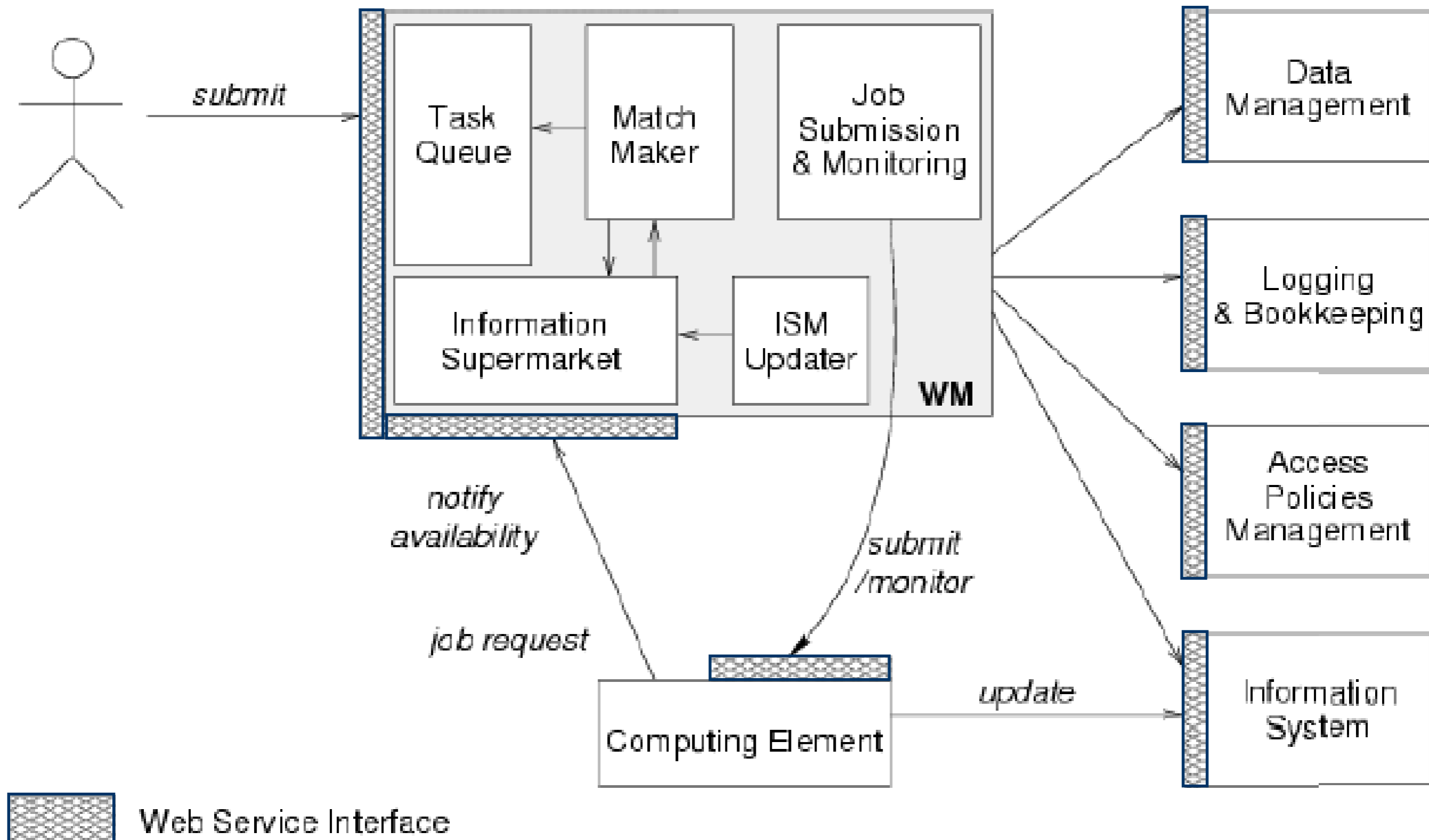
- La elección del CE para ejecutar el job, del grupo que cumpla con los requerimientos (atributo Requirements), esta basado en el **rango** de cada CE, una cantidad expresada como un número de punto flotante. El CE con el mayor **rango** es el seleccionado
 - **Por defecto, el rango es igual a `other .GlueCEStateEstimatedResponseTime`, que representa una estimación del tiempo que transcurrirá desde que el job es enviado hasta que comienza su ejecución**
 - **`Rank = other .GlueCEStateFreeCPUs`;**
Esta expresión le dará el rango mas alto a aquel CE con el mayor número de CPUs disponibles.

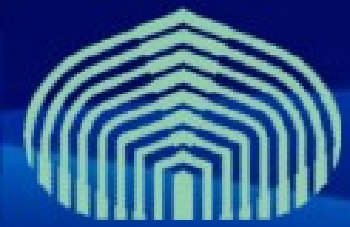


- El servicio WMPProxy es el responsable de proveer acceso a las funcionalidades del WMS a través de una interfaz basada en web-services (SOAP)
- Puede accederse directamente a través de la interfaz WSDL, usando la interfaz de línea de comandos de C++, o el API
- Ha sido diseñado para manejar eficientemente un gran número de solicitudes
 - Provee características y funcionalidades adicionales, como envío por lotes (*bulk submission*) y soporte para sandboxes compartidos y comprimidos para jobs compuestos

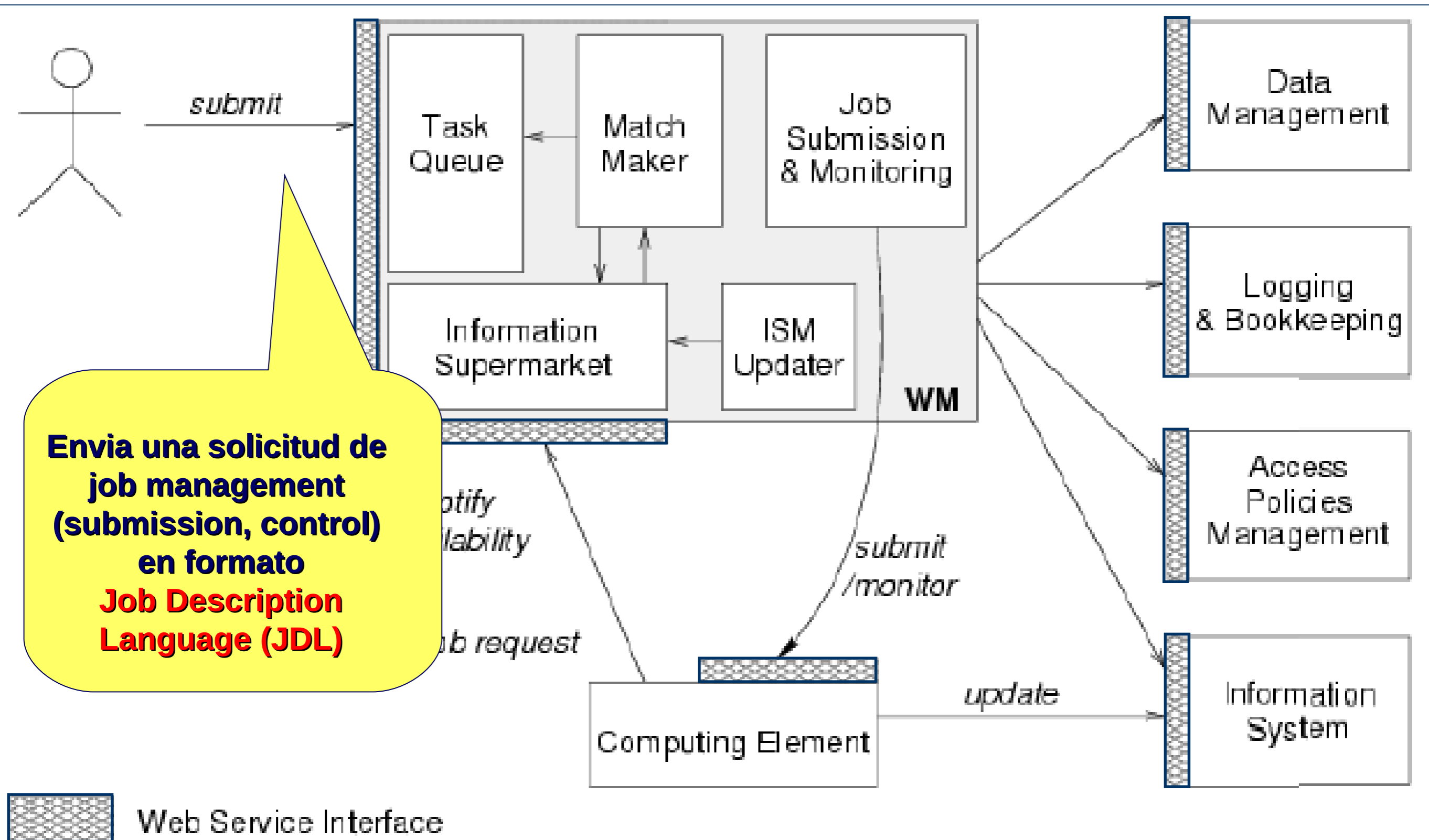


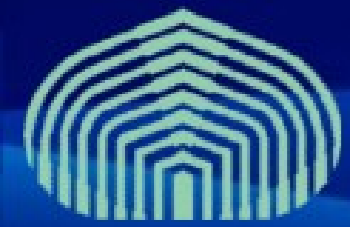
Arquitectura del WMS



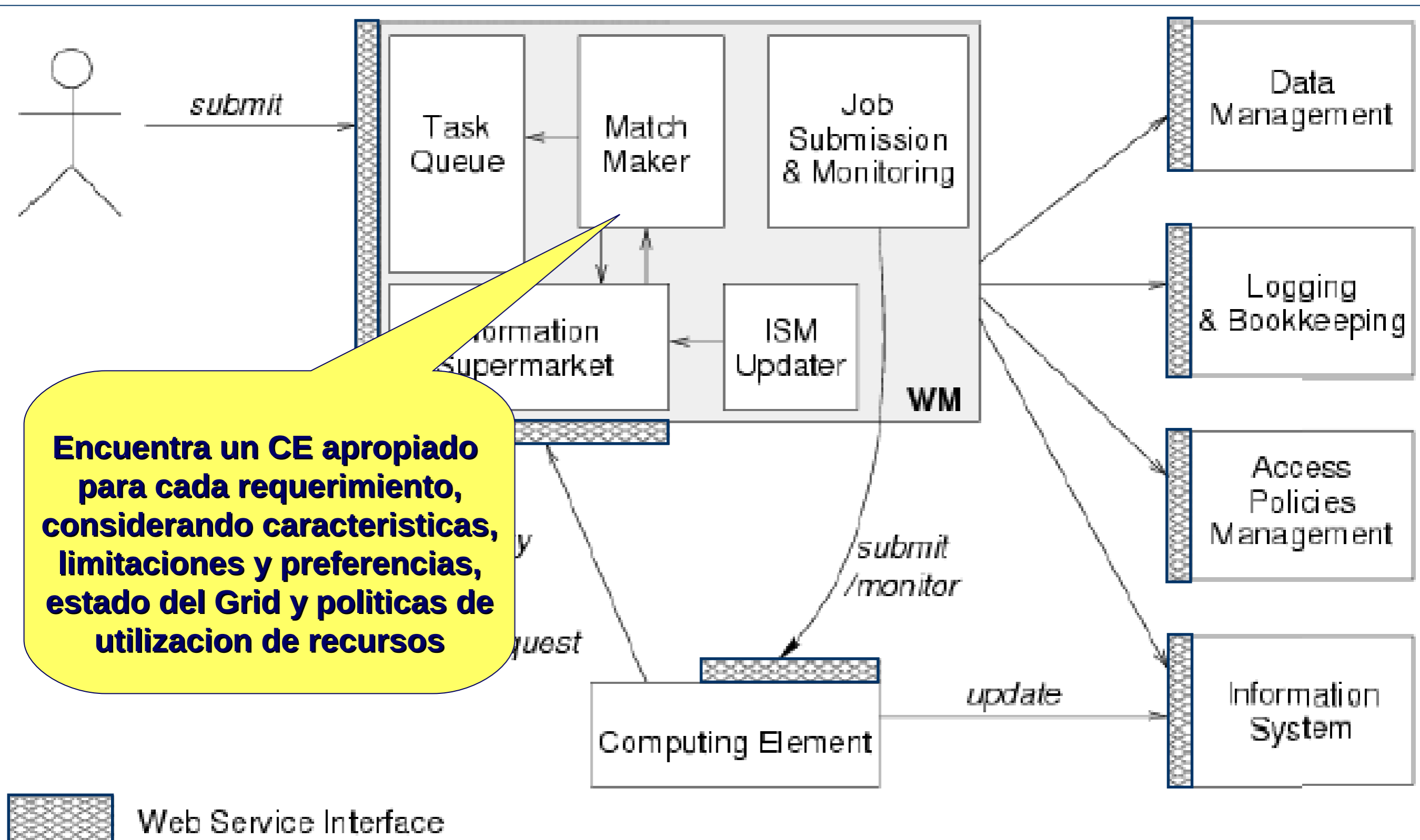


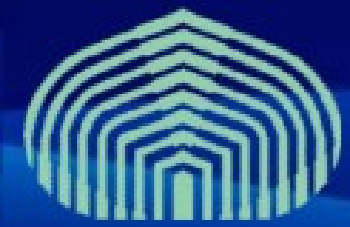
gLite WMS Architecture



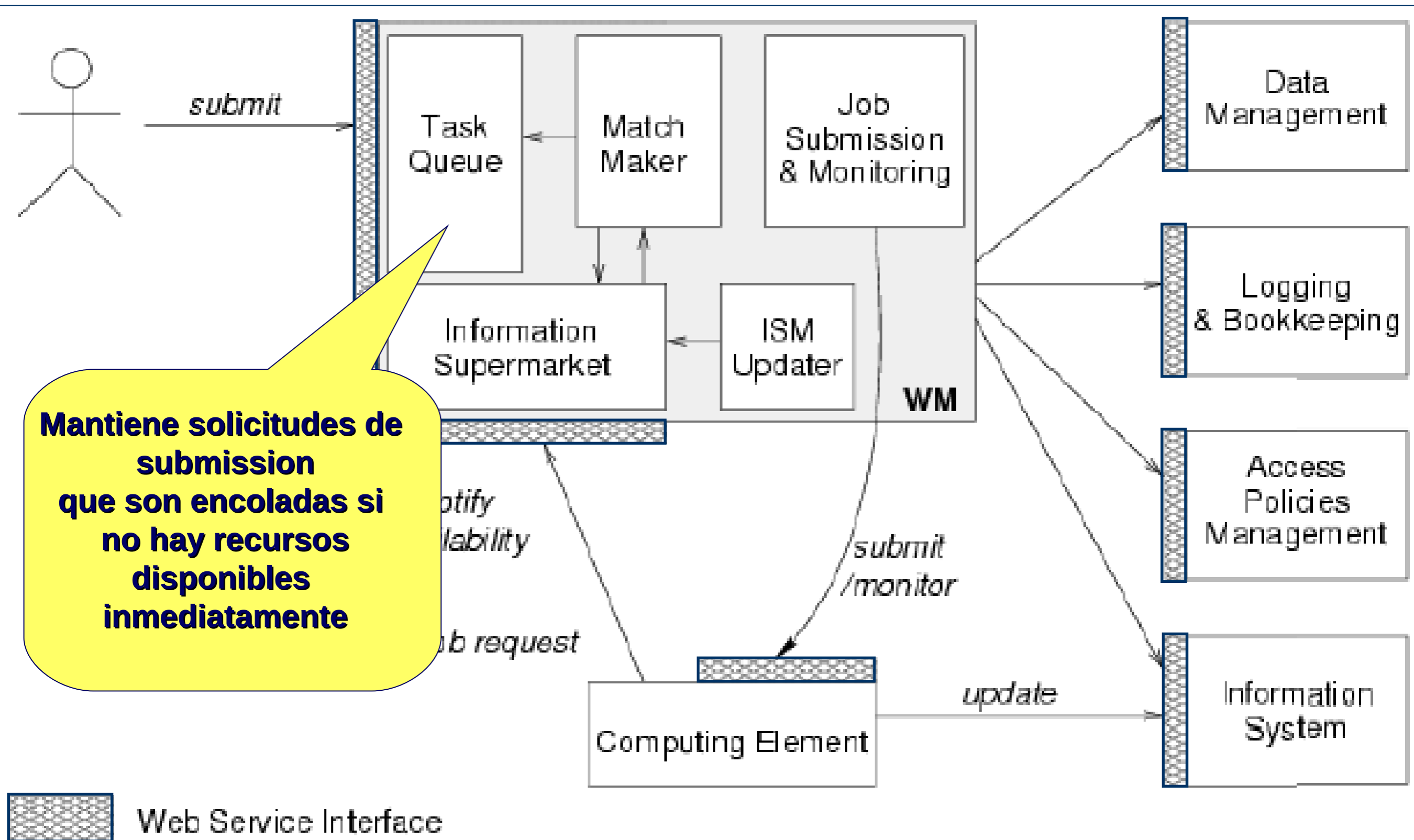


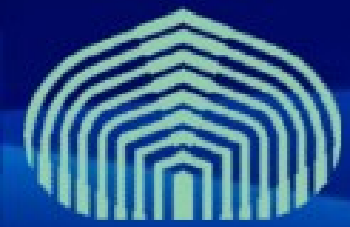
gLite WMS Architecture



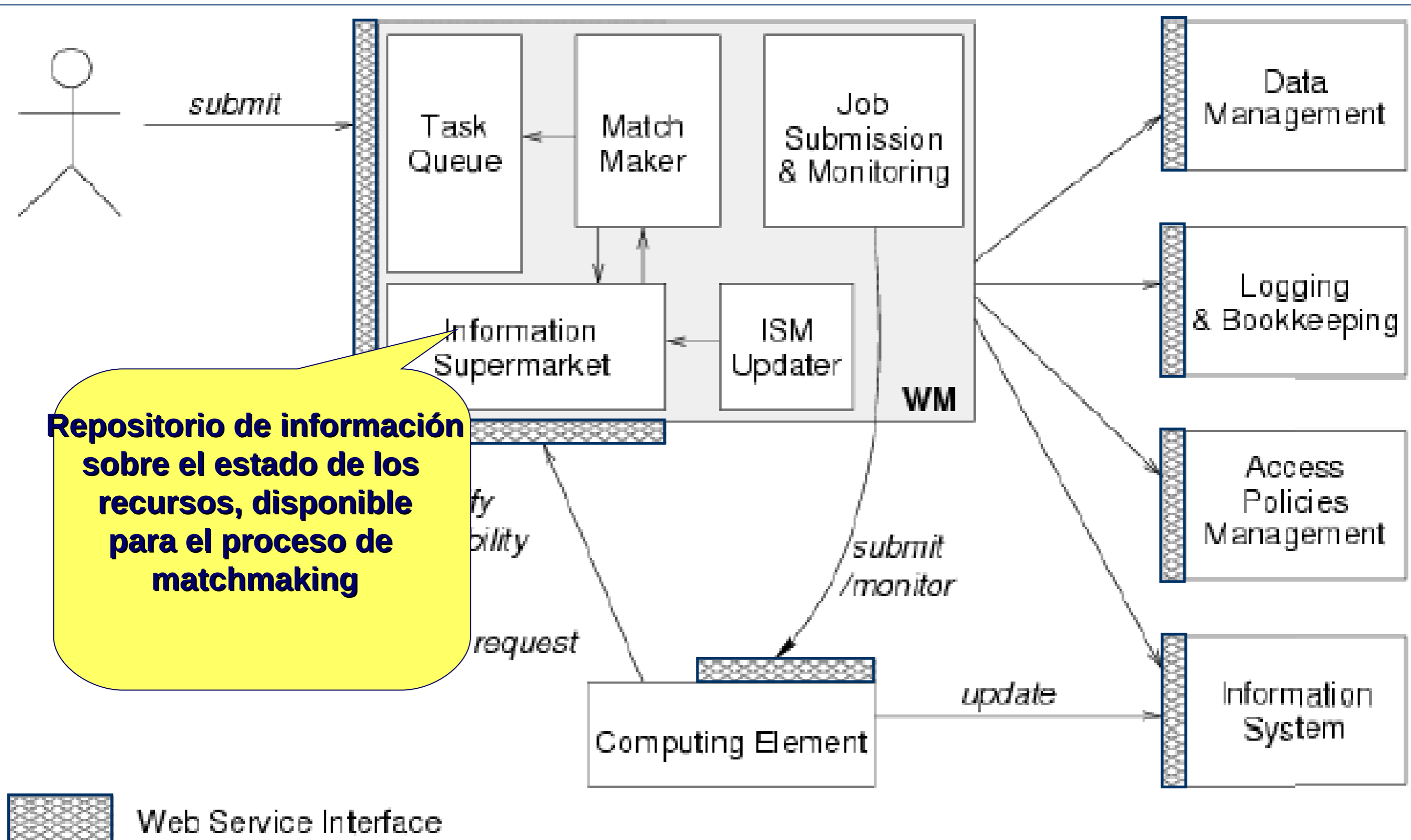


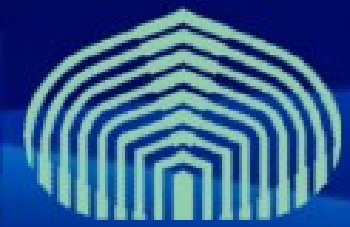
gLite WMS Architecture



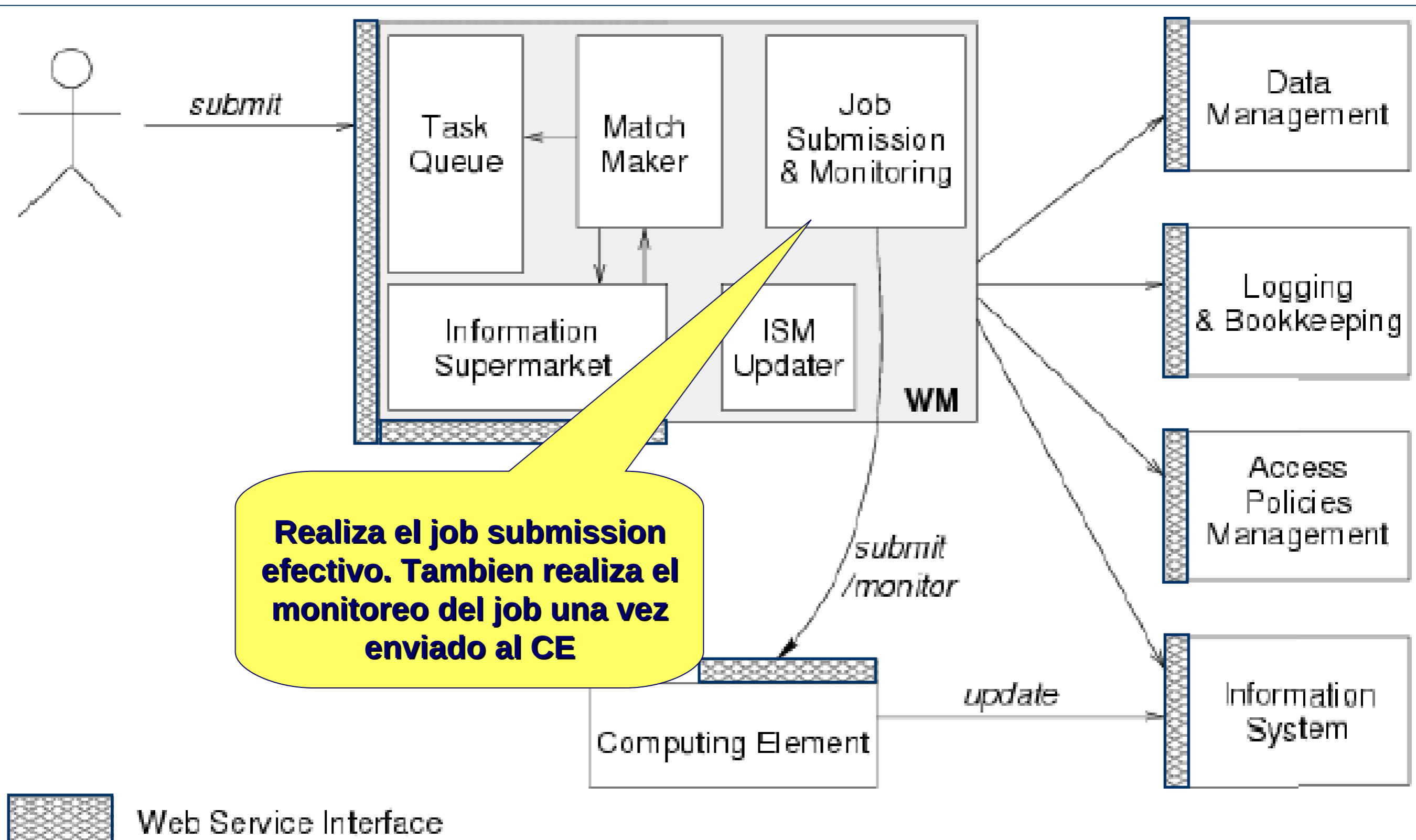


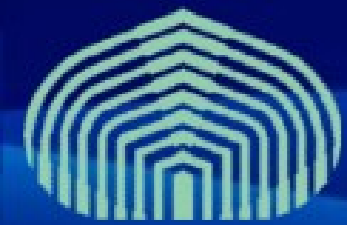
gLite WMS Architecture



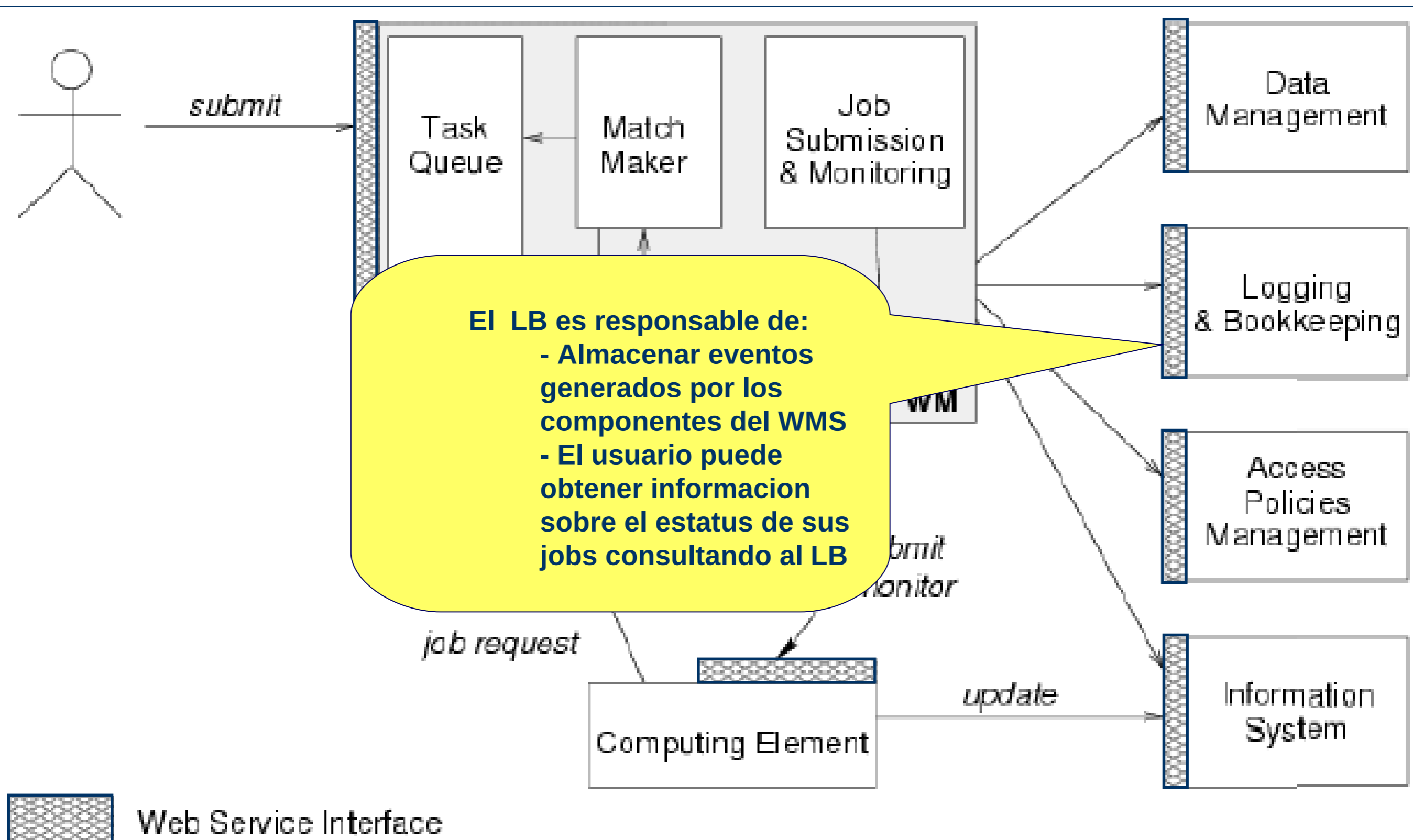


gLite WMS Architecture



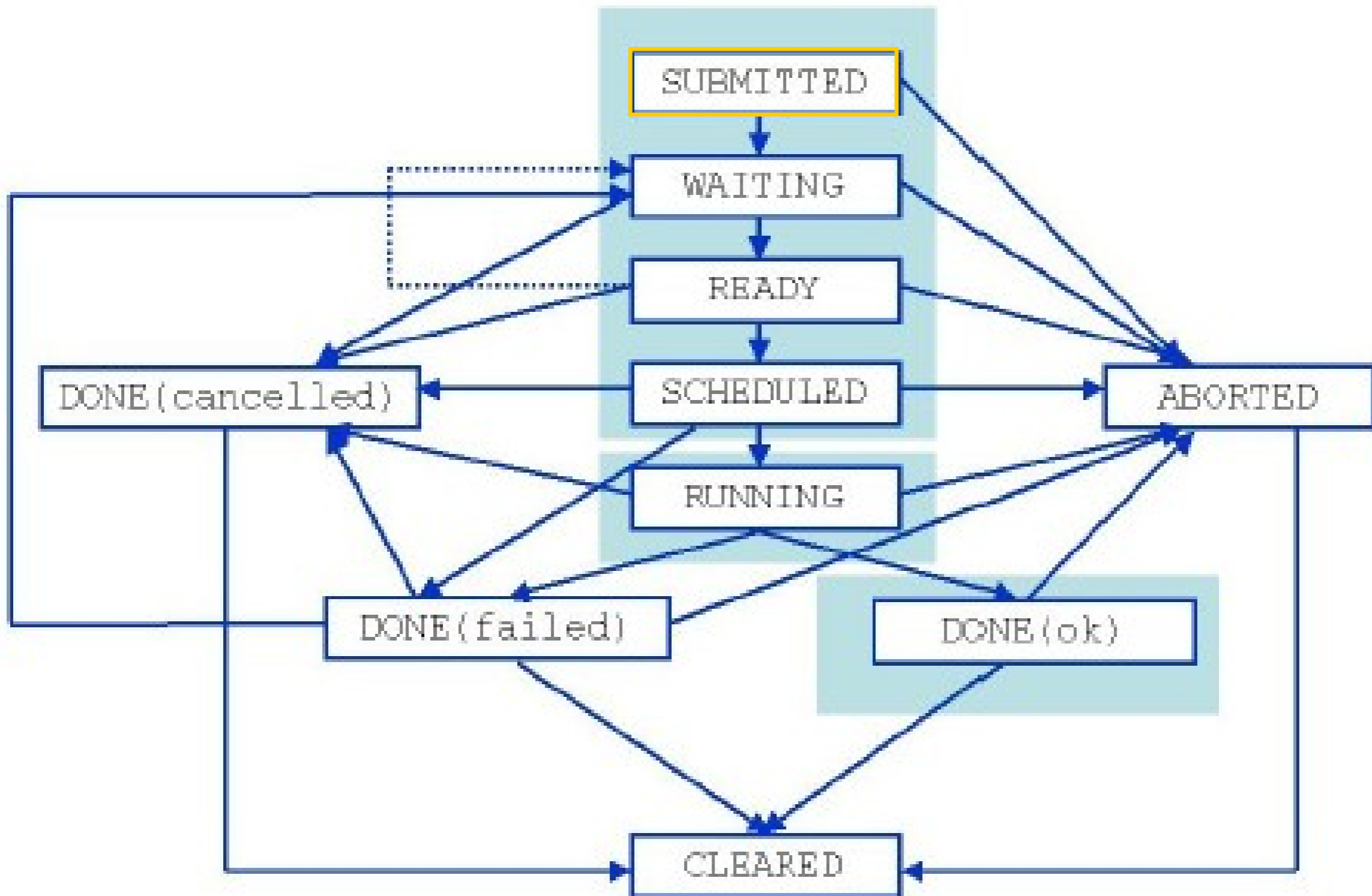


gLite WMS Architecture





Estados de los jobs





Estados de los jobs

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



Delegando un proxy al WMPProxy

- Cada job enviado al WMPProxy debe asociarse a una credencial proxy delegada previamente por el dueño del job al WMPProxy server.
 - Este proxy es usado cuando el WMPProxy requiere interactuar con otros servicios del grid
 - Dos formas de solicitar la delegación de las credenciales de usuario:
 - De forma “**automática**” durante la operación de submission
 - Solicitando una delegación “**explícita**”



- Para delegar una credencial explícitamente, el comando a usar es:
glite-wms-job-delegate-proxy -d <delegID>
donde **<delegID>** es un identificador elegido por el usuario.

Por ejemplo:

```
$ 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
```

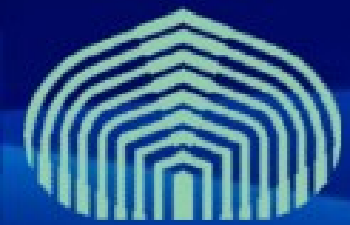
```
=====
```

```
Your proxy has been successfully delegated to the  
WMPProxy:
```

```
https://rb102.cern.ch:7443/glite_wms_wmproxy_server
```

```
with the delegation identifier: mydelegID
```

```
=====
```



Enviando un job simple

- Con un JDL simple, podemos enviarlo al WMPProxy con el comando

`glite-wms-job-submit -a test.jdl`

For the automatic delegation

```
$ glite-wms-job-submit -a test.jdl --url https://rb102.cern.ch:7443/glite_wms_wmproxy_server
```

```
==== glite-wms-job-submit Success =====
```

```
The job has been successfully submitted to the WMPProxy
```

```
Your job identifier is:
```

```
https://rb102.cern.ch:9000/vZKKk3gdB1a6RySxmq\_vQ
```

```
=====
```



- El comando retorna al usuario un **job identifier** (**jobID**), que identifica unívocamente el job y puede utilizarse para realizar otras operaciones sobre el job, como conocer su estado o cancelarlo.
- El formato del jobID es:
https://<LB_hostname>[:<port>]/<unique_string>
- Donde se garantiza que **<unique string>** es única y **<LB hostname>** es el nombre de host del servicio Logging and Bookkeeping (LB) para el job, que usualmente esta en la misma máquina del WMS usada para enviar el job.



Listar matching CE(s) para un job

- Se puede obtener una lista de CEs que cumplen con los requerimientos y restricciones de un job descritos en un archivo JDL:

```
$ glite-wms-job-list-match -d mydelegID --rank  
test.jdl
```

Connecting to the service

```
https://rb102.cern.ch:7443/glite_wms_wmproxy_server
```

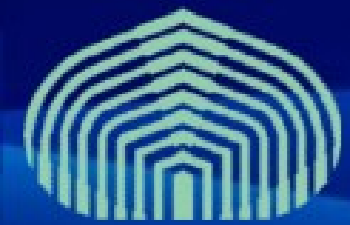
```
=====
```

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

```
=====
```



Obtener el estatus de un 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
*****
```



Cancelar un 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 =====
```

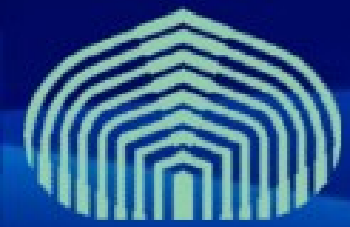
```
The cancellation request has been successfully submitted for the following job(s):
```

```
- https://rb102.cern.ch:9000/P1c60RFsrIZ9mnBALa7yZA
```

```
=====
```

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

CANCELLED



Obtener los resultados

```
$ 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/dae_yabp72aERhofLA6W2-LrJw
=====
```



WMS User Guide

- <https://edms.cern.ch/file/572489/1/WMS-guide.pdf>

WMPProxy guide

- <https://edms.cern.ch/file/674643/1/WMPROXY-guide.pdf>

JDL Attributes Specification

- <https://edms.cern.ch/file/590869/1/EGEE-JRA1-TEC-5908>

