D4.1a - VISHNU Tasks Management Service Package Design

COLLABORATORS

	TITLE : D4.1a - VISHNU Tasks Management Service Package Design	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Benjamin Isnard, Daouda Traoré, Eugène Pamba Capo-Chichi, Kevin Coulomb, and Ibrahima Cissé	April 25, 2013	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME
1	25/03/2011	Deliverable version	SysFera
2	20/05/2011	Modified description of section dedicated to Job Owner and Identifier (section 2.3). Added TMS submitJob sequence diagram (three figures added). Modified TMS Client Class Diagram, TMS Server Class Diagram. Modified TMS Client Side Components, TMS Server Side Components. Added TMS Monitor Daemon description in the definitions of components in section 2.1. Added TMS Monitor Daemon component in TMS Server Side Components. Modified exceptions list of all services in internal API specification chapter. Modified jobSubmit, jobOutputGetResult, jobOutputGetCompletedJobs services in internal API specification chapter.	SysFera
3	21/07/2011	Added SLURM batch scheduler	SysFera
4	10/01/2012	Updates of TMS: Added new internal service named jobSubmit_autom. Added new internal service named getListOfJobs_all. Modified TMS Data Class diagram. Modified TMS Server Class diagram.	SysFera

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
5	24/02/2012	Added LSF batch scheduler	SysFera
6	12/04/2012	Added Grid Engine batch scheduler	SysFera
7	12/01/2013	Added posix SHELL, update for design and new paragraph, plus some data about plugins	SysFera

Contents

List of Figures

Chapter 1

Document presentation

1.1 Document objectives

This document presents the detailed internal design of the Tasks Management Service (TMS) module. The purpose of this module is to handle all aspects of Tasks Management Service within the VISHNU system. The functional and non-functional requirements for this module are those described in the [D1.1a] and [D1.1c] documents (see References). The current document is part of the design phase of the software and therefore its main goal is to define the main components of the system architecture and their relationships.

1.2 Document structure

- Chapter 1 contains a brief overview of the document content.
- Chapter 2 contains a high-level overview of the system architecture except the TMS deployment diagram which is described in [D1.1g] (VISHNU Technical Architecture).
- Chapter 3 describes the internal API used for remote procedure calls through VISHNU.
- Chapter 4 describes the internal classes and data structures except the Vishnu core functions modelization which is included in [D2.1a] (VISHNU User Management Service Module Design)

1.3 References

- [D1.1a]: VISHNU General specifications
- [D1.1b]: VISHNU Spécifications techniques des besoins
- [D1.1c]: VISHNU API Detailed specifications
- [D1.1g]: VISHNU Technical Architecture
- [D2.1a]: VISHNU User Management Service Package Design

1.4 Acronyms

- **API**: Application Programming Interface
- **DB**: DataBase

- **CLI**: Command Line Interface
- **LL**: IBM LoadLeveler software
- **N/A**: Not Applicable
- **SeD**: A Server Daemon (VISHNU agent)
- **SOA**: Service Oriented Architecture
- **TMS**: Tasks Management Service
- **WS**: Web Services

1.5 Glossary

- **Components**: the software components represents a library or an executable program that provides a given interface to other components or to end-users.
- **Serialized type**: this is a class of data (C++ Class) which instances can be serialized in a XML string before being sent over an API (to or from the API). The data is deserialized on the other side of the channel in order to re-build the same instance of the class.

Chapter 2

System Architecture

2.1 Overview of the TMS software infrastructure

We present in this section a detailed description of the TMS package architecture in terms of software components. In addition we show the dependencies between components to highlight their reuse. These components follow a SOA model where each server publish a set of services that can be used by the clients. We present the different software layers from the API (services provided directly to the user) to the database (used by the server). The TMS module is splitted into eight different interrelated components. The diagrams shown in section 2.4 describe the relationships between these components. The definitions of the components are the following:

- **External API** publishes the services provided locally to the user as defined in the detailed specifications. The external API is actually implemented in C++ but wrapping layers are provided for the Python language and for Web Services.
- **Internal API** publishes the services provided by the servers.
- **TMS Client** contains intermediate (proxy) classes providing remote access to the business objects of **TMS Server**.
- **TMS Server** contains all classes implementing business objects that process requests sent by the TMS Client. It loads plugins implementing the different interfaces with all the supported batch schedulers.
- **TMS Monitor Daemon** which the only role is to keep an eye on the job state execution time by checking the status parameter of job table in the Vishnu Database.
- We need one of the following batch schedulers (see STB for details)
  - **Torque API** is the C API provided by the Torque batch scheduler.
  - **LoadLeveler API** is the C API provided by the IBM LoadLeveler batch scheduler..
  - **SLURM API** is the C API provided by the SLURM batch scheduler..
  - **LSF API** is the C API provided by the LSF batch scheduler..
  - **DRMAA API** is the C API provided by drmaa for the Grid Engine batch scheduler..
- **Vishnu Database** stores all data manipulated by the TMS Server.

2.2 Installation prerequisites

2.2.1 Client side

The installation of TMS module on the client needs the following libraries:

- VISHNU UMS libraries: the VISHNU User Management Service that allow the session creation and user authentification.
- C++ and Boost-C++ libraries
- ZMQ

2.2.2 Server side

The installation of TMS module on the server side needs the following libraries:

- VISHNU UMS libraries: the VISHNU User Management Service that allow the session creation and user authentification.
- C++ and Boost-C++ libraries
- A library of a batch scheduler (see STB for detail):

2.3 Job owner and identifier

Each TMS server process runs using a specific local system account dedicated to VISHNU. To allow a TMS server to submit jobs with a user local account on the machine on which the TMS server runs, the TMS server calls an other service and asks the executor of the service to submit the job. We call this service executor *tmsSlave*. *tmsSlave* uses *ssh executable command* by switching the local system account dedicated to VISHNU to user local account, and after connexion on the user local account it calls the underlying batch scheduler API. For example a job submitted by user jdupond on a machine dedicated to LoadLeveler will be visible as a job with identifier "hostname.344.0" and owner "jdupond". From the VISHNU user point of view, the job owner corresponds of user job owner on the submitted machine and the job identifier will be hidden and replaced by the generate global job identifier by VISHNU. For example the previous job will be visible in VISHNU as job "JOB_6767" and userId "jdupond".

This data translation is entirely transparent to the VISHNU user and simplifies job management.

This *tmsSlave* is used only by two services: *submitJob* service and *cancelJob* service.

The figures 2.1, 2.2 and 2.3 show the diagram sequence of TMS submitJob using *tmsSlave*.

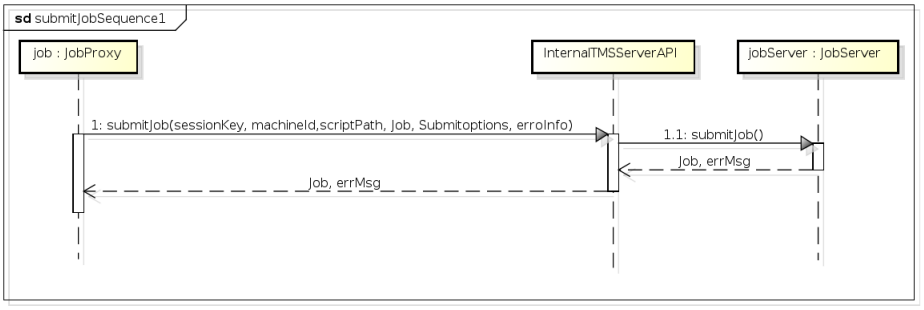

Figure 2.1: TMS submitJob sequence: first sequence

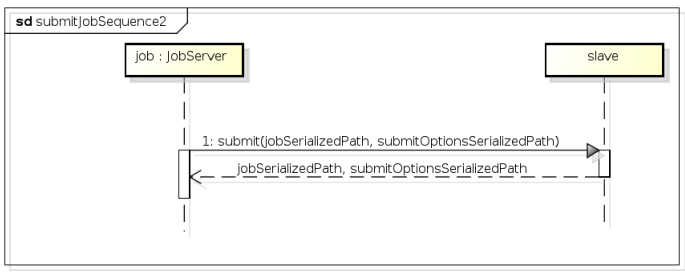

Figure 2.2: TMS submitJob sequence: second sequence

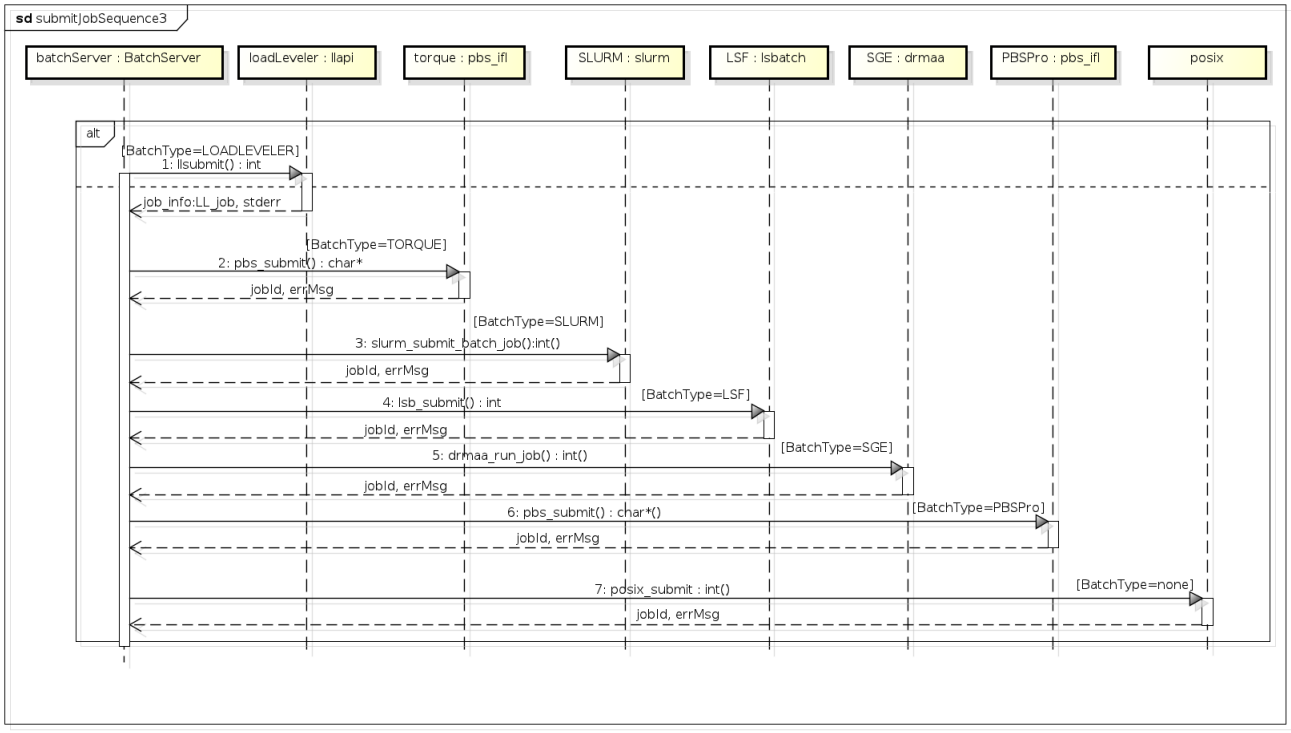

Figure 2.3: TMS submitJob sequence: last sequence

2.4 Architecture diagrams

2.4.1 TMS client-side components

This diagram shows the TMS client side components. Two services among all the services of the TMS external API (see ref. D1.1c) are shown here for example. These services are consumed by serval user interfaces: command-line, web services and python. All the interfaces of the TMS Client component are shown. The TMS client component provides four classes which are defined below:

- **JobProxy** provides the services that allow to submit, to cancel a job and to get information on a job.
- **QueryProxy** provides a generic service that allows to get job information (list of all submitted jobs, information on a specific given job and jobs'progression) and to list all availables queues on a given machine.
- **JobOutPutProxy** provides the services that allow to get the results of all finished jobs or a specific given job.

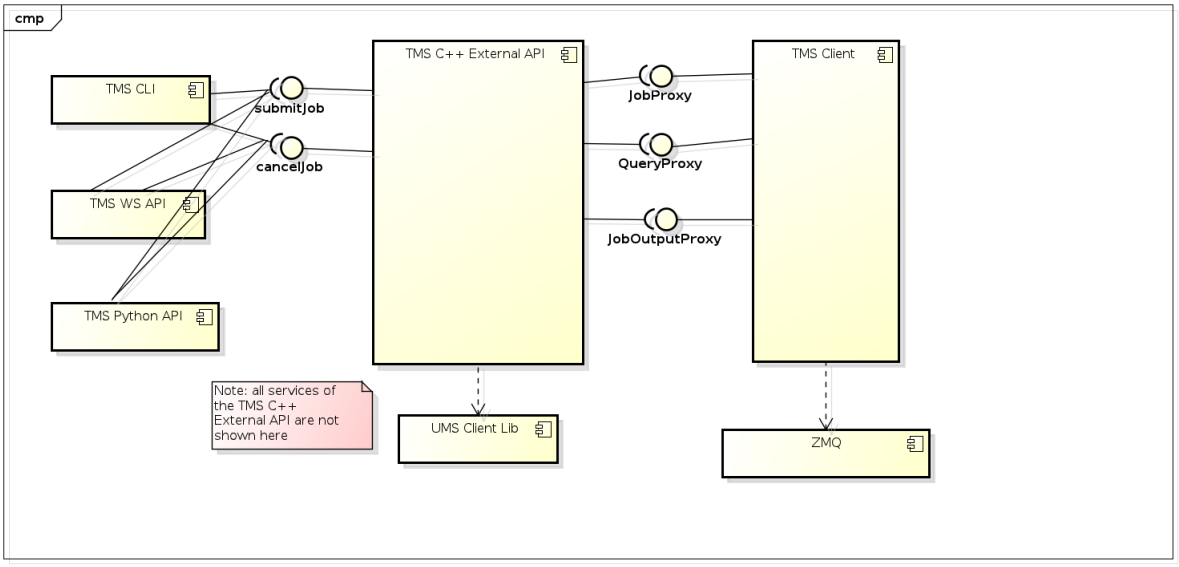

Figure 2.4: TMS client-side components

2.4.2 TMS server-side components

This diagram highlights the TMS server side components. Two services among all the services of the TMS internal API are shown here for example. All the interfaces of the TMS Server component are shown. Nevertheless, it is possible to use TMS without underlying batch scheduler. In this case, the scripts is executed as a classic shell script, it is important to note that this pseudo batch is always available, even when other API are used, but by default not used).

- **JobServer** resolves the services that allow to submit, to cancel a job and to get information on a job.
- **QueryServer** resolves a generic service that allows to get job information (list of all submitted jobs, information on a specific given job and jobs progression) and to list all availables queues on a given machine.
- **JobOutPutServer** resolves the services that allow to get the results of all finished jobs or a specific given job.

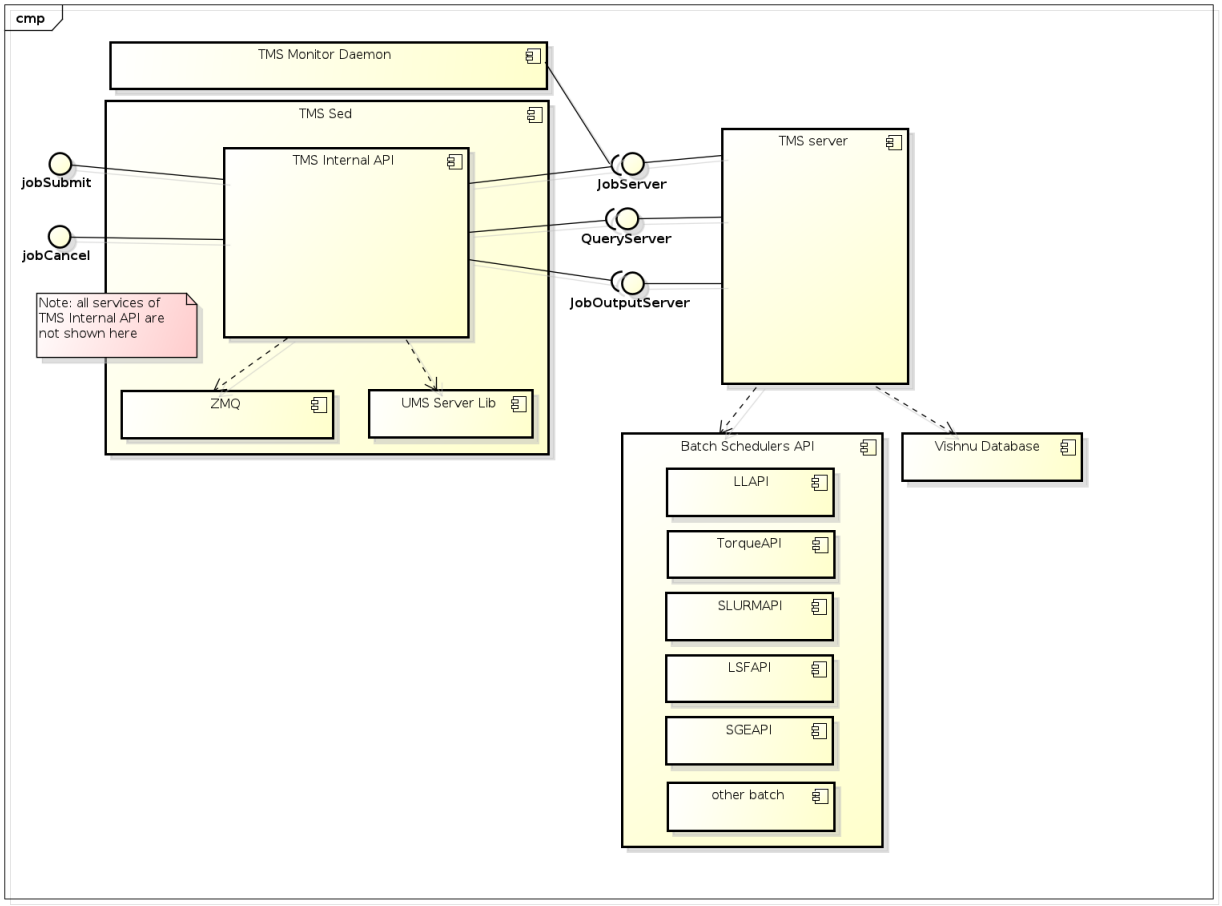

Figure 2.5: TMS server-side components

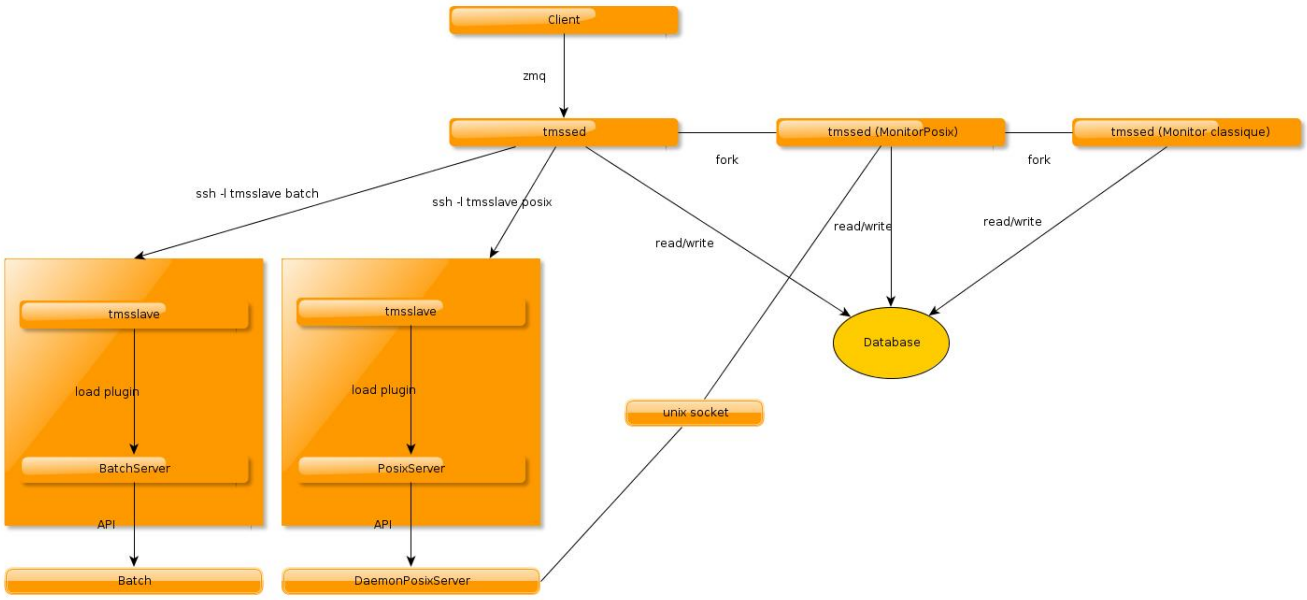

Figure 2.6: TMS communication diagram

2.4.3 Communication Bus Details

This diagram shows the communication paths between the Client host and the TMS servers using the communication Bus. On each remote VISHNU machine on which a batch scheduler library and API are installed a TMS server must be launched (for example in the figure 2.7 we have seven TMS servers launched on seven different nodes). The dispatcher is an optional VISHNU agent that can be executed on a dedicated host or on the same host as the TMS Server. All the communications between the entities here are done using ZMQ librarie and the communications can be tunneled through SSH tunnels if necessary. The diagram shows here all the communication paths in the case where the TMS Server2 is chosen by the dispatcher.

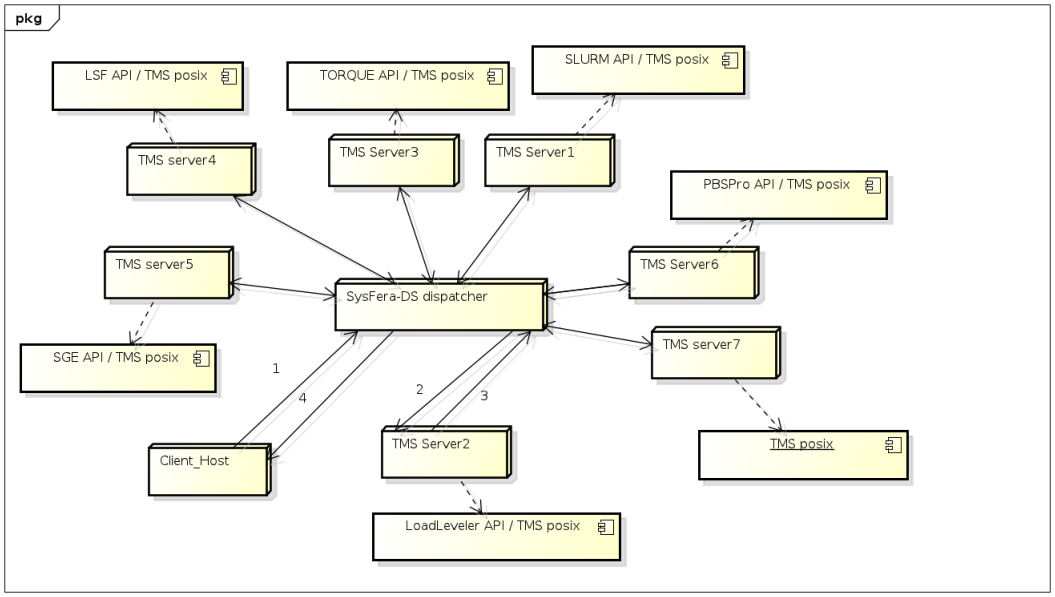

Figure 2.7: Communication Bus Details

Chapter 3

Internal API specification

3.1 Generic definition formats presentation

This section presents the formats used in this chapter to describe the services provided by the internal API.

3.1.1 Service definition format

Access

Here is detailed the access level of the service 'myService' (i.e. the privilege required to use it)

Parameters

The following table contains all the input and output parameters of the service, along with their type and description.

Parameter	Type	Serialized type	Description	Mode
sessionKey	string	n/a	This is an example of a required string input parameter	IN
listOfJobs	string	ListJobs	This is an example of an object output parameter that is serialized as a string	OUT

Description

Here is detailed the purpose of the service 'myService'

Return Value

Here are detailed the different return codes provided by the service.

Name	Description
ERRCODE_VISHNU_OK	The service has been performed successfully.
ERRCODE_UNKNOWN_MACHINE	This is the human-readable generic message that will be available to the user of the API.

Used by this(these) API function(s):

This shows the list of functions from the external Vishnu API (see [D1_1c]) that use this service.

3.2 Definition of the services of the package

3.2.1 Service jobSubmit_MACHINEID

Access

This service can be used by any VISHNU user

Parameters

Parameter	Type	Serialized type	Description	Mode
sessionKey	string	n/a	The session key is the encrypted identifier of the session generated by VISHNU	IN
machineId	string	n/a	Is the id of the machine on which the job must be submitted	IN
scriptContent	string	n/a	Is the content of the script to submit	IN
options	string	SubmitOptions	Is an instance of the class SubmitOptions. It allows the user to submit job by using different options	IN
job	string	Job	The Job object containing the input information (ex: jobPath) and output information (ex: jobId) of the job to submit	INOUT
errorInfo	string	n/a	Additional information provided when an error code is returned	OUT

Description

The jobSubmit_MACHINEID() function submits job on a machine of identifier MACHINEID through the use of a script (scriptContent)

Return Value

An error code is returned when an error occurs during the execution of the service

Name	Description
VISHNU_OK	The service was performed successfully
ERRCODE_INVALID_PARAM	Error invalid parameters
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
ERRCODE_SESSIONKEY_NOT_FOUND	The session key is unrecognized
ERRCODE_BATCH_SCHEDULER_ERROR	The batch scheduler indicates an error
ERRCODE_SSH	Vishnu not available (SSH error)
ERRCODE_SYSTEM	Vishnu not available (System)
ERRCODE_UNDEFINED	Internal Error: Undefined exception
ERRCODE_UNKNOWN_SESSION_ID	The session Id is unknown
ERRCODE_UNKNOWN_MACHINE	The machine id is unknown
ERRCODE_DIET	Vishnu not available (Service bus failure)
ERRCODE_DBERR	Vishnu not available (Database error)
ERRCODE_DBCONN	Vishnu not available (Database connection)

Used by this(these) API function(s):

TMS::submitJob

3.2.2 Service jobCancel_MACHINEID

Access

This service can be used by any VISHNU user

Parameters

Parameter	Type	Serialized type	Description	Mode
sessionKey	string	n/a	The session key is the encrypted identifier of the session generated by VISHNU	IN
machineId	string	n/a	Is the id of the machine on which the job is running	IN
job	string	Job	The Job object containing the input information (ex: jobId)	IN
errorInfo	string	n/a	Additional information provided when an error code is returned	OUT

Description

The jobCancel_MACHINEID() function cancels a job submitted on machine of identifier MACHINEID . If job id is equal to all, all submitted jobs by all users will be cancelled if the user is an administrator, and only jobs submitted by the user will be cancelled if the user is not an administrator.

Return Value

An error code is returned when an error occurs during the execution of the service

Name	Description
VISHNU_OK	The service was performed successfully
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
ERRCODE_SESSIONKEY_NOT_FOUND	The session key is unrecognized
ERRCODE_BATCH_SCHEDULER_ERROR	The batch scheduler indicates an error
ERRCODE_ALREADY_CANCELED	The job is already canceled
ERRCODE_UNKNOWN_JOBID	The job id is unknown
ERRCODE_UNKNOWN_SESSION_ID	The session Id is unknown
ERRCODE_UNKNOWN_MACHINE	The machine id is unknown
ERRCODE_PERMISSION_DENIED	Permission denied
ERRCODE_DIET	Vishnu not available (Service bus failure)
ERRCODE_DBERR	Vishnu not available (Database error)
ERRCODE_DBCONN	Vishnu not available (Database connection)
ERRCODE_UNDEFINED	Internal Error: Undefined exception
ERRCODE_SYSTEM	Vishnu not available (System)
ERRCODE_SSH	Vishnu not available (SSH error)

Used by this(these) API function(s):

TMS::cancelJob

3.2.3 Service JobInfo_MACHINEID

Access

This service can be used by any VISHNU user

Parameters

Parameter	Type	Serialized type	Description	Mode
sessionKey	string	n/a	The session key is the encrypted identifier of the session generated by VISHNU	IN
machineId	string	n/a	Is the id of the machine on which the job is running	IN
job	string	Job	The Job object containing the input information (ex: jobId) and the resulting information	INOUT
errorInfo	string	n/a	Additional information provided when an error code is returned	OUT

Description

The JobInfo_MACHINEID() function gets information on a job submitted on machine of identifier MACHINEID

Return Value

An error code is returned when an error occurs during the execution of the service

Name	Description
VISHNU_OK	The service was performed successfully
ERRCODE_JOB_IS_NOT_TERMINATED	The job is not terminated
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
ERRCODE_SESSIONKEY_NOT_FOUND	The session key is unrecognized
ERRCODE_UNKNOWN_JOBID	The job id is unknown
ERRCODE_UNKNOWN_MACHINE	The machine id is unknown
ERRCODE_UNKNOWN_SESSION_ID	The session Id is unknown
ERRCODE_UNDEFINED	Internal Error: Undefined exception
ERRCODE_DIET	Vishnu not available (Service bus failure)
ERRCODE_DBERR	Vishnu not available (Database error)
ERRCODE_DBCONN	Vishnu not available (Database connection)
ERRCODE_SYSTEM	Vishnu not available (System)

Used by this(these) API function(s):

TMS::getJobInfo

3.2.4 Service getListOfJobs_MACHINEID

Access

This service can be used by any VISHNU user

Parameters

Parameter	Type	Serialized type	Description	Mode
sessionKey	string	n/a	The session key is the encrypted identifier of the session generated by VISHNU	IN
machineId	string	n/a	Is the id of the machine on which the jobs are running	IN
options	string	ListJobsOptions	Additional options for jobs listing	IN
listOfJobs	string	ListJobs	The constructed object list of jobs	OUT
errorInfo	string	n/a	Additional information provided when an error code is returned	OUT

Description

The getListOfJobs_MACHINEID() function gets a list of all submitted jobs submitted on machine of identifier MACHINEID

Return Value

An error code is returned when an error occurs during the execution of the service

Name	Description
VISHNU_OK	The service was performed successfully
ERRCODE_UNKNOWN_SESSION_ID	The session Id is unknown
ERRCODE_UNKNOWN_MACHINE	The machine id is unknown
ERRCODE_BATCH_SCHEDULER_ERROR	The batch scheduler indicates an error
ERRCODE_DBCONN	Vishnu not available (Database connection)
ERRCODE_DBERR	Vishnu not available (Database error)
ERRCODE_DIET	Vishnu not available (Service bus failure)

Name	Description
ERRCODE_SYSTEM	Vishnu not available (System)
ERRCODE_UNDEFINED	Internal Error: Undefined exception
ERRCODE_SESSIONKEY_NOT_FOUND	The session key is unrecognized
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.

Used by this(these) API function(s):

TMS::listJobs

3.2.5 Service getListOfQueues_MACHINEID

Access

This service can be used by any VISHNU user

Parameters

Parameter	Type	Serialized type	Description	Mode
sessionKey	string	n/a	The session key is the encrypted identifier of the session generated by VISHNU	IN
machineId	string	n/a	Is the id of the machine that the user wants to list queues	IN
queueName	string	n/a	The name of queue to query	IN
listofQueues	string	ListQueues	The list of queues	OUT
errorInfo	string	n/a	Additional information provided when an error code is returned	OUT

Description

The getListOfQueues_MACHINEID() function gets queues information of the machine identified by MACHINEID

Return Value

An error code is returned when an error occurs during the execution of the service

Name	Description
VISHNU_OK	The service was performed successfully
ERRCODE_BATCH_SCHEDULER_ERROR	The batch scheduler indicates an error
ERRCODE_DBCONN	Vishnu not available (Database connection)
ERRCODE_DBERR	Vishnu not available (Database error)
ERRCODE_DIET	Vishnu not available (Service bus failure)
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
ERRCODE_SESSIONKEY_NOT_FOUND	The session key is unrecognized
ERRCODE_SYSTEM	Vishnu not available (System)
ERRCODE_UNDEFINED	Internal Error: Undefined exception
ERRCODE_INVALID_PARAM	Error invalid parameters
ERRCODE_UNKNOWN_SESSION_ID	The session Id is unknown
ERRCODE_UNKNOWN_MACHINE	The machine id is unknown

Used by this(these) API function(s):

TMS::listQueues

3.2.6 Service getJobsProgression_MACHINEID

Access

This service can be used by any VISHNU user

Parameters

Parameter	Type	Serialized type	Description	Mode
sessionKey	string	n/a	The session key is the encrypted identifier of the session generated by VISHNU	IN
machineId	string	n/a	Is the id of the machine that the user wants to get jobs progression	IN
options	string	ProgressOptions	Is an object containing the available options jobs for progression	IN
listOfProgress	string	ListProgression	Is the object containing jobs progression information	OUT
errorInfo	string	n/a	Additional information provided when an error code is returned	OUT

Description

The getJobsProgression_MACHINEID() function gets the progression status of jobs submitted on machine of identifier MACHINEID

Return Value

An error code is returned when an error occurs during the execution of the service

Name	Description
VISHNU_OK	The service was performed successfully
ERRCODE_BATCH_SCHEDULER_ERROR	The batch scheduler indicates an error
ERRCODE_DBCONN	Vishnu not available (Database connection)
ERRCODE_DBERR	Vishnu not available (Database error)
ERRCODE_DIET	Vishnu not available (Service bus failure)
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
ERRCODE_SESSIONKEY_NOT_FOUND	The session key is unrecognized
ERRCODE_SYSTEM	Vishnu not available (System)
ERRCODE_UNDEFINED	Internal Error: Undefined exception
ERRCODE_UNKNOWN_MACHINE	The machine id is unknown
ERRCODE_UNKNOWN_SESSION_ID	The session Id is unknown

Used by this(these) API function(s):

TMS::getJobProgress

3.2.7 Service jobOutputGetResult_MACHINEID

Access

This service can be used by any VISHNU user

Parameters

Parameter	Type	Serialized type	Description	Mode
sessionKey	string	n/a	The session key is the encrypted identifier of the session generated by VISHNU	IN
machineId	string	n/a	Is the id of the machine on which the jobs has been submitted	IN
result	string	JobResult	The Job object containing the input information (ex: jobId)	IN
errorInfo	string	n/a	Additional information provided when an error code is returned	OUT

Parameter	Type	Serialized type	Description	Mode
containerId	char*	n/a	Is the identifier of the diet container which contains the job output and error paths	OUT

Description

The jobOutputGetResult_MACHINEID() function gets outputPath and errorPath of a job submitted on machine of identifer MACHINEID

Return Value

An error code is returned when an error occurs during the execution of the service

Name	Description
VISHNU_OK	The service was performed successfully
ERRCODE_ALREADY_CANCELED	The job is already canceled
ERRCODE_ALREADY_DOWNLOADED	The job is already downloaded
ERRCODE_BATCH_SCHEDULER_ERROR	The batch scheduler indicates an error
ERRCODE_DBCONN	Vishnu not available (Database connection)
ERRCODE_DBERR	Vishnu not available (Database error)
ERRCODE_JOB_IS_NOT_TERMINATED	The job is not terminated
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
ERRCODE_SESSIONKEY_NOT_FOUND	The session key is unrecognized
ERRCODE_SSH	Vishnu not available (SSH error)
ERRCODE_SYSTEM	Vishnu not available (System)
ERRCODE_UNDEFINED	Internal Error: Undefined exception
ERRCODE_UNKNOWN_MACHINE	The machine id is unknown
ERRCODE_UNKNOWN_JOBID	The job id is unknown
ERRCODE_UNKNOWN_SESSION_ID	The session Id is unknown

Used by this(these) API function(s):

TMS::getJobOutput

3.2.8 Service jobOutputGetCompletedJobs_MACHINEID

Access

This service can be used by any VISHNU user

Parameters

Parameter	Type	Serialized type	Description	Mode
sessionKey	string	n/a	The session key is the encrypted identifier of the session generated by VISHNU	IN
machineId	string	n/a	Is the id of the machine on which the jobs are been submitted	IN
listOfResults	string	ListJobResults	Is the list of jobs results	OUT
errorInfo	string	n/a	Additional information provided when an error code is returned	OUT
containerId	char*	n/a	Is the identifier of the diet container which contains the completed jobs output and error paths	OUT

Description

The jobOutputGetCompletedJobs_MACHINEID() function dynamically gets outputPath and errorPath of completed jobs of the machine identified by MACHINEID

Return Value

An error code is returned when an error occurs during the execution of the service

Name	Description
VISHNU_OK	The service was performed successfully
ERRCODE_DBCONN	Vishnu not available (Database connection)
ERRCODE_DBERR	Vishnu not available (Database error)
ERRCODE_DIET	Vishnu not available (Service bus failure)
ERRCODE_SSH	Vishnu not available (SSH error)
ERRCODE_SYSTEM	Vishnu not available (System)
ERRCODE_UNDEFINED	Internal Error: Undefined exception
ERRCODE_UNKNOWN_MACHINE	The machine id is unknown
ERRCODE_UNKNOWN_SESSION_ID	The session Id is unknown

Used by this(these) API function(s):

TMS::getCompletedJobsOutput

3.2.9 Service jobSubmit_autom

Access

This service can be used by any VISHNU user

Parameters

Parameter	Type	Serialized type	Description	Mode
sessionKey	string	n/a	The session key is the encrypted identifier of the session generated by VISHNU	IN
autom	string	n/a	The keyword used to submit automatically a job	IN
scriptContent	string	n/a	Is the content of the script to submit	IN
options	string	SubmitOptions	Is an instance of the class SubmitOptions. It allows the user to submit job by using different options	IN
job	string	Job	The Job object containing the input information (ex: jobPath) and output information (ex: jobId) of the job to submit	INOUT
errorInfo	string	n/a	Additional information provided when an error code is returned	OUT

Description

The jobSubmit_autom() function submits a job automatically on a best machine (for now three criterions are used: minimum number of waiting jobs, minimum number of running jobs and the total number of jobs) through the use of a script (scriptFilePath) which must be generic script using VISHNU's generic directives for all batch schedulers

Return Value

An error code is returned when an error occurs during the execution of the service

Name	Description
VISHNU_OK	The service was performed successfully
ERRCODE_INVALID_PARAM	Error invalid parameters
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.
ERRCODE_SESSIONKEY_NOT_FOUND	The session key is unrecognized
ERRCODE_BATCH_SCHEDULER_ERROR	The batch scheduler indicates an error
ERRCODE_SSH	Vishnu not available (SSH error)
ERRCODE_SYSTEM	Vishnu not available (System)

Name	Description
ERRCODE_UNDEFINED	Internal Error: Undefined exception
ERRCODE_UNKNOWN_SESSION_ID	The session Id is unknown
ERRCODE_DIET	Vishnu not available (Service bus failure)
ERRCODE_DBERR	Vishnu not available (Database error)
ERRCODE_DBCONN	Vishnu not available (Database connection)

Used by this(these) API function(s):

TMS::submitJob

3.2.10 Service getListOfJobs_all

Access

This service can be used by any VISHNU user

Parameters

Parameter	Type	Serialized type	Description	Mode
sessionKey	string	n/a	The session key is the encrypted identifier of the session generated by VISHNU	IN
all	string	n/a	The keyword used to list jobs on all machines	IN
options	string	ListJobsOptions	Additional options for jobs listing	IN
listOfJobs	string	ListJobs	The constructed object list of jobs	OUT
errorInfo	string	n/a	Additional information provided when an error code is returned	OUT

Description

The getListOfJobs_all() function allows to list the submitted jobs on all machines

Return Value

An error code is returned when an error occurs during the execution of the service

Name	Description
VISHNU_OK	The service was performed successfully
ERRCODE_UNKNOWN_SESSION_ID	The session Id is unknown
ERRCODE_BATCH_SCHEDULER_ERROR	The batch scheduler indicates an error
ERRCODE_DBCONN	Vishnu not available (Database connection)
ERRCODE_DBERR	Vishnu not available (Database error)
ERRCODE_DIET	Vishnu not available (Service bus failure)
ERRCODE_SYSTEM	Vishnu not available (System)
ERRCODE_UNDEFINED	Internal Error: Undefined exception
ERRCODE_SESSIONKEY_NOT_FOUND	The session key is unrecognized
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey is expired. The session is closed.

Used by this(these) API function(s):

TMS::listJobs

Chapter 4

Internal class and data structures

4.1 Introduction

This chapter introduces the details of the implementation of the different components described in chapter 2 (Architecture). It is composed of three sections:

- **Client modelization**: describes the classes used to implement the *TMS Client* component.
- **Server modelization**: describes the classes used to implement the *TMS Server* component.
- **Data modelization**: describes the data structure used to implement the *TMS Client* component and the *TMS Server* component.

4.2 TMS client modelization

4.2.1 Class diagrams

4.2.1.1 TMS Client Class Diagram

This diagram describes all classes used to communicate with VISHNU System. Each class proxy contains the corresponding data class illustrated on the TMS Data modelization section and the methods usable by the TMS Client component.

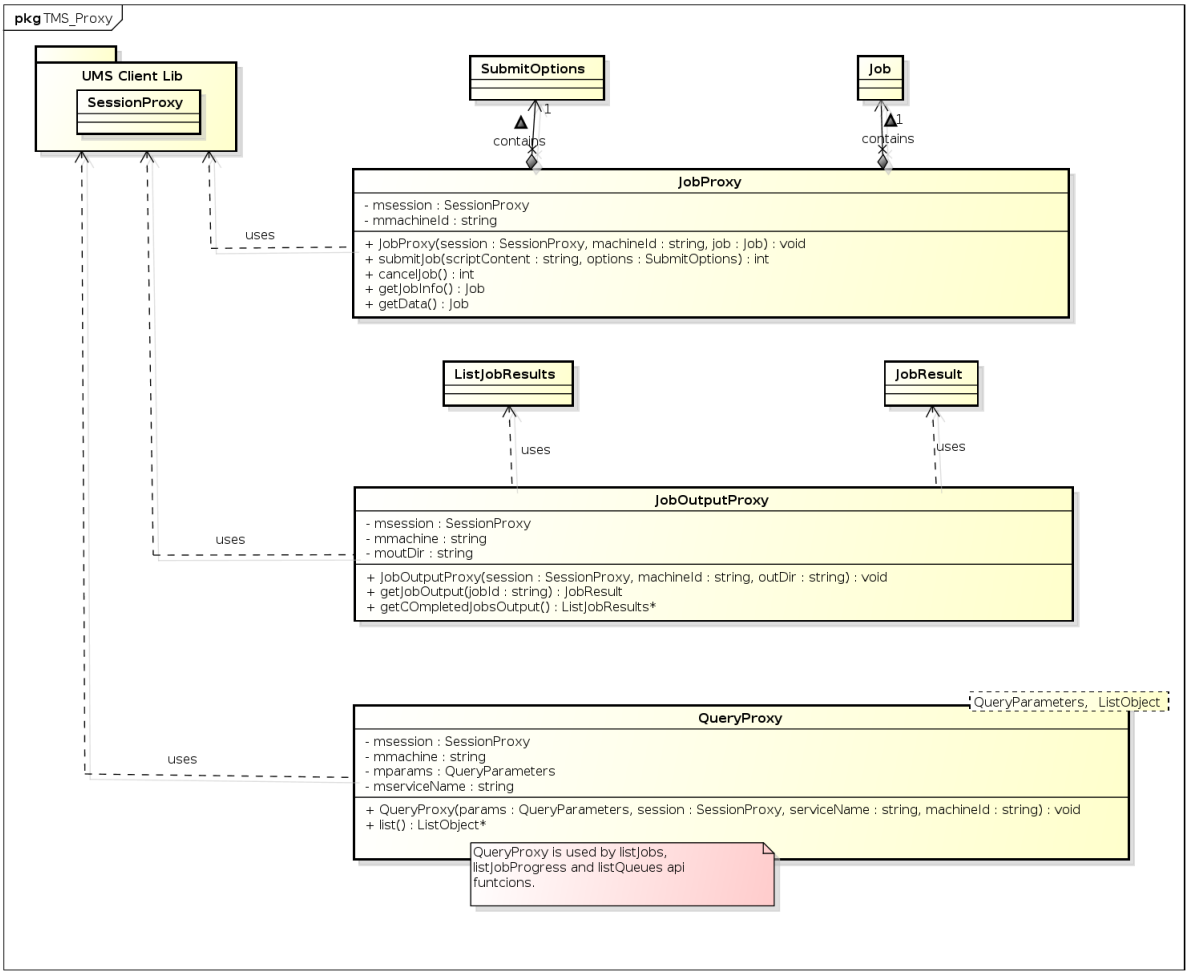

Figure 4.1: TMS Client Class Diagram

4.3 TMS server modelization

4.3.1 Class diagrams

4.3.1.1 TMS Server Class Diagram

This diagram presents the main objects used by TMS server component to process the TMS Client component requests.

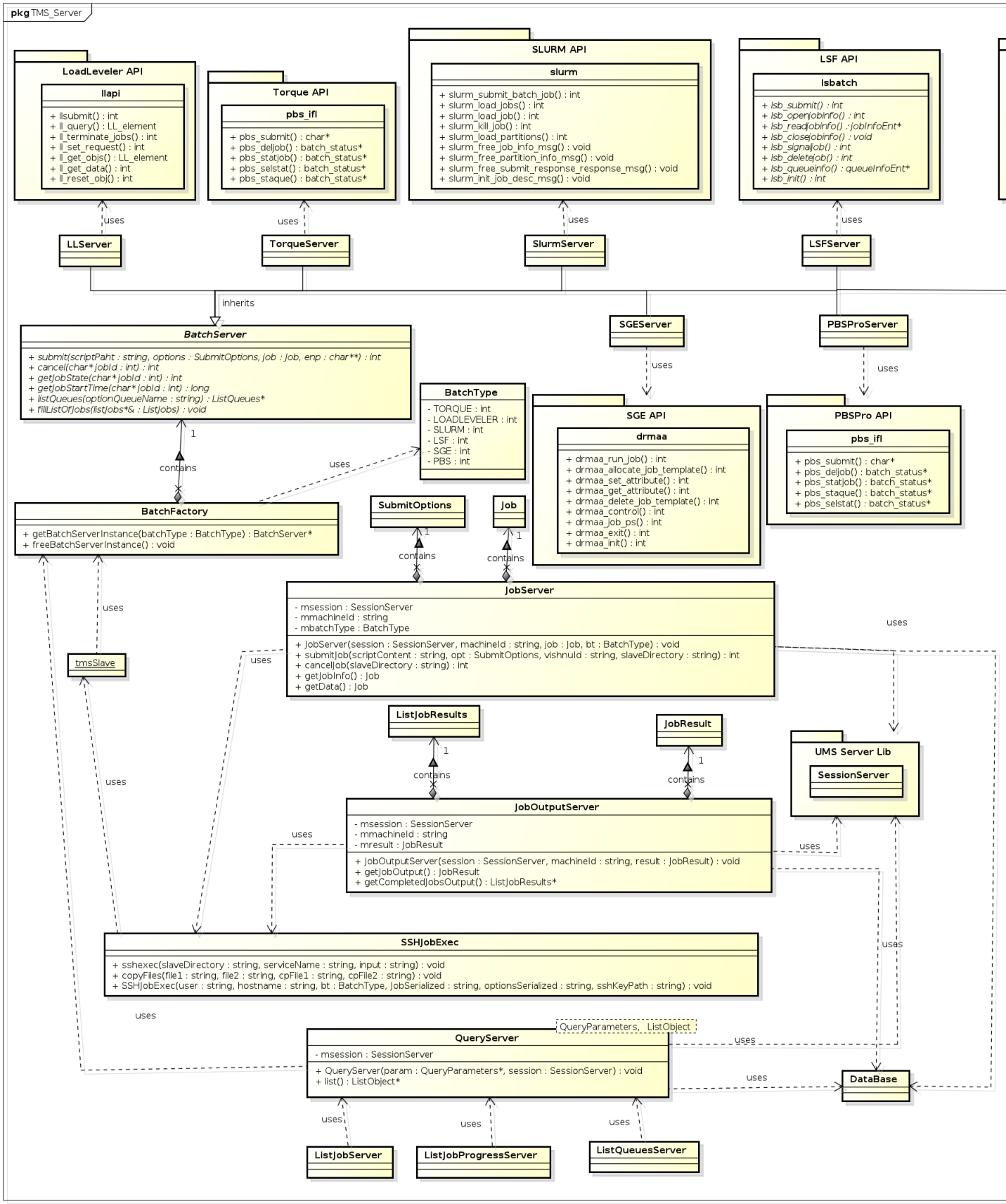

Figure 4.2: TMS Server Class Diagram

4.4 TMS data modelization

4.4.1 Class diagrams

4.4.1.1 TMS Data Class Diagram

This diagram illustrates the structure and the relationship between data manipulated by the components Client and Server.

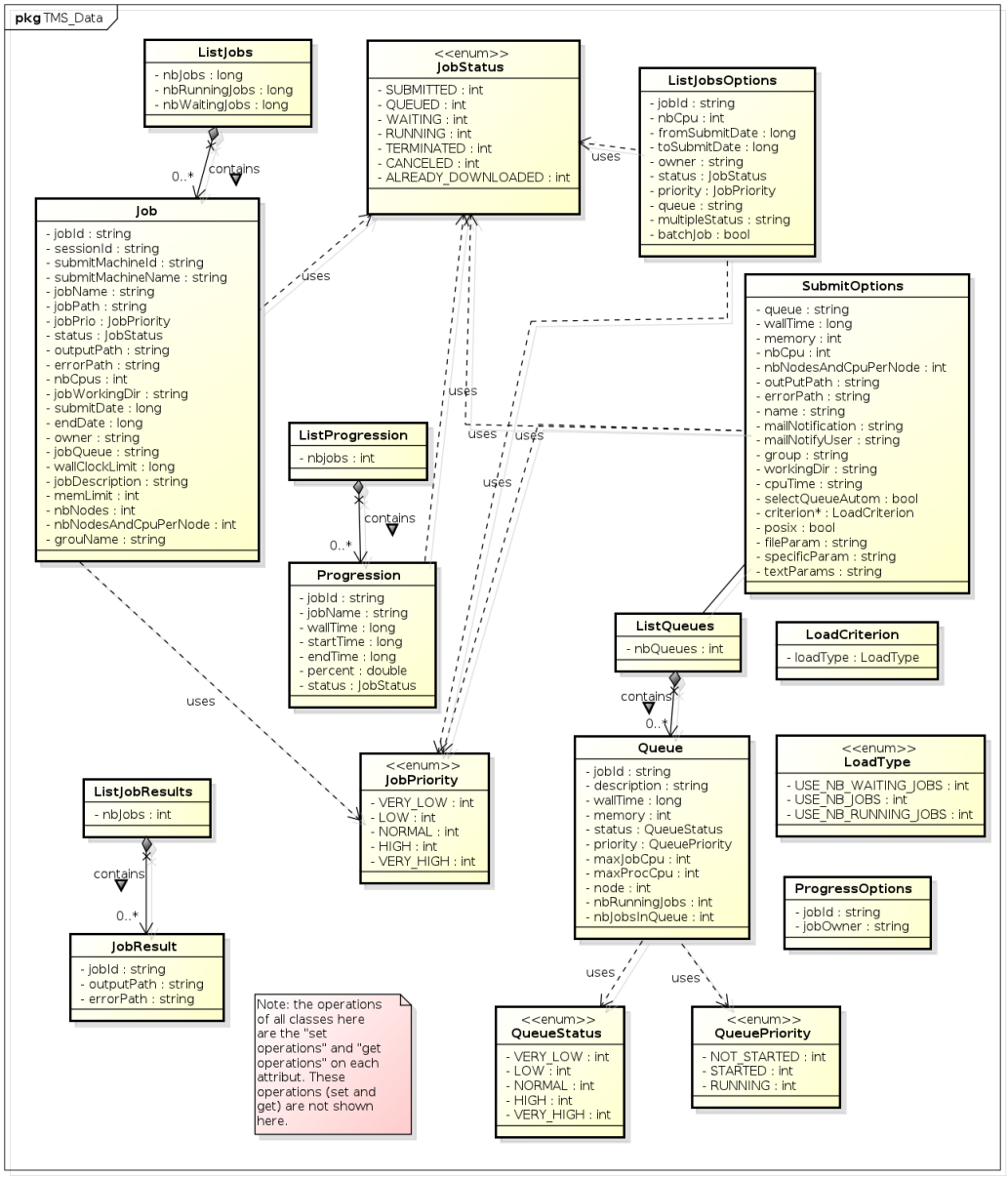

Figure 4.3: TMS Data Class Diagram