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 1, 2011	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
1	25/03/2011	Deliverable version	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 SysFera-DS.
- 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 (SysFera-DS 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.
- **SysFera-DS**: open-source middleware developped by SysFera.

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. The communication and service assignment layer is provided by the SysFera-DS middleware. 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. The services of the external API are implemented using requests sent through the SysFera-DS middleware to this API.
- **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.
- **Sysfera-DS Client API** is the C++ client API provided by the SysFera-DS toolbox.
- **Sysfera-DS Server API** is the C++ server API provided by the SysFera-DS toolbox.
- **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..
- **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
- Sysfera-DS Client libraries

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
- Sysfera-DS Server libraries
- IBM LoadLeveler library
- Torque library

2.3 Job owner and identifier

Each TMS server process runs using a specific local system account dedicated to VISHNU. All VISHNU jobs will be submitted to LoadLeveler or Torque API using this system account and will be identified on these batch schedulers by local identifiers. For example a given job on LL will be visible as a job with identifier "hostname.344.0" and owner "vishnu". From the VISHNU user point of view, these local identifiers and the local job owner will be hidden and replaced by the user's identifier and global job identifiers. 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.

2.3.1 Use constraints of batch scheduler API

Because the TMS server runs with a specific local system account dedicated to VISHNU, the use of the batch scheduler API causes some problems. We present below these problems.

- **Binary execution**: If the user specifies in his job script (file containing job characteristics) to execute some local binaries (installed only in the user local account), the execution of this script will be rejected by the underlying batch scheduler if the local account dicated to TMS has not access to the binary. The figure 2.1 illustrates an example case.
- **Access rights for files and directories**: If the user indicates in his job script to redirect some outputs into files or directories of his home directory, the execution of this job will be rejected by the underlying batch scheduler if the local account dicated to TMS has not access to these files or directories. Sometimes the input data arguments of some commands in the job script can be files or directories, in this case the local account dicated to TMS server must be have read access on these files and directories. The figure 2.1 illustrates an example case

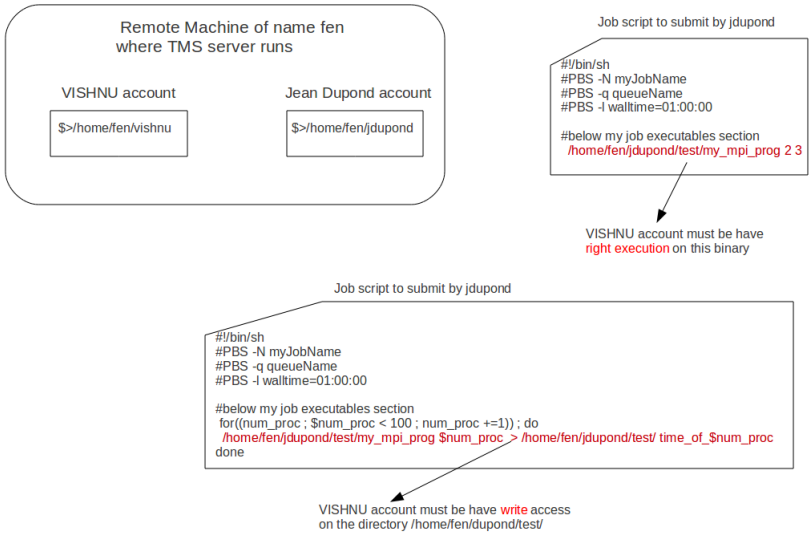

Figure 2.1: Example of binary execution execution problem by TMS server

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

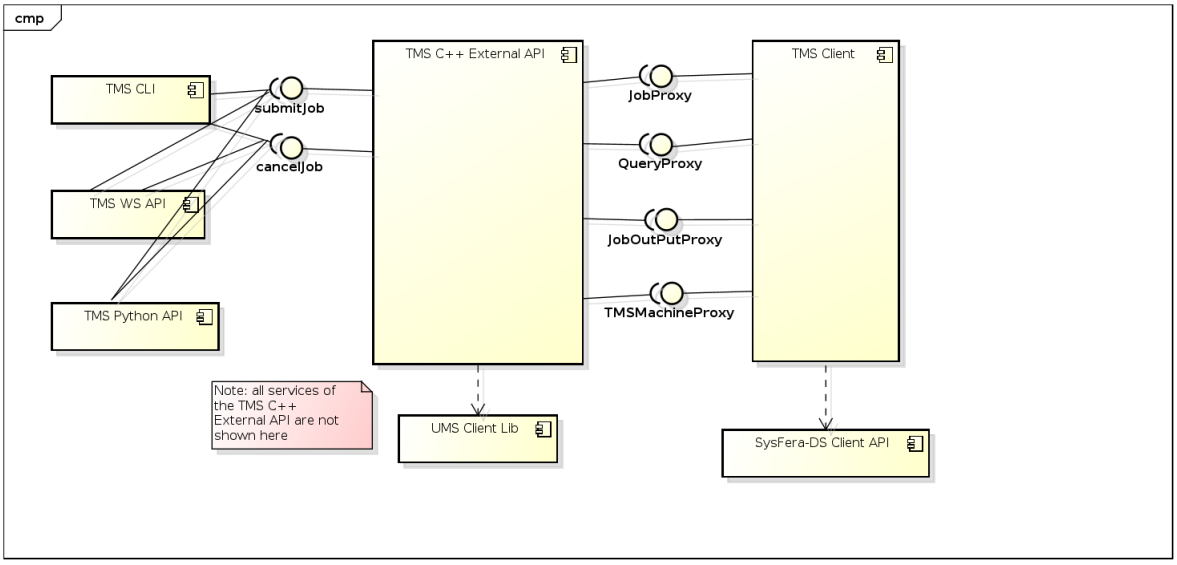

Figure 2.2: TMS client-side components

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. These services are consumed by the TMS Client component through the SysFera-DS API. All the interfaces of the TMS Server component are shown. The TMS Server component uses the Torque and IBM LoadLeveler API as underlying batch sheduler.

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

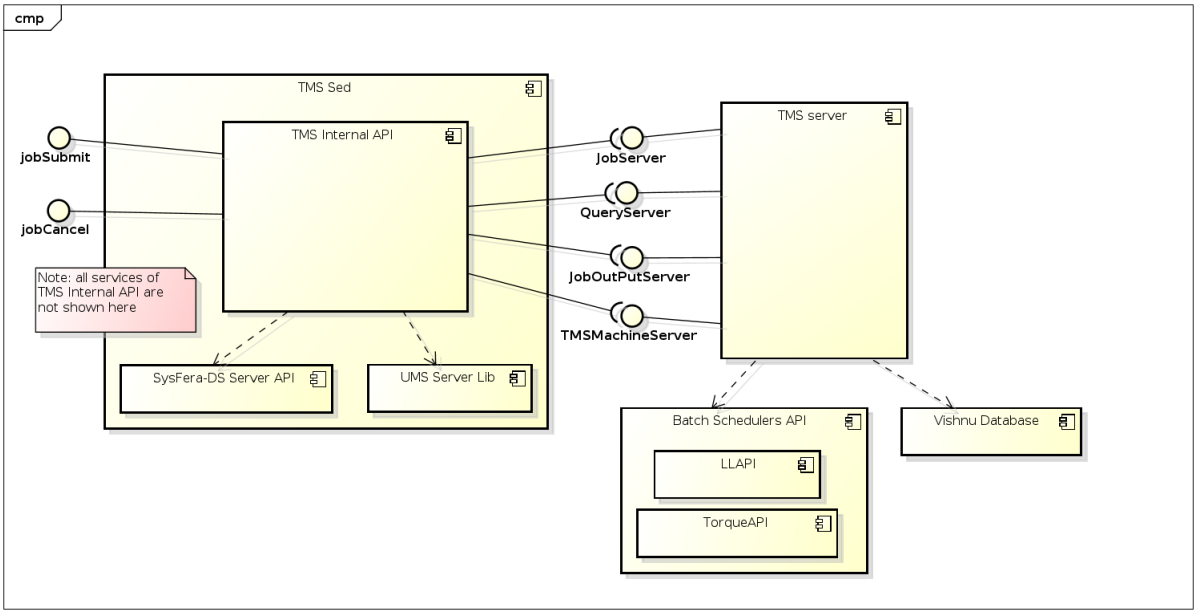

Figure 2.3: TMS server-side components

2.4.3 SysFera-DS Bus Details

This diagram shows the communication paths between the Client host and the TMS servers using the SysFera-DS 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 2.3 we have three TMS servers launched on three different nodes). The SysFera-DS MasterAgent is a SysFera-DS 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 the CORBA IIOP (Internet Inter-ORB) protocol and the communications can be tunneled through SSH tunnels if necessary. The MasterAgent entity is involved in the choice of the TMS server to call. The diagram shows here all the communication paths in the case where the TMS Server2 is chosen by the MasterAgent.

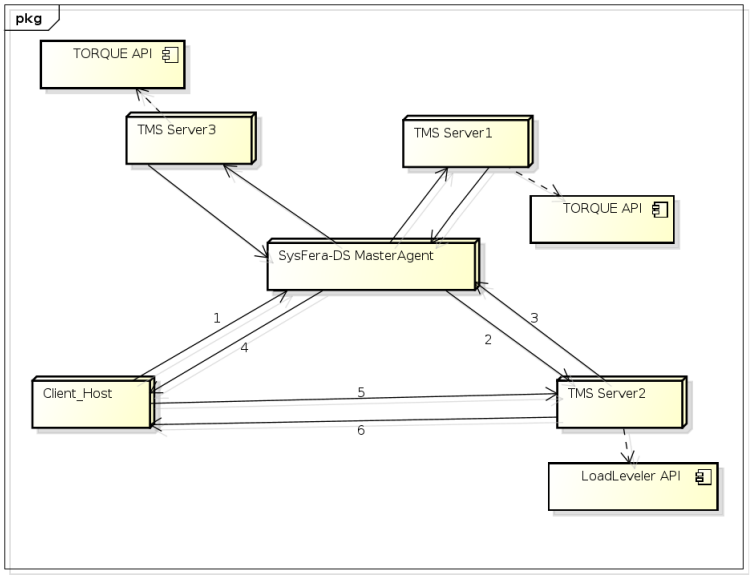

Figure 2.4: SysFera-DS Bus Details

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
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: scriptPath) 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 through the use of a script (scriptFilePath).

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_MACHINE	The machine is unknown
ERRCODE_BATCH_SCHEDULER_ERROR	Indicates an error caused by the underlying batch scheduler
ERRCODE_INVALID_PARAMETER	The provided parameter is invalid
ERRCODE_SESSION_KEY_NOT_FOUND	The sessionKey is unrecognized
ERRCODE_UNKNOWN_BATCH_SCHEDULER	Indicates that the batch scheduler type is unknown
ERRCODE_UNKNOWN_QUEUE	Indicates that the specified queue by the user is unknown
DB_ERROR	A problem occurs with the database
ERRCODE_UNKNOWN_ERROR	The type of error is unknown

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

Parameter	Type	Serialized type	Description	Mode
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 from its id

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_MACHINE	The machine is unknown
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey has expired. The session is closed.
ERRCODE_SESSION_KEY_NOT_FOUND	The sessionKey is unrecognized
ERRCODE_BATCH_SCHEDULER_ERROR	Indicates an error caused by the underlying batch scheduler
ERRCODE_INVALID_PARAMETER	The provided parameter is invalid
ERRCODE_CAN_NOT_CANCEL	Can not cancel this job
ERRCODE_PERMISSION_DENIED	Indicates the requested operation is not allowed for provided user
DB_ERROR	A problem occurs with the database
ERRCODE_UNKNOWN_JOBID	The job id is unknown

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

TMS::cancelJob

3.2.3 Service getInfoOfJob_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 getInfoOfJob_MACHINEID() function gets information on a job from its id

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

Name	Description
ERRCODE_SESSION_KEY_NOT_FOUND	The sessionKey is unrecognized
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey has expired. The session is closed.
ERRCODE_INVALID_PARAMETER	The provided parameter is invalid
ERRCODE_UNKNOWN_MACHINE	The machine is unknown
ERRCODE_UNKNOWN_BATCH_SCHEDULER	Indicates that the batch scheduler type is unknown
DB_ERROR	A problem occurs with the database
ERRCODE_UNKNOWN_ERROR	The type of error is unknown
ERRCODE_UNKNOWN_JOBID	The job id is unknown

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

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_SESSION_KEY_NOT_FOUND	The sessionKey is unrecognized
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey has expired. The session is closed.
ERRCODE_INVALID_PARAMETER	The provided parameter is invalid
ERRCODE_UNKNOWN_MACHINE	The machine is unknown
ERRCODE_UNKNOWN_BATCH_SCHEDULER	Indicates that the batch scheduler type is unknown
ERRCODE_UNKNOWN_ERROR	The type of error is unknown
DB_ERROR	A problem occurs with the database

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

TMS::listJobs

3.2.5 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

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_MACHINE	The machine is unknown
ERRCODE_SESSION_KEY_NOT_FOUND	The sessionKey is unrecognized
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey has expired. The session is closed.
ERRCODE_INVALID_PARAMETER	The provided parameter is invalid
ERRCODE_BATCH_SCHEDULER_ERROR	Indicates an error caused by the underlying batch scheduler
ERRCODE_UNKNOWN_ERROR	The type of error is unknown
DB_ERROR	A problem occurs with the database
ERRCODE_UNKNOWN_JOBID	The job id is unknown

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

TMS::getJobProgress

3.2.6 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
job	string	Job	The Job object containing the input information (ex: jobId) and ouput information (ex: outputPath an errorPath)	OUT
errorInfo	string	n/a	Additional information provided when an error code is returned	OUT

Description

The jobOutPutGetResult_MACHINEID() function gets outputPath and errorPath of a job from its id

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_SESSION_KEY_NOT_FOUND	The sessionKey is unrecognized
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey has expired. The session is closed.
ERRCODE_INVALID_PARAMETER	The provided parameter is invalid
ERRCODE_UNKNOWN_MACHINE	The machine is unknown
ERRCODE_PERMISSION_DENIED	Indicates the requested operation is not allowed for provided user
ERRCODE_UNKNOWN_ERROR	The type of error is unknown
DB_ERROR	A problem occurs with the database
ERRCODE_UNKNOWN_JOBID	The job id is unknown

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

TMS::getJobOutput

3.2.7 Service jobOutPutGetAllResults_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

Description

The jobOutPutGetAllResults_MACHINEID() function dynamically gets outputPath and errorPath of completed jobs

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_SESSION_KEY_NOT_FOUND	The sessionKey is unrecognized
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey has expired. The session is closed.
ERRCODE_INVALID_PARAMETER	The provided parameter is invalid
ERRCODE_UNKNOWN_BATCH_SCHEDULER	Indicates that the batch scheduler type is unknown
ERRCODE_UNKNOWN_ERROR	The type of error is unknown
DB_ERROR	A problem occurs with the database

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

TMS::getAllJobsOutput

3.2.8 Service TMSMachineGetListOfQueues_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
listofQueues	string	ListQueues	The list of queues	OUT
errorInfo	string	n/a	Additional information provided when an error code is returned	OUT

Description

The TMSMachineGetListOfQueues_MACHINEID() function gets queues information

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_PARAMETER	The provided parameter is invalid
ERRCODE_UNKNOWN_MACHINE	The machine is unknown
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey has expired. The session is closed.
ERRCODE_SESSION_KEY_NOT_FOUND	The sessionKey is unrecognized
ERRCODE_BATCH_SCHEDULER_ERROR	Indicates an error caused by the underlying batch scheduler
DB_ERROR	A problem occurs with the database
ERRCODE_UNKNOWN_ERROR	The type of error is unknown

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

TMS::listQueues

3.2.9 Service TMSMachineRefreshPeriodSet

Access

This service can be used by ADMIN users only

Parameters

Parameter	Type	Serialized type	Description	Mode
sessionKey	string	n/a	The Session object which contains the session input information (ex: the session key which 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 set refresh period	IN
value	int	n/a	Is the refresh interval value (in seconds)	IN
errorInfo	string	n/a	Additional information provided when an error code is returned	OUT

Description

The TMSMachineRefreshPeriodSet() function sets the refresh period of output and error files contents

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_MACHINE	The machine is unknown
ERRCODE_SESSION_KEY_NOT_FOUND	The sessionKey is unrecognized
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey has expired. The session is closed.
ERRCODE_UNKNOWN_ERROR	The type of error is unknown
ERRCODE_INVALID_PARAMETER	The provided parameter is invalid
DB_ERROR	A problem occurs with the database

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

TMS::setMachineRefreshPeriod

3.2.10 Service TMSMachineRefreshPeriodGet

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 refresh period	IN
errorInfo	string	n/a	Additional information provided when an error code is returned	OUT

Description

The TMSMachineRefreshPeriodGet() function gets the refresh period of output and error files contents

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_SESSION_KEY_NOT_FOUND	The sessionKey is unrecognized
ERRCODE_SESSIONKEY_EXPIRED	The sessionKey has expired. The session is closed.
ERRCODE_UNKNOWN_MACHINE	The machine is unknown
ERRCODE_INVALID_PARAMETER	The provided parameter is invalid
ERRCODE_UNKNOWN_ERROR	The type of error is unknown

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

TMS::getMachineRefreshPeriod

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.

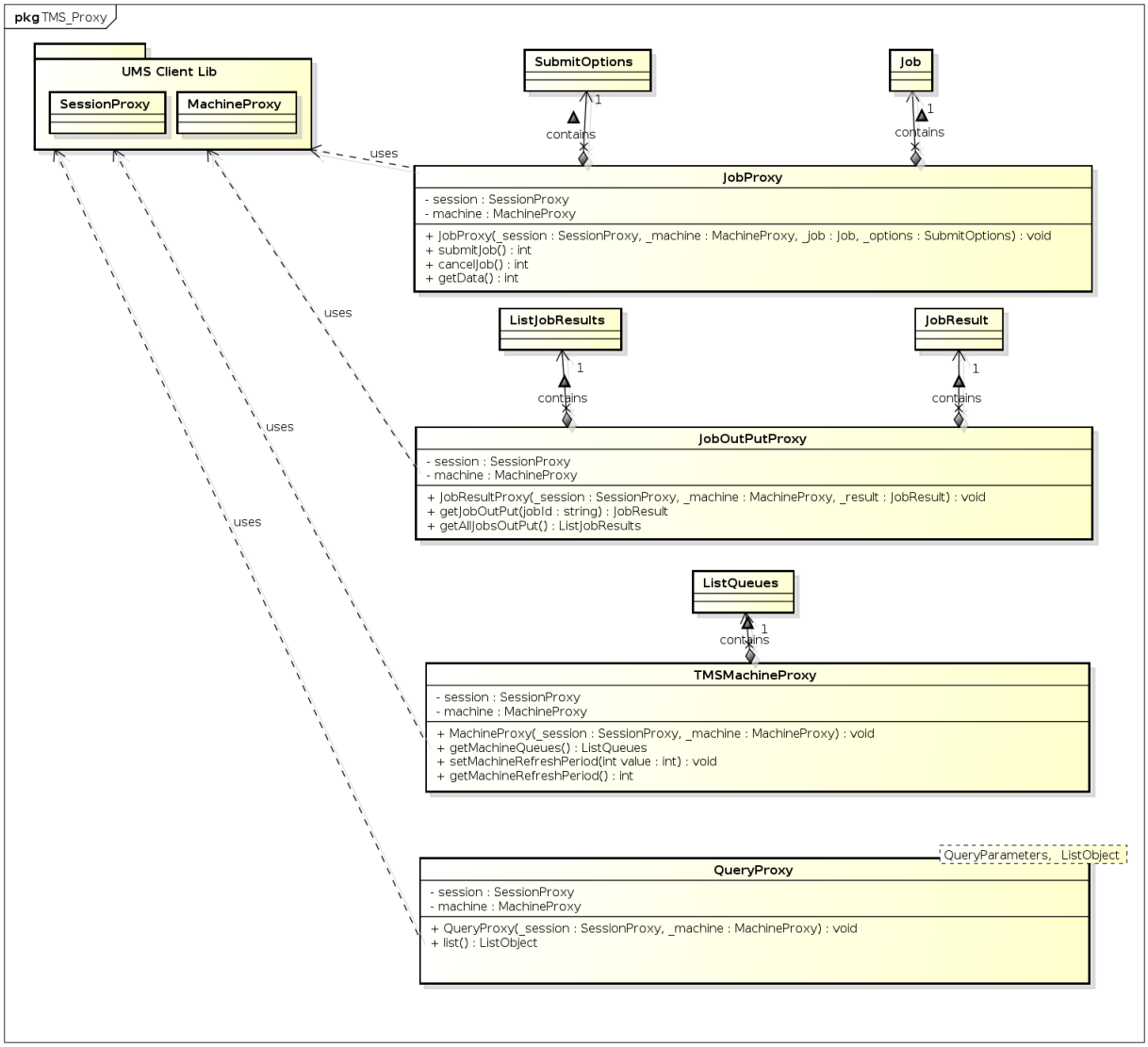

Figure 4.1: TMS Client Class Diagram

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.

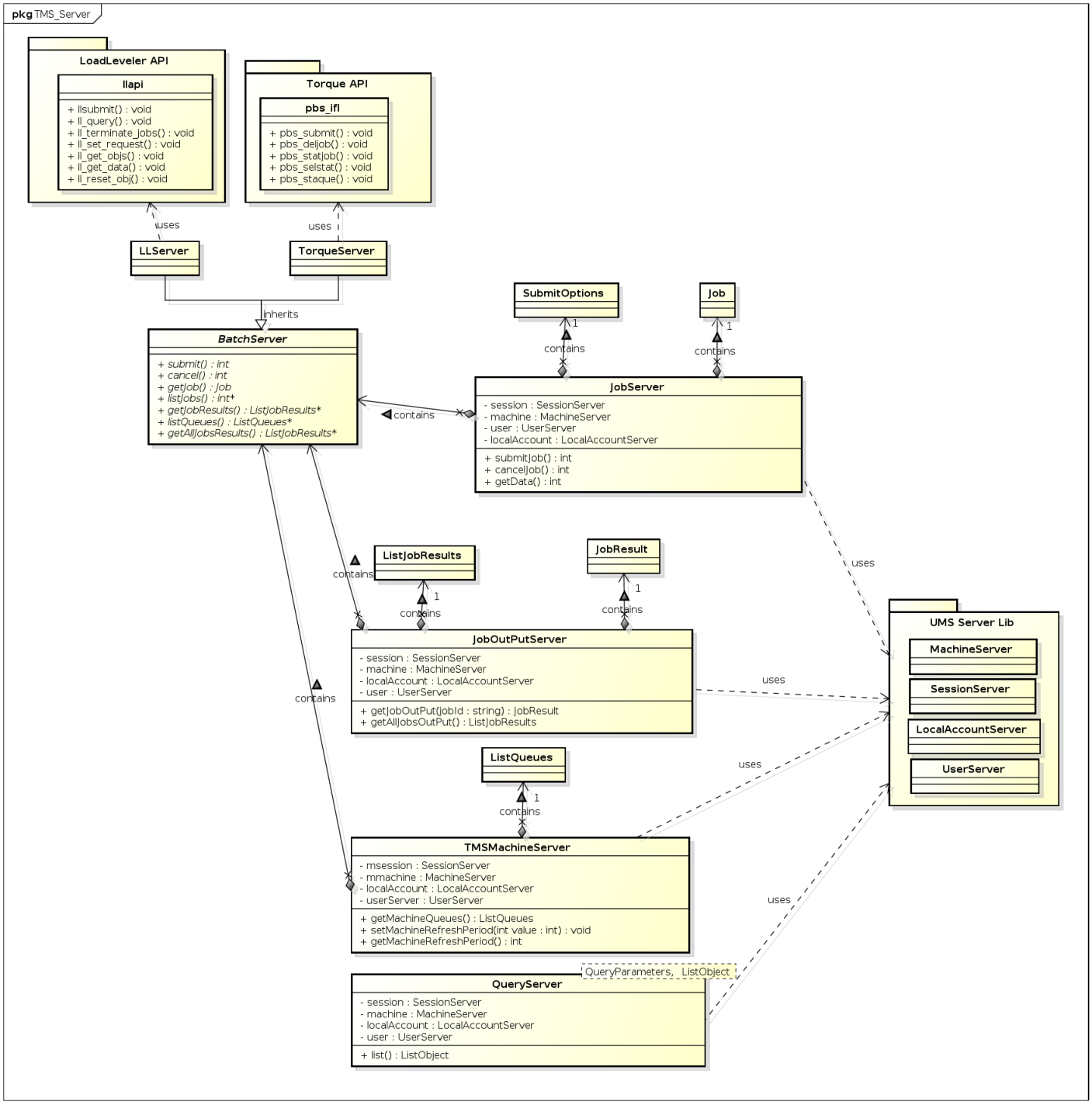

Figure 4.2: TMS Server Class Diagram

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.

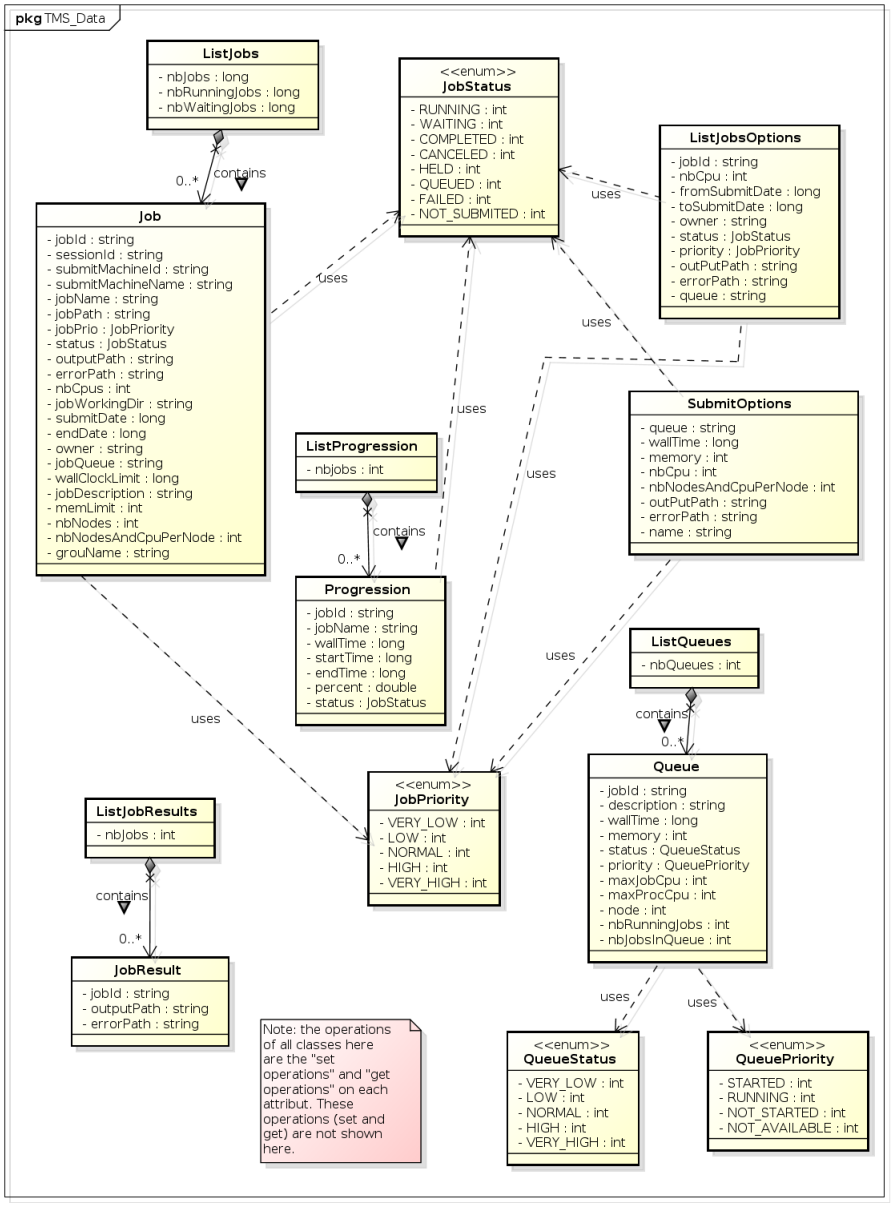

Figure 4.3: TMS Data Class Diagram