Diligent

A DIgital Library
Infrastructure on Grid
ENabled Technology

Deliverable No D1.2.3:

"DL Creation & Management services detailed design report"

September 2006

Document Information

Project	
Project Title:	DILIGENT, A **DI**gital **L**ibrary **I**nfrastructure on **G**rid **EN**abled **T**echnology
Project Start:	1st Sep 2004
Call/Instrument:	FP6-2003-IST-2/IP
Contract Number:	004260
Document	
Deliverable number:	D1.2.3
Deliverable title:	DL Creation & Management services detailed design report
Contractual Date of Delivery:	Month 17
Actual Date of Delivery:	18 October 2006
Editor(s):	CNR – ISTI
Author(s):	H. Avancini, M. Borriello, L. Candela, P. Fabriani, A. Manzi, P. Pagano, P. Roccetti, M. Simi, A. Turli
Reviewer(s):	UNIBAS
Participant(s):	CNR – ISTI, ENG
Workpackage:	WP1.2
Workpackage title:	DL Creation & Management
Workpackage leader:	CNR - ISTI
Workpackage participants:	CNR – ISTI, UoA, CERN, ENG
Est. Person-months:	35
Distribution:	Confidential
Nature:	Report
Version/Revision:	2.0
Draft/Final	Final
Total number of pages: (including cover)	244
File name:	D1.2.3-V2.0.doc/.pdf
Key words:	*Digital libraries, UML, Services Architecture, Services Design*

Disclaimer

This document contains description of the DILIGENT project findings, work and products. Certain parts of it might be under partner Intellectual Property Right (IPR) rules so, prior to using its content please contact the consortium head for approval.

In case you believe that this document harms in any way IPR held by you as a person or as a representative of an entity, please do notify us immediately.

The authors of this document have taken any available measure in order for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated the creation and publication of this document hold any sort of responsibility that might occur as a result of using its content.

This publication has been produced with the assistance of the European Union. The content of this publication is the sole responsibility of DILIGENT consortium and can in no way be taken to reflect the views of the European Union.

The European Union is established in accordance with the Treaty on European Union (Maastricht). There are currently 25 Member States of the Union. It is based on the European Communities and the member states cooperation in the fields of Common Foreign and Security Policy and Justice and Home Affairs. The five main institutions of the European Union are the European Parliament, the Council of Ministers, the European Commission, the Court of Justice and the Court of Auditors. (http://europa.eu.int/)

DILIGENT is a project partially funded by the European Union

Table of Contents

Table of Figures

Document Logs

Issue	Section	Date	Comment	Author
1.0	All	24/03/2006	First version. It contains the first version of the following components: DIS-IP, DIS-IC, DIS-R-GMAClient, DISHLSClient, DIS-Registry, DIS Util, Package Repository, DL Management, Hosting Node Manager, the DVOS Authorization Management, and the VDL Definition Repository.	D1.2.3 Team
1.1	3.2.1, 3.2.3	11/07/2006	Updated HNM, and Package Repository sections to reflect the actual implementation. A11, A12, A13 updated also.	Henri Avancini
1.1.1	3.2.2	14/07/2006	DLManament dependencies section updated to current implementation	Andrea Manzi
1.2	2.1, 2.2; 2.2.1, 2.2.2, 2.2.3 and all subsections;	05/08/2006	Sections completely revised to reflect the new implementations of the DIS-IC and DIS-IP components. Added the new 2.2.3 section with DILIGENTProvider description. All the subsequent sections shifted by one position.	Manuele Simi
1.2	Appendix A	05/08/2006	DIS-IC profile added to Appendix A	Manuele Simi
1.3	2.2.5 and all subsections;	28/08/2006	Updated DISHLSClient operations.	Andrea Manzi
1.3	Appendix C	28/08/2006	Added to Appendix C the DISQueries file	Andrea Manzi
1.4	2.2.6 and all subsections;	30/08/2006	Updated DISRegistry class diagrams, operations and descriptions	Andrea Manzi

1.5	2.2.7 section added, removed old 2.2.6	13/09/2006	Added DIS-Broker service section, removed old util package section	Andrea Manzi
1.5	Appendix A	13/09/2006	Removed RI Profile froms Appendix A Added DIS-Broker Service Profile and gLite (CE|SE|Service|Site) Resource profile. DIS-Registry, DLManagement, DIS-RGMA-Client service profiles updated.	Andrea Manzi
1.6	4.1	15/09/2006	DVOS Common - New Section	P. Roccetti
1.6	4.2	15/09/2006	Authentication Management - Updated introduction with authentication management general diagram Section 4.2.1.2.4 - Logical Operations - Addded naming conventions to define logical operations Section 4.2.2.1.1 - Profile - Added reference to the Authentication API profile Section 4.2.2.2.1 - Profile - Added reference to the Delegation service profile Section 4.2.2.2.5 - Security profile - Added logical operations defined by the Delegation service Section 4.2.2.3.1 - Profile - Added reference to the Credentials Renewal service profile Section 4.2.2.4 - CredentialsRenewal API - New Section	P. Roccetti
1.6	4.3	15/09/2006	Authorization management - Updated introduction with detailed description of services Section 4.3.1.1.1 - Profile - Added reference to the Authorization service profile Section 4.3.1.1.3 - Configuration - Updated configuration parameters table Section 4.3.1.2.2 - Implementation details - Added Command Line user interface description	P. Roccetti

1.6	4.4	15/09/2006	DVOS UserGroupManagement Service - New Section	Paolo Roccetti
1.7	5	15/09/2006	New Section	Marianna Borriello
2.0		26/09/2006	Unibas Revision	Heiko Schuldt, Christoph Langguth, Laura Cristiana Voicu
2.1	4	09/09/2006	4 DVOS (and subsections) – Updated to take the reviewer's comments into account 4.1 DVOS-VOMS interaction – New section 5 Broker and Matchmaker - – Updated to take the reviewer's comments into account	Paolo Roccetti
2.2	1, 2, 3, 6, 7 and all subsections	10/11/2006	Updated to take the reviewer's comments into account	Manuele Simi

Summary

The aim of this report is to present the detailed design of the DILIGENT services constituting the DL Creation & Management functional area. It represents the second revision of the third document of a series of reports dedicated to the design of these services and thus (i) it is based on the results of the previous design documents, i.e. D1.2.1 "DL Creation & Management services specification interim report" and D1.2.2 "DL Creation & Management services specification report", (ii) it takes into account the DILIGENT functional specifications reported in D1.1.1 "Test-bed functional specification", and (iii) it takes into account the DILIGENT software architecture presented in "D1.1.2 Architectural Specification" [1]. It is worth noting that the content of this report reflects the current status of the work in the DL Creation & Management functional area, i.e. it contains the detailed design of the components actually delivered, and it is subject to changes and improvements that will be highligheted in the Documents Log section.

Executive Summary

The objective of the D1.2.3 report is to present the detailed design of the services constituting the DL Creation and Management area of the DILIGENT project.

This report represents the second revision of the third document of a series of deliverables dedicated to the design phase of these services. In particular, the *D1.2.1 DL Creation & Management Services Specification interim report* [5] presenting the first specification of the five services belonging to this area analysed the functional specification and identified the DILIGENT functionality related to these services; the *D.1.2.2 DL Creation & Management Services Specification report* [6] introduced the specification of the services in terms of use-case, logical, and deployment views capturing their main characteristics from the service specification perspectives; D.1.2.3 introduces the identified components in terms of the detailed design perspective acting as a blueprint for software developers. The identified components and the related characteristics are integrated with the specification of the interfaces, the algorithms, the data structures, the data flows and any other detail deemed as relevant to communicate to the application developer the expected software characteristics.

Exploiting the *D.1.2.2 DL Creation & Management Services Specification report* and the *DL Creation & Management Implementation Plan*, the detailed design and the implementation phases are conducted in parallel. Thus, the detailed design specification anticipates and follows the results of the implementation exploiting the results of the early experimentation of the implemented components and reacting with the appropriate adjustments to underestimated or unknown problems related for example to scalability, reliability, and robustness issues. The major effort required in the production of this *on-going* report is thus justified by the rapidity of the implementation to react to the needs of the project.

More precisely, the detailed design phase is divided into sub phases dedicated to produce the design of the components according to the well-established implementation plan. These smallest components are then developed and tested. In the following phases, they will be enriched with the novel designed features in order to form the full-fledged expected component at the end of the development process. Consequently, this report contains the detailed design of those components and those features that have been indicated in the *DL Creation & Management Implementation Plan* as the ones that form the first exploitable release of the Collective Layer. It serves as concrete blueprint for software developers since it represents a valid picture of the software.

The structure of this document has been defined in order to integrate the information provided in *D.1.2.2 DL Creation & Management Services Specification report* with the detailed design choices arising in implementing the expected components in terms of the Service Oriented Architecture paradigm, extended with the technologies of the Web Services Resource Framework, and constrained by the gLite Grid Middleware [15] and by the other open source toolkits, e.g. the Globus Toolkit [22], the Apache Xerces, Xalan, and Xindice [3].

Having clarified these aspects, the report presents:

- the detailed design of all the components constituting the DILIGENT Information Service, i.e. the DIS-IP, the DIS-IC, the DIS-RGMAClient, the DISHLSClient, the DIS-Registry, and DILIGENTProvider;
- the detailed design of the Package Repository, the DL Management, and the Hosting Node Manager as part of the DILIGENT Keeper Service;
- the detailed design of the Dynamic VO Support Service;

- the draft version of the detailed design of the Broker & Matchmaker Service;
- the detailed design of the VDL Definition Repository as part of the VDL Generator Service.

1 INTRODUCTION

The DILIGENT system engineering process is conducted according to the guidelines of the Unified Process methodology. Following this methodology, the architecture of a complex software system is the organization or structure of the system's significant components interacting, through interfaces, with components composed of successively smaller components and interfaces.

This report presents the result of the activity conducted within the design tasks[1] of the *WP1.2 DL Creation & Management* and partially in the context of the implementation tasks[2] of the same WP. The design phase of the WP1.2 spreads out over 12 months and it is further organized into three sub-phases whose final goal is to produce a detailed service specification capable to guide the services implementation phase. The final goal is achieved by producing three reports, each adding further details to the services specification. These reports are respectively (i) the interim report D1.2.1 [5], (ii) the *D1.2.2 DL Creation & Management services specification report* [6], and finally (iii) the *D1.2.3 DL Creation & Management services detailed design report* (this document).

In the *DL Creation & Management* area, there are services in charge to bring together all the resources distributed across a Grid infrastructure and support the creation of new Digital Libraries. The capabilities provided by the these services enable a virtual research organization to dynamically create and modify its own DLs by specifying a set of definition criteria including a number of requirements on the information space (e.g. publishing institutions, subject of the content, document type, level of replication) and on the services (e.g. service type, configuration, lifetime, availability, response time, others) expressing its needs.

In particular, the DL Creation & Management services provide the functionality allowing creating, configuring, monitoring, maintaining, and disposing a DL. This functionality includes:

- the creation of a trusted environment that ensures the controlled sharing of the available resources by exploiting the VO mechanisms provided by the gLite middleware and the development framework (*Dynamic VO Support Service*)
- the implementation of a global strategy that offers a valuable use of the resources supplied by the DILIGENT infrastructure (*Broker & Matchmaker Service*)
- the selection and automatic configuration of the best pool of resources (collections, services, and nodes) forming DLs that fulfill the particular (and possibly temporal) needs of end-user communities (*VDL Generator Service*)
- the orchestration needed to maintain up and running the pool of resources that populates the various DLs and to ensure measurable levels of fault tolerance and QoS (*Keeper Service*)

Finally, the DILIGENT Information Service supports all the other listed functionality providing a complete set of functionality allowing monitoring and discovering of the resources information.

[1] T1.2.1.a Information Service design, T1.2.2.a Broker & Matchmaker Service design, T1.2.3.a Keeper Service design, T1.2.4.a Dynamic VO Support Service design, and T1.2.5.a VDL Generator Service design.

[2] T1.2.1.b Information Service implementation, T1.2.2.b Broker & Matchmaker Service implementation, T1.2.3.b Keeper Service implementation, T1.2.4.b Dynamic VO Support Service implementation, and T1.2.5.b VDL Generator Service implementation.

The overall picture of the DL Creation & Management services and their relationships are reported in the *D1.1.2 Architectural Specification* [1].

The outline of this report is as follows:

- the remaining part of this section contains the rationale of this report and presents the outline of the expected information for each service;
- Section 2 presents the detailed design of the DILIGENT Information Service components;
- Section 2.2.6 presents the detailed design of the Keeper Service components;
- Section 4 presents the detailed design of the Dynamic VO Support Service components;
- Section 5 introduces a draft version of the detailed design of the Broker & Matchmaler Service components;
- Section 6 reports on the detailed design choices about the VDL Generator components;
- Section 7 concludes reporting the follow up of this report.

1.1 Rationale of Detailed Design report

The objective of a detailed design report is to introduce and document in detail the design of a software system. In general, these reports have two audiences. The first one includes engineers and designers interested in how the system works and how effective it is, i.e. (i) the other project partners involved in designing the services they are responsible for that must correctly design the possible service to service interactions, (ii) the DILIGENT designers aimed to discover drawbacks and novel opportunities in delivering the expected functionality, (iii) the research communities interested in the solutions adopted to draw on the ideas present here for future improvement in their own research fields. The second audience includes the system developers that are in charge to concretely realise the components envisaged by the component designers.

Due to these characteristics, the resulting document is mainly a technical document containing a high number of technicalities. It thus intends to satisfy more the information needs of the project technicians than those of other users, e.g. the DILIGENT user communities.

1.2 Document Structure

For each service of the DL Creation & Management Area, an introduction is provided with the aim to briefly recall the role of the service in the DILIGENT project. Then, the components forming the service are presented with their respective technical solutions.

The following two subsections present the structure of the sections belonging the two types of components: *Service* and *Library*.

1.2.1 Service Section Layout

A brief introduction describes the service that corresponds to a WSRF compliant Web Service. The following sections are provided:

- Profile

This section reports the DILIGENT Resource information in accordance with the DILIGENT Resource Model introduced in D1.1.2 deliverable [1].

- Managed Resources

This section presents details on the managed WS-Resource(s); how these resources are created, persisted, etc. It also indicates the resource properties exposed by the service (if any).

- Operations

The section introduces the operations and, for each of these, its signature and description. The operations are those exposed by the WSDL file and the signature refers their automatically generated stubs classes.

- How to use the component

The section gives some technical hints about how to interact with the service using its stubs library.

- Implementation Details

This section presents implementation details such as a description of the implementation classes, the chosen algorithms and any third party software components used to support the implementation of the service.

- Dependencies

This section reports on third party software needed to correctly operate the service.

- Configuration

This section documents the aspects of the service that are customisable as well as the procedures required to perform such personalization tasks.

- Security Profile

This section reports aspects related to the service authorization characteristics in accordance with the authorization framework presented in Sections 4.3.

- Known Bugs and Limitations

This section documents the known bugs and limitations the current implementation suffers from. In particular, the limitations section is dedicated to highlight the differences between the expected functionality and the one provided by the current implementation.

1.2.2 Library Section Layout

A brief introduction provides important details on the usage of the Library. Then the following sections are provided:

- Profile

This section recalls in which profiles the information about the library are reported.

- Operations

This section documents the library's methods providing for each of them its signature and description.

- How to use the component

The section gives some technical hints about how to use the library.

- Implementation Details

This section reports implementation details such as the used algorithms and the third party software components used to implement the library.

- Dependencies

This section reports on third party components needed to correctly operate the software library.

- Configuration

This section documents the aspects of the software library that are customisable as well as the procedure required to exploit the capability.

- Known Bugs and Limitations

This section documents the known bugs and limitations the current implementation suffers from. In particular, the limitations section is dedicated to highlight the differences between the expected functionality and the one provided by the current library implementation.

2 INFORMATION SERVICE

2.1 Introduction

The DILIGENT Information Service (DIS) maintains the most up to date information about the set of available distributed resources that compose the DILIGENT VO together with the status of the DILIGENT services.

It supports the construction of new DLs by providing information about the available DILIGENT Resources. Once a DL is up, the services forming the DL use the DIS to publish their own specific data and consume the information handled to discover other services to interact with.

The support for the DL generation phase is provided via the management of the DILIGENT Resource profiles. These profiles allow describing Services, RunningInstances, DHNs, Collections, CSs, CSInstances and gLite resources following the XML Schemas prepared in the context of the WP1.2[3]. Some of them are automatically generated by the Keeper components while others, such as the Service profiles, have to be manually prepared and registered by the implementers.

From an implementation point of view, the DIS is able to manage information exposed as Resource Property (RP) [18]. The exploitation of the WS-ResourceProperties and WS-ServiceGroup [19] specifications (part of the WSRF specifications family) is the key of the DIS technical solution presented here.

The DIS builds on top the *Aggregator Framework* [23] and relies on the *operation provider* mechanism both implemented in the context of the Globus Project.

By implementing the WS-ServiceGroup specification, the Aggregator Framework allows to build services (called *aggregator services*) that collect and aggregate information. The communication flow is based on the concepts of *Aggregator Source* and *Aggregator Sink*.

The Sources are the producers of data and the Sinks are the collectors of data. The aggregator framework collects data from Aggregator Sources and sends that data to Aggregator Sinks for processing. A Source is always connected to one or more Sinks through a registration. When a new Aggregator Source is registered in an aggregator service, a new Aggregator Sink connected with that source is created by the service.

Aggregator services are self-cleaning, i.e. each registration has a lifetime: if a registration expires without being refreshed, it is removed from the aggregator service. Aggregator Sources distributed with the Java WS Core include modules that query service data, acquire data through subscription/notification, and execute programs to generate information. Available Aggregator Sinks distributed with the framework include modules that implement the WS Index service interface and the WS Trigger service interface. Figure 1 presents a schema of the framework.

[3] https://elibrary.isti.cnr.it/svn_public/diligent_GAR/DILIGENTCommon/DILResourceSchemas/

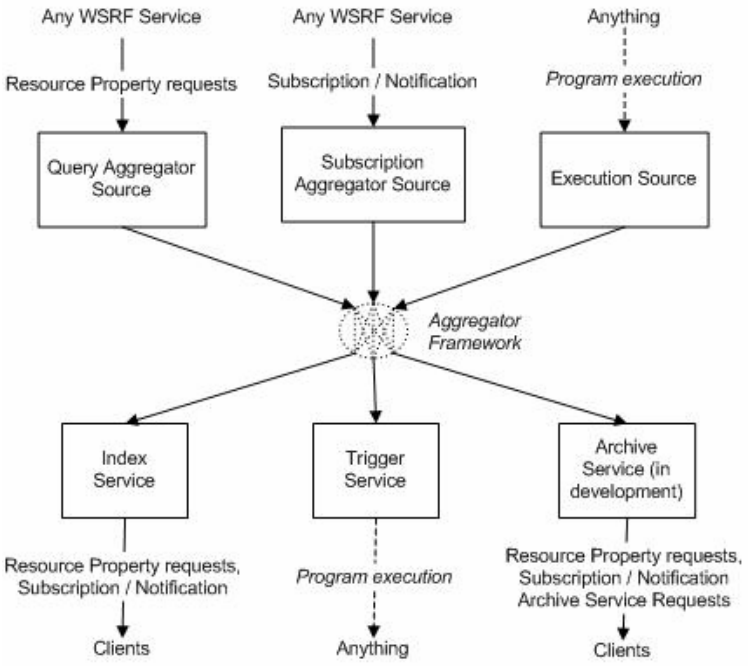

Figure 1. Aggregator Framework: Information Flow

As depicted, a possible kind of Aggregator Source is the WS-Resource Properties and this type of source are used within the DIS. Thanks to specific DIS components, any DILIGENT Service becomes an Aggregator Source. Other DIS components, implemented as aggregator service, collect information from these sources and make them available through an appropriate query interface to consumers.

The DIS Aggregator Sinks acquire information through an extensible interface that can be used to query WSRF services for resource properties. This interface is automatically added to any service by declaring the appropriate *operation providers* in the WSDL and WSDD files. An operation provider is a software module that plugs operations and WS-Resource Properties into a service. A number of standard providers come with Java WS Core allowing to add to a service the needed interface to permit the DIS Aggregator Sinks to query its WS-Resource Properties. In addition, a custom provider, named *DILIGENTProvider*, has been developed to add custom resource properties and further operations to be used for DILIGENT purposes.

2.2 Components

The DILIGENT Information Service is composed by the following components:

- DIS-IP (Library) – The DIS-IP is responsible for registering/unregistering a group of resource properties as Aggregator Source to one or more DIS-ICs. It also allows to register/unregister groups of Topics in the DIS-Broker. A DIS-IP is always installed on each DHN.
- DIS-HLSClient (Library) – The DIS-HLSClient is a library used by DILIGENT services to access the information maintained by the DILIGENT Information Service. Using a DIS-HLSClient, it is possible to query a DIS-IC to discover Profiles or WS-Resource properties. A DIS-HLSClient is always installed on each DHN.

- DIS-Cache (WSRFService) – This service is in charge of building and maintaining a local image of the information that is globally available on the various DIS-IC instances. This service will be available in the Beta release.
- DIS-IC (WSRFService) – This service is the Information Collector (IC) of all the data published in the DIS. It is implemented as Aggregator Sink that collects RPs from the registered (via DIS-IP) Aggregator Sources.
- DIS-R-GMAClient (WSRFService) – This service is in charge of harvesting resource information from the R-GMA Server[4] it has been configured to interact with. The gathered information is manipulated in order to make it compliant with the schema adopted in DILIGENT. Then such information is published as WS-Resources via the DIS-IP and as a DILIGENT Resource of type gLiteResource using the mechanism offered by the DIS-Registry Service.
- DIS-Registry (WSRFService) – This service provides registration and un-registration facilities for the DILIGENT resources profiles. A detailed description of the information contained these profiles is reported in D1.1.2 [1].
- DILIGENTProvider (Library) – This operation provider adds resource properties to the group of properties registered by a service in the DIS-IC. This additional information allows enlarging the spectrum of functionalities offered to identify the source that publishes the data and to perform fine-grained queries.
- DIS-Broker (WSRFService) – This service provides registration/unregistration of Topics (events to be notified on) for DILIGENT Services. This allows clients to subscribe to/unsubscribe from topics without having to know the physical locations of the services that expose them.

Figure 2 shows how the DIS-IP, DIS-HLSClient and DIS-IC play their role in the process of publication and retrieval of information within the DILIGENT Information System.

[4] The R-GMA service is the gLite service playing the same role as the DIS in the context of the gLite infrastructure. Thus, this service provides the list of and the information about the available gLite resources.

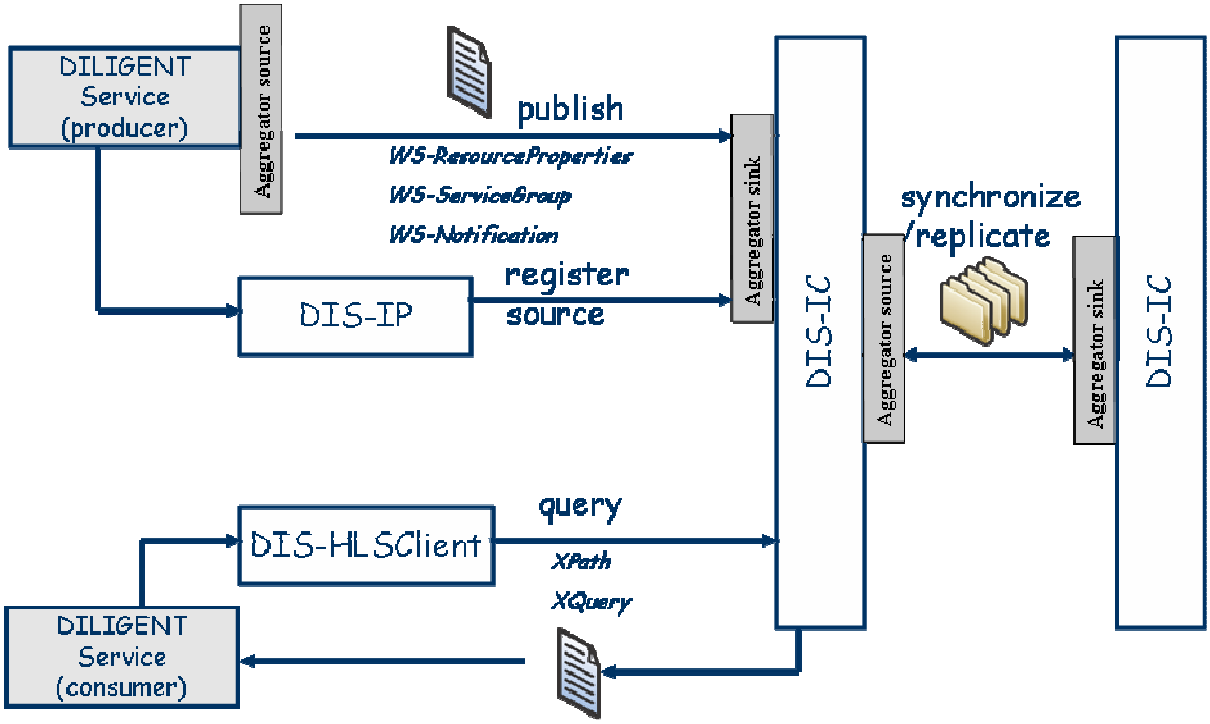

Figure 2. DIS: publish/retrieve schema

Regarding the deployment model, globally, all these components form a distributed DIS architecture. Some of them must reside on each DHN, while others can be distributed on the available hosts over the network.

An example of a possible distribution of these components is depicted in Figure 3.

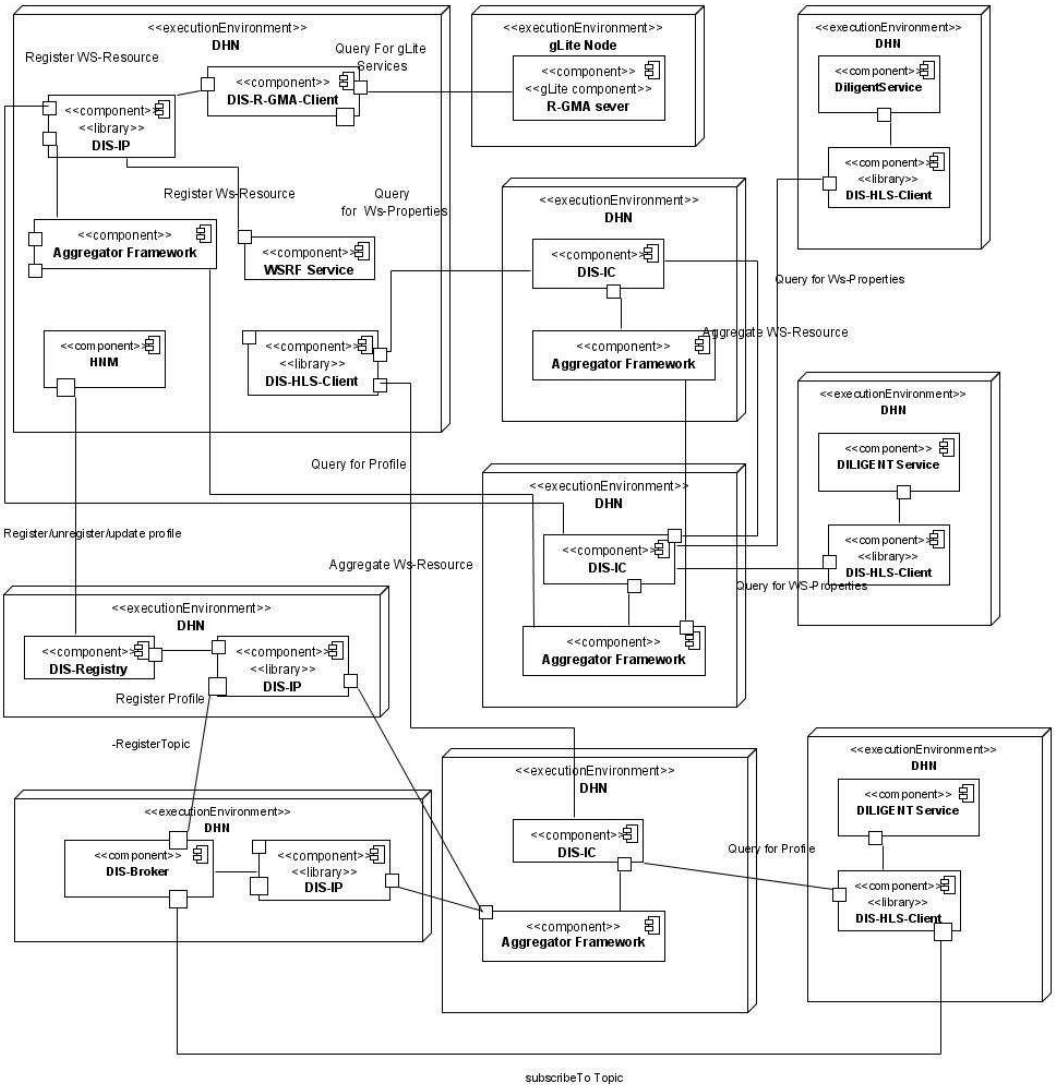

Figure 3. DIS: Main Components and relations

Details about each component are provided in the following sections.

2.2.1 DIS-IP

The DIS-IP is the entry point to feed information in the DIS. It allows to

- publish WS-ResourceProperties and DILIGENT Resource Profiles in the DIS-IC;
- register new Topics in the DIS-Broker (see Section 2.2.7).

In the former case, it transforms a group of resource properties into an Aggregator Source that can interact with the DIS-IC by exploiting the Aggregator Framework facilities. Resource properties can be registered either in pull or push mode (see Section 2.2.1.2). After a registration, they become an Aggregator Source for one or more remote DIS-IC instances (acting as aggregator services). These DIS-IC instances collecting the data make them available to other DILIGENT services.

The DIS-IP works in a strict collaboration with the DILIGENTProvider.

Moreover, it is a DHN mandatory component and this means than it is always available on each node of the infrastructure and can be used by the local services.

2.2.1.1 Profile

The DIS-IP is distributed as part of the DIS-IC component. See Section **2.2.2.1** for the profile that includes the DIS-IP description.

2.2.1.2 Operations

All the operations exposed by the DIS-IP are accessible as static methods of the DIS-IP.

The operation of the DIS-IP can be grouped in two sets: those that allow to manage the registration of resource properties into one or more DIS-IC instances and those that allow to register a list of Topics into a DIS-Broker instance.

Resource properties registration

Registrations can be *Named* or *Anonymous*. Named Registrations permit multiple registrations from the same EndpointReference, allowing a more fine-grained level of registration options. For example, it is possible to register some properties with a certain polling interval and others with a different one, if it is expected that they change with at different timing. On the other hand, Anonymous Registrations register all the resource properties with the same options since only one Registration can be active at the same time from an EndpointReference.

It is important to point out that, if a WS-Resource implements the `PersistentResource` interface for persistence operations, the Registration has to be renewed in the `load()` method each time the resource is restored from its persistent state (see Section 2.2.1.3).

There are two ways to register a group of resource properties: the Pull Mode and the Push Mode. In the Pull Mode, the group of registered properties is periodically polled by the remote DIS-ICs. To use this mode, a registration XML string has to be created and its content has to be passed to the registration operations (described below). It mainly includes the polling time, the refresh interval time, and the name of the properties to publish. The properties are self-cleaning; this means that if they are not renewed they are automatically removed from the DIS content. The DIS-IP, using the refresh interval period specified in the registration file, automatically performs the renewal operation. It is important to point out that, if the `RefreshIntervalSecs` is not specified in the registration file, the resources will never expire and they will be never removed from the DIS. The syntax of the registration file to provide for Pull Mode registration is reported in Section 2.2.1.4.3.

The Push Mode will be supported in the Beta release.

registerPullMode

static void registerPullMode (

org.apache.axis.message.addressing.EndpointReferenceType epr

java.lang.String xmlPullFile,

org.diligentproject.common.provider.DILIGENTPropertySet propSet)

It registers and publishes a group of WS-Resource properties into the DIS using the Pull Mode by creating an Anonymous Registration.

Anonymous Registrations allow to register only a group of resource properties for each WS-Resource (or service), since the `epr` is used as unique key to identify the registration.

After the registration, the set of registered properties becomes an Aggregator Source periodically polled by one or more DIS-IC instances.

registerPushMode

static void registerPushMode (

org.apache.axis.message.addressing.EndpointReferenceType epr

java.lang.String xmlPushFile,

org.diligentproject.common.provider.DILIGENTPropertySet propSet)

This operation will be supported in the Beta release.

registerPullMode

static void registerPullMode (

org.apache.axis.message.addressing.EndpointReferenceType epr,

java.lang.String xmlPullFile,

org.diligentproject.common.provider.DILIGENTPropertySet propSet,

java.lang.String name)

It registers and publishes a group of WS-ResourceProperties into the DIS using the Pull Mode by creating a Named Registration.

As stated, using names for registrations allows to register different groups of properties (possibly with different polling times or different modalities) from the same WS-Resource.

registerPushMode

static void registerPushMode (

org.apache.axis.message.addressing.EndpointReferenceType epr,

java.lang.String xmlPushFile,

org.diligentproject.common.provider.DILIGENTPropertySet propSet,

java.lang.String name)

This operation will be supported in the Beta release.

registerPullModeProfile

static void registerPullModeProfile (

org.apache.axis.message.addressing.EndpointReferenceType epr,

java.lang.String xmlPullFile,

java.lang.String type)

It registers and publishes a group of WS-ResourceProperties corresponding to a DILIGENT profile into the DIS using the Pull Mode. This operation is used only by the DIS-Registry service to register profiles in the DIS-IC.

registerPushModeProfile

static void registerPushModeProfile (

org.apache.axis.message.addressing.EndpointReferenceType epr,

java.lang.String xmlPushFile,

java.lang.String type)

This operation will be supported in the Beta release

remove

static void remove (

org.apache.axis.message.addressing.EndpointReferenceType epr)

It unregisters a group of WS-ResourceProperties from the DIS by destroying the related Anonymous Registration.

remove

static void remove (

org.apache.axis.message.addressing.EndpointReferenceType epr,

java.lang.String name)

It unregisters a group of WS-ResourceProperties from the DIS by destroying the related Named Registration.

removeProfile

static void removeProfile (

org.apache.axis.message.addressing.EndpointReferenceType epr,

java.lang.String diligentID, java.lang.String type)

It removes the DILIGENT Profile identified by the parameter diligentID from the DIS. The `epr` identifies the resource that publishes the profile as its property. The parameter type specifies the type of the resource to which the profile belongs (e.g. DHN, RunningInstance, Service, etc). Only the DIS-Registry uses this operation.

Topics registration

The Topics registration operations allow to interact with the DIS-Broker to register and unregister a list of Topics. See the DIS-Broker section (2.2.7) for a detailed explanation about the use of Topics in DILIGENT.

registerToBroker

static void registerToBroker(

org.apache.axis.message.addressing.EndpointReferenceType epr,

ArrayList<javax.xml.namespace.QName> topicsList)

It registers the list of given topics published by the given `epr` to the DIS-Broker.

Example:

```
ListQname = new ArrayList<QName>();
listQname.add(DILIGENTResourceQNames.RP_RICOUNTERRP);
listQname.add(DILIGENTResourceQNames.RP_EXTERNALRICOUNTERRP);
listQname.add(DILIGENTResourceQNames.RP_SERVICECOUNTERRP);
listQname.add(DILIGENTResourceQNames.RP_COLLECTIONCOUNTERRP);
listQname.add(DILIGENTResourceQNames.RP_CSCOUNTERRP);
```

```
listQname.add(DILIGENTResourceQNames.RP_CSINSTANCECOUNTERRP);
listQname.add(DILIGENTResourceQNames.RP_DHNCOUNTERRP);
listQname.add(DILIGENTResourceQNames.RP_GLITESECOUNTERRP);
listQname.add(DILIGENTResourceQNames.RP_GLITECECOUNTERRP);
listQname.add(DILIGENTResourceQNames.RP_GLITESITECOUNTERRP);
listQname.add(DILIGENTResourceQNames.RP_GLITESERVICECOUNTERRP);

DISIP.registerToBroker(endpoint,listQname)
```

unregisterToBroker

static void unregisterToBroker(org.apache.axis.message.addressing.EndpointReferenceType epr, ArrayList<javax.xml.namespace.QName> topicsList)

It unregisters the list of given topics published by the given `epr` from the DIS-Broker.

2.2.1.3 How to use the component

If a service wants to register its own Topics to the DIS-Broker, it just needs to invoke the `registerTopics()` and `unregisterTopics()` as explained in the previous section.

Otherwise, services that want to publish their resource properties using the DIS-IP must follow these steps:

WSDL file modifications

In the WSDL, a service needs to add the following namespace mapping:

```
xmlns:diligent="http://diligentproject.org/namespaces/common/provider/DILIGENTProvider"
```

and `import DiligentProvider.wsdl` from the standard Java WS Core schema location:

```
<wsdl:import
namespace=http://diligentproject.org/namespaces/common/provider/DILIGENTProvider
location="../../common/provider/DILIGENTProvider/DiligentProvider.wsdl"/>
```

The portType must be extended with the following portTypes in the WSDL:

- GetResourceProperty (supplied by Java WS Core)
- GetMultipleResourceProperties (supplied by Java WS Core)
- QueryResourceProperties (supplied by Java WS Core)
- DiligentProvider

Example:

```
<portType name="YourServicePortType"
      wsdlpp:extends="wsrpw:GetResourceProperty
     wsrpw:GetMultipleResourceProperties
wsrpw:QueryResourceProperties
diligent:DiligentProvider">
[…]
```

Deployment file modifications

In the service deployment descriptor, the following Java WS Core-supplied operation providers for the previous portTypes have to be declared for each <service> element and for those services that aim at exposing their WS-ResourceProperties in the DIS:

- GetRPProvider,
- GetMRPProvider,
- QueryRPProvider
- org.diligentproject.common.provider.DILIGENTProvider

Example:

```
<parameter name="providers"
value="org.diligentproject.common.provider.DILIGENTProvider
  GetRPProvider GetMRPProvider QueryRPProvider"/>
```

Service code

The following packages have to be imported in order to access the DIS-IP functionality:

```
import org.diligentproject.common.provider.*;
import org.diligentproject.informationservice.disip.*;
```

Moreover, the `DILIGENTPropertySet` must to be used to create a collection of resource properties contained in a resource instead of the `SimpleResourcePropertySet` supplied with Java WS Core.

Finally, the service can use the operations described in Section 2.2.1.2 to publish its WS-Resource Properties.

Example:

```
import org.diligentproject.common.provider.*;
import org.diligentproject.informationservice.disip.impl.DISIP;


public class MyResource  implements PersistentResource {


      private ResourcePropertySet propSet;


      public myResource() {
            // create the DILIGENTProperty set (see the DILIGENTProvider
documentation)
            this.propSet = new
DILIGENTPropertySet(ResourceQNames.RESOURCE_PROPERTIES);
String uri = ServiceHost.getBaseURL().toString() + "...my service
name...";
            ResourceKey key = new SimpleResourceKey(new QName("...my
service namespace...","..myKey..."), this.getID());
            EndpointReferenceType epr =
AddressingUtils.createEndpointReference(uri, key);
            //reads registration.xml into String...
            String registrationFile = ....
            //register the RPs as aggregator source
            DISIP.registerPullMode(epr, registrationFile, this.propSet);
      }


      public void load(ResourceKey key) {
```

```
            String uri = ServiceHost.getBaseURL().toString() + "...my
service name...";
            EndpointReferenceType epr =
AddressingUtils.createEndpointReference(uri, key);
//reads registration.xml into String...
              String registrationFile = ....
            //register the RPs as aggregator source
              DISIP.registerPullMode(epr, registrationFile, this.propSet);
      }

}
```

ANT Build File modifications

If Ant is used as build tool, the following lines have to be added to the standard build file, in order to copy the DiligentProvider.wsdl inside the target build directory before compiling the component:

```
      <!-- Sets up the build directory structure -->
      <target name="init">
            .....
      <copy toDir="${schema.dest}">
            <fileset dir="${schema.src}" casesensitive="yes">
                  <include name="wsrf/**/*" />
                  <include name="ws/**/*" />
                  <!--add this line-->
                  <include name="diligentproject/common/**/*"/>
```

2.2.1.4 Implementation Details

The DIS-IP is implemented as a Java static library and the only entry point to its functionality is the `DISIP` class. The component maintains a state composed by the set of active Aggregator Source registrations in a particular moment on the DHN. The `Registration` class models the concept of "active registration".

To perform its specific tasks, the DISIP communicates with:

- the DIS-Broker, via the `DISBroker` class
- the local Aggregator Framework, via the `ServiceGroupRegistrationClient` class

Figure 4 shows the UML Class Diagram including the most meaningful classes of the DIS-IP implementation and their relationships.

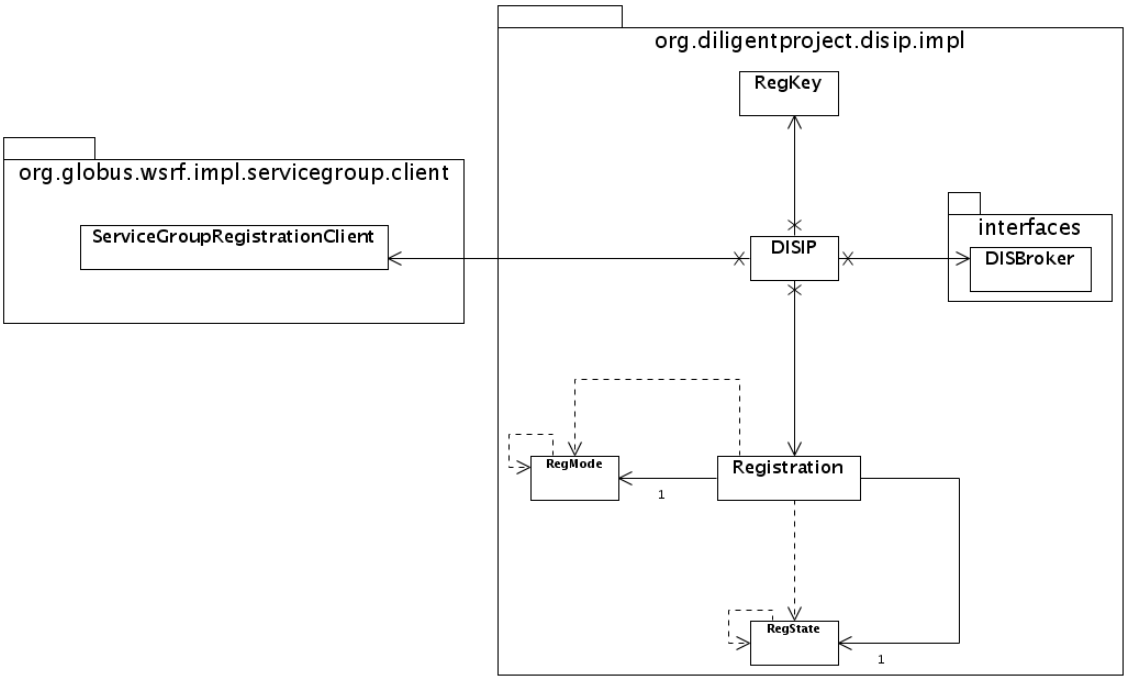

Figure 4. DIS-IP: Global class diagram

2.2.1.4.1 Classes

Each of the classes presented belongs to the package:

`org.diligentproject.informationservice.disip.impl`

DISIP

DISIP
+remove(arg0 : EndpointReferenceType) : void
+remove(arg0 : EndpointReferenceType, arg1 : java.lang.String) : void
+removeProfile(arg0 : EndpointReferenceType, arg1 : java.lang.String, arg2 : java.lang.String) : void
+registerPullMode(arg0 : EndpointReferenceType, arg1 : java.lang.String, arg2 : org.diligentproject.common.provider.DILIGENTPropertySet) : void
+registerPullMode(arg0 : EndpointReferenceType, arg1 : java.lang.String, arg2 : org.diligentproject.common.provider.DILIGENTPropertySet, arg3 : java.lang.String) : void
+registerPushMode(arg0 : EndpointReferenceType, arg1 : java.lang.String, arg2 : org.diligentproject.common.provider.DILIGENTPropertySet) : void
+registerPushMode(arg0 : EndpointReferenceType, arg1 : java.lang.String, arg2 : org.diligentproject.common.provider.DILIGENTPropertySet, arg3 : java.lang.String) : void
+registerToBroker(arg0 : EndpointReferenceType, arg1 : java.util.ArrayList) : void
+unregisterToBroker(arg0 : EndpointReferenceType, arg1 : java.util.ArrayList) : void
+registerPushModeProfile(arg0 : EndpointReferenceType, arg1 : java.lang.String, arg2 : java.lang.String) : void
+registerPullModeProfile(arg0 : EndpointReferenceType, arg1 : java.lang.String, arg2 : java.lang.String) : void

Figure 5. DIS-IP: DISIP class

This is the public interface of the component. All the public methods of this class are static; they have already been described in Section 2.2.1.2, since they are the operations that the DIS-IP exposes to its clients.

Registration

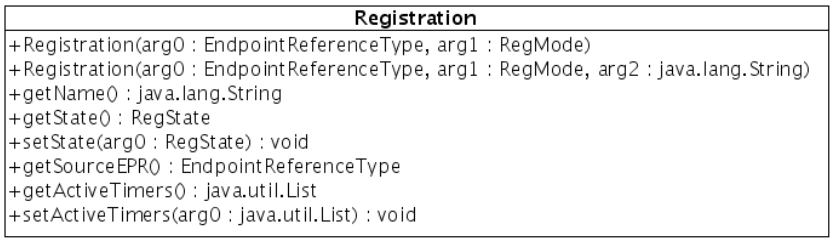

Figure 6. DIS-IP: Registration class

A Registration object represents an active registration of a group of resource properties to one or more remote DIS-IC instances. At Aggregator Framework level, for each of them, a Resource on the Aggregator Sink side is created with a WS-ResourceLifetime. In this case, the remote Aggregator Framework (hosted on the same DHN where the DIS-IC resides) acts as a WS-ServiceGroup, in the meaning specified in [19].

For each Registration, a timer list is created and maintained (`setActiveTimers()` and `getActiveTimers()` methods); when the timers expire, the registration is renewed on the remote Aggregator Framework.

RegMode, RegState, RegKey

These internal helper classes are used in the management of the state of the DIS-IP. They model the modality (push/pull) of a Registration, the state, and the unique key used to retrieve them from the state.

DISBroker

Figure 7. DIS-IP: DISBroker class

This interface creates an abstraction of the DIS-Broker stub classes in order to invoke the `registerTopic` and `unregisterTopic` operations exposed by the DIS-Broker (see Section 2.2.7.3).

ServiceGroupRegistrationClient

The Aggregator Framework supplies this class; it is a client library to manage registrations to WS-ServiceGroups.

Using this library, after a registration request, the DIS-IP creates a ServiceGroupEntry in the appropriate WS-ServiceGroup(s), i.e. where the DIS-IC(s) is (are) hosted. It will then periodically attempt WS-ResourceLifetime lifetime extension on the ServiceGroupEntry. If the DIS-IP detects that the ServiceGroupEntry is no longer available, it will create a new one.

2.2.1.4.2 Dependencies

The DIS-IP requires the following components to be present on the same DHN:

- DIS-Broker stubs classes
- DILIGENTProvider
- Node Access Library

- Aggregator Framework

In each DL or VO, it requires:

- one DIS-IC instance, at least
- one DIS-Broker instance, at least.

2.2.1.4.3 Configuration

At registration time, a DIS-IP client has to pass an XML string including the configuration parameters of the new Registration. For the Pull Mode, the general syntax of the configuration file is the following:

```
<!-- Example of Registration file to use in Pull Mode registrations-->

<ServiceGroupRegistrationParameters
   xmlns:sgc="http://mds.globus.org/servicegroup/client"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing"
   xmlns:agg="http://mds.globus.org/aggregator/types"
   xmlns="http://mds.globus.org/servicegroup/client">

   <!-- Specifies that the registration will be renewed every X
        seconds leave the value empty if you want an endless registration
(deprecated behavior)-->
   <RefreshIntervalSecs>X</RefreshIntervalSecs>

   <!-- <Content> specifies registration specific information -->
   <Content xsi:type="agg:AggregatorContent"
      xmlns:agg="http://mds.globus.org/aggregator/types">

      <agg:AggregatorConfig>
       <agg:GetMultipleResourcePropertiesPollType
      xmlns:service="http://diligentproject.org/namespaces/service/compone
nt/instance"
      xmlns:provider="http://diligentproject.org/namespaces/common/provide
r/DILIGENTProvider">
         <agg:PollIntervalMillis><!-- 60000 --> </agg:PollIntervalMillis>
                    <agg:ResourcePropertyNames> service:Prop1
         </agg:ResourcePropertyNames>
         <agg:ResourcePropertyNames> service:Prop2
         </agg:ResourcePropertyNames>

         <!-- DILIGENT Provider RPs-->
         <agg:ResourcePropertyNames> provider:RunningInstanceID
         </agg:ResourcePropertyNames>
         <agg:ResourcePropertyNames> provider:ServiceID
```

```
          </agg:ResourcePropertyNames>
          <agg:ResourcePropertyNames> provider:ServiceName
          </agg:ResourcePropertyNames>
          <agg:ResourcePropertyNames> provider:ServiceClass
          </agg:ResourcePropertyNames>
          <agg:ResourcePropertyNames> provider:VOName
          </agg:ResourcePropertyNames>
          <agg:ResourcePropertyNames> provider:DiligentType
          </agg:ResourcePropertyNames>
          <agg:ResourcePropertyNames> provider:DHNID
        </agg:ResourcePropertyNames>
        </agg:GetMultipleResourcePropertiesPollType>
       </agg:AggregatorConfig>
      <agg:AggregatorData/>
    </Content>
</ServiceGroupRegistrationParameters>
```

As shown, in addition to the service-specific resource properties, also the DILIGENT RPs (see Section 2.2.3) have to be explicitly registered.

The registration includes two important timings:

RefreshIntervalSecs - The refresh interval of the registration, in seconds. The DIS-IP will attempt to refresh the registration according to this interval, by default incrementing the termination time of the registration by 2 times this interval for every successful refresh. If at any point the termination time for the registration expires the registration will be subject to removal within a maximum of 5 minutes.

PollIntervalMillis - The poll refresh period in milliseconds. The DIS-IC comes to poll the registered resource properties with this polling interval.

2.2.1.5 Known Bugs and Limitations

The Push Mode will be supported in the Beta release.

2.2.2 DIS-IC

The DILIGENT Information Collector (DIS-IC) is the DILIGENT service in charge to collect WS-ResourceProperties (including the Resource Profiles from the DIS-Registry) registered via the DIS-IP and to make them available to be queried by the DIS-HLSClient.

It is implemented as an aggregator service, able to create Aggregator Sinks that query remote Aggregator Sources (in particular QueryAggregatorSources, as those created by the DIS-IP) to harvest resource properties. Then, the collected information is stored in an embedded instance of an eXist XML database[5]. This allows to persist the information as well as to build on top of eXist query APIs, the DIS-IC query interface, based on the XQuery query language[6].

[5] eXist native XML database: http://exist.sourceforge.net/
[6] XQuery 1.0: http://www.w3.org/TR/xquery/

From an architectural point of view, four WSRF Services (DISICRegistrationService, DISICRegistrationServiceEntry, DISICFactoryService and DISICService) which globally implement the Aggregator Sink functionalities (including those related to the storage, indexing, and management of resource information) compose the DIS-IC. The DISICService also exposes the public interface to query and/or delete the stored information.

XML database organization

The eXist database supports the XML:DB Initiative API[7] and therefore the information is organized in Collections as requested by this Initiative.

The following hierarchy of Collections is automatically created at the first start up by the DIS-IC:

```
db (root collection)
|
|-Properties
|
|-Profiles
       |
       |-RunningInstance
       |-DHN
       |-Service
       |-ExternalRunningInstance
       |-CS
       |-CSInstance
       |-Collection
       |-gLiteResource
```

Document structure

In a XML database, usually, there is no need to fix a particular schema for documents to store. However, in the DIS-IC case, a set of "metadata elements" are added to the resource properties documents harvested from the Aggregator Sources to better characterize and manage such information.

The following document structure has been adopted:

```
<Document>
      <ID>internal document identifier</ID>
      <Source>EPR of the RI that publishes this document</Source>
      <EntryKey>Entry key of the ServiceGroupEntry</EntryKey>
      <GroupKey>Group key of the ServiceGroupEntry</GroupKey>
      <TerminationTime>expiration date from the epoch
      </TerminationTime>
      <TerminationTimeHuman>human readable version
      </TerminationTimeHuman>
      <LastUpdateMs>last update time from the epoch</LastUpdateMs>
      <LastUpdateHuman>human readable version</LastUpdateHuman>
```

[7] XML:DB Initiative: API - http://xmldb-org.sourceforge.net/xapi/index.html

```
      <Data>
            (the list of ws-resourceproperties
                  or
            the profile)
             +
             DILIGENT PROVIDER properties
      </Data>
</Document>
```

XQuery

The above information, i.e. the organization of Collections and the structure of the documents stored, are the basic knowledge needed to create queries in the XQuery language able to retrieve information from the DIS-IC. The most common queries are provided by the DIS-HLSClient – thus DILIGENT services can query the DIS-IC using the methods exposed by this library (see Section 2.2.5.1) without dealing with the XQuery language.

However, these methods do not cover all the needs. Therefore, it is always possible to directly send an XQuery query to the eXist database instance embedded in the DIS-IC using the `queryDISIC()` operation of the DIS-HLSClient (see Section 2.2.5).

2.2.2.1 Profile

The DIS-IC service profile is reported in Appendix A.1. It includes the following packages:

- DIS-IC (WSRFService)
- DIS-IP (Library)
- DIS-HLSClient (Library)
- DILIGENTProvider (Library)
- Aggregator Framework (software)

2.2.2.2 Managed Resources

Each time a ServiceGroupEntry is created after a DIS-IP registration, the underlying Aggregator Framework invokes the DISICFactoryService in order to create a new DISICResource resource.

The DISICResource extends the AggregatorServiceGroupResource and when the state of the linked ServiceGroupEntry changes, the `deliver ()` method is invoked and the new state is passed as parameter. The Entry is analyzed and the values of the new resource properties included in the state are extracted and stored in the appropriate Collection of the database instance.

2.2.2.3 Operations

executeXQuery

public String executeXQuery(String xquery) throws BaseFaultType

This operation executes the given XQuery on the embedded database instance of eXist.

The result set of the query is returned in the following format:

```
<Resultset>
```

```
      <Record> .... </Record>
      <Record> .... </Record>
</Resultset>
```

where the "...." are the content identified by the expression specified in the `return` statement of the given XQuery. A query takes into account the Collections hierarchy and the Document structure explained in the previous sections.

Examples:

- the following query retrieves all the Service profiles whose class is 'CSD':

```
for $doc in
collection("/db/Profiles/Service")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource/Profile where $doc/Class/text() eq
'CSD' return $doc
```

- the following query retrieves all the resource properties documents published by the all the WS-Resource of the service

http://dili02.osl.fast.no:8080/wsrf/services/org/diligent/BatchUpdaterService:

```
for $r in (collection("/db/Properties"))//Document/Source where $r/text()
eq
'http://dili02.osl.fast.no:8080/wsrf/services/org/diligent/BatchUpdaterSer
vice' return $r/ancestor::Document/Data
```

deleteProfile

public boolean deleteProfile(DeleteProfileParams params) throws BaseFaultType

This operation removes a Profile document from the XML database given its DILIGENT Resource ID and the resource type (DHN, RunningInstance, etc.).

Example:

```
DISICServiceLocator locator = new DISICServiceLocator();
URL dis_ic_url = new URL(uri);
DISICPortType dis_ic = locator.getDISICServicePortType(dis_ic_url);
DeleteProfileParams prof = new DeleteProfileParams();
prof.setDILIGENTResourceID(“DHNID”);
prof.setProfileType(“DHN”);
dis_ic.deleteProfile(prof);
```

The Profile removal may be performed only by the DIS-Registry. In the Beta release the access to this operation will be ruled by Authorization mechanisms.

deleteResource

public boolean deleteResource(String id) throws BaseFaultType

This operation removes a resource properties document from the XML database given its ID. The ID is the Key of the ServiceGroupEntry connected to the Aggregator Source that published the document.

deleteAllRPs

public void deleteAllRPs() throws BaseFaultType

This operation deletes the Collection “/db/Properties” from the XML database. In the Beta release, the access to this operation will be ruled by Authorization mechanisms.

dispose

public void dispose() throws BaseFaultType

This operation disposes the DISIC service by shutting down the connections and stopping the threads in a safe mode. In the Beta release, the access to this operation will be ruled by Authorization mechanisms.

initialize

public void initialize() throws BaseFaultType

This operation initializes the DIS-IC by opening the appropriate pool of connections to the XML database and starting the threads for administrative and periodic tasks. In the Beta release, the access to this operation will be ruled by Authorization mechanisms.

2.2.2.4 How to use the component

The DIS-IC should not be directly used by other services. The publishing phase is managed by the DIS-IP, while the query operations are managed by the DIS-HLSClient.

2.2.2.5 Implementation Details

The DIS-IC is composed by a set of WSRF services running in the Java WS Core container. The framework specifies what is needed to implement an Aggregator Sink connected with the local Aggregator Framework drives the DIS-IC software architecture.

Figure 8 reports the global class diagram of the DIS-IC component:

- the org.diligentproject.informationservice.disic.impl package groups the classes that implement the Aggregator Sink functionality
- the storage package groups the classes that interact with the XML database instance
- the core package includes the classes that allow to initialize and shutdown the DIS-IC in safe mode and maintain the global service state

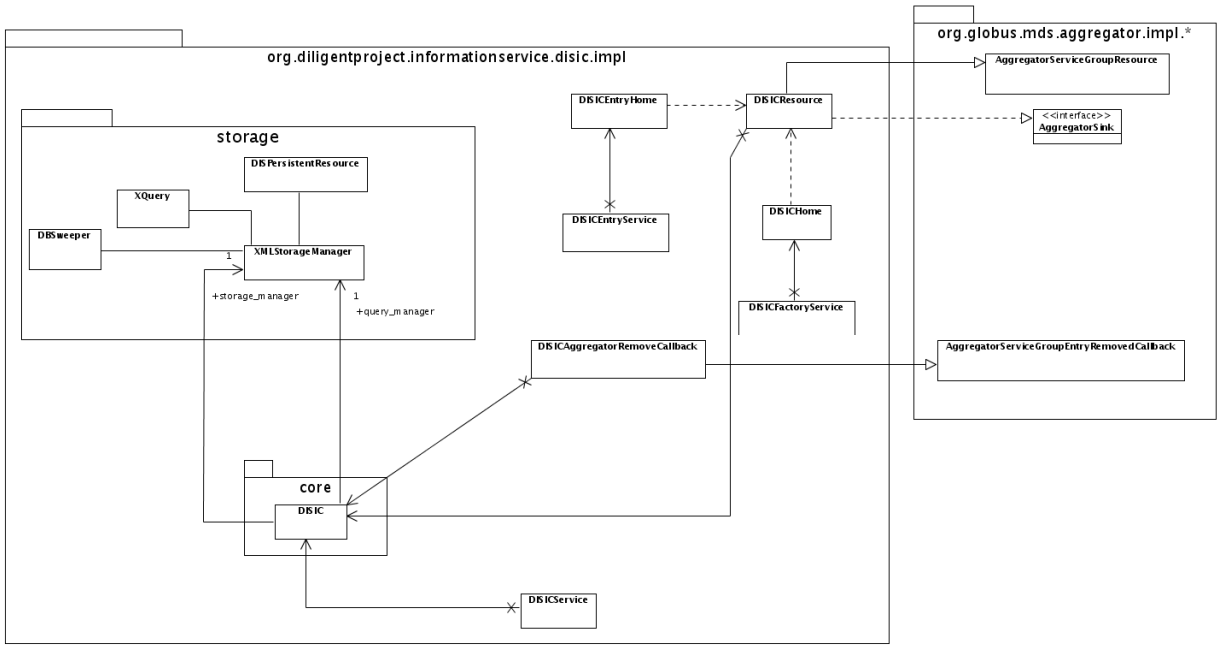

Figure 8. DIS-IC: Global class diagram

2.2.2.5.1 Classes

DISICResource

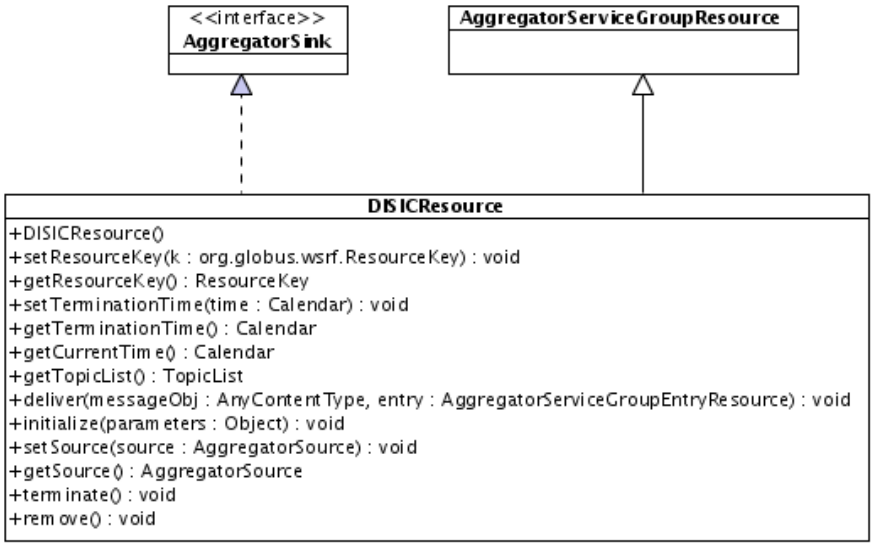

Figure 9. DIS-IC: DISICResource class

This class implements an aggregating in-memory service group resource. For every registered Aggregator Source, the Aggregator Framework creates one connected DISICResource object (acting as an Aggregator Sink for that source). The interface implemented by such objects is invocated each time new data are harvested by the connected service group resource.

DISICService

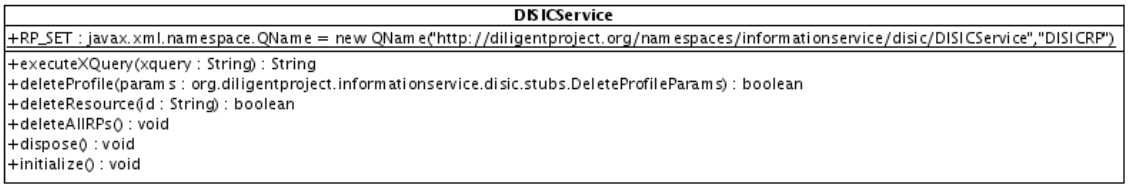

Figure 10. DIS-IC: DISICService class

This is the service class of the DISICService. It exposes the public interface of the DIS-IC. All the methods of this class are described in Section 2.2.2.3.

DISICAggregatorRemoveCallback

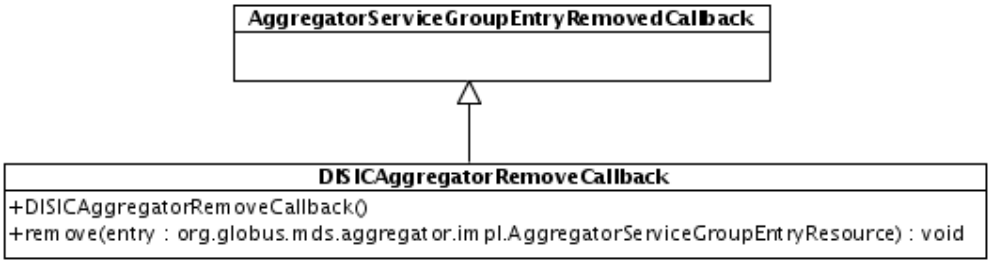

Figure 11. DIS-IC: DISICAggregatorRemoveCallback class

Whenever a AggregatorServiceGroupEntryResource is removed, the corresponding `remove()` method of this class is invoked passing the instance of the resource that has to be removed as parameter. Then, the related resource properties document in the XML database is removed since its AggregatorSource is no longer available.

XMLStorageManager

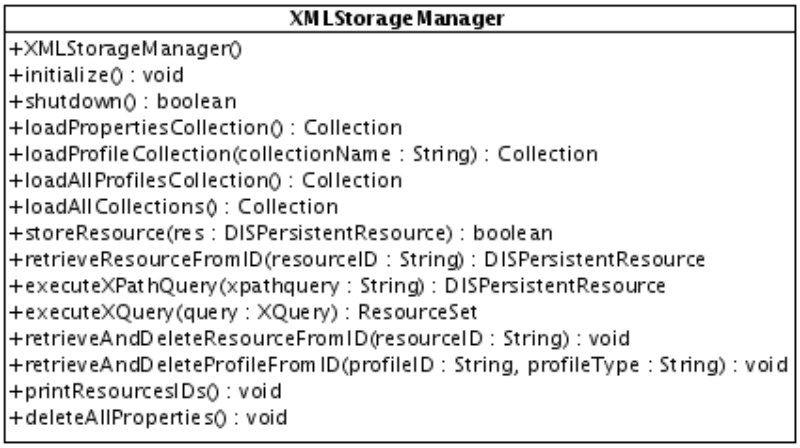

Figure 12. DIS-IC: XMLStorageManager class

This class creates an abstraction layer on the underlying eXist instance. All the operations performed on the XML database are accessible through the XMLStorageManager.

DISPersistentResource

Figure 13. DIS-IC: DISPersistentResource class

This class models the resources to store in the XML database. Starting from the information provided with the `set*` methods, it creates a document compliant with the structure presented in Section 2.2.2.

DISIC

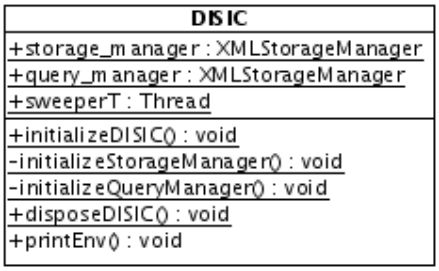

Figure 14. DIS-IC: DISIC class

This class performs the basic initialization of the completely DIS-IC set of services. The `initializeDISIC()` method is invoked at DISICFactoryService service start up; it opens the connections with the embedded eXist instance and starts the Sweeper thread (it periodically performs a cleanup of outdated resource properties documents). The `disposeDISIC()` is invoked at DISICFactoryService service shutdown.

2.2.2.5.2 Stubs

Figure 15 depicts the main classes of the DIS-IC stubs involved in the interaction with DIS-IC from the client point of view. All of them are part of the package:

`org.diligentproject.informationservice.disic.stubs.*`

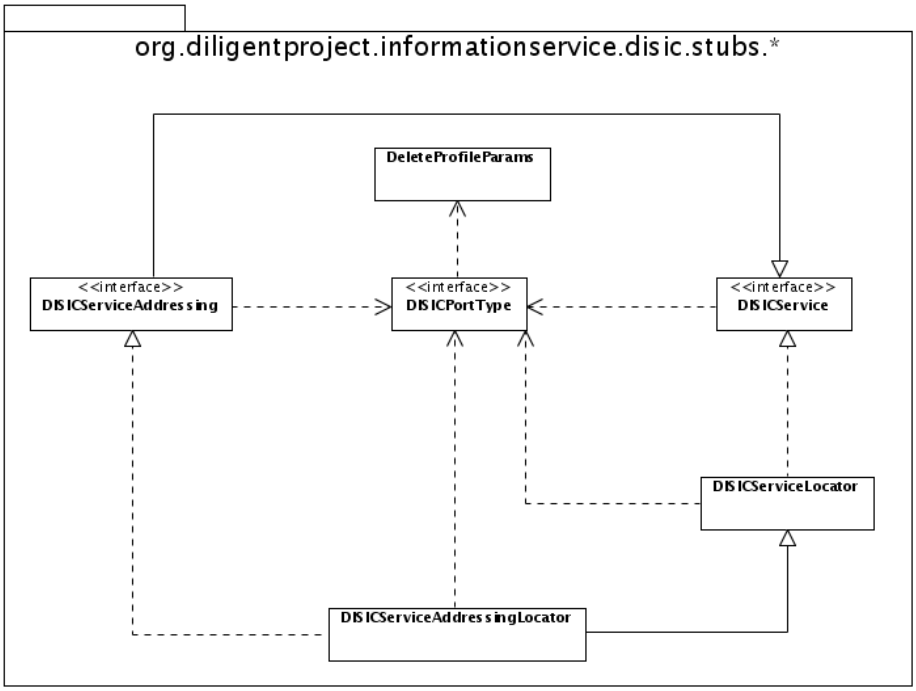

Figure 15. DIS-IC stubs class diagram

These classes should not be directly used by the services that aim at interacting with the DIS-IC, since an API to feed data is provided by the DIS-IP and the query interface is mediated by the DIS-HLSClient.

DISICPortType

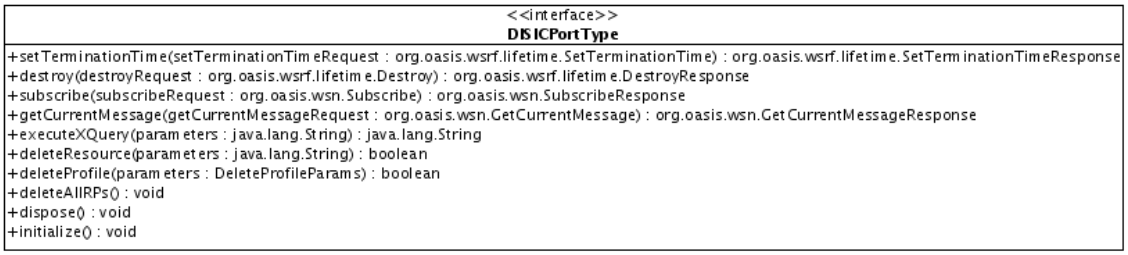

Figure 16. DIS-IC: DISICPortType class

This is the portType class that allows to invoke the DIS-IC operations described in Section 2.2.2.3. The way to access the operations is the same of any other portType class.

Example:

```
DISICServiceLocator locator = new DISICServiceLocator();
DISICPortType dis_ic = locator.getDISICServicePortType(dis_ic_url);
dis_ic.executeQuery(<xquery>);
```

DeleteProfileParams

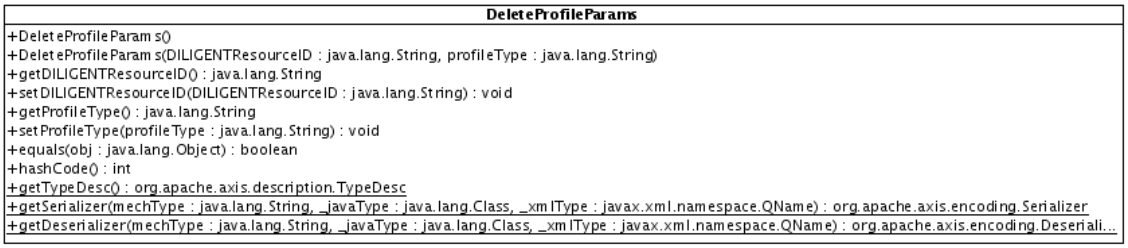

Figure 17. DIS-IC: DeleteProfileParams

Instances of this class are used to pass the parameters to the `deleteProfile()` method of the `DISICPortType` class. The two parameters are the resource ID and the type (DHN, Service, etc.) of the profile to delete. The `setProfileType()` and `setDILIGENTResourceID()` methods have to be used to set them.

2.2.2.5.3 Dependencies

The DIS-IC components depend on the Aggregator Framework and on the eXist XML database. The former is automatically deployed together with the other packages forming the DIS-IC, while the latter has to be installed manually and is a DHN requirement for the Alpha release. The Beta release will also automatically deploy the eXist database on the DHN as a preliminary operation before the deployment of any DIS-IC component.

2.2.2.6 Known Bugs and Limitations

The distributed deployed model of DIS-ICs and replica synchronization will be supported in the Beta release.

2.2.3 DILIGENTProvider

An *operation provider* is a mechanism to plug new operations and resource properties into services; its main purpose it to avoid having to duplicate code in each service. A service just needs to extend its portType and declare the provider in the service deployment descriptor.

The Alpha release of the DILIGENTProvider adds the following resource properties (named DILIGENT RPs):

- *RunningInstanceID*: the DILIGENT Resource ID of the RunningInstance
- *ServiceID*: the DILIGENT Resource ID of the service from which the RunningInstance was generated
- *ServiceName*: the name of the service (e.g. LookupService or DIS-Registry)
- *ServiceClass*: the class of the service (e.g. Index, InformationSystem)
- *DHNID*: the DILIGENT Resource ID of the DHN that is currently hosting the RunningInstance
- *VOName*: the VO (s) the RunningInstance belongs to
- *DiligentType*: what the resource properties document contains (Profile or Properties)

Their goal is to add information to the other RPs published by the service. In this way, it is possible to send queries like "give me all the properties published by the RunningInstance of Service X of Class Y" to the DIS-IC. When a service uses the DILIGENTProvider, these properties become part of the resource properties documents published via the DIS-IP. The last thing to take into account is that the DILIGENT RPs are declared in the namespace:

```
http://diligentproject.org/namespaces/common/provider/DILIGENTProvider
```

2.2.3.1 Profile

The DILIGENTProvider is distributed as part of the DIS-IC component. See Section 2.2.2.1 for the profile that includes the DILIGENTProvider description.

2.2.3.2 Operations

The ALPHA version of the DILIGENTProvider only adds resource properties.

2.2.3.3 How to use the component

The provider is mainly used in conjunction with the DIS-IP to add the DILIGENT RPs at registration time.

Firstly, the DILIGENTProvider has to be included in the WSDL (as a portType extension) and declared in the WSDD file as described in Section 2.2.1.3.

Then, a DILIGENTPropertySet object must be created to collect the resource properties of a resource instead of the usual PropertySets created with the Java WS Core-supplied classes (see the example reported in Section 2.2.1.3).

Finally, the DILIGENT RPs have to be added to the list of service specific resource properties declared in the registration file provided to the DIS-IP (see Section 2.2.1), as reported in the template file in Section 2.2.1.4.3.

2.2.3.4 Implementation details

The DILIGENTProvider is composed by three classes, as depicted in Figure 18, and by a WSDL file to import in the service's WSDL.

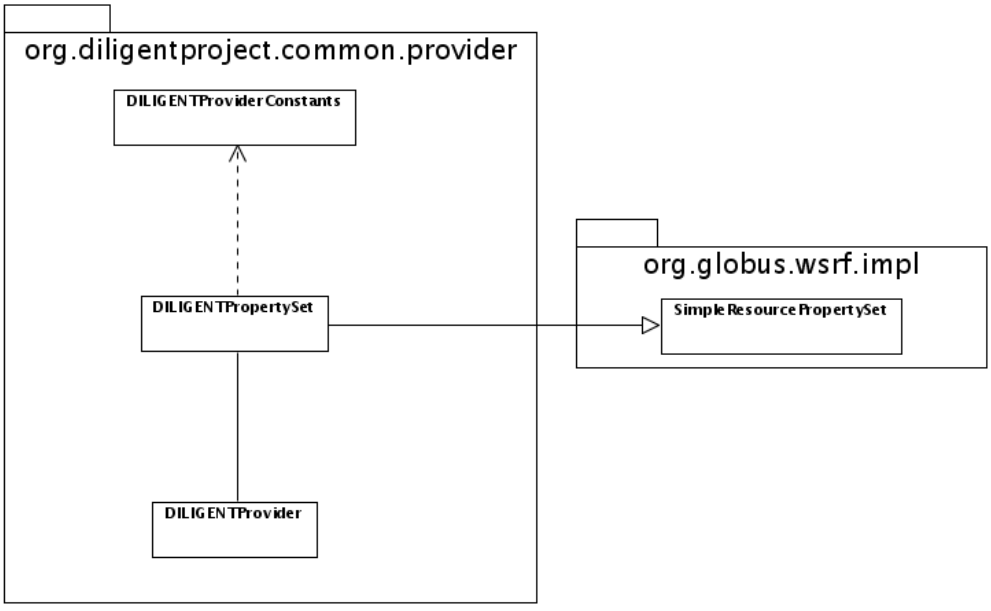

Figure 18. DILIGENTProvider: Global class diagram

2.2.3.4.1 Classes

DILIGENTPropertySet

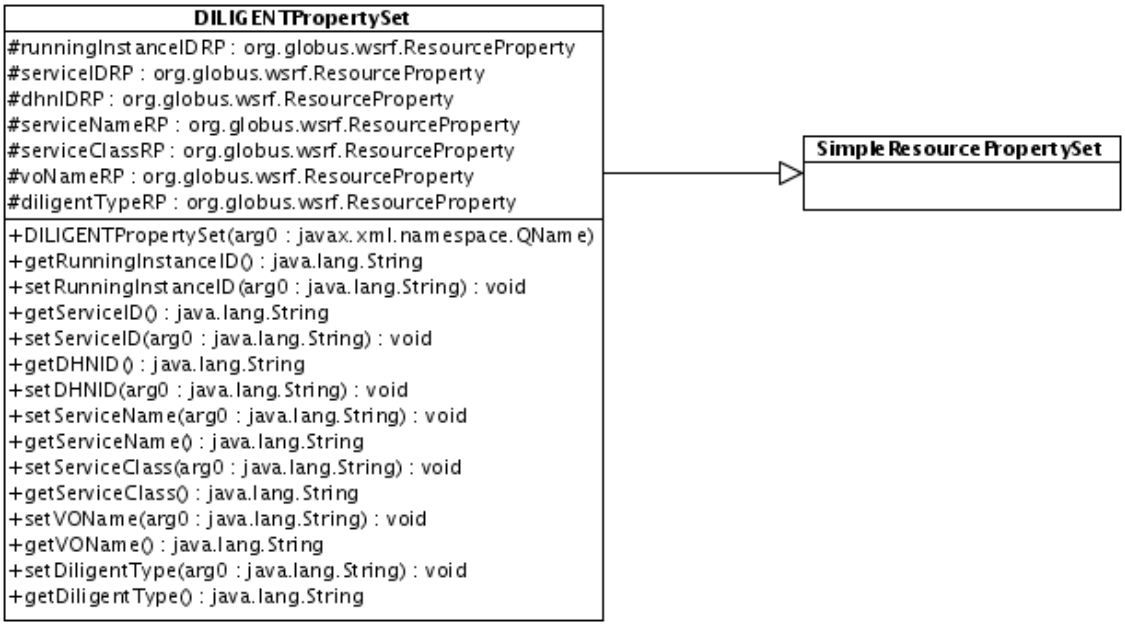

Figure 19. DILIGENTProvider: DiligentPropertySet class

This custom PropertySet class extends the SimpleResourcePropertySet class supplied by Java WS Core. The class provides the getter and setter methods for the DILIGENT RPs.

DILIGENTProviderConstants

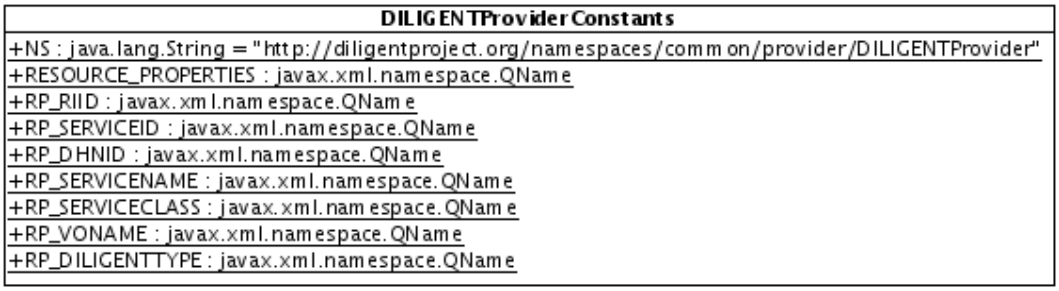

Figure 20. DILIGENTProvider: DILIGENTProviderConstants class

The class groups the definitions of the WS-ResourceProperties added by the DILIGENTProvider.

DILIGENTProvider

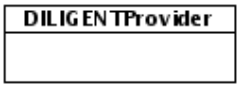

Figure 21. DILIGENTProvider: DILIGENTProvider class

This class has no special methods. It just instantiates the DILIGENTProvider.

2.2.3.4.2 Dependencies

None

2.2.4 DIS-R-GMAClient

This service represents the DIS-R-GMAClient and the gLiteInfoProvider classes reported in the Logical view of the component introduced in D1.2.2. It is in charge to harvest resources information from the gLite R-GMA Server it has been configured to interact with. This information is related to grid resources (CEs, SEs, I/O Servers, etc.) belonging to each gLite based grid infrastructure the DILIGENT system is interacting with. Then, the gathered information is manipulated in order to be compliant with the schema adopted in DILIGENT and published as gLite Resource using the DIS-Registry (static information based on Glue schema specification v1.2) or as WS-Resource Properties on the local DIS-IP where the DIS-R-GMAClient is deployed via the Aggregator Framework support (see DIS-R-GMA-Client class diagram in Figure 22).

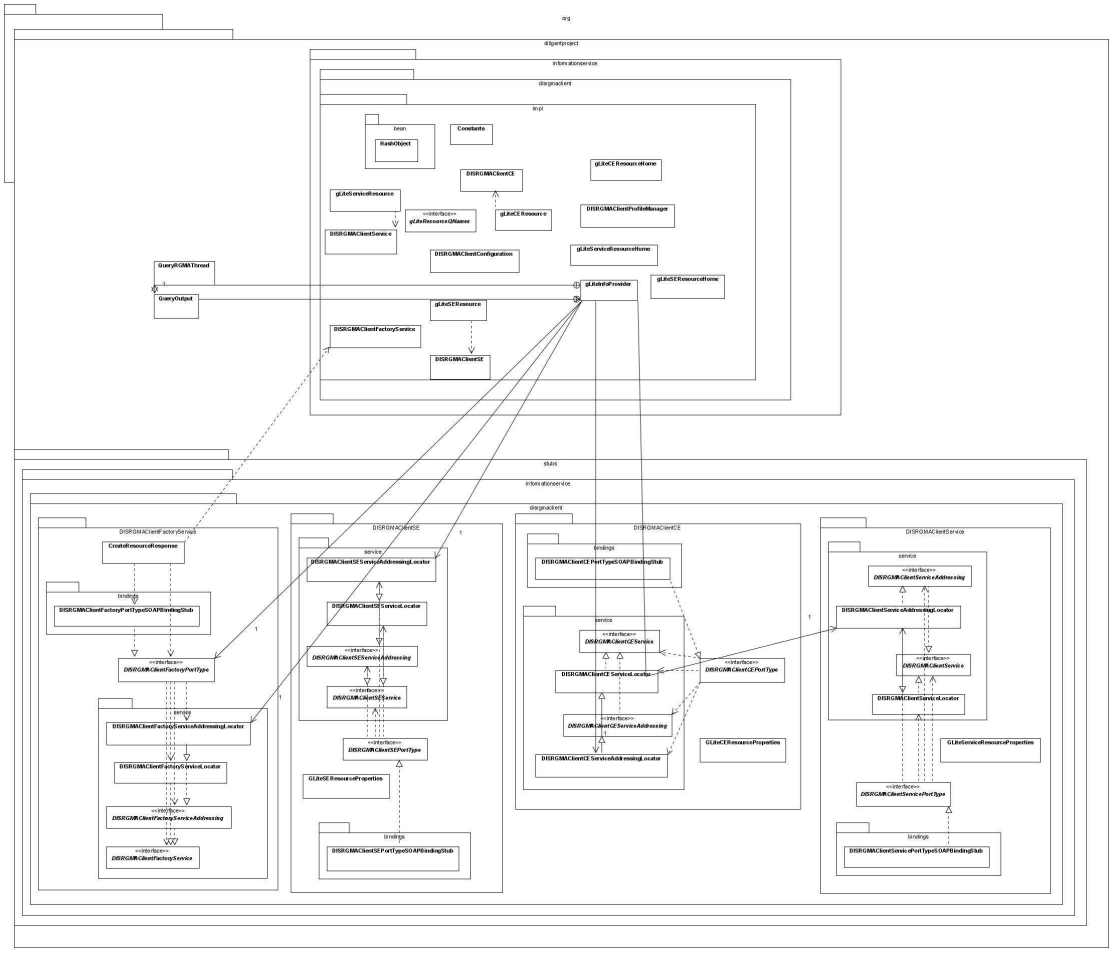

Figure 22. DIS-R-GMAClient: global class diagram

2.2.4.1 Profile

The DIS-R-GMAClient service profile is reported in Appendix A.2. In addition, the DIS-R-GMAClient registers the gLite Resource profiles filled with the information gathered from the R-GMA Server.

We distinguished four types of gLite Resource (and four types of gLiteResource Profiles):

- Service;
- Computing Element:

- Storage Element;
- Site;

Such profiles are compliant with the GLUE Schema 1.2 specification [2] (reported in 3.1). The profiles reported respectively in Appendix A.5, A.6, A.7 A.8.

2.2.4.2 Managed Resources

The service adopts the factory pattern [20]. The DISRGMAClientFactoryService is responsible for the creation of three types of WS-Resources:

- DISRGMAClientService, that exposes WS-Resource Properties containing status information for gLite Services;
- DISRGMAClientSE, that exposes WS-Resource Properties containing status information for gLite Storage Elements;
- DISRGMAClientCE, that exposes WS-Resource Properties containing status information for gLite Computing Elements;

In particular, DISRGMAClientService WS-Resources contain these WS-Resource Properties:

- UniqueID: the gLite UniqueID of the Service;
- Status: a string representing the Service status;
- Status Info: other Status information;

DISRGMAClientSE WS-Resources contains these WS-Resource Properties:

- UniqueID;
- UsedSpace ;
- AvailableSpace ;
- SizeFree;

DISRGMAClientCE WS-Resources contains these WS-Resource Properties :

- UniqueID;
- RunningJobs;
- WaitingJobs;
- TotalJobs;
- EstimatedResponseTime;
- WorstResponseTime;
- FreeJobsSlot;
- MaxWallClockTime;
- MaxCPUTime;
- MaxTotalJobs;
- MaxRunningJobs;

- AssignedJobSlots;
- Priority;

2.2.4.3 Operations

The only public operation implemented by the DISRGMAClientFactoryService is:

createResource

```
CreateResourceResponse createResource(String type)
```

It allows creating WS-Resources of type DISRGMAClientService/CE/SE.

Parameters:

- String type: the type of the resource to create (CE, SE or Service)

Return: a CreateResourceResponse object that contains the EPR of the new WS-Resource created.

The standard GT4 providers provide all the operations offered by the DISRGMAClientService, DISRGMAClientSE and DISRGMAClientCE services:

- SetResourceProperty;
- GetResourceProperty.

2.2.4.4 Implementation Details

2.2.4.4.1 Classes

Every class includes an instance of `org.apache.commons.logging.Log` used by the log4j logging mechanism [4]. All the classes presented in the follow belong to the package `org.diligentproject.informationservice.disrgmaclient.impl`

DISRGMAClientFactoryService

This class (Figure 23) implements the creation of WS-Resources of type DISRGMAClientService/SE/CE. It also instantiates the gLiteInfoProvider thread that is responsible for gathering information from the gLite R-GMA Server.

Figure 23. DIS-R-GMAClient : DISRGMAClientFactoryService class diagram

DISRGMAClientService

This class models the concept of a gLite service.

Figure 24. DIS-R-GMAClient: DISRGMAClientService class diagram

DISRGMAClientSE

This class models the concept of a gLite StorageElement.

Figure 25. DIS-R-GMAClient: DISRGMAClientSE class diagram

DISRGMAClientCE

This class (see Figure 24) models the concept of a gLite Computing Element.

.

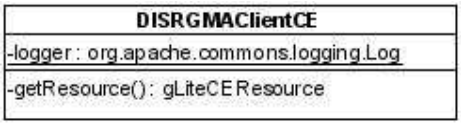

Figure 26. DIS-R-GMAClient: DISRGMAClientCE class diagram

gLiteInfoProvider

This class (Figure 27) implements a thread acting as a "client" of the service. The tasks it performs are:

- contacting gLite R-GMA server using R-GMA client API and retrieving tuples belonging to GLUEService table;
- creating for every gLite service, SE and CE the corresponding DISRGMAClientService/CE/SE WS-Resource;
- invoking the appropriate DIS-Registry operations to register gLite Resource profiles.
- maintaining persistent information about gLiteResources locally (EPR of the resource). Thus, it can periodically update information about the Resources or add some Resources that were not on the R-GMA Server before. This is also important when some gLite Services/Ses/CEs disappear from the R-GMA server: the corresponding WS-Resource is automatically destroyed.

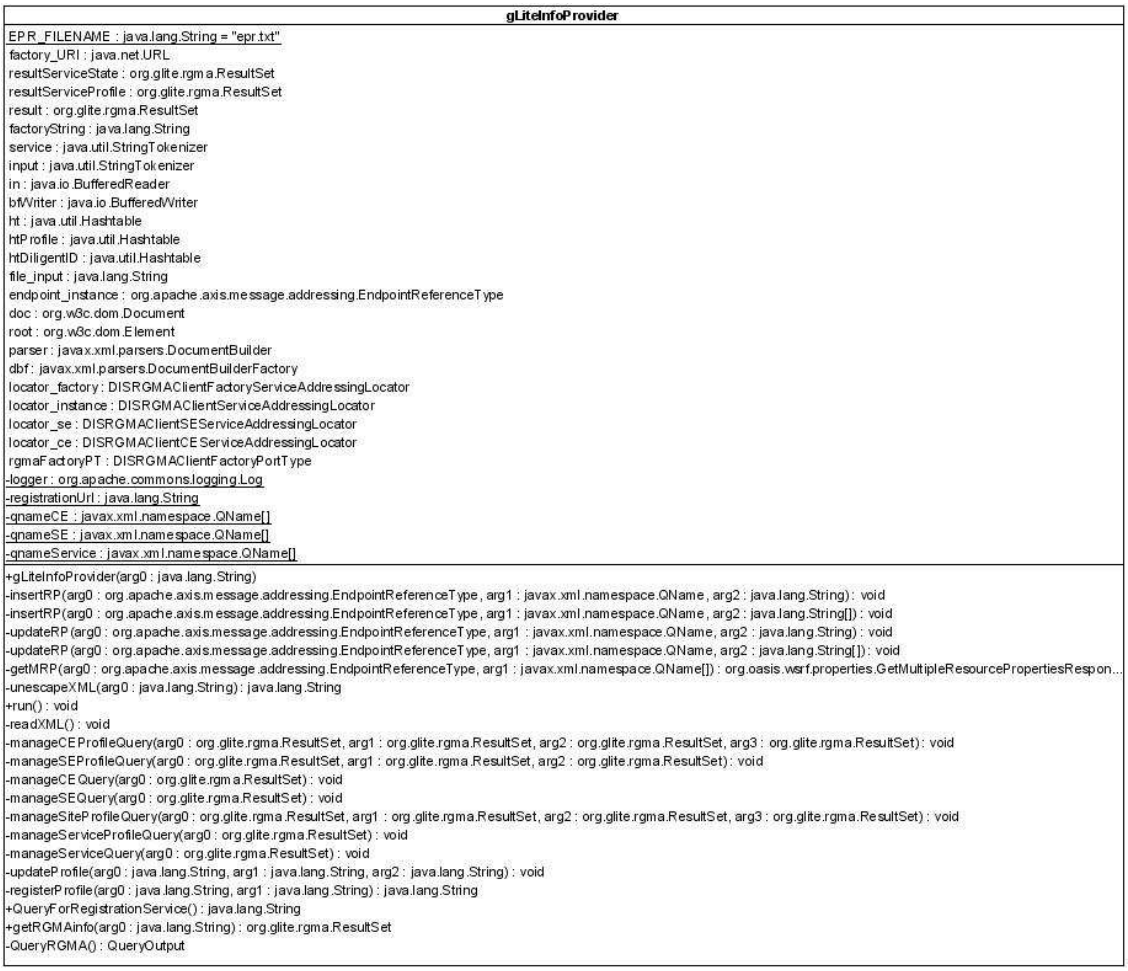

Figure 27. DIS-R-GMAClient: gLiteInfoProvider class diagram

gLiteServiceResource

This class (Figure 28) represents the statefull part of the DISRGMAClientService. It defines the following WS-Resource Properties:

- uniqueIDRP: unique ID of the gLite Service;
- statusRP: status info for gLite Service;
- statusInfoRP: other status information for gLite Services;

All these WS-Resource Properties are also exposed as Topics and registered to the DIS-Broker exploiting the mechanism of subscription brokering. Such resources are registered in the DIS-IC using the local DIS-IP library.

Figure 28. DIS-R-GMAClient: gLiteResource class diagram

gLiteServiceResourceHome

This class is contacted by the DISRGMAClientFactoryService to create WS-Resources of type gLiteServiceResource (see Figure 29). The other resourceHome class description (gLiteSEResource and gLiteCEResource) are omitted because they have the same behaviour as this class.

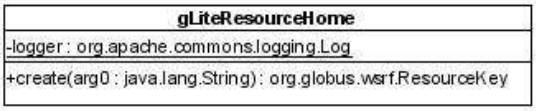

Figure 29. DIS-R-GMAClient: gLiteResourceHome class diagram

gLiteCEResource

This class (Figure 30) is the stateful part of the DISRGMAClientCE. It defines these WS-Resource Properties:

- uniqueIDRP;
- runningJobsRP;
- waitingJobsRP;
- totalJobsRP;
- estimatedResponseTimeRP;
- worstResponseTimeRP;
- freeJobsSlotRP;

- maxWallClockTimeRP;
- maxCPUTimeRP;
- maxTotalJobsRP;
- maxRunningJobsRP;
- assignedJobSlotsRP;
- priorityRP;

All these WS-ResourceProperties are also exposed as Topics and registered to the DIS-Broker exploiting the mechanism of subscription brokering. The resources are registered in the DIS-IC using the local DIS-IP library.

Figure 30. DIS-R-GMAClient: gLiteCEResource class diagram

gLiteSEResource

This class (Figure 31) represents the stateful part of the DISRGMAClientSE. It defines the following WS-Resource Properties:

- UniqueID;
- UsedSpace ;
- AvailableSpace ;
- SizeFree ;

All these WS-ResourceProperties are also exposed as Topics and registered to the DIS-Broker exploiting the mechanism of subscription brokering.

This class is also responsible for registering the WS-ResourceProperties to the DIS-IC using the local DIS-IP library.

Figure 31. DIS-R-GMAClient: gLite SE Resource Class Diagram

gLiteResourceQNames

This class (see Figure 32) defines the WS-Resource namespace and WS-ResourceProperties QNames.

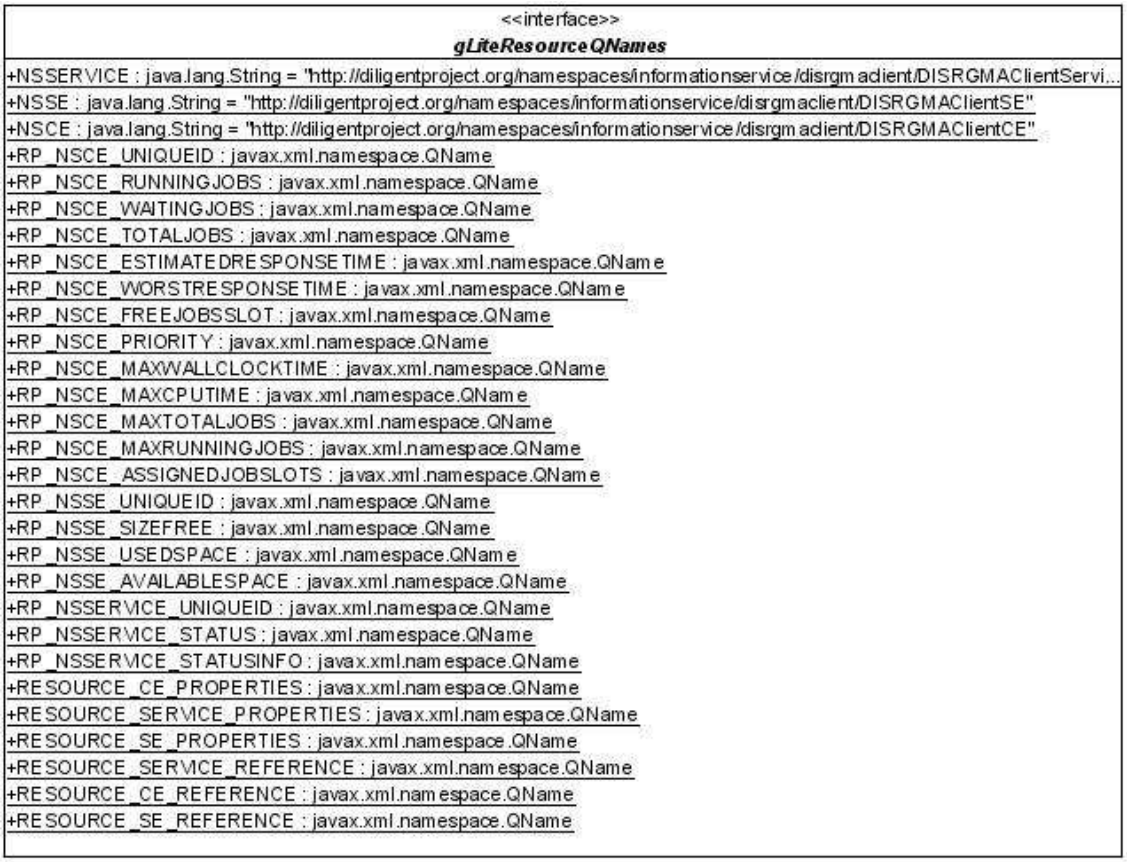

Figure 32. DIS-R-GMAClient: gLiteResourceQNames class diagram

bean.HashObject

This class is used as key for WS-Resource creation and destruction. The gLiteInfoProvider class maintains a HashTable containing the EPRs of the created resources. The *flag* field associated to the EPR is used for checking consistencies between the resources registered on the DIS-IP and the ones gathered from the R-GMA server.

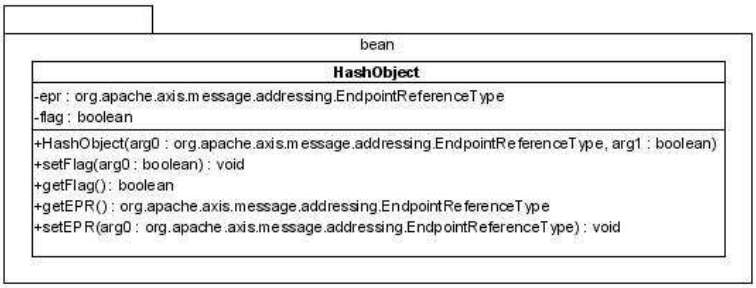

Figure 33. DIS-R-GMAClient: HashObject class diagram

DISRGMAClientProfileManager

This class (see Figure 34) contains methods for gLite Resource profile marshalling/unmarshalling from/to file. The gLiteInfoProvider class uses it in order to create/compare gLite Resource profiles.

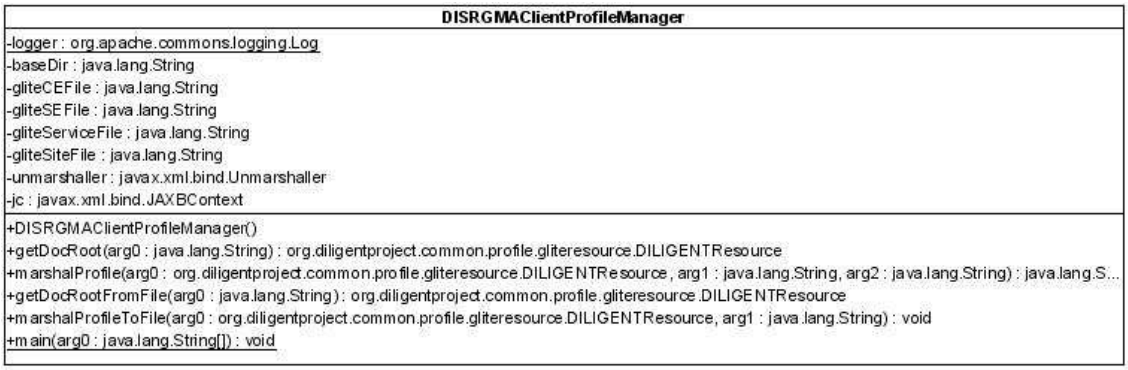

Figure 34. DIS-R-GMAClient: DISRGMAClientProfileManger class diagram

DISRGMAClientConfiguration

This class (see Figure 34) manages resource properties registrations to DIS-IC (via local calls to the DIS-IP library).

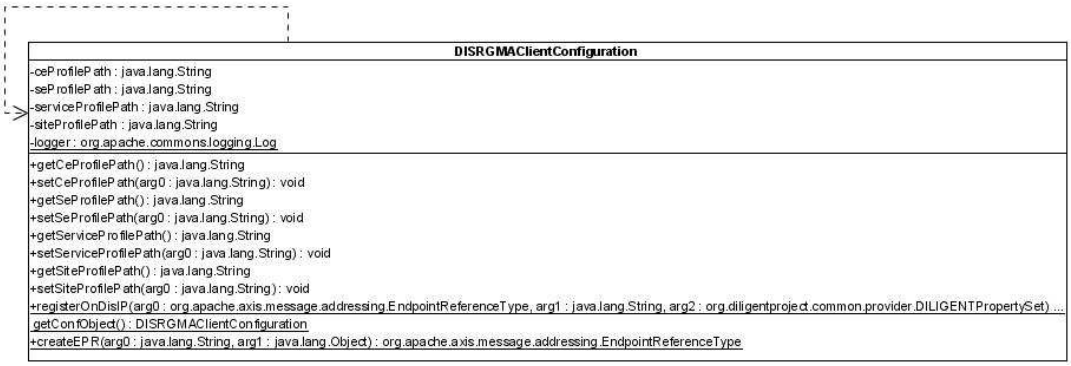

Figure 35. DIS-R-GMAClient: DISRGMAClientConfiguration class diagram

2.2.4.4.2 Stubs

This section contains class diagrams for the stub classes generated at compile time from the DIS-R-GMAClient WSDL files (see Figure 36, Figure 37, Figure 38, and Figure 39).

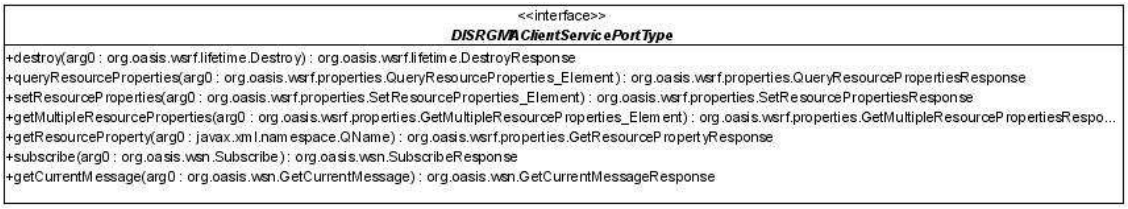

Figure 36. DIS-R-GMAClient: DISRGMAClientServicePortType stub class diagram

Figure 37. DIS-R-GMAClient: DISRGMAClientSEPortType stub Class Diagram

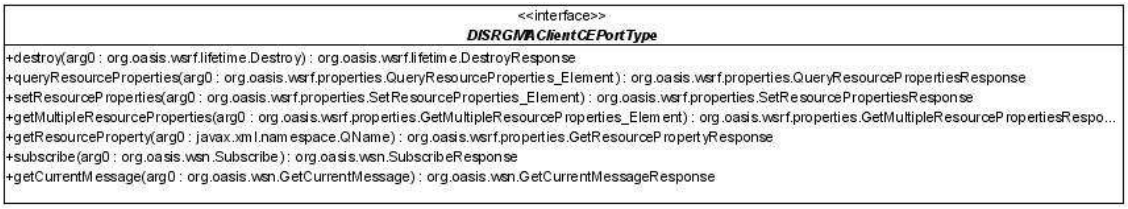

Figure 38. DIS-R-GMAClient: DISRGMAClientCEPortType stub Class diagram

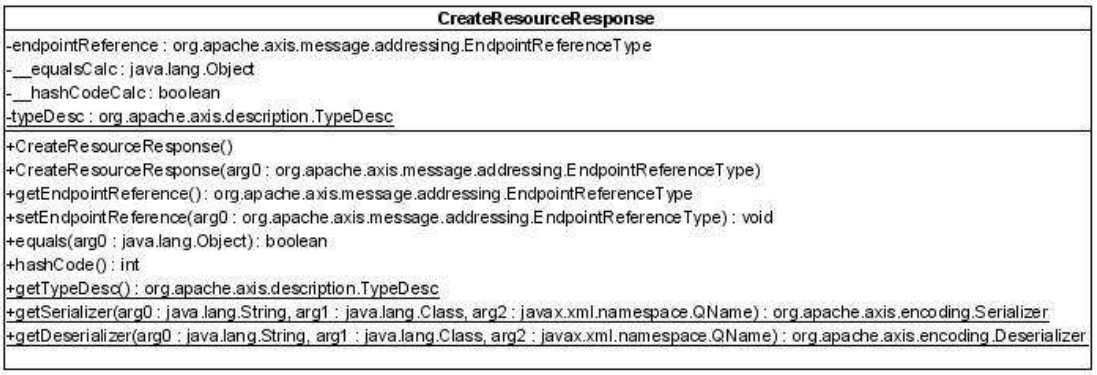

Figure 39. DIS-R-GMAClient: CreateResourceResponse stub class diagram

2.2.4.4.3 Dependencies

DIS-R-GMAClient has to contact a gLite R-GMA server in order to gather information related to gLite Services. This means that the service needs a proper identity in order to be authenticated by the R-GMA server. Following the design of the architectural specification, this service acquires this identity by interacting with the CredentialMapperService[8]. Because this service is not yet available, the current release uses the "classic" gLite authentication method that implies the usage of proxy certificates. The drawback of this solution is represented by the need to equip the DHN node with the gLite UI component installation (that also includes the R-GMA client API).

Only the following subset of gLite UI installation components will be necessary with the use of CredentialMapperService:

- glite-rgma-api-java.jar
- glite-rgma-stubs-servlet-java
- glite-security-delegation-java.jar
- glite-security-util-java.jar
- glite-security-trustmanager.jar

The service dependencies towards other DILIGENT component are:

- DIS-Registry stubs;
- NAL library;
- DISIP library;
- DILIGENTProvider provider;

[8] A component provided by the DVOS Service (Section 4).

- DIS-Broker stubs;

2.2.5 DIS-HLSClient

The DIS-HLSClient is a library that is available on each DHN. It is used by any client (service, portlet, application) hosted on the node to have access to the information maintained by the DILIGENT Information Service while hiding details about communication and information partitioning among instances of the DIS-ICs (see Section 2.2.2).

Using the DIS-HLSClient it is possible to query resource properties documents, discover DILIGENT resources and gather their profiles information by performing queries (XPath and XQuery) based on specifying resource characteristics.

To query information made available by the DIS-IC component, a service has to send appropriate XPath or XQuery queries. Even if it is possible to directly send these queries, the DIS-HLSClient exposes a number of higher-level methods that allow clients to avoid having to deal with the details of these query languages.

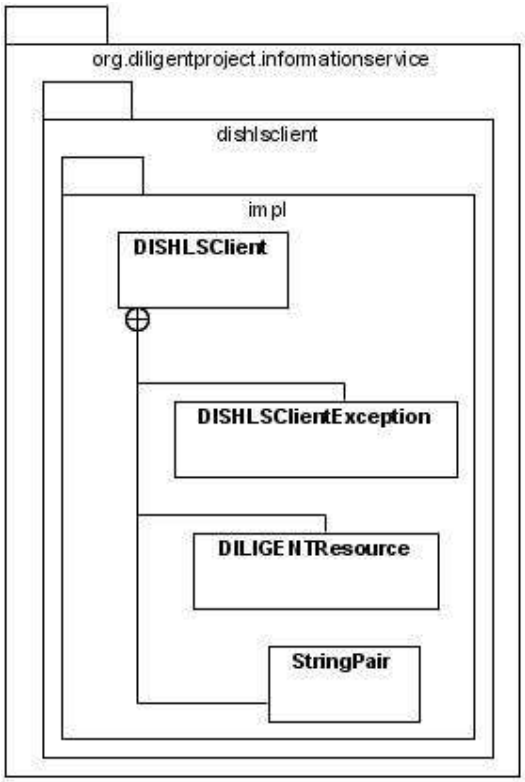

Figure 40. DIS-HLSClient: global class diagram

2.2.5.1 Operations

The operations offered by this library are:

init()

```
static void init()
```

Initialization method used to run the DIS-HLSClient outside the container in a node where the DHN is installed.

Parameters: None.

init

```
static void init(String mapPath, String disQueryPath)
```

Initialization method used to run the DIS-HLSClient outside the container in a node where the DHN is not installed.

Parameters:

- String mapPath: the location on the local file system of the VOMap.xml file.
- String disQueryPath: the location on the local file system of the file with the XQuery templates to send to the DIS-IC.

getAllPublishedEntries

```
String getAllPublishedEntries()
```

It allows retrieving all the resource properties documents published on the DIS-IC.

Parameters: None.

Return: a String containing all the resource properties documents published on the DIS-IC. The result is an XML having this form:

```
<ResultSet>
    <Record>
        <Data>
            WS-Resource Properties xml
        </Data>
    </Record>
    <Record>
    ……….
    </Record
<ResultSet>
```

If the DIS-IC does not contain any WS-Resource Properties the return is `<ResultSet></ResultSet>`.

getResourceEntries

```
String getResourceEntries(EndpointReferenceType eprToQuery)
```

It allows retrieving the resource properties document published from the given EPR. If the publisher service is not a singleton and the method is called with the service EPR, it retrieves all the resource properties document that match the Endpoint Address.

Parameters:

- EndpointReferenceType eprToQuery: this is the EndpointReference to find, it could contain an address or an address and a key.

Return: a String containing the resource properties documents published by the given EndpointReference. The result is an XML having this form:

```
<ResultSet>
    <Record>
        <Data>
            WS-Resource Properties xml
        </Data>
    </Record>
    <Record>
    ……..
    </Record
<ResultSet>
```

If the query does not match any WS-Resource Properties, the return is: `<ResultSet></ResultSet>`.

discoverResourceProperties

`String[][] discoverResourceProperties(EndpointReferenceType eprToQuery)`

It allows querying the DIS-IC in order to find the properties that a resource exposes.

Parameters:

- EndpointReferenceType eprToQuery: this is the EndpointReference to query; it may contain an address or an address and a key.

Return: a 2-dimensional String array that contains in the first column the name of the properties and in the second column their namespace.

getResourcePropertyValue

`String [] getResourcePropertyValue(EndpointReferenceType eprToQuery, String namespace, String localValue)`

It allows querying the DIS-IC in order to find the values of a property that the given EPR publishes.

Parameters:

- EndpointReferenceType eprToQuery: this is the EndpointReference to query; it may contain an address or an address and a key.
- String Namespace: this is the namespace of the property.
- String localValue: this is the name of the property.

Return: an array of strings containing the resource properties values.

getMemberServiceEPRFromAddress

`EndpointReferenceType[] getMemberServiceEPRFromAddress(String addressEPR)`

This method is no longer available in the current version of DIS-HLSClient

getMemberServiceEPRFromPropertyName

`EndpointReferenceType[] getMemberServiceEPRFromPropertyName(String[] propertyNames)`

This method allows retrieving all the Endpoint References that expose the specified Property's LocalNames (i.e., without their URI namespace). By specifying multiple property names, it retrieves the EPRs of all the resources that contain all the specified properties.

Parameters:

- String [] propertyNames: these are the property names (localname without namespace) to look for.

Return: an array of EndpointReferenceTypes that match the query.

Internally the method fills this XQuery template with the appropriated values:

```
for $doc in collection("/db/Properties")//Document where
$doc/Data/child::*[local-name()='*NAME*']/text() eq *VALUE*  return $doc
```

getMemberServiceEPRFromPropertyName

EndpointReferenceType[] getMemberServiceEPRFromPropertyName(String[] propertyNames, String namespaceProperty)

This method allows retrieving all the Endpoints of those resources that publish a resource properties document that contains all the specified properties and it checks if these properties match the desired namespace. If not, it returns an empty array of EndpointReferenceTypes.

Parameters:

- String [] propertyNames: an array of properties local-name to search.
- String namespaceProperty: the namespace of the properties.

Return: an array of EndpointReferenceTypes that match the query.

Internally the method fills this XQuery template with the appropriated values:

```
for $doc in collection("/db/Properties")//Document where
$doc/Data/child::*[local-name()='*NAME*' and namespace-
uri()='*NameSpace*']  return $doc
```

getMemberServiceEPRFromNamespace

EndpointReferenceType[] getMemberServiceEPRFromNamespace (String namespaceProperty)

This method is no longer available in the current version of DIS-HLSClient

getMemberServiceEPRFromPropertyValue

EndpointReferenceType[] getMemberServiceEPRFromPropertyValue(String[] [] propertyNames)

This method allows retrieving all the Endpoints of those resources that publish a resource properties document that matches the specified property names with the specified property values. If not, it returns an empty array of EndpointReferenceType.

Parameters:

- String [] [] propertyNames: the <name=value> couples to search.

Return: an array of EndpointReferenceType that match the query.

Internally the method fills this XQuery template with the appropriated values:

```
for $doc in collection("/db/Properties")//Document where
$doc/Data/child::*[local-name()='*NAME*']/text() eq *VALUE*  return $doc
```

getMemberServiceEPRFromPropertyValue

EndpointReferenceType[] getMemberServiceEPRFromPropertyValue(String[] [] propertyNames, String namespaceProperty)

This method allows retrieving all the Endpoints of those resources that publish a resource properties document that matches the specified property names with the specified property value and that match the specified namespace. Each row of the 2-dimensional array represents a property name and its value. If it does not find anything, it returns an empty array of EndpointReferenceType.

Parameters:

- String [] [] propertyNames: the property-value couples to search.
- String namespace: the namespace of the properties.

Return: an array of EndpointReferenceType that match the query.

Internally the method fills this XQuery template with the appropriated values:

```
for $doc in collection("/db/Properties")//Document where
$doc/Data/child::*[local-name()='*NAME*' and namespace-
uri()='*NameSpace*']/text() eq *VALUE*  return $doc
```

createEPRFromMemberServiceEPR

`EndpointReferenceType[] createEPRFromMemberServiceEPR(String msEPR)`

This method is no longer available in the current version of DIS-HLSClient.

queryDIS

`String queryDIS(String xPathExpression)`

This method allows querying the WS-Resource Properties and profiles registered inside the DIS-IC using an XPath query expression.

Parameters:

- String xPathExpression: the expression to send to the DIS-IC

Return: the result of the query as a String. The result is an XML having this form:

```
<ResultSet>
    <Record>
        <Data>
            WS-Resource Properties
        </Data>
    </Record>
    <Record>
                                    ……….
    </Record
<ResultSet>
```

If the query does not match any resource properties document, the following string is returned:

`<ResultSet></ResultSet>`.

queryRegistryDIS

`String queryRegistryDIS(String xPathExpression)`

This method allows retrieving the DILIGENT Resource Profiles registered inside the DIS-IC matching a given XPath expression.

Parameters:

- String xPathExpression: the expression to send to the DIS-IC.

Return: the result of the query as a String. The result is an XML having this form:

```
<ResultSet>
    <Record>
        <Data>
            DILIGENT Resource Profile
        </Data>
    </Record>
    <Record>
    ……
    </Record
<ResultSet>
```

If the query does not match any resource properties document, the following string is returned:

`<ResultSet></ResultSet>`.

getAllProfilesEntries

`String getAllProfilesEntries()`

This method retrieves from the DIS-IC all the Profiles published in the DIS.

Parameters: none.

Return: all the Profiles found as a String. The result is an XML having this form:

```
<ResultSet>
    <Record>
        <Data>
            DILIGENT Resource Profile
        </Data>
    </Record>
    <Record>
    ……
    </Record
<ResultSet>
```

If the query does not match any resource properties document, the following string is returned:

`<ResultSet></ResultSet>`.

getAllResourceTypeEntries

`String getAllResourceTypeEntries(String resourceType)`

This method retrieves from the DIS-ICs all the Profiles of the given DILIGENT resource.

Parameters:

- String resourceType: the resource type

Return: a string containing all the Profiles retrieved. The string returned has the same structure as the one returned by the getAllProfilesEntries() method.

getAllDHNEntries

`String getAllDHNEntries()`

This method retrieves all the DHN profiles.

Parameters: None.

Return: a string containing all the DHN profiles. The string returned has the same structure as the one returned by the getAllProfilesEntries() method.

getAllGLiteResourceEntries

`String getAllGLiteResourceEntries()`

This method retrieves all the gLiteResource profiles.

Parameters: None.

Return: a string containing all the gLiteResource profiles. The string returned has the same structure as the one returned by the getAllProfilesEntries() method.

getAllDLComponentEntries

`String getAllDLComponentEntries()`

This method is no longer available in the current version of DIS-HLSClient

getAllServiceEntries

`String getAllServiceEntries()`

This method retrieves all the Service profiles.

Parameter: none.

Return: a string containing all the Service profiles. The string returned has the same structure as the one returned by the getAllProfilesEntries() method.

getAllRunningInstanceEntries

`String getAllRunningInstanceEntries()`

This method retrieves all the RunningInstance profiles.

Parameter: none.

Return: a String containing all the RunningInstances profiles. The string returned has the same structure as the one returned by the getAllProfilesEntries() method.

getAllExternalRunningInstanceEntries

`String getAllExternalRunningInstanceEntries()`

This method retrieves all the ExternalRunningInstance profiles.

Parameters: None.

Return: a string containing all the ExternalRunningInstances profiles. The string returned has the same structure as the one returned by the getAllProfilesEntries() method.

getAllCSEntries

`String getAllCSEntries()`

This method retrieves all the CS profiles.

Parameters: None.

Return: a string containing all the CS profiles. The string returned has the same structure as the one returned by the getAllProfilesEntries() method.

getAllCSInstanceEntries

`String getAllCSInstanceEntries()`

This method retrieves all the CSInstance profiles.

Parameters: None.

Return: a string containing all the CSInstance profiles. The string returned has the same structure as the one returned by the getAllProfilesEntries() method.

getAllCollectionEntries

`String getAllCollectionEntries()`

This method retrieves all the Collection profiles.

Parameters: None.

Return: a string containing all the Collection profiles. The string returned has the same structure as the one returned by the getAllProfilesEntries() method.

getAllProfilesID

`String [] getAllProfilesID()`

This method retrieves from the DIS-IC all the DILIGENT Resource IDs of the published DILIGENT resources.

Parameters: None.

Return: an array of String containing the DILIGENT Resource IDs of all the published DILIGENT resources.

Internally the method sends this XQuery query:

```
for $profileID in
collection("/db/Profiles")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource/UniqueID return $profileID
```

getAllResourceTypeIDs

`String [] getAllResourceTypeIDs(String resourceType)`

This method allows querying the DIS-IC in order to find all the DILIGENT Resource IDs of the published DILIGENT resources of the specified type.

Parameters:

- String resourceType: the type of DILIGENT resource to find. The type may be one of the following: "Service", "CS", "CSInstance", "Collection", "RunningInstance", "DHN", "ExternalRunningInstance", "gLiteResource"

Return: an array of String containing the DILIGENT Resource IDs of all the published DILIGENT resources of the specified type.

getAllDHNID

```
String [] getAllDHNID()
```

This method allows querying the DIS-IC in order to find all the DILIGENT Resource IDs of the published DHN DILIGENT resources.

Parameters: None.

Return: an array of String containing the DILIGENT Resource IDs of all the published DILIGENT resources of the specified type, i.e. all the DHNs.

getAllGLiteResourceID

```
String [] getAllGLiteResourceID()
```

This method allows querying the DIS-IC in order to find all the DILIGENT Resource IDs of the published gLiteResource DILIGENT resources.

Parameters: None.

Return: an array of String containing the DILIGENT Resource IDs of all the published DILIGENT resources of the specified type, i.e. all the gLiteResources.

getAllDLComponentID

```
String [] getAllDLComponentID()
```

This method is no longer available in the current version of DIS-HLSClient

getAllServiceID

```
String [] getAllServiceID()
```

This method allows querying the DIS-IC in order to find all the DILIGENT Resource IDs of the published Service DILIGENT resources.

Parameters: None.

Return: an array of String containing the DILIGENT Resource IDs of all the published DILIGENT resources of the specified type, i.e. all the DILIGENT Services.

getAllRunningInstanceID

```
String [] getAllRunningInstanceID()
```

This method allows querying the DIS-IC in order to find all the DILIGENT Resource IDs of the published RunningInstance DILIGENT resources.

Parameters: None.

Return: an array of String containing the DILIGENT Resource IDs of all the published DILIGENT resources of the specified type, i.e. all the RunningInstances.

getAllExternalRunningInstanceID

```
String [] getAllExternalRunningInstanceID()
```

This method allows querying the DIS-IC in order to find all the DILIGENT Resource IDs of the published ExternalRunningInstance DILIGENT resources.

Parameters: None.

Return: an array of String containing the DILIGENT Resource IDs of all the published DILIGENT resources of the specified type, i.e. all the ExternalRunningInstances.

getAllCSID

`String [] getAllCSID()`

This method allows querying the DIS-IC in order to find all the DILIGENT Resource IDs of the published CS DILIGENT resources.

Parameters: None.

Return: an array of String containing the DILIGENT Resource IDs of all the published DILIGENT resources of the specified type, i.e. all the CSs.

getAllCollectionID

`String [] getAllCollectionID()`

This method allows querying the DIS-IC in order to find all the DILIGENT Resource IDs of the published Collection DILIGENT resources.

Parameters: None.

Return: an array of String containing the DILIGENT Resource IDs of all the published DILIGENT resources of the specified type, i.e. all the Collections.

getProfileFromID

`String  getProfileFromID(String ID)`

This method allows querying the DIS-IC in order to retrieve the Profile identified by the given DILIGENT Resource ID.

Parameters:

- String ID: the ID to look for

Return: a String containing the profile of the corresponding DILIGENT Resource.

Internally the method sends this XQuery query:

```
for $DILIGENTResource in
collection("/db/Profiles")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $DILIGENTResource/UniqueID/text()
eq '*ID*' return $DILIGENTResource/Profile
```

getProfileFromID

`String getResourceTypeFromID(String ID)`

This method allows querying the DIS-IC in order to retrieve the ResourceType for given DILIGENT Resources (identified by its ID).

Parameters:

- String ID: the ID to search for

Return: a String corresponding to the DILIGENT Resource type.

Internally the method fills this XQuery template with the appropriated values:

```
for $DILIGENTResource in
collection("/db/Profiles")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $DILIGENTResource/UniqueID/text()
eq '*ID*' return $DILIGENTResource/ResourceType
```

getAuthPoliciesFromID

`String getAuthPoliciesFromID(StringID)`

This method allows querying the DIS-IC in order to retrieve the AuthorizationPolicies of the given DILIGENT Resource (identified by its ID).

Parameters:

- StringID: the ID to search for

Return: a String corresponding to the retrieved Authorization Policy of the given DILIGENT Resource.

Internally the method fills this XQuery template with the appropriated values:

```
for $DILIGENTResource in
collection("/db/Profiles")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $DILIGENTResource/UniqueID/text()
eq '*ID*' return $DILIGENTResource/AuthorizationPolicies
```

queryRegistryDIS

`String queryRegistryDIS (String xPathExpression)`

This method (in the previous version the name was `queryProfiles`) allows directly querying the profiles that the DIS-IC publishes. The XPath Expression is related to the original XML Profile structure and it is dynamically inserted by the code in a wrapper XPath query that allows retrieving from the DIS-IC the desired profile.

Parameters:

- String xPathExpression: This expression is applied to the Profile XML section.

Return: a String containing all the profiles that match the XPath query. The result is an XML having this form:

```
<ResultSet>
    <Record>
        <Data>
            DILIGENT Resource Profile
        </Data>
    </Record>
    <Record>
    ……
    </Record
<ResultSet>
```

If the query does not match any DILIGENT resource profile, the return is: `<ResultSet></ResultSet>`.

Internally the method fills this XQuery template with the appropriated values:

```
for $doc in collection("/db/Profiles")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource/Profile where $doc*XPATH* return $doc
```

queryProfilesForIDAndProfiles

`String [][] queryProfilesForIDAndProfiles (String xPathExpression)`

This method allows directly querying the profiles that the DIS-IC publishes. The xPath Expression is related to the XML Profile structure and it is is dinamically inserted in a wrapper XPath query that allows to retrieve from the DISIC the desired profile. This method returns a multidimensional array that contains in each row:

- the ID of the resource [column 0]

- the Resource Type [column 1]
- the Authorization Policies [column 2]
- the Profile [column 3]

Parameters:

- XPathExpression: This expression is applied to the Profile XML section.

Return: a multidimensional array that contains ID, resourceType, AuthPolicies and profile of the retrieved Profiles.

Internally the method fills this XQuery template with the appropriated values:

```
for $doc in collection("/db/Profiles")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $doc/child::*[local-
name()='Profile']*XPATH* return $doc
```

queryResourceTypeProfilesForIDAndProfiles

`String [][] queryResourceTypeProfilesForIDAndProfiles (String xPathExpression, String resourceType)`

This method allows directly query the profiles that the DIS-IC publishes. The xPath Expression is related to the XML Profile structure and it is dinamically inserted in a wrapper XPath query that allows to retrieve from the DISIC the desired profile of the desidered Resource Type. This method returns a multidimensional array that contains in each row:

- the ID of the resource [column 0]
- the Resource Type [column 1]
- the Authorization Policies [column 2]
- the Profile [column 3]

Parameters:

- xPathExpression: this is the expression that should be applied to the Profile XML section.
- resourceType: it's the type of DILIGENT resource to query.

Return: a multidimensional array that contains ID, resourceType, AuthPolicies, and profile of the retrieved Profiles.

Internally the method fills this XQuery template with the appropriated values:

```
for $doc in
collection("/db/Profiles/*TYPE*")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $doc/child::*[local-
name()='Profile']*XPATH* return $doc
```

queryProfilesForID

`String [] queryProfilesForID (String xPathExpression)`

This method allows to directly query the profiles that the DIS-IC publishes. The XPath expression is related to the XML profile structure and is dynamically inserted in a wrapper XPath query that allows retrieving from the DIS-IC registry the desired profile. This method returns an array that contains in each row the ID of the resource whose profile match the XPath query.

Parameters:

- XPathExpression: This expression is applied to the Profile XML section.

Return: an array that contains the IDs of the corresponding Profiles.

Internally the method fills this XQuery template with the appropriated values:

```
for $doc in collection("/db/Profiles")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $doc/Profile*XPATH* return
$doc/UniqueID
```

queryResourceTypeProfilesForID

`String [] queryResourceTypeProfilesForID (String xPathExpression, String resourceType)`

This method allows directly query the profiles that the DIS-IC publishes. The XPath expression is related to the XML profile structure and is dynamically inserted in a wrapper XPath query that allows retrieving from the DIS-IC registry the desired profile. This method returns an array that contains in each row the ID of the resource whose profile match the XPath query.

Parameters:

- xPathExpression: This expression is applied applied to the Profile XML section.
- resourceType: it's the type of DILIGENT resource to query for.

Return: an array that contains the IDs of the corresponding Profiles.

getAllRunningInstancesOnDHN

`String [] getAllRunningInstancesOnDHN (String DHNId)`

This method allows retrieving all Running Intance IDs of the Running Instance deployed on the DHN identified by the given ID.

Parameters:

- DHNId: the ID of a DHN profile.

Return: an array of IDs corresponding to the Running Instance profiles.

Internally the method fills this XQuery template with the appropriated values:

```
for $doc in
collection("/db/Profiles/RunningInstance")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $doc/child::*[local-
name()='Profile']/DHN[@UniqueID="*VALUE*"] return $doc/UniqueID
```

getElementsFromIDProfile

`String [] getElementsFromIDProfile (String ID, String elementLocalName, String elementNamespace).`

This method queries the DIS-IC in order to retrieve specific Elements of a DILIGENT resource Profile identified by the given ID. If the ID does not have the "elementName" element, it returns null.

Parameters:

- ID: the ID of the DiligentResource to query.
- elementLocalName: the element local name to retrieve
- elementLocalNamespace: the element local namespace to retrieve

Return: an array of String corresponding to all the Elements matching the query.

```
Internally the method fills this XQuery template with the appropriated
values:for $profile in collection("/db/Profiles")//Document where
```

```
$profile/ID/text() eq '*ID*' return $profile/Data/child::*[local-
name()='Profile']/DILIGENTResource/Profile/.//*[local-
name()='*ELEMENTNAME*'][namespace-uri()='*ELEMENTNAMESPACE*']
```

getRISpecificData

String [] getRISpecificData (String serviceClass, String serviceName, String entryName).

This method queries the DIS-IC in order to retrieve the Specific Data section contained in a Running Instance Profile. The Specific Data section of a Running Instance is identified by the Service class and name the Running Instance belongs to, and by the entryName that is one of the endpoint the Runnig Instance exposes.

Parameters:

- serviceClass: the Service Class value of a running instance profile.
- serviceName: the Service Nam value of a running instance profile.
- sentryName: the entryName element of the Running Instance profile.

Return: an array of string each representing the content of a Specific data section matching the specified criteria

Internally the method fills this XQuery template with the appropriated values:

```
for $riProfile in
collection("/db/Profiles/RunningInstance")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource/Profile[AccessPoint/RunningInstanceInte
rfaces/Endpoint[@EntryName="*ENTRYNAME*"]] where
$riProfile/ServiceName/text() eq '*NAME*' and
$riProfile/ServiceClass/text() eq '*CLASS*' return $riProfile/SpecificData
```

getEPRsFromClassAndName

String [] getEPRsFromClassAndName (String serviceClass, String serviceName, String entryName).

This method queries the DIS-IC in order to retrieve the EPRs of the Running Instances generated by the Service having the specified name and classes and providing the EntryName. In the current implementation the methods returns an array of string of URI (in the next planned release it will return an array of EndpointReferenceType).

Parametes:

- serviceClass: the Service Class value of a running instance profile.
- serviceName: the Service Nam value of a running instance profile.
- sentryName: the entryName element of the Running Instance profile.

Return: an array of strings each representing an EPR of a RunningInstance matching the specified criteria

Internally the method fills this XQuery template with the appropriated values:

```
for $riProfile in
collection("/db/Profiles/RunningInstance")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource/Profile where
$riProfile/ServiceName/text() eq '*NAME*' and
$riProfile/ServiceClass/text() eq '*CLASS*' return
$riProfile/AccessPoint/RunningInstanceInterfaces/Endpoint[@EntryName="*ENT
RYNAME*"]/text()
```

queryDISIC

```
String  queryDISIC (String XQuery).
```

Using this method, it is possible to directly query WS-Resource Properties properties as well as profiles by sending an XQuery to the DIS-IC.

In order to define the XQuery, it is important to take into account the organization of Collections managed by the DIS-IC. In Section 2.2.2 the DB structure used internally by the DIS-IC has been illustrated.

Parameter:

- XQuery: a valid XQuery, compliant with the syntax (with the exception of XML Schema-related features) specified in W3C's recommendations outlined in the June 2006 candidate recommendation (http://www.w3.org/TR/xquery/)

Return: a String containing all the records that satisfy the query. The result is an XML having this form:

```
<ResultSet>
    <Record>
            Single Entry of the resultset of the an XQuery query
    </Record>
    <Record>
    ……
    </Record
<ResultSet>
```

If the query does not produce any results the return is: <ResultSet></ResultSet>.

subscribeEPRToTopic

```
EndPointReference[] subscribeEPRToTopic(EndpointReferenceType clientEPR,
QName topic)
```

This method allows contacting DIS-Broker in order to subscribe the given notification consumer to the given topic. The method retrieves the current DIS-Broker EPR using the Node Access Library and exploits DIS-Broker stubs.

Parameters:

- EndpointReferenceType clientEPR: the notification consumer EPR. It could be a different EPR than that of the caller client.
- QName topic: the topic to subscribe to.

Return: the list of notifiers, if any.

unsubscribeEPRFromTopic

```
void unsubscribeEPRFromTopic(EndpointReferenceType clientEPR, QName topic)
```

This method allows contacting DIS-Broker in order to unsubscribe the given notification consumer to the given topic. The method retrieves the current DIS-Broker EPR using the Node Access Library and exploits DIS-Broker stubs.

Parameters:

- EndpointReferenceType clientEPR: the notification consumer `EPR`. It could be a different EPR than that of the caller client.
- QName topic: the topic to unsubscribe to.

Return: void.

2.2.5.2 How to use the component

All DISHLSClient methods are static, so it is not required to create an object of this class to use them (as in the previous version of the component). The DISHLSClient uses an XQuery template file to implement calls to DIS-IC, and at the time of the first method call, internal data structures are loaded with queries template and the appropriate DIS-IC port type.

Therefore, to use the DIS-HLS-Client inside a service running on a DHN, it is required only to import the `DISHLSClient` class and to call the method:

```
import
org.diligentproject.informationservice.dishlsclient.impl.DISHLSClient ;
import
org.diligentproject.informationservice.dishlsclient.impl.DISHLSClientExcep
tion ;
…
DISHLSClient.getAllCSEntries();
…
```

To use the DISHLSClient inside a client that runs in a node where a DHN is installed, it is needed to call the `init()` method:

```
import
org.diligentproject.informationservice.dishlsclient.impl.DISHLSClient ;
import
org.diligentproject.informationservice.dishlsclient.impl.DISHLSClientExcep
tion ;
…
DISHLSClient.init();
DISHLSClient.getAllCSEntries();
```

To use the DISHSLClient inside a client that runs in a node where DHN is not installed, it is required to specify in the `init()` method the location of two files:

- VOMap.xml: it contains the DIS-IC URIs to query;
- DISQueries.xml: the XQuery template file;

```
import
org.diligentproject.informationservice.dishlsclient.impl.DISHLSClient ;
import
org.diligentproject.informationservice.dishlsclient.impl.DISHLSClientExcep
tion ;
…
DISHLSClient.init("<VOMap.path>","<DISQueries.path>");
DISHLSClient.getAllCSEntries();
```

The DISHLSclient class contains also a CLI (Command Line Interface), that can be use to run DISHLSClient methods from command line:

```
java org/diligentproject/informationservice/dishlsclient/impl/DISHLSClient
<VOMap.path> <DISQueries.path>
```

There are also DISHLSClient methods that contact a DIS-Broker in order to subscribe and unsubscribe to/from topics:

- subscribeEPRtoTopic
- unsubscribeEPRfromTopic

This code example describes a client that subscribes to a topic, waiting for notifications for 50 seconds and then unsubscribes it from the topic:

```
public class DISHLSClientSubscribeBroker implements NotifyCallback{
static DISBrokerServiceAddressingLocator brokerLocator = new
DISBrokerServiceAddressingLocator();
      /* This method is called when a notification is delivered */
      public void deliver(List topicPath, EndpointReferenceType producer,
                  Object message) {
            ResourcePropertyValueChangeNotificationElementType notif_elem;
            ResourcePropertyValueChangeNotificationType notif;

            notif_elem =
(ResourcePropertyValueChangeNotificationElementType) message;
            notif =
notif_elem.getResourcePropertyValueChangeNotification();
            System.out.println("Deliver called");
            if (notif != null) {
                  System.out.println("A notification has been delivered");
                  System.out.print("New value: ");
                  System.out.println(notif.getNewValue().get_any()[0]);
            }
      }
      public void run (String topicNamespace,String topicLocalName){

            // The NotificationConsumerManager sets up an endpoint where
            // notifications will be delivered.
            try {
            NotificationConsumerManager consumer;
            consumer = NotificationConsumerManager.getInstance();
            consumer.startListening();
            EndpointReferenceType consumerEPR =
consumer.createNotificationConsumer(this);
            DISHLSClient.init();
            QName topic = new QName(topicNamespace,topicLocalName);
            DISHLSClient.subscribeEPRToTopic(consumerEPR,topic);
            System.out.println("Waiting for notification. Ctrl-C to
end.");
            try {
                  Thread.sleep(50000);
                  } catch (Exception e) {
                        System.out.println("Interrupted while sleeping.");
```

```
                }
            DISHLSClient.unsubscribeEPRFromTopic(consumerEPR,topic);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
      }
      public static void main (String[] args) {
            DISHLSClientSubscribeBroker client = new
DISHLSClientSubscribeBroker();
            client.run(args[0],args[1]);
      }
}
```

2.2.5.3 Implementation Details

2.2.5.3.1 Classes

Every class includes an instance of `org.apache.commons.logging.Log` used by the log4j logging mechanism [4]. All the following classes belong to the package `org.diligentproject.informationservice.dishlsclient.impl`

DISHLSClient

This class is the only entry point to the static methods offered by the library (see Figure 41).

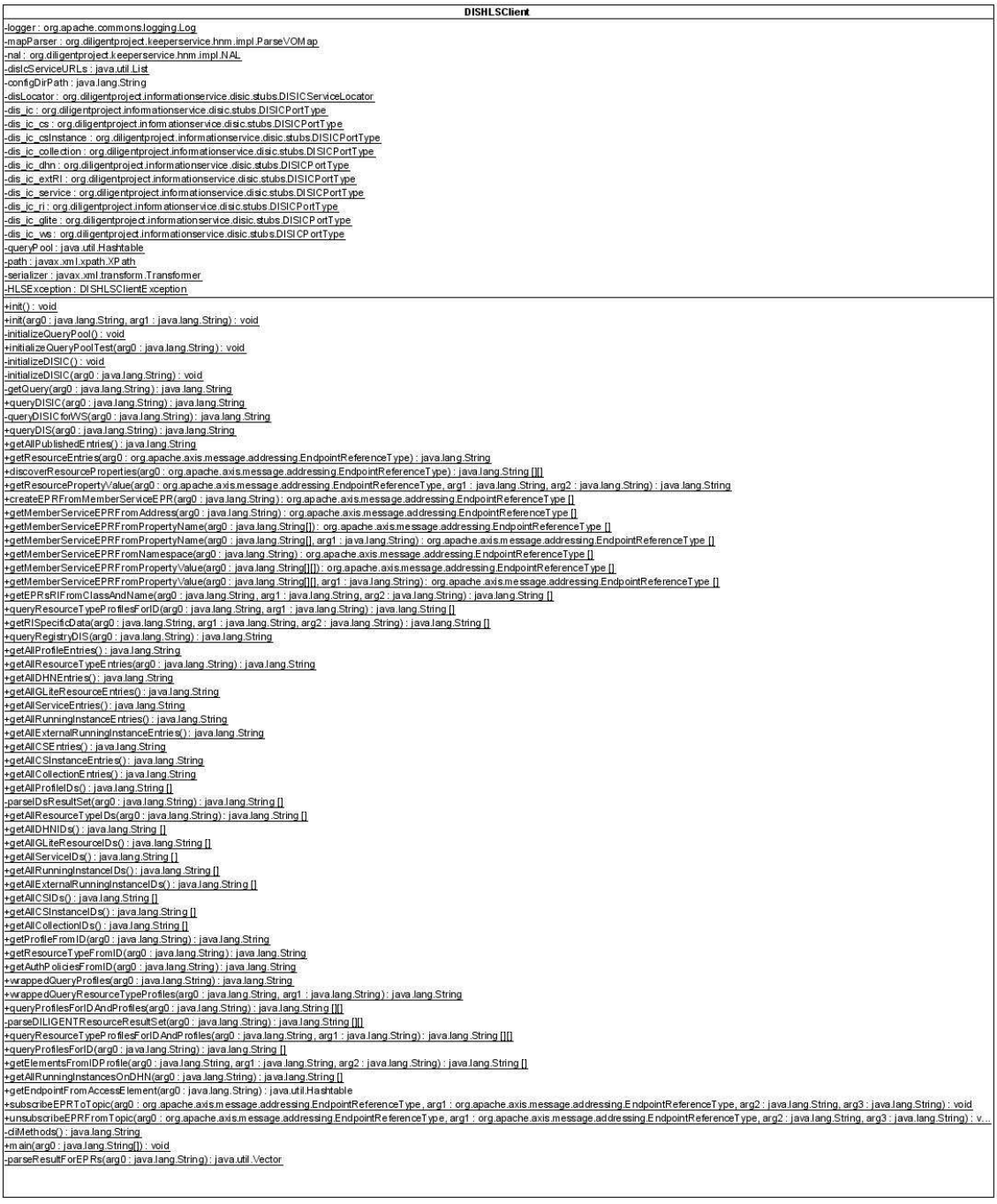

Figure 41. DIS-HLSClient: DISHLSClient class diagram

2.2.5.3.2 Dependencies

This library strictly depends on the organization of the Collections in the DIS-IC XML database (see Section 2.2.2) to build the XPath and XQuery queries to send. The library uses a file (DISQueries.xml) that contains all templates for XQuery queries that are loaded at the first invocation time. The file is located inside the container directory of the HNM service and it is reported in Appendix B.1

The DISHLSClient uses a map file to load the DIS-ICs Endpoints. This file is related to a single VO and it is retrieved using the Node Access Library (see Section 3.2.3). In order to

implement methods for notifications, the DISHLSclient needs also to contact the DIS-Broker and therefore it needs the DIS-Broker stubs. Furthermore, to manage the XPath and XQuery queries, to create DOM Documents, and to contact the DISIC this library needs to import other classes and jars that must be accessible through the classpath:

- xalan.jar
- xercesImpl.jar
- xml-apis.jar
- xsltc.jar
- resolver.jar
- serializer.jar
- org.diligentproject.informationservice.disic.stubs.jar
- org.diligentproject.keeperservice.hnm.nal.jar
- org.diligentproject.informationservice.disbroker.stubs.jar

2.2.6 DIS-Registry

The DIS-Registry service manages DILIGENT Resource profiles (described in D.1.1.2 [1]) for each DILIGENT Resource by providing registration and unregistration facilities for the DILIGENT resources as well as their storage, preservation and publication in the DIS-IC. Regarding the DILIGENT Resource registration, the service:

- validates profiles against DILIGENT resource profiles schema;
- assigns UniqueIds to profiles that don’t have it;
- in case of Service profiles, stores also related packages into PackageRepository;

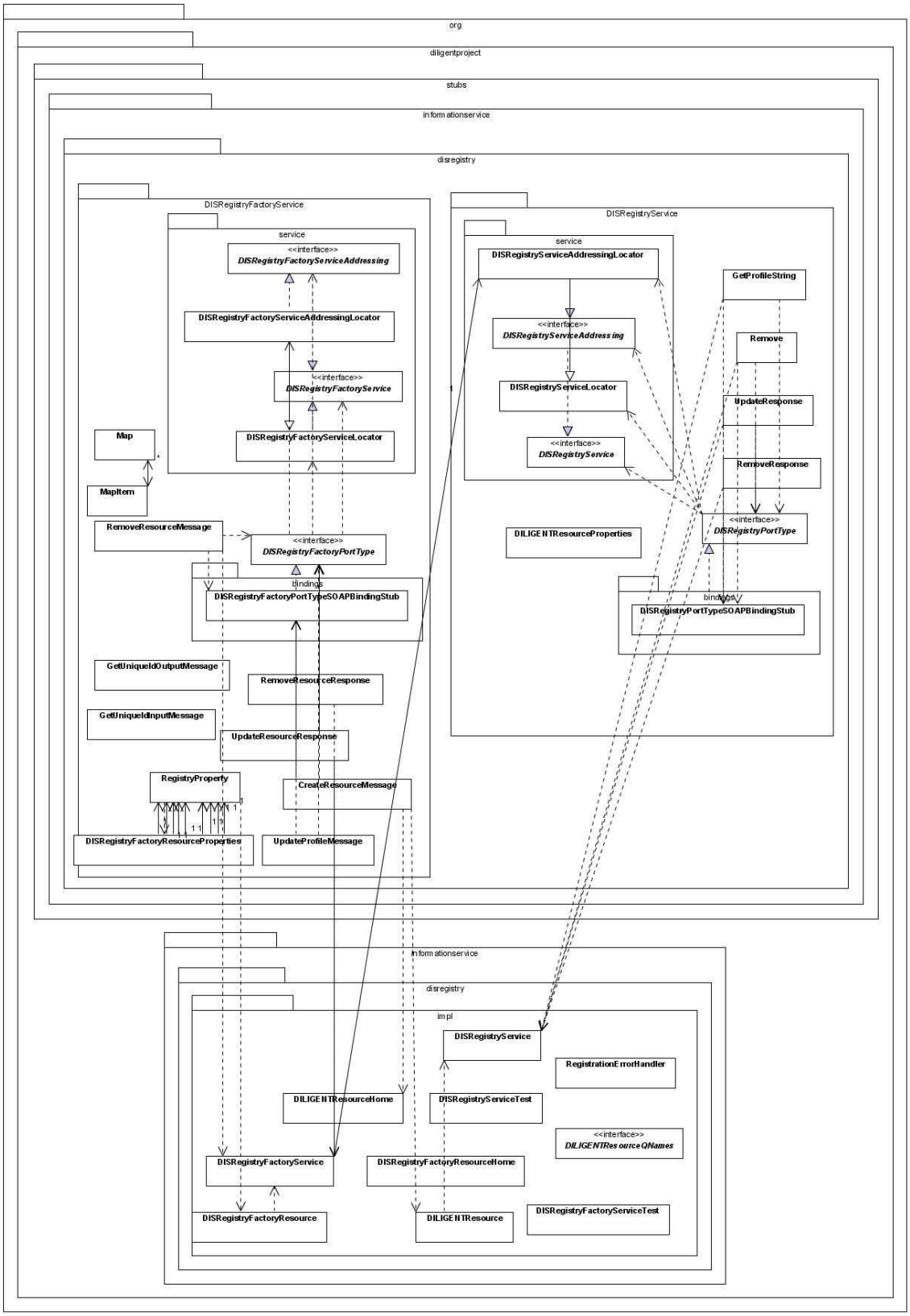

Figure 42. DIS-Registry: global class diagram

2.2.6.1 Profile

The service profile of the DIS-Registry is reported in Appendix A.3.

2.2.6.2 Managed Resources

DIS-Registry

The DILIGENTResource class models the stateful part of the DIS-Registry's WS-Resources. A resource is created each time a new DILIGENT Resource is registered. At creation time, a UniqueID is assigned to the DILIGENT Resource if it is not present yet. A Profile (as described in section 3.1.2 of D.1.1.2) is composed by four fixed elements: UniqueID, ResourceType, AuthPolicies, and Profile.

The UniqueID identifies the resource unambiguously.

The ResourceType discriminates among the different resource types. It is possible to have eight different resource types:

- DHN (Diligent Hosting Node)
- gLiteResource
- Collection
- Service
- RunningInstance
- ExternalRunningInstance
- CS (Compound Service)
- CSInstance (Compound Service Instance)

The AuthPolicies (Authorization Profile) part will be included as soon the integration between this service and the DVOS Authorization Service will be completed. This activity is planned for the DILIGENT Beta prototype release.

The Profile part identifies a resource-specific section that includes the information described in section 3.1.2 of D1.1.2.

The class exposes the "RP_PROFILE" resource property corresponding to the whole resource profile. It QName is the following:

- {http://www.diligentproject.org/namespaces/informationservice/disregistry/DISRegistryService}Profile;

DISRegistryFactoryService

The DISRegistryFactoryService is implemented as a singleton WSRF service with a stateful WS-Resource (named DISRegistryFactoryServiceResource). The resource maintains the mapping between DILIGENT IDs and DISRegistryService's WS-Resource EPRs.

The resource exposes 11 WS-Resource Properties of type RegistryProperty (see section 2.2.6.5.2 for description):

- RP_RICOUNTERRP;
- RP_EXTERNALRICOUNTERRP;
- RP_SERVICECOUNTERRP;

- RP_COLLECTIONCOUNTERRP;
- RP_DHNCOUNTERRP;
- RP_CSCOUNTERRP;
- RP_CSINSTANCECOUNTERRP;
- RP_GLITESECOUNTERRP;
- RP_GLITECECOUNTERRP;
- RP_GLITESITECOUNTERRP;
- RP_GLITESERVICECOUNTERRP.

These properties are updated at profile creation/update/removal time. Each of them maintains the number of DILIGENT Resources of the corresponding type currently registered by the DIS-Registry together with

- the last operation performed
- the related UniqueID,
- the last modified time.

The properties are also registered as Topics in the DIS-Broker, so clients can subscribe to particular events related to DILIGENT Resources (e.g. the death of a Running Instance with a given ID) and receive from the DIS-Registry the appropriate notification.

The DILIGENTResourceQNames class defines the set of DIS-Registry QNames; the corresponding class diagram is shown in Figure 43.

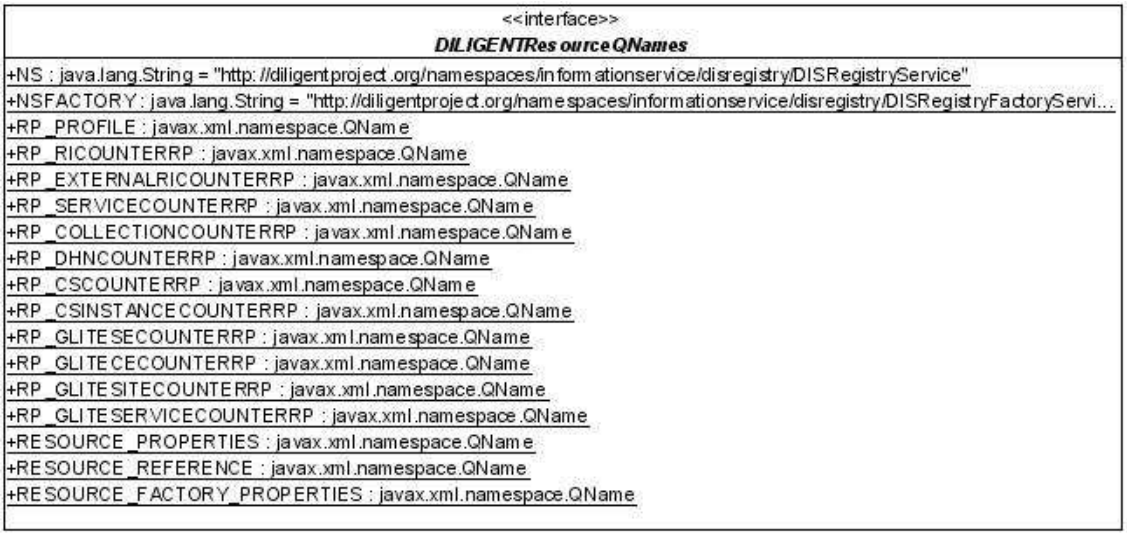

Figure 43. DIS-Registry: DILIGENTResourceQNames class diagram

2.2.6.3 Operations

The DISRegistryFactoryService and the DISRegistryService classes implement the operations globally provided by the DIS-Registry.

DISRegistryFactoryService

createResource

`String createResource (CreateResourceMessage message)`

It allows creating DISRegistryService resources, specifying as parameter (serialized as a String) the resource profile. It returns the UniqueID of the resource created (or the UniqueID pre-specified in the resource profile). This method updates the state of the DISRegistryFactoryService, creating a mapping between the DILIGENT UniqueID and the DISRegistryService WS-Resource EPR.

Parameters:

- CreateResourceMessage message: described later in the stubs section, this parameter contains:
  - String Profile: the String serialization of a DILIGENTResource profile;
  - String RegistrationMode: it can assume one of the following values: `AUTOMATIC_REGISTRATION_MODE` or `REQUEST_REGISTRATION_MODE`. In the first case, the registration ends with the actual modification of the interested VO. In the latter case, the registration ends with a notification to VO managers. In the implementation the DIS-Registry prepared for the alpha prototype, the DISRegistryFactoryService still does not deal with the VOs authorization policies since the integration with the Authorization Service is not yet completed. This feature will be provided with the beta release.
  - String packageURL: during the Service profile registration phase, it is required to specify the location where the software packages are available for downloading. This location is passed to the Package-Repository that will retrieve, validate, and store the software packages implementing the service whose profile is the object of the registration. This parameter is ignored if the given profile is not of type "Service".
  - VO[] suggestedVO: an Array of VOs to register the DILIGENT Resource to. This functionality will be provided with the DILIGENT beta prototype release.

Return: the uniqueID of the resource created.

updateResource

`UpdateResourceResponse updateResource (UpdateProfileMessage message)`

It allows updating an existing DILIGENT Resource profile. In the case that the given resource ID is not present in the DIS, the operation uses the `createResource` operation in order to create a new resource.

Parameters:

- UpdateProfileMessage message: described later on the stubs section, it contains:
  - String diligentID: the DILIGENT ID of the profile to update,
  - String xmlProfile: the serialised string representation of the profile.

Return: None;

removeResource

`RemoveResourceResponse removeResource (RemoveResourceMessage)`

This method allows removing an existing DILIGENT Resource. The removal of an existing resource implies: (i) removing of the mapping information maintained as state of the

service, (ii) destroying of the DILIGENTResource that exposes the related profile, and (iii) calling the remove method of the DIS-IC service.

Parameters:

- RemoveResourceMessage message: described later in the stubs section, it contains:
  - String diligentID: the DILIGENT ID of the profile to delete.
  - String unregistrationMode: this parameter can assume one of the following constant: `AUTOMATIC_UNREGISTRATION_MODE` or `REQUEST_UNREGISTRATION_MODE`. In the first case, the unregistration ends with the actual modification of the interested VO. In the latter case, the registration ends with the notification of the appropriate VO managers. In the implementation the DIS-Registry prepared for the alpha prototype, the DISRegistryFactoryService service still does not deal with the VOs authorization policies since the integration with the Authorization Service is not yet completed. This feature will be provided with the beta prototype release.

DISRegistryService

The following operations are invoked on a qualified EPR. This means that they always operate in the context of a WS-Resource of the DISRegistryService (as described in Section 2.2.6.2).

update

`void update (String Profile)`

It allows updating the given Profile.

Parameters:

- String Profile: serialization of the profile file to update

Return: none.

remove

`void remove ()`

The remove method allows deleting the DILIGENT Resource by destroying the related WS-Resource. Moreover, it deletes from the disk the serialization objects representing them.

Return: none.

getProfileString

`String getProfileString()`

This method returns the Profile managed by the WS-Resource.

Return: the XML string representing the profile.

2.2.6.4 How to use this component

DIS-Registry functionalities may be used only by services entitled to register new DILIGENT Resources, i.e. the following DILIGENT components:

- HNM service (Keeper) in order to create/update DHN profiles and Running Instances profiles;
- DLManagement service (Keeper) in order to remove DHN profiles, create/update/delete Running instance profiles;

- DIS-R-GMAClient service (InformationSystem) in order to create/update/remove gLiteResource profiles;
- CSEngine service (Process Management) in order to create/update/remove CS definition profiles and CS instance profiles;

A DIS-Registry UI has been implemented to simplify, by means of a command line interface, the add/remove/update profile operations:

To add a profile:

```
java DISRegistryUI -factoryUrl:<factoryUrl> -insert -type:<resource_type>
-mode:<reg_mode> -profile:<xmlFile> [-pkg:<pkg_url>] [-vo:<vo_id>]
```

The `package_url` is the location of the package associated with the Service Profile and it is mandatory only for a Service profile registration.

To update a profile:

```
java DISRegistryUI -factoryUrl:<factoryUrl> -update -id:<resource_id> -
profile:<xmlFile>
```

To remove a profile:

```
java DISRegistryUI –remove -factoryUrl:<factoryUrl> -mode:<unreg_mode> -
id:<resource_id>
```

The CLI parameters are the same of the ones explained for the operations of the DISRegistryFactoryService (section 2.2.6.3).

2.2.6.5 Implementation Details

2.2.6.5.1 Class

Every class includes an instance of `org.apache.commons.logging.Log` used by the log4j logging mechanism [4]. All these classes belong to the package `org.diligentproject.informationservice.disregistry.impl`.

DISRegistryService

This class (see Figure 44) represents the stateful part of the DISRegistryService. It implements the operations presented in the previous sections and declared in the DISRegistry.wsdl.

Figure 44. DIS-Registry: DISRegistryService class diagram

DISRegistryFactoryService

This class (Figure 45) implements creation/delete/removal of resources for the DISRegistryService. Moreover, this class registers the DISRegistryFactoryResource WS-Resource-Properties into DIS-IC using DIS-IP library. In the Alpha release, the DIS-IC address is extracted from a local configuration file. With the beta prototype, it will be provided by DLManagement service through the local NAL library. In order to parse this configuration file a utility class has been developed, named `org.diligentproject.keeperservice.hnm.impl.ParseMap`.

It is important to note that it supports replication by means of multiple registration of the created resources in more than one DIS-IC, whilst the partitioning of the registration will be provided with the beta prototype release.

Finally, using the DIS-IP the DISRegistryFactoryService supports the notification of the registered events by means of the registration of its WS-ResourceProperties to the DIS-Broker.

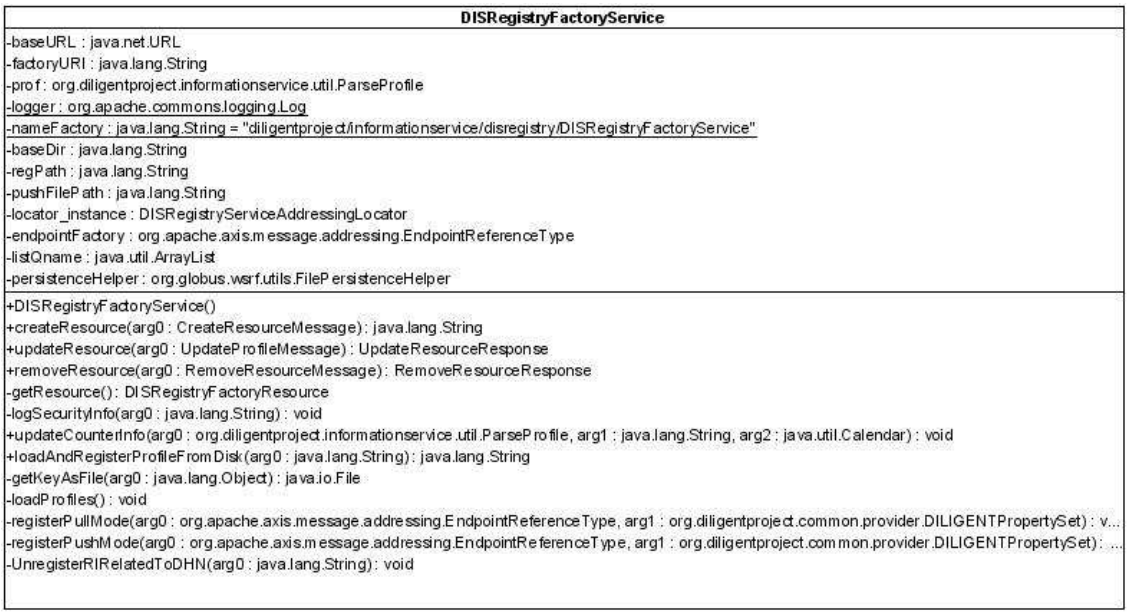

Figure 45. DIS-Registry: DISRegistryFactoryService class diagram

DILIGENTResource

This class (see Figure 46) models the stateful resources of the DISRegistryService. This class publishes the Profile of the DILIGENT Resource it manages as resource properties as previously discussed, and it manages the resource persistence on the disk. By implementing the *PersistenceResource* interface (i.e. its the `load()` and `store()` methods), it's possible to serialize the WS-Resource on disk and recover from Java WS Core container or service crashes. Using the DIS-IP, the DILIGENTResource class implements also the registration of Topics in the DIS-Broker.

Finally, this class implements profile validation against DILIGENT Resource profiles schemas, control of the uniqueness of the resource profile, and in case of Service profiles, it is responsible to contact the Package-Repository to store the related packages.

Figure 46. DIS-Registry: DILIGENTResource class diagram

DILIGENTResourceHome

The DILIGENTResourceHome class (see Figure 47) permits the creation of DILIGENTResource resources, and by extending *ResourceHomeImpl* abstract class, it is responsible to load from local disk persisted resources at service or container start up.

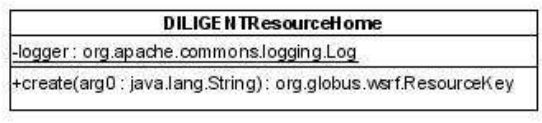

Figure 47. DIS-Registry: DILIGENTResourceHome class diagram

DISRegistryFactoryResource

This class (see Figure 48) models the stateful part of the DIS-RegistryFactory Service following the singleton pattern. The state of this service is maintained by a HashMap handling the relationships between DiligentID and DIS-RegistryService EPR, and by the resource properties described in section 2.2.6.2.

Figure 48. DIS-Registry: DISRegistryFactoryResource class diagram

DISRegistryFactoryResourceHome

This class is the Resource Home of the DIS-Registry-Factory (see Figure 49)

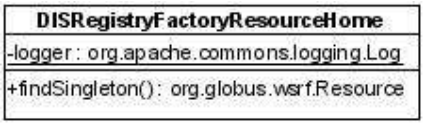

Figure 49. DIS-Registry: DISRegistryFactoryResourceHome class diagram.

2.2.6.5.2 Stubs

org.diligentproject.stubs.informationservice.disregistry.DISRegistryService.DILIGENTResourceProperties

This class represents DILIGENTResourceProperties Set (Figure 50).

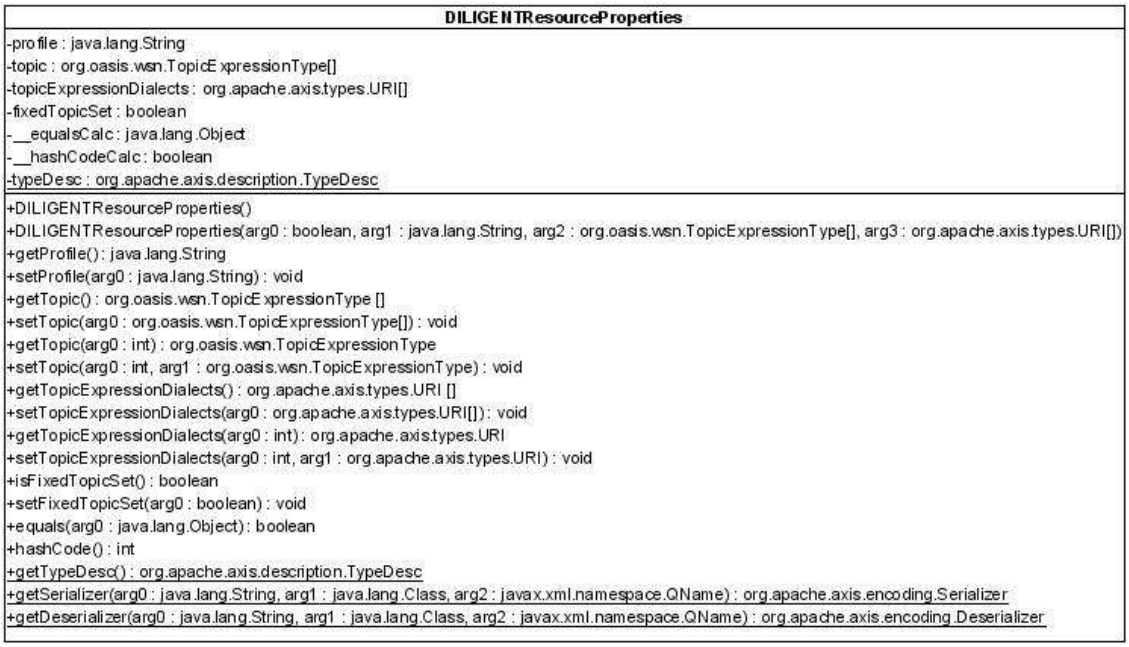

Figure 50. DIS-Registry: DILIGENTResourceProperties class diagram

org.diligentproject.stubs.informationservice.disregistry.DISRegistryFactoryService.DISRegistryFactoryResourceProperties

This class represents DISRegistryFactoryResourceProperties Set.

Figure 51. DIS-Registry: DISRegistryFactoryResourceProperties class diagram

org.diligentproject.stubs.informationservice.disregistry.DISRegistryFactoryService.RegistryProperty

This class (see Figure 52) models the type of all WS-Resource-Properties exposed by the DIS-RegistryFactoryService. The class contains as members:

- operationType: the type of operation performed on the DILIGENTResource (create/delete/update);

- diligentID: the DILIGENT Resource ID last operation refers to;
- counter: the number of resource registered;
- changeTime: last operation time.

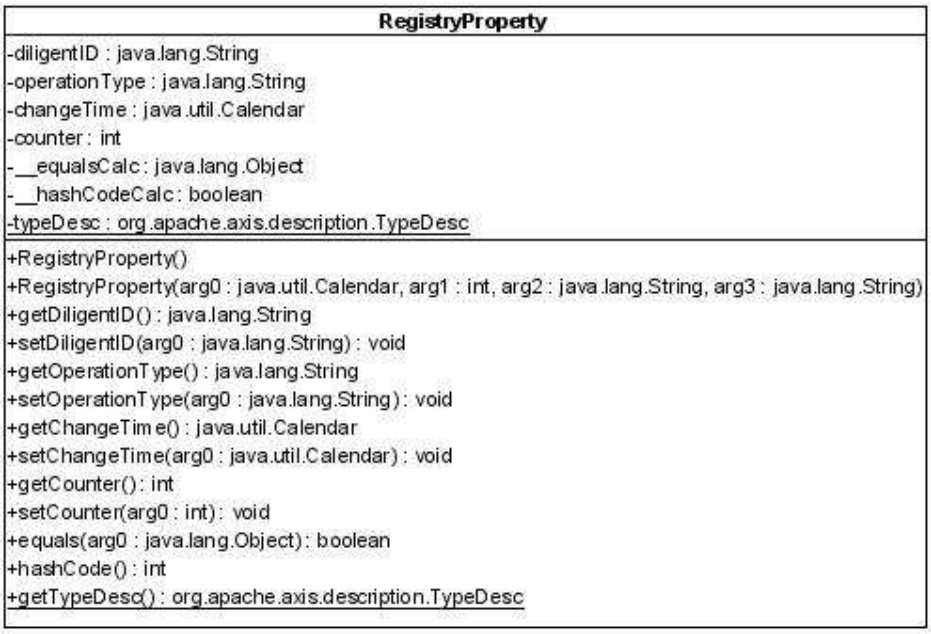

Figure 52. DIS-Registry: RegistryProperty class diagram

org.diligentproject.stubs.informationservice.disregistry.DISRegistryFactoryService.CreateResourceMessage

This class (see Figure 53) contains parameters used for DIS-RegistryService WS-Resource creation (the parameters have been already presented in the `createResource` operation description):

- String profile;
- String registrationMode;
- org.diligentproject.namespaces.dvos.VO[] suggestedVO;
- String packageURL:

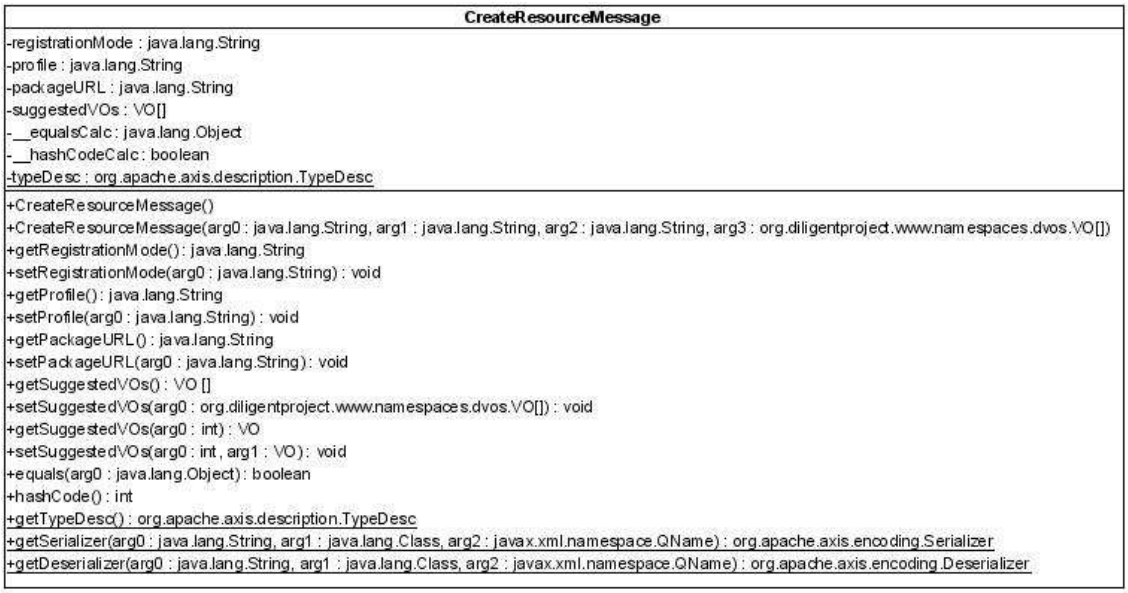

Figure 53. DIS-Registry: CreateResourceMessage class diagram

org.diligentproject.stubs.informationservice.disregistry.DISRegistryFactoryService.UpdateProfileMessage

This class (see Figure 54) is the type of the parameter for `updateResource` method. It contains as members (already presented into `updateResource` operation):

- String diligentID;
- String xmlProfile;

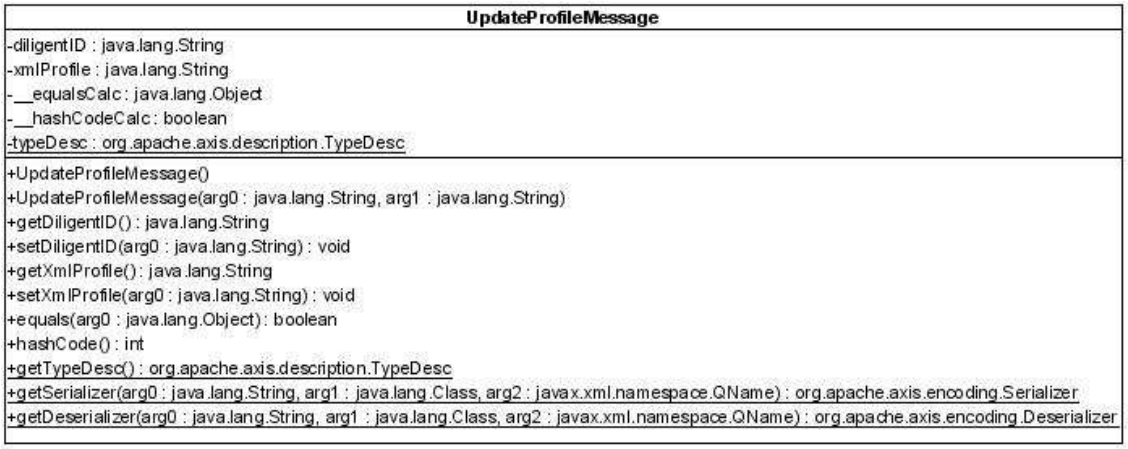

Figure 54. DIS-Registry: UpdateProfileMessage class diagram

org.diligentproject.stubs.informationservice.disregistry.DISRegistryFactoryService.RemoveResourceMessage

This class is the type of the parameter for `removeResource` method. It contains as members (already described into `removeResource` operation):

- String diligentID;

- String unregistrationMode;

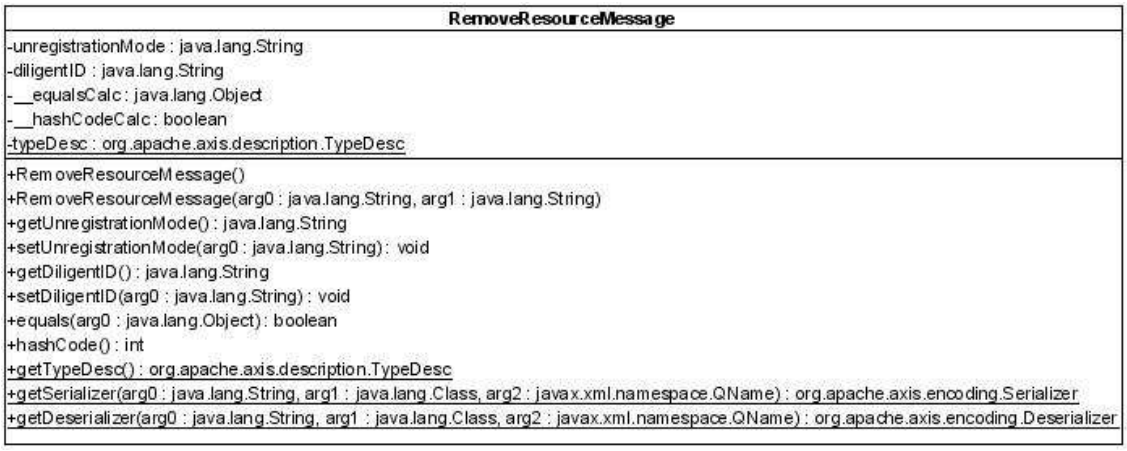

Figure 55. DIS-Registry: RemoveResourceMessage class diagram

2.2.6.5.3 Dependencies

The DIS-Registry Service depends on the following software components:

- DIS-IP library for resource Profiles and WS-Resource-Properties registration;
- DIS-HLSClient library for DIS-IC querying support;
- Package-Repository stubs for package storing;
- DVOS-Common library;
- DILIGENT Provider;

2.2.7 DIS-Broker

This DIS-Broker (see Figure 56) is a WSRF service that adopts the singleton pattern. It provides subscription-brokering capability allowing to a client (i.e. the client part of a service) to subscribe to a topic without knowing the location of the notification producer (the Endpoint Reference identifying it). Moreover, the DIS-Broker service supports the subscription to a topic that the producer has not yet exposed. In this case, it maintains the pending subscription until the notification producer registers its topic.

The implementation of this service is not based on the brokered notification schema since the DIS-Broker subscribes the consumer directly to the notification producer. In that way there is not a central service that is in charge to receive notifications from producers and to deliver them to the right client (as in the brokered notification architecture). However, the DIS-Broker continues its mediation even after the brokerage of the subscription. In fact, the DIS-Broker automatically manages the relocation of notification producers by forwarding the client subscription requests to the new instances of the producers.

This first service implementation prepared for the alpha prototype release also supports the transparent subscription/re-subscription of the client to the new producer instances that comes up after the initial request of subscription.

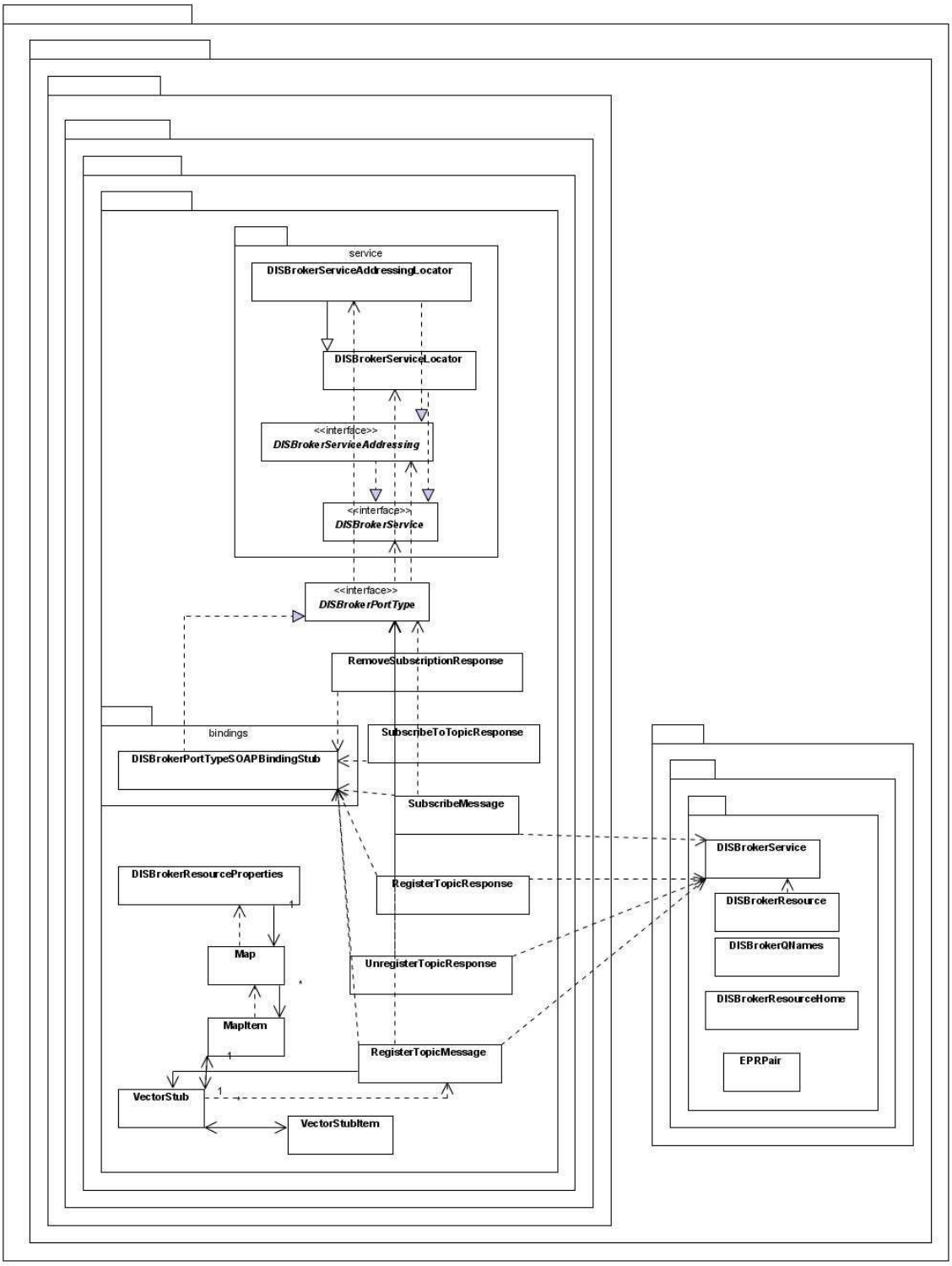

Figure 56. DIS-Broker: global class diagram

2.2.7.1 Profile

The Service Profile of DIS-Broker is reported in Appendix A.4.

2.2.7.2 Managed Resources

In order to register the WS-Resource Properties, the DIS-Broker contacts the DIS-IC by using the local DIS-IP library. The service exposes one resource property:

- RP_TOPICMAPPING {http://diligentproject.org/namespaces/informationservice/disbroker/DISBrokerService}TopicMapping;

This property represents a mapping between topics and the related notification producers. The property can be used to circumvent the DIS-Broker in the subscription/unsubscription phases. In fact, a notification consumer can query the DIS (using DIS-HLS-Client) to find the list of EPRs that expose a particular topic and then it can directly contact them. However, the direct subscription transfers to the client the obligation to implement the advanced features provided by the DIS-Broker and therefore this possibility is strongly discouraged.

2.2.7.3 Operations

The DIS-Broker Service exposes a set of operations for topic registration/unregistration and for subscription/unsubscription. Even if these methods can be directly accessed by calling the DIS-Broker service, it is recommend to use the DIS libraries, i.e. the DIS-IP and the DIS-HLS-Client (the related methods are described in DIS-IP and DIS-HLS-Client sections).

The operations implemented in the DISBrokerService class are:

registerTopic

```
RegisterTopicResponse registerTopic (RegisterTopicMessage message)
```

This operation allows the DILIGENT services to publish their Topics in the DIS infrastructure. It updates the service status by modifying the collection, named topicMapping HashMap, where the mappings between topics and the producer EPR are stored. In the case of pending subscription requests, this method subscribes consumer to the given EPR. It also automatically registers existing consumers interested in the topics to newly registered producers.

Parameters:

- RegisterTopicMessage message: described later in the stubs section, it contains:
  - EndpointReferenceType producerERP: the producer EPR that exposes the Topics;
  - VectorStub topicVector: a vector of Topics that the producer wants to register.

Return: none;

unregisterTopic

```
UnregisterTopicResponse unregisterTopic(RegisterTopicMessage message)
```

This operation allows the DILIGENT services to un-publish their Topics (or a part of them) from the DIS infrastructure. It updates the service status by modifying the collection, named topicMapping HashMap. If no more services expose the same Topics, it removes the entries referred by the given Topics from the internal collection and it un-subscribes all the related consumers.

Parameters:

- RegisterTopicMessage message: described later in the stubs section, it contains:

- EndpointReferenceType producerERP: the producer EPR that wants to un-register its Topics;
- VectorStub topicVector: a vector of Topics that the producer wants to un-register.

Return: None;

subscribeToTopic

SubscribeToTopicResponse subscribeToTopic(SubscribeMessage message)

This method allows consumers to subscribe to a particular Topic (without knowing the EPR of the producers). The consumer is subscribed to all services that expose the given Topic: then it is up to consumers to filter the inappropriate notification message. A consumer can also subscribe itself to a Topic that has not yet published on the DIS-Broker: in this case, the subscription request remains pending until a service registers the given topic. It will also be subscribed to newly appearing producer, even after a successful subscription.

According to the Java WS Core implementation of the WS-Base Notification specification. the producer is in charge to send notifications to consumers. Whilst the DIS-Broker is in charge of dispatching subscription requests to the appropriate producers.

Parameters:

- SubscribeMessage message: described later on the stubs section, it contains:
  - EndpointReferenceType clientEPR: the consumer EPR;
  - String topic: the topic (as a serialized String of the topic QName);

Return: none;

removeSubscription

void removeSubscription(SubscribeMessage message)

This method allows consumers to remove the subscription to a particular topic. The service updates its internal status and destroys *the Subscription Reference* on the consumer side.

Parameters:

- SubscribeMessage message: described later on the stubs section, it contains:
  - EndpointReferenceType clientEPR: the consumer EPR;
  - String topic: the topic string notation;

Return: none;

2.2.7.4 How to use this component

The DIS-Broker service can be seen as a registry supporting subscription mediation between notification consumers and producers. The producers publish the Topics on which they want to support notification for changes, and consumers contact the DIS-Broker in order to subscribe to/unsubscribe from a topic.

The interaction between consumers/producers and the DIS-Broker is implemented by the DIS libraries, DIS-IP and DIS-HLS-Client. A notification producer to register/un-register topics uses the DIS-IP; while on the other hand, a notification consumer exploits the DIS-HLS-Client capabilities to subscribe/unsubscribe to/from Topics. Detailed descriptions are reported in the 2.2.1.3 and 2.2.5.2 Sections.

2.2.7.5 Implementation details

Every class includes an instance of `org.apache.commons.logging.Log` used by the log4j logging mechanism [4]. All these classes belong to the package `org.diligentproject.informationservice.disbroker.impl`

DISBrokerService

This class (see Figure 57) implements all the operations exposed in the WSDL interface (*DISBroker.wsdl*). It registers the WS-Resource Property *TopicMapping* into the DIS-IC by using the local DIS-IP library.

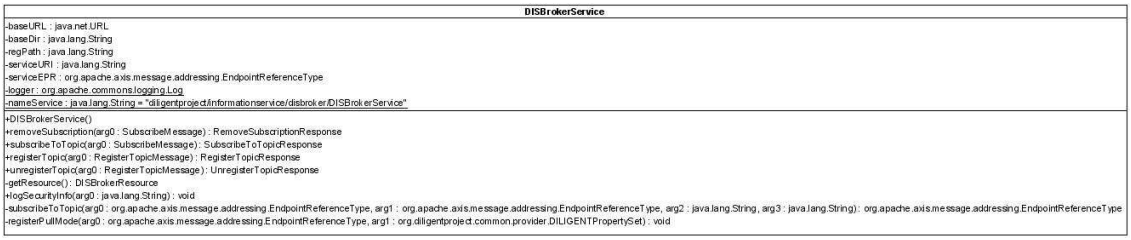

Figure 57. DIS-Broker: DISBrokerService class diagram

DISBrokerResource

This class (see Figure 58) models the singleton resource handling the service status. It maintains the following mappings:

- Topic <-> a Vector of related producers;
- EPRPair <consumerEpr,topic> <-> the related SubscriptionReference EPR;

In addition, it maintains a list of couples <consumerEPR, topic>, i.e. consumers that are waiting for subscriptions.

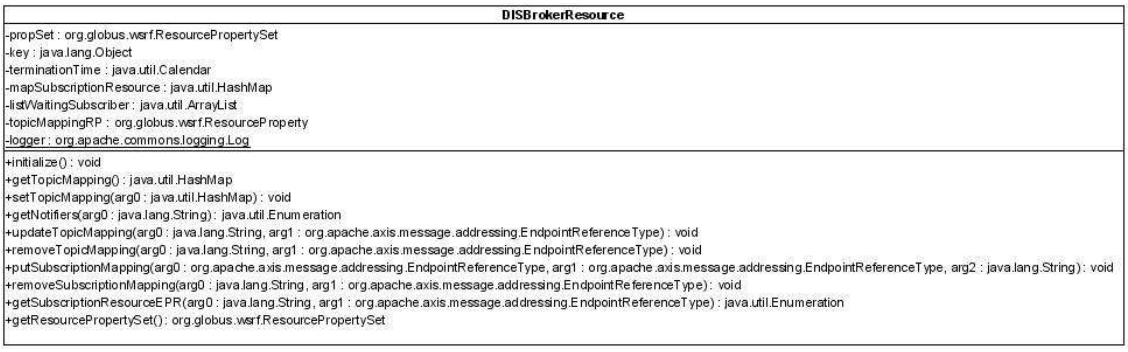

Figure 58. DIS-Broker: DISBrokerResource class diagram

2.2.7.5.1 Stubs

All stubs classes generated at compilation time are depicted in Figure 59, Figure 60, Figure 61, Figure 62 and Figure 63

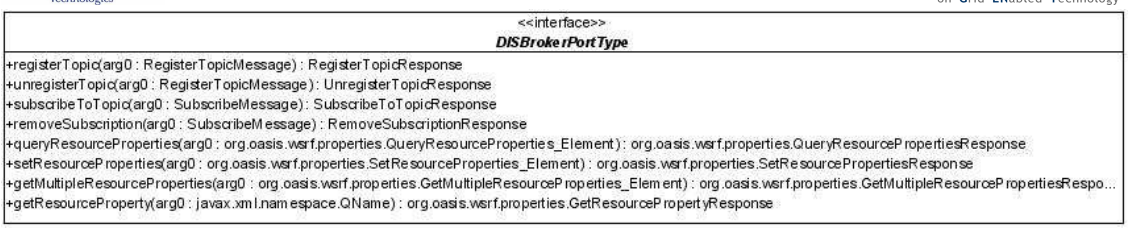

Figure 59. DIS-Broker: DISBrokerPortType Class Diagram

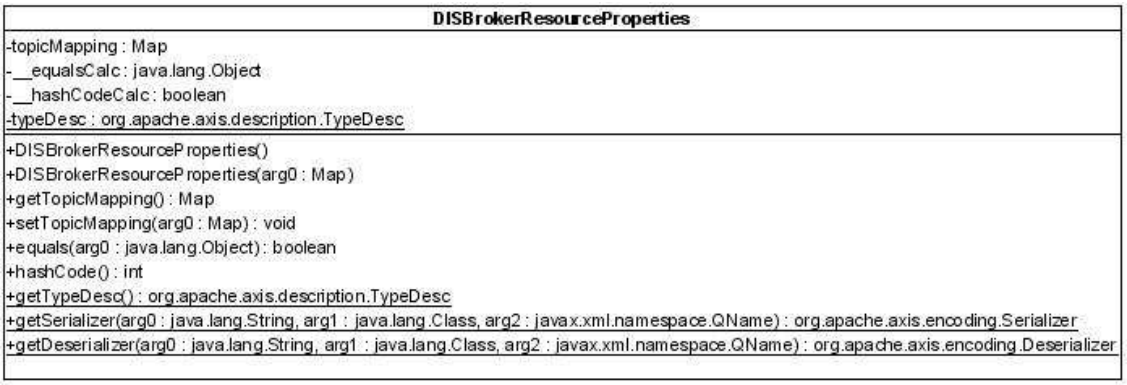

Figure 60. DIS-Broker: DISbrokerResourceProperties class diagram

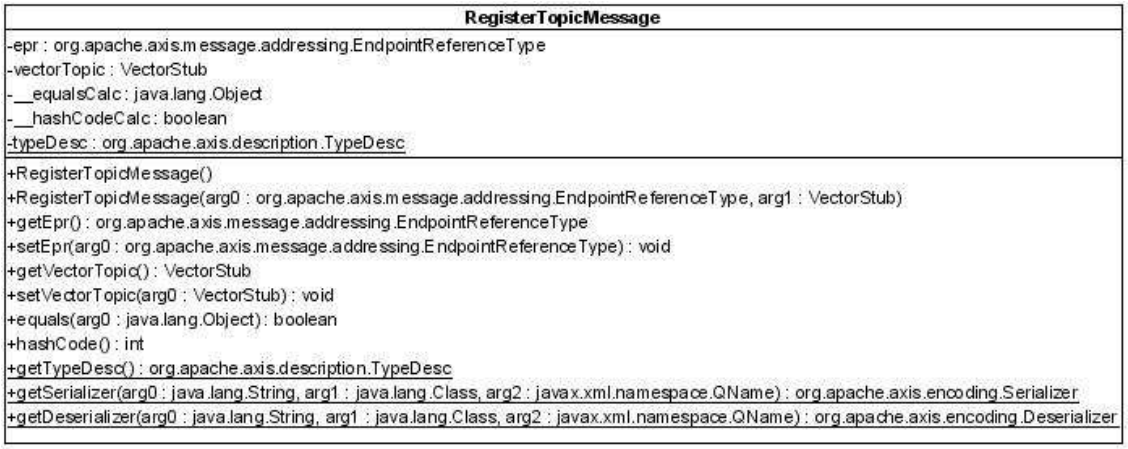

Figure 61. DIS-Broker: RegisetrTopicMessage class diagram

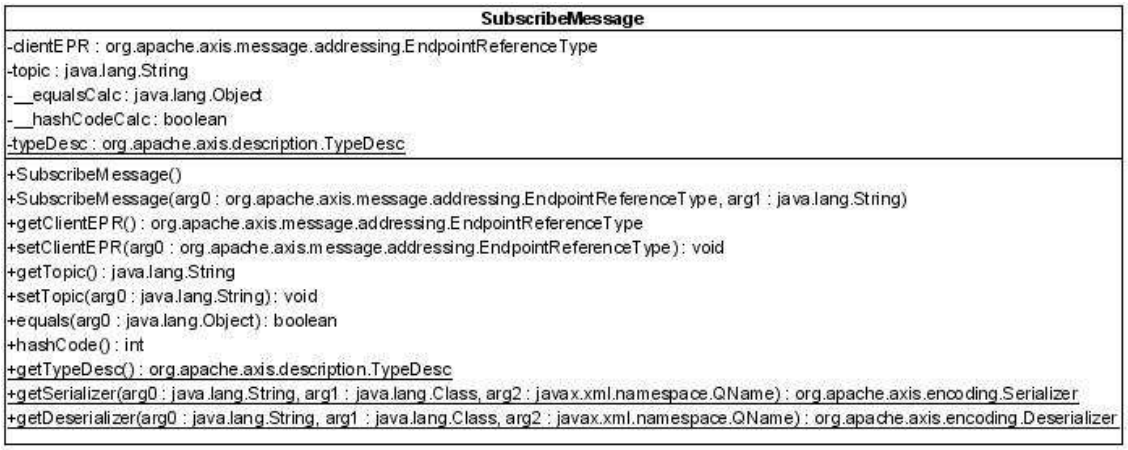

Figure 62. DIS-Broker: SubscribeMessage class diagram

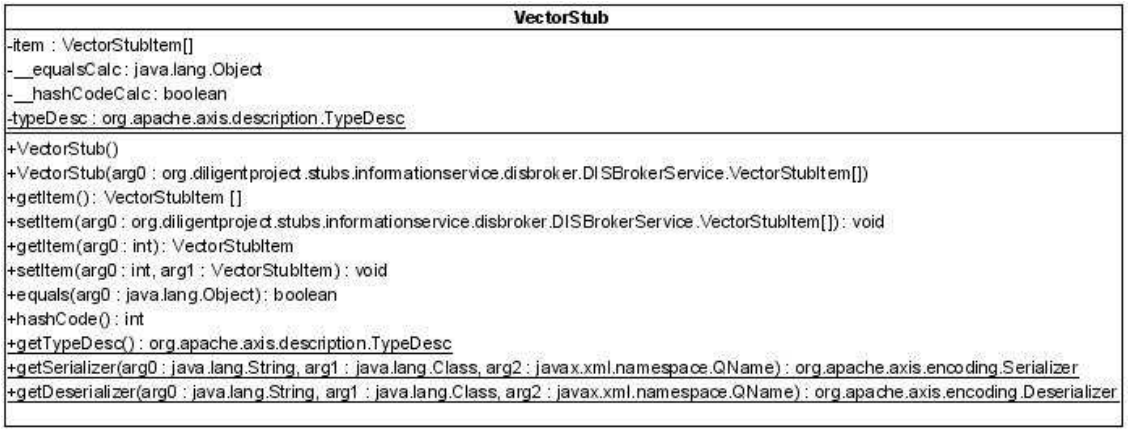

Figure 63. DIS-Broker: VectorStub class Diagram

2.2.7.5.2 Dependencies

The dependencies of the DIS-Broker Service are:

- DILIGENTProvider: to publish WS-Resource Properties in the DIS
- DIS-IP: to publish WS-Resource Properties in the DIS and to register WS-Resource Properties in the DIS.

3 KEEPER SERVICE

3.1 Introduction

The Keeper service takes care of the DL creation and maintenance phases by instantiating the appropriate resources and authorizing users to get access to those resources. In order to perform this activity, the Keeper service supports the package deployment, the storage of DILIGENT resources packages, and the dynamic resources relocation.

The Keeper service is a logical service composed by four software components (see D1.1.2 and D1.2.2 for details) depicted in Figure 64 and described below:

- Package Repository. It validates, stores, and manages the DILIGENT resource packages. It checks packages dependencies and giving access to the stored packages supports the dynamic packages deployment. This repository accepts registration requests coming from the DIS-Registry service, whilst accepts access requests only by the HNM service.
- DL Management. This component coordinates the packages deployment process when a new DL is instantiated and during its lifetime. The operational context that transforms a set of distributed deployed DILIGENT resources into a single application is managed by this component by constructing a DL Map, i.e. a map containing the DL resources locations, their configuration, and the rules to access them.
- Hosting Node Manager (HNM). It manages the Diligent Hosting Node (DHN) by providing the context to deploy the DILIGENT packages accordingly to the DL Management instructions. In particular, when the HNM is deployed, it controls the software dependencies by using the Package Repository, and then it automatically downloads and deploys all the DHN mandatory packages. It also deploys by default the shared Node Access Library (NAL) that exposes a uniform API allowing to query the current node configuration on the node. Moreover, it creates and publishes into the DIS the profiles of all Running Instances deployed in the Java WS Core. Clearly, the HNM must be pre-deployed on each DHN of the DILIGENTinfrastructure.
- K-UI. The K-UI is a graphical user interface that provides the support for all administrative tasks and grants a vision of the DL status.

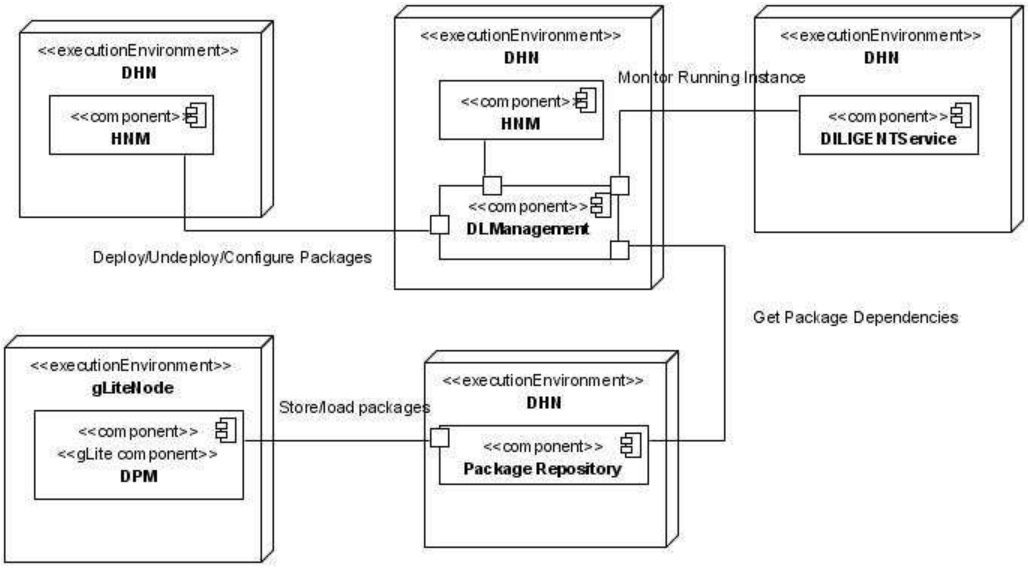

Figure 64. Keeper Service: Main Components and relations

3.2 Components

3.2.1 Package Repository

This component is responsible for the validation, storage, and maintenance of the DILIGENT resource packages. It checks packages dependencies and by ensuring access to them, it supports the dynamic packages deployment.

The Package repository is a WSRF service, implemented as specified by the Singleton Pattern [20]. Accordingly, it splits the implementation into three sub-components: (i) the resource itself that contains Package Repository state, (ii) the resource home that manages the resource, and (iii) the web service that gives access to components functionality. Figure 65 reports the Package Repository class diagram.

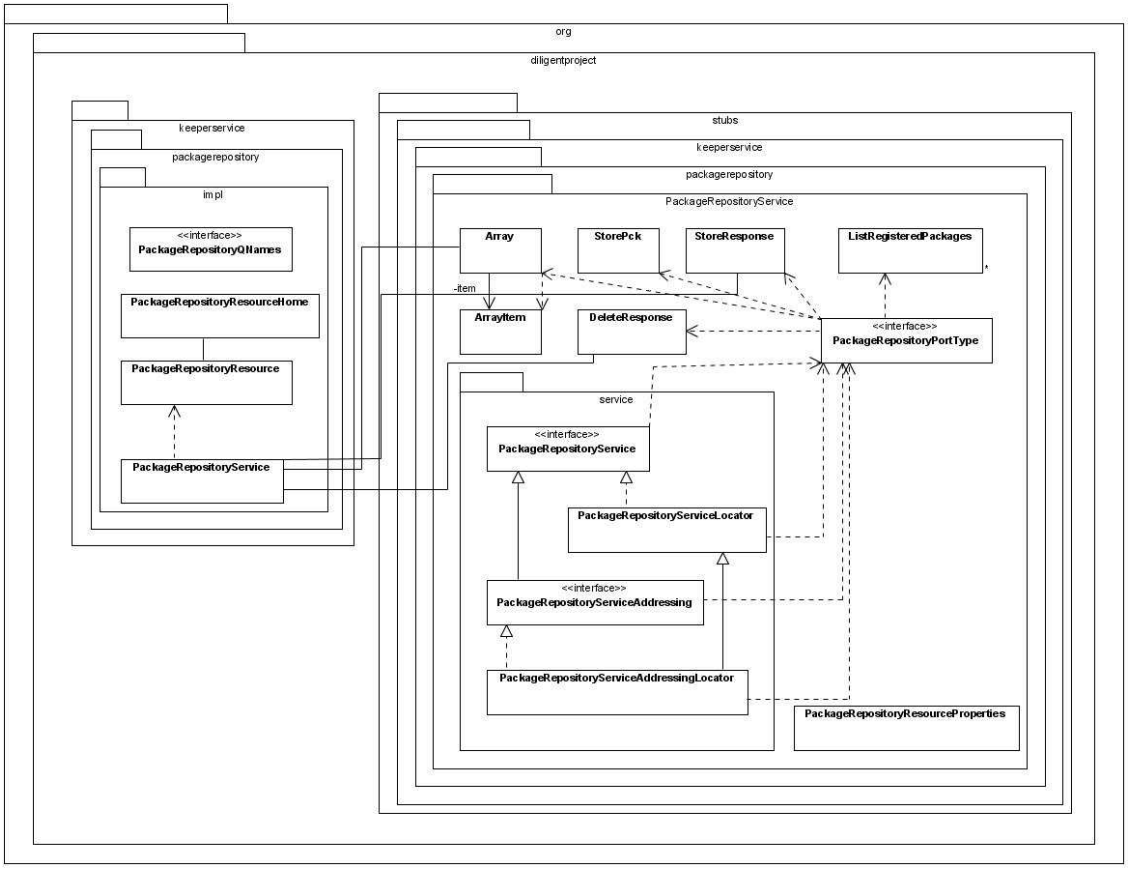

Figure 65. Package Repository: global class diagram

The resource state contains information about the stored software packages. The resource home is used by the web service to create and find the resource. The web service provides to the clients, e.g. DIS-Registry service, access to the Package Repository functionality (see Section 3.2.1.1 below).

3.2.1.1 Managed Resources

Package Repository resource is composed by one Singleton resource with information about the location where DILIGENT software packages are stored. The storage is on a Storage Element (SE) on the Grid even if it also manages a local cache.

When a client, i.e. the DIS-Registry Service, wants to store a package, it gives to the Package Repository service the DILIGENT ID and the URI where the package is available for

download. Instead, when the HNM wants to get a package, it gives to the Package Repository the DILIGENT ID and it gets back the URI from where it is possible to download the package.

3.2.1.2 Operations

The signatures of the actual implemented methods are presented below. These operations allow uploading, deleting, list, and getting a DILIGENT package.

storePck

```
String storePck (StorePck storePackage)
```

This operation stores a package into the Package Repository.

Parameters:

- StorePck: class that contains the `DILIGENT ID` and the `URI` where the package to upload is available.

Return: String with the description of the result.

delete

```
String delete (String diligentID)
```

This operation deletes a stored DILIGENT package from the Package Repository. This implies the removal of the software package from the Grid and from the local cache.

Parameters:

- DILIGENT - ID to delete.

Return: String with the description of the result. Returned values are: "DELETED" or "NOT DELETED".

listRegisteredPackages

```
Array listRegisteredPackages ()
```

This operation lists all IDs of the registered DILIGENT packages.

Parameters: None.

Return: an array with all the DILIGENT IDs of the successfully registered packages.

get

```
String get (String diligentID)
```

The method returns the URI from where the package identified by the specified DILIGENT ID can be downloaded.

Parameters:

- DILIGENT - ID to get.

Returns: the package URI.

3.2.1.3 How to use the component

Access to this component has to be done, as for any other WSRF service, by creating the appropriate portType connected to the EPR of an active instance of the service using the stubs classes distributed with the component.

See section 3.2.1.1 for the signatures.

3.2.1.4 Implementation Details

3.2.1.4.1 Classes

All the following classes belong to the package
`org.diligentprojects.keeperservice.packagerepository.impl`

PackageRepositoryService

This is the service class (see Figure 66) of the Package Repository. It exposes methods to store, delete, get, and list packages.

Figure 66. Package Repository: PackageRepositoryService class diagram

PackageRepositoryResource

This class models the statefull part of the service (see Figure 67). In particular, it maintains information about the Package Repository storage.

Figure 67. Package Repository: PackageRepositoryResource class diagram

PackageRepositoryResourceHome

This is the Resource Home of the service.

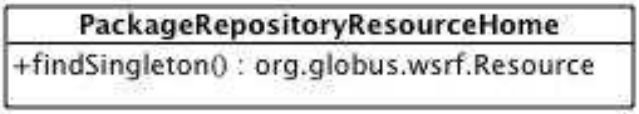

Figure 68. Package Repository: PackageRepositoryResourceHome Class Diagram

3.2.1.4.2 Stubs

All the following stubs belong to the package

`org.diligentprojects.stubs.keeperservice.packagerepository.PackageRepository.`

StorePck

StorePck is the type of the parameter to pass to the `StorePck` () operation of the Package Repository. It contains the `DILIGENT ID` and the `URI` where the package to upload is available.

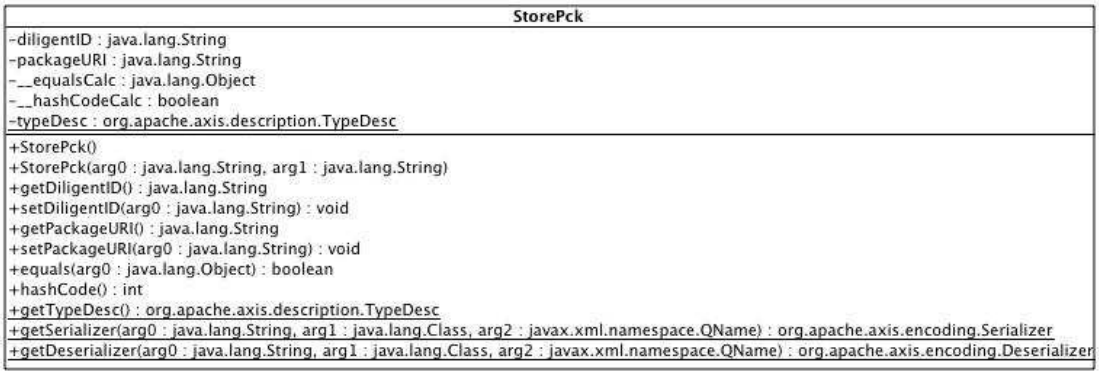

Figure 69. Package Repository: StorePck Class Diagram

Array

This is the returned type of the `listRegisteredPck()` operation.

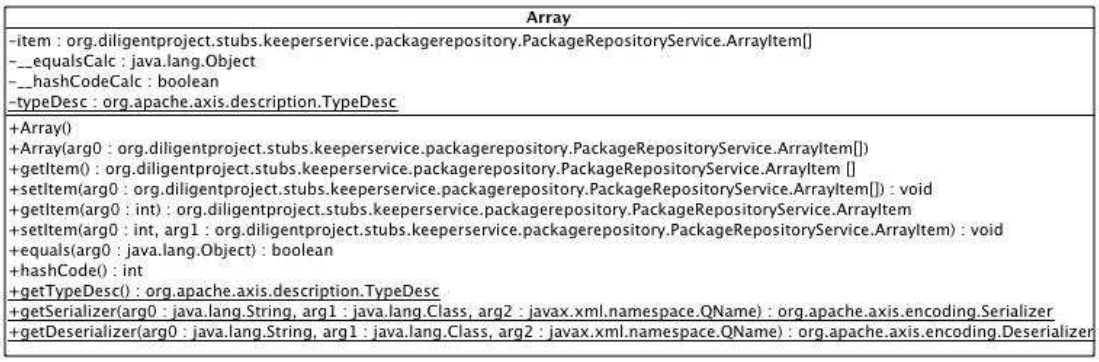

Figure 70. Package Repository: Array Ccass diagram

ArrayItem

The elements of the previous Array class.

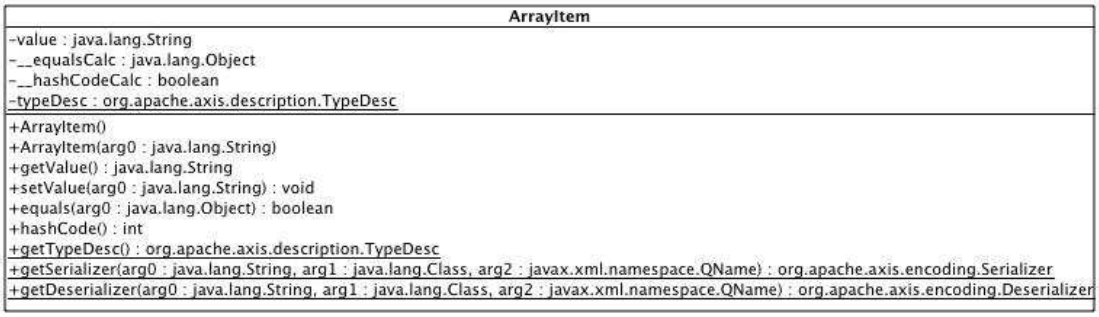

Figure 71. Package Repository: ArrayItem class diagram

3.2.1.4.3 Dependencies

The Package Repository exploits the gLite storage components with a custom java API designed and implemented to access to the existing LFC C API.

3.2.2 DL Management

DL Managament Service groups Keeper functionalities that are in charge of managing and monitoring the service instances forming a DL (see Figure 72).

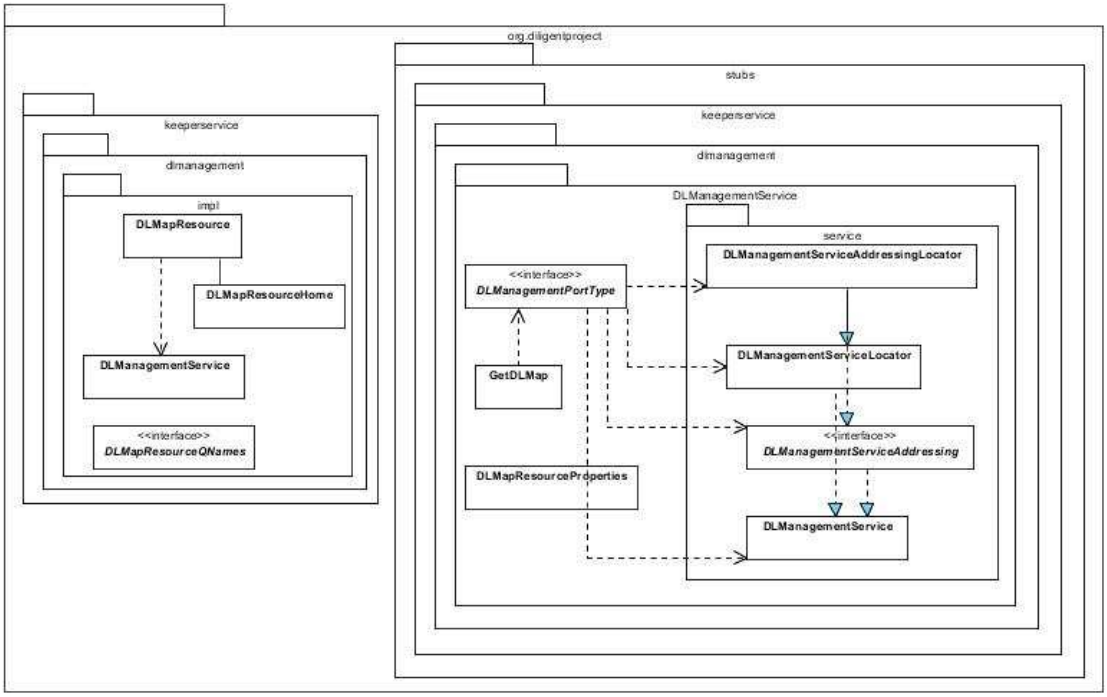

Figure 72. DLManagement: global class diagram

The DL Management creates and maintains the DL Maps and VO Maps.

A *DL Map* is the container of all the information needed to manage the group of services that form the DL. In the DILIGENT distributed environment, the DL Map defines the application context among all services forming the Keeper service class.

A *VO Map* is the group of non DL-specific services acting at VO level (such as the Package Repository and the DIS-Registry) within a particular VO. It allows Collective Layer services to work properly by connecting each other without the need to query the (possible) not available DIS.

3.2.2.1 Profile

The profile of the DL Management service is reported in Section A.9.

3.2.2.2 Managed Resources

The first version of this service prepared for the Alpha prototype release, manages the dissemination of static DLMaps. A statefull resource is created for each active VO or DL. The WS-Resource Property managed by this service is a XML string representing the VOMap or the DLMap.

The structure of the DLMap file is reported in Appendix C.2

The QNames of the service are expressed into DLMapResourceQnames class (see Figure 73).

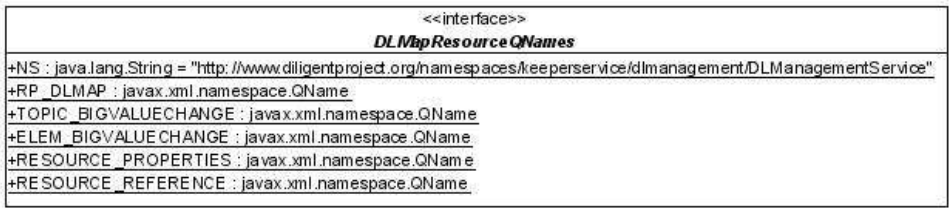

Figure 73. DLManagement: DLMapResourceQnames class Diagram

3.2.2.3 Operations

The functionalities exposed by this service, are the ones concerning the dissemination of DLMap/VOMap Resource Properties.

Clients can query the WS-Resource representing DLMaps or VOMaps or they can subscribe WS-Resource for notifications on DLMap values changes.

getDLMap

`String getDLMap()`

This operation returns the current DLMap Resource Properties value.

Parameters: none;

Return: a String with the current DLMap value.

getVOMap

`String getVOMap()`

This operation returns the current VOMap Resource Properties value.

Parameters: none;

Return: a String with the current VOMap value.

This operation will be supported in the Beta release.

subscribe

`_SubscribeMessage subscribe (org.oasis.wsn.Subscribe subscribemessage)`

The globus standard operation provider guarantees this operation. By implementing the Base-Notification mechanism, this kind of WS-Resource can be set up as notification producer towards clients.

Parameters:

- org.oasis.wsn.Subscribe subscribemessage: contains 2 fields:
  - EndpointReferenceType consumerReference: client `EPR`
  - TopicExpressionType topicExpression: the `Topic` to subscribe to, valued with the `DLMapResourceQNames.RP_DLMAP` constant.

Return: none.

3.2.2.4 Implementation Details

3.2.2.4.1 Classes

All the following classes belong to the package

`org.diligentproject.keeperservice.dlmanagement.impl`

DLManagementService

This class implements the client part of DLManagement WS-Resource (see Figure 74). It exposes methods for getting, updating, and creating DLMap. In this first version of the service, this class is responsible of creating DLMap object by parsing an existing DLMap.xml file stored on the local disk. In order to parse the file, it uses methods from the package `org.diligentproject.keeperservice.mapmanager`

Figure 74. DLManagement: DLManagementService Class diagram

DLMapResource

The class is the statefull part of the DLManagement service. Each resource maintains a DLMap. It also implements the resource persistence by extending the abstract class PersistenceResource and by implementing the `load()` and `store()` methods.

Figure 75. DLManagement: DLMapResource class diagram

3.2.2.4.2 Stubs

DLManagementPortType

This class (Figure 76), generated automatically from wsdl definitions, exposes client side methods for service functionalities calls.

Figure 76. DLManagementPortType Class diagram

3.2.2.4.3 Dependencies

The service depends on:

- DIS-Registry stubs;
- DIS-HLSClient library;
- DIS-IP Library;
- DIS-Broker stubs;
- PackageRepository stubs;
- HNM stubs.

3.2.3 Hosting Node Manager

The HNM is the minimum piece of DILIGENT software that needs to be manually installed on each DILIGENT hosting node (DHN) in order to be able to deploy DILIGENT services on them. The Resource Manager (acting as a DHN owner) manually configures the node (i) by setting up the operating system and the Java WS Core toolkit, (ii) by deploying the HNM WSRF service, and (iii) by providing the required information about the node. Lastly, the DHN owner starts the container.

Once started, the DHN service reads the information provided by the Resource Owner and retrieves from the local system other information to complete the DHN profile. Such a profile is compliant with the XML version of the GLUE schema (see Appendix B). The DHN profile is published and updated using the DIS-Registry Service on a periodical basis.

The DHN profile reports static and dynamic information. The *static* part is related to a sub-set of the GLUE schema that is not affected by the activities performed on the DHN, e.g. the number of hard disks, the processor(s) type(s), etc. The *dynamic* part is related to another sub-set of the GLUE schema information whose values are influenced by the activities performed on the DHN, e.g. the free memory, the available space on the disk(s), etc.

At the start-up, the HNM queries the DL Management service to get the DL Map and un-registers the no longer available services according to the configuration.

After that, for each service profile file, the HNM generates the corresponding running instance profile and publishes it. If the running instance profile already exists, the HNM updates the running instance profile.

On registering the running instances profiles the HNM updates the DHN profile and sent the updated version to the DIS-Registry.

The Node Access Library (NAL) provides different functionalities to access the local node configuration. It allows to obtain the DHN unique ID and the RunningInstance unique ID, the service unique ID and the running instance profile of the caller. Moreover, the NAL, gives access to the DL Map and allows services to set the Specific Data section in their RunningInstance profile.

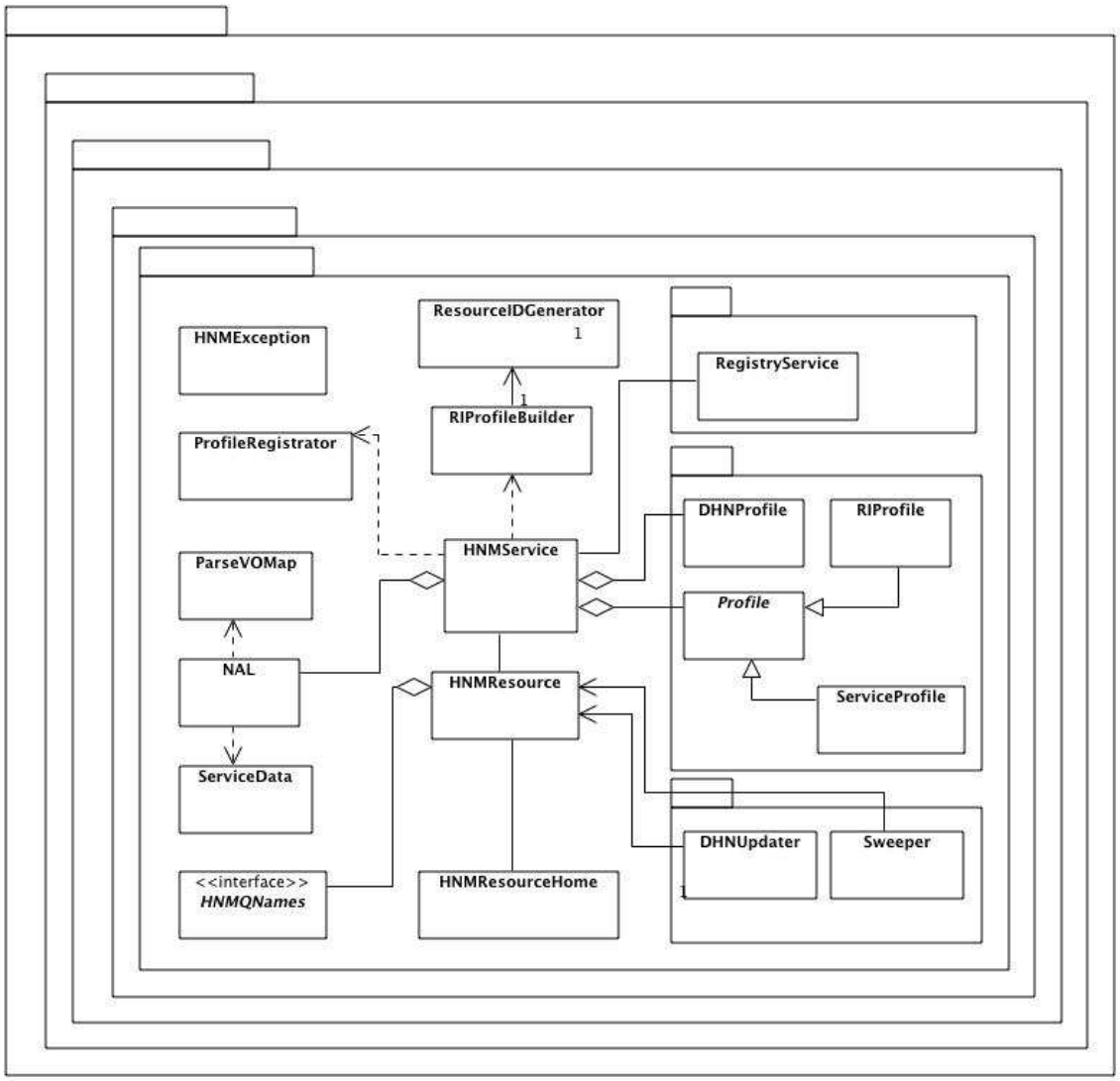

Figure 77. HNM: global class diagram

3.2.3.1 Profile

The HNM service profile is reported in Section A.10.

3.2.3.2 Managed Resources

The HNM resource is composed by a Singleton resource (HNMResource class) maintaining information about the status of the DHN. The DHN profile registered by the resource is presented Section A.11.

When the Resource Owner makes a new node available to DILIGENT infrastructure and starts the Java WS Core container, the HNM WSRF service is created.

3.2.3.3 Operations

The functionalities exposed by the NAL library, are the ones concerning mainly access to the node information. Services can access to this library using the following operations:

getVOMap

```
java.lang.String getVOMap ( java.lang.String voname )
```

Returns the VO map given a VO name.

Parameters:

- voname - a String with the VO name.

Returns: if the required VO map exists, it returns a String with the VO map; else it returns null.

getRIID

```
java.lang.String getRIID ( java.lang.String serviceName,
java.lang.String serviceClass )
```

Returns the Running Instance (RI) unique ID of a given service name and service class.

Parameters:

- serviceName - a String with a valid local service name.
- serviceClass - a String with a valid local service class.

Returns: If the required RI profile exists, return its unique ID else it returns null.

getRIID

```
java.lang.String getRIID ( java.lang.String riProfile )
```

Returns the unique RI ID of a given RI profile.

Parameters:

- riProfile - a String with a valid local RI profile.

Returns: if the required RI profile exists, it returns its unique ID else it returns null.

getServiceID

```
java.lang.String getServiceID ( java.lang.String serviceName,
java.lang.String serviceClass)
```

Returns the Service unique ID of a given service name and service class.

Parameters:

- serviceName - a String with a valid local service name.
- serviceClass - a String with a valid local service class.

Returns: if the required service profile exists, it returns its unique ID else it returns null.

getDHNID

```
java.lang.String getDHNID ( )
```

Returns the local DHN unique ID.

Returns: if the required local DHN is already registered returns a String with the DHN unique ID else it returns null.

getRIProfile

```
org.w3c.dom.Document getRIProfile ( java.lang.String serviceName,
java.lang.String serviceClass)
```

Returns the RI profile of a given service name and service class.

Parameters:

- serviceName - a String with a valid local RI service name.
- serviceClass - a String with a valid local RI service class.

Returns: if the required RI profile exists, it returns it else it returns null.

setSpecificData

```
boolean setSpecificData ( java.lang.String serviceName,
java.lang.String serviceClass, java.lang.String data )
```

Set the specific data tag of the RI profile associated with the given service name and service class.

Parameters:

- serviceName - a String with a valid local service name.
- serviceClass - a String with a valid local service class.
- data - a String containing the specific data section to set

Returns: if the required RI profile exists, it sets the specific data section and returns true else it returns false.

getAllServices

```
java.util.Vector getAllServices()
```

Return a list of all services deployed (id, service class, and service name).

Parameters: none.

Returns: a list of ServiceData class objects.

3.2.3.4 How to use the component

The HNMService does not expose directly any public operation. Local running instances can interact with the Node Access Library in order to access to the DHN configuration and to their own Service and RI profiles.

To use the NAL, each service should create a NAL object and appropriately use described methods.

3.2.3.5 Implementation Details

3.2.3.5.1 Classes

All following classes belong to the package

`org.diligentprojects.keeperservice.hnm.impl.`

HNMService

This is the service class of the HNM service.

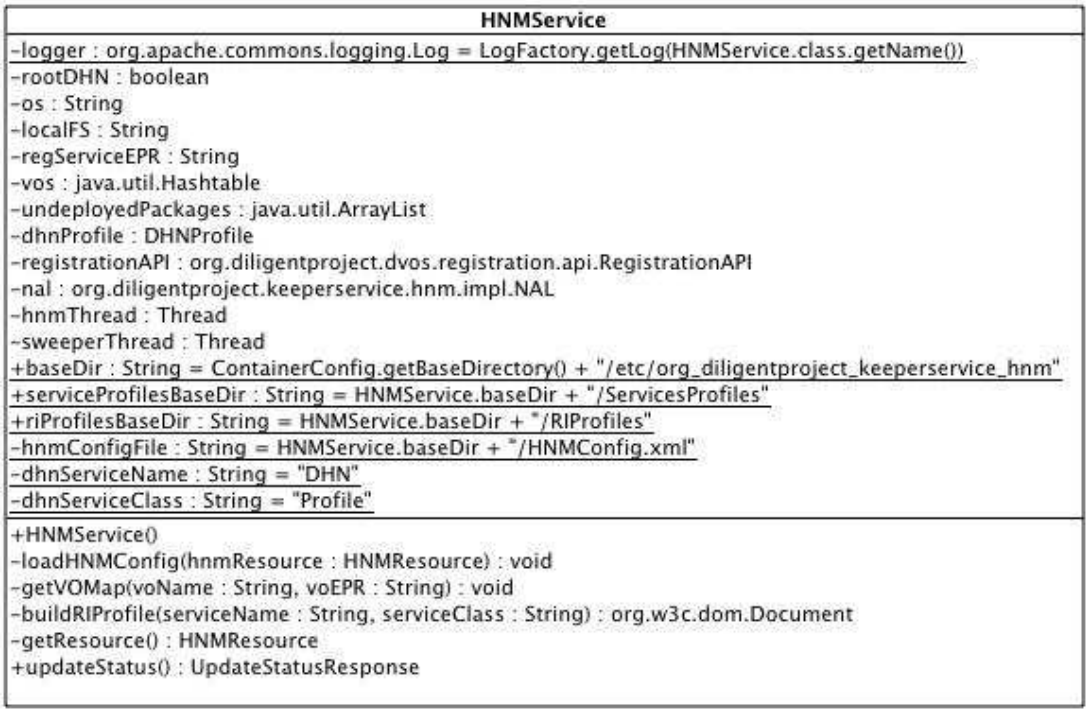

Figure 78. HNM: HNMService class diagram

HNMResource

This class (see Figure 79) together with the HNMQNames and HNMResourceHome compose the resource implementation. In particular, this class handles the DHN profile and status information.

Figure 79. HNM: HNMResource class diagram

HNMResourceHome

This class (see Figure 80) creates and makes accessible the HNM Singleton resource.

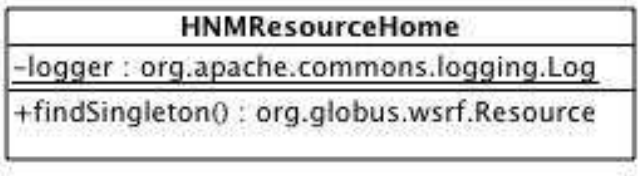

Figure 80. HNM: HNMResourceHome class diagram

ResurceIDGenerator

This class (see Figure 81. HNM: ResourceIDGenerator class diagram) generates unique IDs to assign at profile registration to the newly deployed RunnningInstances.

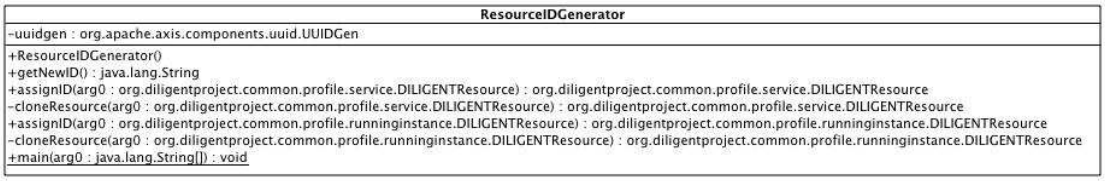

Figure 81. HNM: ResourceIDGenerator class diagram

RIProfileBuilder

This class (see Figure 82. HNM: RIProfileBuilder class diagram) builds a RunningInstance profile starting from a Service profile and a list of deployed packages of that service.

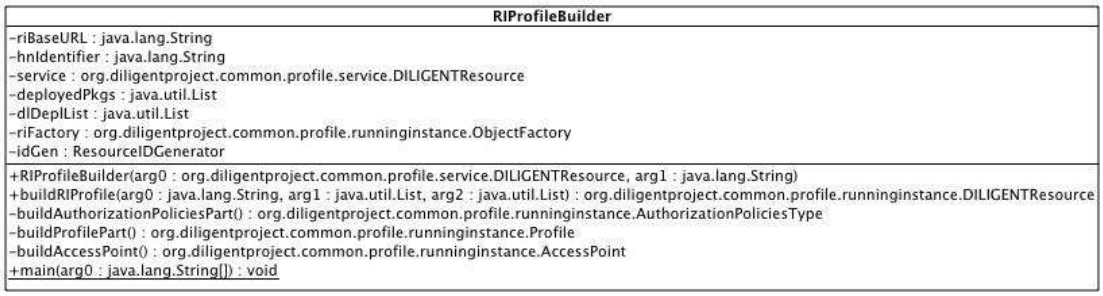

Figure 82. HNM: RIProfileBuilder class diagram

NAL

The following classes (see *Figure 83. HNM: NAL class* diagram and *Figure 84. HNM: ServiceData class* diagram) constitutes the NAL library implementation; they are in charge to give access to the node configuration.

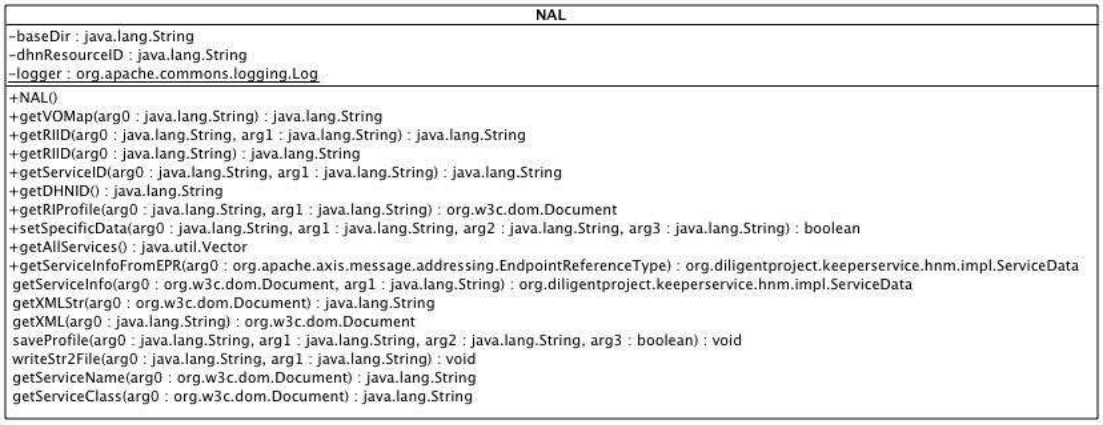

Figure 83. HNM: NAL class diagram

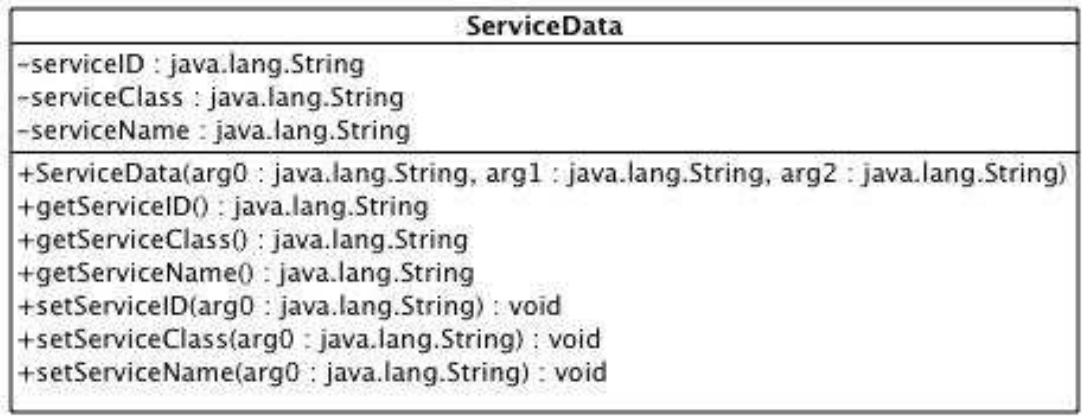

Figure 84. HNM: ServiceData class diagram

3.2.3.5.2 Dependencies

The HNM and NAL uses:

- JavaBeans Act.Fw.1.1 (activation.jar)
- Apache Common IO (commons-io-1.2.jar)
- Diligent common:
  - org.diligentproject.common.profile.dhn.jar,
  - org.diligentproject.common.profile.runninginstance.jar,
  - org.diligentproject.common.profile.service.jar)
- DIS-HLSClient (org_diligentproject_informationservice_dishlsclient.jar)
- DIS-IC (org_diligentproject_informationservice_disic_stubs.jar)
- DIS-IP (org_diligentproject_informationservice_disip.jar)
- DIS-util (org_diligentproject_informationservice_util.jar)
- HNM config (rg.diligentproject.keeper.hnm.hnmconfig.jar)

- JAXB (jaxb-api.jar, jaxb-impl.jar, jaxb-xjc.jar, jaxb1-impl.jar, jsr173_api.jar)
- DLManagement (org_diligentproject_keeperservice_dlmanagement_stubs.jar)
- DIS-Registry (org_diligentproject_informationservice_disregistry_stubs.jar)

3.2.3.6 Known Bugs and Limitations

The Alpha version of the Hosting Node Manager does not support the dynamic deployment.

4 DYNAMIC VO SUPPORT SERVICE

The Dynamic VO Support (DVOS) services are in charge to supply other DILIGENT services with a robust and flexible security framework as well as to provide services in order to manage VO concepts introduced in Section 3.7 of the test-bed functional specification. The Delegationa and CredentialsRenewal services provide authentication support for DILIGENT Users and Services. The Authorization service is in charge to manage permissions to perform action within the infrastructure.

Moreover, UserGroupManagement service is in charge to provide functionalities related to Users and Group Management. All following paragraphs assume than these components are installed on each DHN:

- Authentication API: A library containing authentication utilities
- DVOS Common: A package containing common classes used in WSDL interfaces of DVOS services
- Delegation Stubs: Stub library of the Delegation service
- Delegation Service: Delegation service implementation
- Authorization Stubs: Stub library of the Authorization service
- Authorization API: A library used to contact the Authorization service and enforce authorization

Libraries and services described in this paragraph are needed to enable authentication and authorization functionalities in the DILIGENT platform. The paragraph starts with a brief description of interaction between DVOS and gLite security.

4.1 DVOS and gLite security integration

This paragraph describes how DVOS and gLite security interoperates to provide DILIGENT with required authenticationa and authorization functionities. gLite security, as well as the DILIGENT one, is based on PKI and X509 certificates. This eases the integration between DVOS and gLite authentication. From the VO management point of view, the DVOS integrates the concept of WS-Resource within the VOMS VO model. The diagram in Figure 85 shows DVOS and VOMS interactions in the DILIGENT infrastructure.

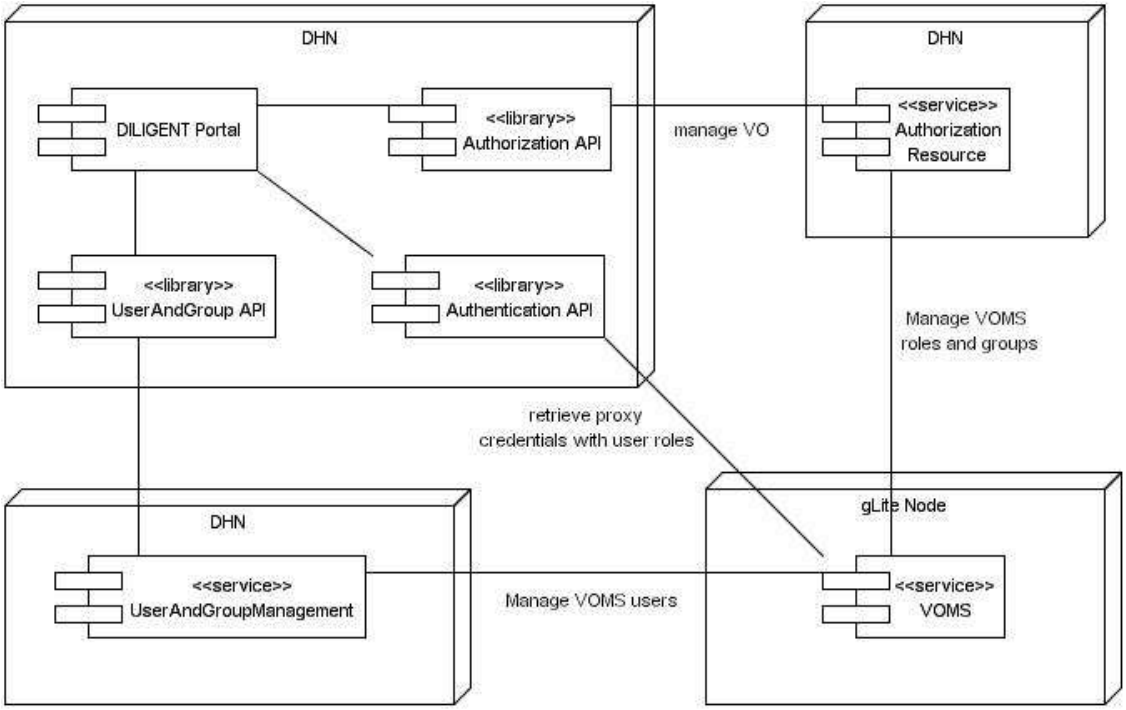

Figure 85, DVOS – VOMS interaction

DILIGENT authorization services and VOMS interact to implement the DILIGENT VO model described in [6]. Each VOMS group is associated to a different VO in DILIGENT.

The VOMS server is in charge to maintain the association between users and VOs in DILIGENT they belong to. Ti is also in charge to maintain the association between users and roles granted to them.

DVOS Authorization service is in charge to maintain associations between VO roles and the set of actions they are entitled to perform.

When a new session is open on the portal, or when the user requests it, new proxy credentials are loaded for the user. These credentials are then enriched with the set of roles held by the user.

When a new VO is created in DILIGENT, a new group is also added to the VOMS group hierarchy. When a new role is created in a VO in DILIGENT, a corresponding role with the same name is also created in the VOMS VO. When a new user is added to DILIGENT, the same user is also registered in the VOMS VO.

VO Managers are in charge to manage associations between user, roles and permission in the DILIGENT VO hierarchy.

4.2 DVOS Common

This component includes some XSD data types and exceptions shared by the DVOS service interfaces.

Figure 86. DVOS Common: global class diagram

4.2.1 Profile

The profile of the DVOS Common library is reported in Appendix A.12 .

4.2.2 Implementation Details

All classes of this library are beans automatically generated by the Axis WSDL2Java tool. Thus, beyond setter and getter methods, they contain serialization and de-serialization methods only. For this reason the classes are presented, but not described in detail.

4.2.3 Dependencies

None

4.2.3.1 Configuration

To correctly deploy the library in the container the `DVOSDataTypes.xsd` file must also be copied in the `$GLOBUS_LOCATION/share/schema` directory.

For this reason, the component is provided in a gar archive. This allows users to deploy it as a normal package in the Java WS Core container.

4.2.3.2 Know Bugs And Limitations

None

4.3 Authentication Management

The purpose of the authentication management is twofold. It provides guidelines about how to use Java WS Core authentication functionalities in DILIGENT, and describes DVOS Authentication Support services.

A diagram of Authentication management is displayed in the next figure.

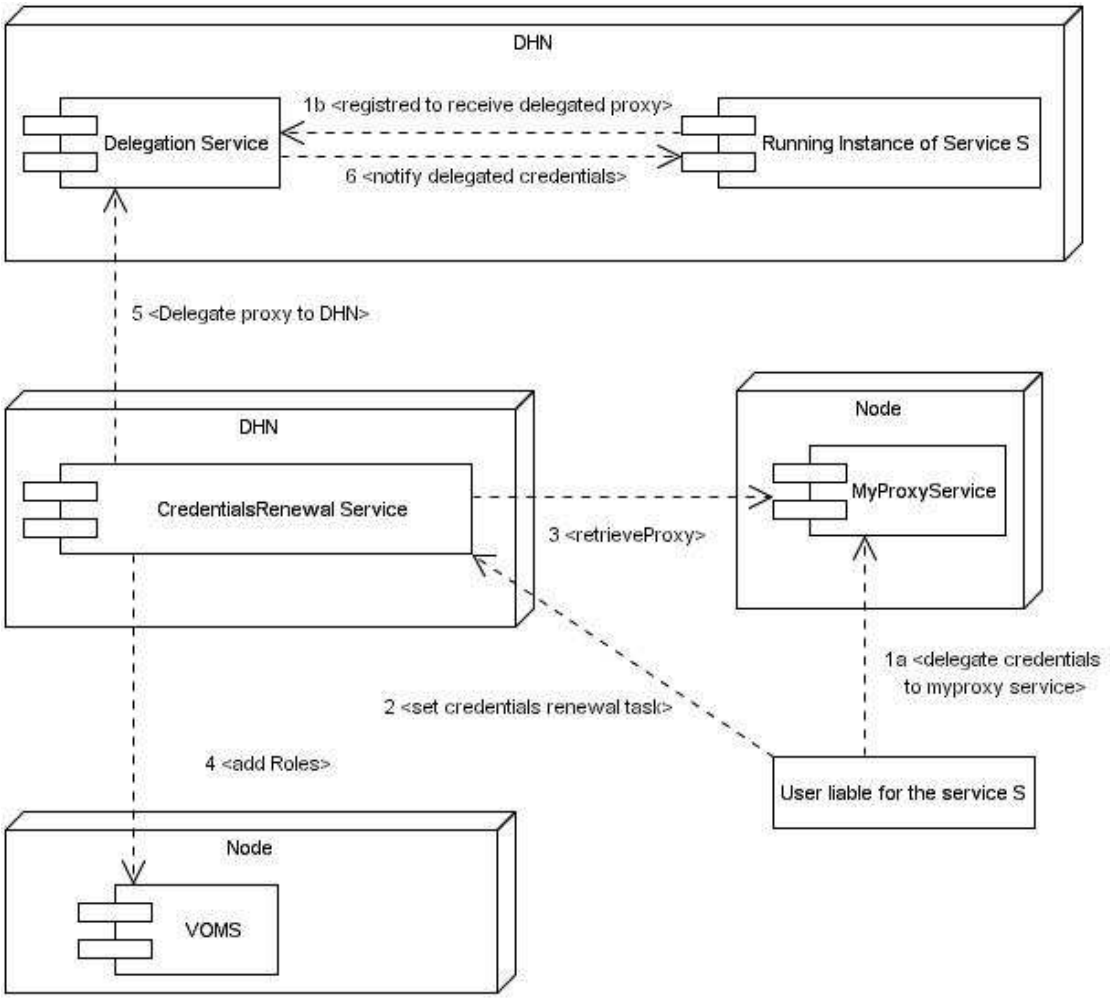

Figure 87. Authentication Management: deployemnt diagram

The User liable for a given service S must delegate a copy of its credentials to the myproxy service. This can be done using the *MyProxy-API library*. Moreover, the user, or the keeper deploying the RI (not shown in the diagram), must instruct the *Credentials Renewal Service*

to periodically delegate these credentials. This can be done using the *Credentials Renewal API library*.

Once set, the Credentials Renewal Service periodically retrieves proxy credentials from the myproxy and, through the *Authentication API library,* add roles needed to operate in the infrastructure, if any. Then, using the *Delegation stubs*, send delegated credentials to the DHN where the RI of the service S is running. The *Delegation Service* receives delegated credentials and dispatches it to the (registered) RI of service S.

This way the RI of service S has periodically renewed proxy credentials to operate in the infrastructure.

4.3.1 DILIGENT Security Configuration

Security configuration for DILIGENT running instances involves two main aspects: Authentication and Authorization.

From the Authentication point of view, the following main issues must be faced:

- How clients authenticate DILIGENT running instances. (RI authentication)
- How DILIGENT running instances can act within the DILIGENT infrastructure. (Client authentication)

From the authorization point of view, the following main issues must be faced:

- How to describe authorization for DILIGENT Running Instances. (Logical operations)
- How to perform verification of authorization when a request is received. (Authorization checks)

Answers to these issues lead to security configuration for DHN and DILIGENT Services described in following paragraphs.

4.3.2 DHN Security Configuration

Each DHN must be manually configured (only once) following these guidelines:

4.3.2.1 Install container credentials

Container credentials are typically installed in the `/etc/grid-security/` directory. The certificate (file `containercert.pem`) should be owned by the user running the container and be read-only for other users. The key (file `containerkey.pem`) should be owned and read-only by the user running the container.

The WS-Core container must be configured as follows in order to use container credentials.

Edit the file `etc/globus_wsrf_core/server-config.wsdd` to add to the `<raglobalConfiguartion>` tag the following parameter: `<parameter name="containerSecDesc" value="etc/globus_wsrf_core/global_security_descriptor.xml"/>`

Edit the `etc/globus_wsrf_core/global_security_descriptor.xml` file to set correct location for container credentials (typically `/etc/grid-security/containercert.pem` and `/etc/grid-security/containerkey.pem`). In addition, comment the `<gridmap>` tag.

4.3.2.2 Install CA certificates

Public certificates for trusted CA must be installed simply copying it in the directory `/etc/grid-security/certificates` of the node hosting the DILIGENT container.

Please refers to the BSCW folder: "`DILIGENT/DILIGENT Infrastructure/CA certificate repository`" to find CA certificates to install.

4.3.2.3 Install VOMS certificates

Public certificates of trusted VOMS must be installed simply copying it in the directory `/etc/grid-security/certificates` of the node hosting the DILIGENT container.

Please refers to the BSCW folder: "`DILIGENT/DILIGENT Infrastructure/CA certificates repository/VOMS server host certificates`" to find VOMS certificates to install.

4.3.2.4 Disable HTTPS protocol

The container must be started with the option "`-nosec`" to disable HTTPS protocol (GSITransport).

Please notice that Message level security (GSISecureMessage and GSISecureConversation) is still available for DILIGENT services running in DHN containers.

4.3.3 Service Security Configuration

Each DILIGENT service must be manually configured (before its registration in the DILIGENT infrastructure) following these guidelines:

4.3.3.1 Web Service Security Descriptor

Set a Web Service Security Descriptor (WSSD) for each WSRF interface. In each WSSD, use the GSISecureConversation as authentication mechanism for all service methods. For those methods running with client's identity, the `<caller-identity>` must be chosen among `RunAs` modes. Other methods must choose the `<service-identity>` mode.

Do not set service credentials in the Web Service Security Descriptor file.

Credentials for DILIGENT services can be obtained registering a CredentialsListener to the DelegationLocalInterface of the local DVOS Delegation service (see paragraph 4.4.2 for details).

4.3.3.2 Service credentials delegation

Credentials can be delegated to each service using the CredentialsRenewalService (see paragraph 4.4.3 for details). The service access path in the container is used as the default credentials identifier. Each service interested in using delegated credentials can simply register a CredentialsListener to its access path and be notified when renewed credentials are delegated to the container (see paragraph 4.4.2).

4.3.4 Authorization configuration

In order to configure authorization for DILIGENT service these guidelines must be followed.

4.3.4.1 Logical operations

Logical operations used by DILIGENT services must be defined in DILIGENT and granted to the DILIGENT VO using the OperationAdministration interface of the Authorization service (see paragraph 4.5.1).

Logical operations IDs could be defined as strings on free format. Nevertheless, a standardization of these IDs could greatly simplify understanding of operation meaning and reduce the risk of naming collision.

For these reasons, Logical Operations IDs should be defined as Qualified Names in the namespace of a DILIGENT service. As Logical Operations can span different services, new namespaces can be defined for Logical Operations.

An example of definition of a Logical Operation could be:

```
http://www.diligentproject.org/namespaces/dvos/registration}registerResour
ces
```

4.3.4.2 Configure Authorization Handler

For each Web Service Security Descriptor a VOAuthorizationPDP handler must be added to enforce VO authorization. This can be done adding the tag:

```
<authz
value="VOAuthorizationPDP:org.diligentproject.dvos.authorization.handler.V
OAuthorizationPDP"/>
```

For detailed information about how to configure the `VOAuthorizationPDP` handler, please refer to Section 4.5.2.2.

Despite credentials used by each service, this handler always uses container credentials to contact the DVOS Authorization service and verify authorization.

4.4 Components

4.4.1 Authentication API

This library provides DILIGENT services with some utility method useful to manage credentials and to set security properties on service stubs.

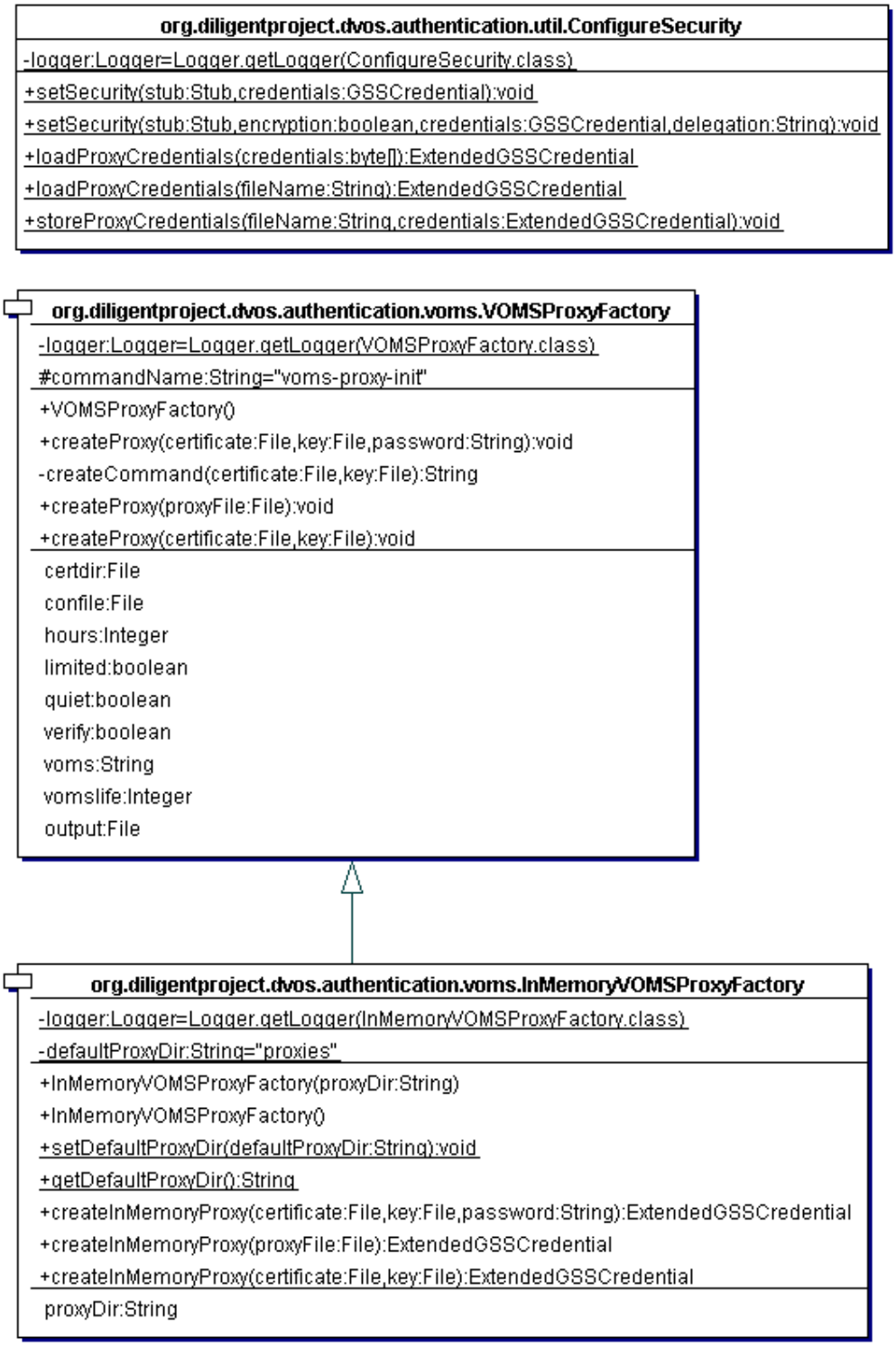

Figure 88. Authorization API: global class diagram

4.4.1.1 Profile

The profile of the Authentication service is reported in Appendix A.13.

4.4.1.2 Implementation Details

All the following classes belong to the package
`org.diligentproject.dvos.authentication.`

util.ConfigureSecurity

setSecurity

`void setSecurity(Stub stub, GSSCredential credentials)`

This method allows to set security properties on service stubs. Encryption is used as default protection level. No delegation is performed.

Parameters:

- stub: The javax.xml.rpc.Stub object where to set security
- credentials: The object containing credentials to be set in stub

setSecurity

`void setSecurity(Stub stub, boolean encryption, GSSCredential credentials, String delegation)`

This method allows to set security properties on service stubs.

Parameters:

- stub: The javax.xml.rpc.Stub object where to set security
- credentials: The object containing credentials to be set in stub
- encryption: If true encryption of messages will be asked to the service, if false signature only is used.
- delegation: The value of delegation required. Allowed values are:
    - org.globus.axis.gsi.GSIConstants.GSI_MODE_NO_DELEG:No delegation is performed.
    - org.globus.axis.gsi.GSIConstants.GSI_MODE_LIMITED_DELEG:Limited delegation is performed (credentials cannot be further delegated).
    - org.globus.axis.gsi.GSIConstants.GSI_MODE_FULL_DELEG:Full delegation is performed (credentials can be further delegated).

loadProxyCredentials

`ExtendedGSSCredential loadProxyCredentials(byte[] credentials)`

Utility method to parse proxy credentials from a byte array.

Parameters:

- credentials: the byte array containing proxy credentials to load

Return: an ExtendedGSSCredential object containing parsed credentials

loadProxyCredentials

`ExtendedGSSCredential loadProxyCredentials(String fileName)`

Utility method to load proxy credentials from a file.

Parameters:

- fileName: the absolute or relative path of file containing proxy credentials to load

Return: an ExtendedGSSCredential object containing parsed credentials

storeProxyCredentials

`void storeProxyCredentials(String fileName, ExtendedGSSCredential credentials)`

Utility method to store proxy credentials to a file.

Parameters:

- fileName: the absolute or relative path of file where to store proxy credentials
- credentials: credentials to be stored on file

voms.VOMSProxyFactory

This class generates proxy certificates containing VOMS Attribute Certificates. It wraps the `voms-proxy-init` command that must be installed locally.

createProxy

`void createProxy(File certificate, File key, String password)`

Create a new proxy certificate from an EEC and an encrypted private key.

Parameters:

- certificate: the file containing the End Entity Certificate.
- key: the file containing the private key.
- password: the password to decrypt the private key

createProxy

`void createProxy(File proxyFile)`

Create a new proxy certificate from another proxy certificate.

Parameters:

- proxyFile: the file containing the Proxy Certificate.

createProxy

`void createProxy(File certificate, File key)`

Create a new proxy certificate from an EEC and an unencrypted private key.

Parameters:

- certificate: the file containing the End Entity Certificate.
- key: the file containing the unencrypted private key.

voms.InMemoryVOMSProxyFactory

This class generates proxy certificates containing VOMS Attribute Certificates and loads them in memory. The file used to generate the certificate is deleted after the creation. By default, the directory used to temporary store proxy credentials is the `proxies`

subdirectory of the current directory where the JVM is running. This class wraps the `voms-proxy-init` command that must be installed locally.

createInMemoryProxy

`ExtendedGSSCredential createInMemoryProxy(File certificate, File key, String password)`

Create a new proxy certificate from an EEC and an encrypted private key. The new proxy is written to a file only at generation time. Then the proxy is loaded in memory and the file is deleted. The file will be created in the current `proxyDir` directory.

Please note that this does not prevent the new proxy to be red from other processes during its persistence on disk. This method should only be intended as an easy way to keep the file system clean from proxies.

Parameters:

- certificate: the file containing the End Entity Certificate.
- key: the file containing the private key.
- password: the password to decrypt the private key

Return: The new credentials just created.

createInMemoryProxy

`ExtendedGSSCredential createInMemoryProxy(File proxyFile)`

Create a new proxy certificate from another proxy certificate. The new proxy is written to a file only at generation time. Then the proxy is loaded in memory and the file is deleted. The file will be created in the `proxyDir` directory. Please note that this does not prevent the new proxy to be red from other processes during its persistence on disk. This method should only be intended as an easy way to keep the file system clean from proxies.

Parameters:

- ProxyFile: the file containing the Proxy Certificate.

Return: The new credentials just created.

createInMemoryProxy

`ExtendedGSSCredential createInMemoryProxy(File certificate, File key)`

Create a new proxy certificate from an EEC and an unencrypted private key. The new proxy is written to a file only at generation time. Then the proxy is loaded in memory and the file is deleted. The file will be created in the current `proxyDir` directory. Please note that this does not prevent the new proxy to be red from other processes during its persistence on disk. This method should only be intended as an easy way to keep the file system clean from proxies.

Parameters:

- certificate: the file containing the End Entity Certificate.
- key: the file containing the unencrypted private key.

Return: The new credentials just created.

4.4.1.3 Dependencies

This library requires VOMS API libraries to contact VOMS daemon and to generate attributed proxy credentials.

4.4.1.4 Known bugs and problems

None

4.4.2 Delegation Service

This service allows clients to delegate proxy credentials to DILIGENT services running on a DHN. Delegated credentials can then be used by co-hosted services (e.g.: to perform service invocations).

Each delegated credential is associated to a credentials identifier unique inside the container. Co-hosted services can subscribe to the local Delegation service using these identifiers to receive notifications about delegated credentials. This can be done using static methods of the `DelegationLocalInterface` class. All services registered to the credentials identifier are notified any time fresh credentials for that identifier are delegated (or canceled) to the DHN.

The Delegation Service class diagram is reported in *Figure 89*.

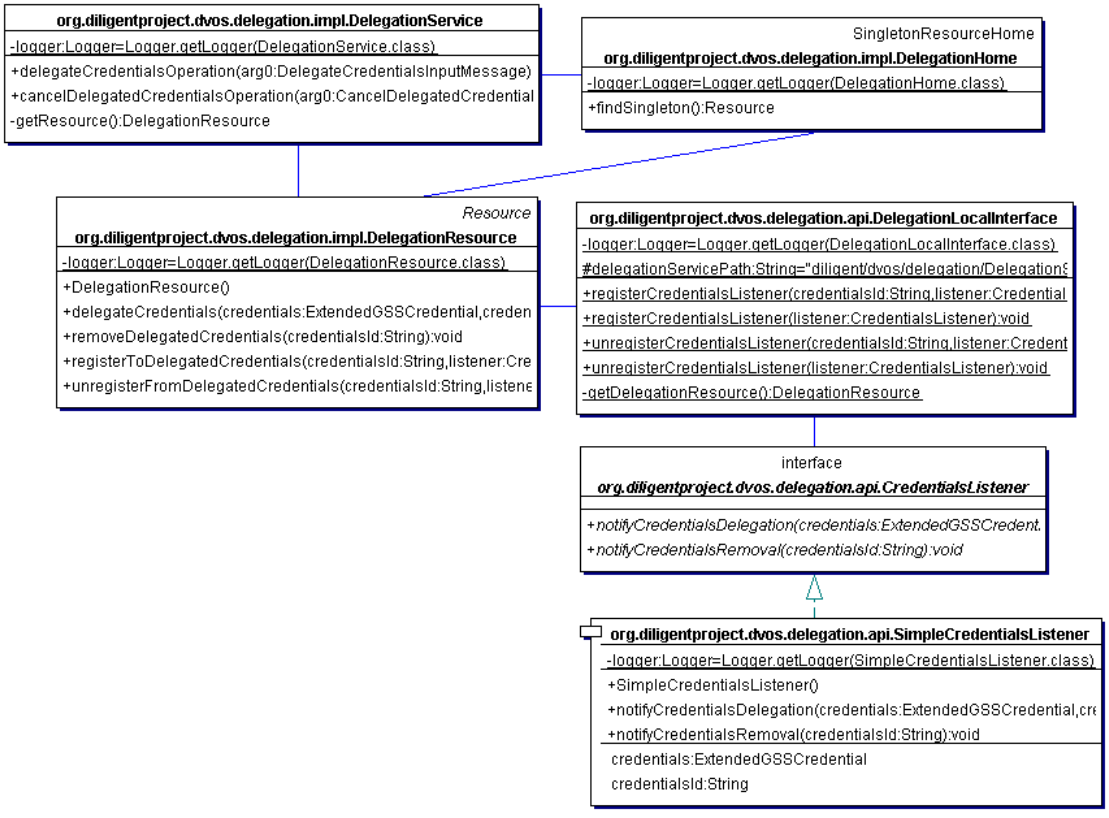

Figure 89. Delegation service: global class diagram

4.4.2.1 Profile

The profile of the Delegation service is reported in Appendix A.14.

4.4.2.2 Managed Resources

The Delegation service is a singleton stateful Web Service bound to a DelegationResource resource. The DelegationResource manage delegated credentials and credentials listeners subscribed by co-hosted services.

For security reasons the information managed by the DelegationResource is not persistent, and is never stored on local file system.

4.4.2.3 Operations

DelegateCredentialsOperation

```
DelegateCredentialsOutputMessage
DelegateCredentialsOperation(DelegateCredentialsInputMessage message)
```

It allows delegating proxy credentials.

Parameters:

- message: a complex type specified by Delegation.wsdl containing delegated credentials and the credentials identifier as strings.

Return: An empty complex type.

cancelDelegatedCredentialsOperation

```
CancelDelegatedCredentialsOutputMessage
cancelDelegatedCredentialsOperation(CancelDelegatedCredentialsInputMessage
message)
```

It allows cancelling delegated proxy credentials.

Parameters:

- message: a complex type specified by Delegation.wsdl containing the identifier of the credentials to be cancelled.

Return: An empty complex type.

4.4.2.4 Implementation Details

All these classes are part of the package

`org.diligentproject.dvos.delegation.*`

impl.DelegationService

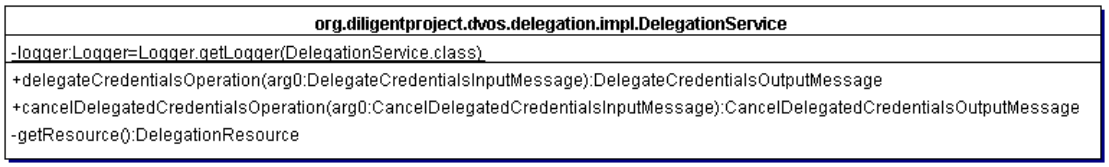

Figure 90. *Delegation Service: DelegationService class diagram*

This class implements operations of the Delegation Service WSDL interface.

impl.DelegationResource

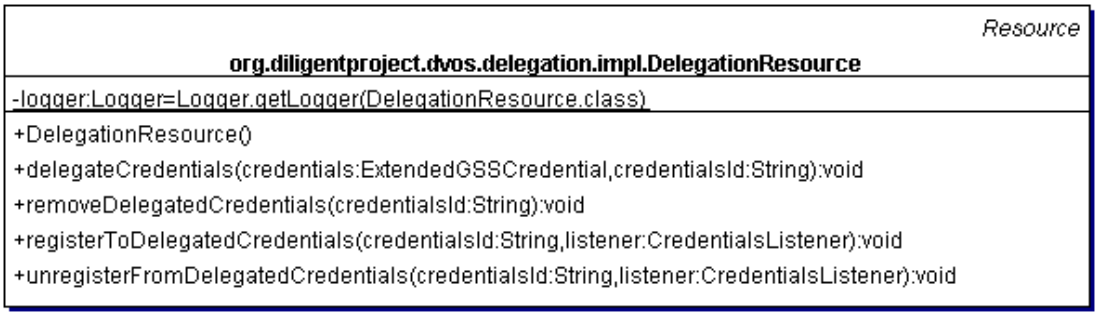

Figure 91. *Delegation Service: DelegationResource class diagram*

This class implements the Delegation resource in charge to store delegated credentials and credentials listeners. It is also in charge to notify listeners when new credentials are delegated to the container.

impl.DelegationHome

Figure 92. *Delegation Service: DelegationHome class diagram*

This class implements the Delegation Home.

api.CredentialsListener

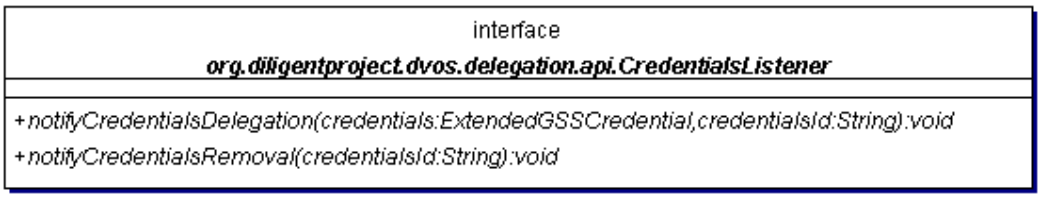

Figure 93. *Delegation Service: CredentialsListener class diagram*

Credentials listeners must implement this interface in order to receive notification about delegated credentials.

api.SimpleCredentialsListener

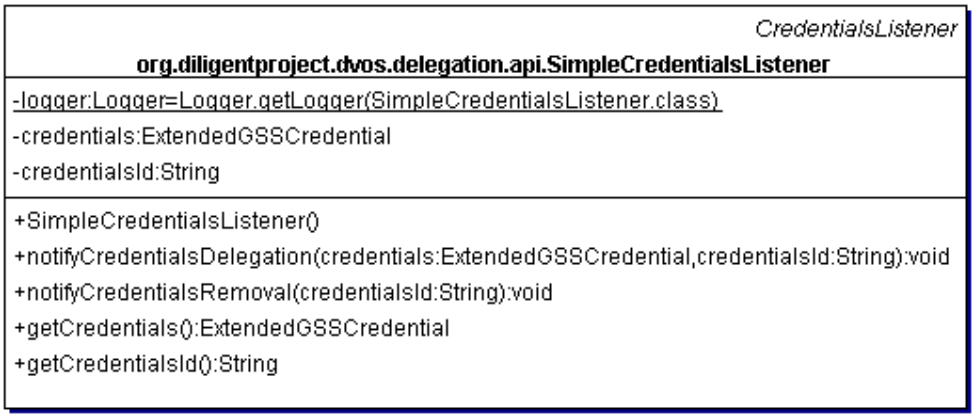

Figure 94. *Delegation Service: SimpleCredentialsListener class diagram*

This class implements the CredentialsListener interface and store delegated credentials in memory. The last copy of credentials and the credentialsId can be retrieved using get methods.

impl.DelegationLocalInterface

Figure 95. *Delegation Service: DelegationLocalInterface class diagram*

This class provides local access to delegation service allowing co-hosted services to register and unregister credentials listeners.

Listeners are bound to a credentialsId. When delegated credentials are received or cancelled for an Id, all listeners registered for that Id are notified.

4.4.2.4.1 Dependencies

No dependencies

4.4.2.4.2 Security Profile

Following logical operation is defined for the Delegation service:

```
{http://www.diligentproject.org/namespaces/dvos/delegation}manageDelegated
Credentials
```

Allows to delegate and remove credentials to a DHN.

4.4.2.4.3 Known bugs and limitations

None

4.4.3 CredentialsRenewal Service

This service allows users to periodically delegate their credentials to DHN. Credentials must be already stored in a MyProxy repository. The service is able to retrieve a copy of credentials from the repository and forward it to the DHN indicated by the user. It can also contact VOMS server(s) adding roles needed by the services to operate.

Delegated credentials are sent to the Delegation service running on the remote DHN. Subscribed services running on that node are thus notified with delegated credentials (see paragraph 4.4.2).

In order to be able to operate properly, the credentials Renewal service requires valid credentials to be present on the local file system. These credentials must be protected by password. The password to decrypt these credentials is red at container startup from standard input.

The Credentials Renewal Service class diagram is reported in *Figure 89*.

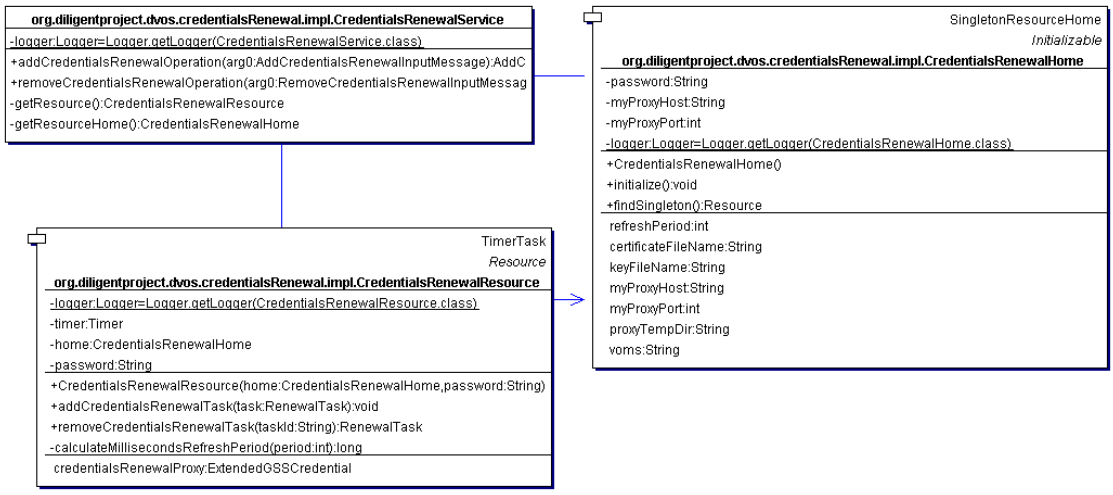

Figure 96. Credentials Renewal: global class diagram

4.4.3.1 Profile

The profile of the Credentials Renewal service is reported in Appendix A.15.

4.4.3.2 Managed Resources

The Credentials Renewal service is a singleton stateful Web Service bound to a CredentialsRenewalResource resource. The CredentialsRenewalResource manages credentials renewal tasks.

4.4.3.2.1 Configuration

Internal configuration

Once the service has been deployed, the deploy-jndi-config.xml (contained in the `/etc` subdirectory) must be configured in order to properly initialize the service. All parameters reported in the following table must be properly set:

Field	Description
`refreshPeriod`	The refresh period of proxy credentials used by the service itself
`voms`	The request string to be included in the voms request for the proxy credentials used by the service

`certificateFileName`	The file where to find the certificate used by the service
`keyFileName`	The file where to read the private key used by the service
`myProxyHost`	The name of the host where the MyProxy service runs
`myProxyPort`	The port of the host where the MyProxy service runs
`proxyTempDir`	The local directory to use for proxy files. Proxies are stored in this directory only for the time needed to delegate them to DHNs, then they are removed.

Table 1. deploy-jndi-config configuration parameters

Moreover, the following command must be executed before to start-up the DHN hosting the CredentialsRenewal service:

umask u=rw,g=,o=

This command set the user mask, defining authorization rules for files created on local file system. This is needed because the voms-proxy-init command requires strict access rights to be set on proxy credentials used to contact a VOMS service. All files created on the local file system will have permissions to read and write for the user and no permissions for the group and others (code 400 in the octal notation of the chmod command). The JVM does not allow doing this programmatically.

At container startup, the user is asked to type on standard input the password required to decrypt credentials of the CredentialsRenewal service.

4.4.3.3 Operations

addCredentialsRenewalOperation

```
AddCredentialsRenewalOutputMessage
addCredentialsRenewalOperation(AddCredentialsRenewalInputMessage message)
```

Add a new credentials renewal task.

Parameters:

- message: a complex type specified by CredentialsRenewal.wsdl containing following fields:
  - myProxyUsername: The MyProxy username to contact MyProxy
  - myProxyPassword: the MyProxy password to use
  - credentialsId: The credentialsId identifying credentials on the delegated DHN (see paraghraph 4.4.2)
  - delegationServiceURL: The URL of the Delegation Service which credentials must be delegated.
  - VOMSRole: the set of requests to be included in the voms-proxy-init command to create a voms proxy.
  - Period: the validity of delegated credentials (in hours). Credentials will be renewed one hour before this limit. Please do not specify a zero refresh period.

Return: An empty complex type.

removeCredentialsRenewalOperation

`RemoveCredentialsRenewalOutputMessage removeCredentialsRenewalOperation(RemoveCredentialsRenewalInputMessage message)`

Remove an existing credentials renewal task. When a credentials task is removed, a cancel request is sent to the DHN where credentials were delegated.

Parameters:

- message: a complex type specified by CredentialsRenewal.wsdl containing following fields:
    - credentialsId: The credentialsId identifying credentials renewal task to cancel (see paraghraph 4.4.2)
    - delegationServiceURL: The URL of Delegation Service identifying the DHN where credentials was delegated (see paraghraph 4.4.2)

Return: An empty complex type.

4.4.3.4 Implementation details

All these classes are part of the package

`org.diligentproject.dvos.credentialsRenewal`

impl.CredentialsRenewalService

Figure 97. Credentials Renewal: CredentialsRenewalService class diagram

This class implements the Credentials Renewal Service interface components and exposes service operations.

impl.CredentialsRenewalResource

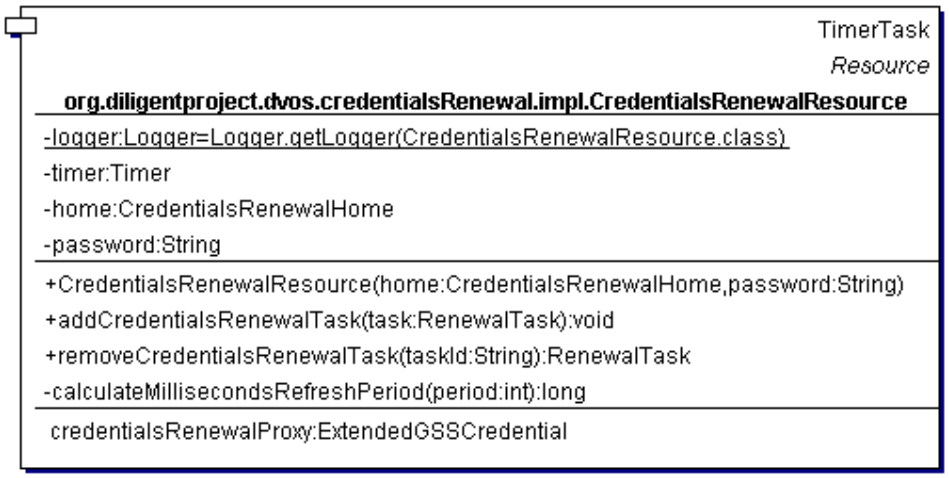

Figure 98*. Credentials Renewal: CredentialsRenewalResource class diagram*

This class implements the Credentials Renewal resource in charge to manage credentials renewal tasks.

impl.CredentialsRenewalHome

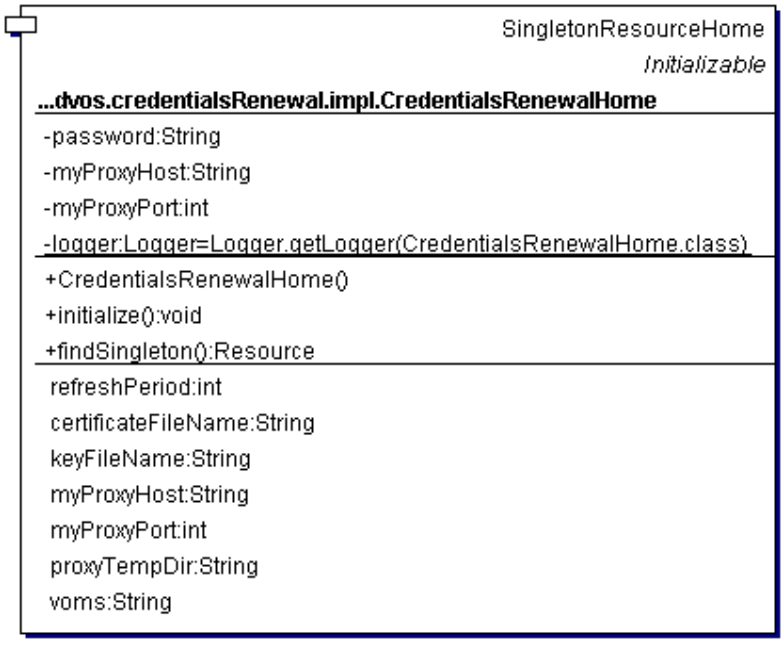

Figure 99*. Credentials Renewal: CredentialsRenewalHome class diagram*

This class implements the Credentials Renewal Home.

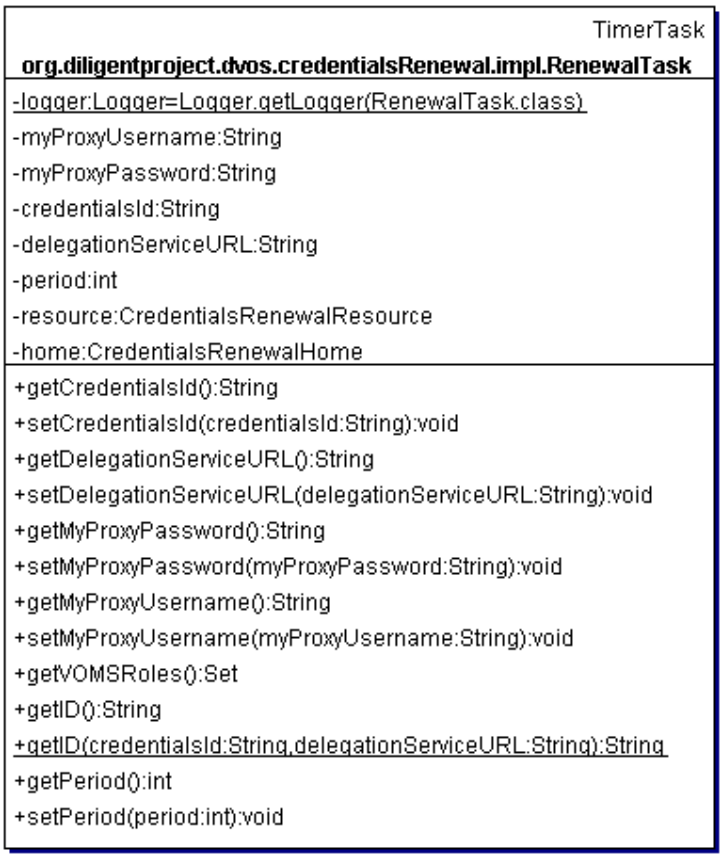

Figure 100*. Credentials Renewal: CredentialsListener class diagram*

This class represents a single credentials renewal task periodically executed to renew credentials.

4.4.3.4.1 Dependencies

This component depends on the voms-proxy-init command that must be installed locally.

It also depends on the Delegation Service stubs.

4.4.3.4.2 Security Profile

The following logical operation is defined for the CredentialsRenewal service:

`org.diligentproject.security.authorizationOperation.dvos.manageCredentialsRenewal`

It allows to delegate and remove credentials to a DHN.

4.4.3.4.3 Known bugs and problems

The logical operation defined for this service does not follows the standard described in paragraph 4.3.4.1. Logical operations should be defined using the QName notation:

- {namespace}localname

the logical operation for this service has been defined before this standardization and must be changed.

4.4.3.5 CredentialsRenewal API

This library enables DILIGENT services to contact the CredentialsRenewal service hiding details of the Authentication management infrastructure. It also provides DILIGENT users with commands to add and remove credentials renewal tasks.

Two classes compose the CredentialsRenewal API: CredentialsRenewalAPI, containing classes to access the CredentialsRenewal service programmatically, and credentialsRenewalUI, containing CredentialsRenewal command line user interface.

Figure 101*. Credentials Renewal: CredentialsRenewal API class diagram*

4.4.3.6 Profile

The profile of the Credentials Renewal API is reported in Appendix A.15 as package included in the CredentialsRenewal Service profile.

4.4.3.7 Operations

org.diligentproject.dvos.credentialsRenewal.api.CredentialsRenewalAPI

This class contains methods to access the CredentialsRenewal service programmatically. It queries the DIS to discover the URL of the CredentialsRenewal service. It allows adding and removing credentials renewal tasks.

getInstance

```
CredentialRenewalAPI getInstance()
```

Build a new instance of CredentialRenewalAPI. Use DIS (DILIGENT Information Service) APIs to locate the CredentialRenewalService to use.

Return: An instance of CredentialRenewalAPI.

getInstance

```
CredentialRenewalAPI getInstance(File mapFilePath)
```

Build a new instance of CredentialRenewalAPI. Use DIS (DILIGENT Information Service) APIs to locate the CredentialRenewalService to use.

Parameters:

- MapFilePath: File object representing the map.xml file needed for DISHLSClient.

Return: An instance of CredentialRenewalAPI.

getInstance

```
CredentialRenewalAPI getInstance(String credentialRenewalServiceURL)
```

Build a new instance of CredentialRenewalAPI using the URL provided.

Parameters:

- CredentialRenewalServiceURL: URL of the credentialRenewal service to contact.

Return: An instance of CredentialRenewalAPI.

addCredentialsRenewal

```
void addCredentialsRenewal(String myProxyUsername, String myProxyPassword,
String credentialsId, String delegationServiceURL, VOMSRoleType[]
VOMSRole, int period)
```

Add a new CredentialsRenewal task to the CredentialsRenewal service.

Parameters:

- myProxyUsername: UserName of a MyProxy account
- myProxyPasswordCertification: Password of a MyProxy account
- credentialsId: credentials id for the delegated credentials
- delegationServiceURL: URL of the remote delegation service where credentials must be sent
- VOMSRole: List of roles to be included in the delegated credentials
- Period: period (in hours) of credentials validity

removeCredentialsRenewal

```
void removeCredentialsRenewal(String credentialsId, String
delegationServiceURL)
```

Remove a CredentialsRenewal task from the CredentialsRenewal service.

Parameters:

- credentialsId: Id of delegated credentials to be removed
- delegationServiceURL: URL of delegation service where credentials are delegated

org.diligentproject.dvos.credentialsRenewal.api.CredentialsRenewalUI

This class allows users to invoke operations of the CredentialsRenewalAPI class from a command line interface. The operations available to users are:

- addCredentialsRenewal: Add a new CredentialsRenewal task.
- removeCredentialsRenewal: Remove a CredentialsRenewal task.

Details on command line arguments can be found using:

```
java CredentialRenewalUI -usage
```

4.4.3.7.1 Dependencies

None

4.4.3.7.2 Known bugs and problems

None

4.5 Authorization Management

The DILIGENT authorization model relies on the underlying Java WS Core authorization framework. It enables DILIGENT Services to separate security issues from service specific behaviour. This is achieved using a chain of authorization handlers managed by the Service Container. These handlers are asked during evaluation of incoming requests in order to permit or deny access to a service. Authorization handlers then contact Authorization Services asking for information about user and resources being accessed.

The purpose of this section is to explain how Authorization Service and handlers can be installed and used.

A picture of components constituting this service is provided in *Figure 102*.

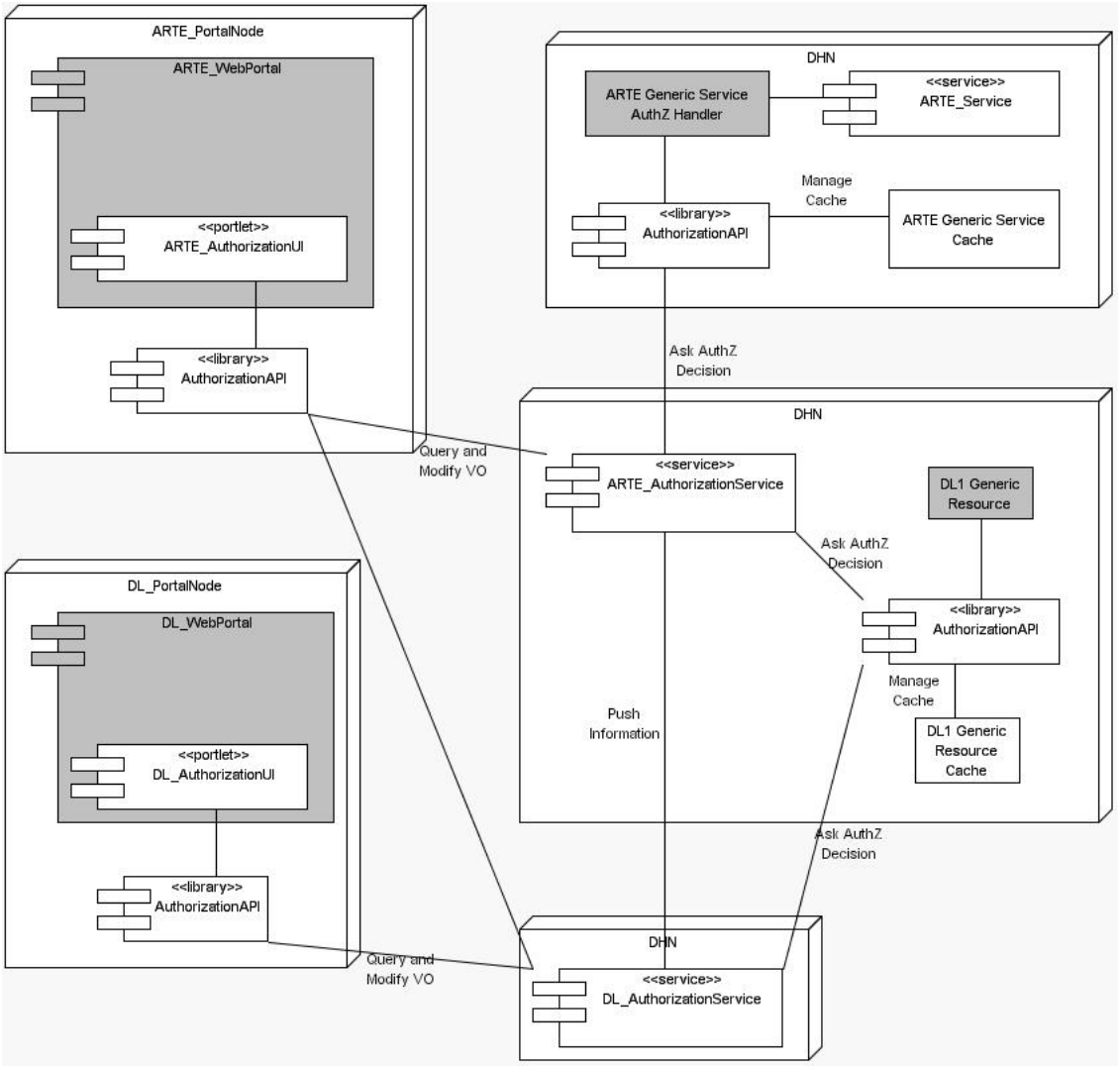

Figure 102*. Authorization Service: Main Components and relatioships*

4.5.1 Authorization Service

The DILIGENT Authorization service is a WSRF-compliant Web Service. It is in charge to store the authorization information of DILIGENT VOs. It also acts as Policy Decision Point (PDP); DILIGENT services can contact it to obtain authorization decisions.

A VOMS VO containing all users and roles backs the DILIGENT VO. The VOMS server is used to create signed Attribute Certificates (AC) to be included in the Proxy Certificate of DILIGENT users. Attribute Certificates contain the set of roles held by the user during the session.

The VO authorization is based on a set of logical operations users can perform over resources. *Resource Managers* have to define these logical operations and grant them to the VO. *VO Mangers* have to define roles, grant them a set of logical operations among those available in the VO, and assign roles to users.

To enforce VO authorization, DILIGENT Services have to contact the Authorization service. This service verifies that roles held by the user have permissions to perform the logical operation. Roles held by the user are included in the proxy certificate attached to service request.

The use of logical operations gives *Resource Managers* a way to group service operations in consistent sets, thus exempting *VO Managers* from the need to know which service operations must be granted a role to perform a logical operation.

This threefold use of the Authorization service leads to the definition of three interfaces:

- OperationAdministration – Used by *Resource Managers* to manage logical operations;
- VOAuthorization – Used by *VO Managers* to manage VO authorization;
- VOQuery – Used by DILIGENT Services to obtain authorization decision.

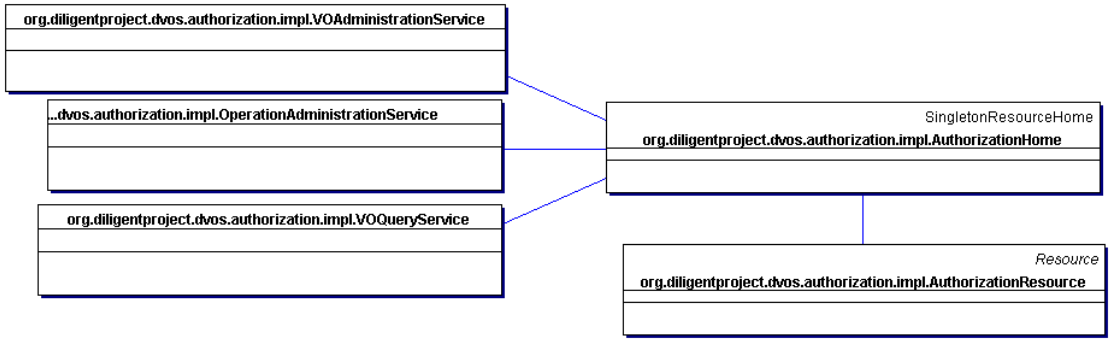

Figure 103*. Authorization Service: global class diagram*

4.5.1.1 Profile

The profile of the Authorization service is reported in Appendix A.16.

4.5.1.2 Managed Resources

Each VO managed by the Authorization service is represented by an instance of the *AuthorizationResource*. The alpha DILIGENT prototype does not include support for VO hierarchy; therefore, the DILIGENT Authorization service adopts the Singleton Resource pattern. Further implementation, including support for VO hierarchy, will adopt the Multiple Resource pattern.

4.5.1.2.1 Configuration

Internal configuration

Once the service has been deployed, the `VOSetup.xml` file (contained in the `/etc` subdirectory) must be configured in order to properly initialize the DILIGENT-VO. All XML elements and attributes reported in the following table must be properly set:

Field	Type	Description
`voInitialStatus.operations.operation`	Element	A logical operation to be granted to the VO. Some logical operations to manage the authorization service itself are already defined. Multiple occurrences of this element can be defined in the `operations` tag.
`voInitialStatus.roles.role`	Element	A role to be defined in the VO. Two roles are at least defined in the `VOSetup.xml` file: "`DILIGENT VO Manager`" and "`DILIGENT Resource Manager`". The "`DILIGENT VO Manager`" has permissions to

		manage VO authorization. The "DILIGENT Resource Manager" has permissions to manage logical operations in the VO. Multiple occurrences of this element can be defined in the roles tag.
voInitialStatus.roles. role.grantPermission	Element	A permission to grant a logical operation to the role. Multiple occurrences of this element can be defined in the role tag.

Table 2. VOSetup configuration parameters

VOMS configuration

As described in the paragraph 4.5.1.2.2 the Authorization service depends on the VOMS. After the VOMS installation and configuration, a VO corresponding to the DILIGENT VO must be created in the VOMS (the name of the VOMS VO can be different from the DILIGENT VO name). DILIGENT users must be registered in the VOMS VO.

DILIGENT roles must be created in the VOMS VO. The name of these roles must correspond to the role name of DILIGENT roles.

4.5.1.2.2 Dependencies

The DILIGENT Authorization service depends on the VOMS to create short-term proxy credentials for users. See paragraph 'VOMS configuration' in Section 4.5.1.2.1 for details about VOMS configuration.

4.5.1.2.3 Security Profile

Two logical operations are defined for the Authorization service itself:

- org.diligentproject.security.authorizationOperation.dvos.adminVO – Used to grant administrative rights over a VO.
- org.diligentproject.security.authorizationOperation.dvos.adminResource – Used to grant administrative rights over resources.

When the DILIGENT VO is created these logical operations are granted to *VO Managers* and *Resource Managers*, in accordance with the VOSetup.xml file (see Section 4.5.1.2.1).

4.5.1.2.4 Known bugs and problems

4.5.2 Authorization API

This library enables DILIGENT services to contact Authorization services hiding thedetails of the Authorization management infrastructure. It provides useful handlers DILIGENT services can use to enforce authorization. Some commands are also implemented allowing DILIGENT users to manage Authorization services through a command line interface.

Three classes compose the core of Authorization API: *VOAuthorizationAPI*, *OperationAdministrationAPI* and *VOQueryAPI*. Each of these classes allows clients to invoke operations on the corresponding interface of authorization service (see Section 4.5.1).

These classes are in charge to discover the location of the authorization service using DIS functionalities. They also set authentication parameters in the service stubs in order to successfully contact the authorization service.

The VOAuthorizationPDP class can be used by DILIGENT services as Policy Enforcement Point handler to enforce VO authorization. It can be configured as described below.

Three classes implement the command line interface for the DILIGENT users: *VOAuthorizationUI*, *OperationAdministrationUI* and *VOQueryUI.*

Figure 104. *Authorization API: globa class diagram*

4.5.2.1 Profile

The profile of the Authorization API is reported in Appendix A.16 as package included in the Authorization Service profile.

4.5.2.2 Implementation Details

All the following classes belong to the package

`org.diligentproject.dvos.authorization.`

api.OperationAdministrationAPI

Logical operations can be added and removed from DILIGENT using instances of this class. Instances of this class can also be used by *Resource Managers* to set Sharing Rules in the DILIGENT VO.

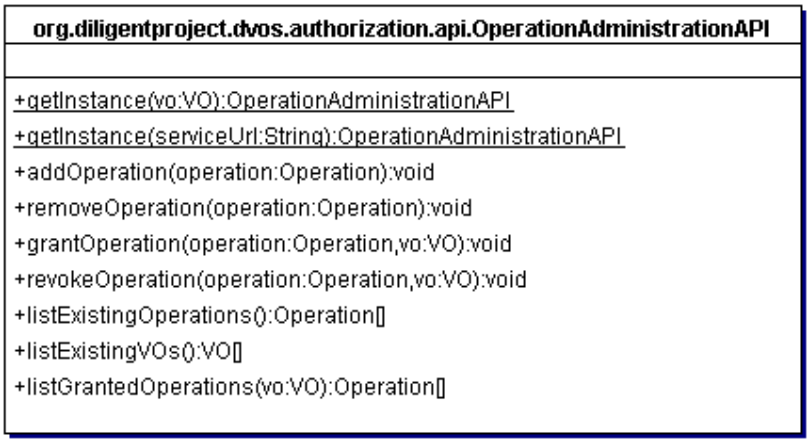

Figure 105. *Authorization API: OperationAdministrationAPI class diagram*

getInstance

`OperationAdministrationAPI getInstance(VO vo)`

Build a new instance of OperationAdministrationAPI starting from the VO name. Use DIS (DILIGENT Information Service) API to locate which OperationAdministration service to use.

Parameters:

- vo: The name of the VO where to manage sharing rules.

Return: the OperationAdministrationAPI object bound to that VO

getInstance

`OperationAdministrationAPI getInstance(String serviceUrl)`

Build a new instance of OperationAdministrationAPI starting from the OperationAdministration service URL.

Parameters:

- serviceUrl: The URL of the OperationAdministration service where to manage sharing rules.

Return: the OperationAdministrationAPI object bound to that URL

addOperation

`void addOperation(Operation operation)`

Add a new logical operation to DILIGENT

Parameters:

- operation: the operation to add.

removeOperation

`void removeOperation(Operation operation)`

Remove a logical operation from DILIGENT.

Parameters:

- operation: the operation to remove.

grantOperation

```
void grantOperation(Operation operation, VO vo)
```

Set a sharing rule in a VO granting an existing operation to that VO.

Parameters:

- operation: the existing operation to grant.
- vo: the existing VO to which the operation is to be granted.

revokeOperation

```
void revokeOperation(Operation operation, VO vo)
```

Unset a sharing rule in a VO revoking an existing operation to that VO.

Parameters:

- operation: the existing operation to revoke.
- vo: the existing VO to which the operation is to be revoked.

listExistingOperations

```
Operation[] listExistingOperations()
```

List existing logical operations.

Return: an array containing existing logical operations.

listExistingVOs

```
VO[] listExistingVOs()
```

List existing VOs.

Return: an array containing existing VOs.

listGrantedOperations

```
Operation[] listGrantedOperations(VO vo)
```

List logical operations granted to a VO.

Parameters:

- vo: the VO where to search for granted Operations.

Return: an array containing logical operations granted to the VO.

api.VOAdministrationAPI

This is the client-side class representing a VOAdministration service. Through instances of this class VO Managers can creates and delete roles, assign and remove roles to users and grant or revoke permissions to roles (being compliant with sharing rules set over VO by Resource Managers). Each instance is bound to a VO and allows managing authorization in that VO only.

Figure 106*. Authorization API: VOAdministrationAPI class diagram*

getInstance

`VOAdministrationAPI getInstance(VO vo)`

Build a new instance of VOAdministrationAPI starting from the VO name. Use the DIS (DILIGENT Information Service) API to locate the VOAdministration service to use.

Parameters:

- vo: the name of the VO where to manage authorization.

Return: the OperationAdministrationAPI object to manage authorization of that VO

getInstance

`VOAdministrationAPI getInstance(String serviceUrl)`

Build a new instance of VOAdministrationAPI starting from the VOAdministration service URL.

Parameters:

- serviceUrl: the URL of the VOAdministration service where to manage authorization.

Return: the OperationAdministrationAPI object to manage authorization of that VO

addUser

`void addUser(User user)`

Add a new user to DILIGENT.

Parameters:

- user: the user to add.

removeUser

```
void removeUser(User user)
```

Remove a user from DILIGENT.

Parameters:

- user: the user to remove.

listAvailableUsers

```
User[] listAvailableUsers()
```

List users can be managed in this VO. This method returns all users of the parent VO of this VO. If this VO represents the DILIGENT-VO (root of the VO hierarchy) then the list of available users is the list of DILIGENT users.

Return: an array containing all users can be managed in the VO.

listAvailableOperations

```
Operation[] listAvailableOperations()
```

List logical operations can be assigned to roles in this VO. This method return all logical operations granted to this VO by Resource Managers.

Return: an array containing all logical operations the VO Manager can grant to roles in this VO.

createRole

```
Role createRole(String roleName, String roleDescription, Role[]
parentRoles)
```

Creates a new role in this VO.

Parameters:

- roleName: the name of the role (cannot be null)
- roleDescription: the description of the role
- parentRoles: an array containing all parent roles of the role being created.

Return: The new role.

deleteRole

```
void deleteRole (Role role)
```

Delete a role (and all its children roles) in this VO.

Parameters:

- role: the role to delete

listRoles

```
Role[] listRoles ()
```

List roles created in this VO.

Return: an array containing all roles created in this VO.

listSubRoles

`Role[] listSubRoles (Role role)`

List children of a role. The role must belong to this VO.

Parameters:

- role: the role for which all the children are to be returned

Return: An array containing all children of the role in this VO.

listSuperRoles

`Role[] listSuperRoles (Role role)`

List parents of a role. The role must belong to this VO.

Parameters:

- role: the role for which all the parents are to be returned

Return: an array containing all parents of the role in this VO.

assign

`void assign(User user, Role role)`

Assign a role to a user. Both the role and the user must exist in this VO.

Parameters:

- user: the user to whom the role is assigned
- role: the role to be assigned to the user

unassign

`void unassign(User user, Role role)`

Remove a role from a user. Both the role and the user must exist in this VO.

Parameters:

- user: the user for whom the role is removed.
- role: the role to be unassigned to the user

grantPermission

`void grantPermission(Role role, Permission permission)`

Grant a new permission to a role. Both the role and the logical operation set in the permission must exist in this VO.

Parameters:

- role: the role to which the permission is to be granted.
- permission: the permission to be granted to the role

revokePermission

`void revokePermission(Role role, Permission permission)`

Revoke the permission to a role. Both the role and the logical operation set in the permission must exist in this VO.

Parameters:

- role: the role to which the permission is to be revoked.
- permission: the permission to be revoked to the role

listRolePermission

`Permission[] listRolePermission(Role role)`

List permissions granted to a role in this VO.

Parameters:

- role: the role for which to list permissions

Return: The list of permissions granted to the role.

listRoleUsers

`User[] listRoleUsers (Role role)`

List users assigned to a role in this VO.

Parameters:

- role: the role for which to list assigned users

Return: the list of users assigned to the role.

listUserRoles

`Role[] listUserRoles(User user)`

List roles assigned to a user in this VO.

Parameters:

- user: the user for whom to list assigned roles

Return: the list of roles assigned to the user.

api.VOQueryAPI

This is the client-side class representing a VOQuery service. Instances of this class can be used to check access to VO resources. Each instance is bound to a VO and performs access checks based on the authorization defined in that VO.

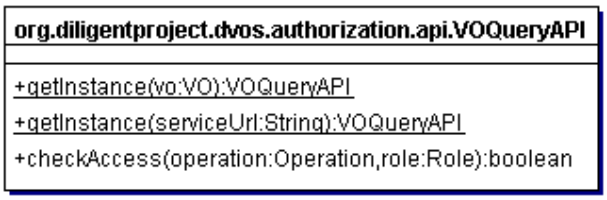

Figure 107*. Authorization API: VOQueryAPI class diagram*

getInstance

`VOQueryAPI getInstance(VO vo)`

Build a new instance of VOQueryAPI starting from the VO name. Use DIS (DILIGENT Information Service) APIs to locate the VOQuery service to use.

Parameters:

- vo: the name of the VO where to check for authorization.

Return: the VOQueryAPI object to check for VO authorization.

getInstance

`VOQueryAPI getInstance(String serviceUrl)`

Build a new instance of VOQueryAPI starting from the VOQuery service URL.

Parameters:

- serviceUrl: the URL of the VOQuery service where to check for VO authorization.

Return: the VOQueryAPI object to check for VO authorization.

checkAccess

`boolean checkAccess(Operation operation, Role role)`

Check if the role has permission to perform the operation in this VO.

Parameters:

- operation: the operation to check for.
- role: the role to check for.

Return: true if the role is entitled to perform the operation in this VO, false otherwise.

ui.OperationAdministrationUI

It represents a simple class that allows to users to invoke the ui.OperationAdministrationAPI operations. It's composed only by a main method.

Users could invoke these following operations:

- addOperation: Add a new logical operation to DILIGENT.
- removeOperation: Remove a logical operation from DILIGENT.
- grantOperation: Set a sharing rule in a VO granting an existing operation to that VO.
- revokeOperation: Unset a sharing rule in a VO revoking an existing operation to that VO.
- listExistingOperations: List existing logical operations.
- listExistingVOs: List existing VOs.
- listGrantedOperations: List logical operations granted to a VO.

Details on command line arguments can be found using:

```
java OperationAdministrationUI -usage
```

ui.VOAdministrationUI

It represents a simple class that allows to users to invoke the ui.VOAdministrationAPI operations. It's composed only by a main method.

Users could invoke these following operations:

- Add User: Add a new user to DILIGENT.
- RemoveUser: Remove a user from DILIGENT.

- List available users: List users can be managed in this VO. This method returns all users of the parent VO of this VO. If this VO represents the DILIGENT-VO (root of the VO hierarchy) then the list of available users is the list of DILIGENT users.
- List available operations: List logical operations can be assigned to roles in this VO. This method returns all logical operations granted to this VO by Resource Managers.
- Role creation: Creates a new role in this VO.
- Role removal: Delete a role (and all its children roles) in this VO.
- List roles: List all roles belonging to this VO.
- List sub roles: List children of a role. The role must belong to this VO.
- List super roles: List parents of a role. The role must belong to this VO.
- Assign: Assign a role to a user. Both the role and the user must exist in this VO.
- Unassign: Remove a role from a user. Both the role and the user must exist in this VO.
- Grant permission: Grant a new permission to a role. Both the role and the logical operation set in the permission must exist in this VO.
- Revoke permission: Revoke the permission to a role. Both the role and the logical operation set in the permission must exist in this VO.
- List role permissions: List permissions granted to a role in this VO.
- List role users: List users assigned to a role in this VO.
- List user roles: List roles assigned to a user in this VO.

Details on command line arguments can be found using:

```
java VOAdministrationUI -usage
```

ui.VOQueryUI

It represents a simple class that allows to users to invoke the ui.VOQueryAPI operations. It's composed only by a main method. Users could invoke these following operations:

- checkAccess: Check if the role has permission to perform the operation in this VO.

Details on command line arguments can be found using:

```
java VOQueryUI -usage
```

handler.VOAuthorizationPDP

This is the handler used to enforce VO authorization policies. This handler extracts VOMS Attribute Certificates (AC) from the certificate attached to the service request and contacts the DVOS Authorization services to check for authorization.

Handler Configuration

Two parameters must be defined in the service WSDD configuration file:

- VOMSCertificateDirectory: The absolute name of a local directory where to find VOMS certificates (needed to validate VOMS Attribute Certificates). The default value is /etc/grid-security/vomsdir/.

- VOAuthorizationHandlerFile: The relative name (to the container directory) of a file containing the configuration for this handler.

The file pointed to by the `VOAuthorizationHandlerFile` parameter must be a property file defining two sets of properties:

- org.diligentproject.dvos.authorization.handler.voQueryServiceURL.<VOName>=<VO_URL>
- org.diligentproject.dvos.authorization.handler.operationMapping.<serviceOperationName>=<logicalOperationId>

where:

`<VOName>` is the name of a VOMS group (corresponding to a VO in DILIGENT)

`<VOURL>` is the URL of the VOQuery Service managing that VO

`<serviceOperationName>` is the QName of a service operation

`<logicalOperationId>` is the logical operation Id to check access for

The property:

```
org.diligentproject.dvos.authorization.handler.operationMapping.default=<logicalOperationId>
```

can be used to define a default mapping to use. If the service operation is not explicitly mapped and a default has not been defined, the QName of the service operation is used as logical operation id.

Keep in mind that characters `#`, `!`, `=`, and `:` in properties files must be written with a preceding slash (`\`) to ensure that they are properly loaded.

4.5.2.2.1 Dependencies

This library depends on VOMS API libraries to extract the AC from the proxy certificate.

4.6 UserGroupManagement Service

This service is in charge to store information about the set of DILIGENT users and DILIGENT groups. The information maintained about DILIGENT users or groups is the minimal set allowing to uniquely identify users and groups in the DILIGENT infrastructure. Other personal information about users and groups is managed by the DILIGENT Personalization service.

The UserGroupManagement Service class diagram is reported in *Figure 108*.

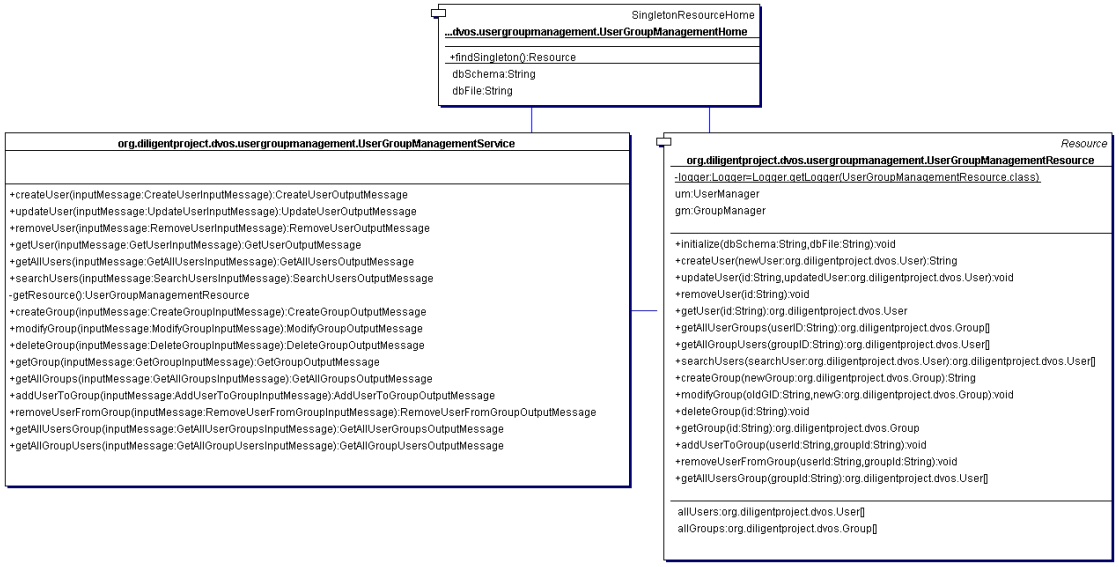

Figure 108. *UserGroupManagement Service: global class diagram*

4.6.1.1.1 Profile

The profile of the UserGroupManagement service is reported in Section Appendix A.17.

4.6.1.1.2 Managed Resources

The UserGroupManagement service is a singleton statefull Web Service bound to a UserGroupManagementResource. The UserGroupManagementResource manages credentials renewal tasks.

4.6.1.1.3 Configuration

No configuration is needed for this service

4.6.1.1.4 Operations

createUser

`CreateUserOutputMessage createUser(CreateUserInputMessage inputMessage)`

It creates a new user.

Parameters:

- message: a complex type specified in the UserGroupManagement.wsdl containing following fields:
    - id: The user Id. This Id is unique in DILIGENT. This input field is ignored by this method.
    - distinguishedName: the Distinguished Name of the user
    - certificationAuthority: The Certification Authority of the user
    - name: The short name of the user in DILIGENT.
    - email: the user's email.

Return: A complex type containing the Id of the user just created

updateUser

`UpdateUserOutputMessage updateUser(UpdateUserInputMessage inputMessage)`

Update a user

Parameters:

- message: a complex type specified in the UserGroupManagement.wsdl containing following fields:
    - id: The Id of the user to be updated.
    - distinguishedName: the new Distinguished Name of the user
    - certificationAuthority: The new Certification Authority of the user
    - name: The new short name of the user in DILIGENT.
    - email: the new user's email.

Return: An empty complex type

removeUser

`RemoveUserOutputMessage removeUser(RemoveUserInputMessage inputMessage)`

Remove a user

Parameters:

- message: a complex type specified in the UserGroupManagement.wsdl containing following fields:
    - id: The Id of the user to be removed.

Return: An empty complex type

getUser

`GetUserOutputMessage getUser(GetUserInputMessage inputMessage)`

Get user's information

Parameters:

- message: a complex type specified in the UserGroupManagement.wsdl containing following fields:
    - id: The Id of the user.

Return: A complex type containing user's information.

getAllUsers

`GetAllUsersOutputMessage getAllUsers(GetAllUsersInputMessage inputMessage)`

Get all users information

Parameters: none

Return: A sequence of complex types containing all users information.

searchUsers

`SearchUsersOutputMessage searchUsers(SearchUsersInputMessage inputMessage)`

Search for the set of users matching the user template provided. All users completely matching the information provided are returned

Parameters:

- message: a complex type specified in the UserGroupManagement.wsdl containing following fields:
    - distinguishedName: the Distinguished Name to search for
    - certificationAuthority: The Certification Authority to search for
    - name: The short name to search for.
    - email: the email to search for.

Return: A sequence of complex type containing the DN, CA, name and email of all users matching the template.

createGroup

```
CreateGroupOutputMessage createGroup(CreateGroupInputMessage inputMessage)
```

Creates a group

Parameters:

- message: a complex type specified in the UserGroupManagement.wsdl containing following fields:
    - id: the group Id. This input field is ignored in this operation.
    - name: The group name
    - description: The group description.

Return: A complex type containing the Id of the group just created

modifyGroup

```
ModifyGroupOutputMessage modifyGroup(ModifyGroupInputMessage inputMessage)
```

Modifies a group

Parameters:

- message: a complex type specified in the UserGroupManagement.wsdl containing following fields:
    - id: the Id of the group to be modified.
    - name: The new group name
    - description: The new group description.

Return: An empty complex type

deleteGroup

```
DeleteGroupOutputMessage deleteGroup(DeleteGroupInputMessage inputMessage)
```

Delete a group

Parameters:

- message: a complex type specified in the UserGroupManagement.wsdl containing following fields:
    - id: the Id of the group to be deleted.

Return: An empty complex type

getGroup

`GetGroupOutputMessage getGroup(GetGroupInputMessage inputMessage)`

Get group information

Parameters:

- message: a complex type specified in the UserGroupManagement.wsdl containing following fields:
    - id: the Id of the group.

Return: A complex type containing information about the group.

getAllGroups

`GetAllGroupsOutputMessage getAllGroups(GetAllGroupsInputMessage inputMessage)`

Get all groups information

Parameters: none

Return: A sequence of complex types containing information about all groups.

addUserToGroup

`AddUserToGroupOutputMessage addUserToGroup(AddUserToGroupInputMessage inputMessage)`

Add a user to a group

Parameters:

- message: a complex type specified in the UserGroupManagement.wsdl containing following fields:
    - userId: the Id of the user to be added.
    - GroupdId: The Id of the group where to add the user

Return: An empty complex type

removeUserFromGroup

`RemoveUserFromGroupOutputMessage removeUserFromGroup(RemoveUserFromGroupInputMessage inputMessage)`

Remove a user from a group

Parameters:

- message: a complex type specified in the UserGroupManagement.wsdl containing following fields:
    - userId: the Id of the user to be removed.
    - groupdId: The Id of the group where to remove the user from

Return: An empty complex type

getAllUsersGroup

```
GetAllUserGroupsOutputMessage
getAllUsersGroup(GetAllUserGroupsInputMessage inputMessage)
```

Get all groups a user belongs to

Parameters:

- message: a complex type specified in the UserGroupManagement.wsdl containing following fields:
    - userId: the Id of the user.

Return: A sequence of complex types containing information about all groups the user belongs to

getAllGroupUsers

```
GetAllGroupUsersOutputMessage
getAllGroupUsers(GetAllGroupUsersInputMessage inputMessage)
```

Get all the users of a group

Parameters:

- message: a complex type specified in the UserGroupManagement.wsdl containing following fields:
    - groupId: the Id of the group.

Return: A sequence of complex types containing information about all the users of the group.

4.6.1.2 Implementation details

All these classes are part of the package

`org.diligentproject.dvos.usergroupManagement`

UserGroupManagementService

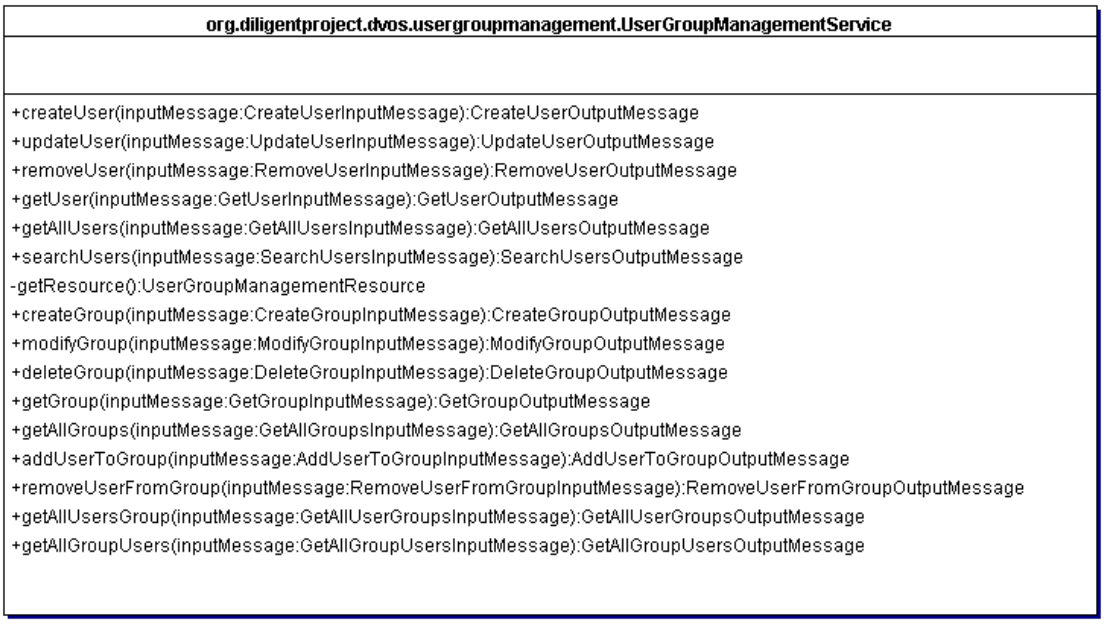

Figure 109. *UserGroupManagement Service: UserGroupManagementService class diagram*

This class implements the UserGroupManagement Service interface components and exposes service operations.

UserGroupManagementResource

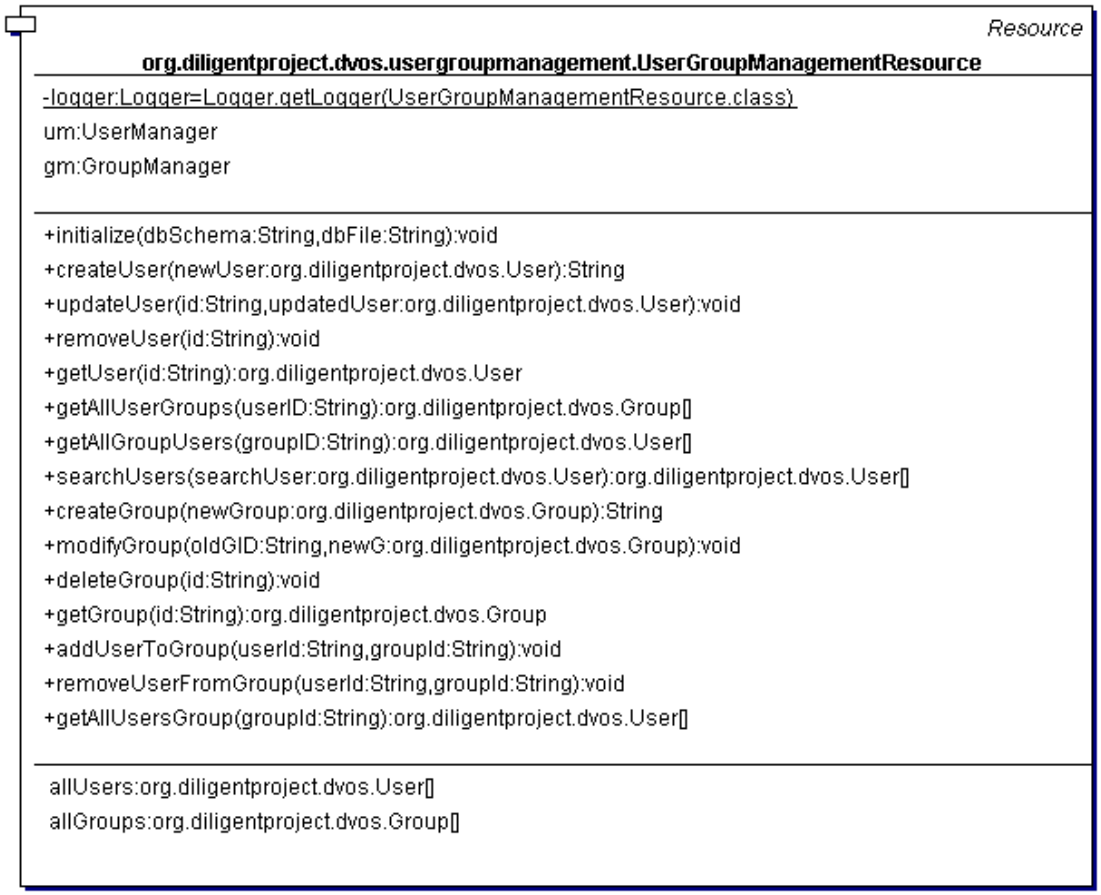

Figure 110*. UserGroupManagement Service: UserGroupManagementResource class diagram*

This class implements the UserGroupManagement resource in charge to manage users and groups.

UserGroupManagementHome

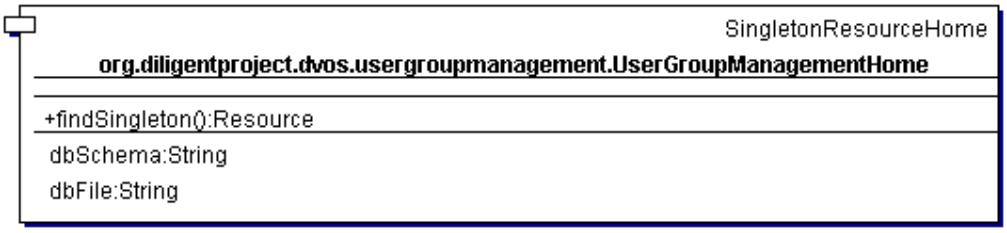

Figure 111*. UserGroupManagement Service: UserGroupManagementHome class diagram*

This class implements the UserGroupManagement Home.

4.6.1.2.1 Dependencies

The DILIGENT Authorization service depends on the VOMS to create short-term proxy credentials for users. See paragraph 'VOMS configuration' in Section 4.5.1.2.1 for details about VOMS configuration.

4.6.1.2.2 Security Profile

Following logical operations are defined for the UserGroupManagement service:

- {http://www.diligentproject.org/namespaces/dvos/usergroupmanagement}manageUsers – Grant write access to service operations related to users (createUser, updateUser, removeUser).
- {http://www.diligentproject.org/namespaces/dvos/usergroupmanagement}listUsers – Grant read access to service operations related to users (getUser, getAllUser, searchUser).

4.6.1.2.3 Known bugs and problems

Logical Operations for this service must be refactored to include group management.

4.6.2 UserGroupManagement API

This library enables DILIGENT services to contact UserGroupManagement service hiding the details of the service. It also provides the DILIGENT users with a command line interface to manage DILIGENT users and groups.

Two classes compose the UserGroupManagement API: *UserGroupManagementAPI*, containing classes to access the UserGroupManagement service programmatically, and *UserGroupManagementUI*, containing UserGroupManagement command line user interface.

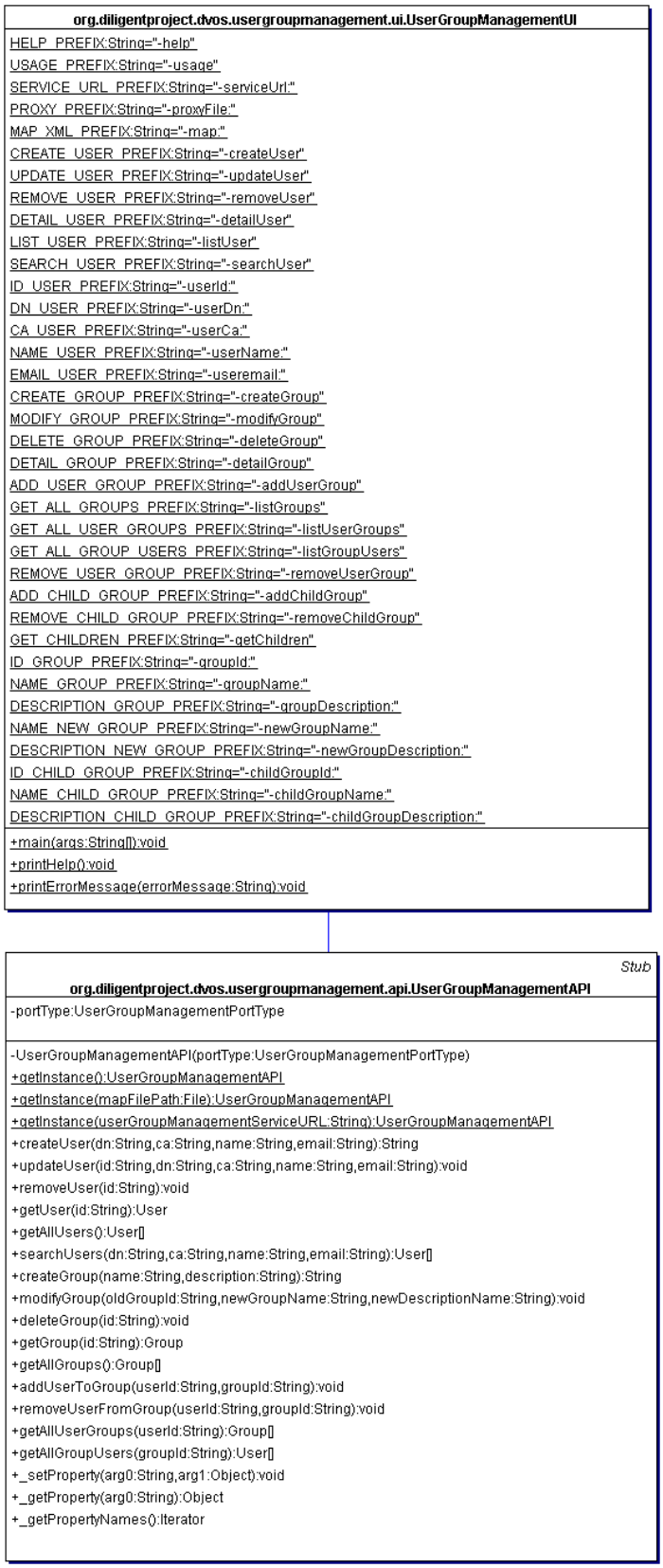

Figure 112. *UserGroupManagement API: global class diagram*

4.6.2.1 Profile

The profile of the UserGroupManagement API is reported in Appendix A.17 as package included in the UserGroupManagement Service profile.

4.6.2.2 Implementation Details

org.diligentproject.dvos.credentialsRenewal.api.UserGroupManagementAPI

This class contains methods to access the UserGroupManagement service programmatically. It queries the DIS to discover the URL of the UserGroupManagement service.

getInstance

```
UserGroupManagementAPI getInstance ()
```

Build a new instance of UserGroupManagementAPI. Use DIS (DILIGENT Information Service) APIs to locate the UserGroupManagementService to use.

Parameters: none

Return: An instance of UserGroupManagementAPI.

getInstance

```
UserGroupManagementAPI getInstance (File mapFilePath)
```

Build a new instance of UserGroupManagementAPI. Use DIS (DILIGENT Information Service) APIs to locate the UserGroupManagementService to use.

Parameters:

- mapFilePath: File object representing the map.xml file needed for DISHLSClient.

Return: An instance of UserGroupManagementAPI.

getInstance

```
UserGroupManagementAPI getInstance(String userGroupManagementServiceURL)
```

Remove a user

Parameters:

- UserGroupManagementServiceURL: URL of the UserGroupManagementService to contact.

Return: An instance of UserGroupManagementAPI.

createUser

```
String createUser(String dn, String ca, String name, String email)
```

Create a new DILIGENT user.

Parameters:

- dn: Distinguished Name of the new user.
- ca: Certification Authority
- name: user name
- email: user email

Return: a String representing the ID of the new user created.

updateUser

`void updateUser(String id, String dn, String ca, String name, String email)`

Update a DILIGENT user.

Parameters:

- id: the id of the user
- dn: new Distinguished Name of the user.
- ca: new Certification Authority
- name: new user name
- email: new user email

removeUser

`void removeUser(String id)`

Remove a DILIGENT user

Parameters:

- id: the id of the user to be removed

getUser

`User getUser(String id)`

Returns information about a DILIGENT User.

Parameters:

- Id: Id of the user to retrieve.

Return: information about the User

getAllUsers

`User[] getAllUsers()`

Returns information about all DILIGENT Users.

searchUsers

`User[] searchUsers(String dn, String ca, String name, String email)`

Search for the set of users matching the user template provided. All users completely matching the information provided are returned

Parameters:

- dn: Distinguished Name of the new user.
- ca: Certification Authority
- name: user name
- email: user email

Return: All users completely matching the information provided

createGroup

`String createGroup(String name, String description)`

Create a new DILIGENT group

Parameters:

- Id: Identifier of the new group
- Name: Name of the new group
- description: Description of the new group

Return: a String representing the ID of the new group created.

modifyGroup

`void modifyGroup(String oldGroupId, String newGroupName, String newDescriptionName)`

Update a DILIGENT group.

Parameters:

- Id: Identifier of the group
- Name: New name of the group
- description: New description of the group

deleteGroup

`void deleteGroup(String id)`

Delete DILIGENT group.

Parameters:

- id: The Id of the group to be deleted

Return: none

getGroup

`Group getGroup(String id)`

Returns information about a Group.

Parameters:

- id: The Id of the group

Return: information about the Group

getAllGroups

`Group[] getAllGroups()`

Returns all Groups.

- Parameters:none

Return: all DILIGENT Groups

addUserToGroup

`void addUserToGroup(String userId, String groupId)`

Add a user to a Group.

Parameters:

- userId: the Id of the user to add to the group
- groupId: the Id of the group where to add the user

Return: none

removeUserFromGroup

```
void removeUserFromGroup(String userId, String groupId)
```

Remove a user from a Group.

Parameters:

- userId: the Id of the user to remove
- groupId: the Id of the group where to remove the user from

Return: none

getAllGroupUsers

```
User[] getAllGroupUsers(String groupId)
```

Get all users of a group

Parameters:

- groupId: the Id of the group

Return: All users belonging to the group

getAllUserGroups

```
Group[] getAllUserGroups(String userId)
```

Return all the groups the user belongs to

Parameters:

- userId: the Id of the user

Return: All the groups the user belongs to

org.diligentproject.dvos.credentialsRenewal.ui.UserGroupManagementUI

This class allows users to invoke operations of the UserGroupManagementAPI class from a command line interface. Operations available to users are:

- createUser: creates a new user
- updateUser: modifies user information
- removeUser: removes a user
- detailUser: get details about a user
- listUser: list users
- searchUser: seach for a user
- createGroup: create a new group

- modifyGroup: modify group details
- deleteGroup: delete a group
- detailGroup: get details about a group
- addUserGroup: add a user to a group
- removeUserGroup: remove a user from a group

Details on command line arguments can be found using:

```
java UserGroupManagementUI -usage
```

5 BROKER & MATCHMAKER SERVICE

5.1 Introduction

The Broker & Matchmaker (BMM) service supports the Keeper service in deploying a new Digital Library on a set of DILIGENT Hosting Nodes (DHNs). In particular, once the Keeper has identified the set of packages needed to build a new DL their requirements and their relationships, the BMM's job is to identify a set DHNs to be used as target hosts for the deployment. The Broker & Matchmaker:

- Supports the Keeper in the deployment of a Digital Library (DL).
- Enables the Keeper to notify a deployment failure.
- Queries the Diligent Information Service (DIS) in order to gather the status of the DHNs.
- Queries the DIS in order to gather the dependencies and the requirements of a package (by keeping the Service Profile).
- Selects the more appropriates set of DHNs to deploy a specified set of packages.
- Stores information related to the associations among packages and DHNs that fail, i.e. the deploy operation has not been successfully executed for them.

This service, as underlined in the previous footnote, will be fully supported in the Beta release; therefore, some details in the description are currently missing.

5.2 Components

5.2.1 Logical view

Broker & Matchmaker architecture is composed of four main packages:

- *DISConnector package*: is in charge of keeping up-to-date tracks of the DHN profiles received from the DIS
- *KeeperConnector package*: allows Keeper to send a matching request, and notifies the Keeper with the response
- *BMMService package*: provides the core functionalities of the BMM Service. It queries the DIS in order to gather information about packages and it forwards the KeeperConnector request to the MatchMaker
- *MatchMaker package*: calculates, by running the matching algorithm, the association among packages and DHNs

No Graphical User Interface is provided by the Broker & Matchmaker, as this service is not meant to interact with human users.

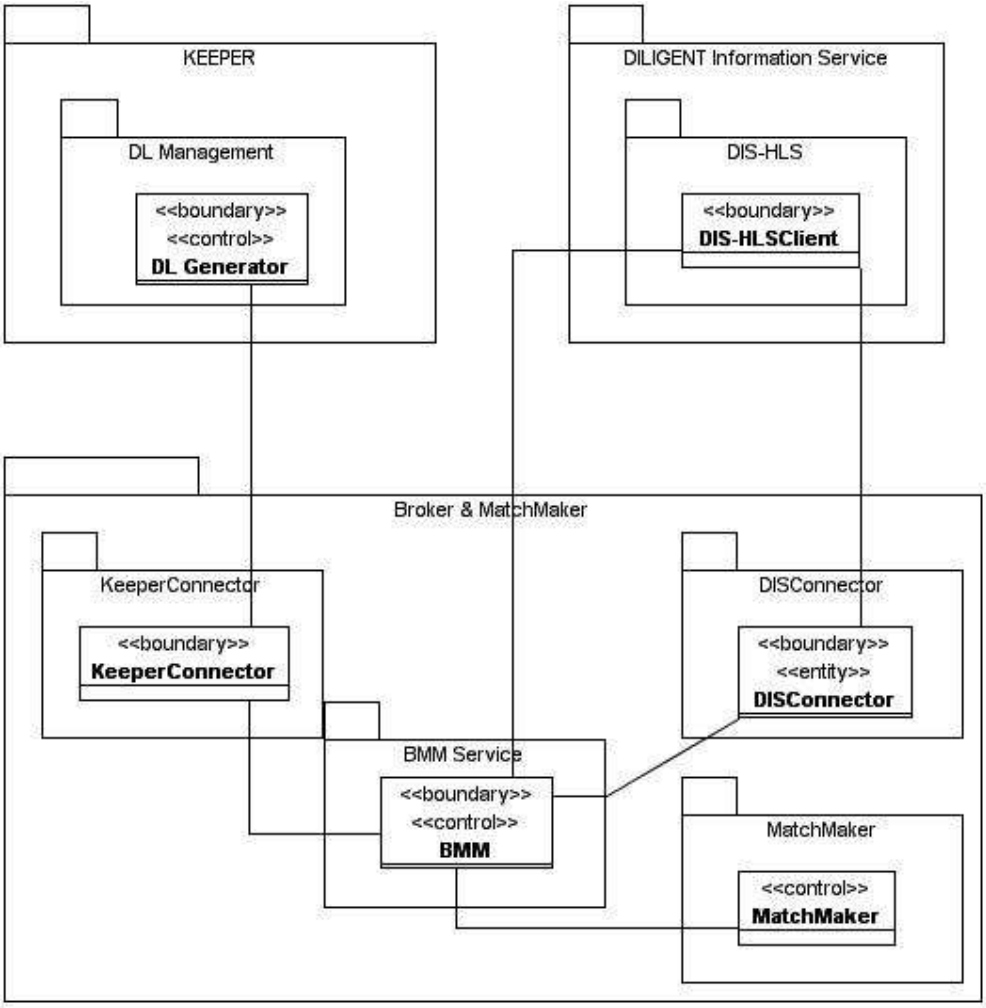

Figure 113 *Broker & Match Maker - Logical View*

5.2.2 Deployment view

From a deployment point of view, two components have been identified. As depicted in the next Figure, these components are designed to be hosted in different networked locations:

- B&MMService – it is composed by the components: BMMService, MatchMaker and DISConntector. To improve availability, multiple instances of this service can be deployed on different DHNs.
- KeeperConnector – provides local access to BMM functionalities. This library is co-hosted with the DL Management Service. It provides the interface which allows a Keeper to contact the B&MMService.

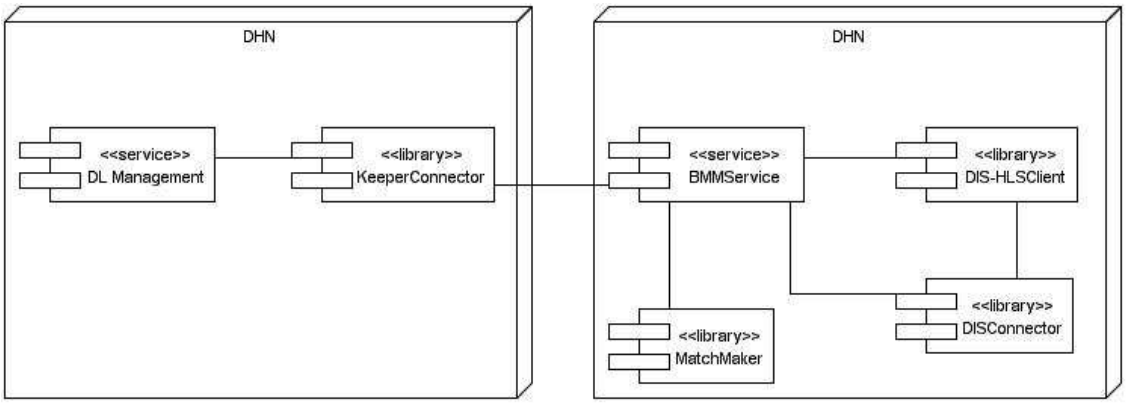

Figure 114. *Broker & MatchMaker - Deployment View*

5.2.3 KeeperConnector

The KeeperConnector is a library that is available on each DHN hosting the DL Management service (see Section 3.2.2). It allows the Keeper to submit a matching request to the Broker & Matchmaker for a DL.

The KeeperConnector delegates the request to the BMM Service. When the response is available, the KeeperConnector notifies the Keeper (a MatchListener object associated to it) with the matching result, i.e. a list of pairs <package, DHN>.

The library allows the Keeper also to notify in case, for some reasons, the deployment <package, DHN> fails.

5.2.3.1 Profile

[to be completed in the next release]

5.2.3.2 Operations

The operations offered by this library are:

matchRequest

```
void  matchRequest(String request, MatchListener listener)
```

It allows the Keeper to send a matching request and to receive the result.

Parameters:

- String request: it contains the Keeper's matching request formatted in XML. The XML Schema and semantics of this request are reported in Appendix D.
- MatchListener listener: this object is in charge of receiving notifications when a response is available. The notification contains the matching output returned by the BMM.

Return: none

notifyMatchingFailure

```
void notifyMatchingFailure(String matchingFailureDescr)
```

It allows the Keeper to notify about deployment failure situation. Whenever the Keeper tries to deploy a package on a DHN and for any reason the deployment fails, the Keeper sends a feedback to the BMM by invoking this operation.

Parameters:

- String matchingFailureDescr: contains at least the pair <package, DHN> causing the failure and optionally other information

Return: none

5.2.3.3 Implementation Details

[to be completed in the next release]

5.2.3.4 Dependencies

[to be completed in the next release]

5.2.3.5 Configuration

[to be completed in the next release]

5.2.3.6 Known Bugs and Limitations

None

5.2.4 BMMService

The DILIGENT Broker & Matchmaker service is a WSRF-compliant Web Service implemented as specified by the Singleton Pattern [20]. This is the coordinator of the BMM. It contacts the DIS to retrieve information about packages. It asks the DISConnector about the DHNs status. It forwards the matching request to the Matchmaker component.

5.2.4.1 Profile

[to be completed in the next release]

5.2.4.2 Managed Resources

[to be completed in the next release]

5.2.4.3 Operations

The operations provided by the service are:

match

```
ID match(String request)
```

It allows submitting a matching request to the service. It returns an ID in order to connect the request to their response allowing clients to retrieve results.

Parameters:

- String request: a string which contains the request specification

Return:

ID: the UUID associated to this request

5.2.4.4 Implementation Details

[to be completed in the next release]

5.2.4.5 Dependencies

[to be completed in the next release]

5.2.4.6 Configuration

[to be completed in the next release]

5.2.4.7 Security Profile

[to be completed in the next release]

5.2.4.8 Known Bugs and Limitations

None

5.2.5 DISConnector

The DISConnector is a library available on DHNs, which will host the Broker & Matchmaker Service. It queries the DIS-HLSClient to gather information about the status of the DHNs.

5.2.5.1 Profile

[to be completed in the next release]

5.2.5.2 Operations

The operations provided by this library are:

getDHNProfiles

```
DHNStatus getDHNProfiles()
```

This method queries the DIS gathering DHNs status data. It parses and filters these data to properly translate the information for the Matchmaker algorithm.

Parameters: none

Return: DHNStatus, the status of DHNs

5.2.5.3 Implementation Details

[to be completed in the next release]

5.2.5.4 Dependencies

[to be completed in the next release]

5.2.5.5 Configuration

[to be completed in the next release]

5.2.5.6 Known Bugs and Limitations

None

5.2.6 Matchmaker

The Matchmaker is a library available on DHNs which hosts the Broker & Matchmaker Service. It implements the matching algorithm representing the core of the BMM Service. The service uses this library to elaborate the matching result.

5.2.6.1 Profile

[to be completed in the next release]

5.2.6.2 Operations

The operation provided by this library is:

findOptimalAllocation

`… findOptimalAllocation(…)`

It applies the matching algorithm to the request in order to calculate the right association among packages and DHNs. It needs the list of packages to be deployed, their requirements, all other constraints (specified in the request) and the information about DHNs status. It returns a list of tuples, where each tuple is in the <package, DHN> format.

Parameters: TBD

Return: TBD

5.2.6.3 Implementation Details

[to be completed in the next release]

5.2.6.4 Dependencies

[to be completed in the next release]

5.2.6.5 Configuration

[to be completed in the next release]

5.2.6.6 Known Bugs and Limitations

None

6 VDL GENERATOR SERVICE

6.1 Introduction

The VDL Generator Service is the service that enables users/communities to create their own DLs. It allows defining a set of criteria that specify the expected characteristics of the new DL; starting from them, it identifies the set of services required to provide the requested features. In particular, the VDL Generator Service:

- Supports users/communities in specifying the criteria that characterize the new DL, e.g. the information space, the required functionality, the QoS parameters;
- Selects the appropriate pool of services and information sources required to implement a DL that satisfies the specified criteria;
- Notifies to the DL Management service the identified services;
- Stores the definition criteria about created DLs for further usage.

This service is also characterized by a high interaction with a user via a graphical user interface. When specifying the architecture of such kind of interactive service, the challenge is to keep the functional core independent from the user interface. In fact, while the core is based on the functional requirements and usually remains stable, the user interface is often subject to changes and adaptation. This base design choice has consequences on the entire design of the VDL Generator Service.

6.2 Components

It is worth noting that the current release of this service is constituted only by the VDL Definition Repository component. The other components will be provided and thus described in a next revision of this report.

6.2.1 VDL Definition Repository

This package contains functionality needed to model the operations for storing, retrieving, and removing VDL Definitions. VDL Definitions are stored as files distributed on the storage elements available on the Grid infrastructure. In the first version of this service, VDL definitions are stored on local file system. This service is designed according to the WSRF Singleton pattern (we need only one resource for storing the Resource status). This component class diagram is shown in Figure 115.

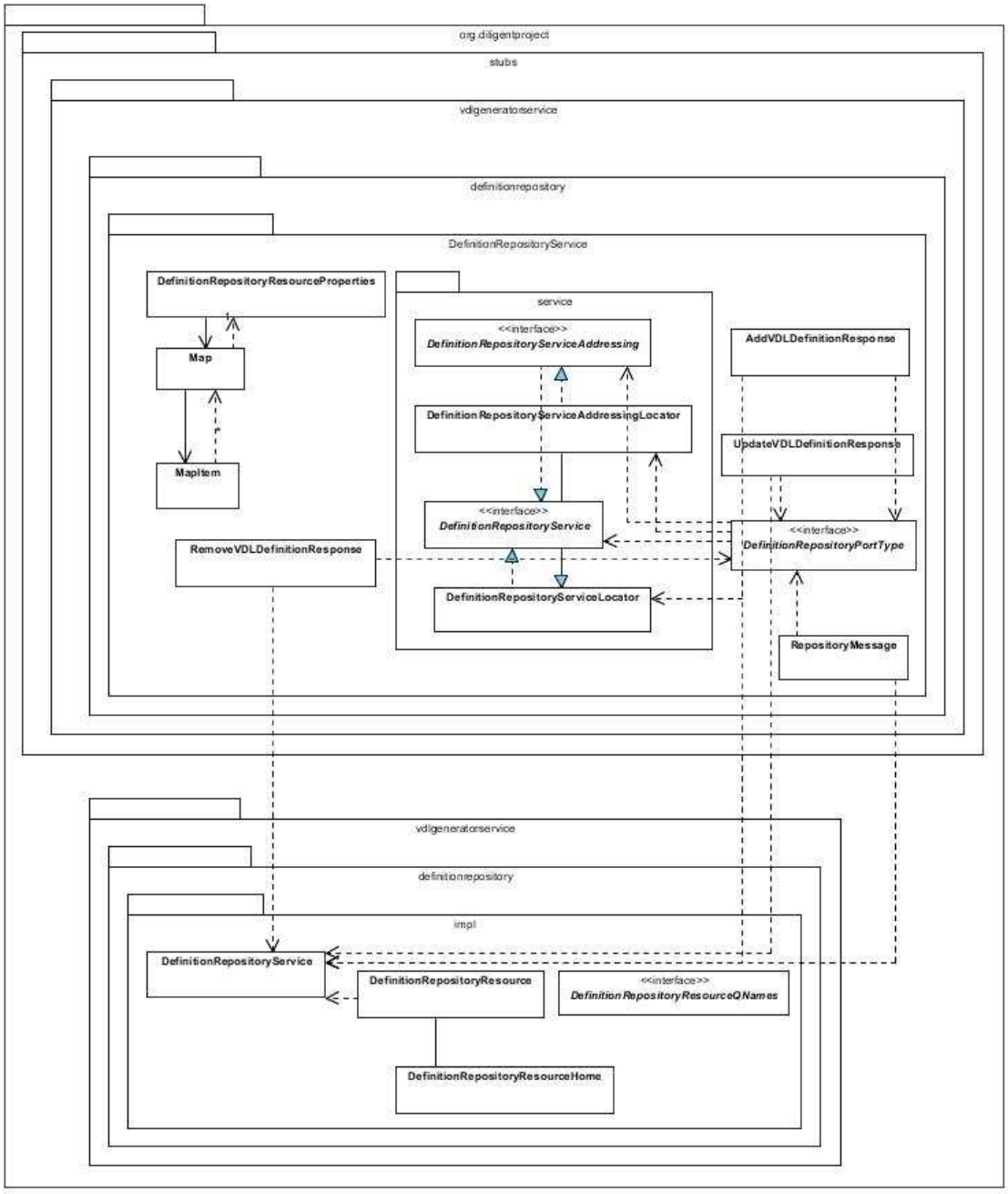

Figure 115. VDL Definition Repository class diagram

6.2.1.1 Profile

The XML profile of the VDLDefinitionRepository service is presented in Section A.18

6.2.1.2 Managed Resources

The WS-Resource-Property managed by this service is called DefintionDB. The QNames associated to this kind of WS-Resource are declared inside DefinitionRepositoryResourceQNames (see Figure 116). This Ws-Resource-Property is implemented as a HashTable that maps DefinitionID to definition file (seen as serialization string of an XML file)

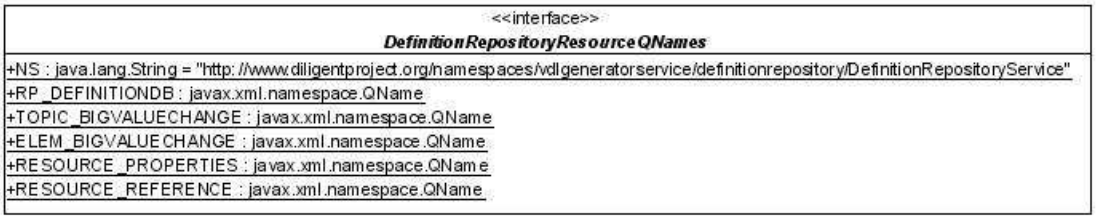

Figure 116. DefintionRepositoryResourceQNames class diagram

6.2.1.3 Operations

The operations, implemented inside the VDLDefintionRepositoryService class, are:

GetVDLDefinition

`String GetVDLDefinition(String DefintionID)`

This functionality represents the retrieval of a previously stored VDL Definition from the Definitions database.

Parameters:

- DefintionID: DefinitionID that is created by the VDLmanager at criteria definition time.

Return: a serialization string of the VDL definition XML file corresponding to the given DefintionID.

removeVDLDefinition

`RemoveVDLDefintionResponse removeVDLDefinition(String DefinitionID)`

This functionality represents the removal of a previously stored VDL Definition to the Definitions database

Parameters:

- String DefintionID: DefinitionID that is created by VDLmanager at criteria definition time.

Return: none.

updateVDLDefintion

`UpdateVDLDefintionResponse updateVDLDefintion(RepositoryMessage mess)`

This functionality represents the update of a previously stored VDL Definition to the Definitions database

Parameters:

- Repositorymessage mess: this bean is created at compilation time from VDLDefintionRepository WSDL file. It contains two strings, one is the DefintionID and the second is the definition file

Return: none

addVDLDefinition

`AddVDLDefinitionResponse addVDLDefinition(RepositoryMessage mess)`

This functionality represents the adjunction of a new VDL Definition to the Definitions database.

Parameters:

- RepositoryMessage mess: This bean is created at compilation time from the VDLDefintionRepository WSDL file. It contains two strings, one is the DefintionID and the second is the definition file

Return: none;

6.2.1.4 Implementation Details

6.2.1.4.1 Classes

The following classes belong to the package

`org.diligentproject.vdlgeneratorservice.definitionrepository.impl.`

DefinitionRepositoryService

This class implements the Web service part of this component (see Figure 117). It is in charge of storing, uploading, deleting, and adding definition to the Definition database. In the first version of this service, the DB is a directory stored in this path:

```
ContainerConfig.getBaseDirectory() +
/etc/org_diligentproject_vdlgeneratorservice_defintionrepository/DB/.
```

Every new definition is stored in one file having DefinitionID as name.

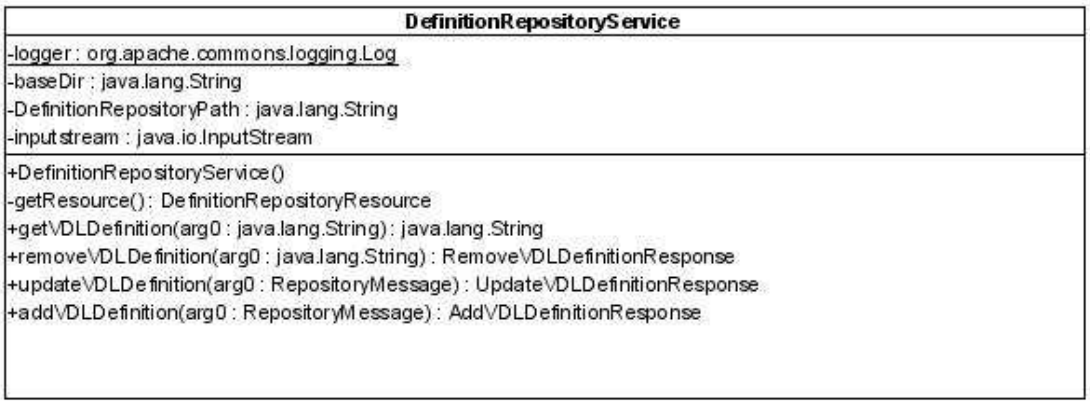

Figure 117. DefinitionRepositoryService class diagram

DefinitionRepositoryResource

This class implements the DefinitionRepositoryResource (see Figure 118). There, the definitionDB resource properties have been declared which are in charge of maintaining DB statefulness.

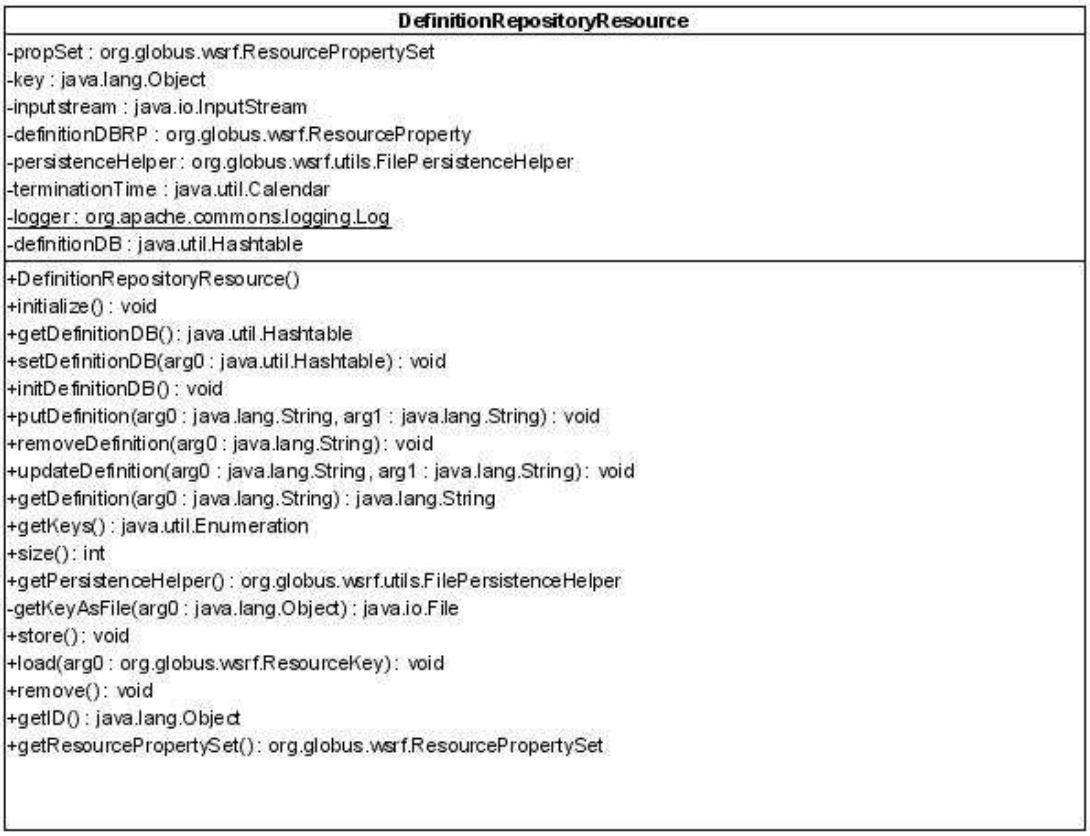

Figure 118. DefinitionRepositorResource class diagram

DefinitionRepositoryResourceHome

This class is an implementation of SingletonResourceHome (because this service reflects Singleton pattern) and exposes only a `findSingleton` method (see Figure 119).

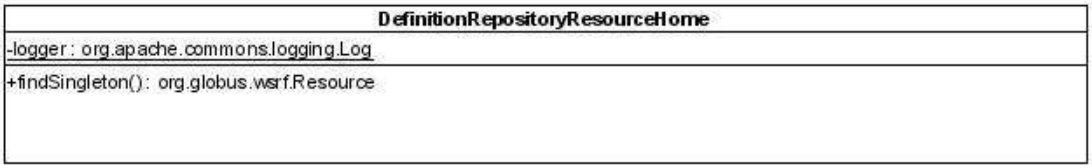

Figure 119. DefitionRepositorResourceHome class diagram

6.2.1.4.2 Stubs

The following stubs belong to the package

`org.diligentproject.stubs.vdlgeneratorservice.definitionrepository.DefinitionRepositoryService`

RepositoryMessage

This class (see Figure 120) is used in add and delete operations. It contains two strings: *definitionId* and *definitionFile*.

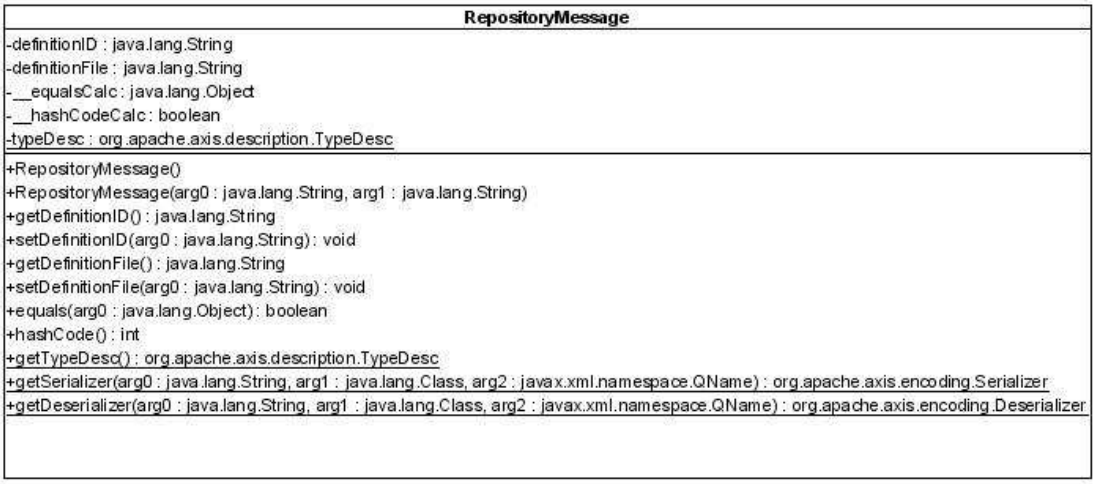

Figure 120. RepositoryMessage class diagram

DefinitionRepositoryPortType

This class (see Figure 121), generated automatically from WSDL definitions, exposes client-side methods for service calls.

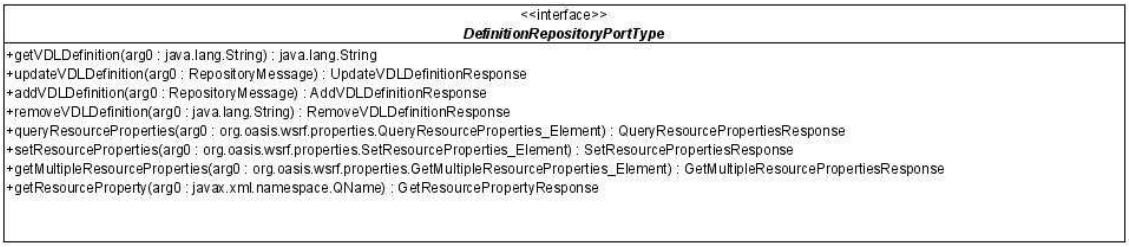

Figure 121. DefinitionRepositoryPortTye class diagram

7 CONCLUSION

This report focuses on the detailed design of the services constituting the DL Creation and Management area of the DILIGENT project. In particular, presenting the DL Creation and Management components with a higher level of detail, it acts as the blueprint for the DILIGENT software developers. The identified components and the related characteristics are integrated with the specification of the interfaces, the algorithms, the data structures, and the data flows presented here.

Due to the "on-going" nature of this document, the document reflects the current Alpha release implementation of the software and does not include all features that will be supported in the future Beta release.

Appendix A. DL Creation & Management Profiles

A.1. DIS-IC Service

```
<?xml version="1.0" encoding="UTF-8"?>
<DILIGENTResource xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <UniqueID>
        <!-- to be assigned by the RegistrationService-->
    </UniqueID>
    <ResourceType>Service</ResourceType>
    <AuthorizationPolicies/>
    <Profile>
        <Class>InformationSystem</Class>
        <Name>DIS-IC</Name>
        <DescriptiveParameters>
            <DescParameter/>
        </DescriptiveParameters>
        <QoS/>
        <DLDependencies/>
        <SpecificData/>
        <PackagesList>
            <Package>
                <PackageName>DIS-IC</PackageName>
                <PackageType>WSRFService</PackageType>
                <Version>1.0</Version>
                <DLMandatory value="1"/>
                <DHNMandatory value="0"/>
                <VOMandatory value="1"/>
                <DisposeInterfaceSupport value="1"/>
                <MultiVersionSupport value="0"/>
                <VOSharingSupport value="1"/>
                <ManifestFile/>
                <InstallScripts/>
                <UninstallScripts/>
                <Dependencies>
                    <Dependency>
                        <Service>
                            <Class>InformationSystem</Class>
                            <Name>DIS-IC</Name>
                        </Service>
                        <PackageName>AggregatorFramework</PackageName>
                        <SameDHN value="1"/>
                        <Priority>1</Priority>
                    </Dependency>
                </Dependencies>
                <DHNRequirements>
                    <Req category="RunTimeEnv" requirement="Variable" value="eXist1.0" operator="eq"/>
                    <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="java1.5" />
                    <Req category="OperatingSystem" operator="eq" requirement="Name" value="Linux" />
                </DHNRequirements>
                <ConfigurationFiles/>
                <WSRFService>
                    <GARArchive>org_diligentproject_informationservice_disic.gar</GARArchive>
                    <BuildFile>build.xml</BuildFile>
                    <DeploymentOptions/>
                    <ArchitecturalPattern>Multiton</ArchitecturalPattern>
                    <WSRFEntry>
                        <EntryName>diligentproject/informationservice/disic/DISICRegistrationServiceEntry</EntryName>
                        <Factory value="0"/>
                        <Parameters/>
                        <Security name="">
                            <securityDescriptor/>
                            <defaultIdentity>
                                <subject/>
                                <CASubject/>
                            </defaultIdentity>
                        </Security>
                        <WSDL>
```

```
                    <wsdl:definitions targetNamespace="http://www.xmlspy.com" name="NCName">
                        <wsdl:documentation>Text</wsdl:documentation>
                        <wsdl:import namespace="http://www.xmlspy.com" location="http://www.xmlspy.com">
                            <wsdl:documentation>Text</wsdl:documentation>
                        </wsdl:import>
                    </wsdl:definitions>
                </WSDL>
            </WSRFEntry>
            <WSRFEntry>
                <EntryName>diligentproject/informationservice/disic/DISICService</EntryName>
                <Factory value="0"/>
                <Parameters/>
                <Security name="">
                    <securityDescriptor/>
                    <defaultIdentity>
                        <subject/>
                        <CASubject/>
                    </defaultIdentity>
                </Security>
                <WSDL>
                    <wsdl:definitions targetNamespace="http://www.xmlspy.com" name="NCName">
                        <wsdl:documentation>Text</wsdl:documentation>
                        <wsdl:import namespace="http://www.xmlspy.com" location="http://www.xmlspy.com">
                            <wsdl:documentation>Text</wsdl:documentation>
                        </wsdl:import>
                    </wsdl:definitions>
                </WSDL>
            </WSRFEntry>
            <WSRFEntry>
                <EntryName>diligentproject/informationservice/disic/DISICFactoryService</EntryName>
                <Factory value="1"/>
                <Parameters/>
                <Security name="">
                    <securityDescriptor/>
                    <defaultIdentity>
                        <subject/>
                        <CASubject/>
                    </defaultIdentity>
                </Security>
                <WSDL>
                    <wsdl:definitions targetNamespace="http://www.xmlspy.com" name="NCName">
                        <wsdl:documentation>Text</wsdl:documentation>
                        <wsdl:import namespace="http://www.xmlspy.com" location="http://www.xmlspy.com">
                            <wsdl:documentation>Text</wsdl:documentation>
                        </wsdl:import>
                    </wsdl:definitions>
                </WSDL>
            </WSRFEntry>
            <WSRFEntry>
                <EntryName>diligentproject/informationservice/disic/DISICRegistrationService</EntryName>
                <Factory value="1"/>
                <Parameters/>
                <Security name="">
                    <securityDescriptor/>
                    <defaultIdentity>
                        <subject/>
                        <CASubject/>
                    </defaultIdentity>
                </Security>
                <WSDL>
                    <wsdl:definitions targetNamespace="http://www.xmlspy.com" name="NCName">
                        <wsdl:documentation>Text</wsdl:documentation>
                        <wsdl:import namespace="http://www.xmlspy.com" location="http://www.xmlspy.com">
                            <wsdl:documentation>Text</wsdl:documentation>
                        </wsdl:import>
                    </wsdl:definitions>
                </WSDL>
            </WSRFEntry>
        </WSRFService>
```

```
        <OtherFiles/>
        <OtherProperties/>
    </Package>
    <Package>
        <PackageName>DIS-IC_stubs</PackageName>
        <PackageType>Library</PackageType>
        <Version>1.0</Version>
        <DLMandatory value="0"/>
        <DHNMandatory value="1"/>
        <VOMandatory value="0"/>
        <DisposeInterfaceSupport value="0"/>
        <MultiVersionSupport value="0"/>
        <VOSharingSupport value="1"/>
        <ManifestFile/>
        <InstallScripts/>
        <UninstallScripts/>
        <Dependencies>
            <Dependency>
                <Service>
                    <Class>InformationSystem</Class>
                    <Name>DIS-IC</Name>
                </Service>
                <PackageName>AggregatorFramework</PackageName>
                <SameDHN value="1"/>
                <Priority>3</Priority>
            </Dependency>
            <Dependency>
                <Service>
                    <Class>InformationSystem</Class>
                    <Name>DIS-IC</Name>
                </Service>
                <PackageName>DIS-IC</PackageName>
                <SameDHN value="0"/>
                <SameDL value="1"/>
                <Priority>3</Priority>
            </Dependency>
        </Dependencies>
        <DHNRequirements>
            <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="java1.5" />
            <Req category="OperatingSystem" operator="eq" requirement="Name" value="Linux" />
        </DHNRequirements>
        <ConfigurationFiles/>
        <Library>
            <Type>stub</Type>
            <IsStubOf>
                <PackageName>DIS-IC</PackageName>
                <Service>
                    <Class>InformationSystem</Class>
                    <Name>DIS-IC</Name>
                </Service>
            </IsStubOf>
            <LibraryFile>org_diligentproject_informationservice_disic_stubs.jar</LibraryFile>
        </Library>
        <OtherFiles/>
        <OtherProperties/>
    </Package>
    <Package>
        <PackageName>DIS-HLSClient</PackageName>
        <PackageType>Library</PackageType>
        <Version>1.0</Version>
        <DLMandatory value="0"/>
        <DHNMandatory value="1"/>
        <VOMandatory value="0"/>
        <DisposeInterfaceSupport value="0"/>
        <MultiVersionSupport value="0"/>
        <VOSharingSupport value="1"/>
        <ManifestFile/>
        <InstallScripts/>
        <UninstallScripts/>
```

```
            <Dependencies>
                <Dependency>
                    <Service>
                        <Class>InformationSystem</Class>
                        <Name>DIS-IC</Name>
                    </Service>
                    <PackageName>AggregatorFramework</PackageName>
                    <SameDHN value="1"/>
                    <Priority>1</Priority>
                </Dependency>
                <Dependency>
                    <Service>
                        <Class>InformationSystem</Class>
                        <Name>DIS-IC</Name>
                    </Service>
                    <PackageName>DIS-IC_stubs</PackageName>
                    <SameDHN value="1"/>
                    <Priority>3</Priority>
                </Dependency>
                <Dependency>
                    <Service>
                        <Class>InformationSystem</Class>
                        <Name>DIS-IC</Name>
                    </Service>
                    <PackageName>DIS-IC</PackageName>
                    <SameDHN value="0"/>
                    <SameDL value="1"/>
                    <Priority>1</Priority>
                </Dependency>
            </Dependencies>
            <DHNRequirements>
                 <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="java1.5" />
                 <Req category="OperatingSystem" operator="eq" requirement="Name" value="Linux" />
            </DHNRequirements>
            <ConfigurationFiles/>
            <Library>
                <Type>shared</Type>
                <LibraryFile>org_diligentproject_informationservice_dishlsclient.jar</LibraryFile>
            </Library>
            <OtherFiles/>
            <OtherProperties/>
        </Package>
        <Package>
            <PackageName>DIS-IP</PackageName>
            <PackageType>Library</PackageType>
            <Version>1.0</Version>
            <DLMandatory value="0"/>
            <DHNMandatory value="1"/>
            <VOMandatory value="0"/>
            <DisposeInterfaceSupport value="0"/>
            <MultiVersionSupport value="0"/>
            <VOSharingSupport value="1"/>
            <ManifestFile/>
            <InstallScripts/>
            <UninstallScripts/>
            <Dependencies>
                <Dependency>
                    <Service>
                        <Class>InformationSystem</Class>
                        <Name>DIS-IC</Name>
                    </Service>
                    <PackageName>AggregatorFramework</PackageName>
                    <SameDHN value="1"/>
                    <Priority>3</Priority>
                </Dependency>
                <Dependency>
                    <Service>
                        <Class>InformationSystem</Class>
                        <Name>DIS-IC</Name>
```

```
                    </Service>
                    <PackageName>DIS-IC</PackageName>
                    <SameDHN value="0"/>
                    <SameDL value="1"/>
                    <Priority>3</Priority>
                </Dependency>
            </Dependencies>
            <DHNRequirements>
                <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="java1.5" />
                <Req category="OperatingSystem" operator="eq" requirement="Name" value="Linux" />
            </DHNRequirements>
            <ConfigurationFiles/>
            <Library>
                <Type>shared</Type>
                <LibraryFile>org_diligentproject_informationservice_disip.jar</LibraryFile>
            </Library>
            <OtherFiles/>
            <OtherProperties/>
        </Package>
        <Package>
            <PackageName>DILIGENTProvider</PackageName>
            <PackageType>Library</PackageType>
            <Version>1.0</Version>
            <DLMandatory value="0"/>
            <DHNMandatory value="1"/>
            <VOMandatory value="0"/>
            <DisposeInterfaceSupport value="0"/>
            <MultiVersionSupport value="0"/>
            <VOSharingSupport value="1"/>
            <ManifestFile/>
            <InstallScripts/>
            <UninstallScripts/>
            <Dependencies/>
            <DHNRequirements>
                <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="java1.5" />
                <Req category="OperatingSystem" operator="eq" requirement="Name" value="Linux" />
            </DHNRequirements>
            <ConfigurationFiles/>
            <Library>
                <Type>shared</Type>
                <LibraryFile>org.diligentproject.common.diligentprovider.gar</LibraryFile>
            </Library>
            <OtherFiles/>
            <OtherProperties/>
        </Package>
        <Package>
            <PackageName>AggregatorFramework</PackageName>
            <PackageType>Software</PackageType>
            <Version>1.0</Version>
            <DLMandatory value="0"/>
            <DHNMandatory value="1"/>
            <VOMandatory value="0"/>
            <DisposeInterfaceSupport value="0"/>
            <MultiVersionSupport value="0"/>
            <ManifestFile/>
            <InstallScripts>
                <File>install.sh</File>
            </InstallScripts>
            <UninstallScripts/>
            <ConfigurationFiles/>
            <Software>
                <Files>
                    <File>aggregator_framework.tar.gz</File>
                </Files>
            </Software>
            <OtherFiles/>
            <OtherProperties/>
        </Package>
    </PackagesList>
```

```
    </Profile>
</DILIGENTResource>
```

A.2. DIS-R-GMAClient Service

```
<DILIGENTResource xmlns:default="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsdlpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor" xmlns:wsrp="http://docs.oasis-
open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.xsd">
   <UniqueID>bc239c10-1bc1-11db-81a5-bc0e2101bb14</UniqueID>
   <ResourceType>Service</ResourceType>
   <AuthorizationPolicies/>
   <Profile>
      <Class>InformationSystem</Class>
      <Name>DIS-R-GMA-Client</Name>
      <DescriptiveParameters>
         <DescParameter/>
      </DescriptiveParameters>
      <QoS/>
      <DLDependencies>
         <DLComponent>
            <Class>InformationSystem</Class>
            <Name>DIS-IC</Name>
            <DescriptiveParametersValue/>
         </DLComponent>
         <DLComponent>
            <Class>InformationSystem</Class>
            <Name>DIS-Broker</Name>
            <DescriptiveParametersValue/>
         </DLComponent>
         <DLComponent>
            <Class>DVOS</Class>
            <Name>RegistrationService</Name>
            <DescriptiveParametersValue/>
         </DLComponent>
         <DLComponent>
            <Class>DVOS</Class>
            <Name>UnRegistrationService</Name>
            <DescriptiveParametersValue/>
         </DLComponent>
      </DLDependencies>
      <SpecificData>Text</SpecificData>
      <PackagesList>
         <Package>
            <PackageName>DIS-R-GMAClient_Service</PackageName>
            <PackageType>WSRFService</PackageType>
            <Version>0.2</Version>
            <DLMandatory value="0"/>
            <DHNMandatory value="0"/>
            <VOMandatory value="1"/>
            <DisposeInterfaceSupport value="1"/>
            <MultiVersionSupport value="0"/>
            <VOSharingSupport value="0"/>
            <ManifestFile/>
            <InstallScripts/>
            <UninstallScripts/>
            <RebootScripts/>
            <Dependencies>
               <Dependency>
                  <Service>
                     <Class>InformationSystem</Class>
                     <Name>DIS-IC</Name>
                  </Service>
                  <PackageName>DIS-IP</PackageName>
                  <SameDHN value="1"/>
                  <SameDL value="0"/>
                  <SameVO value="0"/>
                  <Priority>1</Priority>
               </Dependency>
               <Dependency>
                  <Service>
```

```
            <Class>InformationSystem</Class>
            <Name>DIS-IC</Name>
         </Service>
         <PackageName>DILIGENTProvider</PackageName>
         <SameDHN value="1"/>
         <SameDL value="0"/>
         <SameVO value="0"/>
         <Priority>1</Priority>
      </Dependency>
      <Dependency>
         <Service>
            <Class>InformationSystem</Class>
            <Name>DIS-IC</Name>
         </Service>
         <PackageName>DIS-HLS-Client</PackageName>
         <SameDHN value="1"/>
         <SameDL value="0"/>
         <SameVO value="0"/>
         <Priority>1</Priority>
      </Dependency>
      <Dependency>
         <Service>
            <Class>InformationSystem</Class>
            <Name>DIS-IC</Name>
         </Service>
         <PackageName>DIS-Utils</PackageName>
         <SameDHN value="1"/>
         <SameDL value="0"/>
         <SameVO value="0"/>
         <Priority>1</Priority>
      </Dependency>
      <Dependency>
         <Service>
            <Class>InformationSystem</Class>
            <Name>DIS-Broker</Name>
         </Service>
         <PackageName>DIS-Broker_stubs</PackageName>
         <SameDHN value="1"/>
         <SameDL value="0"/>
         <SameVO value="0"/>
         <Priority>1</Priority>
      </Dependency>
      <Dependency>
         <Service>
            <Class>InformationSystem</Class>
            <Name>DIS-IC</Name>
         </Service>
         <PackageName>Common-Profile</PackageName>
         <SameDHN value="1"/>
         <SameDL value="0"/>
         <SameVO value="0"/>
         <Priority>1</Priority>
      </Dependency>
      <Dependency>
         <Service>
            <Class>DVOS</Class>
            <Name>RegistrationService</Name>
         </Service>
         <PackageName>RegistrationAPI</PackageName>
         <SameDHN value="1"/>
         <SameDL value="0"/>
         <SameVO value="0"/>
         <Priority>1</Priority>
      </Dependency>
      <Dependency>
         <Service>
            <Class>DVOS</Class>
            <Name>UnRegistrationService</Name>
         </Service>
```

```
            <PackageName>UnRegistrationAPI</PackageName>
            <SameDHN value="1"/>
            <SameDL value="0"/>
            <SameVO value="0"/>
            <Priority>1</Priority>
          </Dependency>
          <Dependency>
            <Service>
              <Class>DVOS</Class>
              <Name>RegistrationService</Name>
            </Service>
            <PackageName>DVOS-Common</PackageName>
            <SameDHN value="1"/>
            <SameDL value="0"/>
            <SameVO value="0"/>
            <Priority>1</Priority>
          </Dependency>
        </Dependencies>
        <DHNRequirements>
          <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="java1.5"/>
          <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="glite-rgma-api-java"/>
          <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="glite-rgma-stubs-servlet-java"/>
          <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="glite-security-delegation-java"/>
          <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="glite-security-util-java"/>
          <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="glite-security-trustmanager"/>
        </DHNRequirements>
        <ConfigurationFiles/>
        <WSRFService>
          <GARArchive>org_diligentproject_informationservice_disrgmaclient.gar</GARArchive>
          <BuildFile>build.xml</BuildFile>
          <DeploymentOptions/>
          <ArchitecturalPattern>Factory</ArchitecturalPattern>
          <WSRFEntry>
            <EntryName>diligentproject/informationservice/disrgmaclient/DISRGMAClientFactoryService</EntryName>
            <Factory value="true"/>
            <Parameters/>
            <WSDL>
              <default:definitions
xmlns:tns="http://diligentproject.org/namespaces/informationservice/disrgmaclient/DISRGMAClientFactoryService"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/" name="DISRGMAClientFactory"
targetNamespace="http://diligentproject.org/namespaces/informationservice/disrgmaclient/DISRGMAClientFactoryService">
                <default:types>
                  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://diligentproject.org/namespaces/informationservice/disrgmaclient/DISRGMAClientFactoryService">
                    <xsd:import namespace="http://schemas.xmlsoap.org/ws/2004/03/addressing"
schemaLocation="../../../ws/addressing/WS-Addressing.xsd"/>
                    <xsd:element name="createResource" type="xsd:string"/>
                    <xsd:element name="createResourceResponse">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element ref="wsa:EndpointReference"/>
                        </xsd:sequence>
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:schema>
                </default:types>
                <default:message name="CreateResourceRequest">
                  <default:part element="tns:createResource" name="request"/>
                </default:message>
                <default:message name="CreateResourceResponse">
                  <default:part element="tns:createResourceResponse" name="response"/>
                </default:message>
                <default:portType name="DISRGMAClientFactoryPortType">
                  <default:operation name="createResource">
                    <default:input message="tns:CreateResourceRequest"/>
                    <default:output message="tns:CreateResourceResponse"/>
                  </default:operation>
                </default:portType>
              </default:definitions>
```

```
            </WSDL>
         </WSRFEntry>
         <WSRFEntry>
            <EntryName>diligentproject/informationservice/disrgmaclient/DISRGMAClientCE</EntryName>
            <Factory value="false"/>
            <Parameters/>
            <WSDL>
               <default:definitions xmlns:wsrp="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-
draft-01.xsd" xmlns:tns="http://diligentproject.org/namespaces/informationservice/disrgmaclient/DISRGMAClientCE"
xmlns:wsrlw="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.wsdl"
xmlns:wsrpw="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.wsdl"
xmlns:wsdlpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor" xmlns:wsntw="http://docs.oasis-
open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.wsdl" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/" name="DISRGMAClientCE"
targetNamespace="http://diligentproject.org/namespaces/informationservice/disrgmaclient/DISRGMAClientCE">
                  <default:import location="../../../wsrf/properties/WS-ResourceProperties.wsdl"
namespace="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.wsdl"/>
                  <default:import location="../../../wsrf/lifetime/WS-ResourceLifetime.wsdl" namespace="http://docs.oasis-
open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.wsdl"/>
                  <default:import location="../../../wsrf/notification/WS-BaseN.wsdl" namespace="http://docs.oasis-
open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.wsdl"/>
                  <default:types>
                     <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://diligentproject.org/namespaces/informationservice/disrgmaclient/DISRGMAClientCE">
                        <xsd:element name="UniqueID" type="xsd:string"/>
                        <xsd:element name="RunningJobs" type="xsd:string"/>
                        <xsd:element name="WaitingJobs" type="xsd:string"/>
                        <xsd:element name="TotalJobs" type="xsd:string"/>
                        <xsd:element name="EstimatedResponseTime" type="xsd:string"/>
                        <xsd:element name="WorstResponseTime" type="xsd:string"/>
                        <xsd:element name="FreeJobsSlot" type="xsd:string"/>
                        <xsd:element name="MaxWallClockTime" type="xsd:string"/>
                        <xsd:element name="MaxCPUTime" type="xsd:string"/>
                        <xsd:element name="MaxTotalJobs" type="xsd:string"/>
                        <xsd:element name="MaxRunningJobs" type="xsd:string"/>
                        <xsd:element name="AssignedJobSlots" type="xsd:string"/>
                        <xsd:element name="Priority" type="xsd:string"/>
                        <xsd:element name="gLiteCEResourceProperties">
                           <xsd:complexType>
                              <xsd:sequence>
                                 <xsd:element maxOccurs="1" minOccurs="1" ref="tns:UniqueID"/>
                                 <xsd:element maxOccurs="1" minOccurs="1" ref="tns:RunningJobs"/>
                                 <xsd:element maxOccurs="1" minOccurs="1" ref="tns:WaitingJobs"/>
                                 <xsd:element maxOccurs="1" minOccurs="1" ref="tns:TotalJobs"/>
                                 <xsd:element maxOccurs="1" minOccurs="1" ref="tns:EstimatedResponseTime"/>
                                 <xsd:element maxOccurs="1" minOccurs="1" ref="tns:WorstResponseTime"/>
                                 <xsd:element maxOccurs="1" minOccurs="1" ref="tns:FreeJobsSlot"/>
                                 <xsd:element maxOccurs="1" minOccurs="1" ref="tns:MaxWallClockTime"/>
                                 <xsd:element maxOccurs="1" minOccurs="1" ref="tns:MaxCPUTime"/>
                                 <xsd:element maxOccurs="1" minOccurs="1" ref="tns:MaxTotalJobs"/>
                                 <xsd:element maxOccurs="1" minOccurs="1" ref="tns:MaxRunningJobs"/>
                                 <xsd:element maxOccurs="1" minOccurs="1" ref="tns:AssignedJobSlots"/>
                                 <xsd:element maxOccurs="1" minOccurs="1" ref="tns:Priority"/>
                              </xsd:sequence>
                           </xsd:complexType>
                        </xsd:element>
                     </xsd:schema>
                  </default:types>
                  <default:portType name="DISRGMAClientCEPortType" wsdlpp:extends="wsrpw:GetResourceProperty
wsrpw:GetMultipleResourceProperties wsrpw:SetResourceProperties
wsrpw:QueryResourceProperties wsrlw:ImmediateResourceTermination wsntw:NotificationProducer"
wsrp:ResourceProperties="tns:gLiteCEResourceProperties"/>
               </default:definitions>
            </WSDL>
         </WSRFEntry>
         <WSRFEntry>
            <EntryName>diligentproject/informationservice/disrgmaclient/DISRGMAClientSE</EntryName>
            <Factory value="false"/>
            <Parameters/>
```

```
<WSDL>
<default:definitions xmlns:wsrp="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.xsd" xmlns:tns="http://diligentproject.org/namespaces/informationservice/disrgmaclient/DISRGMAClientSE" xmlns:wsrlw="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.wsdl" xmlns:wsrpw="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.wsdl" xmlns:wsdlpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor" xmlns:wsntw="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.wsdl" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/" name="DISRGMAClientSE" targetNamespace="http://diligentproject.org/namespaces/informationservice/disrgmaclient/DISRGMAClientSE">
<default:import location="../../../wsrf/properties/WS-ResourceProperties.wsdl" namespace="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.wsdl"/>
<default:import location="../../../wsrf/lifetime/WS-ResourceLifetime.wsdl" namespace="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.wsdl"/>
<default:import location="../../../wsrf/notification/WS-BaseN.wsdl" namespace="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.wsdl"/>
<default:types>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://diligentproject.org/namespaces/informationservice/disrgmaclient/DISRGMAClientSE">
<xsd:element name="UniqueID" type="xsd:string"/>
<xsd:element name="SizeFree" type="xsd:string"/>
<xsd:element name="UsedSpace" type="xsd:string"/>
<xsd:element name="AvailableSpace" type="xsd:string"/>
<xsd:element name="gLiteSEResourceProperties">
<xsd:complexType>
<xsd:sequence>
<xsd:element maxOccurs="1" minOccurs="1" ref="tns:UniqueID"/>
<xsd:element maxOccurs="1" minOccurs="1" ref="tns:SizeFree"/>
<xsd:element maxOccurs="1" minOccurs="1" ref="tns:UsedSpace"/>
<xsd:element maxOccurs="1" minOccurs="1" ref="tns:AvailableSpace"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
</default:types>
<default:portType name="DISRGMAClientSEPortType" wsdlpp:extends="wsrpw:GetResourceProperty wsrpw:GetMultipleResourceProperties wsrpw:SetResourceProperties wsrpw:QueryResourceProperties wsrlw:ImmediateResourceTermination wsntw:NotificationProducer" wsrp:ResourceProperties="tns:gLiteSEResourceProperties"/>
</default:definitions>
</WSDL>
</WSRFEntry>
<WSRFEntry>
<EntryName>diligentproject/informationservice/disrgmaclient/DISRGMAClientService</EntryName>
<Factory value="false"/>
<Parameters/>
<WSDL>
<default:definitions xmlns:wsrp="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.xsd" xmlns:tns="http://diligentproject.org/namespaces/informationservice/disrgmaclient/DISRGMAClientService" xmlns:wsrlw="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.wsdl" xmlns:wsrpw="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.wsdl" xmlns:wsdlpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor" xmlns:wsntw="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.wsdl" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/" name="DISRGMAClientService" targetNamespace="http://diligentproject.org/namespaces/informationservice/disrgmaclient/DISRGMAClientService">
<default:import location="../../../wsrf/properties/WS-ResourceProperties.wsdl" namespace="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.wsdl"/>
<default:import location="../../../wsrf/lifetime/WS-ResourceLifetime.wsdl" namespace="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.wsdl"/>
<default:import location="../../../wsrf/notification/WS-BaseN.wsdl" namespace="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.wsdl"/>
<default:types>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://diligentproject.org/namespaces/informationservice/disrgmaclient/DISRGMAClientService">
<xsd:element name="UniqueID" type="xsd:string"/>
<xsd:element name="Status" type="xsd:string"/>
<xsd:element name="StatusInfo" type="xsd:string"/>
<xsd:element name="gLiteServiceResourceProperties">
<xsd:complexType>
<xsd:sequence>
```

```
                                <xsd:element maxOccurs="1" minOccurs="1" ref="tns:UniqueID"/>
                                <xsd:element maxOccurs="1" minOccurs="1" ref="tns:Status"/>
                                <xsd:element maxOccurs="1" minOccurs="1" ref="tns:StatusInfo"/>
                            </xsd:sequence>
                        </xsd:complexType>
                    </xsd:element>
                </xsd:schema>
            </default:types>
            <default:portType name="DISRGMAClientServicePortType"
wsdlpp:extends="wsrpw:GetResourceProperty wsrpw:GetMultipleResourceProperties
wsrpw:SetResourceProperties wsrpw:QueryResourceProperties
wsrlw:ImmediateResourceTermination wsntw:NotificationProducer"
wsrp:ResourceProperties="tns:gLiteServiceResourceProperties"/>
        </default:definitions>
      </WSDL>
    </WSRFEntry>
  </WSRFService>
</Package>
<Package>
  <PackageName>DIS-R-GMAClient_stubs</PackageName>
  <PackageType>Library</PackageType>
  <Version>0.2</Version>
  <DLMandatory value="0"/>
  <DHNMandatory value="0"/>
  <VOMandatory value="1"/>
  <DisposeInterfaceSupport value="0"/>
  <MultiVersionSupport value="0"/>
  <VOSharingSupport value="0"/>
  <ManifestFile/>
  <InstallScripts/>
  <UninstallScripts/>
  <RebootScripts/>
  <Dependencies/>
  <DHNRequirements>
    <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="java1.5"/>
    <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="glite-rgma-api-java"/>
    <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="glite-rgma-stubs-servlet-java"/>
    <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="glite-security-delegation-java"/>
    <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="glite-security-util-java"/>
    <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="glite-security-trustmanager"/>
  </DHNRequirements>
  <Library>
    <Type>stub</Type>
    <IsStubOf>
      <PackageName>DIS-R-GMA-Client_Service</PackageName>
      <Service>
        <Class>InformationSystem</Class>
        <Name>DIS-R-GMA-Client</Name>
      </Service>
    </IsStubOf>
    <Parameters/>
    <LibraryFile>org_diligentproject_informationservice_disrgmaclient_stubs.jar</LibraryFile>
  </Library>
  <OtherFiles/>
  <OtherProperties/>
</Package>
<Package>
  <PackageName>JAXB_software</PackageName>
  <PackageType>Software</PackageType>
  <Version>2.0</Version>
  <DLMandatory value="0"/>
  <DHNMandatory value="1"/>
  <VOMandatory value="0"/>
  <DisposeInterfaceSupport value="0"/>
  <MultiVersionSupport value="0"/>
  <VOSharingSupport value="0"/>
  <ManifestFile/>
  <InstallScripts/>
  <UninstallScripts/>
```

```
            <RebootScripts/>
            <Dependencies/>
            <DHNRequirements>
               <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="java1.5"/>
            </DHNRequirements>
            <Software>
               <Files>
                  <File>jaxb1-impl.jar</File>
                  <File>jaxb-api.jar</File>
                  <File>jaxb-impl.jar</File>
                  <File>jaxb-xjc.jar</File>
                  <File>jsr173_api.jar</File>
               </Files>
            </Software>
            <OtherFiles/>
            <OtherProperties/>
         </Package>
      </PackagesList>
   </Profile>
</DILIGENTResource>
```

A.3. DIS-Registry Service

```
<?xml version="1.0" encoding="UTF-8"?>
<DILIGENTResource xmlns:default="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsdlpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor" xmlns:wsrp="http://docs.oasis-
open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.xsd">
   <UniqueID>gfgdf80-131b-11db-90fgf98-dgfgfgf66ed3f</UniqueID>
   <ResourceType>Service</ResourceType>
   <AuthorizationPolicies/>
   <Profile>
      <Class>InformationSystem</Class>
      <Name>DIS-Registry</Name>
      <DescriptiveParameters>
         <DescParameter/>
      </DescriptiveParameters>
      <QoS/>
      <DLDependencies>
         <DLComponent>
            <Class>InformationSystem</Class>
            <Name>DIS-IC</Name>
            <DescriptiveParametersValue/>
         </DLComponent>
         <DLComponent>
            <Class>InformationSystem</Class>
            <Name>DIS-Broker</Name>
            <DescriptiveParametersValue/>
         </DLComponent>
      </DLDependencies>
      <SpecificData>Text</SpecificData>
      <PackagesList>
         <Package>
            <PackageName>DIS-Registry_Service</PackageName>
            <PackageType>WSRFService</PackageType>
            <Version>0.2</Version>
            <DLMandatory value="0"/>
            <DHNMandatory value="0"/>
            <VOMandatory value="1"/>
            <DisposeInterfaceSupport value="1"/>
            <MultiVersionSupport value="0"/>
            <VOSharingSupport value="0"/>
            <ManifestFile/>
            <InstallScripts/>
            <UninstallScripts/>
            <RebootScripts/>
            <Dependencies>
               <Dependency>
                  <Service>
```

```
                <Class>InformationSystem</Class>
                <Name>DIS-IC</Name>
              </Service>
              <PackageName>DIS-IP</PackageName>
              <SameDHN value="1"/>
              <SameDL value="0"/>
              <SameVO value="0"/>
              <Priority>1</Priority>
            </Dependency>
            <Dependency>
              <Service>
                <Class>InformationSystem</Class>
                <Name>DIS-IC</Name>
              </Service>
              <PackageName>DILIGENTProvider</PackageName>
              <SameDHN value="1"/>
              <SameDL value="0"/>
              <SameVO value="0"/>
              <Priority>1</Priority>
            </Dependency>
            <Dependency>
              <Service>
                <Class>InformationSystem</Class>
                <Name>DIS-Broker</Name>
              </Service>
              <PackageName>DIS-Broker_stubs</PackageName>
              <SameDHN value="1"/>
              <SameDL value="0"/>
              <SameVO value="0"/>
              <Priority>1</Priority>
            </Dependency>
          </Dependencies>
          <DHNRequirements>
            <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="java1.5"/>
          </DHNRequirements>
          <ConfigurationFiles/>
          <WSRFService>
            <GARArchive>org_diligentproject_informationservice_disregistry.gar</GARArchive>
            <BuildFile>build.xml</BuildFile>
            <DeploymentOptions/>
            <ArchitecturalPattern>Factory</ArchitecturalPattern>
            <WSRFEntry>
              <EntryName>diligentproject/informationservice/disregistry/DISRegistryService</EntryName>
              <Factory value="false"/>
              <Parameters/>
              <Security name="">
                <securityDescriptor/>
                <defaultIdentity>
                  <subject/>
                  <CASubject/>
                </defaultIdentity>
              </Security>
              <WSDL>
                <default:definitions xmlns:wsrp="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-
draft-01.xsd" xmlns:tns="http://diligentproject.org/namespaces/informationservice/disregistry/DISRegistryService"
xmlns:wsrlw="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.wsdl"
xmlns:wsrpw="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.wsdl"
xmlns:wsdlpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor" xmlns:wsntw="http://docs.oasis-
open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.wsdl" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/" name="DISRegistry"
targetNamespace="http://diligentproject.org/namespaces/informationservice/disregistry/DISRegistryService">
                  <default:import location="../../../wsrf/properties/WS-ResourceProperties.wsdl"
namespace="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.wsdl"/>
                  <default:import location="../../../wsrf/lifetime/WS-ResourceLifetime.wsdl" namespace="http://docs.oasis-
open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.wsdl"/>
                  <default:import location="../../../wsrf/notification/WS-BaseN.wsdl" namespace="http://docs.oasis-
open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.wsdl"/>
                  <default:types>
```

```
                    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://diligentproject.org/namespaces/informationservice/disregistry/DISRegistryService"><!--  RESOURCE
PROPERTIES -->
                        <xsd:element name="Profile" type="xsd:string"/>
                        <xsd:element name="update" type="xsd:string"/>
                        <xsd:element name="updateResponse">
                           <xsd:complexType/>
                        </xsd:element>
                        <xsd:element name="DILIGENTResourceProperties">
                           <xsd:complexType>
                              <xsd:sequence>
                                 <xsd:element maxOccurs="1" minOccurs="1" ref="tns:Profile"/>
                              </xsd:sequence>
                           </xsd:complexType>
                        </xsd:element>
                     </xsd:schema>
                  </default:types>
                  <default:message name="UpdateInputMessage">
                     <default:part element="tns:update" name="parameters"/>
                  </default:message>
                  <default:message name="UpdateOutputMessage">
                     <default:part element="tns:updateResponse" name="parameters"/>
                  </default:message>
                  <default:portType name="DISRegistryPortType" wsdlpp:extends="wsrpw:GetResourceProperty
wsrpw:GetMultipleResourceProperties          wsrpw:SetResourceProperties
wsrpw:QueryResourceProperties       wsrlw:ImmediateResourceTermination      wsntw:NotificationProducer"
wsrp:ResourceProperties="tns:DILIGENTResourceProperties">
                     <default:operation name="update">
                        <default:input message="tns:UpdateInputMessage"/>
                        <default:output message="tns:UpdateOutputMessage"/>
                     </default:operation>
                  </default:portType>
               </default:definitions>
            </WSDL>
         </WSRFEntry>
         <WSRFEntry>
            <EntryName>diligentproject/informationservice/disregistry/DISRegistryFactoryService</EntryName>
            <Factory value="true"/>
            <Parameters/>
            <Security name="">
               <securityDescriptor/>
               <defaultIdentity>
                  <subject/>
                  <CASubject/>
               </defaultIdentity>
            </Security>
            <WSDL>
               <default:definitions
xmlns:tns="http://diligentproject.org/namespaces/informationservice/disregistry/DISRegistryFactoryService"
xmlns:wsdlpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/" name="DISRegistryFactory"
targetNamespace="http://diligentproject.org/namespaces/informationservice/disregistry/DISRegistryFactoryService"><!--
============================================================                T Y P E S
============================================================-->
                  <default:types>
                     <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://diligentproject.org/namespaces/informationservice/disregistry/DISRegistryFactoryService">
                        <xsd:import namespace="http://schemas.xmlsoap.org/ws/2004/03/addressing"
schemaLocation="../../../ws/addressing/WS-Addressing.xsd"/><!-- REQUESTS AND RESPONSES -->
                        <xsd:complexType name="Map">
                           <xsd:sequence>
                              <xsd:element maxOccurs="unbounded" minOccurs="0" name="item">
                                 <xsd:complexType>
                                    <xsd:sequence>
                                       <xsd:element name="key" type="xsd:string"/>
                                       <xsd:element name="value" type="xsd:string"/>
                                    </xsd:sequence>
                                 </xsd:complexType>
                              </xsd:element>
```

```
                    </xsd:sequence>
                </xsd:complexType>
                <xsd:element name="ProfileMapping" type="tns:Map"/>
                <xsd:complexType name="UpdateProfileMessage">
                    <xsd:sequence>
                        <xsd:element name="DiligentID" type="xsd:string"/>
                        <xsd:element name="xmlProfile" type="xsd:string"/>
                    </xsd:sequence>
                </xsd:complexType>
                <xsd:element name="createResource" type="xsd:string"/>
                <xsd:element name="createResourceResponse">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element ref="wsa:EndpointReference"/>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
                <xsd:element name="updateResource" type="tns:UpdateProfileMessage">
                    <xsd:complexType/>
                </xsd:element>
                <xsd:element name="updateResourceResponse">
                    <xsd:complexType/>
                </xsd:element>
                <xsd:element name="removeResource" type="xsd:string"/>
                <xsd:element name="removeResourceResponse">
                    <xsd:complexType/>
                </xsd:element>
                <xsd:element name="DISRegistryFactoryResourceProperties">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element maxOccurs="1" minOccurs="1" ref="tns:ProfileMapping"/>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
            </xsd:schema>
        </default:types><!--================================================================
MESSAGES                ================================================================-->
        <default:message name="CreateResourceRequest">
            <default:part element="tns:createResource" name="request"/>
        </default:message>
        <default:message name="CreateResourceResponse">
            <default:part element="tns:createResourceResponse" name="response"/>
        </default:message>
        <default:message name="UpdateResourceRequest">
            <default:part element="tns:updateResource" name="request"/>
        </default:message>
        <default:message name="UpdateResourceResponse">
            <default:part element="tns:updateResourceResponse" name="response"/>
        </default:message>
        <default:message name="RemoveResourceRequest">
            <default:part element="tns:removeResource" name="request"/>
        </default:message>
        <default:message name="RemoveResourceResponse">
            <default:part element="tns:removeResourceResponse" name="response"/>
        </default:message><!--================================================================
PORTTYPE                ================================================================-->
        <default:portType name="DISRegistryFactoryPortType">
    wsdlpp:extends=&quot;wsrpw:GetResourceProperty              wsrpw:GetMultipleResourceProperties
wsrpw:SetResourceProperties              wsrpw:QueryResourceProperties&quot;
wsrp:ResourceProperties=&quot;tns:DISRegistryFactoryResourceProperties&quot;&gt;    <default:operation
name="createResource">
            <default:input message="tns:CreateResourceRequest"/>
            <default:output message="tns:CreateResourceResponse"/>
        </default:operation>
        <default:operation name="updateResource">
            <default:input message="tns:UpdateResourceRequest"/>
            <default:output message="tns:UpdateResourceResponse"/>
        </default:operation>
        <default:operation name="removeResource">
```

```
                        <default:input message="tns:RemoveResourceRequest"/>
                        <default:output message="tns:RemoveResourceResponse"/>
                      </default:operation>
                    </default:portType>
                  </default:definitions>
                </WSDL>
              </WSRFEntry>
            </WSRFService>
          </Package>
          <Package>
            <PackageName>DIS-Registry_stubs</PackageName>
            <PackageType>Library</PackageType>
            <Version>0.2</Version>
            <DLMandatory value="0"/>
            <DHNMandatory value="0"/>
            <VOMandatory value="1"/>
            <DisposeInterfaceSupport value="0"/>
            <MultiVersionSupport value="0"/>
            <VOSharingSupport value="0"/>
            <ManifestFile/>
            <InstallScripts/>
            <UninstallScripts/>
            <RebootScripts/>
            <Dependencies/>
            <DHNRequirements>
              <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="java1.5"/>
            </DHNRequirements>
            <Library>
              <Type>stub</Type>
              <IsStubOf>
                <PackageName>DIS-Registry</PackageName>
                <Service>
                  <Class>InformationSystem</Class>
                  <Name>&gt;DIS-Registry</Name>
                </Service>
              </IsStubOf>
              <Parameters/>
              <LibraryFile>org_diligentproject_informationservice_disregistry_stubs.jar</LibraryFile>
            </Library>
            <OtherFiles/>
            <OtherProperties/>
          </Package>
        </PackagesList>
      </Profile>
</DILIGENTResource>
```

A.4. DIS-Broker Service

```
<?xml version="1.0" encoding="UTF-8"?>
<DILIGENTResource xmlns:default="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsdlpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor" xmlns:wsrp="http://docs.oasis-
open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.xsd">
   <UniqueID>9b401d70-1bc1-11db-856d-83284c2b1ef1</UniqueID>
   <ResourceType>Service</ResourceType>
   <AuthorizationPolicies/>
   <Profile>
      <Class>InformationSystem</Class>
      <Name>DIS-Broker</Name>
      <DescriptiveParameters>
         <DescParameter/>
      </DescriptiveParameters>
      <QoS/>
      <DLDependencies>
         <DLComponent>
            <Class>InformationSystem</Class>
            <Name>DIS-IC</Name>
            <DescriptiveParametersValue/>
         </DLComponent>
      </DLDependencies>
      <SpecificData>Text</SpecificData>
```

```
<PackagesList>
  <Package>
    <PackageName>DIS-Broker_Service</PackageName>
    <PackageType>WSRFService</PackageType>
    <Version>0.2</Version>
    <DLMandatory value="0"/>
    <DHNMandatory value="0"/>
    <VOMandatory value="1"/>
    <DisposeInterfaceSupport value="1"/>
    <MultiVersionSupport value="0"/>
    <VOSharingSupport value="0"/>
    <ManifestFile/>
    <InstallScripts/>
    <UninstallScripts/>
    <RebootScripts/>
    <Dependencies>
      <Dependency>
        <Service>
          <Class>InformationSystem</Class>
          <Name>DIS-IC</Name>
        </Service>
        <PackageName>DIS-IP</PackageName>
        <SameDHN value="1"/>
        <SameDL value="0"/>
        <SameVO value="0"/>
        <Priority>1</Priority>
      </Dependency>
      <Dependency>
        <Service>
          <Class>InformationSystem</Class>
          <Name>DIS-IC</Name>
        </Service>
        <PackageName>DILIGENTProvider</PackageName>
        <SameDHN value="1"/>
        <SameDL value="0"/>
        <SameVO value="0"/>
        <Priority>1</Priority>
      </Dependency>
    </Dependencies>
    <DHNRequirements>
      <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="java1.5"/>
    </DHNRequirements>
    <ConfigurationFiles/>
    <WSRFService>
      <GARArchive>org_diligentproject_informationservice_disbroker.gar</GARArchive>
      <BuildFile>build.xml</BuildFile>
      <DeploymentOptions/>
      <ArchitecturalPattern>Singleton</ArchitecturalPattern>
      <WSRFEntry>
        <EntryName>diligentproject/informationservice/disbroker/DISBrokerService</EntryName>
        <Factory value="false"/>
        <Parameters/>
        <Security name="">
          <securityDescriptor/>
          <defaultIdentity>
            <subject/>
            <CASubject/>
          </defaultIdentity>
        </Security>
        <WSDL>
          <default:definitions xmlns:wsrp="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-
draft-01.xsd" xmlns:tns="http://diligentproject.org/namespaces/informationservice/disbroker/DISBrokerService"
xmlns:wsrpw="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.wsdl"
xmlns:wsdlpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor"
xmlns:diligent="http://diligentproject.org/namespaces/common/provider/DILIGENTProvider"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/" name="DISBroker"
targetNamespace="http://diligentproject.org/namespaces/informationservice/disbroker/DISBrokerService">
            <default:import location="../../../wsrf/properties/WS-ResourceProperties.wsdl"
namespace="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.wsdl"/>
```

```
<default:import location="DiligentProvider.wsdl"
namespace="http://diligentproject.org/namespaces/common/provider/DILIGENTProvider"/>
<default:types>
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://diligentproject.org/namespaces/informationservice/disbroker/DISBrokerService"><!-- Type for
EPR/Topic mapping-->
    <xsd:complexType name="map">
      <xsd:sequence>
        <xsd:element maxOccurs="unbounded" minOccurs="0" name="item">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="key" type="xsd:string"/>
              <xsd:element name="value" type="tns:VectorStub"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="subscribeMessage">
      <xsd:sequence>
        <xsd:element name="clientEPR" ref="wsa:EndpointReference"/>
        <xsd:element name="topic" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType><!--types for registerTopic -->
    <xsd:complexType name="VectorStub">
      <xsd:sequence>
        <xsd:element maxOccurs="unbounded" minOccurs="0" name="item">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="value" type="xsd:string"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="registerTopicMessage">
      <xsd:sequence>
        <xsd:element name="epr" ref="wsa:EndpointReference"/>
        <xsd:element name="vectorTopic" type="tns:VectorStub"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="registerTopic" type="tns:registerTopicMessage"/>
    <xsd:element name="registerTopicResponse">
      <xsd:complexType/>
    </xsd:element>
    <xsd:element name="subscribeToTopic" type="tns:subscribeMessage"/>
    <xsd:element name="subscribeToTopicResponse">
      <xsd:complexType/>
    </xsd:element>
    <xsd:element name="removeSubscription" type="tns:subscribeMessage"/>
    <xsd:element name="removeSubscriptionResponse">
      <xsd:complexType/>
    </xsd:element><!--  RESOURCE PROPERTIES -->
    <xsd:element name="TopicMapping" type="tns:map"/>
    <xsd:element name="DISBrokerResourceProperties">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element maxOccurs="1" minOccurs="1" ref="tns:TopicMapping"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
</default:types>
<default:message name="RegisterTopicRequest">
  <default:part element="tns:registerTopic" name="request"/>
</default:message>
<default:message name="RegisterTopicResponse">
  <default:part element="tns:registerTopicResponse" name="response"/>
</default:message>
```

```
                    <default:message name="SubscribeToTopicRequest">
                       <default:part element="tns:subscribeToTopic" name="request"/>
                    </default:message>
                    <default:message name="SubscribeToTopicResponse">
                       <default:part element="tns:subscribeToTopicResponse" name="response"/>
                    </default:message>
                    <default:message name="RemoveSubscriptionRequest">
                       <default:part element="tns:removeSubscription" name="request"/>
                    </default:message>
                    <default:message name="RemoveSubscriptionResponse">
                       <default:part element="tns:removeSubscriptionResponse" name="response"/>
                    </default:message>
                    <default:portType name="DISBrokerPortType" wsdlpp:extends=" wsrpw:GetResourceProperty
wsrpw:GetMultipleResourceProperties                wsrpw:SetResourceProperties
wsrpw:QueryResourceProperties          diligent:DiligentProvider"
wsrp:ResourceProperties="tns:DISBrokerResourceProperties">
                       <default:operation name="registerTopic">
                          <default:input message="tns:RegisterTopicRequest"/>
                          <default:output message="tns:RegisterTopicResponse"/>
                       </default:operation>
                       <default:operation name="subscribeToTopic">
                          <default:input message="tns:SubscribeToTopicRequest"/>
                          <default:output message="tns:SubscribeToTopicResponse"/>
                       </default:operation>
                       <default:operation name="removeSubscription">
                          <default:input message="tns:RemoveSubscriptionRequest"/>
                          <default:output message="tns:RemoveSubscriptionResponse"/>
                       </default:operation>
                    </default:portType>
                 </default:definitions>
              </WSDL>
           </WSRFEntry>
        </WSRFService>
     </Package>
     <Package>
        <PackageName>DIS-Broker_stubs</PackageName>
        <PackageType>Library</PackageType>
        <Version>0.2</Version>
        <DLMandatory value="0"/>
        <DHNMandatory value="0"/>
        <VOMandatory value="1"/>
        <DisposeInterfaceSupport value="0"/>
        <MultiVersionSupport value="0"/>
        <VOSharingSupport value="0"/>
        <ManifestFile/>
        <InstallScripts/>
        <UninstallScripts/>
        <RebootScripts/>
        <Dependencies/>
        <DHNRequirements>
           <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="java1.5"/>
        </DHNRequirements>
        <Library>
           <Type>stub</Type>
           <IsStubOf>
              <PackageName>DIS-Broker_Service</PackageName>
              <Service>
                 <Class>InformationSystem</Class>
                 <Name>DIS-Broker</Name>
              </Service>
           </IsStubOf>
           <Parameters/>
           <LibraryFile>org_diligentproject_informationservice_disbroker_stubs.jar</LibraryFile>
        </Library>
        <OtherFiles/>
        <OtherProperties/>
     </Package>
  </PackagesList>
</Profile>
```

```
</DILIGENTResource>
```

A.5. gLite SE Resource profile

```
<?xml version="1.0" encoding="UTF-8"?>
<DILIGENTResource xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <UniqueID>String</UniqueID>
    <ResourceType>gLiteResource</ResourceType>
    <AuthorizationPolicies>Text</AuthorizationPolicies>
    <Profile>
        <GlueResourceType>GlueSE</GlueResourceType>
        <StorageElement UniqueID="String">
            <SiteRef UniqueID="String"/>
            <InformationServiceURL> </InformationServiceURL>
            <SizeTotal>0</SizeTotal>
            <Architecture>disk</Architecture>
            <StorageArea LocalID="String">
                <Path>String</Path>
                <Type>volatile</Type>
                <Quota>0</Quota>
                <MinFileSize>0</MinFileSize>
                <MaxFileSize>0</MaxFileSize>
                <MaxData>0</MaxData>
                <MaxNumFiles>0</MaxNumFiles>
                <MaxPinDuration>0</MaxPinDuration>
                <ACL>
                    <Rule>String</Rule>
                </ACL>
            </StorageArea>
            <AccessProtocol LocalID="String">
                <Endpoint> </Endpoint>
                <Type>gsiftp</Type>
                <Version>String</Version>
                <Capability>String</Capability>
            </AccessProtocol>
            <ControlProtocol LocalID="String">
                <Endpoint></Endpoint>
                <Type>SRM</Type>
                <Version>String</Version>
                <Capability>String</Capability>
            </ControlProtocol>
        </StorageElement>
    </Profile>
</DILIGENTResource>
```

A.6. gLite Site Resource profile

```
<?xml version="1.0" encoding="UTF-8"?>
<DILIGENTResource xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
    <UniqueID>String</UniqueID>
    <ResourceType>gLiteResource</ResourceType>
    <AuthorizationPolicies>Text</AuthorizationPolicies>
    <Profile>
        <GlueResourceType>GlueSite</GlueResourceType>
        <StorageElement UniqueID="String">
            <SiteRef UniqueID="String"/>
            <InformationServiceURL></InformationServiceURL>
            <SizeTotal>0</SizeTotal>
            <Architecture>disk</Architecture>
            <StorageArea LocalID="String">
                <Path>String</Path>
                <Type>volatile</Type>
                <Quota>0</Quota>
                <MinFileSize>0</MinFileSize>
                <MaxFileSize>0</MaxFileSize>
                <MaxData>0</MaxData>
                <MaxNumFiles>0</MaxNumFiles>
                <MaxPinDuration>0</MaxPinDuration>
                <ACL>
                    <Rule>String</Rule>
                </ACL>
```

```
            </StorageArea>
            <AccessProtocol LocalID="String">
                <Endpoint></Endpoint>
                <Type>gsiftp</Type>
                <Version>String</Version>
                <Capability>String</Capability>
            </AccessProtocol>
            <ControlProtocol LocalID="String">
                <Endpoint></Endpoint>
                <Type>SRM</Type>
                <Version>String</Version>
                <Capability>String</Capability>
            </ControlProtocol>
        </StorageElement>
    </Profile>
</DILIGENTResource>
```

A.7. gLite CE Resource Profile

```
<DILIGENTResource xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ">
    <UniqueID>String</UniqueID>
    <ResourceType>gLiteResource</ResourceType>
    <AuthorizationPolicies>Text</AuthorizationPolicies>
    <Profile>
        <GlueResourceType>GlueCE</GlueResourceType>
        <StorageElement UniqueID="String">
            <SiteRef UniqueID="String"/>
            <InformationServiceURL></InformationServiceURL>
            <SizeTotal>0</SizeTotal>
            <Architecture>disk</Architecture>
            <StorageArea LocalID="String">
                <Path>String</Path>
                <Type>volatile</Type>
                <Quota>0</Quota>
                <MinFileSize>0</MinFileSize>
                <MaxFileSize>0</MaxFileSize>
                <MaxData>0</MaxData>
                <MaxNumFiles>0</MaxNumFiles>
                <MaxPinDuration>0</MaxPinDuration>
                <ACL>
                    <Rule>String</Rule>
                </ACL>
            </StorageArea>
            <AccessProtocol LocalID="String">
                <Endpoint></Endpoint>
                <Type>gsiftp</Type>
                <Version>String</Version>
                <Capability>String</Capability>
            </AccessProtocol>
            <ControlProtocol LocalID="String">
                <Endpoint></Endpoint>
                <Type>SRM</Type>
                <Version>String</Version>
                <Capability>String</Capability>
            </ControlProtocol>
        </StorageElement>
    </Profile>
</DILIGENTResource>
```

A.8. gLite Service Resource Profile

```
<DILIGENTResource xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
    <UniqueID>String</UniqueID>
    <ResourceType>gLiteResource</ResourceType>
    <AuthorizationPolicies>Text</AuthorizationPolicies>
    <Profile>
        <GlueResourceType>GlueService</GlueResourceType>
        <StorageElement UniqueID="String">
```

```
            <SiteRef UniqueID="String"/>
            <InformationServiceURL></InformationServiceURL>
            <SizeTotal>0</SizeTotal>
            <Architecture>disk</Architecture>
            <StorageArea LocalID="String">
                <Path>String</Path>
                <Type>volatile</Type>
                <Quota>0</Quota>
                <MinFileSize>0</MinFileSize>
                <MaxFileSize>0</MaxFileSize>
                <MaxData>0</MaxData>
                <MaxNumFiles>0</MaxNumFiles>
                <MaxPinDuration>0</MaxPinDuration>
                <ACL>
                    <Rule>String</Rule>
                </ACL>
            </StorageArea>
            <AccessProtocol LocalID="String">
                <Endpoint></Endpoint>
                <Type>gsiftp</Type>
                <Version>String</Version>
                <Capability>String</Capability>
            </AccessProtocol>
            <ControlProtocol LocalID="String">
                <Endpoint></Endpoint>
                <Type>SRM</Type>
                <Version>String</Version>
                <Capability>String</Capability>
            </ControlProtocol>
        </StorageElement>
    </Profile>
</DILIGENTResource>
```

A.9. DL Management Service

```
<DILIGENTResource xmlns:default="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsdlpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor" xmlns:wsrp="http://docs.oasis-
open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.xsd">
  <UniqueID>ff699230-1c03-11db-baab-baec964416b0</UniqueID>
  <ResourceType>Service</ResourceType>
  <AuthorizationPolicies/>
  <Profile>
    <Class>Keeper</Class>
    <Name>DLManagement</Name>
    <DescriptiveParameters>
      <DescParameter/>
    </DescriptiveParameters>
    <QoS/>
    <DLDependencies>
      <DLComponent>
        <Class>InformationSystem</Class>
        <Name>DIS-IC</Name>
        <DescriptiveParametersValue/>
      </DLComponent>
      <DLComponent>
        <Class>InformationSystem</Class>
        <Name>DIS-Broker</Name>
        <DescriptiveParametersValue/>
      </DLComponent>
      <DLComponent>
        <Class>Keeper</Class>
        <Name>HNM</Name>
        <DescriptiveParametersValue/>
      </DLComponent>
      <DLComponent>
        <Class>Keeper</Class>
        <Name>PackageRepository</Name>
        <DescriptiveParametersValue/>
```

```
      </DLComponent>
    </DLDependencies>
    <SpecificData>Text</SpecificData>
    <PackagesList>
      <Package>
        <PackageName>DLManagement_Service</PackageName>
        <PackageType>WSRFService</PackageType>
        <Version>0.2</Version>
        <DLMandatory value="1"/>
        <DHNMandatory value="0"/>
        <VOMandatory value=""/>
        <DisposeInterfaceSupport value="1"/>
        <MultiVersionSupport value="0"/>
        <VOSharingSupport value="0"/>
        <ManifestFile/>
        <InstallScripts/>
        <UninstallScripts/>
        <RebootScripts/>
        <Dependencies>
          <Dependency>
            <Service>
              <Class>InformationSystem</Class>
              <Name>DIS-IC</Name>
            </Service>
            <PackageName>DIS-HLSClient</PackageName>
            <SameDHN value="1"/>
            <SameDL value="0"/>
            <SameVO value="0"/>
            <Priority>1</Priority>
          </Dependency>
          <Dependency>
            <Service>
              <Class>InformationSystem</Class>
              <Name>DIS-Broker</Name>
            </Service>
            <PackageName>DIS-Broker_stubs</PackageName>
            <SameDHN value="1"/>
            <SameDL value="0"/>
            <SameVO value="0"/>
            <Priority>1</Priority>
          </Dependency>
          <Dependency>
            <Service>
              <Class>InformationSystem</Class>
              <Name> DIS-Registry</Name>
            </Service>
            <PackageName> DIS-Registry_stubs </PackageName>
            <SameDHN value="1"/>
            <SameDL value="0"/>
            <SameVO value="0"/>
            <Priority>1</Priority>
          </Dependency>
          <Dependency>
            <Service>
              <Class>InformationSystem</Class>
              <Name>DIS-IC</Name>
            </Service>
            <PackageName>Common-Profile</PackageName>
            <SameDHN value="1"/>
            <SameDL value="0"/>
            <SameVO value="0"/>
            <Priority>1</Priority>
          </Dependency>
          <Dependency>
            <Service>
              <Class>Keeper</Class>
              <Name>HNM</Name>
            </Service>
            <PackageName>HNM_stubs</PackageName>
```

```
                <SameDHN value="1"/>
                <SameDL value="0"/>
                <SameVO value="0"/>
                <Priority>1</Priority>
             </Dependency>
             <Dependency>
                <Service>
                   <Class>Keeper</Class>
                   <Name>HNM</Name>
                </Service>
                <PackageName>NAL</PackageName>
                <SameDHN value="1"/>
                <SameDL value="0"/>
                <SameVO value="0"/>
                <Priority>1</Priority>
             </Dependency>
             <Dependency>
                <Service>
                   <Class>Keeper</Class>
                   <Name>PackageRepository</Name>
                </Service>
                <PackageName>PackageRepository_stubs</PackageName>
                <SameDHN value="1"/>
                <SameDL value="0"/>
                <SameVO value="0"/>
                <Priority>1</Priority>
             </Dependency>
          </Dependencies>
          <DHNRequirements>
             <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="java1.5"/>
          </DHNRequirements>
          <ConfigurationFiles/>
          <WSRFService>
             <GARArchive>org_diligentproject_keeperservice_dlmanagement.gar</GARArchive>
             <BuildFile>build.xml</BuildFile>
             <DeploymentOptions/>
             <ArchitecturalPattern>Factory</ArchitecturalPattern>
             <WSRFEntry>
                <EntryName>diligentproject/keeperservice/dlmanagement/DLManagementFactoryService</EntryName>
                <Factory value="true"/>
                <Parameters/>
                <WSDL>
                   <default:definitions
xmlns:tns="http://diligentproject.org/namespaces/keeperservice/dlmanagement/DLManagementFactoryService"
xmlns:wsdlpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/" name="DLManagementFactory"
targetNamespace="http://diligentproject.org/namespaces/keeperservice/dlmanagement/DLManagementFactoryService">
                      <default:types>
                         <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://diligentproject.org/namespaces/keeperservice/dlmanagement/DLManagementFactoryService">
                            <xsd:import namespace="http://schemas.xmlsoap.org/ws/2004/03/addressing"
schemaLocation="../../../ws/addressing/WS-Addressing.xsd"/>
                            <xsd:complexType name="Map">
                               <xsd:sequence>
                                  <xsd:element maxOccurs="unbounded" minOccurs="0" name="item">
                                     <xsd:complexType>
                                        <xsd:sequence>
                                           <xsd:element name="key" type="xsd:string"/>
                                           <xsd:element name="value" type="xsd:string"/>
                                        </xsd:sequence>
                                     </xsd:complexType>
                                  </xsd:element>
                               </xsd:sequence>
                            </xsd:complexType>
                            <xsd:complexType name="createResourceMessage">
                               <xsd:sequence>
                                  <xsd:element name="voName" type="xsd:string"/>
                                  <xsd:element name="mapFile" type="xsd:string"/>
                               </xsd:sequence>
```

```
                    </xsd:complexType>
                    <xsd:element name="createResource" type="tns:createResourceMessage"/>
                    <xsd:element name="createResourceResponse">
                       <xsd:complexType>
                          <xsd:sequence>
                             <xsd:element ref="wsa:EndpointReference"/>
                          </xsd:sequence>
                       </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="updateResource" type="tns:createResourceMessage"/>
                    <xsd:element name="updateResourceResponse">
                       <xsd:complexType/>
                    </xsd:element>
                    <xsd:element name="removeResource" type="xsd:string"/>
                    <xsd:element name="removeResourceResponse">
                       <xsd:complexType/>
                    </xsd:element>
                    <xsd:element name="getVoMap" type="xsd:string"/>
                    <xsd:element name="getVoMapResponse" type="xsd:string"/>
                    <xsd:element name="VoMapping" type="tns:Map"/>
                    <xsd:element name="DLManagementFactoryResourceProperties">
                       <xsd:complexType>
                          <xsd:sequence>
                             <xsd:element maxOccurs="1" minOccurs="1" ref="tns:VoMapping"/>
                          </xsd:sequence>
                       </xsd:complexType>
                    </xsd:element>
                 </xsd:schema>
              </default:types>
              <default:message name="CreateResourceRequest">
                 <default:part element="tns:createResource" name="request"/>
              </default:message>
              <default:message name="CreateResourceResponse">
                 <default:part element="tns:createResourceResponse" name="response"/>
              </default:message>
              <default:message name="RemoveResourceRequest">
                 <default:part element="tns:removeResource" name="request"/>
              </default:message>
              <default:message name="RemoveResourceResponse">
                 <default:part element="tns:removeResourceResponse" name="response"/>
              </default:message>
              <default:message name="UpdateResourceRequest">
                 <default:part element="tns:updateResource" name="request"/>
              </default:message>
              <default:message name="UpdateResourceResponse">
                 <default:part element="tns:updateResourceResponse" name="response"/>
              </default:message>
              <default:message name="GetVoMapRequest">
                 <default:part element="tns:getVoMap" name="request"/>
              </default:message>
              <default:message name="GetVoMapResponse">
                 <default:part element="tns:getVoMapResponse" name="response"/>
              </default:message>
              <default:portType name="DLManagementFactoryPortType">
        wsdlpp:extends=&amp;quot;wsrpw:GetResourceProperty
wsrpw:GetMultipleResourceProperties                                    wsrpw:SetResourceProperties
                    wsrpw:QueryResourceProperties&amp;quot;
  wsrp:ResourceProperties=&amp;quot;tns:DLManagementFactoryResourceProperties&amp;quot;&amp;gt;
                                          <default:operation name="createResource">
                 <default:input message="tns:CreateResourceRequest"/>
                 <default:output message="tns:CreateResourceResponse"/>
              </default:operation>
              <default:operation name="removeResource">
                 <default:input message="tns:RemoveResourceRequest"/>
                 <default:output message="tns:RemoveResourceResponse"/>
              </default:operation>
              <default:operation name="updateResource">
                 <default:input message="tns:UpdateResourceRequest"/>
                 <default:output message="tns:UpdateResourceResponse"/>
```

```
                    </default:operation>
                    <default:operation name="getVoMap">
                      <default:input message="tns:GetVoMapRequest"/>
                      <default:output message="tns:GetVoMapResponse"/>
                    </default:operation>
                  </default:portType>
                </default:definitions>
              </WSDL>
            </WSRFEntry>
            <WSRFEntry>
              <EntryName>diligentproject/keeperservice/dlmanagement/DLManagementService</EntryName>
              <Factory value="false"/>
              <Parameters/>
              <WSDL>
                <default:definitions xmlns:wsrp="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-
draft-01.xsd" xmlns:tns="http://diligentproject.org/namespaces/keeperservice/dlmanagement/DLManagementService"
xmlns:wsrlw="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.wsdl"
xmlns:wsrpw="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.wsdl"
xmlns:wsdlpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor" xmlns:wsntw="http://docs.oasis-
open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.wsdl" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/" name="DLManagement"
targetNamespace="http://diligentproject.org/namespaces/keeperservice/dlmanagement/DLManagementService">
                  <default:import location="../../../wsrf/properties/WS-ResourceProperties.wsdl"
namespace="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.wsdl"/>
                  <default:types>
                    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://diligentproject.org/namespaces/keeperservice/dlmanagement/DLManagementService">
                      <xsd:element name="DLMap" type="xsd:string"/>
                      <xsd:element name="DLMapResourceProperties">
                        <xsd:complexType>
                          <xsd:sequence>
                            <xsd:element maxOccurs="1" minOccurs="1" ref="tns:DLMap"/>
                          </xsd:sequence>
                        </xsd:complexType>
                      </xsd:element><!-- Requests and responses -->
                      <xsd:element name="setDLMap" type="xsd:string"/>
                      <xsd:element name="setDLMapResponse">
                        <xsd:complexType/>
                      </xsd:element>
                      <xsd:element name="getDLMap">
                        <xsd:complexType/>
                      </xsd:element>
                      <xsd:element name="getDLMapResponse" type="xsd:string"/><!-- Resource properties -->
                    </xsd:schema>
                  </default:types>
                  <default:message name="GetDLMapInputMessage">
                    <default:part element="tns:getDLMap" name="parameters"/>
                  </default:message>
                  <default:message name="GetDLMapOutputMessage">
                    <default:part element="tns:getDLMapResponse" name="parameters"/>
                  </default:message>
                  <default:message name="SetDLMapInputMessage">
                    <default:part element="tns:setDLMap" name="parameters"/>
                  </default:message>
                  <default:message name="SetDLMapOutputMessage">
                    <default:part element="tns:setDLMapResponse" name="parameters"/>
                  </default:message>
                  <default:portType name="DLManagementPortType" wsdlpp:extends="wsrpw:GetResourceProperty
wsrpw:GetMultipleResourceProperties wsrpw:SetResourceProperties
wsrpw:QueryResourceProperties wsrlw:ImmediateResourceTermination wsntw:NotificationProducer"
wsrp:ResourceProperties="tns:DLMapResourceProperties">
                    <default:operation name="getDLMap">
                      <default:input message="tns:GetDLMapInputMessage"/>
                      <default:output message="tns:GetDLMapOutputMessage"/>
                    </default:operation>
                    <default:operation name="setDLMap">
                      <default:input message="tns:SetDLMapInputMessage"/>
                      <default:output message="tns:SetDLMapOutputMessage"/>
                    </default:operation>
```

```
                </default:portType>
              </default:definitions>
            </WSDL>
          </WSRFEntry>
        </WSRFService>
      </Package>
      <Package>
        <PackageName>DLManagement_stubs</PackageName>
        <PackageType>Library</PackageType>
        <Version>0.2</Version>
        <DLMandatory value="0"/>
        <DHNMandatory value="0"/>
        <VOMandatory value="1"/>
        <DisposeInterfaceSupport value="0"/>
        <MultiVersionSupport value="0"/>
        <VOSharingSupport value="0"/>
        <ManifestFile/>
        <InstallScripts/>
        <UninstallScripts/>
        <RebootScripts/>
        <Dependencies/>
        <DHNRequirements>
          <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="java1.5"/>
        </DHNRequirements>
        <Library>
          <Type>stub</Type>
          <IsStubOf>
            <PackageName>DLManagement_Service</PackageName>
            <Service>
              <Class>Keeper</Class>
              <Name>DLMmanegement</Name>
            </Service>
          </IsStubOf>
          <Parameters/>
          <LibraryFile>org_diligentproject_keeperservice_dlmanagement_stubs.jar</LibraryFile>
        </Library>
        <OtherFiles/>
        <OtherProperties/>
      </Package>
      <Package>
        <PackageName>MapManager</PackageName>
        <PackageType>Library</PackageType>
        <Version>0.2</Version>
        <DLMandatory value="0"/>
        <DHNMandatory value="0"/>
        <VOMandatory value="1"/>
        <DisposeInterfaceSupport value="0"/>
        <MultiVersionSupport value="0"/>
        <VOSharingSupport value="0"/>
        <ManifestFile/>
        <InstallScripts/>
        <UninstallScripts/>
        <RebootScripts/>
        <Dependencies/>
        <DHNRequirements>
          <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="java1.5"/>
        </DHNRequirements>
        <Library>
          <Type>shared</Type>
          <Parameters/>
          <LibraryFile>org_diligentproject_keeperservice_mapmanager.jar</LibraryFile>
        </Library>
        <OtherFiles/>
        <OtherProperties/>
      </Package>
      <Package>
        <PackageName>JAXB_software</PackageName>
        <PackageType>Software</PackageType>
        <Version>2.0</Version>
```

```
            <DLMandatory value="0"/>
            <DHNMandatory value="0"/>
            <VOMandatory value="0"/>
            <DisposeInterfaceSupport value="0"/>
            <MultiVersionSupport value="0"/>
            <VOSharingSupport value="0"/>
            <ManifestFile/>
            <InstallScripts/>
            <UninstallScripts/>
            <RebootScripts/>
            <Dependencies/>
            <DHNRequirements>
               <Req category="RunTimeEnv" operator="eq" requirement="Variable" value="java1.5"/>
            </DHNRequirements>
            <Software>
               <Files>
                  <File>jaxb1-impl.jar</File>
                  <File>jaxb-api.jar</File>
                  <File>jaxb-impl.jar</File>
                  <File>jaxb-xjc.jar</File>
                  <File>jsr173_api.jar</File>
               </Files>
            </Software>
            <OtherFiles/>
            <OtherProperties/>
         </Package>
      </PackagesList>
   </Profile>
</DILIGENTResource>
```

A.10. HNM Service Profile

```
<?xml version="1.0" encoding="UTF-8"?>
<DILIGENTResource>
    <UniqueID>002694b0-1bc4-11db-b40c-a78a2f5a15a7</UniqueID>
    <ResourceType>Service</ResourceType>
    <AuthorizationPolicies/>
    <Profile>
        <Class>Keeper</Class>
        <Name>HNMService</Name>
        <DescriptiveParameters>
            <DescParameter/>
        </DescriptiveParameters>
        <QoS/>
        <DLDependencies>
            <DLComponent>
                <Class>InformationSystem</Class>
                <Name>DIS-IC</Name>
                <DescriptiveParametersValue/>
            </DLComponent>
            <DLComponent>
                <Class>InformationService</Class>
                <Name>DIS-Registry</Name>
                <DescriptiveParametersValue/>
            </DLComponent>
            <DLComponent>
                <Class>Keeper</Class>
                <Name>PackageRepository</Name>
                <DescriptiveParametersValue/>
            </DLComponent>
            <DLComponent>
                <Class>Keeper</Class>
                <Name>DLManagement</Name>
                <DescriptiveParametersValue/>
            </DLComponent>
        </DLDependencies>
        <SpecificData/>
        <PackagesList>
```

```
<Package>
    <PackageName>HNM</PackageName>
    <PackageType>WSRFService</PackageType>
    <Version>0.9</Version>
    <DLMandatory value="0"/>
    <DHNMandatory value="1"/>
    <VOMandatory value="0"/>
    <DisposeInterfaceSupport value="0"/>
    <MultiVersionSupport value="0"/>
    <VOSharingSupport value="1"/>
    <ManifestFile/>
    <InstallScripts/>
    <UninstallScripts/>
    <RebootScripts/>
    <Dependencies>
        <Dependency>
            <Service>
                <Class>InformationSystem</Class>
                <Name>DIS-IC</Name>
            </Service>
            <PackageName>DIS-HLSClient</PackageName>
            <SameDHN value="1"/>
            <SameDL value="0"/>
            <SameVO value="0"/>
            <Priority>1</Priority>
        </Dependency>
        <Dependency>
            <Service>
                <Class>InformationService</Class>
                <Name>DIS-Registry</Name>
            </Service>
            <PackageName>DIS-Registry_stubs</PackageName>
            <SameDHN value="1"/>
            <SameDL value="0"/>
            <SameVO value="0"/>
            <Priority>1</Priority>
        </Dependency>
        <Dependency>
            <Service>
                <Class>Keeper</Class>
                <Name>DLManagementService</Name>
            </Service>
            <PackageName>DLManagementStub</PackageName>
            <SameDHN value="1"/>
            <SameDL value="0"/>
            <SameVO value="0"/>
            <Priority>1</Priority>
        </Dependency>
        <Dependency>
            <Service>
                <Class>Keeper</Class>
                <Name>HNMService</Name>
            </Service>
            <PackageName>NAL</PackageName>
            <SameDHN value="1"/>
            <SameDL value="0"/>
            <SameVO value="0"/>
            <Priority>1</Priority>
        </Dependency>
        <Dependency>
            <Service>
                <Class>InformationSystem</Class>
                <Name>DIS-IC</Name>
            </Service>
            <PackageName>DIS-IP</PackageName>
            <SameDHN value="1"/>
            <SameDL value="0"/>
            <SameVO value="0"/>
            <Priority>1</Priority>
```

```
            </Dependency>
        </Dependencies>
        <DHNRequirements>
            <Req category="RunTimeEnv" requirement="Variable" value="java1.5" operator="eq"/>
        </DHNRequirements>
        <ConfigurationFiles>
            <File>HNMConfig</File>
            <File>HNMStatus</File>
        </ConfigurationFiles>
        <WSRFService>
            <GARArchive>org_diligentproject_keeperservice_hnm.gar</GARArchive>
            <BuildFile>globus-build-service.sh</BuildFile>
            <DeploymentOptions/>
            <ArchitecturalPattern>Singleton</ArchitecturalPattern>
            <WSRFEntry>
                <EntryName>diligentproject/keeperservice/hnm/HNMService</EntryName>
                <Factory value="false"/>
                <Parameters/>
                <Security name="String">
                    <securityDescriptor/>
                    <defaultIdentity>
                        <subject>String</subject>
                        <CASubject>String</CASubject>
                    </defaultIdentity>
                </Security>
                <WSDL>
                    <definitions name="HNMService"
targetNamespace="http://www.diligentproject.org/namespaces/keeperservice/hnm/HNMService"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://www.diligentproject.org/namespaces/keeperservice/hnm/HNMService"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:wsntw="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-
BaseNotification-1.2-draft-01.wsdl" xmlns:wsdlpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
                        <wsdl:import namespace="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-
ResourceProperties-1.2-draft-01.wsdl" location="../../wsrf/properties/WS-ResourceProperties.wsdl"/>
                        <wsdl:import namespace="http://docs.oasis-open.org/wsn/2004/06/wsn-WS-
BaseNotification-1.2-draft-01.wsdl" location="../../wsrf/notification/WS-BaseN.wsdl"/>
                        <types>
                            <xsd:schema
targetNamespace="http://www.diligentproject.org/namespaces/keeperservice/hnm/HNMService"
xmlns:tns="http://www.diligentproject.org/namespaces/keeperservice/hnm/HNMService"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
                                <!-- Requests and responses -->
                                <xsd:element name="updateStatus">
                                    <xsd:complexType/>
                                </xsd:element>
                                <xsd:element name="updateStatusResponse">
                                    <xsd:complexType/>
                                </xsd:element>
                            </xsd:schema>
                        </types>
                        <message name="updateStatusInputMessage">
                            <part name="parameters" element="tns:updateStatus"/>
                        </message>
                        <message name="updateStatusOutputMessage">
                            <part name="parameters" element="tns:updateStatusResponse"/>
                        </message>
                        <portType name="HNMPortType">
                            <operation name="updateStatus">
                                <input message="tns:updateStatusInputMessage"/>
                                <output message="tns:updateStatusOutputMessage"/>
                            </operation>
                        </portType>
                    </definitions>
                </WSDL>
            </WSRFEntry>
        </WSRFService>
        <OtherFiles/>
        <OtherProperties/>
```

```
            </Package>
            <Package>
                <PackageName>NAL</PackageName>
                <PackageType>Library</PackageType>
                <Version>0.9</Version>
                <DLMandatory value="0"/>
                <DHNMandatory value="1"/>
                <VOMandatory value="0"/>
                <DisposeInterfaceSupport value="0"/>
                <MultiVersionSupport value="0"/>
                <VOSharingSupport value="0"/>
                <ManifestFile/>
                <InstallScripts/>
                <UninstallScripts/>
                <RebootScripts/>
                <Dependencies/>
                <DHNRequirements>
                    <Req category="RunTimeEnv" requirement="Variable" value="java1.5" operator="eq"/>
                </DHNRequirements>
                <Library>
                    <Type>shared</Type>
                    <Parameters/>
                    <LibraryFile>org.diligentproject.keeperservice.hnm.impl.nal.jar</LibraryFile>
                </Library>
                <OtherFiles/>
                <OtherProperties/>
            </Package>
            <Package>
                <PackageName>HNM-stubs</PackageName>
                <PackageType>Library</PackageType>
                <Version>0.9</Version>
                <DLMandatory value="0"/>
                <DHNMandatory value="1"/>
                <VOMandatory value="0"/>
                <DisposeInterfaceSupport value="0"/>
                <MultiVersionSupport value="0"/>
                <VOSharingSupport value="0"/>
                <ManifestFile/>
                <InstallScripts/>
                <UninstallScripts/>
                <RebootScripts/>
                <Dependencies/>
                <DHNRequirements>
                    <Req category="RunTimeEnv" requirement="Variable" value="java1.5" operator="eq"/>
                </DHNRequirements>
                <Library>
                    <Type>stub</Type>
                    <IsStubOf>
                        <PackageName>HNM</PackageName>
                        <Service>
                            <Class>Keeper</Class>
                            <Name>HNM</Name>
                        </Service>
                    </IsStubOf>
                    <Parameters/>
                    <LibraryFile>org.diligentproject.stubs.keeperservice.hnm.jar</LibraryFile>
                </Library>
                <OtherFiles/>
                <OtherProperties/>
            </Package>
        </PackagesList>
    </Profile>
</DILIGENTResource>
```

A.11. DHN Profile sample

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--Sample XML file generated by XML Spy v4.1 U ()-->
<DILIGENTResource xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="D:\ISTI-Works\DILIGENT Services implementation\Common\DILResourceSchemas\DHN\DHN_profile.xsd">
    <UniqueID></UniqueID>
    <ResourceType>DHN</ResourceType>
    <AuthorizationPolicies></AuthorizationPolicies>
    <Profile>
        <DHNDescription>
            <Name>host.name</Name>
<!-- Please complete manually following tags -->
            <Architecture PlatformType="Intel" SMPSize="1" SMTSize="1"/>
<!-- RAMAvailable and VirtualAvailable are automatically filled by the HNM -->
            <MainMemory RAMSize="4054720" VirtualSize="4088532" RAMAvailable="0" VirtualAvailable="0"/>
            <OperatingSystem Name="SL" Release="CERN" Version="3.0.4"/>
            <Processor Vendor="Athlon" Model="i686" ClockSpeed="1" InstructionSet="String" OtherDescription="String" CacheL1="1" CacheL1I="1" CacheL1D="1" CacheL2="1"/>
            <NetworkAdapter InboundIP="1" OutboundIP="1" Name="String" IPAddress="String" MTU="1"/>
            <Benchmark SI00="1" SF00="1"/>
            <RunTimeEnv>
                <Variable>gt_aggr_framework</Variable>
                <Variable>java1.5</Variable>
            </RunTimeEnv>
            <StorageDevice Name="disk1" Type="String" TransferRate="1" Size="58083380"/>
            <StoragePartition Name="/dev/sdb2" Size="58083380" RateRate="1" WriteRate="1"/>
            <LocalFileSystem Name="/dev/sdb2" Root="/home/globus" Size="1" ReadOnly="1" Type="fat32"/>
            <StorageDevice2StoragePartition StorageDeviceName="disk1" StoragePartitionName="/dev/sdb2"/>
            <StoragePartition2FileSystem StoragePartitionName="/dev/sdb2" FileSystemName="/dev/sdb2"/>
<!-- End of complete manually -->
            <Load Last1Min="0" Last5Min="0" Last15Min="0"/>
            <HistoricalLoad Last1H="0" Last1Day="0" Last1Week="0"/>
            <LocalAvailableSpace>567432</LocalAvailableSpace>
            <LastUpdate>String</LastUpdate>
        </DHNDescription>
<!-- Please complete manually following tags -->
        <Site>CNR</Site>
<!-- End of complete manually -->
        <NodeDeployedPackagesList/>
    </Profile>
</DILIGENTResource>
```

A.12. DVOS Common Profile

```
<?xml version="1.0"?>
<DILIGENTResource xmlns:p1="http://schemas.xmlsoap.org/wsdl/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
  <UniqueID></UniqueID>
  <ResourceType>Service</ResourceType>
  <AuthorizationPolicies>
  </AuthorizationPolicies>
  <Profile>
    <Class>DVOS</Class>
    <Name>common</Name>
    <DescriptiveParameters>
      <DescParameter/>
    </DescriptiveParameters>
    <QoS/>
    <DLDependencies>
    </DLDependencies>
    <SpecificData>
    </SpecificData>
    <PackagesList>
      <Package>
        <PackageName>dvos.common</PackageName>
        <PackageType>Library</PackageType>
        <Version>1.0</Version>
        <DHNMandatory value="true"/>
        <DHNRequirements>
          <Req category="Site" requirement="string" value="java1.5" operator="le"/>
        </DHNRequirements>
        <ConfigurationFiles>
```

```
            </ConfigurationFiles>
            <Library>
               <Type>shared</Type>
                     <LibraryFile>dvos.common.jar</LibraryFile>
                  </Library>
         </Package>
               <Package>
                  <PackageName>dvos.common</PackageName>
                  <PackageType>WSRFService</PackageType>
                  <Version>1.0</Version>
                  <DLMandatory value="true"/>
                  <DHNMandatory value="true"/>
                  <VOMandatory value="true"/>
                  <DisposeInterfaceSupport value="true"/>
                  <MultiVersionSupport value="false"/>
                  <VOSharingSupport value="true"/>
                  <Dependencies>
                  </Dependencies>
                  <DHNRequirements>
                     <Req category="Site" requirement="string" value="java1.5" operator="le"/>
                  </DHNRequirements>
                  <WSRFService>
                     <GARArchive>dvos.common.gar</GARArchive>
                     <BuildFile>build.xml</BuildFile>
                     <DeploymentOptions/>
                     <ArchitecturalPattern>Stateless</ArchitecturalPattern>
                     <WSRFEntry>
                        <EntryName></EntryName>
                        <Factory value="false"/>
                        <WSDL>
                           <p1:definitions></p1:definitions>
                        </WSDL>
                     </WSRFEntry>
                  </WSRFService>
                  <OtherFiles>
                     <File></File>
                  </OtherFiles>
                  <OtherProperties></OtherProperties>
               </Package>
      </PackagesList>
   </Profile>
</DILIGENTResource>
```

A.13. Authentication API Profile

```
<?xml version="1.0"?>
<DILIGENTResource xmlns:p1="http://schemas.xmlsoap.org/wsdl/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
   <UniqueID></UniqueID>
   <ResourceType>Service</ResourceType>
   <AuthorizationPolicies>
   </AuthorizationPolicies>
   <Profile>
      <Class>DVOS</Class>
      <Name>authentication</Name>
      <DescriptiveParameters>
         <DescParameter/>
      </DescriptiveParameters>
      <QoS/>
      <DLDependencies>
      </DLDependencies>
      <SpecificData>
      </SpecificData>
      <PackagesList>
         <Package>
            <PackageName>dvos.authentication-api</PackageName>
            <PackageType>Library</PackageType>
            <Version>1.0</Version>
            <DHNMandatory value="true"/>
                  <Dependencies>
```

```
                    <Dependency>
                        <Service>
                            <Class>DVOS</Class>
                            <Name>common</Name>
                        </Service>
                        <PackageName>dvos.common</PackageName>
                        <SameDHN value="true"/>
                        <Priority>4</Priority>
                    </Dependency>
                </Dependencies>
            <DHNRequirements>
              <Req category="Site" requirement="string" value="java1.5" operator="le"/>
            </DHNRequirements>
            <ConfigurationFiles>
            </ConfigurationFiles>
            <Library>
              <Type>shared</Type>
                    <LibraryFile>dvos.authentication-api.jar</LibraryFile>
                </Library>
        </Package>
      </PackagesList>
   </Profile>
</DILIGENTResource>
```

A.14. Delegation Service Profile

```
<DILIGENTResource xmlns:p1="http://schemas.xmlsoap.org/wsdl/">
    <UniqueID>f264e320-1742-11db-8f2d-90d3c7d2b404</UniqueID>
    <ResourceType>Service</ResourceType>
    <AuthorizationPolicies>Text</AuthorizationPolicies>
    <Profile>
        <Class>DVOS</Class>
        <Name>delegation</Name>
        <DescriptiveParameters>
            <DescParameter/>
        </DescriptiveParameters>
        <QoS/>
        <DLDependencies/>
        <SpecificData>Text</SpecificData>
        <PackagesList>
            <Package>
                <PackageName>dvos.common</PackageName>
                <PackageType>Library</PackageType>
                <Version>.9</Version>
                <DLMandatory value="false"/>
                <DHNMandatory value="true"/>
                <VOMandatory value="false"/>
                <DisposeInterfaceSupport value="true"/>
                <MultiVersionSupport value="false"/>
                <VOSharingSupport value="true"/>
                <DHNRequirements>
                    <Req category="Site" requirement="string" value="java1.5" operator="le"/>
                </DHNRequirements>
                <Library>
              <Type>shared</Type>
                    <LibraryFile>dvos.common.jar</LibraryFile>
                </Library>
            </Package>
            <Package>
                <PackageName>dvos.delegation-stubs</PackageName>
                <PackageType>Library</PackageType>
                <Version>.9</Version>
                <DLMandatory value="false"/>
                <DHNMandatory value="true"/>
                <VOMandatory value="false"/>
                <DisposeInterfaceSupport value="true"/>
                <MultiVersionSupport value="false"/>
                <VOSharingSupport value="true"/>
                <!--<Dependencies>
                    <Dependency>
```

```
                        <Service>
                            <Class>DVOS</Class>
                            <Name>registration</Name>
                        </Service>
                        <PackageName>dvos.common</PackageName>
                        <SameDHN value="true"/>
                        <SameDL value="false"/>
                        <SameVO value="false"/>
                        <Priority>4</Priority>
                        </Dependency>
                </Dependencies>-->
                <DHNRequirements>
                    <Req category="Site" requirement="string" value="java1.5" operator="le"/>
                </DHNRequirements>
                <Library>
                    <Type>stub</Type>
                    <IsStubOf>
                        <PackageName>dvos.delegation-service</PackageName>
                        <Service>
                            <Class>DVOS</Class>
                            <Name>delegation</Name>
                        </Service>
                    </IsStubOf>
                    <LibraryFile>dvos.delegation-stubs.jar</LibraryFile>
                </Library>
            </Package>
            <Package>
                <PackageName>dvos.delegation-service</PackageName>
                <PackageType>WSRFService</PackageType>
                <Version>1.0</Version>
                <DLMandatory value="false"/>
                <DHNMandatory value="true"/>
                <VOMandatory value="false"/>
                <DisposeInterfaceSupport value="true"/>
                <MultiVersionSupport value="false"/>
                <VOSharingSupport value="true"/>
                <!--<Dependencies>
                    <Dependency>
                        <Service>
                            <Class>DVOS</Class>
                            <Name>registration</Name>
                        </Service>
                        <PackageName>dvos.common</PackageName>
                        <SameDHN value="true"/>
                        <SameDL value="false"/>
                        <SameVO value="false"/>
                        <Priority>4</Priority>
                    </Dependency>
                </Dependencies>-->
                <DHNRequirements>
                    <Req category="Site" requirement="string" value="java1.5" operator="le"/>
                </DHNRequirements>
                <WSRFService>
                    <GARArchive>dvos.delegation-service.gar</GARArchive>
                    <BuildFile>build.xml</BuildFile>
                    <DeploymentOptions/>
                    <ArchitecturalPattern>Singleton</ArchitecturalPattern>
                    <WSRFEntry>
                        <EntryName>diligentproject/dvos/delegation/DelegationService</EntryName>
                        <Factory value="false"/>
                        <WSDL>
                            <p1:definitions/>
                        </WSDL>
                    </WSRFEntry>
                </WSRFService>
            </Package>
        </PackagesList>
    </Profile>
</DILIGENTResource>
```

A.15. CredentialsRenewal Service Profile

```
<?xml version="1.0"?>
<DILIGENTResource xmlns:p1="http://schemas.xmlsoap.org/wsdl/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
    <UniqueID>String</UniqueID>
    <ResourceType>Service</ResourceType>
    <AuthorizationPolicies>Text</AuthorizationPolicies>
    <Profile>
        <Class>DVOS</Class>
        <Name>credential-renewal</Name>
        <DescriptiveParameters>
            <DescParameter/>
        </DescriptiveParameters>
        <QoS/>
        <DLDependencies/>
        <SpecificData>Text</SpecificData>
        <PackagesList>
            <Package>
                <PackageName>dvos.credential-renewal-stubs</PackageName>
                <PackageType>Library</PackageType>
                <Version>.9</Version>
                <DLMandatory value="true"/>
                <DHNMandatory value="true"/>
                <VOMandatory value="true"/>
                <DisposeInterfaceSupport value="true"/>
                <MultiVersionSupport value="false"/>
                <VOSharingSupport value="true"/>
                <Dependencies>
                    <Dependency>
                        <Service>
                            <Class>DVOS</Class>
                            <Name>common</Name>
                        </Service>
                        <PackageName>dvos.common</PackageName>
                        <SameDHN value="true"/>
                        <SameDL value="true"/>
                        <SameVO value="true"/>
                        <Priority>4</Priority>
                    </Dependency>
                </Dependencies>
                <DHNRequirements>
                    <Req category="Site" requirement="string" value="java1.5" operator="le"/>
                </DHNRequirements>
                <Library>
            <Type>stub</Type>
            <IsStubOf>
                        <PackageName>dvos.credential-renewal-service</PackageName>
              <Service>
                 <Class>DVOS</Class>
                 <Name>credential-renewal</Name>
              </Service>
                    </IsStubOf>
                    <LibraryFile>dvos.credential-renewal-stubs.jar</LibraryFile>
                </Library>
            </Package>
            <Package>
                <PackageName>dvos.credential-renewal-service</PackageName>
                <PackageType>WSRFService</PackageType>
                <Version>1.0</Version>
                <DLMandatory value="true"/>
                <DHNMandatory value="true"/>
                <VOMandatory value="true"/>
                <DisposeInterfaceSupport value="true"/>
                <MultiVersionSupport value="false"/>
                <VOSharingSupport value="true"/>
                <Dependencies>
                    <Dependency>
                        <Service>
                            <Class>DVOS</Class>
```

```
                <Name>common</Name>
            </Service>
            <PackageName>dvos.common</PackageName>
            <SameDHN value="true"/>
            <SameDL value="true"/>
            <SameVO value="true"/>
            <Priority>4</Priority>
        </Dependency>
        <Dependency>
            <Service>
                <Class>DVOS</Class>
                <Name>credential-renewal</Name>
            </Service>
            <PackageName>dvos.credential-renewal-stubs</PackageName>
            <SameDHN value="true"/>
            <SameDL value="true"/>
            <SameVO value="true"/>
            <Priority>4</Priority>
        </Dependency>
        <Dependency>
            <Service>
                <Class>DVOS</Class>
                <Name>authentication</Name>
            </Service>
            <PackageName>dvos.authentication-api</PackageName>
            <SameDHN value="true"/>
            <SameDL value="true"/>
            <SameVO value="true"/>
            <Priority>4</Priority>
        </Dependency>
        <Dependency>
            <Service>
                <Class>DVOS</Class>
                <Name>delegation</Name>
            </Service>
            <PackageName>dvos.delegation-stubs</PackageName>
            <SameDHN value="true"/>
            <SameDL value="true"/>
            <SameVO value="true"/>
            <Priority>4</Priority>
        </Dependency>
    </Dependencies>
    <DHNRequirements>
        <Req category="Site" requirement="string" value="java1.5" operator="le"/>
    </DHNRequirements>
    <WSRFService>
        <GARArchive>dvos.credential-renewal-service.gar</GARArchive>
        <BuildFile>build.xml</BuildFile>
        <DeploymentOptions/>
        <ArchitecturalPattern>Singleton</ArchitecturalPattern>
        <WSRFEntry>
            <EntryName>diligent/dvos/credential-renewal/CredentialsRenewalService</EntryName>
            <Factory value="true"/>
            <WSDL>
                <p1:definitions></p1:definitions>
            </WSDL>
        </WSRFEntry>
    </WSRFService>
    <OtherFiles>
        <File></File>
    </OtherFiles>
    <OtherProperties></OtherProperties>
</Package>
<Package>
    <PackageName>dvos.credential-renewal-api</PackageName>
    <PackageType>Library</PackageType>
    <Version>.9</Version>
    <DLMandatory value="true"/>
    <DHNMandatory value="true"/>
```

```
                <VOMandatory value="true"/>
                <DisposeInterfaceSupport value="true"/>
                <MultiVersionSupport value="false"/>
                <VOSharingSupport value="true"/>
                <Dependencies>
                    <Dependency>
                        <Service>
                            <Class>DVOS</Class>
                            <Name>common</Name>
                        </Service>
                        <PackageName>dvos.common</PackageName>
                        <SameDHN value="true"/>
                        <SameDL value="true"/>
                        <SameVO value="true"/>
                        <Priority>4</Priority>
                    </Dependency>
                    <Dependency>
                        <Service>
                            <Class>DVOS</Class>
                            <Name>credential-renewal</Name>
                        </Service>
                        <PackageName>dvos.credential-renewal-stubs</PackageName>
                        <SameDHN value="true"/>
                        <SameDL value="true"/>
                        <SameVO value="true"/>
                        <Priority>4</Priority>
                    </Dependency>
                    <Dependency>
                        <Service>
                            <Class>DVOS</Class>
                            <Name>authentication</Name>
                        </Service>
                        <PackageName>dvos.authentication-api</PackageName>
                        <SameDHN value="true"/>
                        <SameDL value="true"/>
                        <SameVO value="true"/>
                        <Priority>4</Priority>
                    </Dependency>
                    <Dependency>
                        <Service>
                            <Class>DVOS</Class>
                            <Name>delegation</Name>
                        </Service>
                        <PackageName>dvos.delegation-stubs</PackageName>
                        <SameDHN value="true"/>
                        <SameDL value="true"/>
                        <SameVO value="true"/>
                        <Priority>4</Priority>
                    </Dependency>
                </Dependencies>
                <DHNRequirements>
                    <Req category="Site" requirement="string" value="java1.5" operator="le"/>
                </DHNRequirements>
                <Library>
            <Type>shared</Type>
                    <LibraryFile>dvos.credential-renewal-api.jar</LibraryFile>
                </Library>
            </Package>
        </PackagesList>
    </Profile>
</DILIGENTResource>
```

A.16. Authorization Service Profile

```
<DILIGENTResource xmlns:p1="http://schemas.xmlsoap.org/wsdl/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
    <UniqueID>String</UniqueID>
    <ResourceType>Service</ResourceType>
    <AuthorizationPolicies>Text</AuthorizationPolicies>
    <Profile>
```

```
<Class>DVOS</Class>
<Name>authorization</Name>
<DescriptiveParameters>
    <DescParameter/>
</DescriptiveParameters>
<QoS/>
<DLDependencies/>
<SpecificData>Text</SpecificData>
<PackagesList>
    <Package>
        <PackageName>dvos.authorization-stubs</PackageName>
        <PackageType>Library</PackageType>
        <Version>.9</Version>
        <DLMandatory value="false"/>
        <DHNMandatory value="true"/>
        <VOMandatory value="false"/>
        <DisposeInterfaceSupport value="true"/>
        <MultiVersionSupport value="false"/>
        <VOSharingSupport value="true"/>
        <Dependencies>
            <Dependency>
                <Service>
                    <Class>DVOS</Class>
                    <Name>delegation</Name>
                </Service>
                <PackageName>dvos.common</PackageName>
                <SameDHN value="true"/>
                <SameDL value="false"/>
                <SameVO value="false"/>
                <Priority>4</Priority>
            </Dependency>
        </Dependencies>
        <DHNRequirements>
            <Req category="Site" requirement="string" value="java1.5" operator="le"/>
        </DHNRequirements>
        <Library>
    <Type>stub</Type>
    <IsStubOf>
                <PackageName>dvos.authorization-service</PackageName>
      <Service>
        <Class>DVOS</Class>
        <Name>authorization</Name>
      </Service>
                </IsStubOf>
                <LibraryFile>dvos.authorization-stubs.jar</LibraryFile>
        </Library>
    </Package>
    <Package>
        <PackageName>dvos.authorization-service</PackageName>
        <PackageType>WSRFService</PackageType>
        <Version>1.0</Version>
        <DLMandatory value="true"/>
        <DHNMandatory value="true"/>
        <VOMandatory value="true"/>
        <DisposeInterfaceSupport value="true"/>
        <MultiVersionSupport value="false"/>
        <VOSharingSupport value="true"/>
        <Dependencies>
            <Dependency>
                <Service>
                    <Class>DVOS</Class>
                    <Name>delegation</Name>
                </Service>
                <PackageName>dvos.common</PackageName>
                <SameDHN value="true"/>
                <SameDL value="false"/>
                <SameVO value="false"/>
                <Priority>4</Priority>
            </Dependency>
```

```
            <Dependency>
                <Service>
                    <Class>DVOS</Class>
                    <Name>authorization</Name>
                </Service>
                <PackageName>dvos.authorization-stubs</PackageName>
                <SameDHN value="true"/>
                <SameDL value="true"/>
                <SameVO value="true"/>
                <Priority>4</Priority>
            </Dependency>
            <Dependency>
                <Service>
                    <Class>DVOS</Class>
                    <Name>delegation</Name>
                </Service>
                <PackageName>dvos.delegation-stubs</PackageName>
                <SameDHN value="true"/>
                <SameDL value="true"/>
                <SameVO value="true"/>
                <Priority>4</Priority>
            </Dependency>
            <Dependency>
                <Service>
                    <Class>DVOS</Class>
                    <Name>delegation</Name>
                </Service>
                <PackageName>dvos.delegation-service</PackageName>
                <SameDHN value="true"/>
                <SameDL value="true"/>
                <SameVO value="true"/>
                <Priority>4</Priority>
            </Dependency>
        </Dependencies>
        <DHNRequirements>
            <Req category="Site" requirement="string" value="java1.5" operator="le"/>
        </DHNRequirements>
        <WSRFService>
            <GARArchive>dvos.authorization-service.gar</GARArchive>
            <BuildFile>build.xml</BuildFile>
            <DeploymentOptions/>
            <ArchitecturalPattern>Singleton</ArchitecturalPattern>
            <WSRFEntry>
                <EntryName>diligentproject/dvos/authorization/OperationAdministrationService</EntryName>
                <Factory value="true"/>
                <WSDL>
                    <p1:definitions></p1:definitions>
                </WSDL>
            </WSRFEntry>
            <WSRFEntry>
                <EntryName>diligentproject/dvos/authorization/VOAdministrationService</EntryName>
                <Factory value="true"/>
                <WSDL>
                    <p1:definitions></p1:definitions>
                </WSDL>
            </WSRFEntry>
            <WSRFEntry>
                <EntryName>diligentproject/dvos/authorization/VOQueryService</EntryName>
                <Factory value="true"/>
                <WSDL>
                    <p1:definitions></p1:definitions>
                </WSDL>
            </WSRFEntry>
        </WSRFService>
    </Package>
    <Package>
        <PackageName>dvos.authorization-api</PackageName>
        <PackageType>Library</PackageType>
        <Version>.9</Version>
```

```
                <DLMandatory value="true"/>
                <DHNMandatory value="true"/>
                <VOMandatory value="true"/>
                <DisposeInterfaceSupport value="true"/>
                <MultiVersionSupport value="false"/>
                <VOSharingSupport value="true"/>
                <Dependencies>
                    <Dependency>
                        <Service>
                            <Class>DVOS</Class>
                            <Name>delegation</Name>
                        </Service>
                        <PackageName>dvos.common</PackageName>
                        <SameDHN value="true"/>
                        <SameDL value="false"/>
                        <SameVO value="false"/>
                        <Priority>4</Priority>
                    </Dependency>
                    <Dependency>
                        <Service>
                            <Class>DVOS</Class>
                            <Name>authorization</Name>
                        </Service>
                        <PackageName>dvos.authorization-stubs</PackageName>
                        <SameDHN value="true"/>
                        <SameDL value="true"/>
                        <SameVO value="true"/>
                        <Priority>4</Priority>
                    </Dependency>
                </Dependencies>
                <DHNRequirements>
                    <Req category="Site" requirement="string" value="java1.5" operator="le"/>
                </DHNRequirements>
                <Library>
            <Type>shared</Type>
                    <LibraryFile>dvos.authorization-api.jar</LibraryFile>
                </Library>
            </Package>
        </PackagesList>
    </Profile>
</DILIGENTResource>
```

A.17. UserGroupManagement Service Profile

```
<?xml version="1.0"?>
<DILIGENTResource xmlns:p1="http://schemas.xmlsoap.org/wsdl/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
    <UniqueID>String</UniqueID>
    <ResourceType>Service</ResourceType>
    <AuthorizationPolicies>Text</AuthorizationPolicies>
    <Profile>
        <Class>DVOS</Class>
        <Name>usergroupmanagement</Name>
        <DescriptiveParameters>
            <DescParameter/>
        </DescriptiveParameters>
        <QoS/>
        <DLDependencies/>
        <SpecificData>Text</SpecificData>
        <PackagesList>
            <Package>
                <PackageName>dvos.usergroupmanagement-stubs</PackageName>
                <PackageType>Library</PackageType>
                <Version>.9</Version>
                <DLMandatory value="false"/>
                <DHNMandatory value="true"/>
                <VOMandatory value="false"/>
                <DisposeInterfaceSupport value="true"/>
                <MultiVersionSupport value="false"/>
                <VOSharingSupport value="true"/>
```

```
            <Dependencies>
                <Dependency>
                    <Service>
                        <Class>DVOS</Class>
                        <Name>delegation</Name>
                    </Service>
                    <PackageName>dvos.common</PackageName>
                    <SameDHN value="true"/>
                    <SameDL value="false"/>
                    <SameVO value="false"/>
                    <Priority>4</Priority>
                </Dependency>
            </Dependencies>
            <DHNRequirements>
                <Req category="Site" requirement="string" value="java1.5" operator="le"/>
            </DHNRequirements>
            <Library>
        <Type>stub</Type>
        <IsStubOf>
                    <PackageName>dvos.usergroupmanagement-service</PackageName>
           <Service>
              <Class>DVOS</Class>
              <Name>usergroupmanagement</Name>
           </Service>
                </IsStubOf>
                <LibraryFile>dvos.usergroupmanagement-stubs.jar</LibraryFile>
            </Library>
        </Package>
        <Package>
            <PackageName>dvos.usergroupmanagement-service</PackageName>
            <PackageType>WSRFService</PackageType>
            <Version>1.0</Version>
            <DLMandatory value="true"/>
            <DHNMandatory value="true"/>
            <VOMandatory value="true"/>
            <DisposeInterfaceSupport value="true"/>
            <MultiVersionSupport value="false"/>
            <VOSharingSupport value="true"/>
            <Dependencies>
                <Dependency>
                    <Service>
                        <Class>DVOS</Class>
                        <Name>delegation</Name>
                    </Service>
                    <PackageName>dvos.common</PackageName>
                    <SameDHN value="true"/>
                    <SameDL value="false"/>
                    <SameVO value="false"/>
                    <Priority>4</Priority>
                </Dependency>
                <Dependency>
                    <Service>
                        <Class>DVOS</Class>
                        <Name>usergroupmanagement</Name>
                    </Service>
                    <PackageName>dvos.usergroupmanagement-stubs</PackageName>
                    <SameDHN value="true"/>
                    <SameDL value="true"/>
                    <SameVO value="true"/>
                    <Priority>4</Priority>
                </Dependency>
            </Dependencies>
            <DHNRequirements>
                <Req category="Site" requirement="string" value="java1.5" operator="le"/>
            </DHNRequirements>
            <WSRFService>
                <GARArchive>dvos.usergroupmanagement-service.gar</GARArchive>
                <BuildFile>build.xml</BuildFile>
                <DeploymentOptions/>
```

```
                    <ArchitecturalPattern>Singleton</ArchitecturalPattern>
                    <WSRFEntry>

    <EntryName>diligentproject/dvos/usergroupmanagement/UserGroupManagementService</EntryName>
                        <Factory value="true"/>
                        <WSDL>
                            <p1:definitions></p1:definitions>
                        </WSDL>
                    </WSRFEntry>
                </WSRFService>
            </Package>
            <Package>
                <PackageName>dvos.usergroupmanagement-api</PackageName>
                <PackageType>Library</PackageType>
                <Version>.9</Version>
                <DLMandatory value="true"/>
                <DHNMandatory value="true"/>
                <VOMandatory value="true"/>
                <DisposeInterfaceSupport value="true"/>
                <MultiVersionSupport value="false"/>
                <VOSharingSupport value="true"/>
                <Dependencies>
                    <Dependency>
                        <Service>
                            <Class>DVOS</Class>
                            <Name>delegation</Name>
                        </Service>
                        <PackageName>dvos.common</PackageName>
                        <SameDHN value="true"/>
                        <SameDL value="false"/>
                        <SameVO value="false"/>
                        <Priority>4</Priority>
                    </Dependency>
                    <Dependency>
                        <Service>
                            <Class>DVOS</Class>
                            <Name>usergroupmanagement</Name>
                        </Service>
                        <PackageName>dvos.usergroupmanagement-stubs</PackageName>
                        <SameDHN value="true"/>
                        <SameDL value="true"/>
                        <SameVO value="true"/>
                        <Priority>4</Priority>
                    </Dependency>
                </Dependencies>
                <DHNRequirements>
                    <Req category="Site" requirement="string" value="java1.5" operator="le"/>
                </DHNRequirements>
                <Library>
            <Type>shared</Type>
                    <LibraryFile>dvos.usergroupmanagement-api.jar</LibraryFile>
                </Library>
            </Package>
        </PackagesList>
    </Profile>
</DILIGENTResource>
```

A.18. VDLDefinitionRepository Service Profile

```
<?xml version="1.0" encoding="UTF-8"?>
<DILIGENTResource>
    <UniqueID>
        <!-- to be assigned by the RegistrationService-->
    </UniqueID>
    <ResourceType>Service</ResourceType>
    <AuthorizationPolicies/>
    <Profile>
        <Class>VDLGeneratorService</Class>
```

```
<Name>DefinitionRepository</Name>
<DescriptiveParameters>
    <param/>
</DescriptiveParameters>
<QoS/>
<DLDependencies/>
<PackagesList>
    <Package>
        <PackageName>DefinitionRepository</PackageName>
        <PackageType>WSRFService</PackageType>
        <Version>1.0</Version>
        <DLMandatory value="0"/>
        <VOMandatory value="1"/>
        <DHNMandatory value="0"/>
        <DisposeInterfaceSupport value="1"/>
        <MultiVersionSupport value="0"/>
        <VOSharingSupport value="1"/>
        <ManifestFile/>
        <InstallScripts/>
        <UninstallScripts/>
        <Parameters/>
        <Dependencies/>
        <DHNRequirements/>
        <ConfigurationFiles/>
        <WSRFService>
            <GARArchive>org_diligentproject_vdlgeneratorservice_definitionrepository.gar</GARArchive>
            <BuildFile>build.xml</BuildFile>
            <DeploymentOptions/>
            <ArchitecturalPattern>Singleton</ArchitecturalPattern>
            <WSRFEntry>
                <EntryName> vdlgeneratorservice/definitionrepository/DefinitionRepositoryService </EntryName>
                <Factory value="false"/>
                <Parameters/>
                <Security name="String">
                    <securityDescriptor/>
                    <defaultIdentity>
                        <subject>String</subject>
                        <CASubject>String</CASubject>
                    </defaultIdentity>
                </Security>
                <WSDL>
                    <definitions name="DefinitionRepositoryService"
targetNamespace="http://www.diligentproject.org/namespaces/vdlgeneratorservice/definitionrepository/DefinitionRepositorySer
vice" xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://www.diligentproject.org/namespaces/vdlgeneratorservice/definitionrepository/DefinitionRepositoryService"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:wsrp="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-
ResourceProperties-1.2-draft-01.xsd" xmlns:wsrpw="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-
1.2-draft-01.wsdl" xmlns:wsdlpp="http://www.globus.org/namespaces/2004/10/WSDLPreprocessor"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
                        <wsdl:import namespace="http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-
ResourceProperties-1.2-draft-01.wsdl" location="../../../wsrf/properties/WS-ResourceProperties.wsdl"/>
                        <types>
                            <xsd:schema
targetNamespace="http://www.diligentproject.org/namespaces/vdlgeneratorservice/definitionrepository/DefinitionRepositorySer
vice"
xmlns:tns="http://www.diligentproject.org/namespaces/vdlgeneratorservice/definitionrepository/DefinitionRepositoryService"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
                                <xsd:complexType name="repositoryMessage">
                                    <xsd:sequence>
                                        <xsd:element type="xsd:string" name="definitionID"/>
                                        <xsd:element type="xsd:string" name="definitionFile"/>
                                    </xsd:sequence>
                                </xsd:complexType>
                                <xsd:complexType name="Map">
                                    <xsd:sequence>
                                        <xsd:element name="item" minOccurs="0" maxOccurs="unbounded">
                                            <xsd:complexType>
                                                <xsd:sequence>
                                                    <xsd:element name="key" type="xsd:string"/>
```

```
                                        <xsd:element name="value" type="xsd:string"/>
                                    </xsd:sequence>
                                </xsd:complexType>
                            </xsd:element>
                        </xsd:sequence>
                    </xsd:complexType>
                    <xsd:element name="DefinitionDB" type="tns:Map"/>
                    <xsd:element name="DefinitionRepositoryResourceProperties">
                        <xsd:complexType>
                            <xsd:sequence>
                                <xsd:element ref="tns:DefinitionDB" minOccurs="1" maxOccurs="1"/>
                            </xsd:sequence>
                        </xsd:complexType>
                    </xsd:element>
                    <xsd:element name="getVDLDefinition" type="xsd:string"/>
                    <xsd:element name="getVDLDefinitionResponse" type="xsd:string"/>
                    <xsd:element name="removeVDLDefinition" type="xsd:string"/>
                    <xsd:element name="removeVDLDefinitionResponse">
                        <xsd:complexType/>
                    </xsd:element>
                    <xsd:element name="addVDLDefinition" type="tns:repositoryMessage"/>
                    <xsd:element name="addVDLDefinitionResponse">
                        <xsd:complexType/>
                    </xsd:element>
                    <xsd:element name="updateVDLDefinition" type="tns:repositoryMessage"/>
                    <xsd:element name="updateVDLDefinitionResponse">
                        <xsd:complexType/>
                    </xsd:element>
                </xsd:schema>
            </types>
            <message name="GetVDLDefinitionInputMessage">
                <part name="parameters" element="tns:getVDLDefinition"/>
            </message>
            <message name="GetVDLDefinitionOutputMessage">
                <part name="parameters" element="tns:getVDLDefinitionResponse"/>
            </message>
            <message name="UpdateVDLDefinitionInputMessage">
                <part name="parameters" element="tns:updateVDLDefinition"/>
            </message>
            <message name="UpdateVDLDefinitionOutputMessage">
                <part name="parameters" element="tns:updateVDLDefinitionResponse"/>
            </message>
            <message name="AddVDLDefinitionInputMessage">
                <part name="parameters" element="tns:addVDLDefinition"/>
            </message>
            <message name="AddVDLDefinitionOutputMessage">
                <part name="parameters" element="tns:addVDLDefinitionResponse"/>
            </message>
            <message name="RemoveVDLDefinitionInputMessage">
                <part name="parameters" element="tns:removeVDLDefinition"/>
            </message>
            <message name="RemoveVDLDefinitionOutputMessage">
                <part name="parameters" element="tns:removeVDLDefinitionResponse"/>
            </message>
            <portType name="DefinitionRepositoryPortType"
wsdlpp:extends="wsrpw:GetResourceProperty
                    wsrpw:GetMultipleResourceProperties
                    wsrpw:SetResourceProperties
                    wsrpw:QueryResourceProperties"
wsrp:ResourceProperties="tns:DefinitionRepositoryResourceProperties">
                <operation name="getVDLDefinition">
                    <input message="tns:GetVDLDefinitionInputMessage"/>
                    <output message="tns:GetVDLDefinitionOutputMessage"/>
                </operation>
                <operation name="updateVDLDefinition">
                    <input message="tns:UpdateVDLDefinitionInputMessage"/>
                    <output message="tns:UpdateVDLDefinitionOutputMessage"/>
                </operation>
                <operation name="addVDLDefinition">
```

```
                                    <input message="tns:AddVDLDefinitionInputMessage"/>
                                    <output message="tns:AddVDLDefinitionOutputMessage"/>
                                </operation>
                                <operation name="removeVDLDefinition">
                                    <input message="tns:RemoveVDLDefinitionInputMessage"/>
                                    <output message="tns:RemoveVDLDefinitionOutputMessage"/>
                                </operation>
                            </portType>
                        </definitions>
                    </WSDL>
                </WSRFEntry>
            </WSRFService>
            <OtherFiles/>
            <OtherProperties/>
        </Package>
        <Package>
            <PackageName>DefinitionRepository_stubs</PackageName>
            <PackageType>Library</PackageType>
            <Version>1.0</Version>
            <DLMandatory value="0"/>
            <VOMandatory value="0"/>
            <DHNMandatory value="0"/>
            <DisposeInterfaceSupport value="0"/>
            <MultiVersionSupport value="0"/>
            <VOSharingSupport value="1"/>
            <ManifestFile/>
            <InstallScripts/>
            <UninstallScripts/>
            <Dependencies/>
            <ConfigurationFiles/>
            <Library>
                <Type>stub</Type>
                <IsStubOf>
                    <PackageName>DefinitionRepository</PackageName>
                    <Service>
                        <Class> VDLGeneratorService </Class>
                        <Name> DefinitionRepository </Name>
                    </Service>
                </IsStubOf>
                <Parameters/>
                <LibraryFile>org_diligentproject_vdlgeneratorservice_definitionrepository_stubs.jar</LibraryFile>
            </Library>
            <OtherFiles/>
            <OtherProperties/>
        </Package>
    </PackagesList>
  </Profile>
</DILIGENTResource>
```

Appendix B. GLUE Schema 1.2 – XML Schema

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- GLUE Schema 1.2 - mapping to XML Schema -->
<!-- Author: Sergio Andreozzi (sergio.andreozzi@cnaf.infn.it) -->
<!-- Institution: INFN - Italy -->
<!-- License: see LICENSE file for EGEE Middleware -->
<!-- Revision number: 1 date: 27 November 2005 -->
<xs:schema xmlns="http://infnforge.cnaf.infn.it/glueinfomodel/Spec/V12/R1" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://infnforge.cnaf.infn.it/glueinfomodel/Spec/V12/R1" elementFormDefault="qualified">
    <!-- simple types definition -->
    <xs:simpleType name="UniqueIDType">
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <xs:simpleType name="LocalIDType">
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <xs:simpleType name="DirType">
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
    <!-- enumerations definition -->
    <xs:simpleType name="ServiceStatusEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="OK"/>
            <xs:enumeration value="Warning"/>
            <xs:enumeration value="Critical"/>
            <xs:enumeration value="Unknown"/>
            <xs:enumeration value="Other"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="JobStatusEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Queued"/>
            <xs:enumeration value="Running"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="CEStatusEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Production"/>
            <xs:enumeration value="Queueing"/>
            <xs:enumeration value="Draining"/>
            <xs:enumeration value="Closed"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="LRMSTypeEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="OpenPBS"/>
            <xs:enumeration value="LSF"/>
            <xs:enumeration value="Condor"/>
            <xs:enumeration value="BQS"/>
            <xs:enumeration value="CondorG"/>
            <xs:enumeration value="FBSNG"/>
            <xs:enumeration value="Torque"/>
            <xs:enumeration value="PBSPro"/>
            <xs:enumeration value="SGE"/>
            <xs:enumeration value="NQE"/>
            <xs:enumeration value="fork"/>
            <xs:enumeration value="other"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="LRMSTypeOpenEnum">
        <xs:union memberTypes="LRMSTypeEnum xs:string"/>
    </xs:simpleType>
    <xs:simpleType name="ServiceTypeEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="org.glite.wms"/>
            <xs:enumeration value="org.glite.rgma.LatestProducer"/>
```

```
            <xs:enumeration value="org.glite.rgma.StreamProducer"/>
            <xs:enumeration value="org.glite.rgma.DBProducer"/>
            <xs:enumeration value="org.glite.rgma.CanonicalProducer"/>
            <xs:enumeration value="org.glite.rgma.Archiver"/>
            <xs:enumeration value="org.glite.rgma.Consumer"/>
            <xs:enumeration value="org.glite.rgma.Registry"/>
            <xs:enumeration value="org.glite.rgma.Schema"/>
            <xs:enumeration value="org.glite.rgma.Browser"/>
            <xs:enumeration value="org.glite.rgma.PrimaryProducer"/>
            <xs:enumeration value="org.glite.rgma.SecondaryProducer"/>
            <xs:enumeration value="org.glite.rgma.OnDemandProducer"/>
            <xs:enumeration value="org.glite.voms"/>
            <xs:enumeration value="org.glite.FiremanCatalog"/>
            <xs:enumeration value="org.glite.SEIndex"/>
            <xs:enumeration value="org.glite.Metadata"/>
            <xs:enumeration value="org.glite.ChannelManagement"/>
            <xs:enumeration value="org.glite.FileTransfer"/>
            <xs:enumeration value="org.glite.FileTransferStats"/>
            <xs:enumeration value="org.glite.ChannelAgent"/>
            <xs:enumeration value="org.glite.KeyStore"/>
            <xs:enumeration value="org.glite.FAS"/>
            <xs:enumeration value="org.glite.gliteIO"/>
            <xs:enumeration value="SRM"/>
            <xs:enumeration value="gsiftp"/>
            <xs:enumeration value="org.edg.local-replica-catalog"/>
            <xs:enumeration value="org.edg.replica-metadata-catalog"/>
            <xs:enumeration value="org.edg.SE"/>
            <xs:enumeration value="it.infn.GridICE"/>
            <xs:enumeration value="MyProxy"/>
            <xs:enumeration value="GUMS"/>
            <xs:enumeration value="GridCat"/>
            <xs:enumeration value="edu.caltech.cacr.monalisa"/>
            <xs:enumeration value="OpenSSH"/>
            <xs:enumeration value="MDS-GIIS"/>
            <xs:enumeration value="BDII"/>
            <xs:enumeration value="RLS"/>
            <xs:enumeration value="data-location-interface"/>
            <xs:enumeration value="pbs.torque.server"/>
            <xs:enumeration value="pbs.torque.maui"/>
            <xs:enumeration value="other"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="ServiceTypeOpenEnum">
        <xs:union memberTypes="ServiceTypeEnum xs:string"/>
    </xs:simpleType>
    <xs:simpleType name="SEAccessProtocolEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="gsiftp"/>
            <xs:enumeration value="nfs"/>
            <xs:enumeration value="afs"/>
            <xs:enumeration value="rfio"/>
            <xs:enumeration value="gsirfio"/>
            <xs:enumeration value="dcap"/>
            <xs:enumeration value="gsidcap"/>
            <xs:enumeration value="root"/>
            <xs:enumeration value="https"/>
            <xs:enumeration value="other"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="SEAccessProtocolOpenEnum">
        <xs:union memberTypes="SEAccessProtocolEnum xs:string"/>
    </xs:simpleType>
    <xs:simpleType name="SEControlProtocolEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="SRM"/>
            <xs:enumeration value="org.edg.SE"/>
            <xs:enumeration value="classic"/>
            <xs:enumeration value="other"/>
        </xs:restriction>
```

```
</xs:simpleType>
<xs:simpleType name="SEControlProtocolOpenEnum">
    <xs:union memberTypes="SEControlProtocolEnum xs:string"/>
</xs:simpleType>
<xs:simpleType name="SEArchitectureEnum">
    <xs:restriction base="xs:string">
        <xs:enumeration value="disk"/>
        <xs:enumeration value="tape"/>
        <xs:enumeration value="multidisk"/>
        <xs:enumeration value="other"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="SATypeEnum">
    <xs:restriction base="xs:string">
        <xs:enumeration value="volatile"/>
        <xs:enumeration value="durable"/>
        <xs:enumeration value="permanent"/>
        <xs:enumeration value="other"/>
    </xs:restriction>
</xs:simpleType>
<!--complex types definition -->
<xs:complexType name="ACLType">
    <xs:sequence>
        <xs:element name="Rule" type="xs:string" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="JobType">
    <xs:sequence>
        <xs:element name="GlobalID" type="xs:string" minOccurs="0"/>
        <xs:element name="LocalOwner" type="xs:string" minOccurs="0"/>
        <xs:element name="GlobalOwner" type="xs:string" minOccurs="0"/>
        <xs:element name="Status" type="JobStatusEnum" minOccurs="0"/>
        <xs:element name="SchedulerSpecific" type="xs:string" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="LocalID" type="LocalIDType" use="required"/>
</xs:complexType>
<xs:complexType name="SEAccessProtocolType">
    <xs:sequence>
        <xs:element name="Endpoint" type="xs:anyURI" minOccurs="0"/>
        <xs:element name="Type" type="SEAccessProtocolOpenEnum" minOccurs="0"/>
        <xs:element name="Version" type="xs:string" minOccurs="0"/>
        <xs:element name="Capability" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="LocalID" type="LocalIDType" use="required"/>
</xs:complexType>
<xs:complexType name="SEControlProtocolType">
    <xs:sequence>
        <xs:element name="Endpoint" type="xs:anyURI" minOccurs="0"/>
        <xs:element name="Type" type="SEControlProtocolOpenEnum" minOccurs="0"/>
        <xs:element name="Version" type="xs:string" minOccurs="0"/>
        <xs:element name="Capability" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="LocalID" type="LocalIDType" use="required"/>
</xs:complexType>
<xs:complexType name="OperatingSystemType">
    <xs:attribute name="Name" type="xs:string"/>
    <xs:attribute name="Release" type="xs:string"/>
    <xs:attribute name="Version" type="xs:string"/>
</xs:complexType>
<xs:complexType name="ProcessorType">
    <xs:attribute name="Vendor" type="xs:string"/>
    <xs:attribute name="Model" type="xs:string"/>
    <xs:attribute name="ClockSpeed" type="xs:integer"/>
    <xs:attribute name="InstructionSet" type="xs:string"/>
    <xs:attribute name="OtherDescription" type="xs:string"/>
</xs:complexType>
<xs:complexType name="ProcessorFullType">
    <xs:complexContent>
        <xs:extension base="ProcessorType">
```

```
                <xs:attribute name="CacheL1" type="xs:integer"/>
                <xs:attribute name="CacheL1I" type="xs:integer"/>
                <xs:attribute name="CacheL1D" type="xs:integer"/>
                <xs:attribute name="CacheL2" type="xs:integer"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="LoadType">
        <xs:attribute name="Last1Min" type="xs:integer"/>
        <xs:attribute name="Last5Min" type="xs:integer"/>
        <xs:attribute name="Last15Min" type="xs:integer"/>
    </xs:complexType>
    <xs:complexType name="HostArchitectureType">
        <xs:attribute name="PlatformType" type="xs:string"/>
        <xs:attribute name="SMPSize" type="xs:integer"/>
    </xs:complexType>
    <xs:complexType name="HostArchitectureFullType">
        <xs:complexContent>
            <xs:extension base="HostArchitectureType">
                <xs:attribute name="SMTSize" type="xs:integer"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="MainMemoryType">
        <xs:attribute name="RAMSize" type="xs:integer"/>
        <xs:attribute name="VirtualSize" type="xs:integer"/>
    </xs:complexType>
    <xs:complexType name="MainMemoryFullType">
        <xs:complexContent>
            <xs:extension base="MainMemoryType">
                <xs:attribute name="RAMAvailable" type="xs:integer"/>
                <xs:attribute name="VirtualAvailable" type="xs:integer"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="NetworkAdapterType">
        <xs:attribute name="InboundIP" type="xs:boolean"/>
        <xs:attribute name="OutboundIP" type="xs:boolean"/>
    </xs:complexType>
    <xs:complexType name="NetworkAdapterFullType">
        <xs:complexContent>
            <xs:extension base="NetworkAdapterType">
                <xs:attribute name="Name" type="xs:string"/>
                <xs:attribute name="IPAddress" type="xs:string"/>
                <xs:attribute name="MTU" type="xs:integer"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="BenchmarkType">
        <xs:attribute name="SI00" type="xs:integer"/>
        <xs:attribute name="SF00" type="xs:integer"/>
    </xs:complexType>
    <xs:complexType name="RunTimeEnvType">
        <xs:sequence>
            <xs:element name="Variable" type="xs:string" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="FileSystemType">
        <xs:attribute name="Name" type="xs:string"/>
        <xs:attribute name="Root" type="xs:string"/>
        <xs:attribute name="Size" type="xs:integer"/>
        <xs:attribute name="AvailableSpace" type="xs:integer"/>
        <xs:attribute name="ReadOnly" type="xs:boolean"/>
        <xs:attribute name="Type" type="xs:string"/>
    </xs:complexType>
    <xs:complexType name="StorageDeviceType">
        <xs:attribute name="Name" type="xs:string"/>
        <xs:attribute name="Type" type="xs:string"/>
        <xs:attribute name="TransferRate" type="xs:integer"/>
```

```
        <xs:attribute name="Size" type="xs:integer"/>
    </xs:complexType>
    <xs:complexType name="StoragePartitionType">
        <xs:attribute name="Name" type="xs:string"/>
        <xs:attribute name="Size" type="xs:string"/>
        <xs:attribute name="RateRate" type="xs:integer"/>
        <xs:attribute name="WriteRate" type="xs:integer"/>
    </xs:complexType>
    <xs:complexType name="CEVOViewType">
        <xs:sequence>
            <xs:element name="ACL" type="ACLType" minOccurs="0"/>
            <xs:element name="RunningJobs" type="xs:integer" minOccurs="0"/>
            <xs:element name="WaitingJobs" type="xs:integer" minOccurs="0"/>
            <xs:element name="TotalJobs" type="xs:integer" minOccurs="0"/>
            <xs:element name="FreeJobSlots" type="xs:integer" minOccurs="0"/>
            <xs:element name="EstimatedResponseTime" type="xs:integer" minOccurs="0"/>
            <xs:element name="WorstResponseTime" type="xs:integer" minOccurs="0"/>
            <xs:element name="DefaultSE" type="xs:string" minOccurs="0"/>
            <xs:element name="ApplicationDir" type="xs:string" minOccurs="0"/>
            <xs:element name="DataDir" type="xs:string" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="LocalID" type="LocalIDType" use="required"/>
    </xs:complexType>
    <xs:complexType name="ComputingElementType">
        <xs:sequence>
            <xs:element name="InformationServiceURL" type="xs:anyURI" minOccurs="0"/>
            <xs:element name="Name" type="xs:string" minOccurs="0"/>
            <xs:element name="LRMSType" type="LRMSTypeOpenEnum" minOccurs="0"/>
            <xs:element name="LRMSVersion" type="xs:string" minOccurs="0"/>
            <xs:element name="GRAMVersion" type="xs:string" minOccurs="0"/>
            <xs:element name="HostName" type="xs:string" minOccurs="0"/>
            <xs:element name="GateKeeperPort" type="xs:integer" minOccurs="0"/>
            <xs:element name="JobManager" type="xs:string" minOccurs="0"/>
            <xs:element name="ContactString" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="ApplicationDir" type="DirType" minOccurs="0"/>
            <xs:element name="DataDir" type="DirType" minOccurs="0"/>
            <xs:element name="DefaultSE" type="xs:string" minOccurs="0"/>
            <xs:element name="Status" type="CEStatusEnum" minOccurs="0"/>
            <xs:element name="RunningJobs" type="xs:integer" minOccurs="0"/>
            <xs:element name="WaitingJobs" type="xs:integer" minOccurs="0"/>
            <xs:element name="TotalJobs" type="xs:integer" minOccurs="0"/>
            <xs:element name="EstimatedResponseTime" type="xs:integer" minOccurs="0"/>
            <xs:element name="WorstResponseTime" type="xs:integer" minOccurs="0"/>
            <xs:element name="FreeJobSlots" type="xs:integer" minOccurs="0"/>
            <xs:element name="MaxWallClockTime" type="xs:integer" minOccurs="0"/>
            <xs:element name="MaxCPUTime" type="xs:integer" minOccurs="0"/>
            <xs:element name="MaxTotalJobs" type="xs:integer" minOccurs="0"/>
            <xs:element name="MaxRunningJobs" type="xs:integer" minOccurs="0"/>
            <xs:element name="Priority" type="xs:integer" minOccurs="0"/>
            <xs:element name="AssignedJobSlots" type="xs:integer" minOccurs="0"/>
            <xs:element name="ACL" type="ACLType" minOccurs="0"/>
            <xs:element name="Job" type="JobType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="VOView" type="CEVOViewType" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="UniqueID" type="UniqueIDType" use="required"/>
    </xs:complexType>
    <xs:complexType name="SubClusterType">
        <xs:sequence>
            <xs:element name="Name" type="xs:string" minOccurs="0"/>
            <xs:element name="PhysicalCPUs" type="xs:integer" minOccurs="0"/>
            <xs:element name="LogicalCPUs" type="xs:integer" minOccurs="0"/>
            <xs:element name="TmpDir" type="DirType" minOccurs="0"/>
            <xs:element name="WNTmpDir" type="DirType" minOccurs="0"/>
            <xs:element name="OperatingSystem" type="OperatingSystemType" minOccurs="0"/>
            <xs:element name="Processor" type="ProcessorType" minOccurs="0"/>
            <xs:element name="NetworkAdapter" type="NetworkAdapterType" minOccurs="0"/>
            <xs:element name="MainMemory" type="MainMemoryType" minOccurs="0"/>
            <xs:element name="Architecture" type="HostArchitectureType" minOccurs="0"/>
            <xs:element name="Benchmark" type="BenchmarkType" minOccurs="0"/>
```

```
            <xs:element name="Location" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Name" type="xs:string" minOccurs="0"/>
                        <xs:element name="Version" type="xs:string" minOccurs="0"/>
                        <xs:element name="Path" type="xs:string" minOccurs="0"/>
                    </xs:sequence>
                    <xs:attribute name="LocalID" type="LocalIDType" use="required"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="RunTimeEnv" type="RunTimeEnvType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="UniqueID" type="UniqueIDType" use="required"/>
    </xs:complexType>
    <xs:complexType name="StorageAreaType">
        <xs:sequence>
            <xs:element name="Path" type="DirType" minOccurs="0"/>
            <xs:element name="Type" type="SATypeEnum" minOccurs="0"/>
            <xs:element name="Quota" type="xs:integer" minOccurs="0"/>
            <xs:element name="MinFileSize" type="xs:integer" minOccurs="0"/>
            <xs:element name="MaxFileSize" type="xs:integer" minOccurs="0"/>
            <xs:element name="MaxData" type="xs:integer" minOccurs="0"/>
            <xs:element name="MaxNumFiles" type="xs:integer" minOccurs="0"/>
            <xs:element name="MaxPinDuration" type="xs:integer" minOccurs="0"/>
            <xs:element name="UsedSpace" type="xs:integer" minOccurs="0"/>
            <xs:element name="AvailableSpace" type="xs:integer" minOccurs="0"/>
            <xs:element name="ACL" type="ACLType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="LocalID" type="LocalIDType" use="required"/>
    </xs:complexType>
    <xs:complexType name="CESEBindType">
        <xs:attribute name="CEUniqueID" type="UniqueIDType" use="required"/>
        <xs:attribute name="SEUniqueID" type="UniqueIDType" use="required"/>
        <xs:attribute name="MountInfo" type="xs:string" use="optional"/>
        <xs:attribute name="Weight" type="xs:integer" use="optional"/>
    </xs:complexType>
    <xs:complexType name="Service2ServiceType">
        <xs:attribute name="Service1UniqueID" type="UniqueIDType" use="required"/>
        <xs:attribute name="Service2UniqueID" type="UniqueIDType" use="required"/>
    </xs:complexType>
    <!-- complex elements definition -->
    <xs:element name="Site">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Name" type="xs:string" minOccurs="0"/>
                <xs:element name="Description" type="xs:string" minOccurs="0"/>
                <xs:element name="UserSupportContact" type="xs:string" minOccurs="0"/>
                <xs:element name="SysAdminContact" type="xs:string" minOccurs="0"/>
                <xs:element name="SecurityContact" type="xs:string" minOccurs="0"/>
                <xs:element name="Location" type="xs:string" minOccurs="0"/>
                <xs:element name="Latitude" type="xs:double" minOccurs="0"/>
                <xs:element name="Longitude" type="xs:double" minOccurs="0"/>
                <xs:element name="Web" type="xs:anyURI" minOccurs="0"/>
                <xs:element name="Sponsor" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="OtherInfo" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="Cluster" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="StorageElement" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="Service" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="CESEBind" type="CESEBindType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="Service2Service" type="Service2ServiceType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element ref="Host" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="UniqueID" type="UniqueIDType" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="Service">
        <xs:complexType>
            <xs:sequence>
```

```
                <xs:element name="Name" type="xs:string" minOccurs="0"/>
                <xs:element name="Type" type="ServiceTypeOpenEnum" minOccurs="0"/>
                <xs:element name="Version" type="xs:string" minOccurs="0"/>
                <xs:element name="Endpoint" type="xs:anyURI" minOccurs="0"/>
                <xs:element name="Status" type="ServiceStatusEnum" minOccurs="0"/>
                <xs:element name="StatusInfo" type="xs:string" minOccurs="0"/>
                <xs:element name="WSDL" type="xs:anyURI" minOccurs="0"/>
                <xs:element name="Semantics" type="xs:anyURI" minOccurs="0"/>
                <xs:element name="StartTime" type="xs:dateTime" minOccurs="0"/>
                <xs:element name="Owner" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="Data" minOccurs="0" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Key" type="xs:string"/>
                            <xs:element name="Value" type="xs:string"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
            <xs:attribute name="UniqueID" type="UniqueIDType" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="Cluster">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Name" type="xs:string" minOccurs="0"/>
                <xs:element name="TmpDir" type="DirType" minOccurs="0"/>
                <xs:element name="WNTmpDir" type="DirType" minOccurs="0"/>
                <xs:element name="ComputingElement" type="ComputingElementType" maxOccurs="unbounded"/>
                <xs:element name="SubCluster" type="SubClusterType" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="UniqueID" type="UniqueIDType" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="StorageElement">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="InformationServiceURL" type="xs:anyURI" minOccurs="0"/>
                <xs:element name="SizeTotal" type="xs:integer" minOccurs="0"/>
                <xs:element name="SizeFree" type="xs:integer" minOccurs="0"/>
                <xs:element name="Architecture" type="SEArchitectureEnum" minOccurs="0"/>
                <xs:element name="StorageArea" type="StorageAreaType" maxOccurs="unbounded"/>
                <xs:element name="AccessProtocol" type="SEAccessProtocolType" maxOccurs="unbounded"/>
                <xs:element name="ControlProtocol" type="SEControlProtocolType" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="UniqueID" type="UniqueIDType" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="Host">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Name" type="xs:string" minOccurs="0"/>
                <xs:element name="UpTime" type="xs:integer" minOccurs="0"/>
                <xs:element name="Architecture" type="HostArchitectureFullType" minOccurs="0"/>
                <xs:element name="MainMemory" type="MainMemoryFullType" minOccurs="0"/>
                <xs:element name="OperatingSystem" type="OperatingSystemType" minOccurs="0"/>
                <xs:element name="Processor" type="ProcessorFullType" minOccurs="0"/>
                <xs:element name="Load" type="LoadType" minOccurs="0"/>
                <xs:element name="NetworkAdapter" type="NetworkAdapterFullType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="Benchmark" type="BenchmarkType" minOccurs="0"/>
                <xs:element name="RunTimeEnv" type="RunTimeEnvType" minOccurs="0"/>
                <xs:element name="StorageDevice" type="StorageDeviceType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="StoragePartition" type="StoragePartitionType" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="LocalFileSystem" type="FileSystemType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="RemoteFileSystem" type="FileSystemType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="StorageDevice2StoragePartition" minOccurs="0" maxOccurs="unbounded">
```

```
                    <xs:complexType>
                        <xs:attribute name="StorageDeviceName" type="xs:string" use="required"/>
                        <xs:attribute name="StoragePartitionName" type="xs:string" use="required"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="StoragePartition2FileSystem" minOccurs="0" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:attribute name="StoragePartitionName" type="xs:string" use="required"/>
                        <xs:attribute name="FileSystemName" type="xs:string" use="required"/>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
            <xs:attribute name="UniqueID" type="UniqueIDType" use="required"/>
        </xs:complexType>
        <xs:key name="StorageDeviceNameKey">
            <!-- the storage device name is unique within an host -->
            <xs:selector xpath=".//StorageDevice"/>
            <xs:field xpath="@Name"/>
        </xs:key>
        <xs:key name="StoragePartitionNameKey">
            <!-- the storage partition name is unique within an host -->
            <xs:selector xpath=".//StoragePartition"/>
            <xs:field xpath="@Name"/>
        </xs:key>
        <xs:key name="FileSystemNameKey">
            <!-- the file system  name is unique within an host -->
            <xs:selector xpath=".//LocalFileSystem | .//RemoteFileSystem"/>
            <xs:field xpath="@Name"/>
        </xs:key>
        <xs:key name="NetworkAdapterNameKey">
            <!-- the file system  name is unique within an host -->
            <xs:selector xpath=".//NetworkAdapter"/>
            <xs:field xpath="@Name"/>
        </xs:key>
        <xs:keyref name="StoragePartition2FileSystemSP" refer="StoragePartitionNameKey">
            <xs:selector xpath=".//StoragePartition2FileSystem"/>
            <xs:field xpath="@StoragePartitionName"/>
        </xs:keyref>
        <xs:keyref name="StoragePartition2FileSystemFS" refer="FileSystemNameKey">
            <xs:selector xpath=".//StoragePartition2FileSystem"/>
            <xs:field xpath="@FileSystemName"/>
        </xs:keyref>
        <xs:keyref name="StorageDevice2StoragePartitionSD" refer="StorageDeviceNameKey">
            <xs:selector xpath=".//StorageDevice2StoragePartition"/>
            <xs:field xpath="@StorageDeviceName"/>
        </xs:keyref>
        <xs:keyref name="StorageDevice2StoragePartitionSP" refer="StoragePartitionNameKey">
            <xs:selector xpath=".//StorageDevice2StoragePartition"/>
            <xs:field xpath="@StoragePartitionName"/>
        </xs:keyref>
    </xs:element>
</xs:schema>
```

Appendix C. Configuration files

C.1. DISQueries file. XQuery template used by DIS-HLS-Client Library

```
<?xml version="1.0"?>
<DIS-QUERIES>

  <!-- PARAMETERS in PARAMETRIC QUERIES ARE SPECIFIED WITH the following syntax *param name* -->

  <!-- QUERIES ON Profiles -->

  <QUERY>
    <!-- retrieve all resource profiles -->
    <ID>getAllProfileEntries</ID>
    <VALUE>for $profile in collection("/db/Profiles")//Document/Data/child::*[local-name()='Profile']/DILIGENTResource/Profile
return $profile</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve all resource profiles identifiers -->
    <ID>getAllProfileIDs</ID>
    <VALUE>for $profileID in collection("/db/Profiles")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource/UniqueID return $profileID</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve all the published service profiles -->
    <ID>getAllProfileService</ID>
    <VALUE>for $serviceProfile in collection("/db/Profiles/Service")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $serviceProfile/ResourceType/text() eq 'Service' return $serviceProfile</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve all the published compound service profiles -->
    <ID>getAllProfileCS</ID>
    <VALUE>for $CSProfile in collection("/db/Profiles/CS")//Document/Data/child::*[local-name()='Profile']/DILIGENTResource
where $CSProfile/ResourceType/text() eq 'CS' return $CSProfile</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve all the published External Running Instances profiles -->
    <ID>getAllProfileCSInstance</ID>
    <VALUE>for $CSInstanceProfile in collection("/db/Profiles/CSInstance")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $CSInstanceProfile/ResourceType/text() eq 'CSInstance' return
$CSInstanceProfile</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve all the published collection profiles -->
    <ID>getAllProfileCollection</ID>
    <VALUE>for $collectionProfile in collection("/db/Profiles/Collection")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $collectionProfile/ResourceType/text() eq 'Collection' return
$collectionProfile</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve all the published running instance profiles -->
    <ID>getAllProfileRunningInstance</ID>
    <VALUE>for $riProfile in collection("/db/Profiles/RunningInstance")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $riProfile/ResourceType/text() eq 'RunningInstance' return $riProfile</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve all the published DHN profiles -->
    <ID>getAllProfileDHN</ID>
    <VALUE>for $dhnProfile in collection("/db/Profiles/DHN")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $dhnProfile/ResourceType/text() eq 'DHN' return $dhnProfile</VALUE>
  </QUERY>
```

```
  <QUERY>
    <!-- retrieve all the published gLite profiles -->
    <ID>getAllProfilegLiteResource</ID>
    <VALUE>for $gLiteResourceProfile in collection("/db/Profiles/gLiteResource")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $gLiteResourceProfile/ResourceType/text() eq 'gLiteResource' return
$gLiteResourceProfile</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve all the published External Running Instances profiles -->
    <ID>getAllProfileExternalRunningInstance</ID>
    <VALUE>for $eRIProfile in collection("/db/Profiles/ExternalRunningInstance")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $eRIProfile/ResourceType/text() eq 'ExternalRunningInstance' return
$eRIProfile</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve all the published service Identifiers -->
    <ID>getAllProfileServiceIDs</ID>
    <VALUE>for $serviceProfile in collection("/db/Profiles/Service")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $serviceProfile/ResourceType/text() eq 'Service' return
$serviceProfile/UniqueID</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve all the published compound service Identifiers -->
    <ID>getAllProfileCSIDs</ID>
    <VALUE>for $CSProfile in collection("/db/Profiles/CS")//Document/Data/child::*[local-name()='Profile']/DILIGENTResource
where $CSProfile/ResourceType/text() eq 'CS' return $CSProfile/UniqueID</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve all the published External Running Instances Identifiers -->
    <ID>getAllProfileCSInstanceIDs</ID>
    <VALUE>for $CSInstanceProfile in collection("/db/Profiles/CSInstance")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $CSInstanceProfile/ResourceType/text() eq 'CSInstance' return
$CSInstanceProfile/UniqueID</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve all the published collection Identifiers -->
    <ID>getAllProfileCollectionIDs</ID>
    <VALUE>for $collectionProfile in collection("/db/Profiles/Collection")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $collectionProfile/ResourceType/text() eq 'Collection' return
$collectionProfile/UniqueID</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve all the published running instance Identifiers -->
    <ID>getAllProfileRunningInstanceIDs</ID>
    <VALUE>for $riProfile in collection("/db/Profiles/RunningInstance")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $riProfile/ResourceType/text() eq 'RunningInstance' return
$riProfile/UniqueID</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve all the published DHN Identifiers -->
    <ID>getAllProfileDHNIDs</ID>
    <VALUE>for $dhnProfile in collection("/db/Profiles/DHN")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $dhnProfile/ResourceType/text() eq 'DHN' return $dhnProfile/UniqueID</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve all the published gLite Identifiers -->
    <ID>getAllProfilegLiteResourceIDs</ID>
    <VALUE>for $gLiteResourceProfile in collection("/db/Profiles/gLiteResource")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $gLiteResourceProfile/ResourceType/text() eq 'gLiteResource' return
$gLiteResourceProfile/UniqueID</VALUE>
```

```
  </QUERY>

  <QUERY>
    <!-- retrieve all the published External Running Instances Identifiers -->
    <ID>getAllProfileExternalRunningInstanceIDs</ID>
    <VALUE>for $eRIProfile in collection("/db/Profiles/ExternalRunningInstance")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $eRIProfile/ResourceType/text() eq 'ExternalRunningInstance' return
$eRIProfile/UniqueID</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve the Resource via the specified ID *ID* -->
    <ID>getProfileFromID</ID>
    <VALUE>for $DILIGENTResource in collection("/db/Profiles")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $DILIGENTResource/UniqueID/text() eq '*ID*' return
$DILIGENTResource/Profile</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve the Resource via the specified ID *ID* -->
    <ID>getResourceTypeFromID</ID>
    <VALUE>for $DILIGENTResource in collection("/db/Profiles")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $DILIGENTResource/UniqueID/text() eq '*ID*' return
$DILIGENTResource/ResourceType</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieve the Resource via the specified ID *ID* -->
    <ID>getAuthPoliciesFromID</ID>
    <VALUE>for $DILIGENTResource in collection("/db/Profiles")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $DILIGENTResource/UniqueID/text() eq '*ID*' return
$DILIGENTResource/AuthorizationPolicies</VALUE>
  </QUERY>

  <QUERY>
    <!-- executes the *XPATH* over the Profile section -->
    <ID>wrappedQueryProfiles</ID>
    <VALUE>for $doc in collection("/db/Profiles")//Document/Data/child::*[local-name()='Profile']/DILIGENTResource where
$doc/child::*[local-name()='Profile']*XPATH* return $doc</VALUE>
  </QUERY>

  <QUERY>
    <!-- executes the *XPATH* over the Profile section of the *TYPE* resource -->
    <ID>wrappedQueryResourceTypeProfiles</ID>
    <VALUE>for $doc in collection("/db/Profiles/*TYPE*")//Document/Data/child::*[local-name()='Profile']/DILIGENTResource
where $doc/child::*[local-name()='Profile']*XPATH* return $doc</VALUE>
  </QUERY>

  <QUERY>
    <!-- executes the *XPATH* over the Profile section of the *TYPE* resource -->
    <ID>queryProfilesForID</ID>
    <VALUE>for $doc in collection("/db/Profiles")//Document/Data/child::*[local-name()='Profile']/DILIGENTResource where
$doc/Profile*XPATH* return $doc/UniqueID</VALUE>
  </QUERY>

  <QUERY>
    <!-- executes the *XPATH* over Data section of the Profiles -->
    <ID>queryRegistryDIS</ID>
    <VALUE>for $doc in collection("/db/Profiles")//Document/Data/child::*[local-name()='Profile']/DILIGENTResource/Profile
where $doc*XPATH* return $doc</VALUE>
  </QUERY>

  <QUERY>
    <!-- returns the Endpoint of the RI Profiles having ServiceName = *NAME*, ServiceClass = *CLASS*, and EntryName =
*ENTRYNAME* -->
    <ID>getEPRsRIFromClassAndName</ID>
    <VALUE>for $riProfile in collection("/db/Profiles/RunningInstance")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource/Profile where $riProfile/ServiceName/text() eq '*NAME*' and $riProfile/ServiceClass/text()
```

```
eq '*CLASS*' return
$riProfile/AccessPoint/RunningInstanceInterfaces/Endpoint[@EntryName="*ENTRYNAME*"]/text()</VALUE>
  </QUERY>

  <QUERY>
    <!-- returns the Endpoint of the RI Profiles having ServiceName = *NAME*, ServiceClass = *CLASS*, and EntryName =
*ENTRYNAME* -->
    <ID>getRISpecificData</ID>
<!--     <VALUE>for $riProfile in collection("/db/Profiles/RunningInstance")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource/Profile where $riProfile/ServiceName/text() eq '*NAME*' and $riProfile/ServiceClass/text()
eq '*CLASS*' and $riProfile/AccessPoint/RunningInstanceInterfaces/Endpoint[@EntryName="*ENTRYNAME*"] return
$riProfile/SpecificData</VALUE> -->
    <VALUE>for $riProfile in collection("/db/Profiles/RunningInstance")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource/Profile[AccessPoint/RunningInstanceInterfaces/Endpoint[@EntryName="*ENTRYNAME*
"]] where $riProfile/ServiceName/text() eq '*NAME*' and $riProfile/ServiceClass/text() eq '*CLASS*' return
$riProfile/SpecificData</VALUE>
  </QUERY>

   <QUERY>
    <!-- returns the required Elements in a profile with given ID and namespace-->
    <ID>getElementsFromIDProfile</ID>

    <VALUE>for $profile in collection("/db/Profiles")//Document where $profile/ID/text() eq '*ID*' return
$profile/Data/child::*[local-name()='Profile']/DILIGENTResource/Profile/.//*[local-name()='*ELEMENTNAME*'][namespace-
uri()='*ELEMENTNAMESPACE*']</VALUE>
  </QUERY>


  <!-- QUERIES ON Properties -->

  <QUERY>
    <!-- retrieves all the published WS-ResourceProperties matching the *XPATH* -->
    <ID>getPropertiesByXPath</ID>
    <VALUE>for $doc in collection("/db/Properties")//Document/Data where $doc*XPATH* return $doc</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieves all the published WS-ResourceProperties  -->
    <ID>getAllPublishedEntries</ID>
    <!--<VALUE>for $doc in collection("/db/Properties")//Document/Source return $doc</VALUE>-->
    <VALUE>for $doc in collection("/db/Properties")//Document/Data return $doc</VALUE>
  </QUERY>

  <QUERY>
    <!-- retrieves all the published WS-ResourceProperties  -->
    <!-- the *KEY* part is appropriately managed by the java code -->
    <ID>getResourceEntries</ID>
    <VALUE>for $doc in collection("/db/Properties")//Document[Source="*ADDR*"]*KEY*/Data return $doc</VALUE>
  </QUERY>

   <QUERY>
    <!-- retrieves all the published WS-ResourceProperties + metadata Information  -->
    <ID>getAllPublishedPropertiesEntries</ID>
    <VALUE>for $doc in collection("/db/Properties")//Document return $doc</VALUE>
  </QUERY>

   <QUERY>
    <!-- retrieves all EPR that exported the specified WS-Properties  -->
    <ID>getEPRFromPropertiesName</ID>
    <VALUE>for $doc in collection("/db/Properties")//Document where $doc/Data/child::*[local-name()='*NAME*'] return
$doc</VALUE>
   </QUERY>

   <QUERY>
    <!-- retrieves all EPR that exported the specified WS-Properties  -->
    <ID>getEPRFromPropertiesValue</ID>
    <VALUE>for $doc in collection("/db/Properties")//Document where $doc/Data/child::*[local-name()='*NAME*']/text() eq
*VALUE*  return $doc</VALUE>
   </QUERY>
```

```
  <QUERY>
    <!-- retrieves all EPR that exported the specified WS-Properties  -->
    <ID>getIDandTimeFromPropertiesName</ID>
    <VALUE>for $doc in collection("/db/Properties")//Document where $doc/Data/child::*[local-name()='*NAME*'] return
<RESULT>{$doc/ID}{$doc/LastUpdateHuman}</RESULT></VALUE>
   </QUERY>

   <QUERY>
    <!-- retrieves all EPR that exported the specified WS-Properties  -->
    <ID>getEPRFromPropertiesNameAndNamespace</ID>
    <VALUE>for $doc in collection("/db/Properties")//Document where $doc/Data/child::*[local-name()='*NAME*'][namespace-
uri()='*NameSpace*']  return $doc</VALUE>
   </QUERY>

   <QUERY>
    <!-- retrieves all EPR that exported the specified WS-Properties  -->
    <ID>getEPRFromPropertiesValueAndNamespace</ID>
    <VALUE>for $doc in collection("/db/Properties")//Document where $doc/Data/child::*[local-name()='*NAME*'][namespace-
uri()='*NameSpace*']/text() eq *VALUE*  return $doc</VALUE>
   </QUERY>

   <QUERY>
    <!-- retrieves all RunningInstance ID given the DHN ID  -->
    <ID>getAllRunningInstancesOnDHN</ID>
    <VALUE>for $doc in collection("/db/Profiles/RunningInstance")//Document/Data/child::*[local-
name()='Profile']/DILIGENTResource where $doc/child::*[local-name()='Profile']/DHN[@UniqueID='*VALUE*'] return
$doc/UniqueID</VALUE>
   </QUERY>

</DIS-QUERIES>
```

C.2. DLMap file

```
<?xml version="1.0" encoding="UTF-8"?>
    <DLMap xmlns:wsa=http://schemas.xmlsoap.org/ws/2004/03/addressing
xmlns:tns="http://www.globus.org/namespaces/examples/core/FactoryService">
    <Global>
        <DLName value=""/>
        <Community value=""/>
        <DLManager value=""/>
        <DLDesigner value=""/>
        <DLStartTime value="xsd:dateTime"/>
        <DLTerminationTime value="xsd:dateTime"/>
        <DLMapExpireTime value="xsd:dateTime"/>
    </Global>
    <Sub-VO>
        <Name value=""/>
        <Infrastructure value=""/>
    </Sub-VO>
    <ApplicationList>
        <RunningInstance>
            <DiligentServiceName value=""/>
            <!– retrieved from the VDL Generator list -->
            <DiligentComponentName value=""/>
            <!– retrieved from the VDL Generator list -->
            <Role value=""/>
            <!– retrieved from the VDL Generator list-->
            <JointTime value=""/>
            <!– when the instance joint the DL-->
            <CreationTime value=""/>
            <!– when the instance was created–>
            <AccessPoint>
                <ArchitecturalPattern>Factory</ArchitecturalPattern>
                <EnpointReference>
                    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
                        <xs:import schemaLocation="http://www.w3.org/2005/08/addressing/ws-addr.xsd"/>
                    </xs:schema>
                </EndpointReference>
                <FactoryURI>http://localhost:8080/wsrf/services/factory/QueryParserFactoryService</FactoryURI>
            </AccessPoint>
            <Configuration>
                <Parameter name="" value=""/>
                <!–any deployment parameter (i.e. indexed metadata for the indices) -->
            </Configuration>
        </RunningInstance>
    </ApplicationList>
    <DeployedPackages>
        <Package name="" DHN=""/>
    </DeployedPackages>
    <Security>  </Security>
</DLMap>
```

Appendix D. Broker & MatchMaker Request and Response XML Schema

D.1. BMM_Request

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:annotation>
        <xs:documentation xml:lang="en">
        XML Schema for Requests from Keeper to BMM
        Version 0.01
        Last modified: 2006-09-01
        Official Location:
        Contact: http://www.diligentproject.org
        </xs:documentation>
    </xs:annotation>

    <xs:include schemaLocation="ServiceClasses_list.xsd"/>

<!-- definition of types -->

    <xs:simpleType name="OpType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="eq"/>
            <xs:enumeration value="ne"/>
            <xs:enumeration value="gt"/>
            <xs:enumeration value="ge"/>
            <xs:enumeration value="lt"/>
            <xs:enumeration value="le"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="type">
        <xs:restriction base="xs:string">
            <xs:enumeration value="CPUSpeed"/>
            <xs:enumeration value="MemAmt"/>
            <xs:enumeration value="new"/>
            <xs:enumeration value="reuse"/>
            <xs:enumeration value="reuseIfAvailable"/>
        </xs:restriction>
    </xs:simpleType>

<!-- definition of complex elements -->

    <xs:element name="other_req">
        <xs:complexType>
            <xs:annotation>TBD</xs:annotation>
        </xs:complexType>
    </xs:element>

    <xs:element name="req">
        <xs:complexType>
            <xs:attribute ref="type" use="required">
            <xs:attribute ref="OpType"/>
            <xs:attribute name="value" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>

    <xs:element name="pkg_req">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="req" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="triple_id" type="IDREF" use="required"/>
        </xs:complexType>
    </xs:element>
```

```
    <xs:element name="DL_Req">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="pkg_req" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="other_req" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="triple">
        <xs:complexType>
            <xs:attribute name="id" type="ID" use"required"/> <!-- si può anche pensare come integer -->
            <xs:attribute name="service_name" type="xs:string" use="required"/>
            <xs:attribute name="service_class" type="ServiceClassEnum" use="required"/>
            <xs:attribute name="package" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>

    <xs:element name="packages">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="triple" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

<!-- root element -->

    <xs:element name="request">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="packages"/>
                <xs:element ref="DL_Req" minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

</xs:schema>
```

D.2. BMM_Response

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
    <xs:annotation>
        <xs:documentation xml:lang="en">
       XML Schema for Responses from BMM to Keeper
        Version 0.01
        Last modified: 2006-09-08
        Official Location:
        Contact: http://www.diligentproject.org
        </xs:documentation>
    </xs:annotation>

    <xs:element name="response">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="match" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:attribute name="IDRequest" type="xs:string" use="required"/>
                        <xs:attribute name="package" type="xs:string" use="required"/>
                        <xs:attribute name="DHN" type="xs:string" use="required"/>
                    </xs:complexType>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

</xs:schema>
```

References

[1] Andrade P., Candela L., Pagano P., Simi M. *D1.1.2 Architectural Specification*. DILIGENT Project Deliverable.
[2] Andreozzi S., Burke S., Field L., Fisher S., Kónya B., Mambelli M., Schopf J. M., Viljoen M., Wilson A. *GLUE Schema Specification, version 1.2, 3 Dec. 2005* http://infnforge.cnaf.infn.it/glueinfomodel/uploads/Spec/GLUEInfoModel_1_2_final.pdf
[3] Apache Software Foundation. The Apache XML Project. http://xml.apache.org/
[4] Apache Software Foundation. The Logging Services Project. http://logging.apache.org/
[5] Avancini H., Candela L., Fabriani P., Pagano P., Roccetti P., Simi M. *D1.2.1 DL Creation & Management services specification interim report*. DILIGENT Project Deliverable. June 2005.
[6] Avancini H., Candela L., Fabriani P., Pagano P., Roccetti P., Simi M. *D1.2.2 DL Creation & Management services specification report*. DILIGENT Project Deliverable. September 2005.
[7] Czajkowski K., Ferguson D. F., Foster I., Frey J., Graham S., Sedukin I., Snelling D., Tuecke S., and Vambenepe W. *The WS-Resource Framework*. White paper, 2004.
[8] EGEE. *EGEE Middleware Architecture*. Deliverable DJRA1.1, July 2004. https://edms.cern.ch/file/476451
[9] EGEE. *Design of the EGEE gLite middleware external interfaces*. Deliverable DJRA1.2, September 2004. https://edms.cern.ch/file/487871
[10] EGEE. *Software and associated documentation*. Deliverable DJRA1.3, April 2005. https://edms.cern.ch/document/567624
[11] Foster I., Kesselman C., and Tuecke S. *The Anatomy of the Grid: Enabling Scalable Virtual Organization*. The International Journal of High Performance Computing Applications, 15(3):200-222, 2001.
[12] Foster I., Kesselman C., Nick J., and Tuecke S. *The Physiology of the Grid: An Open Grid Service Architecture for Distributed Systems Integration*. Open Grid Service Infrastructure Working Group, Global Grid Forum, June 2002.
[13] Foster I. and Kesselman C. *The Grid 2: Blueprint for a New Computing Infrastructure.* Morgan Kaufmann, Second edition, November 2003.
[14] Foster I., Frey J., Graham S., Tuecke S., Czajkowski K., Ferguson D. F., Leymann F., Nally M., Storey T., Vambenepe W., and Weerawarana S. *Modeling Stateful Resources with Web Services*. White paper, 2004.
[15] gLite A Lightweight Middleware for Grid Computing. http://glite.web.cern.ch/glite/
[16] Global Grid Forum. Grid Monitoring Architecture Working Group. http://www-didc.lbl.gov/GGF-PERF/GMA-WG/
[17] Graham S., Hull D., Murray B. Web Services Base Notification 1.3 (WS-BaseNotification). OASIS Public Review Draft 02. November 2005.
[18] Graham S. and Treadwell J. *Web Services Resource Properties 1.2 (WS-ResourceProperties)*. OASIS Committee Specification. January 2006.
[19] Maguire T., Snelling D., and Banks T. Web Services Service Group 1.2 (WS-ServiceGroup). OASIS Committee Specification. January 2006.
[20] Sotomayor B., Childers L. Globus Toolkit 4: Programming Java Services. Morgan Kaufmann, 2005
[21] Srinivasan L. and Banks T. *Web Services Resource Lifetime 1.2 (WS-ResourceLifetime)*. OASIS Committee Specification. January 2006.
[22] The Globus Alliance. The Globus Toolkit. http://www.globus.org/toolkit/
[23] The Globus Alliance. The GT4 Aggregator Framework. http://www.globus.org/toolkit/docs/4.0/info/aggregator/
[24] Tierney B., Aydt R., Gunter D., Smith W., Swany M., Taylor V., Wolski R. A Grid Monitoring Architecture. GGF Memo, August 2002.

[25] Vambenepe W., Graham S., Askary S., and Niblett P. *Web Services Topics 1.3 (WS-Topics)*. OASIS Public Review Draft 01. December 2005.