D4.3.2

Prototype Implementation of the Infrastructure Services Layer

WP 4.3 Grid Infrastructure Services Layer

Dissemination Level: Public

Lead Editor: Antonis Litke, ICCS/NTUA

30/01/2006

Status: draft

SIXTH FRAMEWORK PROGRAMME

PRIORITY IST-2002-2.3.1.18

Grid for complex problem solving

Proposal/Contract no.: 004293

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. **"Collective Work"** means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.
b. **"Derivative Work"** means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.
c. **"Licensor"** means the individual or entity that offers the Work under the terms of this License.
d. **"Original Author"** means the individual or entity who created the Work.
e. **"Work"** means the copyrightable work of authorship offered under the terms of this License.
f. **"You"** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;

b. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats, but otherwise you have no rights to make Derivative Works. All rights not expressly granted by Licensor are hereby reserved, including but not limited to the rights set forth in Sections 4(d) and 4(e).

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any reference to such Licensor or the Original Author, as requested.
b. You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation. The exchange of the Work for other copyrighted works by means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.
c. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work, You must keep intact all copyright notices for the Work and give the Original Author credit reasonable to the medium or means You are utilizing by conveying the name (or pseudonym if applicable) of the Original Author if supplied; the title of the Work if supplied; and to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work. Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.
d. For the avoidance of doubt, where the Work is a musical composition:
    i. **Performance Royalties Under Blanket Licenses**. Licensor reserves the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital

performance (e.g. webcast) of the Work if that performance is primarily intended for or directed toward commercial advantage or private monetary compensation.

ii. **Mechanical Rights and Statutory Royalties**. Licensor reserves the exclusive right to collect, whether individually or via a music rights agency or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions), if Your distribution of such cover version is primarily intended for or directed toward commercial advantage or private monetary compensation.

e. **Webcasting Rights and Statutory Royalties.** For the avoidance of doubt, where the Work is a sound recording, Licensor reserves the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions), if Your public digital performance is primarily intended for or directed toward commercial advantage or private monetary compensation.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted

under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
b. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
c. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
d. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

Context

Activity **<4>**	Detailed Architecture, Design & Implementation
WP **<4.3>**	Grid Infrastructure Services Layer Architecture, Design & Implementation
Dependencies	This is the accompanying report prepared as the D4.3.2 Prototype Implementation of the Infrastructure Services Layer deliverable. This deliverables uses specifically input from the D431 Architecture of the Infrastructure Services Layer V1 report.

Contributors:

Contributors (in alphabetical Order):	Reviewers:[1]
Annalisa Terracina (DATAMAT)	
Antonis Litke (NTUA)	
Francesco D'Andria (ATOS)	
Josep Martrat (ATOS)	
Raul Bori (ATOS)	
Jesus Movilla (TID)	
Ignaz Mueller (USTUTT-HLRS)	
Kleopatra Konstanteli (NTUA)	
Mario del Campo Melgar (TID)	
Sotiris Chatzis (NTUA)	
Giuseppe Laria (CRMPA)	

Approved by: QM

[1] Due the nature of this document being a central document of the project it was not possible to determine completely independent reviewers. The approach chosen was to assign sections not written by the authors themselves to be reviewed and consolidate the results.

Table of Contents

Abbreviations

Akogrimo	Access To Knowledge through the Grid in a Mobile World
A4C	Authentication, Authorization, Accounting, Auditing and Charging
AR	Advanced reservation
CPU	Central Processing Unit
CSG	Candidate Set Generator
EMS	Execution Management Services
EPR	Endpoint Reference
EPS	Execution Planning Service
JM	Job Manager
MUWS	Management Using Web Services
OASIS	Organization for the Advancement of Structured Information Standards
OpVOBroker	Operative VO Broker
OGSA	Open Grid Services Architecture
SC	Service Consumer
SP	Service Provider
SLA	Service Level Agreement
SOAP	Simple Object Access Protocol
QoS	Quality of Service
VO	Virtual Organization
WSDM	Web Services Distributed Management
WSRF	Web Services Resource Framework
WSDL	Web Service Definition Language
XML	Extensible Markup Language

1. Summary

This document describes in short the services prototypes that have been developed under WP4.3 Grid Infrastructure Services Layer of the Akogrimo project. It is the accompanying document of the D4.3.2 "Prototype Implementation of the Infrastructure Services Layer" which is a deliverable of prototype nature.

The services that are presented in this document have been based on the architecture that has been designed in the deliverable D431 "Architecture of the Infrastructure Services Layer V1".

2. Introduction - Overview

The components that have been identified as building blocks of the WP4.3 architecture are the Execution Management Services component, Data Management, Monitoring component, Service Level Agreement Enforcement, Policy Manager, Metering component and the Security framework that will be applied. All these (including the sub-components they comprise) have been described in deliverable D431 "Architecture of the Infrastructure Services Layer V1". The partners that are involved in the workpackage have focused their development efforts in a set of modules especially for the prototype. These modules along with their interactions are presented in this report.

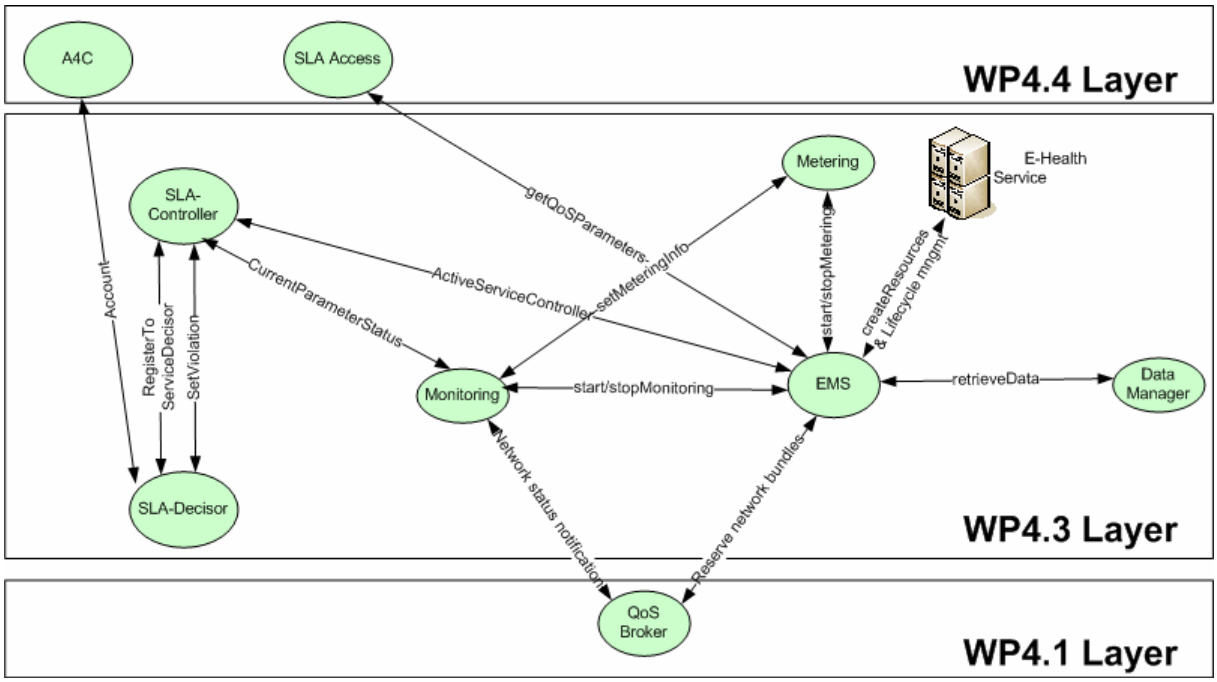

Figure 1 The layout of the WP4.3 developed services and their interaction with the other layers

In Fig. 1 we present an overview of the developed services in WP4.3 and their interactions within and out of the worpackage limits in order to conceptualize the functionality of the modules that have been developed. The objectives of the services, their functionality and the technologies involved are presented in the sequel of this report.

3. The developed services

3.1. Execution Management Services

3.1.1. Objectives

The Execution Management Service comprises the central controller of the business service execution. It is responsible for the establishment of communication and coordination between the various services that perform the execution and monitoring of the requested business services according to the user requirements specified in the user's contracts.

3.1.2. Functionality

The Execution Management service is developed on the basis of the WSRF specification. It is a persistent service that directs the accomplishment of the requirements specified in the contracts of the Akogrimo clients. In more details, it coordinates the operation of the various modules responsible for the execution, monitoring and SLA enforcement of the business services. The functionality of the service can be categorized in two generic classes: the external interface (outside layer 4.3) and the internal interface (inside layer 4.3) that shall be presented in the following paragraphs. The Execution management service exposes two operations:

- ***boolean createResources (String customerID, String slaID)*:** The Workflow Engine uses this operation to invoke the Execution Management Service. The EMS contacts the SLA-access service in order to retrieve the quality of service parameters that are defined in the contract with the specified id (slaID) by invoking the *getQosParameters* operation of the SLA-access service. After retrieving the quality of service parameters, the EMS uses this info in order to find the most appropriate host for the execution of the E-Health service that the client wants and performs a reservation by creating a resource for this service in the selected host and saving the endpoint reference that corresponds to this resource in a database. The EMS sets the time period that the E-Health service resource will be available and also performs a subscription to this E-Health service resource in order to receive notification messages from it. Afterwards, it contacts the SLA-Controller service and creates a corresponding resource. The EMS activates this SLA-Controller resource and also invokes the Monitoring service by calling the *startMonitoring* operation at the time specified in the contact.
- ***String getEPR(String slaID):*** This method returns the endpoint reference of the E-Health service resource that corresponds to specified slaID.

3.1.3. Involved technologies

The technologies needed for the implementation of the Execution Management Service are listed below:

- WSRF and WS-Notification specifications.
- Java WS-Core 4.0.1 included in the GT4 toolkit.
- Java JDK 1.5.0

3.2. Data Management Services

3.2.1. Objectives

The Data Management component is in charge of storing data, transferring data from one location to another and in general of keeping tracks about data stored.

The Data Management is able to handle large amount of data. It is able to handle different kind of data formats like text, binary, etc.

The Data Management module has been developed has a web service and it is not linked to other components of the layer. This means that it can be used as a stand alone service.

3.2.2. Functionality

The Data Management component is implemented as a web service and it is based on the RFT (Reliable File Transfer) component of the Globus Toolkit 4.

It exposes the following methods:

- *upload*
- *retrieve*
- *transfer*

The methods listed above reflect the Data Management behaviour. The first two methods, *upload* and *retrieve*, identify the data management module as a data manager itself. The third method, *transfer*, let the data manager acts as a third party instrument for data transfer.

The idea behind the upload and retrieve methods is that data should be first uploaded to the Data Manager. Once uploaded data are identified by a logical name that correspond to a physical name. Please note that for the first implementation data replication is not foreseen; the implementation will be changed when data replication will be inserted in the module features.

Once the data are uploaded to the Data manager, data can be retrieved in any moment in time.

The data transfer permits to move data from one location to another. It is implicit that, in order to do that, some software should be installed on the machines that want to use this third party feature.

Each one of the methods above needs some parameters in order to work. Table 1 reports the complete synopsis for each method.

Method name	Synopsis	Input parameters	Output parameters
upload	*FileID = upload(userID, srcUrl, filename)*	*userID* *srcUrl* *filename*	*FileID* unique identifier of the uploaded file
retrieve	*retrieve(userID, filename, destUrl)*	*userID* *filename*	none The file is retrieved to the

		destUrl	*destUrl* location
transfer	*transfer(userID, srcUrl, destUrl)*	*userID* *srcUrl* *destUrl*	None The file is copied to the *destUrl* location

Table 1

The meaning of the parameters mentioned in Table 1 are explained below.

- *String userID*

  This parameter is a string and it identifies the user. In future implementation this parameter will contain significant information about the user and about his identity. At the moment it is just used by the Data Manager to identify the user in its own internal logic.

- *String filename*

  This parameter is a string and identifies the name of the file that should be stored/retrieved.

- *String srcUrl*

  This parameter is a string and identify the source url of the file. It should be of the form:

  protocol://hostname/path/to/file/filename

  Protocol could be file or gsiftp.

- *String destUrl*

  This parameter is a string and identify the destination url of the file. It should be of the form:

  protocol://hostname/path/to/file/filename

  Protocol could be file or gsiftp.

- *String FileID*

  This output parameter is a string and it is returned by the *upload* method. It is the unique identifier of the file stored. It is of the form:

  protocol://hostname/path/to/file/filename

3.2.3. Involved technologies

The Data Management module is based on the following technologies:

- GT4 , in particular Reliable File Transfer
- Java 1.4.2
- Tomcat 5.0.28

3.3. Monitoring Services

3.3.1. Objectives

The Monitoring module of the Akogrimo architecture has the function of collecting data regarding to a service execution from different sources and sending these data to the interested modules.

In a first stage, the modules sending data to the Monitoring one are:

- QoS Broker. This module sends to Monitoring a change in the Network Bundle.
- Metering. This module sends to Monitoring the CPU Load and the Number of Instantiations associated to a certain service.

The module receiving this information from the Monitoring is the SLA Controller, which needs it to fix a service violation.

3.3.2. Functionality

The Monitoring component is modelled like a WS-Resource, implemented following the WSRF and WS-Notification specifications and using the GT4 toolkit.

For the Monitoring module begin to work it exposes the method startMonitoring, that will be called by the EMS module.

The following step is the Monitoring receiving the measured data from the QoS Broker (Network Bundle parameter) and Metering (CPU Load and Number of Service Invocations parameters). For this purpose, the Monitoring module exposes two methods: setMeteringData and setQoSBrokerData, invoked, respectively, by the Metering and QoS Broker modules for sending this information related to the service performance.

With the received information, the Monitoring component makes some calculations and sends to the SLA Controller the values of the parameters.

Once the Monitoring finalize its work, the EMS will invoke the stopMonitoring method, analogous to the startMonitoring one.

3.3.3. Involved technologies

The following technologies are needed for the implementation of the Monitoring module:

- WSRF and WS-Notification specifications.
- Java WS-Core 4.0.1 included in the GT4 toolkit.
- Java JDK 1.5.0.

3.4. SLA Enforcement Services

3.4.1. Objectives

The SLA Enforcement is split in two main services; SLA Controller and the SLA Decisor.

The SLA Enforcement services are in charge of

- Receiving the QoS measurements sent by monitoring
- Checking in execution phase the fulfilment of the SLA contract agreed between a service provider and a service client
- Notifying the existing violations

The SLA Enforcement system has a supervisor role and it is responsible for the verification of the contract conditions (QoS thresholds) of all running services, alerting about any abnormal situation and taking quick and effective decisions.

3.4.2. Functionality

Every time a new application/business service is created by EMS in Akogrimo environment, a new instance of SLA Controller is created and associated to it as well. The SLA Controller instance receives the QoS measurements of the assigned service and verifies that the measurements are within the threshold defined in the SLA contract. In the negative case, a QoS violation occurs and this fact is notified to SLA Decisor. For this first prototype, SLA Decisor is a service that simply logs these events. In a second stage, this subsystem, in cooperation with Policy manager, will be in charge of deciding what necessary actions must be undertaken according to the associated policy and the status of the system (normally notify this to EMS and/or VO Manager).

With regard to ***SLA Controller*** module the following functional capabilities have been implemented in the first prototype:

Method	
ActiveService Controller	**ActiveServiceController (string serviceID, objQoS objQoSData, string contractID, string topic)** EMS enables a SLA-Controller instance passing QoS parameters and thresholds. The SLA-Controller will be subscribed to suitable monitoring topics. *ServiceID:* contains ID of the service has to be controlled. *ObjQoSData:* contains QoS parameters to be measured (extracted from contract). *contractID*: String that contains the document identifier In this activation method, the SLA Controller creates the necessary WS-Notification mechanism to communicate with SLA Decisor.
CurrentParameter Status (*finally not used)	**objectServiceInfo CurrentParametersStatus()** Monitoring notifies periodically the QoS measurement to SLA-Controller *objectServiceInfo:* contains the measurements of the SLA parameters of a specific service.
receiveInfoby Monitoring	**String receiveInformatioByMonitornig(string serviceID, objQoS objQoSData)** This method exposed by SLA Controller allows monitoring subsystem to inform about the QoS measurements with regard to a service *ServiceID:* contains ID of the service has to be controlled. *ObjQoSData:* contains QoS parameters to be measured.

In turn the ***SLA Decisor*** module implements the following methods for first prototype:

RegisterTo Decisor (implemented but not used in 1st prototype)	**string ActiveSLADecisor(EndpointReferenceType sourceEPR, string ServiceId))** The SLA-Controller is registered to a SLA-Decisor. Before registration, the SLA-Decisor determines a maximum number of possible SLA-Controller instances that can be maintained at the same time. When the new registration implies to exceed the limit, the SLA-Decisor creates a new instance itself, and the registration is performed over the new instance. *string (return):* contains the EPR of the SLA-Decisor (In this case EPR means *URL + Reource ID*). Note: This method will be implemented in the 2nd cycle for scalability reasons

SetViolation (WS-Notification)	**receiveNotify (objViolation Violation)** The SLA-Decisor receives the violation message from the SLA-Controller that in turn will simply write a log, a simple XML file stored in a repository that contains information like: • *Time of the Violation;* • *ServiceID;* • *Kind of violation;* • *Value;* • *Bundle;* *Violation:* contains information related to the violation. In fact this method corresponds to the notification mechanism between both modules.

3.4.3. Involved technologies

SLA Controller and SLA Decisor are developed in Microsoft .NET platform and therefore they use the .NET framework 1.1 with WS Enhancements (WSE 2.0 SP3)

SLA Controller is a transient Grid service while the SLA Decisor is a persistent GS. They make use of WSRF.NET implementation of the University of Virginia.

The communication between both modules (SLA Controller and SLA Decisor) is based on WS-Notification specification.

In terms of SLA Contract definition, the WS-Agreement specification is considered. It defines the structure of agreements and their templates and it can be extended and complemented by other terms.

However, the analysis of approaches gathered in WSLA (IBM specification) has been taken into account to define the contract template.

3.5. Metering Service

3.5.1. Objectives

The Metering service provides the functionality required for the measurement and logging of the resources that the system dedicates to the execution of the business services (E-Health) services that are requested by the customers. This information is necessary for the accounting system's operations.

3.5.2. Functionality

The Metering service shall record for each host the following performance parameters:

- CPU utilization

- Disk utilization

The Metering service is invoked by the Execution Management service when the execution of a requested by a client E-health service starts. The Metering service periodically records the parameters listed above and invokes the *setMeteringData* provided by the Monitoring service in order to notify it about the values of these parameters. The notification message passed to the Monitoring service through the *setMeteringData* operation is an XML-like string that includes the IP address of the host and all the necessary information about the parameters that are being monitored.

3.5.3. Involved technologies

The technologies needed for the implementation of the Metering service are listed below:

- WSRF and WS-Notification specifications.
- Java WS-Core 4.0.1 included in the GT4 toolkit.
- Java JDK 1.5.0