D5.1.2

Integrated Prototype

Version 1.0

WP 5.1 Architecture Integration Report

Dissemination Level: Public

Lead Editors: Alfonso Sánchez-Macián & Victor Villagrá, UPM

13/03/2006

Status: Final

SIXTH FRAMEWORK PROGRAMME

PRIORITY IST-2002-2.3.1.18

Information Society

Grid for complex problem solving

Proposal/Contract no.: 004293

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.
c. For the avoidance of doubt, where the work is a musical composition:
    i. **Performance Royalties Under Blanket Licenses**. Licensor waives 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.
    ii. **Mechanical Rights and Statutory Royalties**. Licensor waives the exclusive right to collect, whether individually or via a music rights society 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).
d. **Webcasting Rights and Statutory Royalties**. For the avoidance of doubt, where the Work is a sound recording, Licensor waives 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).

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.

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 credit as required by clause 4(b), as requested.
b. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; 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.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE MATERIALS, 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, 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 5	**Integration and Application Case Studies**
WP 5.1	**System Integration**
Task 5.1.2	**Integrated Prototype**
Dependencies	**This deliverable uses specifically the input of the deliverable D5.1.1 for defining the manual of the Integrated Prototype of Activity 5.**

Contributors:

Giuseppe Laria (CRMPA)

Nadia Romano (CRMPA)

Giulio Negro (CRMPA)

Georgina Gallizo (TID)

Ignatz Mueller (USTUTT)

Patrick Mandic (USTUTT)

Per-Oddvar Osland (Telenor)

Fredrik Solsvik (Telenor)

Brynjar Viken (Telenor)

Ian Johnson (CCLRC)

Brian Ritchie (CCLRC)

Tom Kirkham (CCLRC)

Alfonso Sanchez-Macian (UPM)

Jose I. Moreno (UPM)

Víctor A. Villagrá (UPM)

Reviewers: Peer reviewed among contributors

Version	Date	Authors	Sections Affected
0.1	27-Feb-05	All Contributors in WP5.1	Initial inputs for all sections
0.2	13-Mar-05	Victor Villagrá (UPM), Alfonso Sanchez-Macián (UPM)	Complete revision of the document

Executive Summary

This document presents a manual for the installation and configuration of the Akogrimo Integrated Prototype.

Table of Contents

List of Figures

List of Tables

Abbreviations

A4C	Accounting, Authentication, Authorization, Auditing, Charging
Akogrimo	Access To Knowledge through the Grid in a Mobile World
BVO	Base Virtual Organization
CM	Context Manager
ECG	Electrocardiogram
EMS	Execution Management Service
EPR	End Point Reference
GASS	Grid Application Support Service
GT4	Globus Toolkit version 4
MD	Monitoring Daemon
MT	Mobile Terminal
OpVO	Operative Virtual Organization
PR	Participant Registry
SA	Service Agent
SAML	Security Assertions Markup Language
SLA	Service Level Agreement
UA	User Agent
VO	Virtual Organization
WF	Work Flow
WS	Web Service
XML	Extensible Markup Language

1. Introduction

Scope

The purpose of this document is the description of the guidelines to execute the Akogrimo Integrated Prototype. The deployment task has been performed in the Stuttgart Integration Site and validated with a simplified e-health testbed.

The following sections distribute the target machines in compliance with the mapping agreed during the integration and deployment phase.

Intended audience

This document is primarily intended for

- Partners from WP5.1 and other WPs involved in system integration.
- Technical integrators involved in the deployment of the Akogrimo prototype.

2. Integration Site Architecture

The architecture of the Stuttgart Integration Site as described in D5.1.1 is presented again including all the modules that will be explained in the following sections. Figure 1 is showing the Akogrimo physical architecture, with all the modules included in the prototype and integrated in their respective target machines, while Figure 2 is showing the specific network configuration of the integration site deployed in the University of Stuttgart and used for the validation of the prototype with a simplified e-health testbed.

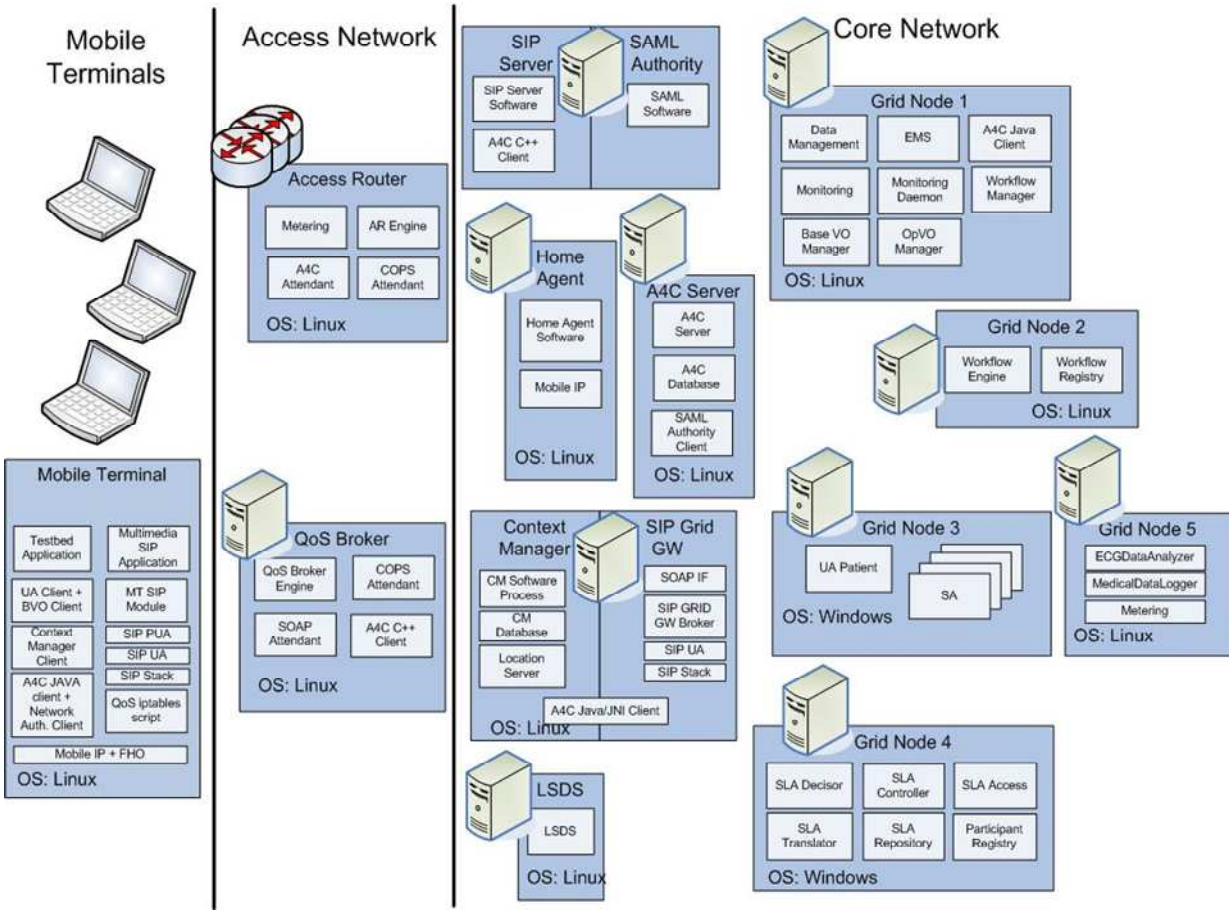

Figure 1: AKOGRIMO Physical Architecture

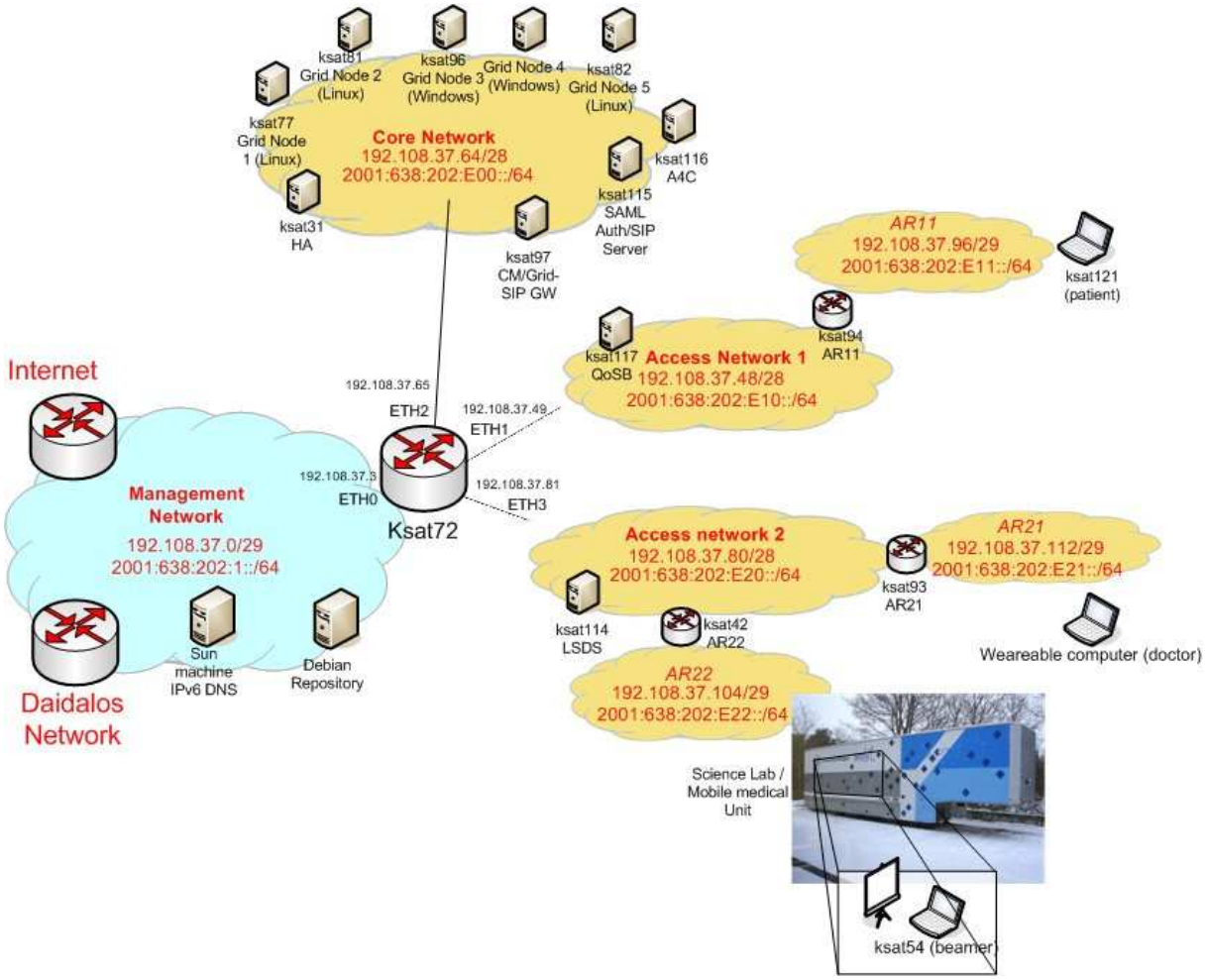

Figure 2: Integration Site Networks

Next sections will present the specific details for executing and configuring the modules of each target system.

3. Mobile Terminals

There are three mobile terminals available in the integration site. As a simplified e-health application is running in the prototype, there are some differences between them due to the specific testbed hardware requirements and applications.

3.1. Mobile IPv6

The mobile IP infrastructure in the MT is divided in two parts, a patched kernel and a set of utilities to support this functionality. To start the MIPv6 support in the MTs is necessary to run the command:

```
mip6d -c /etc/mip6d.conf
```

The "-c" option is followed by the path to the configuration file.

This functionality requires MIPv6 running in the Home Agent to perform the binding of the Care-of-address to the Home-address.

3.2. MT SIP infrastructures

SIP infrastructures on the MTs can be tested using the multimedia SIP application included for the simplified e-health testbed. Provided that all related software has been currently installed and configured in the corresponding machines, the general procedure to start it is the following.

1. Start a terminal window and change privileges to superuser, by typing:

   ```
   sudo –s
   ```

2. Start MT Multimedia Application. There is a shell script you can use for this proposal. Go to the installation directory (`/opt/akogrimo/mt_sip_module`) and type:

   `./MTMultimediaApp restart`.

3. This will stop and start again the application if running. For more information on how to use the script, please type:

   `./MTMultimediaApp` or `./MTMultimediaApp usage`

4. After starting, you should see in the terminal several logs, which indicates the current status of the component (the component can be also configured to store logs in a file or not to show any logs by editing the <user>.xml file located in the installation directory, see annex A.2). If everything worked, you should see the login window.

5. Please check if the application has detected audio and/or video capture devices (you can see it in the logs). The application will inform if no audio either video capture devices are available.

6. Then you are allowed to use one of the preloaded profiles or to create a new one by editing the different fields of the login window.

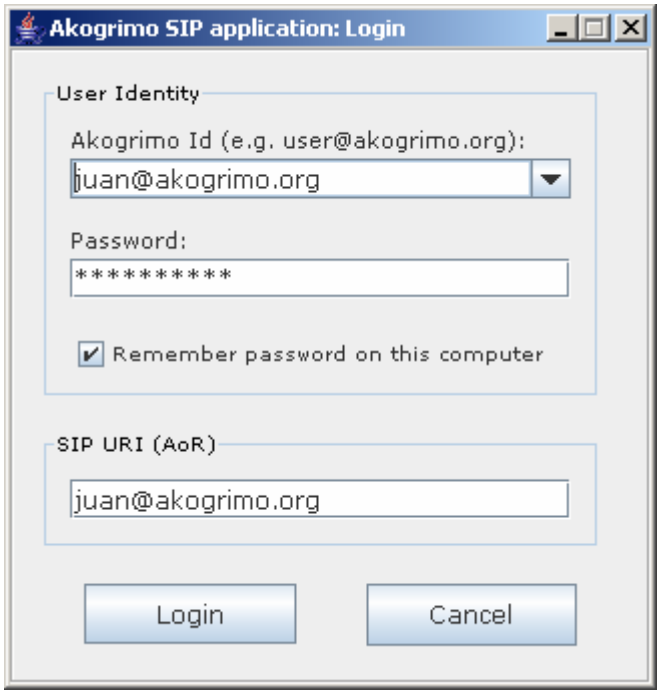

Figure 3 Login Window

7. Then push the login window and continue using the graphical user interface. Please be sure the connection with the A4C Server has been successfully established, and the A4C client java library properly configured. If not, the script will stop waiting a reply for the A4C Server, some log error will be shown or some logs indicating A4C Server connection time out will appear. If everything was ok, you should see the main window.

Figure 4 Main Window

8. You can choose to publish your presence information by clicking on the "Presence…" button and choosing an xml file (pidf/rpid format) which will be automatically published in

the SIP server. You can modify the contents of the files or add others to the directory `/opt/akogrimo/mt_sip_module/UserProfile/Presence`.

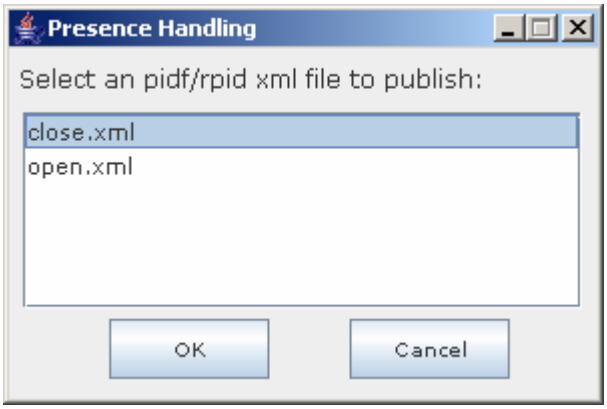

Figure 5 Window for presence/context handling

9. Next interactions with the graphical interface are intuitive and easy to use. It will inform you about incoming events (e.g. a grid-initiated call and a grid-initiated call transfer request, enabling you to control the media data during a multimedia session and so on.

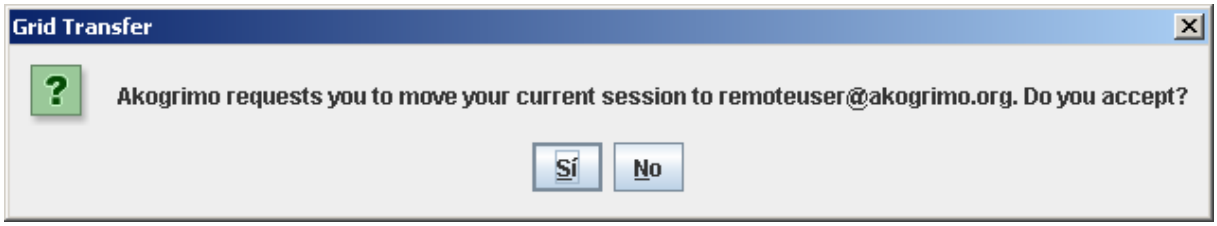

Figure 6 Sample of application information on incoming request

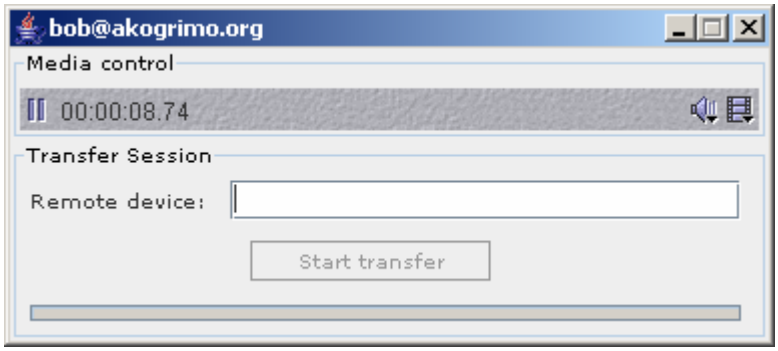

Figure 7 Window for audio control sample

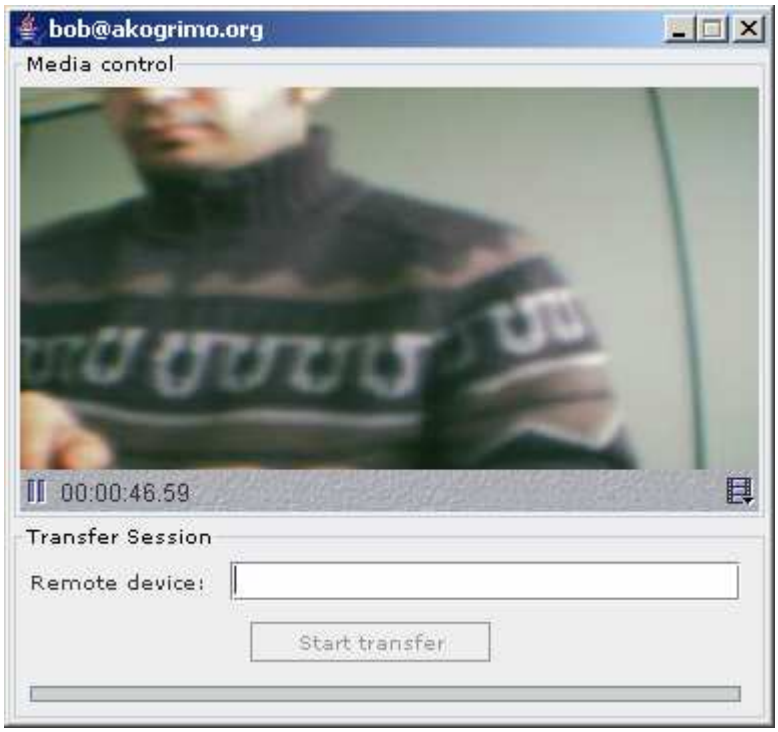

Figure 8 Window for video control sample

10. To stop a media session, just close the corresponding window. Audio and video appears in separate windows to enable user-triggered transfer each media to different devices (note that this functionality will be not available for the demo, only grid-triggered transfer). Note also that establishment of RTP traffic takes some time; during this time, media windows are disabled. Some logs should indicate what is going on.

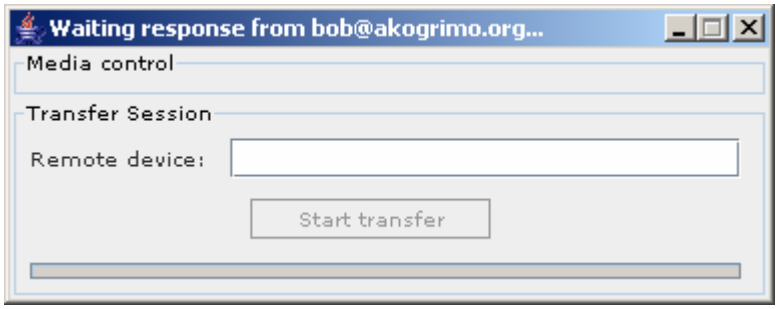

Figure 9 Sample of media windows during RTP establishment

3.3. Testbed Application

Some modules of the eHealth testbed are installed in the mobile terminals. They are composed by several services and are started following these steps:

1. Log in as user ehealthdemo.
2. Start Tomcat to run the services:

```
/opt/jakarta-tomcat-5.0.30/bin/startup.sh
```

3. Change to root user:

```
sudo su
```

4. Start Grid FTP Server to send and receive data:

```
/opt/gridftp-4.0.1/sbin/globus-gridftp-server &
```

4. Network Target Systems

4.1. Access Routers

Three access networks and their routers are available in the architecture to allow the connection of the mobile terminals. Traditional IPv6 routing software is used.

To start the Access Router software that communicates with the QoS Broker:

1. Log-in as root

   ```
   sudo –s
   ```

2. Enter into the directory where the Access Router software is installed.

   ```
   cd /opt/arm-1.2/bin
   ```

3. Start and run the software

   ```
   ./arm.sh start
   ./arm.sh run
   ```

In case something fails and you have problems restarting the access router software, you can perform a clean up by issuing the command

```
./arm.sh clean
```

4.2. Home Agent

The mobile IP infrastructure in the HA is divided in two parts, a patched kernel and a set of utilities to support this functionality. To start the MIPv6 support in the HA is necessary to run the command:

```
mip6d -c /etc/mip6d.conf
```

The "-c" option is followed by the path to the configuration file.

This functionality requires a MIPv6 DNS with Home Agent support. This DNS is installed in a different machine and it is started using the command:

```
/etc/init.d named start
```

4.3. QoS Broker

The QoS Broker software is installed at the directory `/home/boss/anqosbr`. To run this software:

1. Log-in as root

   ```
   sudo –s
   ```

2. Enter into the directory where it is installed

   ```
   cd /home/boss/anqosbr
   ```

3. Run the program:

```
./broker
```

The QoS Broker is then started and ready to connect to other modules.

4.4. SIP Server and SAML Authority

4.4.1. SIP Server

SIP Server is currently showing logs messages by default in the console from it was started, although it can be configured also to redirect these logs to a file. Log level can be changed by editing file `/etc/ser/ser.cfg` (find the line `debug = 3` and type a higher level of detail; 3 should be enough).

Some modules have their own debug system and trace levels independent from the general log framework of the SIP Server. To modify them, modify the lines

```
modparam("esc", "trace_level",   1)
modparam("a4c", "trace_level",   2)
```

and type the desired level of detail.. Current values should be enough, so in principle you should not to change this file at all.

In order to star the SIP server, type (from any directory):

```
sudo ser start
```

After starting, you should see general information (like aliases, the IP address and the port where the SIP Server is listening, current debug level…) and some particular information regarding esc module (Presence Agent) and a4c module. If there are no error messages, then the SIP Server is running properly. You should see some logs regarding registration, proxy transactions, presence publication and delivery of presence information.

To stop the SIP Server, type:

```
sudo pkill ser
```

There is a script utility useful to monitor how the component is running. From a console type:

```
export SIP_DOMAIN=akogrimo.org
```

and

1. `sudo serctl ul show` if you want to see the Location Service status (users currently registered). Type yes when requested.
2. `sudo serclt moni` to monitor ser activity. If ser is running, you should see some information about the status of the SIP Server internal cycles (should be moving) and the requests and replies that the component has handled.

4.4.2. SAML Authority

SAML Authority is started with the tomcat service. In order to start it, if needed due to a stop of the service, the procedure is the following:

1. Start the tomcat service by doing:

   ```
   sudo /opt/samlauthority/tomcat/bin/startup.sh
   ```

2. To check if the tomcat service is running, it should be:

Open a browser and check that the SAMLAuthority is available connecting to the URL http://ksatXX.ipv6.rus.uni-stuttgaart.de/axis/services (or the adequate IP Address of the machine where the SAML-Authority is running), and click in the wsdl file

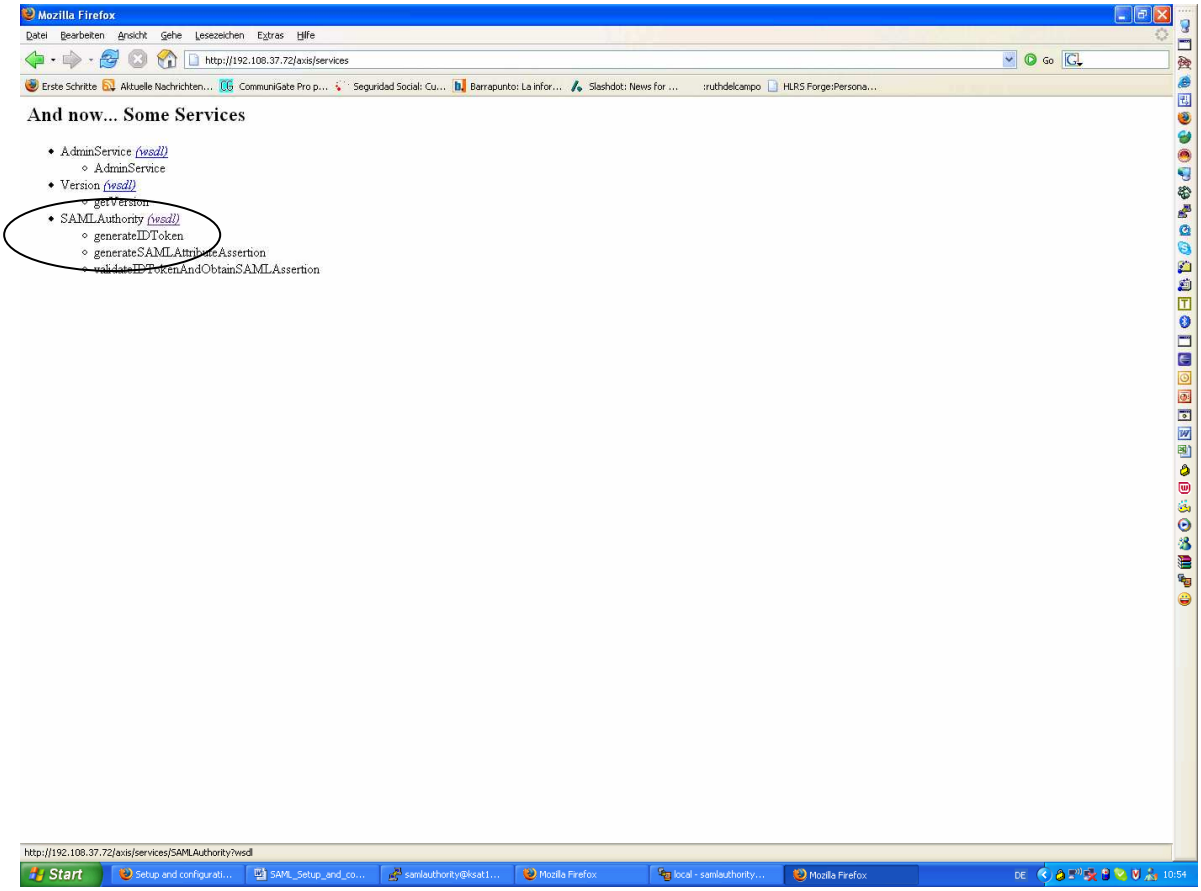

Figure 10: SAML Authosity Service

3. To stop the tomcat service:

```
sudo /opt/samlauthority/tomcat/bin/shutdown.sh
```

4.5. A4C Server

A4C Server software connects to the SAML Authority so it should be started after this other module. To run the software:

- Log in in the machine where it is installed as the user "a4c".
- Execute:

```
./A4CServer/startA4CServer.sh
```

4.6. Context Manager and SIP Grid Gateway

4.6.1. SIP Grid Gateway

This component represents a Web Service that is deployed on a Tomcat 5.0.28 application server. The own component includes the required Axis libraries to run as a standard Web Service. It is very important to remark that is a Standard Web Service not a stateful Web Service (WSRF platform).

The SOAP interface component has been deployed on a Tomcat 5.0.28 whose `CATALINA_HOME=/opt/tomcat/`. In order to setup the services this component offers to the Akogrimo platform (i.e. grid call and grid transfer), the procedure is the following:

1. Change user to sipgridgw. From a terminal, type

   ```
   su – sipgridgw.
   ```

2. Start SIP Grid Gateway Broker Engine. There is a shell script you can use for this proposal. Go to the installation directory (`/opt/akogrimo/sip_grid_gw`) and type

   ```
   ./brokerEngine restart
   ```

   This will stop and start again the service if running. For more information on how to use the script, simply type:

   ```
   ./brokerEngine or ./brokerEngine usage
   ```

3. After starting, you should see in the terminal several logs, which indicates the current status of the component (the component can be also configured to store logs in a file or not to show any logs by editing the SIPGridGw.xml file, see annex A.1). If everything worked, last log you can see should say "SIP Grid Gateway Broker Engine started". Please be sure the connection with the A4C Server has been successfully established, and the A4C client java library properly configured. If not, the script will stop waiting reply for the A4C Server, some log error will be shown or some logs indicating A4C Server connexion time out will appear. SIP Grid Gateway can work even if the A4C Server is not running; but obviously, results will be different than expected.

4. Be sure WS interface is currently working. In order to check it, go to the directory `$CATALINA_HOME/bin`, by typing:

   ```
   cd $CATALINA_HOME/bin
   ```

5. In case that the environment variable is not defined, please define it for this session and include `export CATALINA_HOME=<tomcat-installation-dir>` (typically `/opt/tomcat`) in the .bashrc file of the user home for forthcoming sessions.

6. Execute the Tomcat shell script startup.sh

   ```
   ./startup.sh
   ```

7. Go to the tomcat logs file to test if the startup process has been successfully. The Tomcat logs file is placed at `$CATALINA_HOME/logs/catalina.out`. In order to stop the Tomcat container the shell script `$CATALINA_HOME/bin/shutdown.sh` should be executed in the same way as the startup.sh previously described.

In order to perform local tests using the Grid Gateway Broker Engine script, type from the installation directory:

`Ctrl+C` to continue using the same terminal window

`./brokerEngine gridcall AkogrimoID1 AkogrimoID2` to launch a grid call between both users

`./brokerEngine gridtransfer AkogrimoID1 AkogrimoID2 TargetDeviceURI` to transfer the current session between `AkogrimoID1` and `AkogrimoID2` from device being used by `AkogrimoID1` to `TargetDeviceURI`.

The best option for testing is to use the SOAP interface local test, which is testing the whole path and automatically launch a grid call followed by a grid transfer request. Currently it launches a grid call between bob@akogrimo.org and alice@akogrimo.org, and then it requests for a session transfer from bob@akogrimo.org to the device registered as carol@akogrimo.org. It is possible to change the involved users and devices by editing the file `/opt/akogrimo/sip_grid_gw/soapif/build.xml`. For making this test, move to `/opt/akogrimo/sip_grid_gw/soapif` and type:

```
sudo ant run-client
```

4.6.2. Context Manager

4.6.2.1. Overview

The following figures are showing the detailed architecture of the context manager., with the components and files which are part of it.

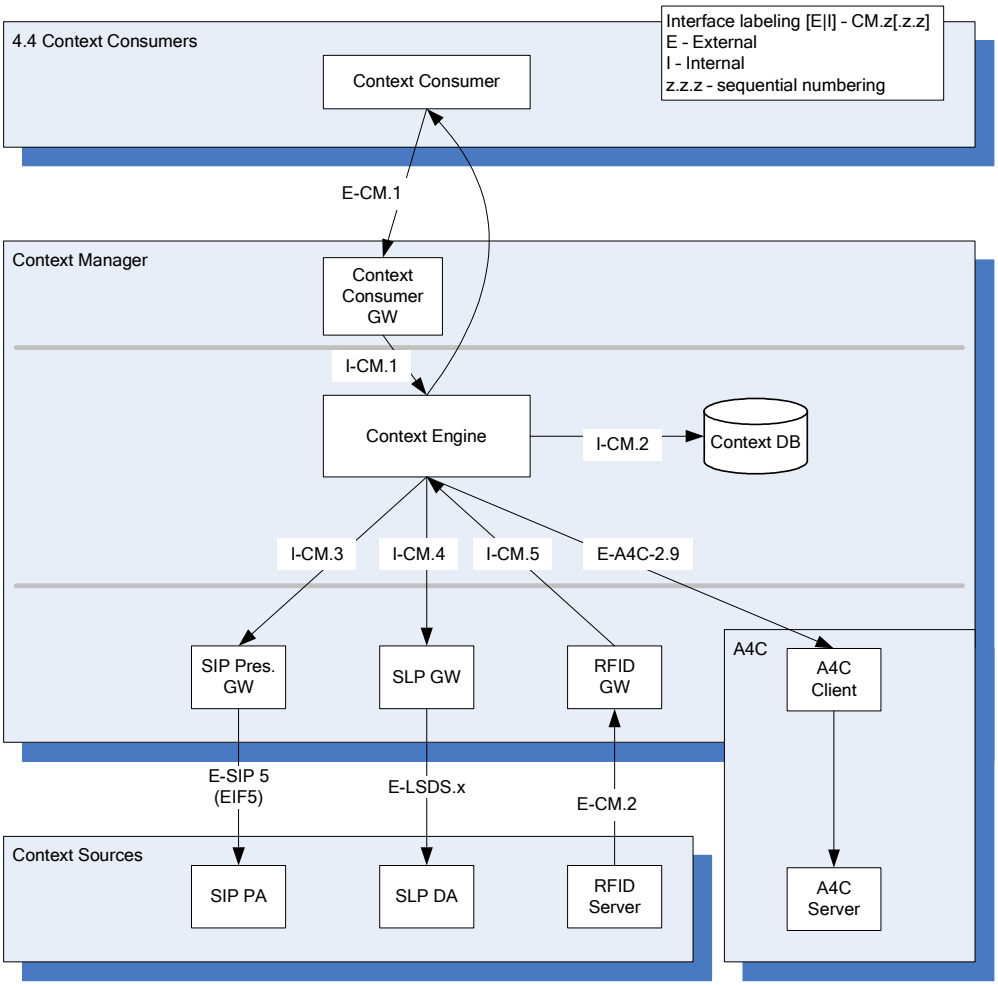

Figure 11: Context Manager overview

Table 1: Overview of CM subcomponents and selected files

Sub-component	Purpose	Special presumptions
CM: Context Consumer GW	Interface to Context Manager Interface to Context Consumer for returning results	Tomcat 5.5+, Axis 1.2.1+
CM: Context Engine	Main part of CM	
SLP GW	Local service discovery	OpenSLP 1.3, GCC 4.0, make
RFID GW	Communication with RFID SW	
SIP Presence GW	Interface to SIP PA	

Sub-component	Purpose	Special presumptions
CM Client (@ MT)	Communicating User Agent Profile (UAProf) to the Context Manager.	
Context Database	Holds user context and subscribers	
3rd party: SIP Communicator	Part of SIP Presence GW	
3rd party: RFID SW (JPCSC 0.8.0)	Receives input from RFID readers	

4.6.2.2. Configuration

Unless stated, configuration refers to /etc/contextmanager/ContextManager.properties

4.6.2.2.1. Database

- `db.url = jdbc:mysql://localhost:3306/contextdb`
  Update if database server is different from localhost, or if port number is different from 3306
- `db.user = contextmanager`
  change if different username
- `db.password = mysql`
  Set password for db.user
- `db.numConnections = 2`
  Max number of simultaneous calls to the database. Increase if needed for scalability

4.6.2.2.2. Engine

Configure RMI interfaces provided by Engine

- # --- RfidContextDestination : a RMI interface provided by contextengine, and used by rfidGw
  `cm.rmiserver.rfid.url = //localhost/RfidContextDestination`
- # --- SipPresenceContextDestination: a RMI interface provided by contextengine,
  # and used by SIP presence GW
  `cm.rmiserver.sippresence.url = //localhost/SipPresenceContextDestination`
- # --- ContextManagerServer: a RMI interface provided by contextengine, and used by ContextManagerWS
  `cm.rmiserver.contextmanagerWS.url = //localhost/ContextManagerServer`

4.6.2.2.3. SLP GW

- SLPGateway url, update to match server address for the machine hosting the process
  `slp.gw.url = //ksatXX.ipv6.rus.uni-stuttgart.de/SLPGw`
- SLP DA
  `slp.da = ksat114`
- SLP range refers to radius in meters when searching for SLP services within a certain area
  `slp.range = 50`

4.6.2.2.4. SIP Presence GW

- url for rmi interface server
  `sip.gw.url = //localhost/sippresencegw`
- realm for the searched presentity
  `sip.gw.presentity.realm = ksatXX.ipv6.rus.uni-stuttgart.de`
- Username and password for the SIP Presence GW
  Used when the SIP Presence GW sends REGISTER
  `sip.gw.watcher.username = akogrimo_wp42_cm`
  `sip.gw.watcher.password = akogrimo_wp42_cm`

- The following parameters are configured in sip-communicator.xml
  - Public address for the SIP Presence GW
    <PUBLIC_ADDRESS value="sip:akogrimo_wp42_cm@akogrimo.org"/>
  - Registrar address, username (for registering the SIP Presence GW)
    <REGISTRAR_ADDRESS value="`192.108.37.72`"/>
    <USER_NAME value="akogrimo_wp42_cm"/> (same as in ContextManager.properties)
  - Set domain and realm for the SIP server where SIPPresenceGW registers
    <DEFAULT_DOMAIN_NAME value="`192.108.37.72`"/>
    <DEFAULT_AUTHENTICATION_REALM value="`192.108.37.72`"/>

4.6.2.2.5. RFID GW

- Change `localhost` to the appropriate address for the host for RFID GW
  `rfid.gw.url = //localhost/rfidgw`

4.6.2.2.6. A4C GW

- Address to A4C server
  `a4c.server = a4c-server.ifi.unizh.ch`
  `or`
  `a4c.server = ksat116.ipv6.rus.uni-stuttgart.de`

4.6.2.3. Running CM

4.6.2.3.1. General

The following processes need to be started and stopped manually:

- System: The following must be running
  - RMIregistry
  - Tomcat and axis
- Context Manager processes: Scripts may be run from any location on Linux. Scripts must be run as root (sudo –s).
  - Engine: `$> startengine.linux`
  - Sippresencegw: `$> startsippresencegw.linux`
  - slpgw: `$> startslpgw.linux`[1]
  - rfidgw: `$> startrfidgw.linux`
  - To stop, press Ctrl-c (or use `$> kill -9 <process numer>`)
  - On windows, replace `.linux` with `.bat`
- ContextManagerWS (Web Service for Context consumer interface)
  - Enter address "http://localhost:8080/manager/html" in a browser window.
  - Under "WAR file to deploy", klick "Browse" and select file ContextManagerWS.war. Klick "Deploy". If you also wish to run the Context Consumer simulator, deploy ContextConsumerWS.war as well.
  - Test client for simulating input from Context Consumer: Click "ContextManager", then "Test"

4.6.2.4. RFID Server

4.6.2.4.1. Configuration

`reader.positions.config` need special attention. You need to edit the url of the rfidgateway in order to communicate with the contextmanager, change `rfid.gw.url` .

You also have to edit the positions of the readers connected. The digit signifies what number reader connected to the reader the position is valid for. The readers are counted from 0 and up. And the current readers we use have two readers in them, the first one being a slot in, while the second on each reader is the contactless one. If you got multiple readers connected and want to use the contactless readers, make sure to specify odd numbers for the reader positions in this file.

[1] Note that the SLP Directory Agent has to be running (as configured in Section 4.7). I.e. the slpd process has to be started on this machine.

4.6.2.4.2. Running

If installed properly, `pcscd`, which is required for `rfidserver`, will be running. If it does not, make sure to start in prior to starting `rfidserver`. `Rfidgw` and `contextmanager` need to be running on a computer too.

- `startrfidserver.linux` – runnable from any location, run as root (sudo –s).

4.7. LSDS

The Local Service Discovery Server should be started as root using the command:

```
slpd -c /etc/slp.conf -r /etc/slp.reg
```

5. Grid Target Systems

5.1. Grid Node 1

5.1.1. Containers

Several of the modules developed in this machine use the Globus infrastructure. A script has been developed to run all of them:

```
/etc/init.d/start-akogrimo-containers
```

5.1.2. Monitoring Component

Monitoring component is currently installed, configured and running on the machine hosting the Grid Node 1. It is started with the Containers script presented in the previous section. It could also be started separately following the next guidelines.

In order to setup the services this component offers to the Akogrimo platform a link between producer information components like Metering and QoSBroker and the corresponding SLA-Controller. The procedure to start up the component is the following:

1. Change user to monitoring. From a terminal, type

   ```
   su – monitoring
   ```

   with the adequate password when requested.

2. Start GT4 WS-Core java container. To do so, you can achieve it in two different ways:

   - Go to $GLOBUS_LOCATION/bin and execute

     ```
     globus-start-container –nosec
     ```

   - Executing a shell script you can use for this purpose. Go to the installation directory (/opt/akogrimo/monitoring/) and type

     ```
     ./gt4-monitoring-manage start
     ```

   This will start the GT4 WS-Core java container an, consequently, also starts up the monitoring service. This script also allows stopping the container

   ```
   ./gt4-monitoring-manage stop
   ```

   to check whether the java container is alive

   ```
   ./gt4-monitoring-manage alive
   ```

   to restart the java container

   ```
   ./gt4-monitoring-manage restart
   ```

3. After starting, you should check (in the same console) whether the monitoring component has started successfully. To know that you should see the following logs:

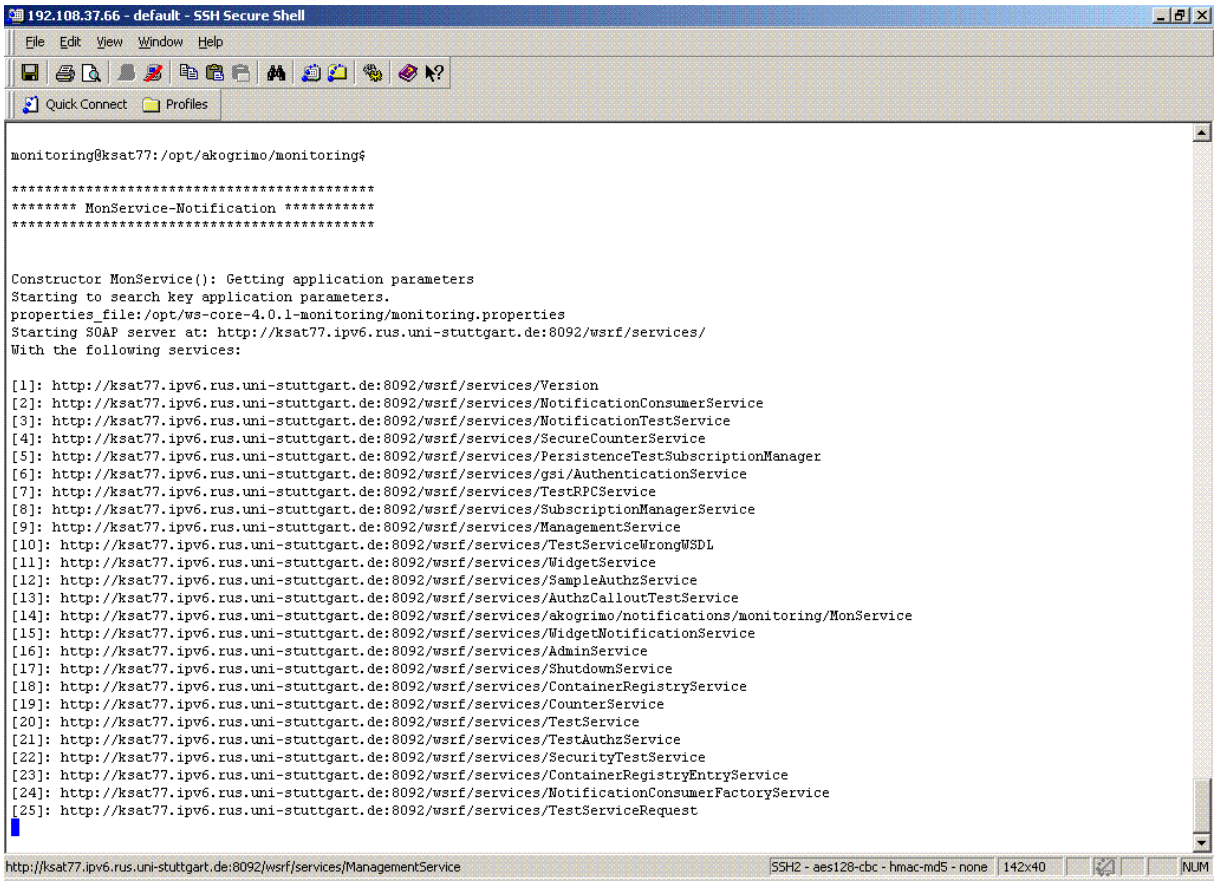

Figure 12: Logs corresponding to the startup process.

Please, verify that the monitoring service is present in this list. This is the corresponding url:

http://machine:8092/wsrf/services/akogrimo/notifications/monitoring/MonService

In order to perform local tests, please go to the directory

`/opt/akogrimo/monitoring/tests/construccion/`

and execute:

`ant run-clientStart` (this tests the interface with EMS)

`ant run-clientSetMetering` (this tests the interface with Metering)

`ant run-clientQoSBroker` (this tests the interface with QoSBroker)

After making the tests, please go to the data directory (/opt/akogrimo/monitoring/data/) and remove all the files ending with *.file*. There should be at least.

`ServiceRegistry.file`

`ServiceHost.file`

And one more file per service that it has been initialized by the "ant run-clientStart" test.

5.1.3. Data Management

Data Manager is currently installed in the Grid Node 1. To initialize the Data Manager:

- Log in as user 'wf' and run:

  ```
  grid-proxy-init -valid 1000:0
  ```

  When asked for a password, enter the appropriate one. It should be seen in the console:

  Your identity:

  /O=Grid/OU=GlobusTest/OU=simpleCA-thevoicepc2.datamat.it/OU=datamat.it/CN=lisa

  Enter GRID pass phrase for this identity:

  Creating proxy ..... Done

  Your proxy is valid until: Mon Apr 10 07:44:31 2006

- Again, as user 'wf' run:

  ```
  cd /jakarta-tomcat-5.0.28/bin
  ./startup.sh
  ```

  This command will start tomcat on port 8080

- As user 'root':

  ```
  ./opt/globus-4.0.1/sbin/globus-gridftp-server
  ```

5.1.4. Monitoring Daemon

5.1.4.1. Startup

To start the Monitoring Daemon service, just start the globus container, with the script presented in section 5.1.1.

It can also be done in a separate way:

```
export GLOBUS_LOCATION=/opt/ws-core-4.0.1-mondaemon-wfmgr
export PATH=$GLOBUS_LOCATION/bin:$PATH
# unless it's already in there
globus-start-container &
```

It may be wise to check that the container isn't actually running first!

Use "netstat -tlp" to check for this (see below for more); if there isn't an entry for port 8093, then the container isn't running.

5.1.4.2. Testing

In the directory /home/brian/workspace/MonitoringDaemon of the Grid Node 1 target machine, there is a script called `run-client.sh` which allows to make some local tests of this component.

For this testing, it is needed an environment like the previous one, plus GLOBUS_JARS set as follows:

```
export GLOBUS_JARS=$(ls $GLOBUS_LOCATION/lib/*.jar | tr '\n' ':')
```

The script performs several notify and getContext operations.

If all goes well, the output should look something like this:

```
~/workspace/MonitoringDaemon$ ./run-client.sh
Notify Context Change...
Get Context...
Returned context on next line:
13.12.00.01
(Did you get it?)
Notify Context Change (to little display)...
Get Context again ...
Returned context on next line:

(Should be empty. Did you get it?)
Restore big display to brian...
brian's context: 13.12.00.01
Get Cooked Context...
Returned context on next line:
cooked-display-id
(Did you get it?)
Get unknown user context...
Returned context on next line:

(Did you (not) get it?)
Get context for bob@akogrimo.org
Returned context:
ksat54.ipv6.rus.uni-stuttgart.de
(end of returned context)
Get context for alice@akogrimo.org
Returned context:
ksat54.ipv6.rus.uni-stuttgart.de
(end of returned context)
```

Also for testing, there is a script that can be used to set a user's context directly. The script is in

`/home/brian/workspace/MonitoringDaemon/run-notify.sh`

and should be used as:

```
run-notify.sh <user-id> <context-file-name>
```

The directory contains several example context document files. So for example,

```
./run-notify.sh bob@akogrimo.org bobActualContext.xml
```

will (at present) cause subsequent getNotify("bob@akogrimo.org") calls to return "beamer@ksat57...". as indicated in the XML file.

5.1.4.3. Stopping

To stop the container (e.g. to redeploy after a change), first make your window nice and wide, and then use:

```
netstat -tlp
```

Look in the listing for an entry whose value in the LocalAddress column is *:8093 (the port on which the container runs). On the far right there should be a string of the form "<nnnnn>/java" (e.g. "22894/java"; <nnnnn> is the process-id. Now stop the process by running

```
kill -TERM <nnnnn>
```

5.1.5. EMS

The EMS component is executed when starting Globus Container (see section 5.1.1).

5.1.6. BVO Manager, OpVO Manager and Workflow Manager

These components are executed when starting Globus Container (see section 5.1.1).

5.2. Grid Node 2

5.2.1. Workflow Engine and Workflow Registry

These two components have been deployed into Tomcat, so it is necessary to start this application.

As user 'wf' run:

```
cd /jakarta-tomcat-5.0.28/bin
 ./startup.sh
```

This command will start tomcat on port 8090.

5.3. Grid Node 3

5.3.1. User Agent component

In this section, we are going to describe the configuration of the User Agent component.

You can find the GUI to test this component at:

```
C:\Programme\Akogrimo\VOManagement\UA and SA
Services\Client\UA\Src\AgentsClient\bin\Debug\AgentsClient.exe
```

5.3.1.1. Current Configuration

For demonstrating this testbed, the following UA instances were created for the user bob@akogrimo.org

- http://ksat96.ipv6.rus.uni-stuttgart.de/VOManagement/UserAgent/UserAgent.asmx#dedddcaf-1525-4122-bc36-25e8fadb7f01
- http://ksat96.ipv6.rus.uni-stuttgart.de/VOManagement/UserAgent/UserAgent.asmx#daa7f4bb-a15c-423a-bbaa-b30d43f1d033
- http://ksat96.ipv6.rus.uni-stuttgart.de/VOManagement/UserAgent/UserAgent.asmx#5a93180c-98c8-49b0-a316-06465971236c

Furthermore, all these instances are configured with the following methods and parameters:

- http://ksat96.ipv6.rus.uni-stuttgart.de/VOManagement/UserAgent/UserAgent.asmx#dedddcaf-1525-4122-bc36-25e8fadb7f01

```
<MethodsList>
    <Method name="startFlow">
        <Parameters>
            <string description="userId"></string>
            <string description="doctorId"></string>
        </Parameters>
    </Method>
</MethodsList>
```

So, it is possible to invoke the startFlow method which start the Business process functionality.

5.3.1.2. How to create a new configuration

5.3.1.2.1. Preliminary Steps

- Remove any file with .dll extension in the folder **C:\WINDOWS\Temp**
- Start User Agent GUI

5.3.1.2.2. User Agent Creation (create() step in UI client → UA)

- Switch on the Create panel and click on the Create button and the User Agent EPR will show the EPR for the new UA instance.

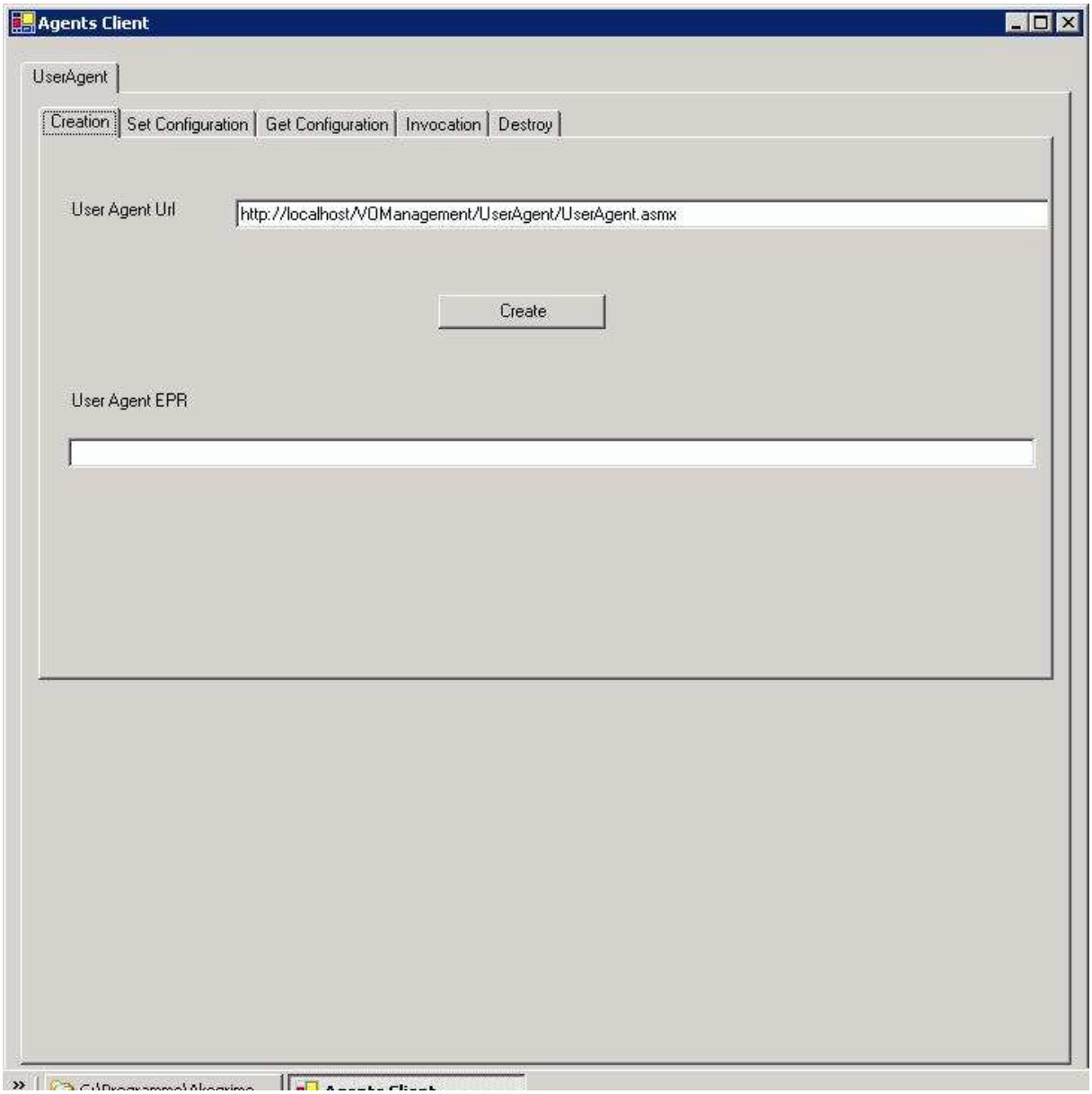

Figure 13: User Agent Creation

5.3.1.2.3. User Agent Configuration (configureUA() step in UI client → UA)

- Switch on the Set Configuration panel

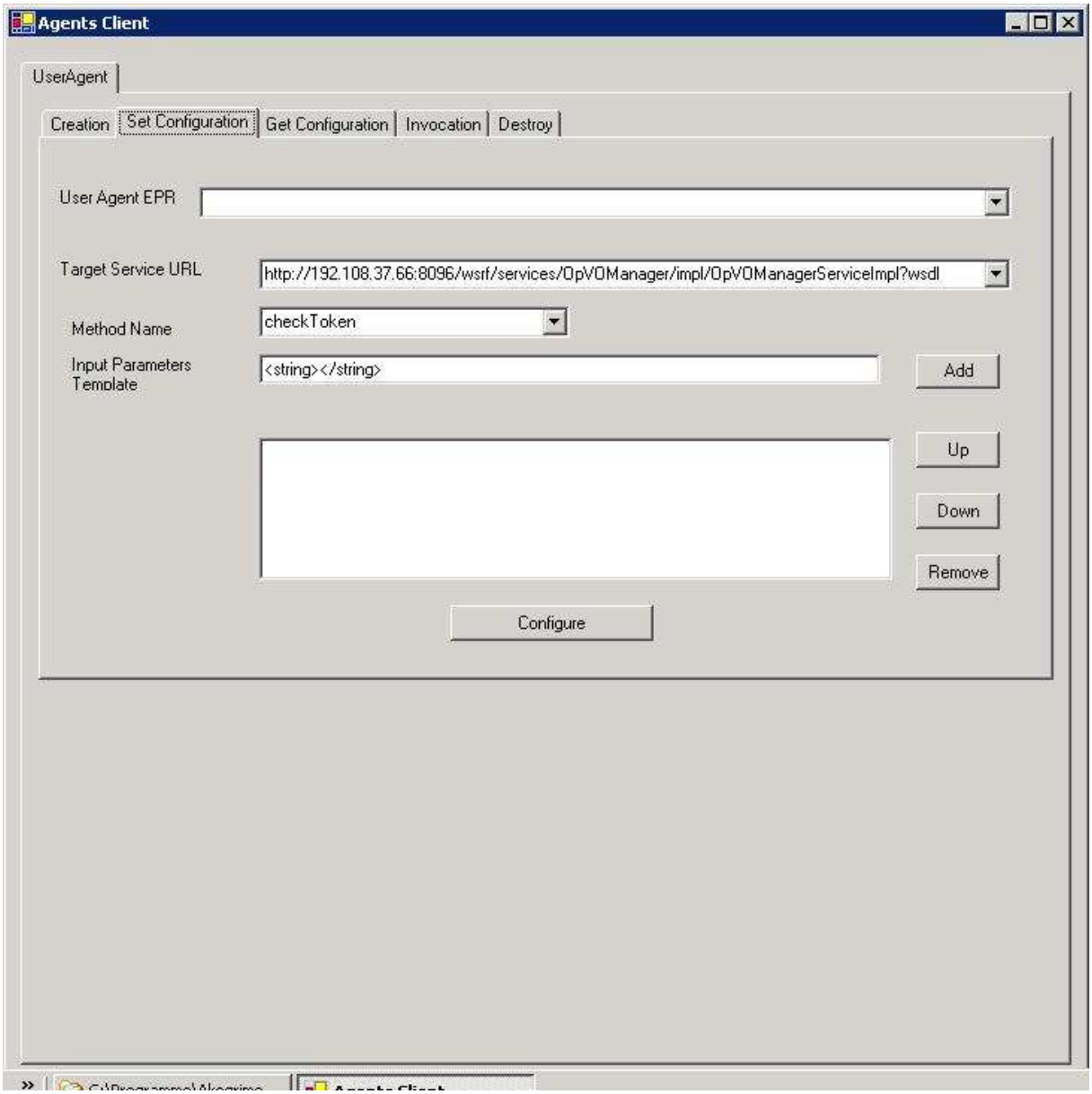

Figure 14: User Agent Configuration

- In **User Agent EPR** specifies the EPR of the UA to configure
- Set the OpVOManager:
  - Target service URL:
    - **http://ksatXX.ipv6.rus.uni-stuttgart.de:8096/wsrf/services/OpVOManager/impl/OpVOManagerServiceImpl?wsdl**
  - Method Name:
    - checkToken
  - Input Parameters Template:
    - <string description="login"></string>, then use Add button
    - <string description="password"></string>, then use Add button

- Click on the **Configure** button.
- Without modify UA EPR, set the Business Workflow properties:
  - Target service URL:
    - http://ksatXX.ipv6.rus.uni-stuttgart.de:8090/usecases/eHealthDemo.wsdl
  - Method Name:
    - startFlow
  - Input Parameters Template:
    - <string description="userId"></string>, then use Add button
    - <string description="doctorId"></string>, then use Add button
- Click on the **Configure** button.

5.3.2. HE Setup Client

In this section we describe how to interact with EMS service to retrieve information useful for SAs configuration. The HE Set Up is performed using a client currently installed in the target machine. You can find the client in:

C:\Programme\Akogrimo\VOManagement\UA and SA services\Client\SA\EnvironmentSetUpClient\EnvironmentSetUpClient\bin\Debug\ EnvironmentSetUpClient.exe

5.3.2.1. Start Set up phase

The client reads from a configuration file **C:\Programme\Akogrimo\VOManagement\UA and SA Services\Client\SA\EnvironmentSetUpClient\EnvironmentSetUpClient\bin\Debug\ SetUpParameters.xml** which contains information about the EMS service and Service Agents. Currently, it contains the correct values for the demo, so we don't have to modify it.

To run HE set up, you have just to start the **EnvironmentSetUpClient.exe** client and wait for its set up message.

5.4. Grid Node 4

5.4.1. SLA Access Configuration

It is possible to find the GUI for testing this component at:

H:\Program Files\Akogrimo\SLA\Client\SLAAccessClient.exe

and follow these steps:

- Start **SLAAccessClient.exe** GUI

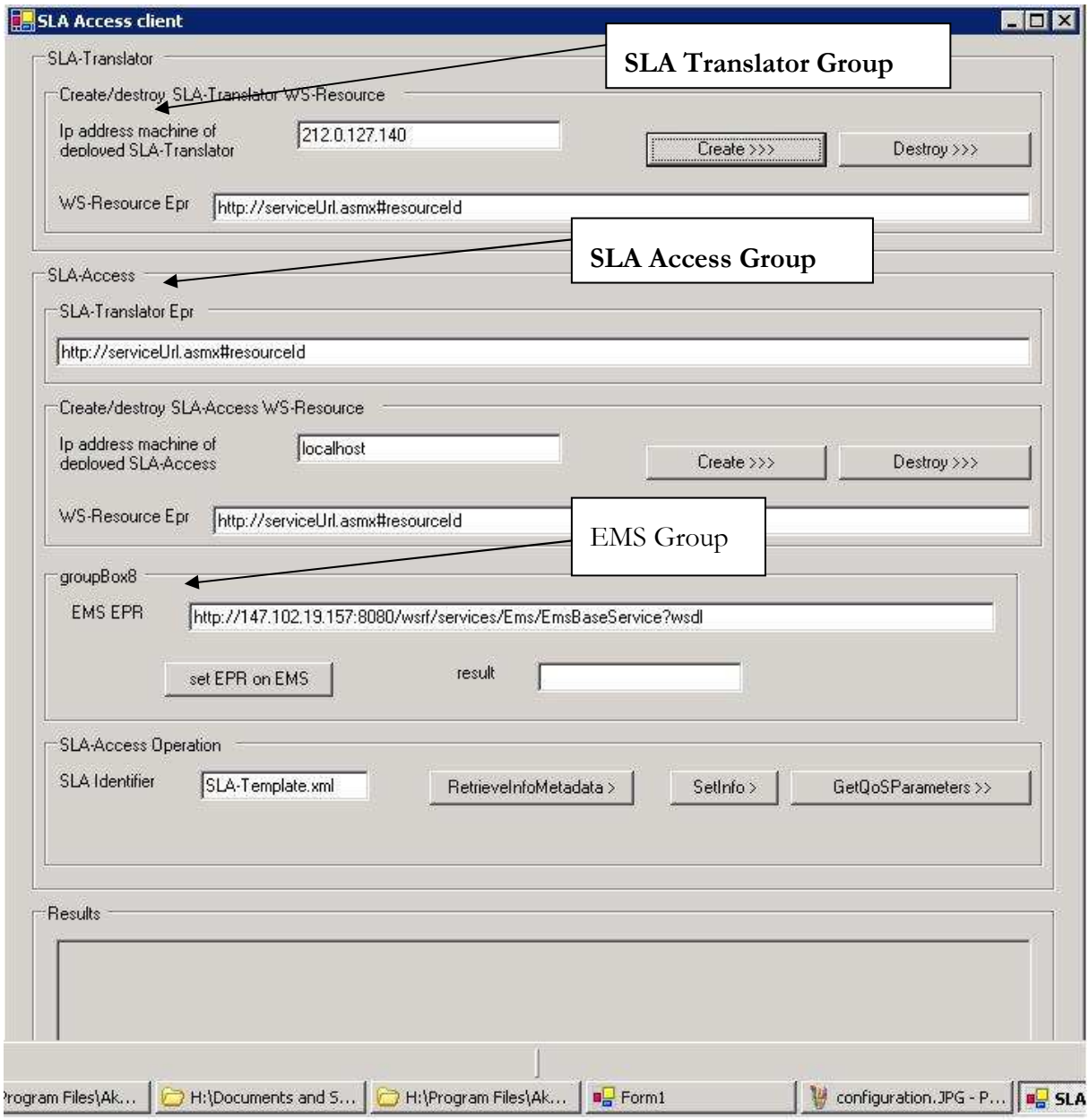

Figure 15: SLA Access Client

- SLA-Translator Group
  - Specify the IP of the machine where the SLA-Translator is running
  - Use **Create** button to start the creation of a SLA-Translator instance
  - In WS-Resource Epr you will see the EPR for the SLA-Translator instance
- SLA Access Group
  - Click on "Create" button. This will create an SLA-Access instance on localhost and will initialize it with the EPR of the SLA-Translator created in the previous steps.
  - In WS-Resource Epr you will see the EPR for the SLA-Access instance

- EMS Group
    - Update the URL of EMS service (it should be machine:**8080**/wsrf/services/Ems/EmsBaseService?wsdl)
    - Click on "Set EPR on EMS" button. This will configure the EMS service with the current instance of the SLA-Acess service

5.4.2. Participant Registry Components

It is possible to find the GUI for testing this component at:

H:\Program Files\Akogrimo\VOManagement\ParticipantRegistry\NewClient\ ParticipantRegistryDemoClient.exe

5.4.2.1. Current Configuration

During the integration meeting, the Participant Registry has been configured according to the BVO Setup, OpVO setup and UpdateParticipantRegistry.

We have created three participants and we have added them to Akogrimo collection:

1. alice@akogrimo.org (doctor role)
2. bob@akogrimo.org (patient role)
3. carol@akogrimo.org (beamer role)

It is possible to use the GUI to retrieve information about this configuration performing the following steps:

1. Start **ParticipantRegistryDemoClient.exe** client
2. Switch to **CheckStettings** panel
3. As ParticipantId value specify the participant identifier ({alice,bob,carol}@akogrimo.org)
4. Use **GetEPR** button to take the EPR of the ParticipantInfo service for the user.

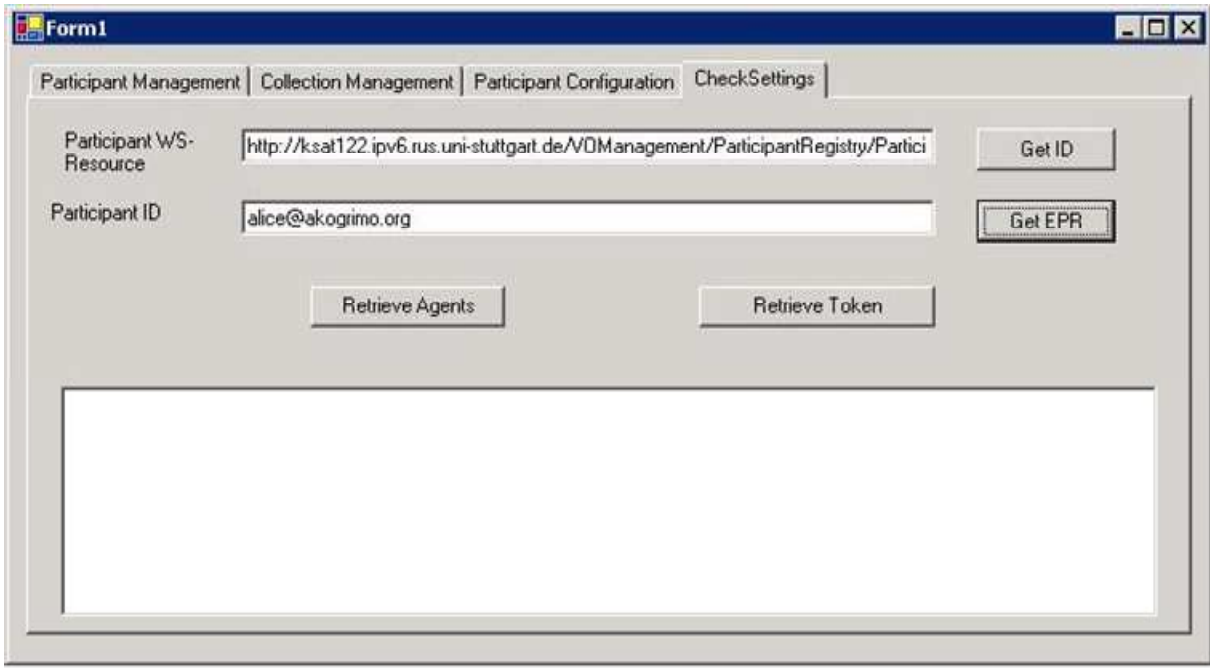

Figure 16: Getting Participant Registry EPR

5. Then use the **RetrieveAgents** button to take the list of User Agents created for this user

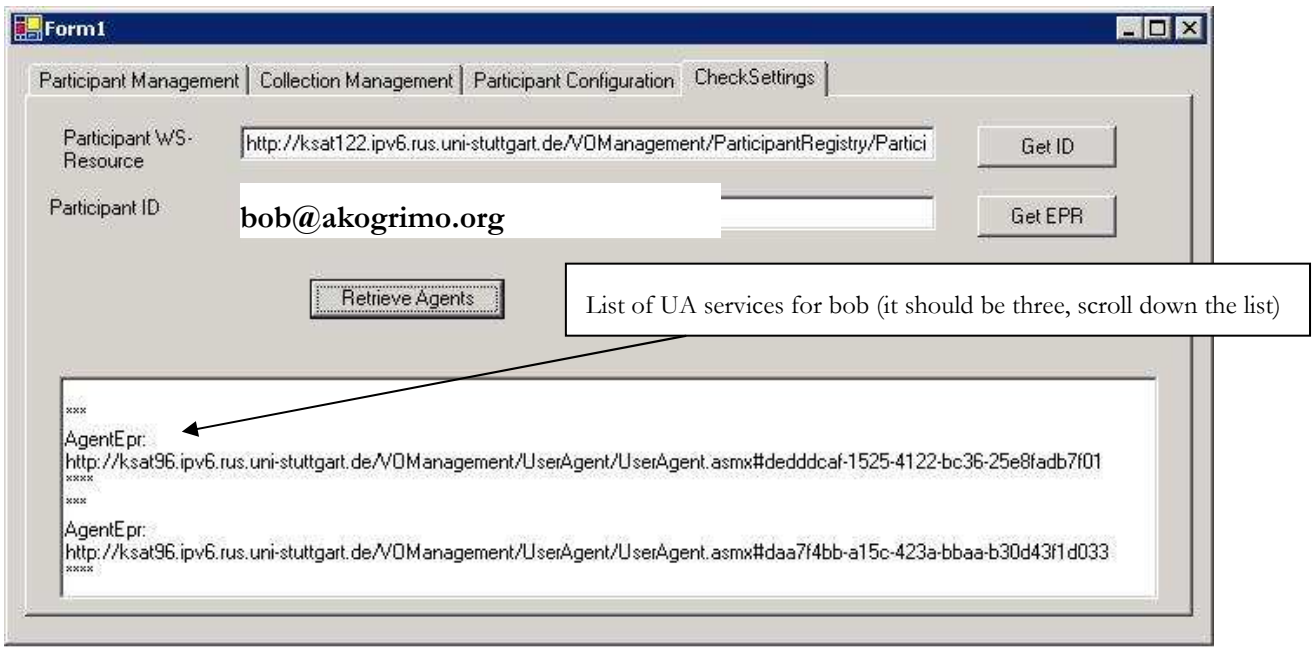

Figure 17: Participant Registry information

6. Or use the **RetrieveToken** button to take the couple of entry (User Agent, token) for the current user

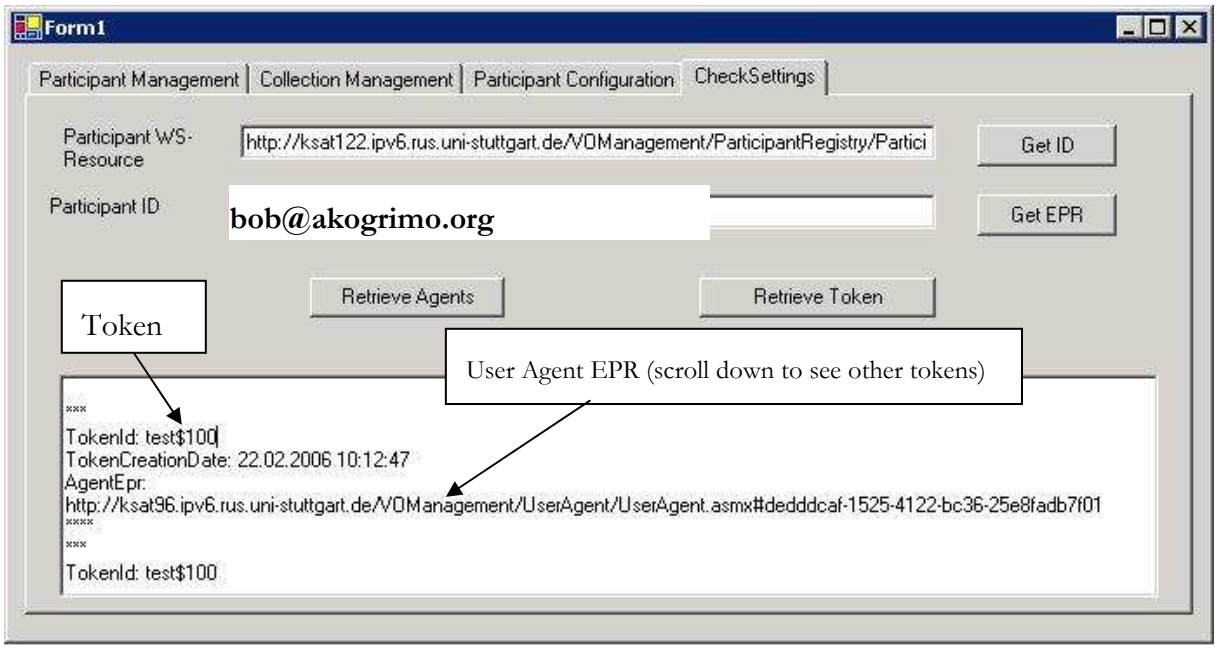

Figure 18: Retrieving Participant Registry Token

5.4.2.2. How to create a new configuration

Below we describe how to use the GUI if some problems occur with the previous configuration.

5.4.2.2.1. Preliminary Steps

- Start **ParticipantRegistryDemoClient.exe** client
- Remove users.

  Switch on ParticipantManagement panel and specify as ParticipantId alice@akogrimo.org (bob@akogrimo.org, carol@akogrimo.org) and the use Remove button.

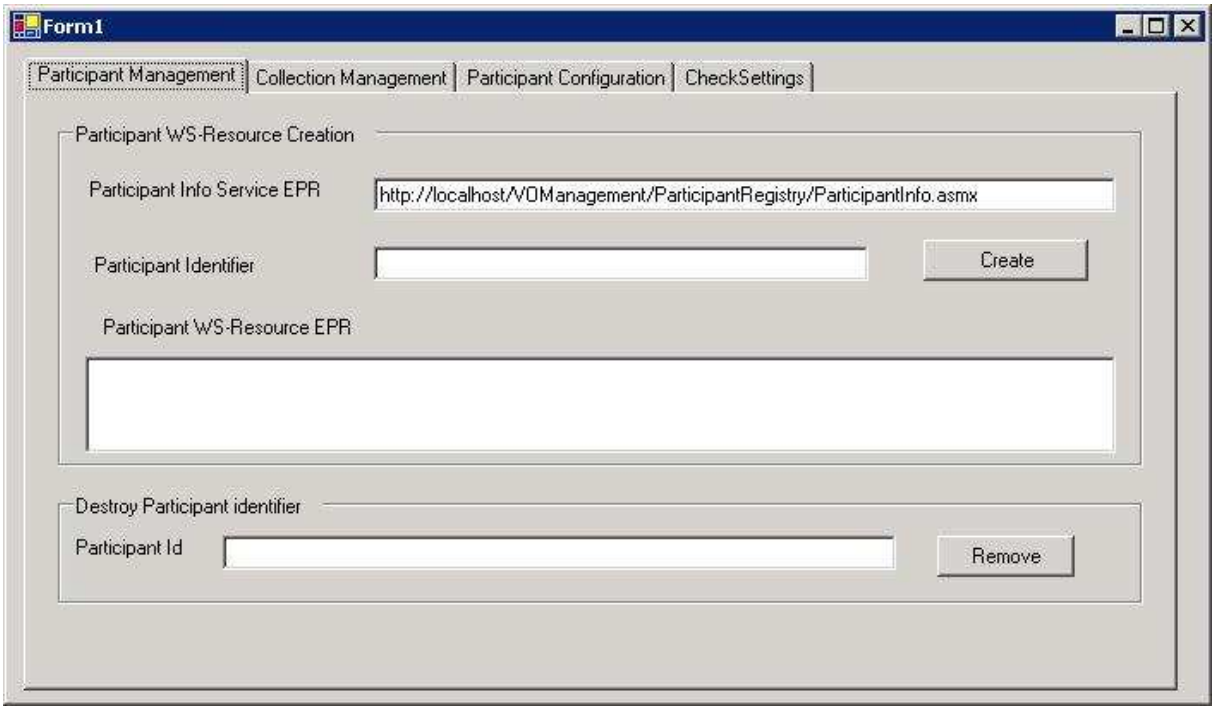

Figure 19: Creating a new Participant Registry Configuration

- For each Akogrimo participant, run steps described in 5.4.1.2.2 – 5.4.1.2.3 – 5.4.1.2.4 - 5.4.1.2.5

5.4.2.2.2. BVO SetUP

- Switch on "Participant Management" panel
- Insert Participant Identifier ({alice, bob, carol}@akogrimo.org)
- Start the creation using **Create** button

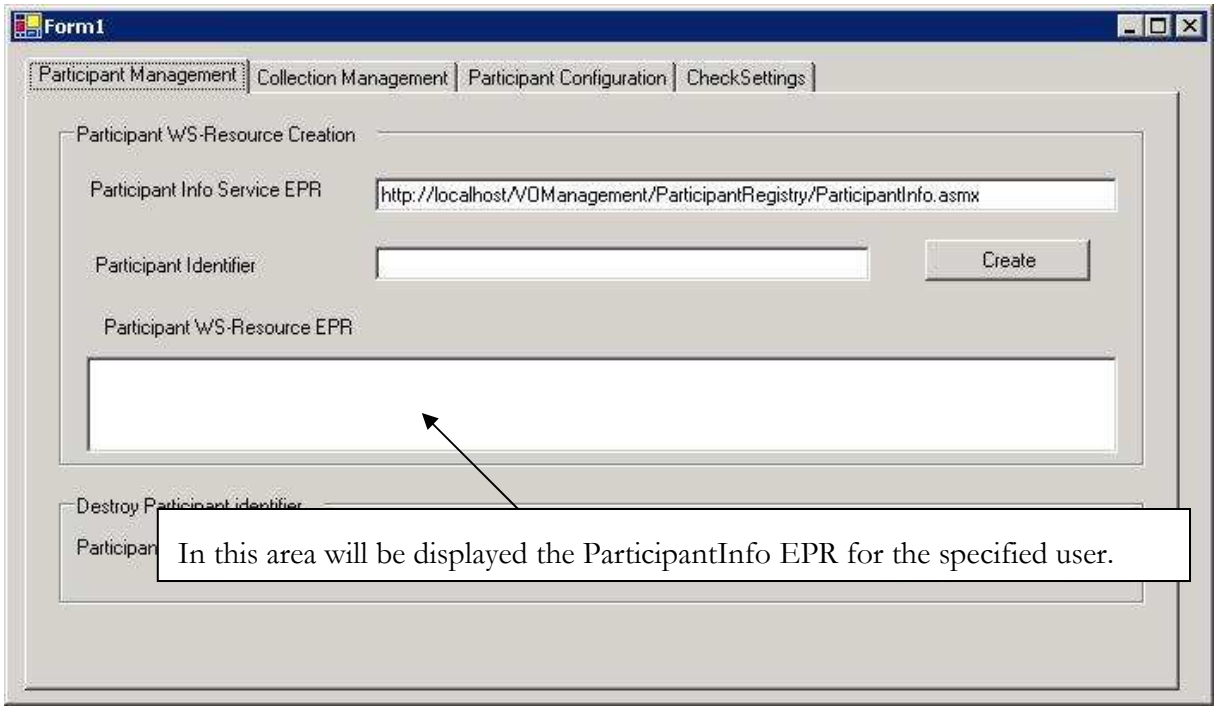

Figure 20: Participant WS-Resource Creation

- Switch on Collection Management panel to add the user to Akogrimo collection. You have just to hit the button **Add to Collection** because the needed values are updated automatically according to the creation phase.

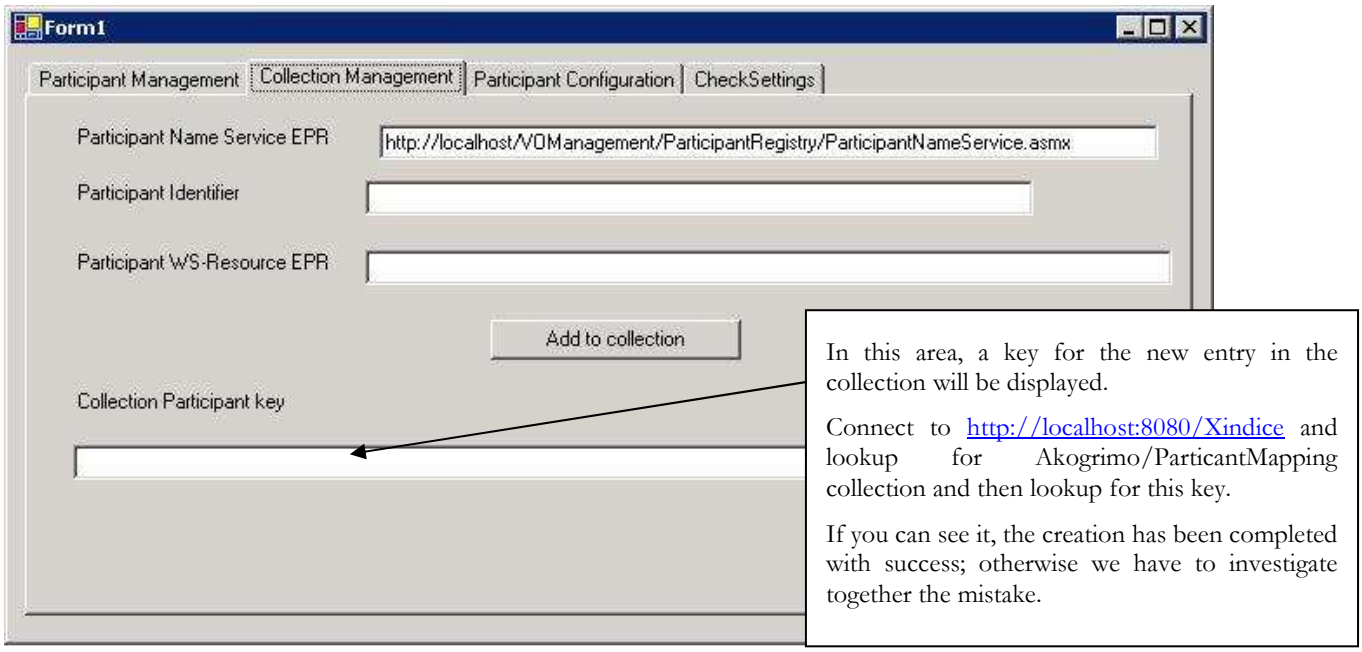

Figure 21: Adding the user to the BVO collection

5.4.2.2.3. OpVO SetUP (UI Client → UA)

This step allows you to create a new User Agent instance, as it is described in section 5.4.2.1. In the GUI will be displayed the User Agent EPR, copy this value because you need of it in the next step.

5.4.2.2.4. Update Participant Registry

- Switch on "**Participant Configuration**" panel.
- Participant WS-Resource EPR automatically shows the EPR for the current user you are configuring. If this value is not correct for you, change it according to your configuration.
- Insert in UserAgent EPR the EPR of the user Agent instance previously created as described in 2.2.3
- Insert in Token the string test$100
- Use Add User Agent and Token button

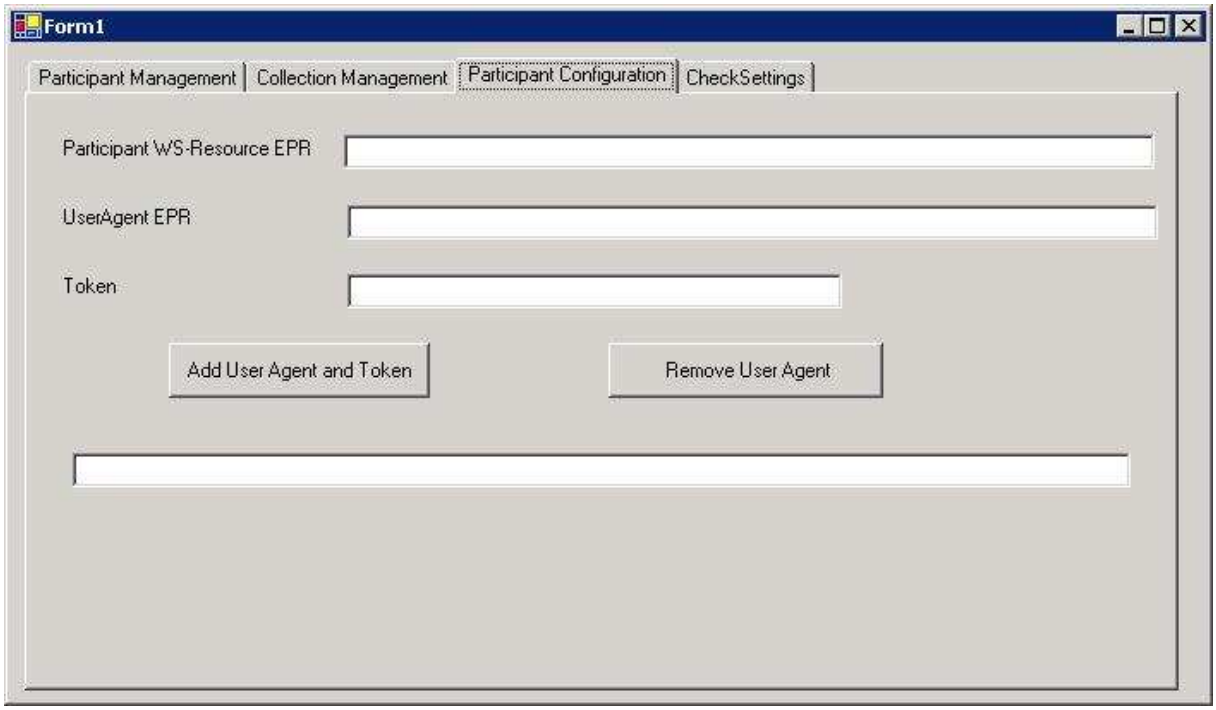

Figure 22: Updating Participant Registry

5.4.2.2.5. Check Configuration (Optional)

Run the steps described in 2.1 to check your configuration.

Now, you have configured an end user as Akogrimo user

ATTENTION: Repeat steps described in 5.4.2.2.2 – 5.4.2.2.3 -5.4.2.2.4 -5.4.2.2.5 for the next user

5.4.3. SLA Decisor and Controller

Both the SLAController and SLADecisor are Grid services deployed in C#.

A factory mechanism provided by a Web Service is used to create a new instance of the SLA Controller. This Web Service resides in the Internet Information Server that is started when the computer is turned on. The EMS is the service responsible of the SLA Controller Instance creations.

The SLA Decisor is a persistent Grid Service. At this moment the SLA Controller instance creates a new instance of the SLADecisor and then builds a notification mechanism, but in the second cycle the SLADecisor will be a persistent Grid Service created in the start up phase.

5.5. Grid Node 5

5.5.1. MedicalDataLogger and ECGDataAnalyzer

These two eHealth services are installed in the Grid Node 5 target machine. They are started following these steps:

1. Log in as user ehealthdemo.
2. Start Tomcat to run the services:

   ```
   /opt/jakarta-tomcat-5.0.30/bin/startup.sh
   ```

3. Change to root user:

   ```
   sudo su
   ```

4. Start Grid FTP Server to send and receive data:

   ```
   /opt/gridftp-4.0.1/sbin/globus-gridftp-server &
   ```

5.5.2. Metering

This component is started with the Globus container script:

```
/opt/ws-core-4.0.1/bin/globus-start-container –nosec
```

6. Logging

Some of the modules available produce logging data. Next table indicates how this information is generated and where it is shown.

Table 2: Logging Information

Module / Application	**Type of logging**	**Details**
Multimedia Application's SIP UA	FILE	"MTSIPUA_log.txt" and "MTSIPUA_debug.txt". The _log file provides information on the underlying SIP traffic, messages sent and received and time when they were sent/received. The _debug file provides data about what is going on in the UA.
SIP infrastructures in MT	SCREEN / FILE	Configurable. Currently shown in the console window where the MT SIP Multimedia Application is started.
MT Testbed Application	SCREEN	A log showing the connection status between MT Application and A4C Server is directly displayed on the console.
ECGDataVisualizer	NO	
ECGDataGenerator	NO	
MIPV6 daemon running on MTs and HA	SCREEN	Standard output. Can be redirected to file
Access Router software	SCREEN	Standard output. Can be redirected to file
QoS Broker software	SCREEN	Standard output. Can be redirected to file
SIP Server	SCREEN / FILE	Configurable. Currently shown in the console window where the SIP Server is started.
SAML Authority	FILE / DB	Tomcat Log file: "/opt/samlauthority/tomcat/logs/catalina.out" Data stored in MySQL, database "samldb", table: "samlstorage"
A4C Server and Client	SCREEN /FILE	By default it is sent to the screen, but it can be configured to be sent to the syslog. The level of detail can also be configured.
SIP Grid Gateway	SCREEN / FILE	Configurable. Currently shown in the console window where the SIP Grid Gateway is started.
SIP Grid Gateway SOAP Interface	FILE	Tomcat log file "/opt/tomcat/logs/catalina.out"

SIP Grid Gateway's SIP UA(SIP Grid GW)	FILE	Two files named "sipgridgw_log.txt" and "sipgridgw_debug.txt". The _log file provides information on the underlying SIP traffic, messages sent and received and time when they were sent/received. The _debug file provides data about what is going on in the UA.
Context Manager	SCREEN + FILE	CM consists of four processes plus a WS interface. - Context Engine: "/var/log/ContextEngine.log" - SLP GW (Gateway): "/var/log/slpgw.log" - SIP Presence GW: "/var/log/CM_sippres.log" - RFID GW: currently no log file, output is to screen. To be done: Logfile at "/var/log/rfidgw.log" - ContextManagerWS: Activity is logged in the tomcat log at "/opt/apache.../log/catalina.out"
RFID Server	SCREEN + FILE	File is located at "<homedir>/rfidserver.log " where <homedir> is "/home/contextmanager/rfid/" or (most likely) "/home/boss/rfid", depending on the installation.
LSDS	FILE	One log file: "/var/log/slpd.log"
EMS	SCREEN	In the terminal on which the Globus WS-Container has been started
Monitoring	FILE	Use of log4j "/opt/akogrimo/monitoring/data/monitoring.log"
BaseVO Manager Service	NO	It might be considered in the next version
OpVO Manager Service	NO	It might be considered in the next version
Monitoring Daemon	NO	It might be considered in the next version
Workflow Manager	NO	It might be considered in the next version
Data Manager	FILE	Located at "/tmp/log4j-akogrimo.log"
Workflow Registry	FILE	Tomcat log file: "/home/wf/jakarta-tomcat-5.0.28/logs/catalina.out"
Workflow Engine	FILE	Log of the executed workflow. it could be downloaded by the web page that shows the process execution (http://192.108.37.74:8090/BpelAdmin/active_processes.jsp)
Service Agents	FILE	Several files (two for each SA): C:\Inetpub\wwwroot\EcgDataAnalyzer\InputTrace.webinfo

		C:\Inetpub\wwwroot\EcgDataAnalyzer\OutputTrace.webinfo C:\Inetpub\wwwroot\EcgDataVisualizer\InputTrace.webinfo C:\Inetpub\wwwroot\EcgDataVisualizer\OutputTrace.webinfo C:\Inetpub\wwwroot\Medicaldatalogger\InputTrace.webinfo C:\Inetpub\wwwroot\Medicaldatalogger\OutputTrace.webinfo C:\Inetpub\wwwroot\ECGDataGenerator\InputTrace.webinfo C:\Inetpub\wwwroot\ECGDataGenerator\OutputTrace.webinfo
User Agent	FILE	Three files: C:\Inetpub\wwwroot\VOManagement\UserAgent\InputTrace.webinfo C:\Inetpub\wwwroot\VOManagement\UserAgent\OutputTrace.webinfo C:\Programme\Akogrimo\VOManagement\UA and SA services\Log\UA\UserAgent.log
SLA Controller	FILE	Two XML files located at: "H:\akogrimo\slacontroller". The first log (Output) stores the QoSvalue provided by the monitoring; The second (OutputLog) is an internal log file.
SLA Decisor	FILE	XML file stored at "H:\akogrimo\slaDecisor"
SLA Translator	FILE	XML file stored at "H:\akogrimo\log"
ParticipantRegistry	FILE	Two files ("ParticipantRegistryInputSoapTrace.webinfo" and "ParticipantRegistryOutputSoapTrace.webinfo") located at: "H:\Inetpub\wwwroot\VOManagement\ParticipantRegistry" A third file "ParticipantRegistry.log" at: "H:\Program Files\Akogrimo\VOManagement\ParticipantRegistry\Log"
SLA Access	FILE	Two files ("SLAAccessInputSoapTrace.webinfo" and "SLAAccessOutputSoapTrace.webinfo") located at: "H:\Inetpub\wwwroot\SLAManagement\SLAAccess" A third file "SLAAccess.log" at: "H:\Program Files\Akogrimo\SLA\Log"
ECGDataAnalyzer	FILE	Tomcat Log: /opt/jakarta-tomcat-5.0.30/logs/catalina.out
MedicalDataLogger	FILE	Tomcat Log: /opt/jakarta-tomcat-5.0.30/logs/catalina.out
Metering Service	FILE	Tomcat Log: /opt/jakarta-tomcat-5.0.30/logs/catalina.out

7. Run Order

This section defines the order to follow when running the different modules when dependencies exist:

1. Mobile IPv6 software in the Home agent and the Mobile Terminals.
2. Mobile IPv6 Domain Name System (/etc/init.d named start)
3. Local Service Discovery Server
4. SAML Authority
5. A4C server
6. SIP Server
7. SIP Grid Gateway
8. Context Manager
9. Execution Management System
10. Monitoring
11. Monitoring Daemon
12. BVO Manager
13. OpVO Manager
14. ParticipantRegistry, SLA Access, UA and SAs.
15. Workflow Engine.
16. E-health testbed application (services and GridFTP)
17. Data Manager.

References

[1] D5.1.1 Architecture Integration Report

Annex A. XML configuration files

This annex describes briefly the content of the different configuration files of the SIP Grid Gateway and the Multimedia SIP application, indicating the fields that must be adjusted.

A.1. SIP Grid Gateway configuration file

The SIP Grid Gateway configuration file can be found at:

`/opt/akogrimo/sip_grid_gw/SIPGridGw.xml`

Its content is the following (subkeys have been indented for clarification purposes):

```
<?xml version="1.0"?>
<!-- SIPGridGw configuration file-->
<Akogrimo>
      <brokerengine
            rmiport="1127"
            debugmode="2"
            logfile="sipgridgwbe.txt">
      </brokerengine>
      <sipua
            id="SIPGridGw"
            domain="akogrimo.org"
            display="SIPGridGw"
            transport="UDP"
            ip="[2001:638:202:e00:230:1bff:feb5:bb58]"
            port="5070">
      </sipua>
      <proxy
            ip="[2001:638:202:e00:213:d4ff:fe36:dab1]"
            port="5060">
      </proxy>
      <A4C
            library="/usr/lib/jni/libA4CClientJava.so"
            xmlfile="/etc/a4c/nas_SIPGridGw.local.xml"
            server="ksat116.ipv6.rus.uni-stuttgart.de">
      </A4C>
</Akogrimo>
```

In principle, this information should not be modified. Please be sure that the SIP User Agent IP address being used (key "sipua", subkey "ip" appears in the /etc/hosts file with the alias of the machine). Some decisions have been adopted to avoid conflicts with the Context manager, which is hosted in the same machine:

- The SIP User Agent listening port should be 5070 to avoid conflicts with the Context Manager SIP User Agent, which is using default 5060.
- RMI port has been chosen also to avoid conflicts with the Context Manager (currently using the default RMI port, 1099).
- The A4C client interface is using port 1819 because Context Manager is using the default 1811 (see `/etc/a4c/nas_SIPGridGw.local.xml` file).

Finally, it is important not forget to check the correctness of the SIP proxy address (which is currently listening on that address and port).

A.2. Multimedia Application configuration file

After first installation, application installs the default.xml file. Each time a new user is registered using the same machine, a new file named <AkogrimoUser@domain>.xml is created in the same directory. The content of the default file is as indicated below (subkeys have been indented for clarification purposes):

```
<?xml version="1.0"?>
<!-- User configuration file-->
<Akogrimo>
      <user
            akoid="-- Enter Akogrimo ID"
            pwd=""
            savepwd="false">
      </user>
      <sipua
            id="--Enter SIP URI"
            domain=""
            display=""
            localtag="gdg76t-tfd267te-saduy"
            transport="UDP"
            ip="[2001:638:202:e11:213:ceff:fe5b:fb73]"
            port="5060">
      </sipua>
      <A4C
            library="/usr/lib/jni/libA4CClientJava.so"
            xmlfile="/etc/a4c/nas_MTSIPMMApp.local.xml"
            server="ksat116.ipv6.rus.uni-stuttgart.de">
      </A4C>
      <proxy
            ip="[2001:638:202:e00:213:d4ff:fe36:dab1]"
            port="5060">
      </proxy>
      <buddylist>
```

```
	</buddylist>
	<sessiondata>
		<!—- Type 0 indicates MEDIA session-->
		<session type="0">
			<!—- Media session Type 1 indicates AUDIO-->
			<media
				type="1"
				tx="22222"
				codec="0/3/4/5/14/16/17">
			</media>
			<!-- Media session Type 2 indicates VIDEO-->
			<media
				type="2"
				tx="22226"
				codec="26/32/34">
			</media>
		</session>
	</sessiondata>
</Akogrimo>
```

It is important to check that the SIP User Agent IP address being used (key "sipua", subkey "ip") is the right one and it appears in the /etc/hosts file with the alias of the machine. If not, JMF library will not work properly. Also check the IP address and listening port of the SIP Server.

Note that the A4C client interface is using port 1819 because eHealth application is using the default 1811 (see `/etc/a4c/nas_MTSIPMMApp.local.xml` file).

Also check that each MT is using a different port to transmit RTP audio and video data. The best practise is to have a gap of 4 between the audio and video transmission port (to enable the debug version to work) and to have 10 units of difference between the different MTs (e.g. 222XX for patient, 223XX for the doctor´s MT and 224XX for the beamer).

Note the set of codecs (identified by the RTP payload) that the linux version of the JMF library can transmit and receive. Not all codecs are supported by all devices.

Annex B. Context Manager Installation

B.1. Presumptions

- `Contextmanager*.deb` is available.
- Linux Ubuntu PC with necessary software is available, including (see Table 1 for further details)
  - Ant
  - gcc 4.0
  - Java 1.5+
  - MySQL 4.1.10a
  - Tomcat and Axis (see Table 1)
- SLP:
  - OpenSLP v1.3.0 libraries must be available.
  - The JNI classes have to run on a Linux machine.
  - The SLP DA can be on a separate machine.
- Database: MySQL 4.1 is installed. Note: User and password must be updated in the configuration for CM, see Section 4.6.2.2.
- RMI Registry is started. Start it from where installed, e.g.
  - Windows: C:\Program Files\Java\jdk1.5.0_06\bin\rmiregistry.exe
  - Linux: Execute `rmiregistry` from the command line
- No special requirements wrt IP4/IP6

B.2. Compile/build

Context Manager is distributed as a debian package (contextmanager*.deb) ready to install. If this is the case for you, proceed to Section 0. If you wish to compile and build from source, follow instructions in this section.

Go to directory where the file `build.xml` for Context Manager is located. Normally this is
`c:\akogrimo\dev\akogrimo\wp42\contextmanager> (Windows)`
or
`\user\akogrimo\dev\akogrimo\wp42\contextmanager> (Linux)`

`ant` is used to compile, build, and create installation files contextmanager.zip and contextmanagerwin.zip

From the command prompt (same directory as build file), execute the following command:

```
$> ant target
```

Table 3: Available targets for `ant` in CM installation

Target	Explanation
compile	compiles project
jar (default)	executes target "compile", then creates .jar file

clean	cleans everything: All the built stuff and debug files.
install	executes target "jar", then creates install directory and copies needed files
dist	executes target "install", then creates contextmanager.zip and contextmanagerwin.zip file from install directory

To make the debian package:

- `$>ant install` will create a folder `install`
- Copy `install/contextmanager` to a Linux machine
- On Linux, run the `./makedeb` script

B.3. Install

B.3.1. General

This procedure describes installation on Linux.

- Download (e.g. from Forge at HLRS) `contextmanager*.deb` and place it in a temporary folder on the machine where you want to install Context Manager.
- Install by doing `dpkg -i <nameoffile>.deb`
- This will create Context Manager -folders with files and subfolders, see below for locations.
- Review and if necessary change the configuration files.

Table 4: File locations after CM installation

File type	Folder	Files
Applications (.jar)	/usr/share/contextmanager	contextengine.jar, rfidgw.jar, sippresencegw.jar, slpgw.jar
Database definition (.sql)	/usr/share/contextmanager/database	ContextDb.sql
Configuration files (.properties, .xml, .all)	/etc/contextmanager	CM_sippres.log.properties, policy.all, ContextEngine.log.properties, sip-communicator.xml, ContextManager.properties,

		slpgw.log.properties
Log files (.log)	/var/logs	CM_sippres.log, ContextEngine.log, slpgw.log Note: Axis logs under $tomcat_home[2]/logs
Web services	$tomcat_home/webapps	ContextManagerWS.war, ContextConsumerWS/*

B.3.2. Database

Install the database by executing the sql /usr/share/contextmanager/database/ContextDb.sql in MySQL.

- MySQL Query Browser: File – Open Script – Execute
- On the command line:

```
$>  mysql -u root –p
(enter root password for mysql)
mysql> source /usr/share/contextmanager/database/ContextDb.sql;
mysql> quit;
```

B.3.3. SLP

After the ContextManager has been installed (see Section **¡Error! No se encuentra el origen de la referencia.**) on the target machine, the following must be done to compile the native C code:

a) Change directory to /usr/share/contextmanager/

```
$> cd /usr/share/contextmanager/
```

b) It is required that Makefile is updated with the correct path to openslp and java

Paths: Makefile uses the following variables that have to be changed if openslp or java are installed at different locations.

```
openslpdir = /usr/local/openslp-1.3.0/
javadir = /usr/lib/j2sdk1.5-sun/
```

c) Compile native C code:

```
$> make
```

[2] $tomcat_home is /opt/apache-tomcat-5.5.12 at ksat97

Annex C. RFID Server Installation

C.1. Presumptions

Hardware:

- 2 pieces of RFID reader type SCR311 DI, made by SCM Microsystems: http://www.epsys.no/scm/scm_readers.htm#scr331di
- 2 fitting Smart Cards with RFID tags.

Software:

- RFIDserver available either in zip format or in deb format.
- Linux Ubuntu PC
- Contextmanager installed and running on a computer accessible by network.
- Ant
- Gcc
- Java 1.5
- PCSC dameon installed and running (packages pcscd and pcsclite1-dev)

C.2. Compile/build

RFIDServer will be distributed as a debian package (rfidserver*.deb) ready to install. If you wish to compile and build from source follow the instructions in this section.

Go to the root directory of the RFIDServer source files. Execute the following commands:

`make` - to compile native code needed for communication with the RFID readers.

`ant` - to compile java code.

If you wish to make debian package and are on a linux machine:

```
ant install
./makedeb
```

C.3. Install

You need to do the following in order to install this Linux.

- Download rfidserver*.deb
- Install by `dpkg –i <nameoffile>.deb`
- Review and if necessary change the configuration files.

Log files will be placed in `/var/log/` folder named `rfidserver.log`

Configuration files are in `/etc/rfidserver/` and are `RFIDserver.log.properties` and `reader.positions.config`