# D5.3.3

## Updated Architecture Evaluation Report

Version 1.0

## WP 5.3 Architecture evaluation and assessment

## Dissemination Level: Public

Lead Editor: Giuseppe Laria, CRMPA

22/10/07

Status: Final

**SIXTH FRAMEWORK PROGRAMME**

**PRIORITY IST-2002-2.3.1.18**


*Grid for complex problem solving*

*Proposal/Contract no.: 004293*

This is a public deliverable that is provided to the community under the license Attribution-NoDerivs 2.5 defined by creative commons http://www.creativecommons.org

**This license allows you to**

- to copy, distribute, display, and perform the work
- to make commercial use of the work

**Under the following conditions:**

**Attribution**. You must attribute the work by indicating that this work originated from the IST-Akogrimo project and has been partially funded by the European Commission under contract number IST-2002-004293

**No Derivative Works**. You may not alter, transform, or build upon this work without explicit permission of the consortium

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

This is a human-readable summary of the Legal Code below:

*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 all partners of the Akogrimo consortium that have participated in the production of this text
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.
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 5.3** | **Architecture evaluation and assessment** |
| **Task 5.3.1** | **Qualitative architecture evaluation** |
| **Dependencies** | **This deliverables uses specifically the input of the deliverables D5.3.2, D5.3.2b, D3.1.3 and D4.x.3** |

| **Contributors:** | **Reviewers:** |
|---|---|
| **CRMPA**:<br>**TID**: Mario Del Campo Melgar<br>**USTUTT**: Robert Piotter, Patrick Mandic<br>**ITAv**: Nuno Inacio<br>**ITInn**: Alfonso Sanchez Mancian, Mike Boniface<br>**CCLRC**: Julian Gallop, Tom Kirkham | Annalisa Terracina (ElsagDatamat)<br>Patrick Ratz (UniZh)<br>Francesco D'Andria (ATOS)<br>Kleopatra Kostanteli (NTUA)<br>Fredrik Solsvik (TN) |

**Approved by: Victor Villagra (QM)**

| Version | Date | Authors | Sections Affected |
|---|---|---|---|
| 0.1 | 03/12/07 | CRMPA | Template |
| 0.2 | 22/12/07 | CRMPA | Section 1, section 5 |
| 0.4 | 31/01/07 | CRMPA | Section 5, introduction to all sections |
| 0.5 | 16/02/07 | All | Section 2, section 3 |
| 0.6 | 02/03/07 | All | Section 4 |
| 0.7 | 13/04/07 | CRMPA | Template updated to include a gap analysis section |
| 0.8 | 11/05/07 | All | First version of section 6 |
| 0.9 | 18/06/07 | All | Updated version of section 6 |
| 0.91 | 22/06/07 | CRMPA | Executive Summary, Conclusions. Final draft ready |
| 0.92 | 18/10/07 | CRMPA | Implementation of internal review feedbacks |
| 1.0 | 22/10/07 | CRMPA | Final version after Quality Check |

# Table of Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **3PCC** | Third Party Call Control |
| **Akogrimo** | Access To Knowledge through the Grid in a Mobile World |
| **API** | Application Programming Interface |
| **A4C** | Accounting, Authentication, Authorization, Auditing, Charging |
| **BP** | Business Process |
| **BVO** | Base Virtual Organization |
| **DHCM** | Disaster Handling and Crisis Management |
| **EAP** | Extensible Authentication Protocol |
| **ECG** | Electrocardiogram |
| **EMS** | Execution Management Service |
| **FS** | File System |
| **GrSDS** | Grid Service Discovery Service |
| **GSI** | Grid Security Infrastructure |
| **GT4** | Globus Toolkit version 4 |
| **LDS** | Local Discovery Service |
| **MDS** | Meta Directory System |
| **Mgr** | Manager |
| **MT** | Mobile Terminal |
| **NAS** | Network Authentication Service |
| **OGSA** | Open Grid Service Architecture |
| **DAI** | Data Access and Integration |
| **OpVO** | Operative Virtual Organization |
| **PANA** | Protocol for carrying Authentication for Network Access |
| **PKI** | Public Key Infrastructure |
| **QoS** | Quality of Service |
| **REC** | Recommendation |

| | |
|---|---|
| **RFT** | Reliable File Transfer |
| **SA** | Service Agent |
| **SAML** | Security Assertions Markup Language |
| **SIP** | Session Initiation Protocol |
| **SLA** | Service Level Agreement |
| **SOA** | Service Oriented Architecture |
| **SOAP** | Simple Object Access Protocol |
| **SSL** | Secure Sockets Layer |
| **TCP** | Transmission Control Protocol |
| **UA** | User Agent |
| **VO** | Virtual Organization |
| **WS** | Web Service |
| **XML** | Extensible Markup Language |

# Executive Summary

This deliverable presents the evaluation of the final architecture design. The results have the goal of providing final feedbacks to be taken into account for further improvements of Akogrimo infrastructure after the end of the project.

This document starts from work done in [1] and evaluates if the provided recommendations have been addressed by the updated architecture design. The analysis show that most of the identified weaknesses have been covered even if areas for further improvements still exist. In particular, it is clear that several changes have been introduced in order to provide features for improving non functional requirements such as availability, scalability and performance, but some additional work need to be done in order to allow the exploitation of Akogrimo results in real life applications.

As a part of the update of the scenario based evaluation, some additional analysis have been performed, in particular:

a. Evaluation of the missing capability in order to meet requirements coming from the demonstrator scenario: this analysis has identified some necessary changes to be introduced in the final version of the design in order to meet some key requirements related to the need of having an OpVO not centrally controlled through a workflow. These changes are not particularly invasive but they are necessary to run the demonstrator appropriately.

b. Evaluation of the countermeasures provided by the design with respect to some possible threats arising from the study of the attack model: this analysis resulted in the selection of one attack model (from user to Base VO or OpVO) for a more detailed investigation through the definition of tests to be performed on the final demonstrator.

c. Gap Analysis of the Akogrimo architecture: the first problem to perform this gap analysis resulted from the identification of the target architecture to be used for evaluating the gap. In order to overcome this problem the OGSA architecture has been taken as target architecture enriching it with additional capability to cover the mobile paradigm. The evaluation shows that Akogrimo design presents some gaps (in particular, with respect to capabilities related to self management and resource management areas).

Finally a more detailed analysis of the tests reported in [4] has been done in Annex A and it supports the evaluation that recommends detailed design of solutions meeting stronger performance and availability requirements in order to support future exploitation of Akogrimo in real case application.

# 1. Introduction

This document provides the final results of the architecture evaluation process.

The assessment focuses on understanding at which extent the final Akogrimo architecture design has addressed the recommendations provided in D5.3.2a (see [1]) providing final feedbacks for design improvements beyond the Akogrimo project life cycle.

Apart from the above general focus this document includes some additional information that provide:

- A qualitative evaluation about the capability of the current architecture to address the requirements introduced by the final demonstrator scenario that has been designed merging the different test beds: Disaster Handling and Crisis Management, eHealth and eLearning.
- Extended analysis of first prototype testing results in order to provide detailed feedbacks about what the causes/consequences could be of specific results.

The overall goal is to collect the evaluation and testing results achieved so far in order to compare them with the final version of Akogrimo architecture design: the result is an analysis that gives a first comprehensive assessment of the architecture with respect to the goal to create a blueprint for a Next Generation Grid embracing the mobility paradigm. This analysis will be further improved in the final output of WP5.3 when the test beds will be run on the final infrastructure and the validation process will be performed as well.

In order to achieve the above objectives this documents uses the results of other deliverables, in particular:

- D5.3.2a (see [1]) that includes the recommendations resulting from the first qualitative evaluation and references to the initial requirements and evaluation criteria used to carry out that recommendations
- D5.3.2b (see [4]) that provides the results of first prototype testing and preliminary conclusions about the results of those tests
- D3.1.3 (see [2]) that is the final overall architecture design. Actually, this document evaluates the final architectural choices explained in D3.1.3
- ID4.1.3, ID4.2.3, ID4.3.3, ID4.4.3 (see [5][6][7][8]) that provide details about the architecture choices explained in [2]. These internal deliverables have been used when the overall description did not provide enough details to perform an evaluation.
- D2.3.4 that describes the DHCM scenario and then the additional requirements to be met by Akogrimo

## 1.1. Placement in the Akogrimo Project

This report is a deliverable of the Akogrimo work package WP5.3 "Architecture Evaluation and Assessment" and it is related to the second evaluation cycle. Together with the other official report planned in this evaluation cycle (D5.3.4) it provides a first set of feedbacks on the final achievements of the project related to the architecture design. As already mentioned, the goal is to provide assessments and recommendations for improvements that can be covered beyond the end of the project, contributing to provide understanding about the final existing gap to make the Akogrimo outcomes useful for commercial exploitation.

With respect to the overall Akogrimo, the deliverable is strictly related to WP3.1 (architecture design), WP2.3 (scenario definition), WP5.2 (test bed definition) and the workpackages of activity 4 (Detailed architecture design and implementation).

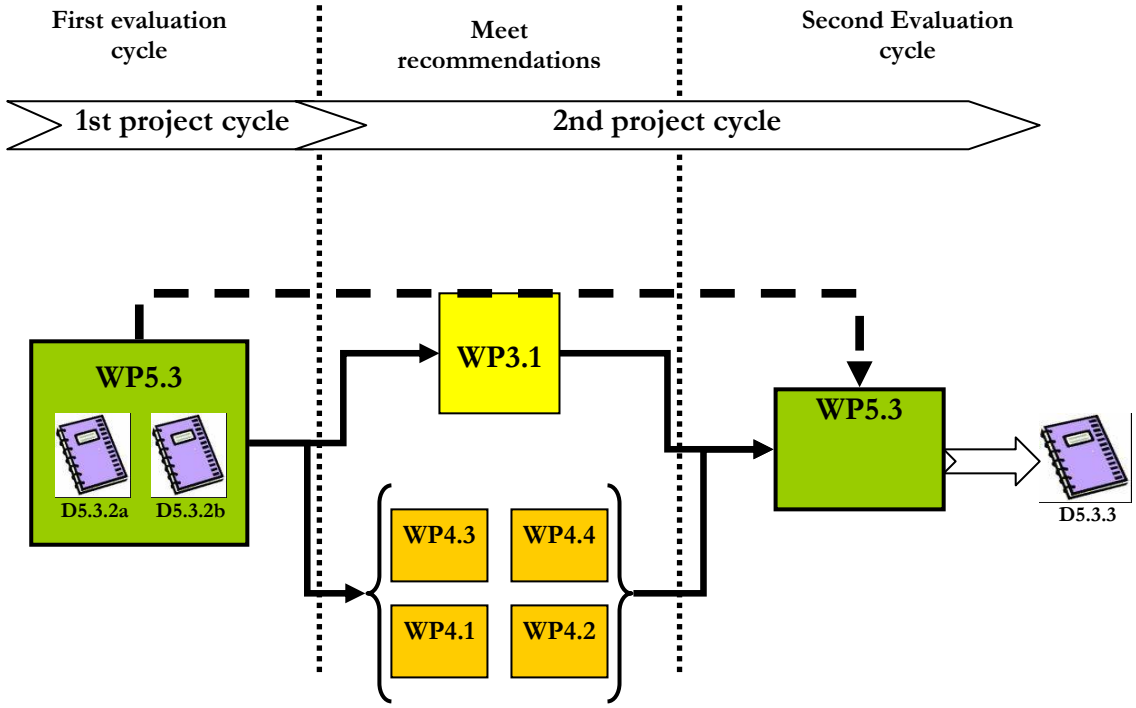


**Figure 1 - Dependencies between D5.3.3, other WPs and project cycles**

This document represents the first step of the second evaluation process: it consists of collecting the previous evaluation results (previous WP5.3 outcomes) and to evaluate if they have been met in WP3.1 and WP4.x. Furthermore additional requirements coming from WP2.3 and WP5.2 will be taken into account as well.

The steps of the second evaluation cycle will be:

- Testing and evaluation of the final infrastructure prototype collecting results in the internal report "Final Prototype Evaluation Report"
- Validation of the infrastructure running the test beds. Results will be collected in D5.3.5 "Test Bed Evaluation Report"

## 1.2. Content and Document Structure

The document is organized as follows:

- *Section 1 (Introduction)*: is the current section;
- *Section 2 (Recommendations and updated architecture design)*: provides an assessment of the final architecture design with respect to the goal of meeting the recommendations in D5.3.2a.
- *Section 3 (Akogrimo Attacks model)*: provides an evaluation of the countermeasures provided by the design with respect to some possible threats.
- *Section 4 (Additional requirements coming from demonstrator scenario)*: evaluates the architecture with respect to the additional requirements introduced by the DHCM scenario
- *Section 5 (gap analysis)*: provides a gap analysis using as target architecture the OGSA model enriched with additional requirements to cover the mobile paradigm.

- *Section 6 (final conclusions)*: summarizes the evaluations with respect to the non functional attributes providing the final assessment of the Akogrimo.

# 2. Recommendations met by the updated architecture design

This section assumes a general understanding of the qualitative architecture evaluation approach followed in Akogrimo and of the related evaluation results (for details see respectively [3] and [1]).

In particular, the qualitative evaluation results summarized in D5.3.2a arose at the end of the first project cycle some recommendations that had to be met by the architecture design update.

The goal of this section is to understand at which extent those recommendations were met. At this aim, the following subsections report the subset of validation scenarios that have required for architecture changes as result of the evaluation process. This subset is reported providing here for each of those scenarios the related tables already filled in and extracted by D5.3.2a ([1]).

Here the meaning of these tables is explained again for a better readability of the document. They did have the general objective of describing the potential weakness of the architecture with respect to the defined evaluation scenario, proposing possible changes to the design itself evaluating the estimated cost to introduce them.

Table 1 includes five different columns labelled as:

- Scenario ID: provides the unique identifier that refers to the scenario across the different deliverables
- Involved components: makes a list of Akogrimo architecture components involved in order to carry out the related scenario
- Description of interactions among components: provides a brief description about how the different components interact together to achieve the scenario
- Required changes: describes the identified changes to be introduced in order to meet the scenario
- Cost estimation: gives a rough estimation of the cost necessary to introduce the proposed changes. Six possible values were defined: Zero (no changes required); not available (additional investigation needed); Low (minor changes: no new components to be introduced); Medium (additional components have to be implemented); High (It is necessary to introduce huge changes to the existing components and, in case, adding new components as well); Very High (significant changes in the overall architecture have to be introduced).

**Table 1 - Template of table extracted from D5.3.2a**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| | | | | |
| **Evaluation of the updated architecture design** | | | | |
| | | | | |

Each table includes an additional part labelled "Evaluation of the updated architecture design" that will provide feedbacks about the improvements introduced in the architecture design update and about how they meet the required changes.

This section is divided in five sub-sections one for each of the following sub-attributes: interoperability, scalability, availability, performance and security. Each sub-section contains a set of tables: one for each scenario that required some changes in the architecture design at the end of the first evaluation process.

## 2.1. Interoperability

### 2.1.1. Scenarios evaluation

**Table 2 – Updated Architecture Evaluation with respect to scenario I.1.3**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| I.1.3 | SIP Client, Server 3PCC Service | User initiated SIP session transfer: E.g. two users are in a SIP managed video conference. User A decides to transfer his session from one mobile terminal to another. The target MT may not comply with OpVO policies, e.g. it may be a public display not suitable for a conversation between doctor and patient. | The required changes are unclear. It should be understood how user initiated changes to network connections can be controlled by OpVO policies. | Not Available |
| **Evaluation of the updated architecture design** | | | | |
| The new device will be cleared of policy via SLA before invoked. | | | | |

**Table 3 - Updated Architecture Evaluation with respect to scenario I.1.4**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| I.1.4 | Workflow Engine, SIP Broker 3PCC Service | The Workflow Engine starts a conference call between two or more OpVO participants using the SIP Broker 3PCC Service. | The SIP Broker 3PCC Service has to check if the conference call is compliant with the OpVO policies. Therefore it has to check the context of the participants of the conference. E.g. if the doctor is using a public display he should not be allowed to talk to a patient because conversations between patient and doctor should not be made public. | Low |
| **Evaluation of the updated architecture design** | | | | |
| The devices will be cleared of policy via SLA before invoked | | | | |

**Table 4 - Updated Architecture Evaluation with respect to scenario I.2.3**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| I.2.3 | Service A, EMS in Middleware B. | A hosting environment B, that does not run Akogrimo, asks through its EMS B the instantiation of a Service A in an Akogrimo based hosting environment A. Service A is instantiated by Akogrimo EMS A and the manageability information of this service should be available for the EMS B. | The architecture does not include components that can act as bridge between EMSs coming from different Grid middlewares.<br>It has to be provided a method for interchanging information between the EMS components belonging to different middlewares. | Low/Medium |
| **Evaluation of the updated architecture design** | | | | |
| This feature is not supported in a native way in the updated architecture design, but it could be implemented in the following way. The non-akogrimo EMS B that belongs to another service provider domain must be trusted by the Akogrimo EMS, and also some sort of SLA must be used for this functionality. For example, if a non Akogrimo service provider asks Akogrimo EMS to execute an akogrimo service, the Akogrimo EMS must make sure that the requestor is who he claims to be, not all requests should be accepted or else everyone will have access to Akogrimo. On the other hand, the non Akogrimo service provider will expect quality of service, so an SLA should be established and of course the non akogrimo client must pay for all of this. For summarizing:<br>1. The non akogrimo EMS must have a certificate from a Certification Authority (CA) that Akogrimo trusts.<br>2. It needs an SLA<br>In practice, that non Akogrimo EMS must perform all the steps that are needed in order to execute akogrimo services:<br>1. Contact SLA Negotiator and negotiate a contract.<br>2. Perform reservation using EMS.<br>3. Request the execution of the service he wants.<br>As stated before, the non Akogrimo EMS must have a certificate from the Akogrimo Certification Authority and this certificate will be presented during the interactions. Only if it is valid will EMS proceed to reservation and execution. In order for this feature to be supported the EMS negotiation phase must be changed, but these are minor additions/changes because the EMS design is very flexible and scalable. | | | |

**Table 5 - Updated Architecture Evaluation with respect to scenario I.2.4**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| I.2.4 | Service B, EMS in Akogrimo. | Akogrimo EMS asks for instantiating a service B in a hosting environment that does not run Akogrimo. | As the above, but in this case if the middleware B does not provide useful information for the managed execution of the service inside an Akogrimo OpVO, this instance cannot be used. | Low/Medium. |
| **Evaluation of the updated architecture design** | | | | |
| This feature depends on the rules and policies that the non Akogrimo hosting environment uses. As in the previous scenario, the non Akogrimo hosting environment should take the following actions:<br>1. Authenticate the Akogrimo EMS.<br>2. Give it an SLA.<br>3. Grant access to the services in the environment. | | | | |

**Table 6 - Updated Architecture Evaluation with respect to scenario I.3.2**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| I.3.2 | User, VO Mgr, A4C | User accesses the VO Mgr to use VO management services: VO Mgr requests authorization of the user to the A4C | None, as long as the VO Mgr is treated like every other service | None-low |
| **Evaluation of the updated architecture design** | | | | |
| In this case no changes are needed, the architecture design is the same. In principle, the mobile user is able to perform the required action (e.g. adding new VO member) because the VO manager and Participant Registry are Web Service and all methods to be invoked in order to add a new member are available as web method. Actually, it is not allowed to perform this action because just the administrator using dedicated tool is enabled to add new members. | | | | |

**Table 7 - Updated Architecture Evaluation with respect to scenario I.4.1**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| I.4.1 | MT, ECGData Analyzer, ECGData Visualizer, ECGData Generator, BVO Manager, UA, SA, A4C | MT->UA: Each communication towards the OpVO passes through a UA (a WS-Resource). Then external components (e.g. MT) have to invoke the UA to use the Akogrimo infrastructure features.<br>SA->ECGDA, ECGDV, ECGDG: At the same time, in order to invoke external components Akogrimo infrastructure assumes that these external components are WS- Resource or WS.<br>MT->A4C: Finally, external components need to communicate with the A4C subsystem of Akogrimo infrastructure. This is the only communication that can be established using both WS based interface and/or some client API provided by Akogrimo infrastructure | 1. Interface changes: in the Akogrimo infrastructure for each different application it is necessary to change the SA interface. It is foreseen a tool in order to automate the SA generation for each specific application<br>2. Application components: they need to be able to perform SOAP invocation and/or to include the A4C client API<br>3. Application logic: the application need to be have a service oriented logic and it has to be based on Web-service components orchestration | 1. low<br>2. low: depend on the application logic<br>3. potentially high if the application is not based on components |

## Evaluation of the updated architecture design

The architecture design has been updated to meet the required changes, in particular:

1. The factory service has been design that is in charge to create at runtime service agent service. This component receives as input the description of a web service interface and dynamically creates a Service Agent exposing the required interface. The Factory Service returns to the service requestor the Endpoint Reference Type identifying the created Service Agent. When SA are invoked they forward the request to EMS service belonging to SP domain providing the application.
2. The architecture design of Akogrimo infrastructure still needs to be invoked through SOAP message a part from the A4C component (also specific Java APIs are available to invoke it).
3. In the updated design the SA does not invoke directly services in the SP domain but each invocation passes through the EMS that will execute the job related to the specific required service. Summarizing the application does not need to be a Grid Service, anymore.

In general, we can conclude that the updated architecture provides new features to meet the required changes. In particular, for each numbered point:

1. The proposed solution is evaluated to fully address the recommended change
2. This is not a required change but rather the identification of a requirement fixed by the design (the same for the following item). This requirement is still valid but it is directly related to the choice of a SOA for Akogrimo then it was expected to continue to have this requirement
3. This requirement is not valid, yet. In fact, the use of EMS as front end of the SP domain allows having a greater flexibility in the application logic (each application component has to be an executable manageable by the EMS). At the same time, the new approach can arise some issues because the EMS can be a possible bottleneck. Some architectural solution should be introduced to solve this potential issue.

## 2.2. Scalability

### 2.2.1. Scenarios evaluation

**Table 8 - Updated Architecture Evaluation with respect to scenario S1.1**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| S.1.1 | Deployed service, Execution Management Services | The Execution Management Services have to be informed about the availability of the new machine. Once the new machine is available as new resource the EMS can perform workload management functions and transfer the existing service to the new machine. | None, if available in GT4. | Zero |
| **Evaluation of the updated architecture design** | | | | |
| D3.1.3 [2] section "4.7. EMS, Service Migration, Monitoring" describes the process of service migration. The new machine has to have the required service installed in a GT4 container. Service deployment itself is not handled by the EMS and is currently not supported by GT4. A machine with a deployed service can be advertised to the MDS and used as a replacement for an overloaded machine, thereby increasing the scalability. | | | | |

**Table 9 - Updated Architecture Evaluation with respect to scenario S.1.2**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| S.1.2 | Deployed service, Execution Management Services, Resource Management Services | The Resource Management Service should be able to deal with a hardware failure so that the EMS is informed of that the resource is no longer available and the services that were running on that machine have to be restarted on a different machine. | None, if available in GT4. | Zero |
| **Evaluation of the updated architecture design** | | | | |
| The described behaviour is supported by the Akogrimo EMS. D3.1.3 [2] section "4.7. EMS, Service Migration, Monitoring" describes the process of service migration. The described scenario is fully supported. The replacement machine has to have GT4 installed and the required service deployed. | | | | |

**Table 10 - Updated Architecture Evaluation with respect to scenario S.1.3**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| S.1.3 | Deployed service, Execution Management Services, Resource Management Services | S.1.3 is similar to S.1.1 except that the new machine differs in hardware features like memory, CPU speed, hard disk capacity from the existing machines. The steps executed are identical. The EMS should be able to deal with heterogeneous resources. | None, if available in GT4. | Zero |

| Evaluation of the updated architecture design |
|---|
| See S.1.1 – the difference in hardware is a parameter of the selection process of a replacement machine. This process is controlled by the EMS and supported by the MDS and the EMS Advertisement service and client. |

**Table 11 - Updated Architecture Evaluation with respect to scenario S.1.4**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| S.1.4 | Deployed service, Execution Management Services, Resource Management Services | S.1.4 is similar to S.1.2 except that the removed machine has special hardware features. Also here the EMS should be able to deal with heterogeneous resources. | None, if available in GT4.<br>The use of an information service should be introduced in the architecture | Low. |
| **Evaluation of the updated architecture design** | | | | |
| Analogue to S.1.2 this scenario is fully supported. The difference in hardware is a parameter of the selection process of a replacement machine/service. | | | | |

**Table 12 - Updated Architecture Evaluation with respect to scenario S.2.1**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| S.2.1 | EMS and deployment service. | The EMS detects that a machine could host a specific service. Then the EMS interacts with the deployment service on the specific host in order to deploy on the fly the executable necessary to publish the Web Service on the host. | The architecture description does not describe in detail the functionalities of the deployment subsystem and the sequence to have this deployment. Then, in principle, the scenario is supported by Akogrimo architecture but the way this functionality works to carry out a complete evaluation should be described in great detail.<br>Furthermore, it is not clear which kind of interface the EMS exposes to allow the deployment of a Web Service inside the Hosting Environment. | Medium if the lack of description reflects a shortcoming of the architecture. |
| **Evaluation of the updated architecture design** | | | | |
| The updated architecture design has been updated to meet the required changes, in particular EMS deploys a service in two different phases:<br>• the first is Negotiation and Discovery<br>• the second is Advanced Reservation.<br>In the first phase EMS is in charge of discovering the resources (inside the Service Provider domain) needed for the execution of the business services based on low level QoS parameters passed by the SLA-Negotiator service. The EMS discovers available resources through the use of an Index service that is included and deployed by default into the Java WS container of the GT4 as part of the WS MDS subsystem.<br>In the second phase the EMS performs advance service reservation on the discovered resources. Advance service resource reservation is achieved by creating an instance resource of the requested business service on the most suitable host and managing its lifecycle. The business services are developed on the WS-Resource Factory pattern that enables the management of multiple resources through the use of a factory service that creates instance resources of the service. Whenever the EMS wants to create a new business resource, it contacts the factory service that corresponds to the business service. The factory service returns to the EMS an endpoint reference (EPR) to the newly created business resource. In case no match | | | |

is found the EMS receives a null EPR

The above description provides some basic information about the interactions behind an invocation to the EMS requiring the availability of a service in the hosting environment. The updated architecture design of EMS subsystem is more detailed about this topic and provide enough information to understand how the system works. In particular, sequence diagram both for Negotiation and Advanced Reservation phase are available. Furthermore the used interfaces to allow deployment service are well defined.

**Table 13 - Updated Architecture Evaluation with respect to scenario S.2.2**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| S.2.2 | EMS and deployment. | The EMS detects that a machine cannot host a specific service anymore (The reasons that can bring to un-deploy a service are not specified in the architecture design).<br>Then the EMS interacts with the deployment service on the specific host in order to delete the executables available on the host. | As above (S.2.1) | As above (S.2.1) |

| Evaluation of the updated architecture design |
|---|
| After the deployment phase the EMS is in charge to manage and monitor the service execution. Once the execution of the business service has started, it is managed and monitored in order to achieve continuous conformance to the contractual terms of SLAs. In case of execution failure or failure to meet the SLA criteria and depending on the explicit type of failure, EMS is able to reallocate the execution. In order to detect possible failures and maintain the state of the execution between possible reallocations, a fault-detection in conjunction with a recovery scheme is being used. The EMS establishes a WS-Notification mechanism with the business services in order to receive notification messages every time a property of the business resource changes its value. Resource properties provide to the EMS a view on the current state of the resource. Every time the EMS receives a notification message it stores the new value of the resource property. If a failure occurs during the execution of the business service and the creation of new business resource is needed (either on the same or different location), the EMS will set the resource properties to the last known ones before the failure occurs. This is a first approach to a recovery scheme used by the EMS that makes use of the WS-Resource specification in order to maintain the state of the business resources. |

**Table 14 - Updated Architecture Evaluation with respect to scenario S.2.3**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| S.2.3 | EMS, deployment, monitoring. | The EMS may implement a checkpointing or replication scheme for fault tolerance interacting with the deployment service and with monitoring service to decide when to move the execution from a machine to another one. | The architecture clearly states the possibility to provide this functionality and identifies also the component that will provide it.<br>A detailed evaluation cannot be performed because an in depth description is not available. Anyway the architecture seems to provide all the components to provide these capabilities and the use of WS-Resource goes in the right way to manage status replication. | Low: it is necessary to design the deployment service. Anyway the other components are designed to interact with it |

| Evaluation of the updated architecture design |
|---|
| EMS architecture design meets the required changes by using two service (Monitor service and Metering service) together with the WS-Notification mechanism to monitor the execution of the business service. In this way EMS is always informed on the status of the service and, if a failure occurs, the EMS can perform all actions foreseen in the recovery mechanism, that is:<br>• To stop monitoring and metering on the business service;<br>• To destroy all resources related to the execution of the service;<br>• To create new business resource either on the same or different location.<br>As anticipated in the previous evaluation, the architecture design already provided statements about the availability of this feature. The updated design describes in more details the behaviour of the system in case of failure behind the EMS, in particular some choices are well appreciated (e.g. use of WS-Notification mechanism that should allow for interoperability with monitoring and metering services). A suggestion for future improvements could be to investigate how to monitor the single service instance instead of the machine hosting it. |

**Table 15 - Updated Architecture Evaluation with respect to scenario S.2.4**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| S.2.4 | GrSDS, "Service Requestor" | The involved components are the subcomponents of the Grid Service Discovery subsystem.<br>The GSDS will contact the LDS or the GrSDS depending on the kind of coming request. | The GrSDS can be a potential bottleneck that could limit the number of connection for searching. In general, the overall subsystem should be distributed in order to guarantee the scalability both for increasing the published service and the number of connection | Potentially low, depending on the difficult to manage the different replica |

| Evaluation of the updated architecture design |
|---|
| The updated architecture design shows that the service discovery server is divided into two parts: the service repository and the service discovery proxy. The service repository is principally replaceable, whereas the proxy stays the same.<br>The GrSDS is a directory service, that is, a kind of yellow pages; it can't be properly distributed as it should maintain the oneness of the VO. For this reason in the prototype there is only one instance of this service; in a real scenario the interface of the service could be made redundant with the same technologies used for the present web servers |

**Table 16 - Updated Architecture Evaluation with respect to scenario S.2.5**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| S.2.5 | Participant Registry | The involved component is the participant registry subsystem.<br>Each participant is a WS-Resource that allows to manage the role change. | Distribution of the service in order to manage the increasing number of participants | Low |

## Evaluation of the updated architecture design

The Participant Registry service is actually constituted by three services that provide different kind of information on the BVO/OpVO and their participants. These services are:

- Participant Info: this service will contain and manage all information related to participant features in the context of BVO/OpVO. This service will allow creating a WS-Resource associated to each participant and storing information related to the specific participant. In fact, it is able to manage information related to each different participant about:
- VO info: This service will manage information related to VO (BVO or OpVO) about its participants and its manager identifiers (name identity and endpoint reference). There will be a WS-Resource for each BVO/OpVO storing the mentioned information
- ParticipantNameService: This service will manage the mapping table in order to maintain an association between participant identifier and the endpoint reference of the resource.

The required changes have not been implemented. Anyway the architectural choices partially allow for supporting the distribution. In fact, binding a Participant Info WS-Resource for each participant allows for distributing this service on different machines and redirecting the resource creation through a load balancing mechanism. The implementation of this feature is not evaluated necessary for the scope of the prototype.

Suggestion for future work: investigating possible distribution mechanisms also for VO info service and ParticipantNameService.

**Table 17 - Updated Architecture Evaluation with respect to scenario S.3.1**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| S.3.1 | Service A, Network Access Point A, Number of Users. | A Number of Users try to access to a Service A through the Network Access Point A. | It has to be provided an alternative Network Access Point and a method for redirecting in case of overload. The architecture devolve to the EMS this role but it is not clear the behaviour and how it will managed | Not Available.<br>In principle, Medium. |

| Evaluation of the updated architecture design |
|---|
| The updated architecture does not foresee this case, there isn't any sort of backup plan. However, a normal PC (with speeds > 1GHz) should be well able to handle any traffic load. The real bottleneck is at radio signal level, not the Network Access Point level (Access Router). It will have to have overlapping wlans (for having handovers it will have to have some degree of overlap), so users from the overloaded Network Access Point A which are close enough to a second Network Access Point B can move to it. |

**Table 18 - Updated Architecture Evaluation with respect to scenario S.3.2**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| S.3.2 | Resource A, Number of Users. | A Number of Users try to access simultaneously to the Resource A. | It has to be provided a replica of the Resource and a method for redirecting in case of overload. The architecture devolve to the EMS this role but it is not clear the behaviour and how it will managed | Not Available.<br>In principle, Medium. |
| **Evaluation of the updated architecture design** | | | | |
| In the current architecture design is not foreseen that more than one user try to access to the same resource simultaneously. Each time a client performs a reservation, a resource for the service is created by the EMS. This resource belongs only to this client, by enforcing security and authorization only this client will be able to access this resource. If, for some reason, two requests arrive at the same time for the same resource, both of them will be accepted unless the operations of the service are synchronized. The results of the execution will be unpredictable. It is entirely up to the service provider to perform proper synchronization and the EMS is not responsible for this. | | | | |

## 2.3. Availability

### 2.3.1. Scenarios evaluation

**Table 19 Updated Architecture Evaluation with respect to scenario A.1.1**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| A.1.1 | The key service, The machine/device where the service is installed, EMS. | It has not been identified a clear description of architecture details that could manage this scenario. | To duplicate the services in other machines/devices and to implement a method for redirecting the execution in case of failure. It is not possible to provide further details. | Not Available. (Potentially high). |

**Evaluation of the updated architecture design**

The EMS has a recovery mechanism to deal with services' failures. This recovery mechanism consists of the following two actions:

1. To maintain the status of the services that are being executed. All Akogrimo services are grid services, meaning that they are stateful web services, and EMS maintains their status by keeping track of the values of their resource properties, so if and when a service fails, the EMS has its last status stored in disk storage.
2. When a service fails the EMS receives an exception. Depending on the type of the exception the EMS takes a recovery action. These are the possibilities:
    - the service crashes because of an internal fatal error, i.e. the service doesn't work like it should -> EMS reallocates the resource. This means that it discovers a new candidate service, creates a new resource, sets its status to the latest good one and repeats execution;
    - the service resource is lost because of a container restart, etc. The EMS will recreate the resource on the same machine and repeat execution.

**Table 20 - Updated Architecture Evaluation with respect to scenario A.1.2**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| A.1.2 | The key service, the network, the mobile terminal, EMS. | It has not been identified a clear description of architecture details that could manage this scenario. | To establish alternative network operators for connecting and to implement a method for connecting using this new operator in case of failure. It is not possible to provide further details, actually more investigations needed. | Not Available<br>(Potentially High if the architecture design doesn't provide these capabilities at all). |
| **Evaluation of the updated architecture design** | | | | |
| As the above scenario, the EMS has a recovery mechanism to deal with the unavailability of a service due to a network failure. This recovery mechanism consists of the following two actions:<br>1. To maintain the status of the services that are executed. All Akogrimo services are grid services, this means that they are stateful web services, and EMS maintains their status by keeping track of the values of their resource properties, so if and when a service fail, the EMS has its last status stored in disk storage.<br>2. When a service fails the EMS receives an exception. Then, the EMS will force another attempt on the same resource. If that fails too, EMS will reallocate it. | | | |

**Table 21 - Updated Architecture Evaluation with respect to scenario A.2.3**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| A.2.3 | New VO user that offers new services or extends services;<br>new/ extended Services;<br>Workflow Mgr. | New user enters the VO and adds new services or extensions to existing Workflows.<br>Workflow Manager is informed about the changed workflows.<br>Workflow Manager provides new Workflows, might change ongoing workflows, continues ongoing workflows that were suspended due to missing services now available. | For providing new workflows, none.<br>For continuing suspended workflows, none. These features are already provided.<br>For changing workflows, supposable none, as it should only influence the future of the workflow, not the past.<br>It is worth mentioning that the architecture design doesn't allow to choose new services when an OpVO is running. The services can be changed between a set identified in the phase of OpVO creation. | None – low<br>High if it is requested to check the availability of new services inside the BVO and changing the OpVO according with the new available services. |
| **Evaluation of the updated architecture design** | | | | |
| The Akogrimo architecture owns yet the required functionality to carry out the required features, although it is not explicitly described. The design includes the Service Agent, that allows the disengagement of the service reference used by the workflow from the real service instance used. A mechanism is described for replacing the running services in case of failure; the same mechanism could be used to introduce new services in the OpVO environment; a minimal architecture adjustment is required to notify the new services to the involved components and start the negotiation. The other involved component in this scenario are: OpVO Mng, OpVOBroker, Negotiator | | | | |

**Table 22 - Updated Architecture Evaluation with respect to scenario A.3.1**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| A.3.1 | It was not possible to identify a set of modules able to manage this scenario. Maybe the AR or QoS Broker are good candidate but the evaluation needs more in depth investigations | | | Not Available |
| **Evaluation of the updated architecture design** | | | | |
| In the updated architecture design we are using mobile IP and therefore as long as we have more than one line, if one of them fails we can set up the system to choose another one and make a transparent handover. If there was no other line, from the point of view of the system, it would be a matter of choosing another service/user in case it were replicable. This could be done either by having a reliable server pooling or by good WF coding that takes into account that these things could happen. | | | | |

**Table 23 - Updated Architecture Evaluation with respect to scenario A.3.2**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| A.3.2 | A4C, Participant Registry | The A4C server and the participant registry store at different levels the user/VO participant profiles. | A4C: for this component the architecture foresees a central database.<br>Participant Registry: this component stores information in a native XML database. Also in this case a replication mechanism description is missing.<br>A database replication should be introduced. This change can require different difficulties depending on the geographical distribution of these databases. The use of WS-Resource as access point to the database could simplify the work in the participant registry case. | Medium |
| **Evaluation of the updated architecture design** | | | | |
| The architecture design does not explain how to realize a mechanism for database replication. In general the design focuses on the different blocks of the architecture and how they interact without analysing how the single blocks can be replicated in order to address specific needs such as fault tolerance, mirroring, load distribution,…In any case the current design does not leave out the possibility to introduce them in order to address particular requirements. | | | | |

**Table 24 - Updated Architecture Evaluation with respect to scenario A.4.1**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| A.4.1 | All Akogrimo components | The usual interactions that the components require corresponding with the different possible events of Akogrimo | Include hardware and software monitoring tools:<br>• Network monitoring (SNMP software): In order to trace and supervise the network performance and obtain analytic data<br>• Logger monitoring: The Akogrimo software must contain logger objects to gather log info of all expected and unexpected events<br>• Component specific software tool: Use of specific monitoring tool (if there is any), when using a required tool for developing Akogrimo software (e.g, when using Axis to deploy some Web Service for Akogrimo, there's a helpful monitoring tool, called tcpmonitor, to trace the SOAP request and response Akogrimo messages) | Low/medium |
| **Evaluation of the updated architecture design** | | | | |
| D3.1.3 [2] section "4.7. EMS, Service Migration, Monitoring" describes the process of monitoring and failure recovery of an Akogrimo virtual hosting environment. Degradation of network QoS or loss of connection is reported by the QoS-Broker to the monitoring component of the hosting environment. Depending on the SLA for a specific service a counter measure might be initiated, a replacement service might be allocated or a SLA violation might be reported to the OpVO. Re-routing network traffic and load balancing in the network are assumed to be done by the network operator, but these topics are not specifically dealt with in the Akogrimo project. Higher level failure handling is done in the OpVO by application specific procedures that deal with SLA violations and replacing a failed service. Event logging is considered to be a component internal implementation detail. Logging accounting relevant events for auditing purposes is handled by the A4C infrastructure. | | | | |

**Table 25 - Updated Architecture Evaluation with respect to scenario A.5.1**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| A.5.1 | Old SIP Server, New SIP Server. | The requests for the old SIP Server will be redirecting to the new one. | It has to be included a method and/or a module for redirecting the requests from the old SIP Server to the new one. | Medium. |
| **Evaluation of the updated architecture design** | | | | |

The SIP Server is responsible for SIP message routing (if no Access Network (AN) SIP proxies available it acts also as a proxy), and includes a SIP Registrar Server and a SIP Presence Agent. It would be beneficial to have a mechanism to recover this information in case of failure, to enable the operation of an alternative SIP Server. If not, all registrations and publications will be lost and all MT should perform these processes again. If we want to make this failure transparent to the users, several options are envisaged:

1. Use of a distributed database, which could be read at the new server start-up, which could have the same IP address than the one which failed.
2. The primary SIP Server could re-route REGISTER, PUBLISH and SUBSCRIBE messages to the secondary SIP server to have it ready when required. In this case, the secondary server must have a different IP address, since both will be running simultaneously.

These mechanisms have not been implemented in the prototype, but will not modify the proposed architecture in a significant way. Having a non-collocated SIP PA with the SIP Server will minimize also the availability of this information in case of failure (in this case, publications and subscription will not have to be replicated). This can be done easily, by setting up the PA in another machine.

The existence of AN SIP proxies is highly beneficial in order to make the server's failure transparent to the MTs. AN SIP proxies are responsible to route the SIP messages from MTs to the corresponding SIP servers, typically by using the server's IP address. In case of SIP Server's failure, they have to be reconfigured only if the new one have a different IP address, while the MT will remain accessing the SIP infrastructures through the same proxy. MT SIP modules should have a list of the alternative proxies they must contact in case of failure (e. g. no replay from the proxy/server). This feature has not been implemented in the current prototype.

Finally, a SIP Broker failure and its substitution by a secondary one will imply:

- It will have no effect in further grid-triggered calls, but a grid-triggered transfer of an existing call could not be performed, since only previously grid-triggered calls can be further transferred. The storage and loading of this information when the secondary SIP Broker is started would be beneficial. The prototype does not implement it.
- - The service mobility feature (or grid service *sipification*) could be affected, since the sip gSDP sessions (between the MTs and the SIP Broker) used to exchange the nomadic/mobile service information will be lost. In this case, the EMS will not receive notifications about a change on the services availability. In order to make it possible, the new SIP Broker would have to setup the EMS interface and to setup again an SDP session with the terminals in which services are running. So the storage and loading of the SIP Broker "state" (which consist on a list of ongoing gSDP sessions) would be highly beneficial. Not implemented in the prototype.

## 2.4. Performance

### 2.4.1. Scenarios evaluation

**Table 26 - Updated Architecture Evaluation with respect to scenario P.1.3**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| P.1.3 | All Akogrimo Services | - | By using the Hardware Specification of P.1.2, the maximum number of users/request/responses as well as delay/latency and throughput has to be calculated | Medium-high |
| **Evaluation of the updated architecture design** | | | | |
| Some Akogrimo services have been tested (D532b) and the target of these tests was checking the performance while the service execution in terms of CPU, Memory, Disk, and other. These tests were executed on some critical components that could become bottleneck for the Akogrimo workflow execution; these components are: EMS, SLA subsystem (SLA-Decisor, SLA-Controller, Monitoring, Metering), SIP Server, network infrastructure (Access Routers, QoS-Broker, Mobile Terminal, Home Agent). The results of all performance tests, on average, accomplish the minimal expected results. | | | | |

**Table 27 - Updated Architecture Evaluation with respect to scenario P.3.1**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| P.3.1 | The components involved in Policy are not clearly defined and described, yet. | To be provided. | To define clearly the components involved in policy issues. | Not available. |

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| **Evaluation of the updated architecture design** | | | | |
| There are two modules involved in Policy, that is, two Policy Manager modules, each of one at a different layer, namely, Network Infrastructure Layer and Application Services Layer. | | | | |

**Table 28 - Updated Architecture Evaluation with respect to scenario P.4.1**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| P.4.1 | EMS; A4C server, SAML authority. | The EMS through the subcomponents (Advanced Reservation Service, monitoring, SLA controller) is able to foresee and check the status of processes (and related services). Through the management of reservation it will be possible to avoid reaching the upper limit. Anyway, the business service execution is always monitored with respect to the Service Level Agreement and, in case of violations, possible penalties will be applied.<br>The A4C server receives an authentication request and it contacts with the A4C storing the requestor credentials, if he/she doesn't belong to the administrative domain of the first A4C server. The A4C server will contact with the SAML authority to receive a token. | None in the EMS.<br>The A4C and SAML servers don't live behind an EMS then they should have ad hoc mechanism to manage the authentication request queue. It should be introduced a replication mechanism. | Low/Medium |
| **Evaluation of the updated architecture design** | | | | |
| The architecture design of the A4C and SAML server does not describe explicitly changes about the request queue. We have to notice that the SAML authority is not a standalone service, but it is integrated into the design of the A4C server. The A4C service does not seem act as bottleneck for the goal of the demo. Some tests can be executed to verify the real need of replication, and in positive case, implemented replicating the access interface and maintaining one single account database. | | | | |

**Table 29 - Updated Architecture Evaluation with respect to scenario A.5.1**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| P.5.1 | User, Call Agent, VO Mgr, Doctor | The user contacts the emergency call agent, which will contact the existing OpVO to manage the emergency, locating a doctor and initiating a direct connection between the doctor and the user.<br>Finally it informs the call agent about its success.<br>If this fails to be done in less than 15 seconds, the call agent will continue its automated call with the user. | Creating a VO, locating a doctor (or any other required service) and establishing a call must be faster then 15 seconds (all together, therefore about less then 5 seconds each) | VO, Service Discovery: Medium<br>SIP: low |
| **Evaluation of the updated architecture design** | | | | |
| The timer in the SIP Broker is configurable, so its value could be established for convenience. The only remaining feature is the implementation of a "cancel" operation for the doctor to know that the call failed. | | | | |

**Table 30 - Updated Architecture Evaluation with respect to scenario P.5.2**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| P.5.2 | A GRID Simulation Service | The simulation service is contacted for some task and replies with an estimated time for this job. | The service must be able to guess (pre-calculate) the time of a job. To be precise, the overall workload should be included in this calculation. | Depending on the service, medium – high |
| **Evaluation of the updated architecture design** | | | | |
| The tests made in the first phase of the project do not cover this aspect because a Grid Simulation Service was not available according with the test bed implementation plan. As this GRID Simulation service has been dropped from the final demonstrator (it was part of the eLearning scenario) this recommendation is not valid anymore. | | | | |

## 2.5. Security

**Table 31 - Updated Architecture Evaluation with respect to scenario SA.1.2**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| SA.1.2 | An invalid identified user, A4C Server. | An invalid identified user tries to authenticate in the A4C Server. | It has to be provided security methods for avoiding the false authentication and informing about the intrusion attack. | Low. |
| **Evaluation of the updated architecture design** | | | | |
| If an invalid identified user tries to authenticate in the A4C Server, this module will not provide the tokenID and the login application will not start. Each Akogrimo user has a unique ID and password to login into the platform. Then, the only way for an invalid user to authenticate correctly in the A4C server is to steal the ID and the password of a valid user, and this issue is not related to the platform itself. | | | | |

**Table 32 - Updated Architecture Evaluation with respect to scenario SA.2.2**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| SA.2.2 | An invalid identified user, BVO Manager. | The BVO Manager verifies the tokens provided by an invalid identified user. | It has to be provided security methods for the BVO Manager to reject the tokens provided by an invalid identified user and to inform about the intrusion attack. | Low. |
| **Evaluation of the updated architecture design** | | | | |
| The BVO Manager invokes the A4C Server in order to have the authentication of the incoming request and if the token is not valid it is rejected | | | | |

**Table 33 - Updated Architecture Evaluation with respect to scenario SI.1.2**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| SI.1.2 | MT, NAS, A4C and SAML authority. | The MT will contact the NAS (using EAP protocol encapsulated in PANA protocol) that will invoke the A4C server. The A4C server requires to the SAML authority an ID token for the current session. | None. To be tested with invalid invocation to check if they are rejected.<br>It is not clear in which way the SOAP message exchange (between SAML authority and A4C server) is secured between A4C and SAML server. | Not Available |
| **Evaluation of the updated architecture design** | | | | |
| The test between the involved components has been made in the first phase of the project. This test highlights that all invalid invocations are detected and rejected. The SAML Authority is an internal subcomponent of the A4C Server (D423), therefore the communication between A4C and SAML Authority is managed internally. | | | | |

**Table 34 - Updated Architecture Evaluation with respect to scenario SY.1.1**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| SY.1.1 | Reliable File Transfer service | Given two URLs the Reliable File Transfer service moves data between two locations. | Transfer should be secure, if transfer aborts, original state should be restored (do not overwrite files before fully transferred). | Low-medium |
| **Evaluation of the updated architecture design** | | | | |
| RFT has been changed with OGSA-DAI. Required changes have to be referred to OGSA-DAI, therefore has to be considered OGSA-DAI implementation. | | | | |

**Table 35 - Updated Architecture Evaluation with respect to scenario SY.1.2**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| SY.1.2 | File Transfer | A copy of a sending file is stored locally until this file reach its destination. | Keep a copy of data while sending is not acknowledged. | Low |
| **Evaluation of the updated architecture design** | | | | |
| See evaluation of SY.1.1. | | | | |

**Table 36 - Updated Architecture Evaluation with respect to scenario SY.1.3**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| SY.1.3 | Storage | A received file is duplicated and distributed immediately. | Use of a distributed FS, initiate duplicating features after file transfer. | Medium |
| **Evaluation of the updated architecture design** | | | | |
| DMS doesn't handle explicitly duplicating features. As we use OGSA-DAI we can use it as distributed FS. | | | | |

**Table 37 - Updated Architecture Evaluation with respect to scenario SY.1.4**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| SY.1.4 | Storage | All data is duplicated and distributed. | Use of a distributed FS. | Medium |
| **Evaluation of the updated architecture design** | | | | |
| See SY.1.3. | | | | |

**Table 38 - Updated Architecture Evaluation with respect to scenario SY.2.1**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| SY.2.1 | All components sending messages | All messages are signed. | Provide full PKI for all components in Akogrimo. | Medium-high |
| **Evaluation of the updated architecture design** | | | | |
| The architecture design foresees all messages can be signed using the security method based on PKI and use of certificates X.509 certificates with Akogrimo specific tokens and certificates (IDToken, VO Certificates). The details about the Akogrimo PKI have been provided in [6]. | | | | |

**Table 39 - Updated Architecture Evaluation with respect to scenario SY.2.2**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| SY.2.2 | Network connections | All communication channels are SSL. | Only allow SSL connections for sending and receiving messages. | Low |
| **Evaluation of the updated architecture design** | | | | |
| The update design includes this feature. | | | | |

**Table 40 - Updated Architecture Evaluation with respect to scenario SY.2.3**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| SY.2.3 | All components sending messages | The sent messages have acknowledge of receipt. | 1. Use TCP as transport protocol; 2. Implement ACKs | 1. none; 2. medium |
| **Evaluation of the updated architecture design** | | | | |
| The acknowledge of receipt is not implemented. When a message is not received properly, the module sending it receives an exception. If all goes well, the sending module receives a non error code. | | | | |

**Table 41 - Updated Architecture Evaluation with respect to scenario SY.2.5**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| SY.2.4 | Not identified | There are methods for resending messages. | 1. Use TCP and a packet cache (to not bother the application with resends); 2. Use TCP and let the application handle resends, or 3. Implement ACK/Retransmission. | 1. none, 2. low, 3. medium |
| **Evaluation of the updated architecture design** | | | | |
| GT4 does not foresee the resending of messages, it should be done programmatically. | | | | |

**Table 42 - Updated Architecture Evaluation with respect to scenario SY.2.4**

| cenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| SY.2.5 | Not identified | There are methods to request the sending of the message again | 1. Use TCP and a packet cache (to not bother the application with resends); 2. Use TCP and let the application handle resends, or 3. Implement ACK/Retransmission. | 1. none, 2. low, 3. medium |
| **Evaluation of the updated architecture design** | | | | |
| This feature is not foreseen. Moreover, the calls are linked and it would be complicated to implement. | | | | |

**Table 43 - Updated Architecture Evaluation with respect to scenario SY.3.1**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| SY.3.1 | Not identified | The malicious program attacks critical components | The possibility of this should be excluded in advance. Attacks must be recognized and tackled by deleting the malicious software | High |
| **Evaluation of the updated architecture design** | | | | |

If the malicious program is part of a regular Akogrimo component, e.g., a virus infiltrated entity, the Akogrimo architecture does not provide security mechanisms to avoid the attack, since this attack is defined to be out of the scope of the project.

If the malicious program is not part of the Akogrimo platform, but tries to act on behalf of an Akogrimo component or to pretend it is part of Akogrimo, several security mechanisms protect the platform against the attack depending on the domain where the attack takes place. The key concept in each domain (user, network, BVO, Service Provider) is based on the mechanisms provided by a PKI. Using X.509 certificates guarantees high reliability concerning the identity of the participants. These certificates used together with Akogrimo specific tokens and certificates (IDToken, VO Certificates) make sure that no malicious program is able to pretend a faked identity.

If the malicious program does not pretend to be an Akogrimo component, the attack is easily discovered, because an identity can be only validated when the entity is part of Akogrimo.

**Table 44 - Updated Architecture Evaluation with respect to scenario SY.3.2**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| SY.3.2 | Not identified | All the programs are replicated or distributed. | Services have to be recognized not by address, replicated data should be safe and only readable by the service it belongs to. | Medium |
| **Evaluation of the updated architecture design** | | | | |

Each Akogrimo service is addressed by its simplified SIP address (service@domain). This is not a security issue, since at the VO setup the service gets its VO certificate which contains among other information the service name (simplified SIP address). This VO certificate is signed by the VO manager, so it guarantees that only the certificate holder is allowed to assign the SIP address to his service.

Then the use of replicated data is only a security problem when these data can be used for pretending an identity or for obtaining knowledge about user's credentials, e.g. a password. This problem can be divided into three different areas of possible attacks:

An attacker can try to obtain user's login credentials. Using the credentials, he/she could make it possible to login to Akogrimo using the user's account. This should be avoided, but since it is not an Akogrimo specific problem, the usual security mechanisms are used (e.g., using an encrypted connection for submitting passwords) but the solution of the problem is declared as to be out of the scope of the project..

An attacker could use an Identity Token that he/she obtained by eavesdropping on a connection. Since the Identity Token is signed by the issuer and contains among other information a serial number, the validation entity will reject the token, because either the signature is invalid or the serial number is not correct.

The VO Certificate contains the simplified SIP address of the VO member. That means that even if an eavesdropper gets knowledge about the reusable VO Certificate, it will be useless, because he/she won't be able to bind his/her address to it without losing the signature validity. Additionally the VO Certificate is encrypted, so the attacker won't be able to use it for obtaining any specific knowledge.

Due to the requirements of a fast and reliable connection, only the crucial data are encrypted. Thus, if the encryption is not needed, every eavesdropper is able to follow the communication. The encrypted data can only be decrypted by the target receiver, because of the use of a reliable PKI.

**Table 45 - Updated Architecture Evaluation with respect to scenario SZ.1**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| SZ.1 | A4C Server, authenticated user at the A4C | The authenticated user invokes a non authorized service by the A4C (e.g. tries to access to a non authorized network, or makes a SIP call without a given permission). | Need further investigations. | Not available |
| **Evaluation of the updated architecture design** | | | | |
| If an authenticated user tries to invoke a service, which is not part of the Akogrimo platform, there is no way of avoiding it, because the Akogrimo Policy Management Systems components cannot check it. On the other hand, a service that is not part of Akogrimo will not ask for an access decision or a user profile at an Akogrimo entity.<br>If the user wants to use a service, which usage would be in conflict to other services, e.g. a non secured network for accessing a service that requires a secure connection, the Policy Management of the services should detect it and should refuse the access. | | | | |

**Table 46 - Updated Architecture Evaluation with respect to scenario SZ.2**

| Scenario Id | Involved components | Description of interactions among components | Required changes | Cost estimation |
|---|---|---|---|---|
| SZ.2 | VO Authorization Service, authenticated user at the A4C and at the VO, VO Service | The user invokes a VO Service that is not permitted based on his/her role. | Need further investigations. | Not available |
| **Evaluation of the updated architecture design** | | | | |
| Each authenticated A4C user is assigned to a specific role in the VO participants' registry. Since the access to a resource is protected by a policy that evaluates user's role, the user is not able to invoke the service; the Policy Management System of the service won't allow the access. | | | | |

# 3. Akogrimo attack models

Based on the Security Matrix and attack model defined in [2] the chosen security mechanisms are evaluated to show their impact on the overall security of the Akogrimo platform. Since many attacks were considered to be out of scope of Akogrimo or are very unlikely, this section focuses only on those attacks that have a high relevance within Akogrimo and also a high impact.

Attacker

Target

| | User | Terminal | OpVO | BaseVO | Customer Domain | Network Provider | Service Provider |
|---|---|---|---|---|---|---|---|
| **User** | 4/4/4 relevant but out of scope | 4/2/2 | 1/2/1 | 2/3/2 | 4/2/2 | 4/3/3 Not specific to Akogrimo | 1/1/1 Accounting; SLA |
| **Terminal** | 4/3/2 | 4/3/2 | 4/3/2 | 4/3/2 | 4/3/2 | 4/3/2 | 4/3/2 |
| **OpVO** | 1/1/1 | 1/1/1 | 1/2/1 | 1/3/1 | 1/2/1 | 4/4/4 | 1/3/1 |
| **BaseVO** | 1/1/1 | 1/1/1 | 1/2/1 | 4/4/4 | 1/2/1 | 2/3/2 | 1/2/1 |
| **Customer Domain** | 4/1/1 | 2/2/2 | 1/3/1 | 1/3/1 | 4/3/3 | 4/3/2 Not specific to Akogrimo | 1/3/1 |
| **Network Provider** | 4/3/3 | 4/2/2 | 2/3/2 | 2/3/2 | 4/4/4 | 4/3/2 Not specific to Akogrimo | 4/4/4 |
| **Service Provider** | 2/2/2 | 3/3/3 | 1/1/1 | 1/3/1 | 1/2/1 | 1/3/1 | 4/3/1 |

**Relevance in Ako/Likelihood in Ako/Impact**
1 high - 4 low

**Figure 2 - Security focus within Akogrimo**

The following section provides for each matrix's field marked in red an evaluation of the Akogrimo security infrastructure design with respect to the related attack model. This evaluation has the goal of providing:

- A more detailed description of the possible attacks behind a specific field of the matrix
- Initial feedbacks about the countermeasures of Akogrimo against this attack.

This is a preliminary evaluation because an attack model will be selected within the set analysed in the following sections for a more in depth analysis to be performed on the integrated demonstrator that will be released as final result of the project activities.

## 3.1. Security infrastructure evaluation

This subsection provides a table for each attack model. Below its template is reported.

**Table 47 – Table used for attack model description**

| Target | Attacker | Possible Attacks |
|---|---|---|
| | | |
| **Evaluation of the updated architecture design** | | |
| | | |

The rational of this table is explained here for a better readability of the document:

- Fields labelled with "Target" and "Attacker" are used to identify a specific attack into the matrix in Figure 2
- Field labelled with "Possible Attacks" provides a rough description about how the attacker could try to attack the target
- Field labelled with "Evaluation of the updated architecture design" provides information about:
  - The countermeasures (if any) that the Akogrimo security infrastructure can take in place in order to defend against the described attacks
  - The possible problems that could arise in relation with a specific kind of attack.

The following part of this section will report a table for each field marked in red of the security matrix (Figure 2).

**Table 48 – Attacks from OpVO to User**

| Target | Attacker | Possible Attacks |
|---|---|---|
| User | OpVO | The OpVO could return fake information to the user.<br>Someone could impersonate the OpVO. |
| **Evaluation of the updated architecture design** | | |
| The updated architecture provides means to avoid these attacks:<br>• SLAs are used to assure that information privacy policies are broken.<br>• The BVO monitors the actions of the OpVO, so illegal actions are limited.<br>• Authenticity of endpoints is assured via certificates, to avoid OpVO impersonation.<br>By means of these certificates, encrypted connections can be established.<br>The amount of information about the user that can be handled by the OpVO is limited. The OpVO will know that is running a job for User A, and the services running but it will not know in deeper detail is the identity of User A. | | |

**Table 49 – Attacks from SP to User**

| Target | Attacker | Possible Attacks |
|---|---|---|
| User | Service Provider | The Service Provider could breach the user's privacy.<br>The Service Provider could return fake/wrong results.<br>Someone could impersonate a Service |
| **Evaluation of the updated architecture design** | | |
| SLAs are defined to assure that the service results are the ones expected by the user.<br>The user does not usually interact directly with the service provider, but rather with the User Agent.<br>Certificates are used to assure authenticity of communication end-points, and to bootstrap encrypted connections. | | |

**Table 50 – Attacks from User to OpVO**

| Target | Attacker | Possible Attacks |
|---|---|---|
| OpVO | User | The User could attack the OpVO by providing incorrect data to services in the OpVO.<br>The User could breach the privacy of the OpVO information and leak it outside the domain.<br>Somebody could impersonate the user. |
| **Evaluation of the updated architecture design** | | |
| The updated architecture provides means to avoid these attacks:<br>1. The User is limited in its actions as the Base VO acts as a filter of information from the UA to the OpVO and applies policy on the requests the user makes.<br>2. The data sent between the user and the OpVO will be encrypted and identity managed by a certification mechanism.<br>3. SLAs are used to assure that information privacy policies are broken. | | |

**Table 51 – Attacks from Terminal to OpVO**

| Target | Attacker | Possible Attacks |
|---|---|---|
| OpVO | Terminal | The terminal could expose data from the OpVO to unwanted receivers.<br>A rouge terminal could influence context change and steal data.<br>Deliberately misleading data could be passed to the OpVO. |
| **Evaluation of the updated architecture design** | | |

| |
|---|
| SLAs are used to assure that information privacy policies are broken.<br>Policy is in place within the Base and OpVO's to ensure correct terminals are used in the case of context change.<br>The other services in the OpVO will pick this data up and stop the workflow with a error code. |

**Table 52 – Attacks from OpVO to OpVO**

| Target | Attacker | Possible Attacks |
|---|---|---|
| OpVO | OpVO | An OpVO could be attacked by another OpVO invoking services of the other one providing malicious data.<br>The only way an OpVO could influence another OpVO is by attacking the Base VO or service providers that another OpVO relies on. |
| **Evaluation of the updated architecture design** | | |
| OpVO's are distinct entities and each of them is provided and issues specific certificate. This implies that an OpVO can be invoked just from services belonging to the same OpVO<br>The knowledge of OpVO's is stored in BaseVO specific services to which other OpVO's don't have access.<br>The SLA management system in Akogrimo ensures that OpVO's can't request extravagant demands from service providers in order to protect other users. | | |

**Table 53 – Attacks from BaseVO to OpVO**

| Target | Attacker | Possible Attacks |
|---|---|---|
| OpVO | Base VO | The Base VO starts the Operative VO and has ultimate control over its destruction. Thus if the BaseVO's behaviour began to attack the OpVO it could do this by feeding in the wrong workflow data and destroying it before it has completed. |
| **Evaluation of the updated architecture design** | | |
| The protection against this will be the internal policies of the Base VO in relation to OpVO creation and destruction. | | |

**Table 54 – Attacks from Service Provider to OpVO**

| Target | Attacker | Possible Attacks |
|---|---|---|
| OpVO | Service Provider | Here the service provider can attack the OpVO by providing incorrect data and services when requested to do so.<br>It could also expose the data sent to services to parties outside of the agreement with the OpVO.<br>Possible man in the middle attacks can also be made by attackers to intercept data and pose as fake service providers. |
| **Evaluation of the updated architecture design** | | |
| SLA is a means by which these threats are initially addressed.<br>Sensitive data is broken up by using multiple service providers.<br>Quality can be checked by performing the same job (with a known result) on multiple services and checking the results.<br>The data sent between the Service Provider and the OpVO will be encrypted and identity managed by a certification mechanism. | | |

**Table 55 – Attacks from User to BaseVO**

| Target | Attacker | Possible Attacks |
|---|---|---|
| Base VO | User | The user (participant playing the role of user) is not allowed to use any feature of the BVO: the BVO Manager will not authorize any requests (e.g. publishing a service, creating an OpVO,…) coming from a participant providing credentials playing the role of user.<br>On the other hand, the user will access the BVO (or better an OpVO) through a UA then he/she will be able to attack the BVO if:<br>1. He/she will be able to provide stolen credentials and then acting on behalf of a participant having extended rights (e.g. a customer)<br>2. The user is able to leverage on an improper authentication/authorization handling by the BVO infrastructure |
| **Evaluation of the updated architecture design** | | |
| The Akogrimo design leverages on the concept of digital identities that are associated to a real identity in the home network domain. The digital identities are the ones valid in the BVOs. The architecture design includes the following countermeasures with respect to the above attacks:<br>1. the user cannot use a stolen digital identity because the home domain maintains the association real identity->digital identities and the BVO Manager always asks the Home domain A4C if the network authenticated requestor is associated to the digital identity he/she is trying to use. In order to succeed in this attempt the user should steal the credentials to access the network as well.<br>2. An improper handling could rise due to an improper behaviour of the following components:<br>a. The A4C authenticates a wrong couple of IDToken/digital identity.<br>b. The PR includes this digital identity with a profile that includes a specific role.<br>c. The PM includes the specified role associating the rights required to perform the actions asked by the requestor.<br>An appropriate testing of those components should guarantee from problems related to improper behaviour. | | |

**Table 56 – Attacks from Terminal to BaseVO**

| Target | Attacker | Possible Attacks |
|---|---|---|
| Base VO | Terminal | The terminal could attempt to perform administrative operations on the BaseVO or try to access third party information. |
| **Evaluation of the updated architecture design** | | |
| The terminal does not usually interact directly with the BaseVO and its operations are performed by the way of the UserAgent inside the OpVO.<br>Hence the BaseVO will reject any attempt of the terminal to execute operations since its don't own the credentials provided from the OpVO to only its members.<br>There is a phase in which the terminal interacts with the BaseVO: after the access into the network, it retrieves from the BaseVO the list of the available OpVOs and afterwards requires to connect to one of these.<br>This operation is a read-only operation and doesn't modify any information. The terminal could attempt to access information other than own authorized access. The BaseVO can protect itself from this attack by checking the identity exposed by the requestor with the identity authenticated by the network provider; the later feature is provided by the A4C server intercommunication.<br>Finally the attack could success by a misconfiguration of the A4C servers. | | |

**Table 57 – Attacks from OpVO to BaseVO**

| Target | Attacker | Possible Attacks |
|---|---|---|
| Base VO | OpVO | The BaseVO is a management entity in respect to the OpVOs and each OpVO, in normal operations, interacts with it accessing information from the Policy Manager and the Participant Registry. In an attack scenario, beyond the above operations, the OpVO could attempt to alter the information provided to other entities from the Policy Manager and the Participant Registry or attempt to request the BaseVO Manager operation for OpVOs other then itself. |
| **Evaluation of the updated architecture design** | | |
| All the service interfaces of the BaseVO are Web Services and they are protected by WS-Security specification; this should assure some common security issues (privacy, integrity, non-repudiation).<br>The other attack targets could be the Policy Manager and the Participant Registry that, as information providers of the whole VO, play a critical role; but:<br>• The information of the Policy Manager are not modifiable by any Akogrimo component, but only by a human operator by means of an admin client;<br>• The Participants Registry allow the update of the participant only to the BaseVO Manager, in the same domain, and deny the update to any other external entity; the domain membership can be verified by means of the IDToken.<br>The BaseVO Manager too, it can be target of attack in order to execute operation (e.g. the creation of a new unauthorized OpVO), but, even for this, the same security protection of the Participant Registry are valid.<br>The security protection above relies on the strength of architecture in the lower layers and it doesn't involve the component implementation.<br>Indeed, the protection of information provider services takes the base from access control of faked information insertion and, therefore, the access control to service operations; the later is not a direct issue of the service implementation. | | |

**Table 58 – Attacks from Customer Domain to BaseVO**

| Target | Attacker | Possible Attacks |
|---|---|---|
| Base VO | Customer Domain | The BaseVO Manager in the BaseVO domain contacts the A4C infrastructure for authentication and authorization purposes. Then, the customer domain could attack to the BaseVO if:<br>1. The customer domain provides deliberately wrong information in the authentication process.<br>2. The customer domain provides deliberately wrong information in the authorization process.<br>3. The customer domain provides wrong accounting information to the BaseVO. |
| **Evaluation of the updated architecture design** | | |
| See Table 55 – Attacks from User to BaseVO | | |

**Table 59 – Attacks from Customer to OpVO**

| Target | Attacker | Possible Attacks |
|---|---|---|
| OpVO | Customer Domain | See Table 58 – Attacks from Customer Domain to BaseVO |
| **Evaluation of the updated architecture design** | | |
| See Table 58 – Attacks from Customer Domain to BaseVO | | |

**Table 60 – Attacks from Service Provider to BaseVO**

| Target | Attacker | Possible Attacks |
|---|---|---|

| Target | Attacker | Possible Attacks |
|---|---|---|
| Base VO | Service Provider | The Service Provider could attack the BaseVO publishing deliberately services with incorrect data. |
| **Evaluation of the updated architecture design** | | |
| These incorrect data will result in failure of the negotiation phase or, if the service provider will continue to provide deliberately incorrect negotiation data, the SLA is a means by which these threats are initially addressed. | | |

**Table 61 – Attacks from OpVO to Customer Domain**

| Target | Attacker | Possible Attacks |
|---|---|---|
| Customer Domain | OpVO | Since the user agent, acting on behalf of the user, is involved in the OpVO, it has to present user's Identity Token to policy decision entities in the VO. The OpVO can try to modify the token, that means, the OpVO could try to tamper customer's identity<br>An OpVO is established to provide specific service-bundles to the consumer. If the OpVO's behaviour is malicious, the OpVO could provide wrong services – or no service at all |
| **Evaluation of the updated architecture design** | | |
| The OpVO cannot tamper consumer's identity, the Identity Token is secured and cannot be faked by the OpVO.<br>The OpVO is fully controlled by the Base VO. That means, an attack from the OpVO is seen as a violation of the trust agreement/contract between the Base VO and the Customer Domain. The punishment for violating the agreement should be high enough that a violation is not considered. | | |

**Table 62 – Attacks from BaseVO to Customer Domain**

| Target | Attacker | Possible Attacks |
|---|---|---|
| Customer Domain | Base VO | The BVO has knowledge about user's identity to provide individualized services. It could try to tamper customer's identity<br>The Base VO is responsible for creating the OpVOs, the user wants to have, and thus it could violate the agreement regarding OpVOs: It can create wrong OpVOs or even no OpVO at all. |
| **Evaluation of the updated architecture design** | | |
| The Base VO cannot tamper user's identity, since it is bind to a secured Identity Token which cannot be faked by the Base VO.<br>The proposed architecture requires trust between the Base VO and the Customer Domain, established, e.g., with signed contracts. If the punishment for violating the contract is high enough, such a violation will not be considered. | | |

**Table 63 – Attacks from Service Provider to Costumer Domain**

| Target | Attacker | Possible Attacks |
|---|---|---|
| Customer Domain | Service Provider | The Service Provider monitors all actions related to a specific customer and his services. Thus, it could send false/faked accounting data to the customer domain<br>The Service Provider could try to tamper customer's identity |
| **Evaluation of the updated architecture design** | | |

Since the Service Provider Domain is the only domain that handles accounting and monitoring of the use of services, it cannot be avoided that a Service Provider sends false data. But this risk can be softened, if only trusted A4C Clients are allowed to report monitoring data to A4C Servers. This trust, in turn, can be established by contracts between the domain owners.

The Service Provider cannot tamper consumer's identity, since it is bind to a secured Identity Token which cannot be faked by the Service Provider.

**Table 64 – Attacks from OpVO to Service Provider**

| Target | Attacker | Possible Attacks |
|---|---|---|
| Service Provider | OpVO | The OpVO could attack the Service Provider in the following ways:<br>1. Executing wrong worklows instead of the right ones associated to the services.<br>2. Creating incorrectly the Services Agents.<br>3. Providing stolen credentials and then acting on behalf of other OpVOs having extended rights<br>4. Leveraging on an improper authentication/authorization handling by the Service Provider domain |
| **Evaluation of the updated architecture design** | | |
| The SLA as result of the negotiation establishes which services can be invoked by the OpVO then invocations to different services from the ones defined by the SLA are rejected. A potential vulnerability could be the invocation of services using malicious parameters that could generate failure in the Service Provider domain.<br>In general, item 3 and 4 in the list above could arise from an improper behaviour of the GSI which the Akogrimo Service Provider security infrastructure is based on. | | |

**Table 65 – Attacks from Base VO to Service Provider**

| Target | Attacker | Possible Attacks |
|---|---|---|
| Service Provider | Base VO | The BaseVO could violate the service agreement of the VO or alter accounting information. |
| **Evaluation of the updated architecture design** | | |
| The BaseVO has a critical role in the VO management and it has a hierarchical priority over the Service Provider.<br>But in this case the Service Provider plays a role different from other Akogrimo components (see OpVO) and it, though entering in the VO, preserve own independency.<br>The Service Provider offers services to the VO, but doesn't grant the control of own structure to the VO. This is achieved with the application of policies provided by a local instance of the Policy Manager. If the BaseVO delivers policies conflicting with the local policies the result is a denial-of-service, but not a damage to the Service Provider.<br>Another attempt of attack could be the faking of the service accounting information. This can be easily verified as both the Service Provider and the BaseVO own an A4C server and malfunctioning or intentional corruption can be detected comparing the records.<br>So the security protection relies on the correct configuration of the Service Provider Policy Manager and A4C server. | | |

**Table 66 – Attacks from Customer Domain to Service Provider**

| Target | Attacker | Possible Attacks |
|---|---|---|
| Service Provider | Customer Domain | The customer domain could attack the Service Provider if:<br>1. Provides incorrect authentications/authorizations of users (the right ones are not permitted and the wrong ones are able to access).<br>2. Provides wrong users´ profiles.<br>3. Modifies the accounting records. |

| Evaluation of the updated architecture design |
|---|
| In this case, the architecture design foresees a flow of information that starts from the customer go through the VO and finally reach the Service Provider domain. The countermeasures in these cases are related to the ones involving the customer domain against the VO domain and the VO domain against the Service Provider domain (see related tables). |

**Table 67 – Attacks from NP to Service Provider**

| Target | Attacker | Possible Attacks |
|---|---|---|
| Service Provider | Network Provider | The network provider (through the SIP Broker) is responsible for offering the EMS information about services running on nomadic/mobile terminals, so a possible attack can consist on the provision of unrealistic information (incorrect services EPRs or wrong services availability). This wrong information can be introduced by the SIP Broker itself (the element which interfaces with the Service Provider), the SIP Server (which informs about the availability or not of a mobile device) or the MT (which is responsible to provide the real services information when it receives the corresponding request from the SIP Broker).<br>• Possible results of such kind of attack could be:<br>• Services running on mobile devices will not be able to be invoked (e.g. if a wrong EPR is provided).<br>• Services will not be invoked when available (if information about its availability is falsified). |
| **Evaluation of the updated architecture design** | | |
| As described earlier, this attack could involve the manipulation (or tampering) of platform infrastructures belonging to the core network, which, as a general rule, will not be very accessible to attackers. This fact minimizes the impact of this kind of attacks. Anyway, some kind of trusted relationship with the service provider entities would be also beneficial.<br>In the other hand, the authentication infrastructures build a barrier against a potential tampering of a MT, avoiding that a non-authorised entity can introduce wrong information on the platform; additionally, at the service provider side, if the EMS detects that something wrong is happening about the services running on a concrete machine, it can decide to find a more suitable (and secure) candidate. | | |

# 4. Additional requirements coming from demonstrator scenario

This section does have the goal of evaluating the Akogrimo architecture with respect to potential additional requirements to be met in order to run the demonstrator scenario on top of Akogrimo infrastructure.. At this purpose, first the elicited additional requirements will be described in section 4.1 and 4.2, finally, an evaluation is performed providing feedbacks that will explain how the architecture is able to carry out the scenario related to the requirements.

The demonstrator scenario has been defined integrating the three test beds of the project (see [10] for details):

1. eHealth testbed
2. eLearning testbed
3. Disaster Handling and Crisis Management (DHCM) testbed

The main focus of the demonstrator scenario will be on the DHCM test bed but additional capabilities coming from the other test beds were integrated in the scenario in order to have a comprehensive demonstration. (details about demonstrator scenario are available in [14] and in [15]).

The requirements are targeted at the infrastructure developers within activity 4. For each requirement we have identified target components and a priority based on the following classification.

1. Essential for successfully development of the final Akogrimo demonstrator
2. Would be required for production deployment but not critical for development within the lifetime of Akogrimo
3. Interesting feature but beyond the scope of Akogrimo

In section 4.3 an evaluation of the architecture is provided with respect such requirements and the resulting recommendations focus on the aspects that are considered essential for successfully deployment of the Akogrimo demonstrator according with the above classification.

## 4.1. Functional requirements

The functional requirements define the capabilities required by the Akogrimo infrastructure to support the final demonstrator application.

### 4.1.1. Planning Phase

**Table 68 – planning phase requirements**

| ID | Requirement | Target Component | Priority |
|---|---|---|---|
| F1 | *A Civil Contingency Officer (CCO) shall be able to define a group of organisations that can participate in emergency response operations.*<br>During the planning phase the CCO will locate different organisations and their resources within the local area that can be used during the response. These could include organisations | Base VO | 2 |

| ID | Requirement | Target Component | Priority |
|---|---|---|---|
| | such as the emergency services, schools, highways agency, media, etc. | | |
| F2 | *Organisations participating in a civil contingency should be able to publish services with metadata descriptions.*<br>The organisations participating in a CCO should be able to publish the resources that are willing to offer during a response. For example, most organisations will provide services that monitor status information about their resources or the environment (ambulance availability, deployed bronze responders, number of beds, surveillance video, etc). This information forms the basis for the common operational picture and is structured into information bundles that are delivered to different responders. | Service Discovery | 2 |
| F3 | *A Civil Contingency Officer shall be able to compose and publish workflow templates that represent civil contingency plans.*<br>A Civil Contingency Plans describe the tasks that need to be undertaken in response to an emergency and the resources required for those tasks. In Akogrimo, orchestration is described using abstract workflow descriptions and should be used to describe such plans | Workflow | 2 |
| F4 | *A Civil Contingency Officer shall be able to discover services provided by organisations participating in a civil contingency plan.* | Service Discovery | 2 |

### 4.1.2. Response Phase

**Table 69 – Response phase requirements**

| ID | Requirement | Target Component | Priority |
|---|---|---|---|
| F5 | *Bronze responders using a mobile terminal shall be able to participate in many civil contingency plans.*<br>For example, a police bronze responder may participate in the OpVO for assessing the emergency situation but may also be involved as a resource. | OpVO | 1 |
| F6 | *Silver command shall be able to instantiate different civil contingency plans based on analysis of the common operational picture.*<br>The nature of IEM is that multiple plans will be created to response to different threats and vulnerabilities. These plans should be able to be initiated during the operation of silver command. For example, silver command may decide to deploy a mobile medical centre and select the appropriate plan for doing so. Encoding the entire emergency response in one | OpVO | 1 |

| ID | Requirement | Target Component | Priority |
|---|---|---|---|
| | workflow is not possible. | | |
| F7 | *Silver command shall be able to dissolve a specific contingency plan and resources associated with its operation on completion of its objective.*<br>This completes the lifecycle of the OpVO. | OpVO | 2 |
| F8 | *Silver command shall be able to initiate planning for the recovery phase.*<br>During the response phase, the OpVO for recovery needs to be initiated allowing suitable resources to be put in place once there are no more life threatening situations. | OpVO | 1 |
| F9 | *Silver command shall be able to recruit new organisations and their services during the execution of a civil contingency plan.*<br>During the execution of a specific continency plan new resources may be discovered out-of-band from a previously unknown organisation. This has implications for the creation of the BaseVO and OpVO. | BaseVO, OpVO | 2 |
| F10 | *Silver command shall be able to view video surveillance resources.* | EMS, SIP Infrastructure | 1 |
| F11 | *Silver command shall be able to initiate communication with a human expert user.* | EMS, SIP Infrastructure | 1 |
| F12 | *Silver command shall be able to establish location context for deployed bronze responders.*<br>Location context for resources and vulnerabilities to determine associated risks, threats and resulting actions at Silver Command. | Context Manager | 1 |
| F13 | *Silver command shall be able to broadcast information bundles to bronze responders.*<br>The main challenge is here is how data is distributed between the different clients and services. The current e-Health prototype uses the workflow to orchestrate communication but in the DHCM scenario there will be ad hoc communication between bronze responders and silver command. | EMS, SIP Infrastructure | 1 |
| F14 | *Silver command shall be able to personalise information bundles based on context of the subscriber.* | Context Manager | 2 |
| F15 | *Bronze responders shall be able to subscribe to information bundles distributed by the common operational picture.*<br>See F13 | | 1 |

| ID | Requirement | Target Component | Priority |
|---|---|---|---|
| F16 | *Bronze responders shall be able to submit multimedia status reports to the common operational picture.*<br>See F13 | EMS, SIP Infrastructure | 1 |

### 4.1.3. Recovery Phase

The final demonstrator will not focus on the recovery phase because the Akogrimo infrastructure can provide a major support in the first two phases Therefore, at this stage the recovery phase remains under specified and will not contribute directly to system requirements on AC4.

## 4.2. Non functional requirements

The non-functional requirements define the operational constraints that must be applied when using the functional capabilities defined above.

### 4.2.1. Maintainability

**Table 70 – Maintainability requirements**

| ID | Requirement | Target Component | Priority |
|---|---|---|---|
| M1 | *The Akogrimo infrastructure must be easy to install, maintain, and administer.*<br>This is an essential requirement for application service development, demonstration, and exploitation. | All | 2 |
| M2 | *The Akogrimo infrastructure shall be modular in nature, with well defined interfaces between services to allow for future upgrade, re-writing of specific services.* | All | 2 |
| M3 | *The Akogrimo infrastructure shall be capable of recruiting new service providers, application services and service consumers, once properly authorized. This should be possible without major disruption, for example re-installation of infrastructure components.* | BaseVO, OpVO, Service Discovery | 2 |

### 4.2.2. Reliability

**Table 71 - Reliability requirements**

| ID | Requirement | Target Component | Priority |
|---|---|---|---|
| R1 | *Clients using a mobile terminal to access information* | EMS, SIP Infrastructure, User | 1 |

| ID | Requirement | Target Component | Priority |
|---|---|---|---|
| | *resources should be resilient to handovers between network access points.* | Agent, Home Agent (MobileIPv6) and Operating System features (MobileIPv6 and Fast handover) | |
| R2 | *Clients accessing services provided by mobile terminals should be resilient to handovers between network access points by those mobile terminals.* | EMS, SIP Infrastructure, Service Agent, Home Agent (MobileIPv6) and Operating System features (MobileIPv6 and Fast handover) | 1 |
| R3 | *Civil Contingency Officers should be able to encode redundancy paths into workflows describing civil contingency plans and the Akogrimo infrastructure should automatically invoke these recovery alternatives in the case of service failure.* | Workflow, EMS | 2 |
| R4 | *A list of explicitly defined faults and error codes must be created for Akogrimo infrastructure components.*<br>This will allow application developers and end-users to unambiguously specify which faults have occurred and the actions they need to take to recover. | All | 2 |

### 4.2.3. Security

**Table 72 - Security Requirements**

| ID | Requirement | Target Component | Priority |
|---|---|---|---|
| S1 | *The Akogrimo infrastructure shall support the full lifecycle for establishing and dissolving relationships between collaborating organisations.*<br>The key relationships between the participants in a civil contingency plan will be static. However, the planning phase cannot include all eventualities and there will be occasions when new resources need to be recruited during operational response. For example, silver command may be notified out-of-band about new information feeds being broadcast by media sources that provide vital information about the geography at risk and nature of a threat. Alternatively, there may be new threats that were not envisaged requiring new experts to contribute to the common operational picture. | BaseVO, OpVO, Sevice Discove-ry | 2 |
| S2 | *The Akogrimo infrastructure should allow access control policies to be* | Grid | 2 |

| ID | Requirement | Target Component | Priority |
|---|---|---|---|
| | *dynamically updated based on context information.*<br>The dynamic nature of DHCM response means that policies associated with information bundles will change based on the context of a bronze responder. The context directly impacts the risk associated with disclosing information. For example, if a bronze responder moves towards a perceived life-threatening situation then it may be essential to create new authorisations giving them permission to access to new information about the specific threat that was previously inaccessible to them.<br>This is similar to the e-Health scenario where a doctor may be able to view highly classified patient records within the location of a surgery but not while he/she is doing his or her weekly shop in a local supermarket. There will be situations when patient information may need to be delivered to a supermarket location, for example, if a paramedic is treating the patient following a terrorist attack at the supermarket. So, the access control policy associated with data should be dynamic and based on the situation. In this case, this could include the patient location, patient condition, healthcare professional role and other perceived threats in the location. | Security, Context Manager, Policy Manager, VO Manager | |
| S3 | *Akogrimo should comply with governance regulations for the processing of personal data.*<br>The DHCM applications deployed within the Akogrimo infrastructure will process personal data (location context generated by nomadic mobile services, patient records, etc). Such information systems operate within a strict regulatory framework that is enforced to ensure the protection of personal data against processing and outlines conditions and rules in which processing is allowed. Data privacy regulations allow subjects to self-determine how information is disclosed. There are many such regulations at European level and additional legislation implemented within member states. According to EU Directive 95/46/EC, if a healthcare provider maintains personal data on its patients the healthcare provider is identified as a data controller and is responsible for protecting that data against unauthorised use.<br>Typically, a healthcare provider implements the legislation by authoring a security policy that mandates working practices and security technology requirements (key sizes, algorithms). If a healthcare provider wants to access personal data within another organisation they are identified as a data processor. For the communication to occur between data controller and data processor consent must be obtained from the patient and a contract between the two parties must exist that defines conditions such as the type of data processing, data | A4C, Grid Security | 2 |

| ID | Requirement | Target Component | Priority |
|---|---|---|---|
| | anonymisation requirements and how long the data can be stored by the data processor. | | |
| S4 | *The Akogrimo infrastructure shall support authentication based on existing authentication infrastructures.*<br>Each organisation participating within the DHCM scenario will have existing authentication mechanisms for issuing and validating identity security tokens (Active directory, certification, smart cards, etc). These authentication mechanisms should not be replaced but integrated with identity mechanisms within Akogrimo. | A4C, Grid Security, SAML Authori-ty, PBNM | 2 |
| S5 | *The Akogrimo infrastructure shall support flexible policy-driven security requirements for the exchange of data information between parties across untrustworthy communication networks such as authentication, integrity and confidentiality.*<br>Each organisation will make decisions about the risk associated with data they control in respect to any information exchange. These risk need to be directly translated into security policies which define the authentication, integrity and confidentiality requirements for the resulting messages. These security requirements will change depending upon context.<br>The Akogrimo infrastructure should allow organisations to specify security policies and provide capabilities to determine message policies for specific information exchanges between two parties. | Policy Manager, A4C, Grid Security | 2 |
| S6 | *Both service consumer and service provider must instigate a formally declared logging procedure for security reasons. This will include access logging, and should allow security breaches to be detected and appropriate action taken.* | All | 2 |
| S7 | *Infrastructure and application services exposed on the Akogrimo network should meet appropriate Quality Assurance standards for safety critical systems.* | All | 2 |

### 4.2.4. Portability

**Table 73 - Portability requirements**

| ID | Requirement | Target Component | Priority |
|---|---|---|---|
| P1 | *Akogrimo infrastructure shall be capable of supporting application services deployed on LINUX, UNIX and Windows operating systems.*<br>The heterogeneity of information systems, operating systems and hardware platforms deployed within DHCM organizations | All | 3 |

| ID | Requirement | Target Component | Priority |
|---|---|---|---|
| | requires Akogrimo to be a portable infrastructure. | | |
| P2 | *Akogrimo infrastructure shall be capable of supporting mobile client applicationss deployed on LINUX, UNIX, Windows, other Mobile operating systems and terminal interface infrastructures.* | Mobile Terminal Services | 3 |

### 4.2.5. Performance

**Table 74 - Performance requirements**

| ID | Requirement | Target Component | Priority |
|---|---|---|---|
| PE1 | *The response time for the creation of the bronze emergency assessment OpVO must be completed within 5 seconds.*<br>The initial assessment workflow should be established very quickly and will typically involve only a few organisations and resources. This is a reference value provided by the DHCM application stakeholders. The same applies for all values mentioned hereafter in PE2, PE3, PE4. | All compoents in OpVO creation (UA, EMS. SAs, OpVO Broker, SLA Negotiator, SIP Broker, GrSDS, WF Engine) | 2 |
| PE2 | *The response time for the creation of the silver command OpVO must be completed within 20 second.*<br>The organisations and resources deployed by silver command will vary depending upon the emergency response required. However, the services required by silver command to establish the common operational picture should be accessible within 20 seconds even if all of the information sources have not come online | OpVO | 2 |
| PE3 | *The response time for application service invocation from mobile terminals should be within 5 seconds.* | EMS, SIP Infrastructure | 2 |
| PE4 | *The response time for notification messages from silver command to mobile clients should be within 5 seconds.* | EMS, SIP Infrastructure | 2 |
| PE5 | *A single OpVO must support at least 200 bronze responders communicating video using the Akogrimo infrastructure.*<br>Currently, Bristol City Council has a limited set of communication devices used by the bronze response team (about 20). The Akogrimo infrastructure should provide an order of magnitude improvement on current communication technologies. | SIP Infrastructure | 2 |

## 4.3. Assessment with respect to Akogrimo architecture

### 4.3.1. Evaluation with respect to functional requirements

The following table provides the evaluation of the Akogrimo architecture for each of the functional requirements listed in Table 68 and Table 69

**Table 75 - Architecture Evaluation with respect DHCM functional requirements**

| ID | Evaluation of required changes | Cost estimation |
|---|---|---|
| F1 | *A Civil Contingency Officer (CCO) shall be able to define a group of organisations that can participate in emergency response operations.*<br>This functional requirement involves the process of Base VO population registering new organizations that could be involved in a contingency plan. The VO management subsystem includes the Participant Registry component that has the role of maintaining information about the participants. | None or low if the data structure has to be changed to include new information |
| F2 | *Organisations participating in a civil contingency should be able to publish services with metadata descriptions.*<br>The function related to the publication of services from Service Providers registered into the BVO is already available through the Grid Service Discovery Service (GrSDS). The GrSDS is a sort of yellow page, since it stores static information. Information about the status of the single service or resource is available in the SP domain itself using the Grid Information Service (GIS) integrated from GT4. | None |
| F3 | *A Civil Contingency Officer shall be able to compose and publish workflow templates that represent civil contingency plans.*<br>Publishing workflow template is a key function in Akogrimo using a dedicated registry (Workflow Registry). The search process is not semantically enhanced then the infrastructure can not support a semantic search | None. |
| F4 | *A Civil Contingency Officer shall be able to discover services provided by organisations participating in a civil contingency plan.*<br>This requirement is complementary to F2 and it is met by GrSDS and GIS | None |
| F5 | *Bronze responders using a mobile terminal shall be able to participate in many civil contingency plans.*<br>Using a mobile terminal to access the OpVO and to provide services to be invoked by the OpVO is an available functionality | None if there is not interactions with the OpVO. Else potentially |

| ID | Evaluation of required changes | Cost estimation |
|---|---|---|
| | and has been already tested using the first prototype release. Participating in many civil contingency plans does not add complexity if the OpVOs underpinning these civil contingency plans do not have to interact among them as result of the interactions with the mobile terminal. | high. |
| F6 | *Silver command shall be able to instantiate different civil contingency plans based on analysis of the common operational picture.*<br>This requirement implies that several OpVOs have to be created and executed in parallel to manage a crisis. Creation of several OpVOs is possible and it has been already tested in simple cases. To be tested how the performance decreases when the number of OpVOs increases. The assumption that the OpVOs do not have to interact among them is still valid here. | None if there is no interactions with the OpVO. Else potentially high.<br>It is recommended to execute tests to evaluate potential performance problems. |
| F7 | *Silver command shall be able to dissolve a specific contingency plan and resources associated with its operation on completion of its objective.*<br>This functionality is available just at architectural level: to be implemented. | Low |
| F8 | *Silver command shall be able to initiate planning for the recovery phase.*<br>Putting in place suitable services to initiate an OpVO is a function already covered by the search and negotiation phase of the OpVO creation process. | None |
| F9 | *Silver command shall be able to recruit new organisations and their services during the execution of a civil contingency plan.*<br>The execution of a civil contingency plan implies the execution of an OpVO. The identification of service (and related service provider) to be bounded to the OpVO is performed when the OpVO is initiated. If a service fails and the SP is not able to replace it internally (the EMS provides functionalities to do that) the OpVO is not able to continue its execution.<br>The architecture design provides possible solutions but they are not detailed and they are not implemented. | High |
| F10 | *Silver command shall be able to view video surveillance resources.*<br>SIP Infrastructure integrates with EMS provides this capabilities. This functionality has been already tested. | None |
| F11 | *Silver command shall be able to initiate communication with a human expert user.* | None |

| ID | Evaluation of required changes | Cost estimation |
|---|---|---|
| | Infrastructure integrates with EMS provides this capabilities. This functionality has been already tested. | |
| F12 | *Silver command shall be able to establish location context for deployed bronze responders.*<br>The architecture and implementation provides this functionality through the Context manager.<br>Changes to be introduced if it is necessary a GPS based localization. | None.<br>Medium for GPS integration. |
| F13 | *Silver command shall be able to broadcast information bundles to bronze responders.*<br>This is critical functional requirement because the current architecture does not support it. In fact this requirement implies an unstructured communication between parties involved in a contingency plan and then in an OpVO. Instead an OpVO is structured in the sense that everything is controlled through a workflow. In order to meet this requirement it is necessary to envisage an unstructured OpVO that allows communication between User Agent (UA) and Service Agent (SA) without passing through a workflow. At the same time it is necessary to continue to have an OpVO in order to be sure that the communication (among the involved parties) is still controlled and there is not a completely loose of control management (e.g. through the EMS). To achieve this goal the OpVO creation process has to be modified in order to have as result a more general OpVO that allows also for direct communication among UAs and SAs. | Medium. There are several services to be modified even if the cost of each of them is low. |
| F14 | *Silver command shall be able to personalise information bundles based on context of the subscriber.*<br>See F13 | |
| F15 | *Bronze responders shall be able to subscribe to information bundles distributed by the common operational picture.*<br>See F13 | |
| F16 | *Bronze responders shall be able to submit multimedia status reports to the common operational picture.*<br>See F13 | |

### 4.3.2. Evaluation with respect to non functional requirements

The following table provides the evaluation of the Akogrimo architecture for each of the non functional requirements listed from Table 70 to Table 74

**Table 76 - Architecture Evaluation with respect to the demonstrator scenario non functional requirements**

| ID | Evaluation of required changes | Cost estimation |
|---|---|---|
| M1 | The Akogrimo infrastructure is a complex system that requires several competencies to be installed, maintained and administered. All the components of this infrastructure include installation packages that can be easily used following the associated guides. The overall maintenance and administration should be simplified through the design and implementation of an integrated administrative interface. | Medium |
| M2 | The Akogrimo architecture was designed following the principles of SOA and then it is modular and open to future upgrades | None |
| M3 | In general the recruitment of new service providers, consumers and services requires an update of environment configuration. Nothing needs to be re-installed. Rather installation of dedicated components in the Service Provider or Consumer environment need to done to interact with Akogrimo infrastructure | None |
| R1 | Akogrimo infrastructure supports handovers between network access points. | None |
| R2 | See R1 | None |
| R3 | The Business Process enactor subsystem (part of Akogrimo architecture) is based on BPEL and allow for designing redundancy paths. Furthermore the Akogrimo architecture provides capabilities to execute these redundancy paths as result of event notified (e.g. context changes) at run time.<br>Furthermore the EMS subsystem supports management of service failures inside the Service provider domain.<br>The architecture design support more complex failure through and management (e.g. fatal failures in the Service Provider domain can be thrown to the VO level and managed at this level) but the current infrastructure does not implement these capabilities. | Medium: in case the advanced failure management should be managed. |

| ID | Evaluation of required changes | Cost estimation |
|---|---|---|
| R4 | The components of Akogrimo create log files and information about faults, but a comprehensive fault logging system is not available. To examine and understand current fault codes requires a low level understanding of the implementation. | Medium: such a logging system should be completely designed and implemented, but it is out of the scope of the Akogrimo prototype. |
| S1 | See evaluation of functional requirement F7 in chapter 4.3.1 | |
| S2 | The access authorization in Akogrimo follows a role based approach then for each role different access policies are defined. As long as the role and associated access policies are pre-defined it is possible to modify the role associated to an identity.<br>At the moment, the implementation does not support fully dynamic changes of roles but it is done through an administrative GUI. To have a fully dynamic update, the VO manager should receive notifications and modify the identity roles accordingly. The most of the interfaces to do that already exist (e.g. to modify the role in the PR, to identify several type of context changes), then the overall design is not affected, just an additional capability should be added to the VO Manager. | Low |
| S3 | Akogrimo provides a set of primitives to manage secure access but to meet this requirement additional features should be defined on top of them.<br>The current design does not meet this requirement. | High |
| S4 | The communication between Akogrimo infrastructure and organizations providing services uses standards message and transport levels security mechanism. On the other side, messages are signed using tokens issued by Akogrimo infrastructure then even if the organizations support such mechanism it is necessary to study in each specific case how to map credentials in order to perform authentication with existing home infrastructure. | Depends on the existing home infrastructure. |

| ID | Evaluation of required changes | Cost estimation |
|---|---|---|
| S5 | The Akogrimo infrastructure provides a policy manager that is not related to specific typology of policies. It allows to store and manage policies that can be defined for different purposes (e.g., security, SLA violation).<br>Of course for each defined policies it is necessary to design and implement the related policy decision making. The design does not support the specific case but it is open to be extended at this purpose. | Depends on the difficulty to design and implement the policy decision making. |
| P1 | Akogrimo architecture is based on the service paradigm then the architecture is portable on different platform.<br>The implemented prototype includes services developed on *LINUX* and *WINDOWS* platform, but duplicated implementation does not exist for the components of the infrastructure. Then all the features of the infrastructure are not available for services running on all platforms. This is a limit of the prototype. | Medium: to meet this requirement requires a high implementation effort (though the design does not have to be modified). It is out of the prototype scope. |
| P2 | See P1. | High: in the particular case of the mobile terminal some features are not available on all OSs. |
| PE1-PE5 | In order to understand at which extension these requirements are met, it is necessary to define a quantitative evaluation based on tests to be performed on the final prototype. | |

### 4.3.3. Recommendations

After evaluating the architecture with respect to the demonstrator requirements, some key aspects have been identified bringing to some updates to be introduced and tests to be performed. Though the evaluation above does have a general purpose and aims at providing feedbacks for improvement of Akogrimo results after the end of the project, the following recommendations just focus on aspects that are necessary to run successfully the demonstrator on top of the Akogrimo infrastructure.

**REC01.** The Akogrimo architecture design provides most of the critical functionalities to run the demonstrator but in the evaluation reported in section 4.3.1, it is clear that an important gap exists with respect to the requirements from F13 to F16. This gap needs to be covered in order to enable the execution of the demonstrator scenario.

**REC02.** To have a rough evaluation of the behaviour of the infrastructure in case of faults it is suggested to define a couple of tests covering the following fault cases (these tests do not have to introduce any change to the current implementation):

1. How the EMS manages a crash in the Service Provider domain
2. To design alternative paths in the workflow that execute alternative simple tasks (e.g. stop the workflow) in case a fault is thrown to the Workflow engine.

**REC03.** To have a rough evaluation of the authentication and authorization mechanism it is suggest to define a couple of tests covering the following aspects (these tests do not have to introduce any change to the planned infrastructure implementation):

1. Authentication and authorization of a user subscribed in the VO following the overall process that starts from the network login to the VO services invocations
2. To modify the role associated to a user to check if the authorization is denied after this change

**REC04.** To define tests in order to verify the behaviour in case of handover between network access points.

**REC05.** In order to evaluate the performance of the system it is necessary to define a set of tests related to:

1. Response time to create an OpVO (this test should be done considering different definitions of OpVO)
2. Response time for application service invocations
3. Response time for message notification
4. Scalability of the OpVO in terms of number of participants

The threshold values provided in Table 74 have to be considered indicative, the results of the suggested tests have to be evaluated to understand the existing gap to meet the required performance and to evaluate the effort necessary to cover that gap.

# 5. Architecture Gap Analysis

The goal of this section is to identify areas among the initial Akogrimo objectives that have not been addressed by the final design. For this purpose a comparison analysis is performed between the features of a hypothetical target architecture that should cover all those areas and the features included in the final architecture design. In this way it is possible to identify which features, respect to the target architecture are missing.

This section is organized as follows:

1. Subsection 5.1 summarizes the Akogrimo initial objectives as introduced in the Description of Work (see [12])
2. Subsection 5.2 describes the identified features that are supposed to be provided in order to meet the identified goals.
3. Subsection 5.3 provides information about the gap between the final architecture and the target one using a matrix representation.

## 5.1. Akogrimo Objectives

According to the Akogrimo Annex I (see [12]), "*the global objectives and visions of Akogrimo are centred on the notion of the 'Next Generation Grid' (NGG)*". Annex I also underlines that a formal definition of NGG is not available then it is necessary to refers to description provided by experts: "*a result of an evolution process comprising three layers – namely the NGG applications, the NGG application enabling technologies and the NGG core layer. The set of envisioned NGG applications, in addition to 'heroic computing', now includes e-business, e-health, e-government and e-learning. In the 'NGG enabling layer' – a comprehensive set of existing and evolving technologies such as Web services, semantic Web, data mining technologies etc. are mentioned. The 'infrastructure layer' completes the envisioned NGG framework by the provision of a service portfolio according to the OGSA vision and beyond.*" (quoted from the Annex I).

In the same section of the Annex, it is also emphasized how Akogrimo considers the mobility paradigm a key aspect of NGGs: "*A blueprint and instance of a/the Next Generation Grid – embracing the knowledge and mobility paradigm*".

Annex I provides also the requirements from a technical viewpoint and they are summarized in the following bullet list (for details see [12]):

- NGGs have to be based on, and vertically co-operate with, mobility enabled IPv6 infrastructures, network related middleware such as AAA and QoS, Mobility Management systems.
- Personalization, security, privacy and trust will be based on innovative identity management.
- Resource sharing – one of the key concepts of the Grid – needs to become mobilized, virtual and dynamic - leading to the 'Mobile Dynamic Virtual Organization' (MDVO).
- Expand the potential of the Grid adding context awareness
- Supporting the service provision process with an accounting model that takes into account and solves the issues coming from the multi domain and dynamic environment of a MDVO.
- Supporting definition of contracts based on Service Level Agreements and control of compliance during the operation phase.
- Allowing dynamic adaptation of applications on changing situations. It implies a vertical integration because it is necessary to adapt the services throughout the various layers of Akogrimo platform.

## 5.2. NGG embracing the mobility paradigm

In order to identify the key features of a NGG embracing the mobility paradigm the following approach has been followed due to the lack of a "standard definition".

As first step the "classic" NGG[1] vision has been revised an enriched with additional challenges. Looking at the quoted text in section 5.1, in the worldwide research community there is a general agreement about the central role played by the Open Grid Service Architecture (OGSA, see [13]) as reference model for the envisaged NGG framework.

The OGSA describes a set of capabilities largely independent each other so that a Grid systems[2] can implement a subset of them depending on its purpose. Below Table 77 provides just a summary of these capabilities, for details refer to the OGSA v1.5 document (see [13]):

**Table 77 – Capabilities required by the OGSA model**

| Capability | Description | Subcomponents |
|---|---|---|
| Infrastructure services | They are not services per se but assumptions related to the infrastructure on which all the other services should be built. They mainly deal with standards/specifications and properties to be provided by each OGSA service | **Web Service Foundations**: the overall architecture is assumed to follow the SOA principles and the Web Service technologies are assumed as basilar infrastructure |
| | | **Naming**: OGSA uses a three level convention for naming schema based on the WS-Addressing Endpoint Reference (EPR) model. |
| | | **Security**: OGSA assumes the use of WS-Security standard protocols to manage authentication, authorization, and message protection. Higher-level protection mechanisms are strongly suggested if required by the scenario (e.g. based on IPsec or TLS) |
| | | **Representing state**: services have to be statefull and the WSRF or WS-Management specification families are assumed as reference. |
| | | **Notification**: notification mechanisms based on the publish and subscribe pattern are assumed to be used |
| | | **Transactions**: "*for the purposes of the OGSA architecture, we assert that transactions layer on top of the basic architecture in a transparent fashion*" |

[1] Not including the mobility paradigm.

[2] Here the term Grid system is used instead of NGG because that is the terminology applied in the referred OGSA document.

| Capability | Description | Subcomponents |
|---|---|---|
| | | **Orchestration:** the composition of existing services is a key aspect in OGSA. Existing mechanisms are suggested for providing this capability (e.g. choreography, orchestration, workflow) |
| | | **Interoperability**: ensuring interoperability is a must. At this aim OGSA leverage as much as possible on existing WS-specification standards |
| Execution Management services | EMS addresses problems regarding the execution of units of work, including their placement, provisioning, and lifetime management. The EMS capability includes:<br>• Finding execution candidate locations.<br>• Selecting execution location.<br>• Preparing for execution.<br>• Initiating the execution.<br>• Managing the execution | **Service container**: it is the entity that contains running entities (e.g. a Web Service) |
| | | **Job manager**: it is is a higher-level service that encapsulates all of the aspects of executing a job, or a set of jobs, from start to finish. |
| | | **Resource selection service**: it can be a set of services that interacting among them decide where executing a job. |
| Data Services | Data services should provide the following set of functionalities: data transfer, storage management, simple access, queries, federation, location management, update, transformation, security mapping extension, resource and service configuration, metadata catalogues, data discovery, provenance.<br>OGSA does not impose designing/implementing all of them but just the subset that is required by the specific requirements. | The basic structure of data services includes services for:<br>• Storage management<br>• Data management<br>• Transfer<br>• Registries<br>This list can be extended. |
| Resource Management Services | These services serve for managing in several forms the resources available through the Grid. | **Manageability interfaces**: provides functions for introspection and retrieve manageability information |
| | | **Manageability functionalities**: they are implemented from management services that are able to understand manageability information retrieved through the manageability interfaces. |
| Security Services | OGSA requires a security architecture that supports, integrates, and unify widely used | **Credential Validation and Trust services**: guarantee verification of asserted identity. |

| Capability | Description | Subcomponents |
|---|---|---|
| | security models, mechanism, protocols, platforms, and technologies. These existing security solutions have to be improved in order to meet the classic property of a Grid-specific application: it can span through multiple administrative domains. The OGSA model identifies a set of capabilities that should provided by a set of related services (at least partially). | **Trust, Attribute and Bridge/Translation services** allows for:<br>• mapping identities between different domains<br>• conversions between different type of credentials. |
| | | **Authorization service**: it is in charge of taking policy-based access control decision |
| | | **Audit service**: produces records tracking security relevant events. |
| Self-Management Services | The work on this topic is still at a preliminary stage (though it is considered really relevant). The current version of OGSA mainly focuses on the service level attainment and it does not focus on describing the components but the mechanisms by which it is done. | **Self configuring mechanism**: they allow to apply policies in order to modify the system configuration (e.g. deployment of new services). |
| | | **Self healing mechanisms**: they allow to detect hostile behaviour in order to take corrective actions that can safe the overall system. |
| | | **Self optimizing mechanisms**: they deal with enforcing an SLA and meeting it optimizing the available resources |
| Information services | Information services provide data to be consumed by interested parties. The term information is related both to dynamic (e.g. consumed by monitoring services) and to static data (e.g. published for discovery purpose). OGSA information services follow the publish though consumption information model. OGSA recommends for adopting multiple information services able to manage general and use cases specific information. | **Discovery service**: it can be a registry providing information about services and resources |
| | | **Message infrastructure**: it is a common system allowing message delivery to the interested parties. |
| | | **Logging service**: optimizes interaction between logs producer and consumers |
| | | **Monitoring service**: it is a special purpose service providing information labelled with field for ordering purpose. In some cases (e.g. real time monitoring) it could meet strict requirements |
| | | **General information and monitoring service:** it is a general purpose service that should embrace all the above components. |

After having identified the basilar set of capabilities that should be addressed by a NGG architecture, additional capabilities will be provided hereafter in order to explicitly embrace the mobility paradigm as well.

**Table 78 – Additional requirements for capabilities required by the OGSA model**

| Capability | Additional Requirements | |
|---|---|---|
| Infrastructure services | **Mobile IPv6** | Mobility: Mobile IPv6 is used to provide transparent terminal mobility |
| | | Network transparency:. A user will not have reduced functionality when he is not using a different network |
| | | Roamability: Using MIPv6, it is possible for a user to switch between different networks without losing the connection |
| | | Non-IP protocol support: Akogrimo, has an all-IPv6 approach, there is no support to non-IP protocols |
| | **SIP Infrastructure** | Signalling: The SIP event notification framework is used to provide and achieve the status of SIP resources over the network |
| | | Hand-offability: SIP-aware applications can transfer their sessions from one machine to another |
| | | Signalling: SIP protocol is used to establish session in an IP network |
| Execution Management services | Mobile services management | |
| | Negotiation: To provide a way to stablish a SLA contract between the service execution requestor and the Service Provider. The SLA contract might include precise conditions of the service execution that should be fulfilled at service runtime | |
| | Monitoring: To provide a way to verify the fulfilment of the SLA contract conditions during the service execution, and inform in case of service violations. | |
| Resource Management Services | Context awareness | |
| | Signalling: The SIP event notification framework is also used to provide and achieve the status of SIP resources over the network – SIP presence as a source of context information | |
| Security Services | Integration between network and service level security mechanism | |
| Self-Management Services | Adaptive business process | |
| | SLA adaptation also on the basis of QoS | |
| Information services | Context information | |
| | Accounting across multiple domains | |
| | Signalling: the SIP Protocol will be used to retrieve information (e.g. the Service EPR, availability/unavailability, …) about grid services running on nomadic or mobile devices | |

| | |
|---|---|
| VO management (new capability) | VO Creation: involves all the actions that need to set up a VO and then the interactions among all the parties involved in a VO |
| | VO Operation: it is related to the features that need for providing all the functionalities of the VO expected by the customer |
| | VO Evolution: it is related to the capabilities to manage the lifecycle of the VO from its creation to its dissolution, including corrective actions taken to react to events arising during the VO operation |
| | VO Monitoring: it is related to features to monitor the execution of the VO as a whole and to take corrective actions in order to react with respect to unforeseen events (e.g. failures) |
| | VO Dissolution: it is related to the features for a smooth dissolution of the VO. |

## 5.3. Comparative Analysis results

In this section, a comparison between the features identified in the previous section 5.2 and the capabilities provided by the Akogrimo architecture is provided.

This analysis was performed referring to the architecture designed in the different Akogrimo WP4.x:

- WP4.1: Mobile Network Architecture, Design & Implementation
- WP4.2: Mobile Network Middleware Architecture, Design & Implementation
- WP4.3: Grid Infrastructure Services Layer
- WP4.4: Grid Application Support Services Layer

The following subsections identify the capabilities met in each WP4.x providing an evaluation about the gap that exists in each WP4.x with respect to the final goal to be achieved by Akogrimo.

### 5.3.1. WP4.4 analysis

WP4.4 deals with the Akogrimo infrastructure related to the management of VOs and, more in general, with features that represent the front end of Akogrimo with respect to the applications layer.

The WP4.4 architecture includes 5 main building blocks:

- VO Management
- Business Process Enactment
- SLA High Level
- Security infrastructure
- Grid Service Discovery Service

Each of them covers some of the capabilities listed in Table 72 and 73.

More in detail:

- The Infrastructures Services represent capabilities that are met across all the building blocks

- The Execution Management Services are partially met by the VO management building block.
- Data services are not part of WP4.4 (see WP4.3)
- Resource Managements service are not covered in WP4.4
- Security services are partially covered by WP4.4
- Self Management services are partially covered by WP4.4
- Information services are partially covered by WP4.4
- VO management services are partially covered by WP4.4

#### 5.3.1.1. Infrastructure services

"They are not services per se but assumptions related to the infrastructure on which all the other services should be built. They mainly deal with standards/specifications and properties to be provided by each OGSA service" quoted from Table 77. As a consequence the infrastructure services list is not expected to be covered by a specific building block but rather the building block should be built on them. That is the rationale behind the WP4.4 design as well, in particular the infrastructure services are not designed in the frame of the WP but existing solutions are used providing such capabilities (e.g. WSRF.NET, GT4 core,…).

The WP4.4 includes application support level services: all the building blocks have been designed following the Service Oriented model and in particular the Web Service reference model.

On the basis of above assumptions all the WP4.4 building blocks are built on a subset of the infrastructure services: Web Service Foundations, Naming, Representing State, Interoperability.

In some cases (notification, orchestration, security) the infrastructure services are related just to specific building blocks because they are the only blocks that use a specific feature (e.g. notification) or provide a specific feature to the Akogrimo infrastructure (e.g. orchestration).

The following matrix in table 74 provides a summary of mapping between WP4.4 building blocks and infrastructure services.

Finally it is worth mentioning that some features (e.g. mobile IPv6) are not covered into WP4.4 but they are covered by other WP4.4 so they do not represent a gap in the overall Akogrimo architecture.

**Table 79 – WP4.4 design gap with respect to the Infrastructure services**

| | | Infrastructure services | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **Web Service Foundation** | **Naming** | **Security** | **Represent state** | **Notification** | **Transaction** | **Orchestr.** | **Interop.** | **Mobile IPv6** | **SIP** |
| **WP4.4** | VO Management | X | X | | X | X | | | X | | |
| | Business process enactment | X | X | | X | X | | X | X | | |
| | SLA high level | X | X | | X | | | | X | | |
| | Security infrastructure | X | X | X | X | | | | | | |
| | GrSDS | X | X | | X | | | | X | | |
| **Comment** | | All WP4.4 are based on WS foundations | All WP4.4 are based on the EPR model | The security infrastructure is based on WS-Security | All the WP4.4 state full services apply the WSRF pattern | In some cases WP4.4 services use notification and it is based on WS-Notification then on a publish and subscribe mechanism | | Orchestration is based on BPEL. No choreography is used. Service orchestration is key in Akogrimo | Interoperability is a key feature in WP4.4. all services use standards communication protocols based on WS specification | WP4.4 is transparent to Mobile IPv6 though all services use IPv6<br>WP4.4 is transparent to SIP Infrastructure though uses SIP Infrastructure features.<br>However, WP4.4 uses the related features as result of integration with other WPs, in particular the following features are used: location awareness, network transparency, hand-offability, roamability. | |

### 5.3.1.2. Execution management services

The execution management services deal with the management of units of work execution. The management spans from planning phase when execution is negotiated and resources allocated to support till to the actual execution when required by the user.

These features are managed by the underlying Grid infrastructure and then WP4.4 does not feet these capabilities, they are already covered by WP4.3. Anyway, the VO management building block includes negotiation features related to the requestor part while the offering is addressed in WP4.3.

The following matrix summarizes the above considerations.

**Table 80 – WP4.4 design gap with respect to the Execution Management Services**

| | | Execution Management Services | | | | | |
|---|---|---|---|---|---|---|---|
| | | Service container | Job manager | Resource selection service | Monitoring | Mobile service management | Negotiation |
| **WP4.4** | VO Management | | | | | | X |
| **Comments** | | WP4.4 does not include "execution management services". It covers just the part of the negotiation process. | | | | | |

### 5.3.1.3. Data services

WP4.4 does not include any type of data services.

### 5.3.1.4. Resource Management services

Resource management involves two different aspects:

- The availability of a manageable resource, that is a resource exposing management information
- The availability of management services able to understand management information and to take decisions.

Resource Management Services should define "standard" interfaces to be exposed by manageable services in order to make available information to be consumed by management services.

Examples of manageability functions can be:

- Monitoring the quality of a service
- Enforcing a service level agreement
- Controlling a task
- Managing a resource lifecycle
- …

Examples of Web Service specifications addressing this objective are WSDM and WSM. Though Akogrimo architecture provides some manageability functions (e.g. monitoring, SLA enforcing) they are not exposed using existing specification and this is clearly a gap that should be covered.

**Table 81 – WP4.4 design gap with respect to the Resource Management Services**

| | Resource Management Services | | | |
|---|---|---|---|---|
| | Manageability interfaces | Manageability functionalities | Context awareness | Signalling |
| **WP4.4** | WP4.4 does not include any kind of manageability features. This is a gap of the architecture that should be covered. The specific case related to the context awareness and signalling is covered by other WPs. | | | |

### *5.3.1.5. Security Services*

The Akogrimo security infrastructure span across all WP4.x; WP4.4 security infrastructure covers some of the mentioned security features focusing mainly on message level security.

In particular, the security infrastructure guarantees:

- Verification of identity through an integrated authentication mechanism between network and VO domains. The authentication is guaranteed through a network of trusted A4C servers (a component of Akogrimo architecture, see [6] for details)
- Mapping of identities between network and Base VO domain in order to support the integrated authentication mechanism
- An authorization mechanism based on Web Service security specifications that allows to have a fine grained access control authorizing invocation at method level.

The Akogrimo CA issues the credentials. The problem of converting credentials issued by different authorities has only been managed between mobile user and VO through the User Agent that represents a sort of bridge between external world and VO domain.

Finally, audit services have been introduced in the overall Akogrimo architecture also if a detailed design has not been provided.

The following matrix summarizes the above conclusions.

**Table 82 – WP4.4 design gap with respect to the Security Services**

| | | Security Services | | | | |
|---|---|---|---|---|---|---|
| | | Credential Validation and Trust services | Trust, Attribute and Bridge/Translation services | Authorization services | Audit services | Integration between network level and service level security |
| **WP4.4** | Security infrastructure | X | X | X | | X |

| | |
|---|---|
| **Comments** | WP4.4 security infrastructure does not include audit services and translation of credentials between domains. All the other primitives are supported including a particular kind of trust between network level authentication and service level access authentication. The integration with network level includes a specific identity management model that represents an identity mapping mechanism between different domains. |

### 5.3.1.6. Self Management service

Self management services have a really important roles because they can guarantee reliability of the infrastructure. Akogrimo architecture covers partially these capability and at different levels through the WP4.x.

The WP4.4 design does not include any feature to self manage situations that can be a potential risk for the overall system safety (e.g. failure of some WP4.4 services or hostile behaviours).

The Business Process Enactment building block has been designed to provide self adaptation during the execution of workflows if external events happen. While the design of the building block is quite flexible with respect to the type of manageable events (they can be defined by the workflow designer), the implemented architecture is characterized by the capability of managing context aware adaptation. The design is generic enough to be not coupled to specific context information but, in practice, there is some limitation due to the type of context information providers integrated into the infrastructure.

**Table 83 – WP4.4 design gap with respect to the Self Management Services**

| | | Self Management | | | | |
|---|---|---|---|---|---|---|
| | | Self configuring mechanism | Self healing mechanism | Self Optimizing mechanism | Adaptive business process | SLA adaptation |
| **WP4.4** | **Business process enactment** | | | | X | |
| **Comments** | | WP4.4 does not include any kind of self configuration and healing mechanism. except for the Business Process Enactment. The BPR supports mechanisms for self adapting the business process execution taking into account external events happening during the workflow execution. Akogrimo supports self optimisation mechanism (see WP4.3) | | | | |

### 5.3.1.7. Information service

WP4.4 does not cover in a comprehensive way the information services capabilities included in Table 72.

That is due to:

- The requirements of WP4.4 mainly lead to the needs of a directory service providing a sort of yellow page about published services
- Adoption of ad hoc solution (as in the case of logging) for each module of the WP4.4 building blocks without designing a common solution

A mechanism for integrated accounting is design but just partially implemented (there is not detailed design and implementation in WP4.4).

**Table 84 – WP4.4 design gap with respect to the Information Services**

| | | Information Services | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Discovery service | Message infrastructure | Logging service | Monitoring services | Signalling | General inform. and monit. service | Accounting |
| **WP4.4** | GrSDS | X | | | | | | X |
| **Comments** | | WP4.4 includes a directory service that acts as a sort of yellow page of the VO to allow publication of services. The other services are not directly supported in WP4.4. In general the Akogrimo design does not include "message infrastructure and general information and monitoring" at all. A common approach for logging is not available even if several logs are available.<br>A mechanism for integrated accounting between several domains has been designed (it is part of WP4.2). A partial gap exists with respect to Information Service | | | | | | |

### *5.3.1.8. VO management*

One of the main goals of Akogrimo was to design and implement an infrastructure for VO management. This capability is introduced at WP4.4 level and it is based on the introduction of Base VO and Operative VO concepts: the VO management building block design covers the VO creation, operation and dissolution phases of a VO lifecycle.

The introduction of the monitoring daemon component (Business Process Enactment building block) introduces the capability of monitoring the VO evolution trough the possibility of modifying its lifecycle taking into accounts upcoming events. Monitoring of OpVO services during the OpVO execution are not available.

**Table 85 – WP4.4 design gap with respect to the VO Management**

| | | VO Management | | | | |
|---|---|---|---|---|---|---|
| | | VO Creation | VO Operation | VO Evolution | VO Monitoring | VO Dissolution |
| **WP4.4** | VO Management | X | X | X | | X |
| **Comments** | | WP4.4 provides VO management capabilities. The design is not comprehensive of all the required capabilities in table 75 and in particular there is a lack of monitoring related to the services that are involved to manage the VO execution. The VO dissolution is also managed not in a complete automatic way with some lacks in the release of all involved resources. | | | | |

#### 5.3.1.9. Summary of identified gaps

The performed analysis shows that the existing WP4.4 design covers most of the selected target capabilities. Though those capabilities are sufficient to run appropriately the Akogrimo testbeds (in any case some changes are required, see section 4), the gap analysis showed that some gap exist.

In particular, the most relevant aspects that should benefit of a design improvement are:

- Resource management.

  The improvement should focus on two different areas:

  - Design of a resource management framework based on web service standards/specifications (e.g. WSDM, WSM). This framework has to provide general features to be used across the different work package
  - Design of a service management system based on this framework that will provide manageability functions related to the requirements of WP4.4 building blocks.

- Security services.

  Akogrimo deals with distributed environments and across domains communication. Apart from some limitations of the current detailed design (e.g. missing audit services), the most relevant identified gap is the lack of support for interactions between domains using different security infrastructures. The design of "bridges" allowing this kind of interoperation is still missing.

- Self management.

  The design presents a clear gap in this area because there are not features for self management of failures or hostile behaviour. The architecture design should cover this gap in order to guarantee a reliable infrastructure.

- VO Management.

  WP4.4 meets many requirements related to VO management, anyway, areas for further improvement have been identified: they are mainly related to the VO execution monitoring and failure managements.

### 5.3.2. WP4.3 analysis

The WP4.3 layer involves the Basic Grid Infrastructure Services [7]. The Grid Infrastructure Services Layer is positioned on top of the Mobile Network Middleware Layer and below the Application Services Layer. It is able to establish communication with all the layers, including the Mobile Network Layer. Its key aim is the provisioning and management of the Akogrimo Grid through Open Grid Services Architecture (OGSA) compliant grid services. [13]

The WP4.3 architecture includes 8 main building blocks:

- Execution Management Services.
- Data Management Service.
- Monitoring Service.
- Service Level Agreement Enforcement Services.
- Metering Service.
- Policy Manager.
- Semantic Service Discovery Service.

- Overall Security in Grid Layer.

Each of them covers some of the capabilities listed in Table 77 and Table 78.

More in detail:

- The Infrastructure Services represent capabilities that are met across all the building blocks.
- The Execution Management Services are met by the Execution Management Services building block.
- Data Services are met by the Data Management Service building block.
- Resource Management Services are partially met by the Execution Management Services building block.
- Security Services are partially covered by WP4.3.
- Self Management Services are partially met by the Execution Management Services, Monitoring Service, Service Level Agreement Enforcement Services, Metering Service and Policy Manager building blocks.
- Information Services are met by the Monitoring Service, Metering Service and Semantic Service Discovery Service building blocks.
- VO management services are not covered by WP4.3.

#### 5.3.2.1. Infrastructure services

"They are not services per se but assumptions related to the infrastructure on which all the other services should be built. They mainly deal with standards/specifications and properties to be provided by each OGSA service" quoted from Table 77.

Looking at this definition it is clear that the infrastructure services list is not expected to be covered by a specific building block but rather the building blocks should be built on them. That is the rationale behind the WP4.4 design as well, in particular the infrastructure services are not designed in the frame of the WP but existing solutions are used providing such capabilities (e.g. WSRF.NET, GT4 core,…).

The WP4.3 includes Grid Infrastructure Level Services: all the building blocks have been designed following the Service Oriented model and in particular the Web Service reference model.

On the basis of these assumptions all the WP4.3 building blocks are built on a subset of the infrastructure services: Web Service Foundations, Naming, Security, Representing State, Notification and Interoperability.

In some cases (notification, security) the infrastructure services are related just to specific building blocks because they are the only blocks that use a specific feature (e.g. notification).

The following matrix in Table 86 provides a summary of mapping between WP4.3 building blocks and infrastructure services.

**Table 86 – WP4.3 design gap with respect to the Infrastructure Services**

| | | Infrastructure services | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **Web Service Foundations** | **Naming** | **Security** | **Representing state** | **Notification** | **Interoperability** | **Mobility** | **SIP** |
| **WP4.3** | Execution Management Services | X | X | X | X | X | X | | X |
| | Data Management Service | X | X | X | X | | X | | |
| | Monitoring Service | X | X | X | X | X | X | | |
| | Service Level Agreement Enforcement Services | X | X | | X | X | X | | |
| | Metering Service | X | X | X | X | X | X | | |
| | Policy Manager | X | X | | X | | X | | |
| | Semantic Service Discovery Service | X | X | X | X | | X | | X |
| | Overall Security in Grid Layer | X | X | X | X | | X | | |
| **Comment** | | All WP4.3 are based on WS Foundations | All WP4.3 are based on the EPR model | The security infrastructure is based on WS-Security | All the WP4.3 state full services apply the WSRF pattern | In some cases WP4.3 services use notification and it is based on WS-Notification then on a publish and subscribe mechanism | Interoperability is a key feature in WP4.3. All services use standards communication protocols based on WS specification | WP4.3 is transparent to Mobile IPv6 though all services use it | WP4.3 directly uses features provided by SIP Infrastructure in the management of services on mobile terminal. |

### 5.3.2.2. Execution Management Services

The execution management services deal with the management of units of work execution. The management spans from planning phase when execution is negotiated and resources allocated to support till to the actual execution when required by the user.

These features are managed by the Execution Management Services building block within WP4.3.

The following matrix summarizes the above considerations.

**Table 87– WP4.3 design gap with respect to the Execution Management Services**

| | | Execution Management Services | | | | | |
|---|---|---|---|---|---|---|---|
| | | Service container | Job manager | Resource selection service | Monitoring | Mobile service management | Negotiation |
| **WP4.3** | Execution Management Services | X | X | X | X | X | X |
| **Comments** | | All Execution Management Services are carried out by the Execution Management Services (EMS) building block in WP4.3. | | | | | |

### 5.3.2.3. Data Services

The Data Services deal with data transfer, storage management, simple access, queries and so on.

These features are managed by the Data Management Service building block within WP4.3. It has been designed and developed to satisfy the basic requirements of the Data Management in a Grid environment. Taking into account the functional requirements of the Akogrimo domain applications, the focus has been put over three main areas: the transfer of data from one location to another, the storing of data and finally the access to the data stored.

The following matrix summarizes the above considerations.

**Table 88 – WP4.3 design gap with respect to the Data Services**

| | | Data Services | | | |
|---|---|---|---|---|---|
| | | Data Transfer | Storage Management | Simple Access | Queries |
| **WP4.3** | Data Management Service | X | X | X | X |
| **Comments** | | All Data Services are carried out by the Data Management Service (DMS) building block in WP4.3. | | | |

### 5.3.2.4. Resource Management Services

Resource management involves two different aspects:

- The availability of a manageable resource, that is a resource exposing management information
- The availability of management services able to understand management information and to take decisions.

Resource Management Services should define "standard" interfaces to be exposed by manageable services in order to make available information to be consumed by management services..

Examples of manageability functions can be:

- Controlling a task.
- Managing a resource lifecycle.

Example of Web Service specifications addressing this objective are WSDM and WSM. Though the Execution Management Services building block within WP4.3 provides some manageability functions, they are not exposed using existing specification and this is clearly a gap to be covered due to the distributed WP4.3 does not meet this type of capabilities and this is clear a gap to be covered.

**Table 89 – WP4.3 design gap with respect to the Resource Management Services**

| | | Resource Management Services | | | |
|---|---|---|---|---|---|
| | | Manageability Interfaces | Manageability Functionalities | Context Awareness | Signalling |
| **WP4.3** | Execution Management Services | | X | | X |
| **Comments** | | WP4.3 partially met manageability features through its Execution Management Services building block. Furthemore it use signalling capabilities provided by other WPs in order to manage the selection of service during the negotiation phase. | | | |

### 5.3.2.5. Security Services

The Akogrimo security infrastructure span across all WP4.x, then also WP4.3 security infrastructure covers some of the mentioned security focusing mainly on message level security.

In WP4.3 it is used the security infrastructure that is provided by the grid middleware platforms such as GT4 and WSRF.NET. These infrastructures support the functionality that OGSA specifies as credential and trust services, authentication and authorization, identity mapping, privacy, logging, etc.

Furthermore, in WP4.3 there is a service named Policy Manager that aims at enforcing various policies in Akogrimo.

In any case, the credentials are issued by the Akogrimo CA and the problem of converting credentials issued by different authorities is addressed just in a superficial way at design level.

Finally Audit Services have been introduced in the overall architecture but a detailed design was not covered by Akogrimo architecture

The following matrix summarizes the above conclusions.

**Table 90 – WP4.3 design gap with respect to the Security Services**

| | | Security Services | | | | |
|---|---|---|---|---|---|---|
| | | Credential Validation and Trust Services | Trust, Attribute and Bridge/Translation services | Authorization services | Audit services | Integration between network level and service level security |
| **WP4.3** | Security infrastructure | X | | X | | X |
| **Comments** | | WP4.3 security infrastructure does not include audit services and translation of credentials between domain. The integration between network level and service level security is supported by a chain of trust, that is, if the VO trusts the network and the EMS trusts the VO, then the EMS and the whole grid layer trusts the network. | | | | |

### 5.3.2.6. Self Management Services

Self Management Services have a really important role because they can guarantee reliability of the infrastructure. Akogrimo architecture covers just partially these capability and at different levels through the WP4.x.

The WP4.3 design includes the SLA Enforcement building block to self manage situations that can be a potential risk for the overall system performance (e.g. an SLA violation).

The following matrix summarizes the above conclusions.

**Table 91 – WP4.3 design gap with respect to the Self Management Services**

| | | Self Management | | | | |
|---|---|---|---|---|---|---|
| | | Self Configuring mechanism | Self Healing mechanism | Adaptive Business process | Self Optimizing mechanism | SLA Adapatation |
| **WP4.3** | SLA Enforcement | | | X | | X |
| **Comments** | | WP4.3 does not include any kind of self configuration and healing mechanism. WP4.3, through its SLA Enforcement building block, supports self optimizing mechanism. Furthermore features to detect SLA violations are available as well, though adaptive behaviour with respect to such detection violations is not provided. | | | | |

### 5.3.2.7. Information Services

WP4.3 partially met the Information Services capabilities included in Table 77. The Metering, Monitoring and Semantic Service Discovery Services building blocks within WP4.3 carry out these functionalities.

The following matrix summarizes the above conclusions.

**Table 92 – WP4.3 design gap with respect to the Information Services**

| | | Information Services | | | | |
|---|---|---|---|---|---|---|
| | | **Discovery service** | **Message infrastructure** | **Logging service** | **Monitoring services** | **General information and monitoring service** |
| **WP4.3** | Metering Service | | | | X | |
| | Monitoring Service | | X | X | | X |
| | Semantic Service Discovery Service | X | | | | |
| **Comments** | | The WP4.3 Semantic Service Discovery Service (SSDS) building block is a central repository of service meta-information, including low level, high level parameters as well as SLAs. The collection of such meta-information in a central repository enables advanced browsing, searching and discovery mechanisms. Besides, the WP4.4 includes the GrSDS for discovering services. In general the Akogrimo design does not include "message infrastructure and general information and monitoring service" at all, but the WP4.3 Monitoring Service building block uses the producer-consumer approach, providing to the SLA Enforcement with the information produced by the Metering service. A common approach for logging is not available even if several logs are available.<br>A mechanism for integrated accounting between several domains has been designed (it is part of WP4.2). A partial gap exists with respect to Information Service. | | | | |

### 5.3.2.8. Summary of identified gaps

The performed analysis shows that the existing WP4.3 design covers most of the selected target capabilities. Though those capabilities are evaluated sufficient to run appropriately the Akogrimo testbeds (in any case some changes are required, see section 4), the gap analysis demonstrates that some gaps exist in order to achieve a comprehensive solution.

In particular, the most relevant aspects that should benefit of a design improvement are:

- Infrastructure Services:

The improvement should focus on two different aspects:

- The WP4.3 Data Management Services, Policy Manager and Semantic Service Discovery Service building blocks do not support the Notification mechanism.
- The WP4.3 SLA Enforcement Services and Policy Manager building blocks do not implement security mechanisms. The SLA Enforcement Services foresees the possibility of using an X509-based system in order to protect the communication between the services.

- Resource Management Services.

  The improvement should focus on two different areas:

  - Design of a resource management framework based on web service standards/specifications (e.g. WSDM, WSM). This framework has to provide general features to be used across the different work packages.
  - Design of a service management system based on this framework that will provide manageability functions related to the requirements of WP4.3 building blocks.

- Security Services.

  Akogrimo deals with distribute environment and across domains communication, apart from some limitations of the current detailed design (e.g. missing audit services), the most relevant gap of the current security design is a lack of support for interactions between domains using different security infrastructures through the design of bridges allowing this kind of interoperation.

- Self Management Services.

  The design presents a gap in this area because there are not features for self configuring and self healing mechanisms. The architecture design should cover this gap in order to guarantee a reliable infrastructure.

### 5.3.3. WP4.2 gap analysis

This section gives an overview of these topics in Akogrimo related to the Mobile Network Middleware, which are covered in WP4.2. The goal of WP4.2 is to give the higher Grid layers the possibility to use a comfortable and reliable network infrastructure.

The five main parts of WP4.2 are:

- SIP Presence Handling
- A4C System
- Identity Management
- Context Management
- Service Discovery

These components are dealing with some of the capabilities mentioned in Table 77 and Table 78.

In detail, the following topics are addressed:

- Infrastructure Services are met in the SIP Presence Handling and the Context Management
- Execution Management Services are not part of WP4.2
- Data Services are not part of WP4.2

- Resource Management Services are provided in the SIP Presence Handling and the Context Management
- Security Services are used and established for A4C and Identity Management
- Self-Management Services are not part of WP4.2
- Information Services are partly covered in the Service Discovery topic

### 5.3.3.1. Infrastructure services

Context management and Presence handling support and extend the mobility features provided in Akogrimo. Mobility is supported on one side by the SIP Infrastructure implementation as nomadicity and on the other side by MIPv6 for full real-time mobility. Context and Presence enrich and take benefits of this mobility.

- SIP Presence Handling: It uses the SIP infrastructure to allow users and services to export their current presence status to all subscribers that request to be informed about it.
- Context Management: Fetches information from several sources to aggregate all the required information regarding the current context of a component, such as location (RFID-based, GPS…), presence (using SIP Infrastructure), etc. Subscriptions can be made to be informed about events on a certain subject.

**Table 93 – WP4.2 design gap with respect to the Infrastructure services**

| | | **Infrastructure services** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **Web Service Foundations** | **Naming** | **Security** | **Representing state** | **Notification:** | **Transactions** | **Orchestration** | **Interoperability** | **Mobile IPv6** | **SIP** |
| **WP4.2** | SIP Presence Handling | | X | X | X | X | | | X | X | X |
| | Context Management | X | | X | X | | | | | X | X |
| **Comment** | | Transactions and Orchestration are not part of WP4.2 and therefore these issues are not covered. | | | | | | | | | |

### 5.3.3.2. Execution Management Services

WP4.2 does not include any Execution Management Services.

### 5.3.3.3. Data Services

WP4.2 does not include any Data Services.

### 5.3.3.4. Resource Management Services

SIP Presence Handling and Context Management can be used to acquire the state, in the broad sense of the word, of any components that implement clients of these infrastructure services. On the other hand, a subscriber can obtain concrete information about specific changes of this status.

**Table 94 – WP4.2 design gap with respect to the Resource Management Services**

| | | Resource Management Services | | |
|---|---|---|---|---|
| | | **Manageability interfaces** | **Manageability functionalities** | **Context Awareness** |
| **WP4.2** | SIP Presence Handling | | | X |
| | Context Management | | | X |
| **Comments** | | Some other components complement these services (and feed from them) working at different levels, such as the Monitoring daemon | | |

### 5.3.3.5. Security Services

The network layer is the main source of trust and security in the Akogrimo concept: An initial login is done when the user selects a network and this login information is spread, related to the Single Sign On paradigm, to all other domains, but without violating user's privacy or the platform security. The WP4.2 components acting as Security Services are the following two:

- The A4C System is in charge of many authentication, authorization, accounting, auditing and charging issues. It is also responsible for issuing user's attributes to other authorization components in the network layer as well as in the Grid layer.
- The Akogrimo Identity Management components, here to be mentioned the SAML Authority and the Akogrimo Certification Authority, validate user's credentials for authentication issues and forward this information to other layers.

**Table 95 – WP4.2 design gap with respect to the Security Services**

| | | Security Services | | | | |
|---|---|---|---|---|---|---|
| | | Credential Validation and Trust services | Trust, Attribute and Bridge/Translation services | Authorization services | Audit services | Integration between network level and service level security |
| **WP4.2** | A4C | X | X | X | X | X |
| | Identity Management | X | | | | X |
| **Comments** | | WP4.2 provides services in each of the fields of the Security Services demanded in Table 72 and 73. | | | | |

### 5.3.3.6. Self-Management Services

WP4.2 does not include any Self-Management Services.

### 5.3.3.7. Information Services

Due to the layer-specific structure of the Akogrimo workpackages, WP4.2 covers only these issues of Information Services proposed in the Tables 72 and 73 that are related to the network infrastructure:

- Accounting is done by the A4C Server, a centralized platform component that handles besides authentication, authorization, auditing and charging also accounting issues.
- A Discovery Service is implemented in Akogrimo's Service Discovery approach, which enables the possibility to find network services as well as Grid services.

The implementation of the other services is not part of WP4.2, but assigned to other workpackages in activity 4.

**Table 96 – WP4.2 design gap with respect to the Information Services**

| | | Information Services | | | | | |
|---|---|---|---|---|---|---|---|
| | | Discovery service | Message infrastructure | Logging service | Monitoring services | General information and monitoring service | Accounting |
| **WP4.2** | A4C | | | | | | X |
| | Service Discovery | X | | | | | |
| **Comments** | | WP4.2 deals only with the network related services and provides therefore neither logging and monitoring services nor defines the message infrastructure. | | | | | |

#### 5.3.3.8. Summary of identified gaps

The analysis done in the former sections shows, that WP4.2 does not have large gaps related to the concept demanded in the Tables 72 and 73. Although there are many components that are mentioned in the tables and not covered by WP4.2, this workpackage benefits from the layered architecture of the Akogrimo platform.

Execution Management Services, Data Services, Self-Management Services and Information Services are not part of WP4.2 and do not have to be worked out in that workpackage. But this indicates not a disqualification of the platform, because all those services are covered by other workpackages, dealing with other (mostly higher) layers.

In the analysis of the demanded services that are (partially) covered by WP4.2, there can be also found some inconsistency related to the Tables 72 and 73. These missing services are no gaps, too; the problem is also derived from Akogrimo's layered architecture:

- Infrastructure Services: Although WP4.2 is only network layer related, it covers (partly) most of the Infrastructure Services. The missing Transactions and Orchestration cannot be mentioned as a gap, since those services are not WP4.2 specific.
- Resource Management Services: The observed gaps between the demands in the Tables 72 and 73 are also related to the layered workpackage structure. The services are covered in other workpackages in activity 4.
- Security Services: Since the network layer is the main source of trust and security in Akogrimo, it is clear that WP4.2 must not miss any topic. Therefore, no gap can be noticed.
- Information Services: Due to the fact, that most of the Information Services are related to higher layer in the Akogrimo structure, it is not a gap with respect to the whole platform that WP4.2 covers only a little of the demanded services.

As an overall statement, it can be mentioned that WP4.2 covers all issues from Tables 72 and 73 which are related to the network infrastructure. Gaps that can be observed are addressed in other workpackages in activity 4.

### 5.3.4. WP4.1 gap analysis

This section gives an overview of topics in Akogrimo related to the Mobile Network, which are covered in WP4.1. The goal of WP4.1 is to provide the underlying mobile network infrastructure.

The four main parts of WP4.2 are:

- Terminal Mobility
- Network Quality of Service
- SIP Service Provisioning
- Policy Based Network Management

Each of these parts of WP4.1 covers some of the capabilities mentioned in Table 77 and Table 78.

In particular, the following topics are addressed:

- Infrastructure Services are met in all parts of WP4.1
- Execution Management Services are not part of WP4.1
- Data Services are not part of WP4.1
- Resource Management Services are not part of WP4.1.

- Security Services not part of WP4.1
- Self-Management Services are not part of WP4.1
- Information Services are met by the SIP Service Provisioning

### 5.3.4.1. Infrastructure services

- Terminal mobility is provided by MIPv6. It allows users to roam across different access networks in a transparent manner.
- Network QoS: The network components that handle network quality of service. Included here are components such as Access Router that form the basis of Akogrimo's network infrastructure.
- SIP Service Provisioning: This is the SIP infrastructure and related services.
- Policy Based Network Management: Enables easier and more centralized configuration of the network elements.

**Table 97 – WP4.1 design gap with respect to the Infrastructure services**

| | | **Infrastructure services** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **Web Service Foundations** | **Naming** | **Security** | **Representing state** | **Notification:** | **Transactions** | **Orchestration** | **Interoperability** | **Mobile IPv6** | **SIP** |
| **WP4.1** | Terminal Mobility | | | X | X | | | | X | X | |
| | Network QoS | | | X | X | | | | X | | |
| | SIP Service Provisioning | | | X | X | | | | X | | X |
| | PBNM System | | | X | X | | | | X | | |
| **Comment** | | None | | | | | | | | | |

#### 5.3.4.2. Execution Management Services

WP4.1 does not include any Execution Management Services.

#### 5.3.4.3. Data Services

WP4.1 does not include any Data Services.

#### 5.3.4.4. Resource Management Services

WP4.1 does not include any Resource Management Services. The SIP infrastructure provides some functionality regarding this item, but it is considered to be the scope of WP4.2.

#### 5.3.4.5. Security Services

WP4.1 does not include any Security Services.

#### 5.3.4.6. Self-Management Services

WP4.1 does not include any Self-Management Services.

#### 5.3.4.7. Information Services

WP4.1 does not include any Information Services. The SIP infrastructure provides some functionality regarding this item and it allow the invocation of nomadic/mobile service .

#### 5.3.4.8. Summary of identified gaps

While WP4.1 by its own does not fill all the requirements, the remaining WPs do fill most of them. The missing gaps are explained by fact that 4.1 is a network oriented WP, and most of its design reflects that orientation.

Taking into account that WP4.1 forms the basis of the Akogrimo infrastructure, it is normal that apart from Infrastructure Services, few of the other services are covered by WP4.1. Instead, workpackages other than WP4.1 cover those.

# 6. Summary of final evaluation

This final section has the goal of providing a brief summary of the evaluation results distributed across the document. Furthermore, some sources of problems have been identified in order to give final feedbacks about research areas that could be investigated after the end of the project in order to improve the achieved results and to make a step forward towards the exploitation of Akogrimo in real life applications. The following paragraphs provide such summary with respect to the non functional requirements: interoperability, scalability, availability, performance and security.

## 6.1. Interoperability

Looking at the analysis from section 2.1, the updated architecture design has covered the most of open issues that arose during the first evaluation cycle. In particular, the following considerations can be done:

**REC01.** It is possible to state that Akogrimo is designed having interoperability as one of its guides. In fact, each module has been designed taking into account SOA based on the Web Service paradigm and open standards or specifications for defining communication among the Akogrimo modules.

**REC02.** From the viewpoint of integration of existing applications in the Akogrimo infrastructure, the main weakness of the first design was the need for migrating such application towards a Web Service model. This strong requirement is now outdated. In fact, the introduction of integration between SA and EMS allows for maintaining the Web Service communication paradigm but the SA and EMS can play now the role of a sort of bridge, so that different kinds of applications (e.g. web service, batch jobs) can be managed, even if they are still virtualized as Web Service by the infrastructure itself in order to communicate with them.

**REC03.** Though Akogrimo components are built using Web Service standards or specifications such as the most current Grid middleware, an interesting topic that should be investigated is how the Akogrimo platform can interact with other Grid middleware and how services deployed in other Grid middleware can be invoked from Akogrimo based system and vice versa. Of course this kind of interoperability cannot be guaranteed just by the use of standard protocols and specifications. This is an investigation that could be performed beyond the end of the project because the interoperability with other middleware can represent a relevant advantage for exploitation of Akogrimo results. It is worth mentioning that such a study requires an in depth understanding of the middleware that Akogrimo would interoperate with. Thus, it will be necessary to select the most relevant one (at least from exploitation goals viewpoint).

## 6.2. Scalability

The first architecture evaluation underlined some weakness in the Akogrimo design, mainly about the capabilities of managing an increased number of deployed services, hardware and/or external users in the Service Provider domain.

The final design (see [2] and [7]) describes how this capability is provided by the EMS together with the MDS:

- The former during the negotiation phase discovers a candidate set of services to be invoked and the related machine inside the Service Provider domain and selects a winner service.
- The latter maintains an updated list of such services and machines.

The introduction of this mechanism (partially part of GT4, as well) in the final design allowed to address most of the scenarios that were not met by the first architecture (see Table 8 to Table 14).

Though the updated design has provided clear improvements with respect to the recommendations of the first architecture evaluation, there a couple of topics that are could be further investigated:

**REC04.** There are some components that could create a bottleneck when the number of user increases. Thus an appropriate policy to scale them should be defined for real cases application of Akogrimo. These components are:

a. EMS: each invocation to a Service Provider domain passes through the EMS and it can become clearly a bottleneck as soon as the number of SP's customers increase

b. A4C: the authentication and authorization requests of user registered in the same network domain pass through the same A4C server (both request for authorize network access and for authenticate VO use)

c. GrSDS and PR: they are respectively the "yellow pages" and the participant registry of the VO. When the number of published services and VO participants increase they could become a bottleneck

d. OpVO creation management services: though each OpVO has a dedicated instance of services managing the OpVO creation and its execution, increasing the OpVO number in a Base VO can create bottleneck problems in the machine hosting the management services instances.

The current design of the modules that could become a potential bottleneck seems to allow for addressing such a problem by adopting well known solutions in their areas. In particular, all services that are distributed by nature (because different instances manage different entities) could be distributed physically as well; while other services (such as the GrSDS) can be made redundant with technologies already used (e.g. Web Servers).

**REC05.** Large scale scalability: an infrastructure as the one designed in Akogrimo is expected to be deployed in geographically distributed environment and new domains can join an existing Akogrimo Base VO. This implies that the system has to continue to work appropriately when new Service Provider and/or customer join the Base VO. This will increase the number of requests and then the overall infrastructure will need to be scaled in order to meet the increased request load. While the Akogrimo design being based on the SOA paradigm seems to be well designed to meet this kind of requirement, the configuration of the infrastructure does not seem to be a simple administrative task in in terms of scalability. In fact, the different demos, that have required an increased number of physical resources and the management of an increased number of services, have shown that the infrastructure is able to scale at different level but they have also shown that the re-configuration is not an easy administrative task. Additional feedbacks about this aspect will be provided by the execution of the final demonstrator that is supposed to be deployed in a geographically distributed site.

## 6.3. Availability

The analysis of the final architecture design shows improvements at several levels in terms of mechanisms for recovery from failures:

a. At SP domain level, the EMS includes recovery mechanisms to deal with service failures

b. At OpVO level, the SA can be thought as proxy for redirecting the invocations from the OpVO to other SPs, if the primary ones are not available any more

c. At WF level, alternative paths can be designed in order to recover from external failures that could result in an overall workflow execution failure.

d. At network level, failing lines can be replaced transparently with alternative lines (if they are available).

Though the above mechanisms have introduced improvements in the order to address availability issues, there are some components of the architecture that play a key role and their failure can affect the overall behaviour of the system. They are mainly the ones already mentioned in section 6.2, and in particular:

a. The A4C server: its unavailability will result in a failure in each authentication/authorization request

b. The SIP server: unavailability of the SIP server will result, for example, in unavailability of mobile services and of presence awareness capability.

c. VO management services: its unavailability will results in failures of VO operations and many of them are related.

d. The EMS: its unavailability will result in the related SP unavailability.

**REC06.** Due to their key role, such components cannot be a single point of failure and appropriated redundant design has to be foreseen when using Akogrimo in real applications.

## 6.4. Performance

In general performance attributes have to be evaluated through a quantitative evaluation. It is worth mentioning that the Akogrimo architecture has been implemented as a prototype, thus, the performance evaluation will provide only indicative feedbacks, as the prototype is not supposed to meet performance requirements which are typical in a real case application.

**REC07.** At this aim, the following tests are suggested to be performed on the final demonstrator (see section 4.3.3, as well):

a. Response time to create an OpVO

b. Response time for application service invocation

c. Response time for notification messaging

## 6.5. Security

The security matrix in Figure 2 shows that six relevant kinds of attacks have been identified (the ones labelled with the high level relevance, likelihood and impact):

- Users attack BVO or OpVO

- Terminals attack BVO or OpVO
- OpVO attacks SP
- Service Provider attacks User domain

For each of them an evaluation has been provided in section 3.1, describing how the architecture addresses the vulnerability related to those attacks. The design of the security infrastructure presents some critical aspects in relation with the communication across the boundaries of different domains involved in the end- to- end communication.

This communication is critical because it involves communications between domains based on different security technologies (e.g. between network and VO domain) and then they could present more vulnerabilities.

On the basis of this considerations, it is clear that the theoretical analysis performed in section 2.5 and 3.1 should be supported by quantitative evaluation performed on the final prototype.

**REC08.** At this aim, it is not possible to focus on all of the attacks identified by the security matrix but it is necessary to focus on one specific attack. It is recommended here to define tests related to the attacks from User to BVO/OpVO because it involves communication among many domains (i.e. network, customer and VO domain) and because it is based on the Akogrimo identity model that is the most relevant Akogrimo innovation introduced in the frame of the security design.

# References


[1] D5.3.2a "Architecture Evaluation Report" - members of Akogrimo consortium

[2] D3.1.3 "The mobile Grid reference architecture" – members of Akogrimo consortium

[3] D5.3.1 "Assessment Metrics, Test Cases and Guidelines" - members of Akogrimo consortium

[4] D5.3.2b "Prototype Evaluation Report" - members of Akogrimo consortium

[5] ID4.1.3 – "Updated Network Layer Architecture" - members of Akogrimo consortium.

[6] ID4.2.3 – "Updated Network Middleware Architecture" - members of Akogrimo consortium

[7] ID4.3.3 – "Updated Grid Infrastructure layer architecture" - members of Akogrimo consortium

[8] ID4.4.3 – "Architecture design update" - members of Akogrimo consortium

[9] Akogrimo Deliverable 2.3.2 – Validation Scenarios – members of Akogrimo consortium

[10] ID5.2.1 – "Test Bed Definition" - members of Akogrimo consortium

[11] http://www.grids-center.org/news/clusterworld/0505GridFinal.pdf

[12] Akogrimo annex I v2.0

[13] OGSA http://forge.gridforum.org/sf/go/doc13553?nav=1

[14] ID7.1.1 "Detailed Storyboard" - members of Akogrimo consortium

[15] D7.1.1 "Final Demonstrator" - members of Akogrimo consortium (to be delivered by 31/10/07)

# Annex A. Evaluation with respect to the results of first prototype testing

## A.1. FT-CA-Presence Awareness

### A.1.1. Requirement to be addressed

The platform must gather basic context information.

### A.1.2. Normal operation or failures in operation

The user uses the RFID card to allow the detection of his location with respect to a specific device inside a room. The device is contacted in order to start a service on it (e.g. patient data visualization). The detection of user from the RFID card reader does not guarantee the current presence of the user in the room when the service will be started.

A failure occurs if the RFID card reader does not detect correctly the presence and/or if the information is not notified to the interested parties.

### A.1.3. Potential problems

Test execution has pointed out some difficulties to configure the initial context due to some problems in detecting the presence information using the RFID card and the related detector device. These difficulties result just on the necessity of passing several time the RFID card on the detector device before having the RFID detection alert from the device.

The test has pointed out a percentage of failures (connection not established) but from a careful test reading it arises that they do not depend on wrong presence information, but on failure in the communication with some services involved in the workflow execution. In particular the percentage of failure is higher when the user is located close to a MT when the visualization involves mobile service invocations.

### A.1.4. Feedbacks about potential aspects to be investigated

To achieve a more accurate diagnose of the test, a network monitor system could be useful to detect network troubles (congestion, missing packets, etc...); moreover it can aid to distinguish client-side from server-side problems.

The RFID device could have defective hardware and a check is needed to prevent a breakdown in the course of a demo.

## A.2. FT-CA-Location Change

### A.2.1. Requirement to be addressed

Context information might change during service provisioning. The platform must take care of changes and notify applications and services accordingly.

### A.2.2. Normal operation or failures in operation

The starting context is BD (Big Display) or MT (Mobile Terminal) and for each test there is a context change moving respectively nearby the MT or the BD.

A normal operation consists in moving the visualization from the MT to the BD (from BD to MT respectively). If the visualization is not transferred when the user is nearby the other device there is a failure.

In the same test we have evaluated the presence awareness and after the context change. So we have measured the number of successful presence detection (i.e. the visualization of data appears in the BD or MT) with respect to the overall number of tests for each starting context (i.e. 50 tests for each) but the successful ones (i.e. the visualization appears) were respectively 41 and 37 for BD and MT. For each successful detection, a context change was tested (i.e. transfer from BD to MT or from MT to BD when the new location is detected and the percentage of successful transfer was 74% and 84% respectively).

### A.2.3. Potential problems

Similar considerations are valid here about the presence detection through the RFID card.

### A.2.4. Feedbacks about potential aspects to be investigated

Also in this case the considerations about the failure causes explained in FT-CA-PA are still valid.

## A.3. FT-Conference between Mobile Users

### A.3.1. Requirement to be addressed

The objective of this test is to verify that a conference can be correctly established between two mobile users.

### A.3.2. Normal operation or failures in operation

The test results was 100% right. All conferences were correctly established and there was not any kind of failures.

### A.3.3. Potential Problems

There are not specific potential problems associated to this test, but the conference will not be established if:

- Some of the users have not logged in.
- It could not satisfy the required QoS.
- Some of the users have a call established at the moment of test is being performed .

### A.3.4. Feedbacks about potential aspects to be investigated

The established communications through the multimedia application are bidirectional. The behaviour of the multimedia application in the situation of allowing multiple unidirectional

incoming calls (i.e. multiple incoming video streams) at the same time of trying to establish a conference with another mobile user should be tested.

It would be desirable to test the multiconference facility.

## A.4. FT-SLA_M-CPU usage

### A.4.1. Requirement to be addressed

The typical objectives of a performance requirement are to ensure that the business application or component:

a. Has an adequate capacity to handle actual usage conditions.

b. Minimizes latency and response times for its users and client applications.

c. Behaves according to required schedules.

The goal of low latency and response short time are strictly dependent on computational power (CPU). So to assure a good level of quality of service, it is needed to know in each instance how much CPU power is available and run corrective actions when a violation is detected.

### A.4.2. Normal operation or failures in operation

For each test a number of random CPU violations has been simulated. Different values have been used for each simulation.

### A.4.3. Potential problems

No problem detected.

### A.4.4. Feedbacks about potential aspects to be investigated

The saturation of the CPU usage could set back the violation handler process; probably the management functionality should have higher priority over the other services.

## A.5. FT-SLA_M-Disk usage

### A.5.1. Requirement to be addressed

The typical objectives of a performance requirement are to ensure that the business application or component:

a. Has an adequate capacity to handle actual usage conditions.

b. Minimizes latency and response times for its users and client applications.

c. Behaved according to required schedules.

d. Has an adequate throughput to support its users and client applications

### A.5.2. Normal operation or failures in operation

For each test a number of random disk space violations has been simulated. Different values have been used for each simulation.

### A.5.3. Potential problems

No problem detected.

### A.5.4. Feedbacks about potential aspects to be investigated

The disk space saturation can "hang" the hosting environment; the same care as for FT-SLA_M-D is valid.

## A.6. FT-AA-Failure/Success

### A.6.1. Requirement to be addressed

- Avoid that a user without right credentials can be authenticated within the Akogrimo environment.
- The main objective of the identification requirement is to ensure that all of the important externals are identified before granting access.
- Ensure that externals are actually who or what they claim to be.

### A.6.2. Normal operation or failures in operation

Actually the wrong authentications have been detected just with a rejected authentication using right credentials.

The percentage is against the overall number of test (6%) and against the test with right credentials (10% in rounded brackets).

### A.6.3. Potential problems

The latter is more significant. Anyway the rejected authentications cannot be considered as evaluation errors of the authentication server as they are the result of server unavailability.

### A.6.4. Feedbacks about potential aspects to be investigated

The demo and test was executed in a LAN network environment and in this scenario the server unavailability for network problem is a strange event; perhaps it needs better focus.

## A.7. FT-AA-TokenFailure

### A.7.1. Requirement to be addressed

Deny access to a malicious user in order to avoid he/she accessing services of another user with a wrong IDToken.

### A.7.2. Normal operation or failures in operation

This test has been done in sequence to the FT-AA-F/S. In particular,, we have 27 accesses with right IDToken returned after the first successful authentication. 23 accesses with wrong IDToken have been simulated.

### A.7.3. Potential problems

Also in this case as in FT-AA-F/S, the failures have been detected also in the requests with right IDToken and the rejected authentications are a result of server unavailability.

### A.7.4. Feedbacks about potential aspects to be investigated

Therefore the same comment of the FT-AA-F/S test is valid.

## A.8. VT-Data Manager

### A.8.1. Requirement to be addressed

The primary requirement that is addressed by the VT-Data Manager test is to detect problems in the network or CPU utilization when large amounts of data are transferred via GridFTP.

### A.8.2. Normal operation or failures in operation

The normal process is the continuous transmission of ECG data from the mobile terminal to the Medical Data Logger service and the ECG Analyzer. Additionally, the data can be transferred to one or more ECG Data Visualizer services. The test involves on the one hand the GridFTP library provided by GT4 and on the other hand the application services that act as data producer and data consumer. The following table shows the result of the performed tests:

**Table 98 – Results of VT-Data Manager test**

| | | Nr. of test cycles | Network QoS | % of successful transfers | COMMENTS |
|---|---|---|---|---|---|
| VT-DM | 1-1 | 20 (290 kByte chunk size) | n/a | 100% | ksat82 (ECG Analyzer + Medical Data Logger) showed a 100% CPU utilisation at the start of the test (first data transfer). Later data transfers resulted in only 50% CPU utilisation. ksat77 (Data Manager) had at most a 50% CPU utilisation. ECG Data Visualizer (ksat54) and ECG Data Generator (ksat121) worked well.<br>The larger chunk size resulted in a crash, and no data could be transferred. |
| | 1-10 | 4 (9.5 MByte chunk size) | n/a | 0% | |

### A.8.3. Potential problems

As this test was performed to detect architectural problems, we will not cover here the obvious implementation problems that resulted in a crash for larger chunk sizes. Architectural problems could not be detected. Also the transfer of data via GridFTP is rather a standard operation in a Grid environment. According to the May 2005 description of GT4 by Lee Liming and Ian Foster

"GT4: What's in It for You?" GridFTP is capable to transfer data as fast as 80 percent of the raw Iperf performance. In their tests the performance was limited by the disk subsystem or the network interface card, but not by the software. Obviously the data transfer in the prototype implementation could be optimized, but that was beyond the aim of the testbed implementation.

### A.8.4. Feedbacks about potential aspects to be investigated

Depending on the application requirements it could be investigated if the Mobile IPv6 Fast Handover, which can result in an interruption of the network connection for several seconds, imposes any problem for either the GridFTP library or the application services. But this is rather an implementation problem than an architectural shortcoming.

## A.9. ST-Heavy Load-A4C Server

### A.9.1. Requirement to be addressed

The ST-Heavy Load-A4C-Server test aims at validating the use of the A4C infrastructure in an environment with many simultaneous users where token requests and validations are frequent operations. The token generation occurs during log-in and the token validation before service use.

### A.9.2. Normal operation or failures in operation

During normal operation the user name and password are validated and a token is generated and sent back to the mobile terminal. Using the received token the user may get access to services.

**Table 99 – Results of ST-Heavy Load-A4C-Server test**

| | Token Generation | Token Validation | | COMMENTS |
|---|---|---|---|---|
| **ST-HL-AR** | 200 / sec. | 70 / sec. | | CPU utilisation in the A4C server and the SAML authority remained below 50%. This indicates that the processing power of the CPU is not a bottleneck. |

The results obtained by the test indicate that the prototype implementation and configuration of the A4C server could handle far more requests than necessary to support the scenario.

### A.9.3. Potential problems

Potential problems could occur if the number of requests exceeds the capacity of the A4C infrastructure. In that case log-in or service use could fail.

### A.9.4. Feedbacks about potential aspects to be investigated

There are no open issues related to token generation and validation by the A4C server.

# A.10. ST-HL-OpVO-User Agent

## A.10.1. Requirement to be addressed

The time that the User Agent needs to process an invocation is to be measured. The UA does the following (Figure 1):

- Contacts OpVO manager to validate token;
- Invoke a service method on the workflow engine.

In this test we define these parameters:

**Table 100 – Scenario Parameters**

| Scenario Parameters | |
|---|---|
| Number of UA | 10 |
| Average number of invocations | 45 per min |

## A.10.2. Normal operation or failures in operation

This test has been executed increasing the number of invocations per minute (fixing number of UAs). The result is the average value of successful invocations per min. This test has been performed increasing each minute one invocation per UA.

The final results show that increasing the number of invocation there is a decreasing of performance (increasing of response time).

## A.10.3. Potential problems

Some minor reliability problems are pointed out, as well. (See Appendix)

## A.10.4. Feedbacks about potential aspects to be investigated

No comment.

# A.11. ST-HNR-VOPI-Participant Registry

## A.11.1. Requirement to be addressed

With this test we are going to estimate:

- Response time
- Maximum number of participant
- Maximum number of parallel tasks
- Database operation response time and database used memory space.

### A.11.2. Normal operation or failures in operation

- This is the stress test related to the registration of a new participant. This test has been performed many times and each time with an increased number of participants. For each participant a registration request is started and for this reason the number of participants is the number of parallel registration requests. The response time has been evaluated taking into account the number and the size of WSRF.Net DB and Akogrimo collection entries. A registration request involves the creation of a ParticipantInfo instance and then the insertion of this instance into Akogrimo collection.
- This test evaluates the performance with respect to the profile updating and retrieving. It has been performed considering different number of existing database entries. For each fixed value several tests have been performed with several sizes of updated/retrieved profiles.
- This test evaluates the performance with respect to the retrieving of participant info. It has been performed considering different number of existing registered participants. For each fixed value different tests have been performed with several numbers of concurrent access and sizes of query responses.

### A.11.3. Potential problems

We need to remark that after the restart of WSRF.Net components, the first request is slower: it is a known behaviour as the IIS hosting environment needs to reload (and initialize) the code in memory at the first invocation. Therefore the test execution has to schedule a "warm up" stage to put the service into the working condition.

### A.11.4. Feedbacks about potential aspects to be investigated

We have noted that the response time increases as the number of registered participants (performed test) increases and then the DB entries should affects this parameter but deleting all the DB entries there is not an improvement. For these reasons further investigations are required in order to understand which components of ASP.Net pipeline, WSRF.Net pipeline and service use an incorrect allocation/de-allocation mechanism of resources.

## A.12. PT-SIP Broker Response Time

### A.12.1. Requirement to be addressed

The objective of this test is to estimate the delay introduced by the A4C Server in the SIP Broker response time under different workload conditions, because this is the unique module introducing delay.

### A.12.2. Normal operation or failures in operation

As expected, the A4C Server introduced a little delay under demanding workload conditions. The test failed (a big delay) under having an A4C occasionally running very slow.

### A.12.3. Potential Problems

If one of the users doesn't repond before the configurable time-out expires, the SIP Broker will not suffer problems, but the SIP call request remains in its mobile terminal and it should be desirable to cancel it.

### A.12.4. Feedbacks about potential aspects to be investigated

Currently, the SIP Broker supports multiple calls simultaneously. Then, it would have sense (and desirable) to repeat the test and measure the response time in function of a variable number of SIP call requests.

## A.13. PT-SIP Server Memory Usage

### A.13.1. Requirement to be addressed

The objective of this test is to measure the memory consumed by the SIP server in function of a different number of users publishing simultaneously their presence information, and also in function of a different size of their presence file.

### A.13.2. Normal operation or failures in operation

The test results are normal, with a little growth of the memory usage in the situation of having too many users and a big presence file.

### A.13.3. Potential Problems

None identified.

### A.13.4. Feedbacks about potential aspects to be investigated

None identified.

## A.14. FT-Network Handover

### A.14.1. Requirement to be addressed

The verification that a Mobile IPv6 handover is performed using Fast Handover.

### A.14.2. Normal operation or failures in operation

Using the FHO application, a Mobile IPv6 handover is triggered.

### A.14.3. Potential problems

No problem detected.

### A.14.4. Feedbacks about potential aspects to be investigated

FHO can be further optimised. Use of specific drivers for wireless cards, optimised for this operation can drastically improve the performance. Development of those kinds of drivers is however, out of the scope of Akogrimo.

## A.15. FT-Bandwidth Reservation

### A.15.1. Requirement to be addressed

Performing a network QoS reservation. This involves:

- Access Router detects network flow
- Access Router asks QoS Broker for rules to apply to the network flow
- QoS Broker checks user profile and responds to router

### A.15.2. Normal operation or failures in operation

After configuring the QoS Broker, a network flow is started from one mobile terminal to another.

### A.15.3. Potential problems

No problem detected.

### A.15.4. Feedbacks about potential aspects to be investigated

None identified