D3.1.1

Overall Architecture Definition and Layer Integration

WP 3.1 Overall Architecture

Dissemination Level: Public

Lead Editors: Jürgen Jähnert, USTUTT/RUS
Stefan Wesner, USTUTT/HLRS

07/07/2005

SIXTH FRAMEWORK PROGRAMME
PRIORITY IST-2002-2.3.1.18

Grid for complex problem solving
Proposal/Contract no.: 004293

License

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

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

1. Definitions

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

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

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

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

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

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

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

a. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any reference to such Licensor or the Original Author, as requested.

b. You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation. The exchange of the Work for other copyrighted works by means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.

c. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work, You must keep intact all copyright notices for the Work and give the Original Author credit reasonable to the medium or means You are utilizing by conveying the name (or pseudonym if applicable) of the Original Author if supplied; the title of the Work if supplied; and to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work. Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

d. For the avoidance of doubt, where the Work is a musical composition:

   i. **Performance Royalties Under Blanket Licenses**. Licensor reserves the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work if that performance is primarily intended for or directed toward commercial advantage or private monetary compensation.

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

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

5. Representations, Warranties and Disclaimer

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

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

7. Termination

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

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

8. Miscellaneous

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

Context

Activity 3	**Architectural framework**
WP 3.1	**Overall Architecture Definition and Layer Integration**
Dependencies	The Description of Work, DoW, places the Overall Architecture to be developed in this Deliverable D3.1.1 of WP3.1 in relation to the following Workpackages and their Deliverables: D3.1.1 is influenced by • WP1.1 Market and Regulations, WP2.2 Environment, WP2.3 Test-bed Definition o D2.1.1 The Akogrimo market players o D2.1.2 Regulation report o D2.2.1 Report on the socio-economic environment o D2.2.4 Report on State of the Art o D2.3.1 Test-bed Definition o D4.2.1 Overall Network Middleware Requirements report D3.1.1 influences and is influenced by • WP3.2 Business Modelling and its forthcoming Deliverables o D3.2.1 The Akogrimo consolidated Value Chain o D3.2.2 The Akogrimo Business Modelling Framework Finally, D3.1.1 influences the implementation Workpackages WP4.1, Network, WP4.2 Network Middleware, WP4.3, Grid Infrastructure, and WP4.4 Grid Applications – according to the specializations and restrictions of the Overall Architecture elaborated in WP2.3/D2.3.2, 'Validation Scenarios'.

Contributors

Contributors (in alphabetical Order):	Reviewers:[1]
Annalisa Terracina (DATAMAT) Anniello Rovezzi (CRMPA) Antonis Litke (NTUA) Arantxa Toro (TID) Brynjar Aage Viken (Telenor) Burkhard Stiller (Universtiy of Zurich) Christian Loos (University of Hohenheim) Cristian Morariu (University of Zurich) Eduardo Oliveros (TID) Francesco Verdino (CRMPA) Giuseppe Laria (CRMPA) Ignaz Müller (USTUTT-HLRS) Jan Wedvik (Telenor) Jürgen Jähnert (USTUTT-RUS) Martin Waldburger (University of Zurich) Nuno Inácio (IT Aveiro) Patrick Mandic (USTUTT-RUS) Per-oddvar Osland (TN) Pierluigi Ritrovato (CRMPA) Robert Piotter (USTUTT-HLRS) Rui L.A. Aguiar (IT Aveiro) Ruth del Campo (USTUTT-RUS) Stefan Wesner (HLRS) Stefano Beco (DATAMAT) Victor A. Villagra (UPM)	Stefan Wesner (USTUTT-HLRS) Christian Loos (University of Hohenheim) Julian Gallop (CCLRC) Juan Burgos (TID) Mario del Campo (TID) Rosa Vieira (TID)

Special Thanks to Paul Christ (formerly USTUTT-RUS now retired) who reviewed large parts of this documents with short notice and was providing useful input and help during the preparation of this document

Approved by: Stefan Wesner, University of Stuttgart, Germany, as IP Manager

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

Version	Date	Authors	Sections Affected
0.1	29.12.04	Jürgen Jähnert, Stefan Wesner, Paul Christ	Initial Version
0.1.1	01.01.05	Paul Christ	Extending the ToC; integrating partner contributions so far
0.3	6.3.05	Juergen Jaehnert	All
0.4	19.4.05	Juergen Jaehnert	All
0.5	18.5.05	Juergen Jaehnert	
0.6	20.5.2005	Juergen Jaehnert	All
0.7	19.6.2005	Julian Gallop	Took section 5.2.4 and split it into 2: 5.2.4 is now labelled VO Management but is now really about VO Management; new 5.2.5 is the material on SIP and SOAP, now labelled correctly.
0.8	24.6.2005	Julian Gallop	Minor revisions to those same sections
1.0	07.07.2005	Stefan Wesner	Incorporation of comments from the internal reviewing process and drafts of the deliverables from WP4.x

Table of Contents

List of Figures

List of Tables

Abbreviations

Abbreviation	Expansion
A4C	Authentication, Authorisation, Auditing, Accounting, Charging
Akogrimo	Access To Knowledge through the Grid in a Mobile World
BP	Business Process
CORBA	Common Object Request Broker Architecture
CSTA	Computer Supported Telephone Applications
DCOM	Distribute Component Object Model
DMTF	Distributed Management Task Force
ED	End Device
EE	External Entity
EN	Enterprise Network
GIS	Grid Information Service
HTTP	Hypertext Transfer Protocol
ID	Initial Device
IETF	Internet Engineering Task Force
IP	Internet Protocol
MDVO	Mobile Dynamic Virtual Organisation
MM	Multimedia
MOWS	Management Of Web Service
MSMQ	Microsoft® Message Queue
MT	Mobile Terminal
MUWS	Management Using Web Services
NO	Network Operator
OGSA	Open Grid Service Architecture
OMA	Open Mobile Alliance

Abbreviation	Expansion
OpVo	Operational Virtual Organisation
PDA	Personal Digital Assistant
QoS	Quality of Service
RTP	Realtime Protocol
SAML	Security Assertions Markup Languages
SDP	Session Description Protocol
SIMPLE	SIP extension for ??
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SMTP	Simple Mail Transport Protocol
SP	Service Provider
SSO	Single Sign On
TCO	Total Cost of Ownership
TCP	Transport Control Protocol
TMF	Telemanagement Forum
UDDI	Universal Description, Discovery and Information Protocol
UMTS	Universal Mobile Telecommunication System
UNICORE	Uniform Access to Computing Resources
URI	Uniform Resource Identifier
VO	Virtual Organisation
WLAN	Wireless Local Area Network
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language

1. Executive Summary

This document describes the Akogrimo architecture after the first iteration process of the architecture definition phase. The architecture is expected to be refined during the following iteration loop. The major task of WP3.1 is to enable the parallel development on different layers within Activity 4.

The Akogrimo project is driven by the basic idea that Next Generation Grids should be built on Next Generation Networks. This means that an Akogrimo NGG must be able to address the needs of an environment where users experience potentially fast changing context (Bandwidth, Device capabilities, Location, …), different access network providers and local services while aiming to participate in complex collaborations using resources provided by service providers from different organisations.

In order to address this challenging task to build an overall architecture for such a highly dynamic environment in this document the basic concepts and terms are defined. The approach chosen in this document is to define the concepts on a conceptual layer using UML class diagrams showing the connections and dependencies between them. Additionally in a more technical layer important aspects are described in greater detail such as different types of mobility.

A specific part for solving the problem on how to allocate responsibilities is to look at issues that span several layers. These cross layer issues are crucial for a successful integrated grid middleware The major elements identified are problem around identity and security management, cross organisational accounting, the handling of context and the interrelation of SIP and SOAP as basic protocol element.

The last part of this document is dedicated to the identified building blocks and their allocation to the different layers which will be further detailed in the respective deliverables of Activity 4. The last section of the document provides a list of the subsystems and the interactions on a high level view.

2. Guide through this document

This document is seen to be a special deliverable as it is supposed to be the entry point for a technical persons interested in the Akogrimo Project. For this reason this document start with a general description of the opportunities and challenges addressed within Akogrimo and introduced the concept of cross-layer cooperation.

The following section 4 is aiming at the definition of fundamental terms such as the "Mobile Grid" as understood by Akogrimo and external roles to the Akogrimo System. This section is related and inspired by D3.2.1 The Akogrimo consolidated Value Chain.

One of the major challenges identified during the architecture definition process was the lack of clear definitions of terms and concepts in particular in the Grid domain and the duplication of terms with different meaning within domains and in particular across domains. For this reason basic concept and termns and selected cross layer issues are tackled in sections 5 and 6.

As the major goal of this document is to clearly define the boundaries between the different layers in order to allow the Activity 4 workpackages to organise their detailed design and implementation work in section 7 the identified Building Blocks are allocated to one of these layers and described. This section partially contains information described already in greater detail in other deliverables in particular D4.2.1 Overall Network Middleware Requirements report contain a more detailed description of the components in 7.2.1. The details on the interchange between the blocks are subject to be detailed and updated during the design and implementation phase in Activity 4.

The document is concluded with diagrams showing for selected key scenario the interaction between the high level components (section7) and a listing of the identified interfaces in an informal description in section 8.

Additionally a short summary of the requirements from the testbeds are provided in section Annex A. This section does not provide a complete view so that the interested reader is directed to D2.3.1 Testbed Definition and D2.3.2 Validation Scenarios.

This document will be updated in the second cycle of this project and we appreciate all comments you might have. You may either use the facility provided by our content management system in order to place comments directly at the location of this document in the download section of our web site at http://www.mobileGrids.org or send a comment to the following mail address: akogrimo-info@forge.hlrs.de

The most recent version of this deliverable and all other public deliverables can always be downloaded from the Akogrimo web site here

3. Akogrimo Specific Challenges and Opportunities

The integration of modern, mobility aware networks with Grid concepts looks at a first glance artificial. This section will provide a motivation why this combination is beneficial from a technical viewpoint. The economic dimension, which is at least equally important, is addressed in the existing and forthcoming deliverables of WP2.1, WP3.2 and WP6.3 in particular D3.2.1, the Akogrimo consolidated Value Chain and D3.2.2 The Akogrimo Business Modelling Framework. However this integration also brings additional challenges to be solved. Akogrimo, following the fundamental Grid concepts, is accordingly embracing the paradigm of resource sharing in the context of Virtual Organizations. As a major differentiator from other projects enabling the commercial exploitation of Grids, Akogrimo is considering highly dynamic Virtual Organisations (VO) that need to adapt to changing contexts, to take into account the location of a service and feature the concept of a user. This character of Akogrimo's VO concept in turn requires the vertical, i.e. cross-layer integration and interoperation of 'lower layer' functionalities such as Mobility management and user-related AAA functions with Grid middleware.

Similar considerations and related work can be found in Grid projects targeting as well architectures for mobile environments. In particular the Wireless Grid project [8] and a Mobile Grid project in Korea [6] and to some extend also the GridLab project [7] have addressed in their architecture similar basic requirements. Recently also in the commercial sector attempts to design architectures for this domain have been published as in [5].

3.1. Opportunities and Challenges

Many Grid solutions as of today are targeting rather static Virtual Organizations where the participating organizations are infrequently changing or the resources to be shared are provided by one single company. For this kind of settings the motivation for using Grids is driven by considerations around reduction of Total Cost of Ownership (TCO) often in conjunction with reorganisation of hardware infrastructure from large mainframe systems towards cluster based solutions.

We believe that the chosen Akogrimo approach to rely for basic properties of the overall system on the developments outside the standard Grid world is providing new opportunities in particular in the following areas:

- User and identity management
- Cross organisational accounting
- Integration with multimedia collaboration tools
- Device and Context Awareness

The following table elaborates on these issues:

Area	Specific Challenges and Opportunities
User and Identity Management	The security and identity models in the Grid domain are not designed for a large number of users (for example the UNICORE security model see [2]). For typical deployments of Grids the number of users is several hundred maybe some thousand users. Furthermore as rather

Area	Specific Challenges and Opportunities
	static Virtual Organisation has been considered a centralised user and identity model has been chosen. For Akogrimo the more advanced systems from the network domain, designed for several million users, will enable different solutions in particular federated identity models which are seen as important property of a real dynamic Virtual Organisation.
Cross Organisational Accounting	The realisation of an accounting system spanning several companies is not yet sufficiently addressed in Grid solutions. Either accounting is completely handled out of band of the Grid solution or a central management is realised such as in [3]. For the kind of dynamic Virtual Organisations considering mobility new solutions are needed. Solutions from the network domain enabling this kind of cross organisational accounting for network services might be expanded to cover accounting for Grids in a similar way.
Device and Context Awareness	For existing Grid solution the network is seen as a simple transport layer. For supporting mobile participants applications and services must be aware of the current device capabilities and the context (e.g. bandwidth, network provider, network type, "home" or "foreign", and possibly the geographic location etc.). Without the provision of context it is hard to see how mobile participants could be supported appropriately. The way how context data is provided to the Grid middleware is again a key difference to existing solution providing portal based access to the Grid resources such as targeted in [6] [7] [8]. If a component within a Grid suddenly disappears this is typically considered to be a failure situation and recovery mechanisms for overcoming this failure are started (e.g. replacing the service with another one). In a mobile environment this must not necessarily be a failure condition. A device (maybe even driven by a user decision) might have changed the underlying access network or entered a tunnel. An architecture for Mobile Grids must support adaptation to such kind of changing conditions
Integration with Multimedia Collaboration Tools	The integration of the user in the Grid middleware enabling workflows spanning not only compute or data services is becoming possible as Grids are using the same identity and accounting model as the multimedia communications. So in contrast to the approach taken in AccessGrid [4] the security and identity model from Grids is not pushed down to be used for collaborative applications such as Videoconferencing but the other way around. ??? hab I et verstanden

3.2. The concept of Cross-Layer Cooperation

As mentioned above, Akogrimo's VOs are context-aware. Some related context changes will be handled **intra-layer**, e.g. within the Grid Infrastructure Layer an SLA management system may replace a non-performing service with another one. In addition – and this is Akogrimo-specific – also (mobile) users and user devices as potential members of a Virtual Organisation including their changing contexts are considered. These context properties such as "Location" and "Mobility" or even SIP-based presence attributes need to be addressed by what we call **cross-layer** or **inter-layer** integration. As an example consider the move of a session from a Workstation to a Mobile Device (Network Infrastructure Layer). The changing context (different bandwidth, screen size, …) is notified through the SIMPLE protocol [21] to a Context Manager in the Network Middleware Layer, see section 7.3.1 which is sending a notification based on the Web Service Notification family of specifications to several subscribers. One of the subscribers is the Business Process Enactment Engine (Application Support Layer) that put the further execution of the workflow on pause. Additionally the Operational Virtual Organisation Manager Service is starting a process of finding from the workflow repository the appropriate workflow to be executed in this case (Network Middleware) and is adding a new service provider and monitoring infrastructure (Grid Infrastructure Layer).

Another – key cross-layer issue in Akogrimo is the vertically integration of AAA providing the needed parameters in only one round-trip via the related access network and core network protocol protocols and mechanisms directly to the SOAP-based Authentication and Authorization based ones. Furthermore AAA is expected to be an instrument for realising cross organisational accounting also across different layers from network over grid middleware up to the application layer.

4. Conceptual Model

The conceptual model describes key aspects of the architecture of Akogrimo. In order to address the question how these concepts are related to the components of the architecture, in the following, some diagrams include both, concepts and components. Components are indicated by a grey background colour.

4.1. Services

Services are the logical entities that are provided, managed and coordinated within a Virtual Organisation. Services are accessed via an interface that is part of the service description. Service Level Agreement (SLA) documents are used to describe and constrain the quality of service. Services can be consumed by end users or service aggregators. Aggregated services are exposed like generic services. If the service aggregator also manages the service execution he is also a service provider.

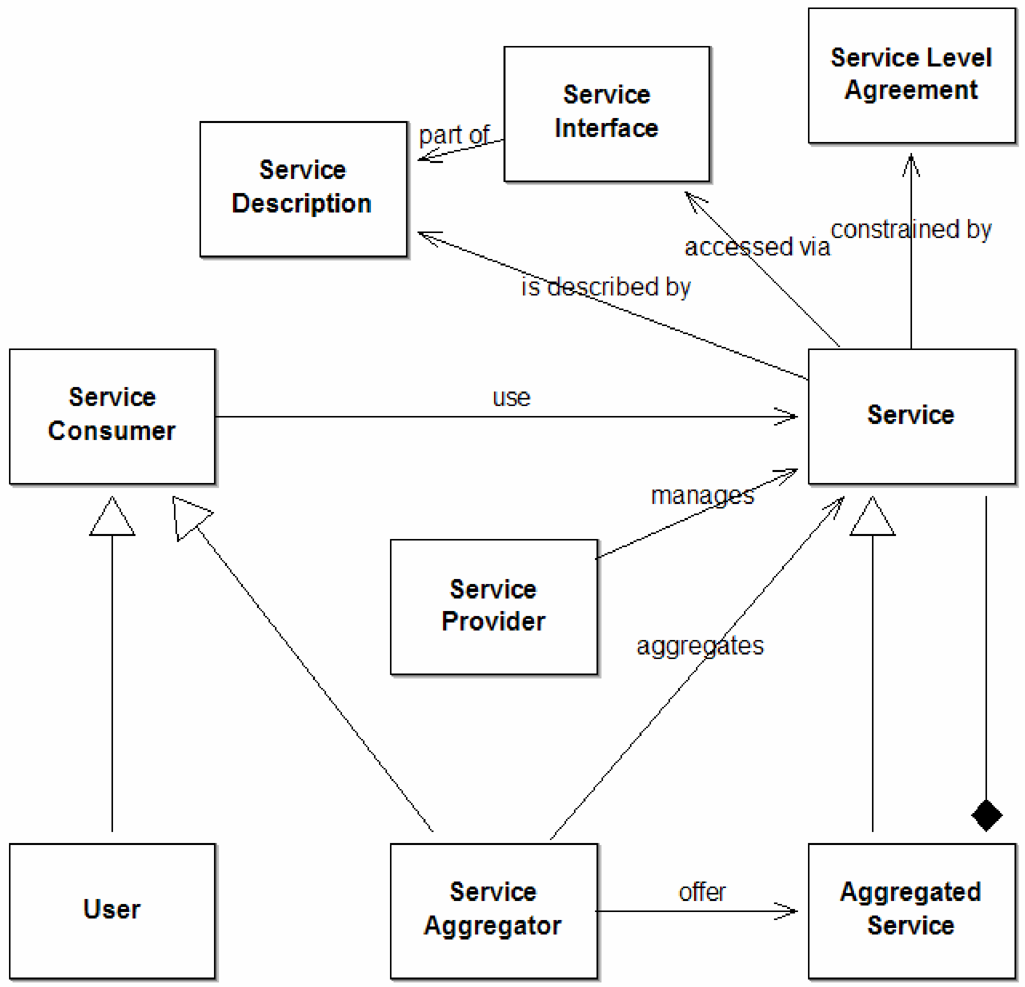


Figure 1 – The Akogrimo Service concept

4.1.1. Local Services

Local services are hosted by a device that can be associated with a geographical location. Local devices of interest to the VO may be general purpose devices like printers and displays or may be specific to the business process, like medical devices or other application specific equipment.

4.2. Service Provider

A service provider manages one or more services. The hosting environment of the service provider may contain several computational and hardware resources. Services and resources are monitored to assure that they are used in compliance with local policies. These policies are enforced by the local policy manager, which uses the monitoring component to check the resource and service usage.

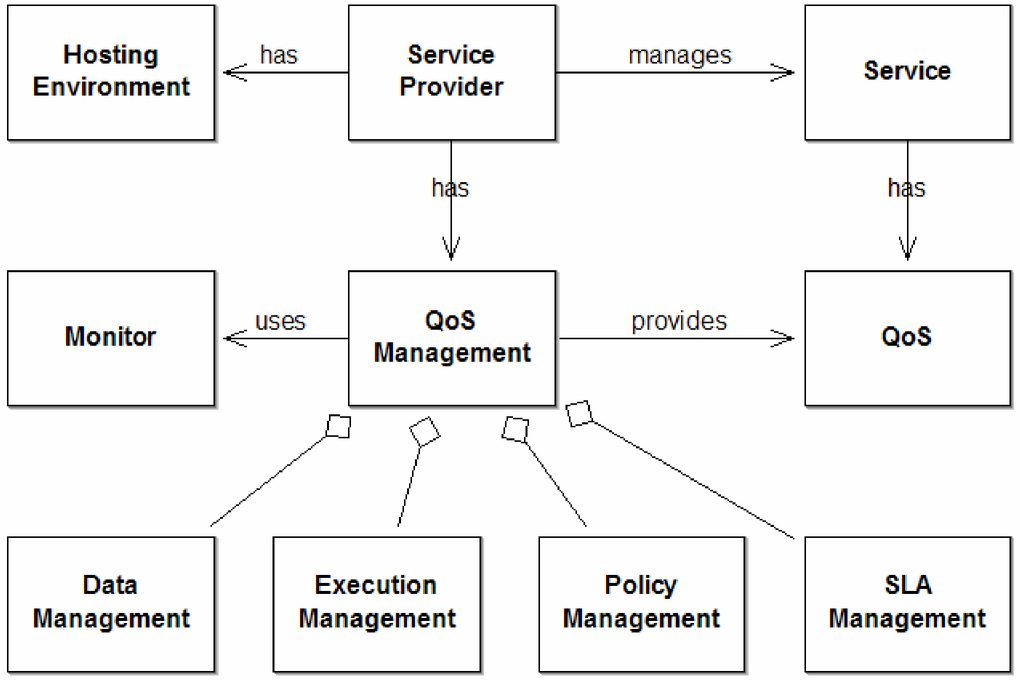


Figure 2 – The Akogrimo Service Provider concept

Quality of service management might be more or less sophisticated according to the facilities of a service provider. Large scale service provides may even have redundant resources to ensure reliability. Also the execution and data management technologies could be very different for single PC hosted services and for server farm hosted services. A monitoring component is used to provide the QoS management with information about the current performance.

4.3. Network

Devices and specifically user terminals are connected to an access network which provides mechanisms to query and specify the quality of service (QoS). From the access network, which may use technologies like Bluetooth or WLAN, data transport continues in the core network. Because of the user's connection to a device their context information can be attributed with network QoS parameters.

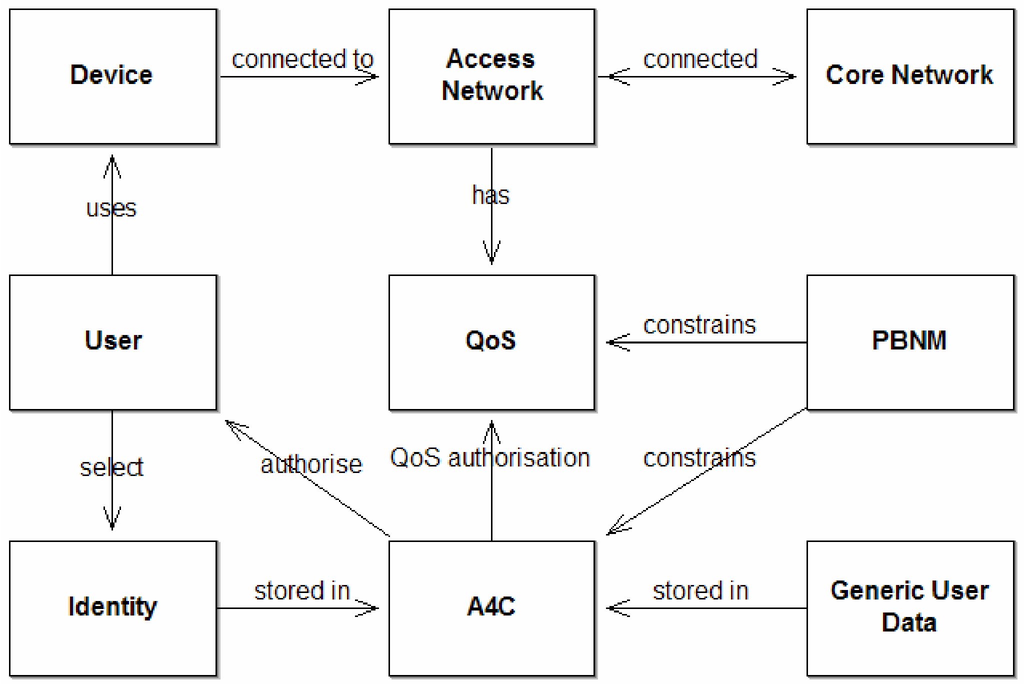


Figure 3 – The Akogrimo Network concept

Before being authorised to use the access network a user has to authenticate with the A4C server. In this process a user selects an identity. Generic user data and associated identities are stored in the A4C server.

4.4. Context

Context information is connected to a user via the device(s) he is using. Context information is used to adapt the workflow, e.g. by adapting the services to the context or by changing the workflow. Context information is essential to workflow adaptation in an environment with mobile users. The context may include user attributes like spoken languages or disabilities, environmental information like indoors/outdoors, geographical location information, device capabilities like display resolution and network capabilities like the bandwidth.

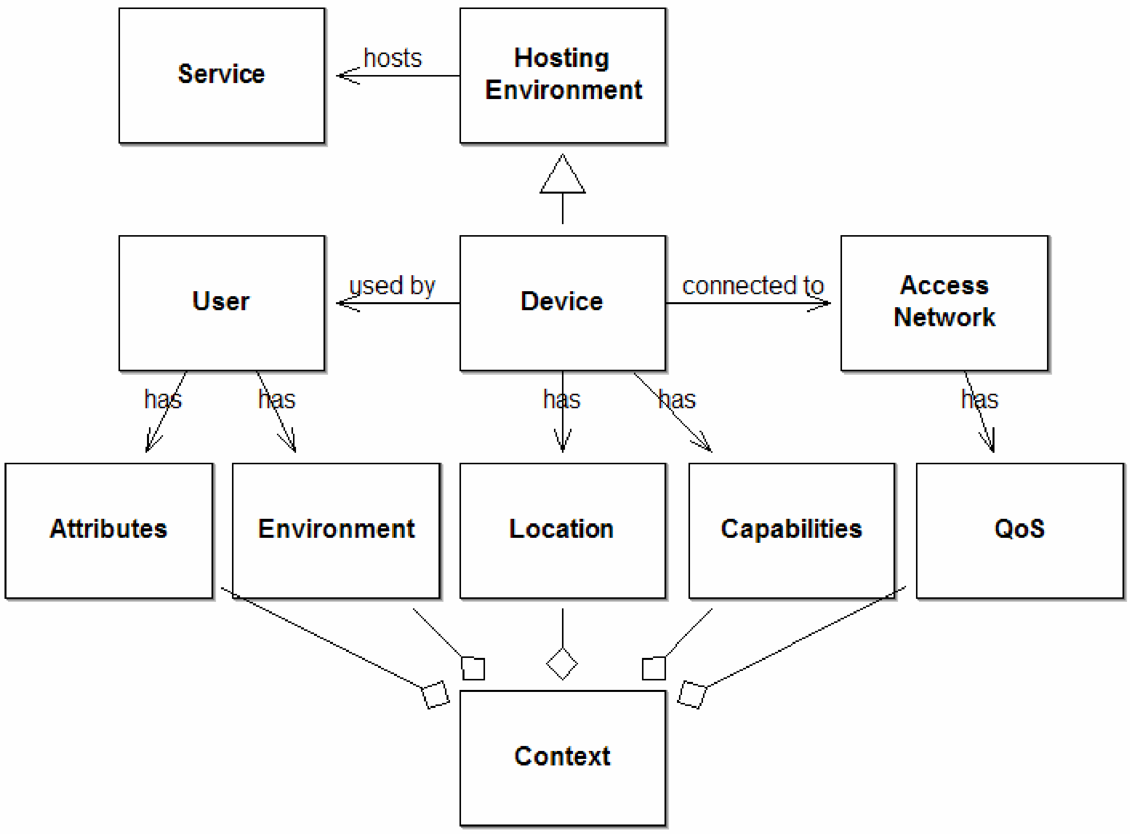


Figure 4 – The Akogrimo Context concept

4.5. Virtual Organisation

A Virtual Organisation (VO) provides services and the means to manage and coordinate them. In order to implement a business process an operational Virtual Organisation (OpVO) is created out of a base VO. The base VO is a Virtual Organisation that is not running a specific business process, but provides the mean for creating and supporting it. The base Virtual Organisation provides the means to register users, services and other meta-data like SLAs and workflow templates. These repositories are used by the operational VO when a business process is instantiated and executed.

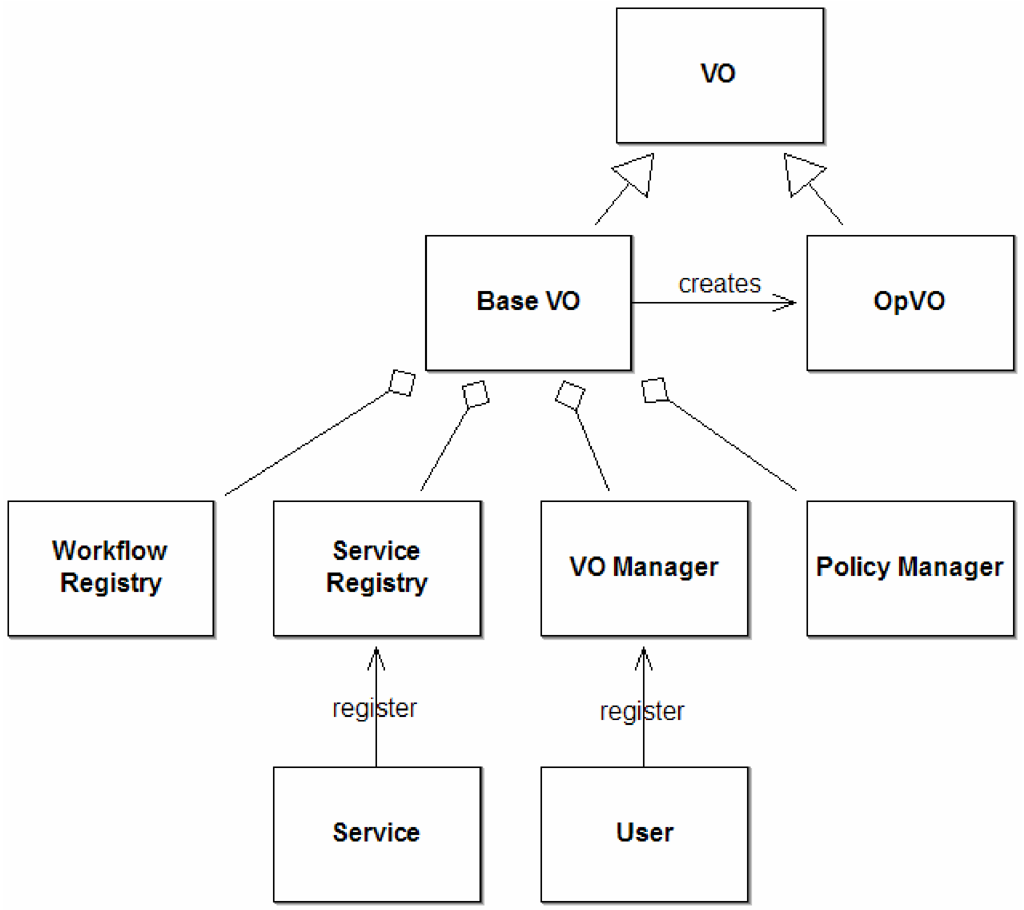


Figure 5 – The Akogrimo Base VO concept

A VO may use a policy manager to store and distribute policies that apply to VO members and services. Services may have their own associated policy manager.

4.6. Operational Virtual Organisation

The purpose of the operational VO is to instantiate a business process and to manage the execution. Services of the operational VO are offered to the users through the operational VO broker (OpVO Broker). The OpVO Broker uses the service discovery server to find the services that will be used in execution of the workflow that is part of the business process. The business process consists of one or more workflows that are set up and executed by the business process management component (BP Management).

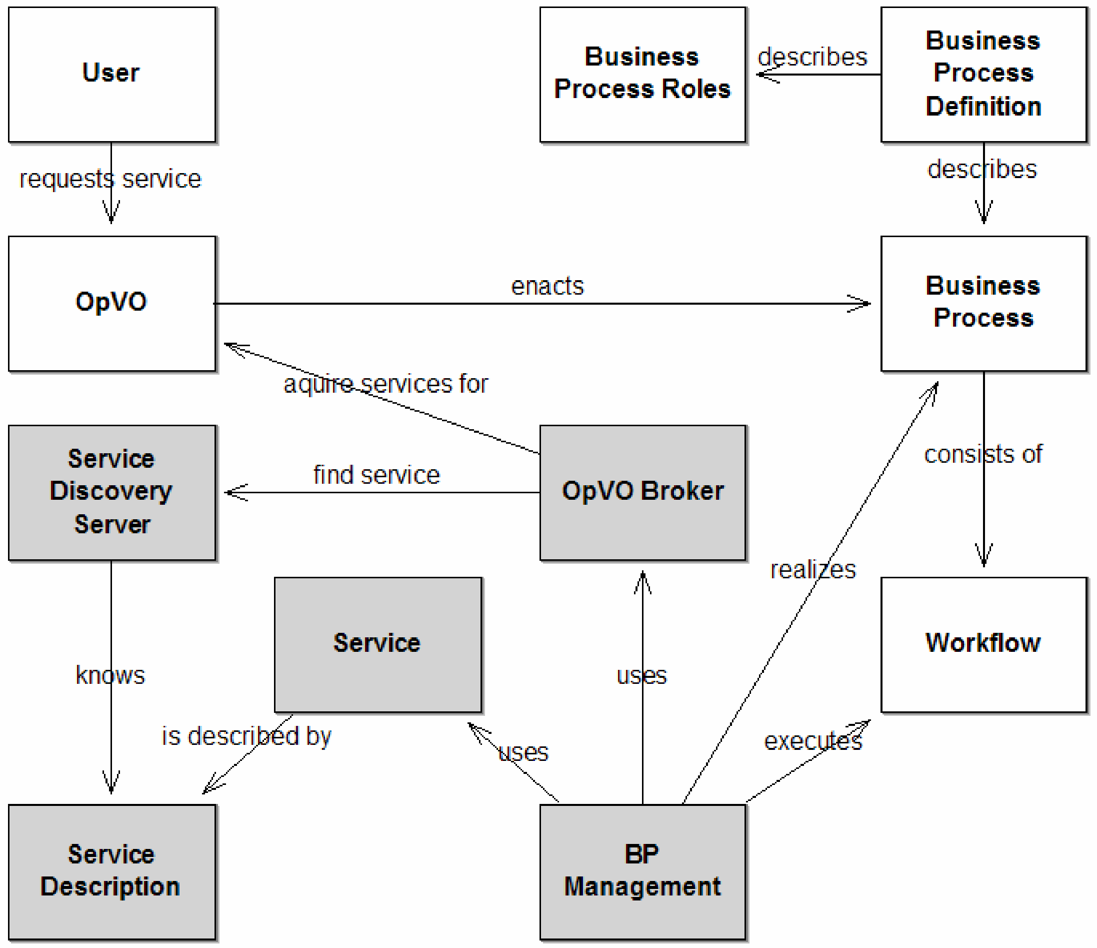


Figure 6 – The Akogrimo Operational VO – Components and concepts

4.7. Quality of Service

In an environment with mobile users, ensuring quality of service is a complex task. Not only the hosting environment of a service has to be managed but also the network through which the user accesses the service. Different access technologies have different characteristic with respect to reliability or bandwidth. Due to mobility even changing the access technology on the fly could be desirable as well as support for disconnected operation. Parameters from the context information of a user are used to adjust QoS. E.g. a video stream could be adjusted to the display resolution of the users' terminal. The display resolution would be part of the user's context.

Both service provider and network operator have means to control and adjust the quality of service. Each of them applies policies to their administrative domain. VO level policies can influence the overall service quality if service provider and network operator support this. Service Level Agreements (SLAs) are used between service provider and service consumer to agree on quality of service parameters. In case of SLA violations local QoS management facilities may apply countermeasures. A service provider can e.g. use load balancing to counter a decrease in response time. If a service violation can not be handled locally it is escalated to the operational VO level. The situation is then handled according to the business process definition.

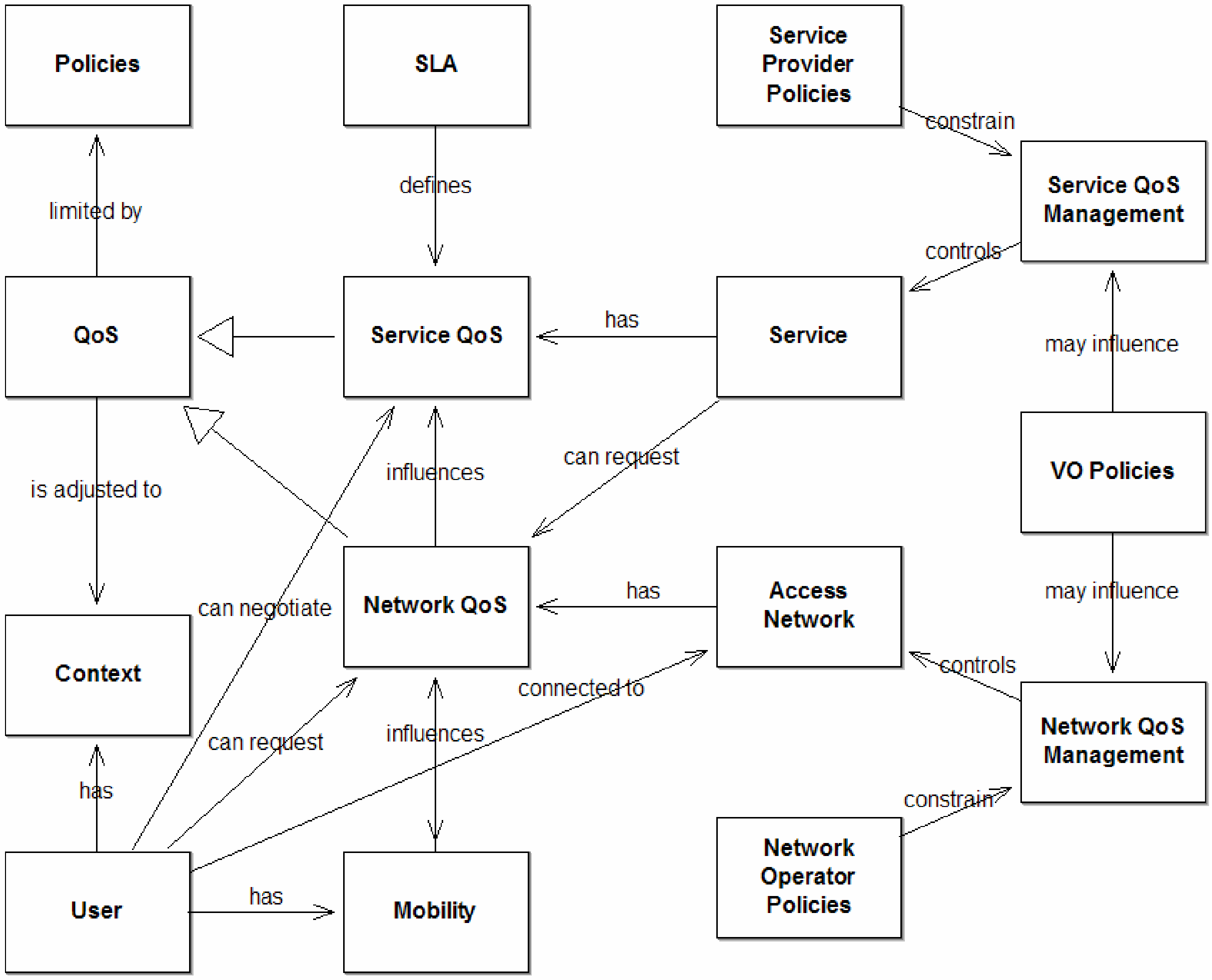


Figure 7 – The Akogrimo Quality of Service concept

4.8. Roles and Identity

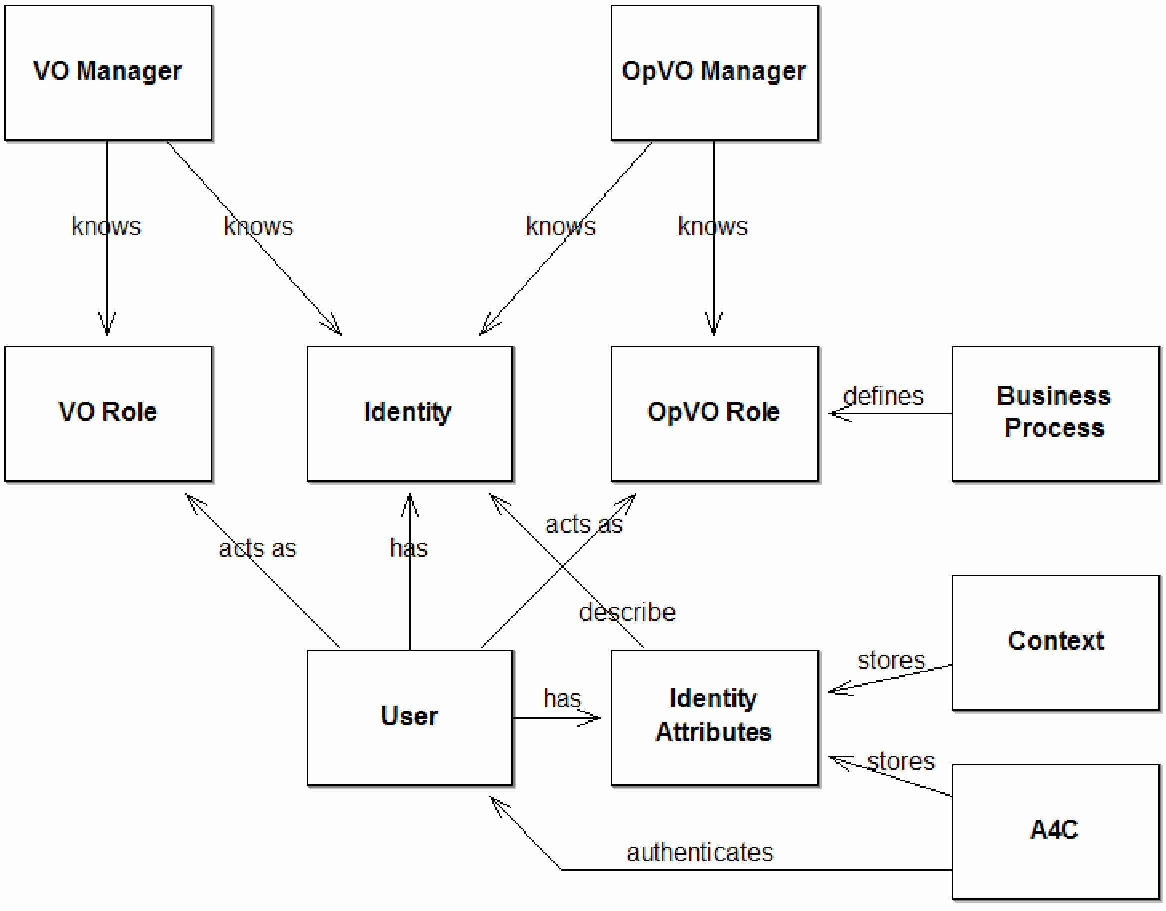


Figure 8 – Roles and Identity in Akogrimo

When users connect to a network they are authenticated by the A4C server. The A4C server stores identity attributes like name, address or employer. The user's context also provides identity attributes. When users act in a VO they also have a role. The mapping of identity to role is done by the VO manager component in the base VO and by the OpVO manager component in the operative VO. The roles that are used in the OpVO correspond to the roles defined in the business process.

4.9. Authentication and Authorisation

Authorisation of service usage can depend on several pieces of information. E.g. a user's identity attributes may include a company identifier, provided by the A4C, and the current location, provided by the context. The authorisation logic might then be to grant access only to users from a specific company and only if they are currently in a specific country. The authorisation logic can also check the user's role and the SLA to apply. E.g. the SLA might state that the user can use the service only 10 times. Also local access policies of the hosting environment and global VO access policies may restrict the use of the service.

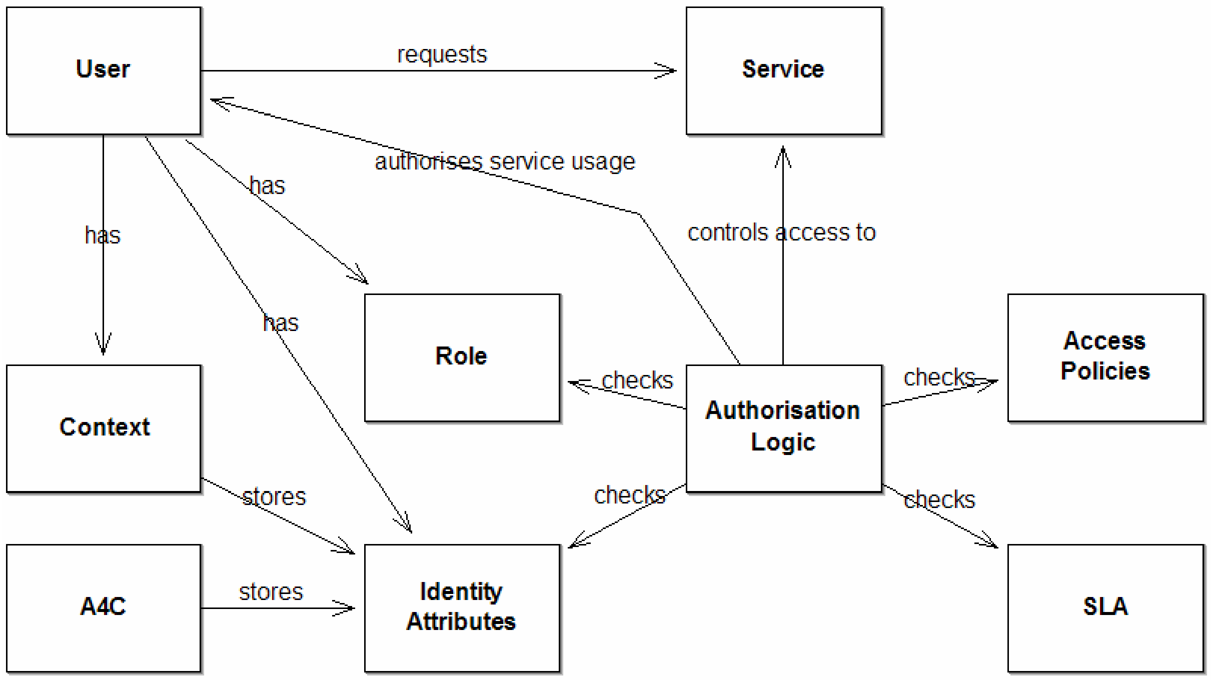


Figure 9 – Service Authorisation

4.10. Accounting and Charging

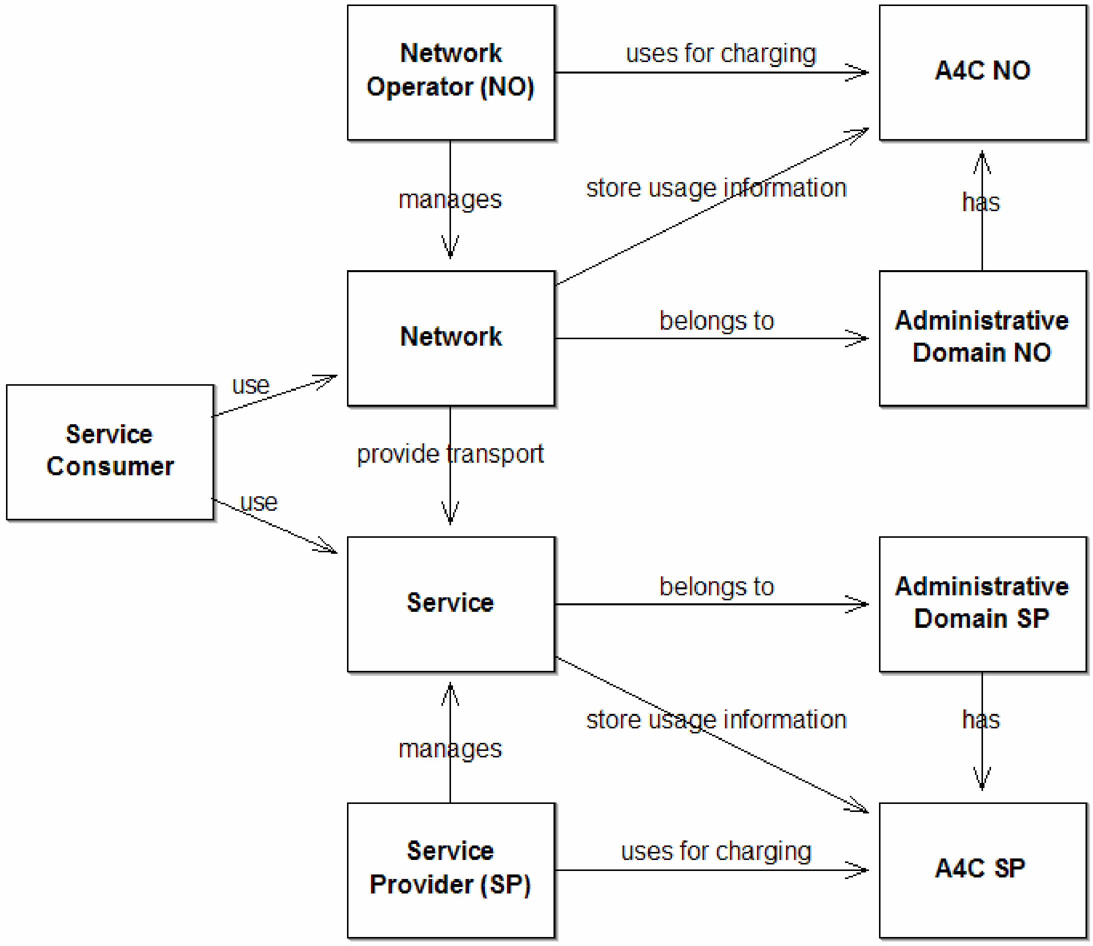


Figure 10 – Accounting and Charging in Akogrimo

Using services of the VO should finally lead to a bill that is presented to the service consumer. Basis for the final charge is the usage of services and data transport across the network. Usage information about network usage is stored in the A4C server of the network operator ("A4C NO"). Analogous usage information about service usage is stored in the A4C server of the service provider ("A4C SP"). Service providers and network operators may control several administrative domains and aggregate information stored in the A4C servers of these domains.

On OpVO level charging information is aggregated from the different administrative domains of the services used in the business process. Also charging information from the network operator's A4C servers is collected in the A4C server of the operative VO. This charging information is used to calculate a price for the services offered by the OpVO.

5. Concepts and Terms

In the previous chapter the basic concepts that build the basis for Akogrimo has been put into a common context. However, not all concepts fit into a conceptual model diagram and need a more text oriented definition. In the following sections these kinds of concepts are explained.

5.1. Enterprise Network

The Akogrimo project is aiming at highly dynamic, "evolutionary" Virtual Organisations that must adopt changes in context and state by dynamically adding/removing services or partners from the Virtual Organisation. This requires ad-hoc discovery of services, negotiation and policy deployment and configuration. This kind of functionality can be only provided if all interfaces to e.g. discovery services und metadata are fully standardized if this scenario should be performed in a complete open market. As we do not believe that this level of standardization is likely to happen or even meaningful at all as metadata should be domain or even application specific we assume for Akogrimo that the Virtual Organisations are built from partners and services offered in an *Enterprise Network*. Accordingly, an Enterprise Network is a consortium of enterprises who have agreed on the interfaces needed to form an Akogrimo-VOs.

5.2. Mobile Dynamic Virtual Organization

We follow in Akogrimo a definition of Virtual Organisation which is more close to original definition in economics as defined in The term 'Virtual Organization' (VO) as described in [12][13][14] is an organizational model describing the rules of interaction between companies not limited to IT resources. Combining this with the definition of the Grid community ([15]), and the requirements from business to business interactions in [16][17] Virtual Organisation is defined as:

> *A Virtual Organization (VO) is understood as a temporary or permanent coalition of geographically dispersed individuals, groups, organizational units or entire organizations that pool resources, capabilities and information to achieve common objectives. Virtual Organizations can provide services and thus participate as a single entity in the formation of further Virtual Organizations. This enables the creation of recursive structures with multiple layers of "virtual" value-added service providers.*

The Akogrimo project extends/change this definition to:

> *A Mobile Dynamic Virtual Organization (MDVO) is a temporary or permanent coalition of geographically dispersed potentially mobile individuals, groups, organizational units or entire organizations that pool resources, capabilities and information, selected from the resources of an Enterprise Network, to contribute to the VO according to the dynamically established contracts typically driven by one ore more business processes.*
>
> *Virtual Organizations can provide services and thus participate as a single entity in the formation of further Virtual Organizations. This enables the creation of recursive structures with multiple layers of "virtual" value-added service providers.*

5.3. Refined definition of a Mobile Grid

For the definition of a Mobile Grid it is necessary to define first the term Grid. As the meaning of this term has no universally accepted definition and is interpreted in different ways for this article we follow the definition proposed in [18]

> *A Grid provides an abstraction for resource sharing and collaboration action across multiple administrative domains...*

Applying this definition on Mobile Grids with the obvious additional element of Mobility where a resource is not bound to a certain location and/or to a certain device the definition of a mobile Grid is consequently

> *A mobile Grid is a Grid with at least one Mobile Grid Resource. A Mobile Grid resource is actively participating in the Grid so that the resource can take the role of either a service consumer or a service provider. A Mobile Grid resource is an active member of at least one Virtual Organization and is involved actively in executing workflows.*

5.4. Administrative Domain

Within Akogrimo the notion of administrative domain is twofold. First, there are different service providers (e.g. network operators) offering their services on the same conceptual level. And of course there are different service providers offering services on different conceptual levels.

A handover of a mobile terminal from one network to another one is first a technological handover, which in the simplest case, occurs within an administrative domain. However, this kind of handover might occur as well if an operator changes "regional" boundaries which would then result in a handover between administrative domains i.e. switching from a Telefonica to a Portugal Telecom network while crossing the Spanish – Portuguese frontier.

Further, an administrative domain might also be a SAN of other Grid resources as well as pure network content or even parts of a SAN offered by a company. This service provider, content provider of resource provider might operate on their own with or without federations to other operators of the same of different conceptual level.

5.5. Session

The most usual concept of session is associated with the availability of exchange data between two or more entities. In fact, IETF RFC 3261 [22] defines a session like "an exchange of data between an association of participants". Involved parties may have to reach a agreement regarding how this data transfer will be during the session setup process. Data could be transferred more less in a continuous way (as occurs in general for multimedia transactions), or it is possible that there is a period of time in which there is no data transfer at all (i.e. in a eHealth scenario a user may request for a simulation service, which results may be delivered to the user later). When the data transfer finished, session is terminated.

The common characteristic of all of these sessions is the availability for transferring data, with independence of the nature of these data. We can have audio sessions, video sessions, accounting sessions (to exchange accounting data)… From now on, we will reserve the term "data session" or simply "session" for this kind of transactions involving data transfer.

It is possible to make an additional definition, the "Akogrimo user session", which make possible to link certain events, actions and service consumptions together. One "Akogrimo user session" begins when the user is registered (authenticated) in the system and (depending on his profile and the existing SLAs) is able to access to the different services that the platform can provide. These services may imply the establishment of different kind of data sessions, so during a "user session" several "data sessions" can take place. For instance, a previously authenticated eLearning user can have access to different kind of services such as multimedia communications or simulation services. When using these services, the corresponding sessions (data sessions) may be established. When user is logged out from the platform, user session finishes. This definition imposes some restrictions to the accounting subsystem, which would need that all services make use of the user session identifier in order to relate all consumed services to this user session.

Numerous protocols have been authored that carry various forms of real-time multimedia session data such as voice, video, or text messages. The Session Initiation Protocol (SIP) [21] works in concert with these protocols by enabling Internet endpoints (called user agents) to discover one another and to agree on a characterization of a session they would like to share. For locating prospective session participants, and for other functions, SIP enables the creation of an infrastructure of network hosts (called proxy servers) to which user agents can send registrations, invitations to sessions, and other requests. SIP is an agile, general-purpose tool for creating, modifying, and terminating sessions that works independently of underlying transport protocols and without dependency on the type of session that is being established

5.6. Mobility Management with SIP

SIP provides its own mechanisms to manage the mobility, such as the following:

- **Device Mobility**: Users capacity to move themselves with their devices through different networks, without available services change.
- **Personnel Mobility**: Users capacity to send and receive calls and access subscribed telecommunication services, regardless neither device nor place where they access. Besides, is the network skill to identify user when he is moving. This capacity is based on use single personnel identity.
- **Session Mobility:** Session Mobility is the seamless transfer of media of an ongoing communication session from one device to another.

Session mobility involves both transfer and retrieval of an active session. Transfer means to move the session on the current device to one or more other devices. Retrieval means to remotely transfer a session currently on another device to the local device. This may mean to return a session to the device on which it had originally been before it was transferred to another device. For example, after discovering a large video monitor, a user transfers the video output stream to that device. When he walks away, he returns the stream to his mobile device for continued communication. One may also retrieve a session to a device that had not previously carried it. For example, a participant in an audio call on his IP phone may leave his office in the middle of the call and transfer the call to the mobile device as he is running out the door.

Session media may either be transferred completely to a single device or be split across multiple devices. For instance, a user may only wish to transfer the video of his session while maintaining the audio on his PDA. Alternatively, he may find separate video and audio devices and wish to transfer one media service to each. Furthermore, even the two directions of a full-duplex session may be split across devices. For example, a PDA display may be too small for a good view of the other call participant, so the user may transfer video output to a projector and continue to use the PDA camera.

Two different methods are proposed in order to perform one transfer: third-party call control (SIP External Entity SIP-EE) and the REFER method:

- **Session Mobility with SIP External Entity**: In this case the SIP-EE establishes separate SIP sessions with both devices (Source Device and Target Device), but a media stream is established between both devices. The shortcoming of this approach is that it requires the SIP-EE to remain active to maintain the sessions.
- **Session Mobility with REFER Method**: A user may need to transfer a session completely because the battery on his mobile device is running out. Alternatively, the user of a stationary device who leaves the area and wishes to transfer the session to his mobile device, he will not want the session to remain on the stationary device when he is away, since it will allow others to easily tamper with his call. In such case, Session Handoff mode, which completely transfers the session signalling and media to another device, is useful.

5.6.1. Device Mobility

The following figure shows how SIP handles the device mobility before to establish the call between both hosts. The case in which the user has active sessions is described in section 5.6.3.

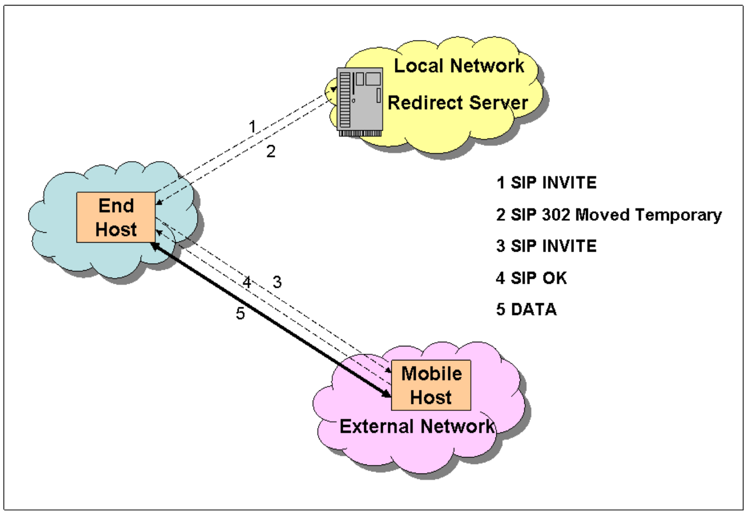


Figure 11: Device Mobility before call establishment

In this case, the End Host (End User) wishes to establish a communication with other host (Mobile Host) which is placed into an external network

When the End Host sends the message SIP INVITE to its Local Network to establish the communication with the Mobile Host, the Redirect Server of the Local Network knows the current position of the Mobile Host. So, the Redirect Server delivers to End Host the current position in order to it redirects the INVITE request to suitable address. After this is established the data interchange between both hosts.

5.6.2. Personnel Mobility

The personnel mobility is the capacity to locate user, regardless his position as well as the device which is using. So, the user can reach a service, as from PC, PDA as from Mobile phone, regardless access network (local network or visiting network).

SIP plays a very important role about personnel mobility, because SIP translates to *single overall address* all user connection address (for example, name@terra.es, 91555 55 55, name@tid.es...).

Thus, the services are not bound to own identifier of the device but *single overall address* (as showsthe following figure).

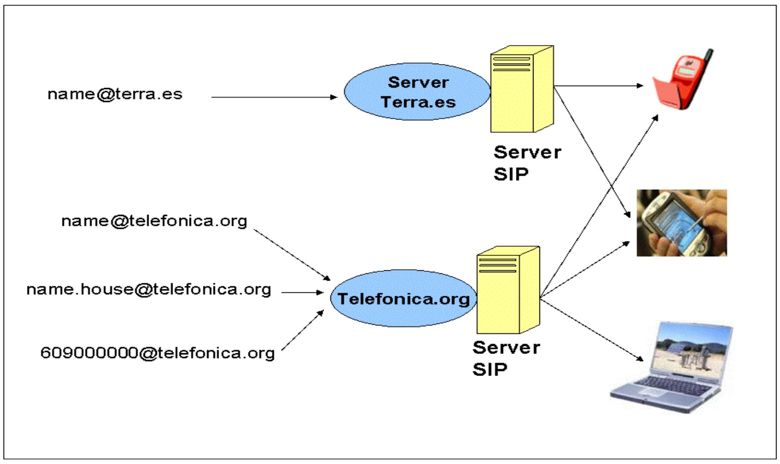


Figure 12: Personnel Mobility with SIP

The figure shows as End User can be identified as several names and different domains. This is possible because, when the connection address reaches the SIP Server, this translates it to *single overall address.*

The translation mechanisms of the SIP servers are basically mapping mechanisms, which use personnel databases with different number plans that let recognize different address as the same.

Any device (PC, PDA or Mobile Phone) can access to these databases.

5.6.3. Session Mobility

The following figure shows the message flow for transferring a session to a single device. It follows Third Party Call Control Flow (SIP-External Entity) which is recommended as long as the endpoints will immediately answer. However, the SDP content here differs somewhat from that flow.

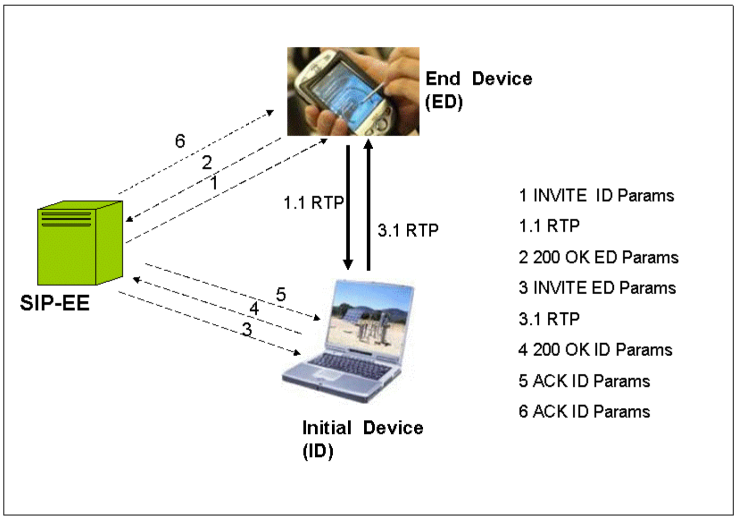


Figure 13: SIP Session Mobility. Transfer to a Single Device

The SIP-EE sends a SIP INVITE request (1) to the End Device (ED) used for the transfer, requesting that a new session be established. If the ED receives an empty SDP body in message (1), it will be unaware of the new sender, and will not play the content of the RTP packets with the new SSRC, until it receives message (6). During this lapse, not only will media not be played on any device, but the media segment sent will be lost completely.

The ED response message (2) contains an SDP body that includes the address and ports it will use for any media. The SIP-EE updates the session with the Initial Device (ID) by sending an INVITE message (re-INVITE) containing the local device media parameters in the SDP body.

The ID sends a 200 OK response (4), and includes in its body, the media parameters that it will use, which may or may not be the same as the ones used in the existing session. The SIP-EE sends an ACK message (6) to the End Device, which includes these parameters in the body if they have changed. The SIP-EE has established separate SIP session with the Initial Device and the End device, but a media flow has been established between both devices.

In order to split the session across multiple devices, the SIP-EE establishes a new session with each local device through a separate INVITE request and updates the existing session with the ID with an SDP body that combines the media parameters it receives in their responses. For instance, in order to transfer an audio and video call to two devices, it creates an audio session with one device and a video session with another.

The SIP-EE may later retrieve the session by sending a re-INVITE to the ID with its own media parameters, causing the media streams to return. It then sends a BYE message to each local device to terminate the session.

The following figure shows the session transfer when is used the SIP REFER method. REFER message is a request sent by a "referrer" to a "referee," which "refers" it to another URI, the "refer target", which may be a SIP URI to be contacted with an INVITE or other request, or a non-SIP URI, such as a web page. This URI is specified in the "Refer-To" header of the REFER message.

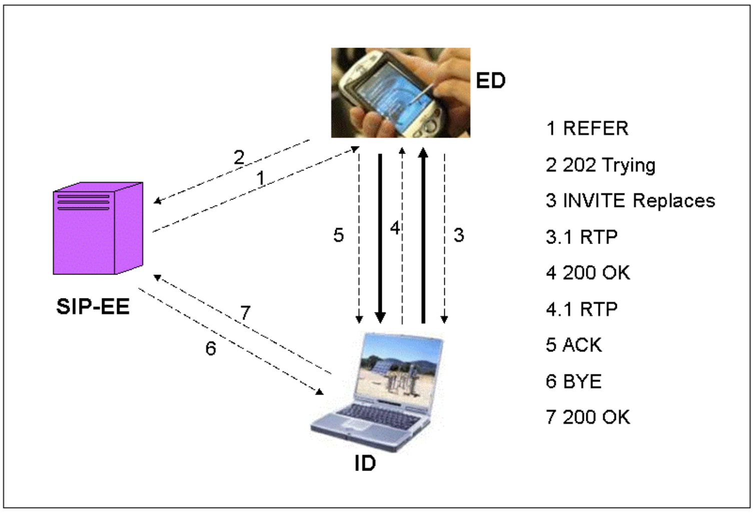


Figure 14 Session Transfer with REFER

The SIP-EE sends the REFER request (1) to an End Device(ED). This message refers the ED to invite the refer target, the Initial Device (ID), into a session. The "audio" and "video" tokens following the URI are caller capabilities. Here they are used to inform the referee that it should initiate an audio and video session with the ID. Is also included the "Replaces" header which is to be included in the INVITE request. The "Replaces" header identifies an existing session that should be replaced by the new session. Here, the End Device requests that the Initial Device replaces its current session with the SIP-EE with the new session.

In order for a device to retrieve a session in Session Handoff mode, it must initiate a session with the ID that replaces the ID existing session.

The following figure shows one scenario where one running session user wishes to transfer the session to other device (for instance, from his mobile phone to his PDA)

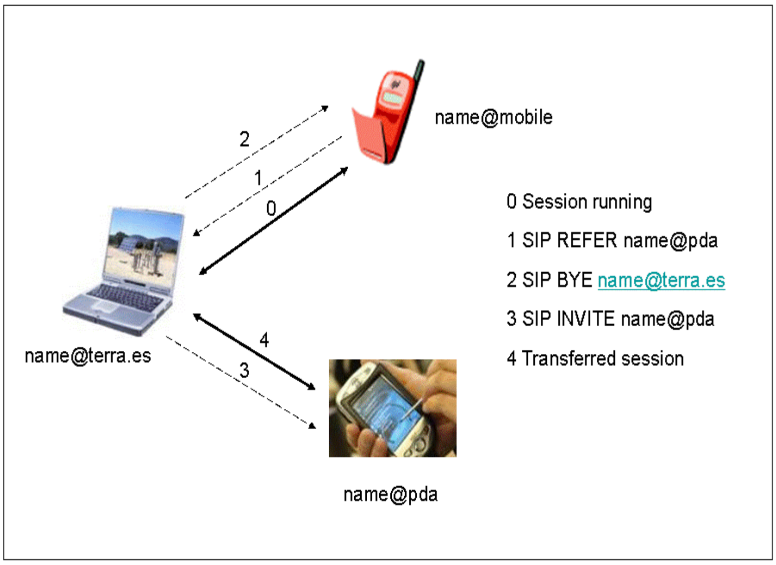


Figure 15 Session Transfer with SIP

At the beginning, the user named (name@mobile) is keeping one session with user (name@terra.es), and then he decides to transfer the session to his PDA, therefore, he sends a control message (SIP REFER) to end user (nombre@terra.es) which contains the new address (name@pda) where he wants to continue the session. When user (nombre@terra.es) gets this new address, sends an INVITE message to new device (PDA), and disconnect the old session with the first device (name@mobile). Finally is established the session between user (name@terra.es) and PDA device (name@pda).

5.7. Context and Device Capabilities

The relevance of the context awareness is closed to the generalisation of the mobility. In parallel with the increasing of the popularity of the mobile devices, the importance of the usability and functionality is increasing as well. Within this mobile environment, we have encountered that the optimal functioning of an application from the user point of view depends on a variety of circumstances: some information related to the user himself, like his/her identity, preferences or physical location; some properties of the physical location itself - temperature, light, noise…; the availability of resources that the application can made use of, like device capabilities, free usable nearby devices (printers) or different network access points…All of these information sources that can have an influence on the optimal functioning of an application can be grouped into the term “context”.

Context awareness means that devices, applications and systems have information about the circumstances under which they are operating and can react accordingly. A context-aware service is more flexible, adaptable and autonomous so as to respond accordingly to the highly changing computing environments such as location, terminal size, or network features without disturbing end user. For example, a context aware mobile phone can detect that the user is in a meeting and activate the vibration mode instead of the ring; a videoconference application can switch seamlessly to a new wireless access point with higher bandwidth availability, or to a nearby device with better screen resolution. So context awareness is a crucial aspect of the Akogrimo infrastructure in order to improve the performances of end user applications

The nature of the context sources shows a higher variability, so it is crucial to analyse them in order to find common characteristics that enable the design of context awareness systems. Some kind of context information trends to be more less stable, showing static or quasi-static properties (i.e. this device has a touch screen); however, there is some information that trends to change quickly, especially in mobile environments (i.e. the network access point when he is moving with active sessions). In the last case the context systems have to assure the coherence of the dynamic information in order to avoid inconsistencies. In the other hand, different context sources usually mean alternative context representations: there are important differences between raw context data and processed context information with relation to its possible usage by the context-aware systems. Finally, the context information from different sources is complementary, so it is possible to infer a more complex (high level) information from basic and isolated sources (i.e. it is possible to cross the geographical location of a user with a digital map of the zone to determinate the building or the room in which the user is in).

The general structure of a context awareness system is depicted in the following figure:

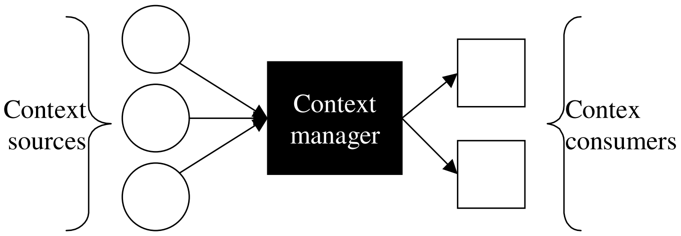


Figure 16 Structure of a Context Awareness System

Context data, obtained from different context sources, are distributed to the interested parties (context consumers) through a mediator entity, the context manager. This element filters relevant data and converts it to a uniform format prior to deliver it to the interested destination. Additionally it has to solve the possible inconsistencies and infer high level context data from basic context information.

The context source is the user device which provides context information, or any sensor connected to own device.

Some context information consumers in Akogrimo are the following: Call handler (SIP), Service Logic, Service Support (A4C, SLA…)

5.8. Identity

The term of identity refers to "the collection of characteristics by which an entity is recognized or known". An identity is always bound to an entity. However, identity is an abstract term since the collection of characteristics by which an entity can be described with, may be seen from different points of view, and they will lead in different sets of descriptive attributes. For instance, types of characteristics could be physical, emotional, biological, structural…. When the identity is used, the set of characteristics is restricted to a finite subset and the election of which characteristics to use depend on the application where the identity is going to be used.

In a digital world, the concept of digital identity refers to the finite set of characteristics (digital information) that represent the entity that is used in a distributed network interaction with other entity.

Identity in Akogrimo

In Akogrimo, the users are represented with multiple identities along the networks.
The user identity is the set of information that is attributable about the person.
A study from Andre Durand, published in the DigitalID Magazine, defines the user identity as a fourth layer concept. Also, the Daidalos project has also used the distinction of the four layers in order to develop its identity model:

- The first layer refers to the physical identity, such as DNA or fingerprint.
- The second layer refers to personal identity, such as mood or preferences. Personal identity is controlled and owned by the end user.
- The third layer is the relational or issued layer, since they exist within the relationship with another entity, such as employee or customer identity. The entity but not the end user controls this identity.
- The fourth layer, called in Daidalos Virtual Identity, represents some parts of the personal and relational identity. This identity is owned by the end user in order to protect its privacy.

The user identity can also be classified regarding on the probability of change. In this way, we could classify the identity as static or dynamic one.

- Static identity: It is referred to attributes that predominant do not change (or they don't do it often). This includes personal data like name, address, date of birth…etc. Regarding the layer classification, physical identity and relational identity tend to behave in a static way.
- Dynamic identity: The set of attributes that are dynamically generated like location or presence of a user. The dynamic identity may also need help from additional external entities who can generate these values and store them. Second layer, namely personal identity is considered dynamic. Additionally, because fourth layer is partially third and fourth layer, can be considered of dynamic and static nature.

In Akogrimo, the end user uses his identity in order to communicate with other entities of the Akogrimo environment. The Akogrimo project will use the third and fourth layers of identity depending on the entities it is communicating with and on the purpose of the communication.

Akogrimo uses static and also dynamic information of the user, which contributes to provide personalized information without mobility constraints.

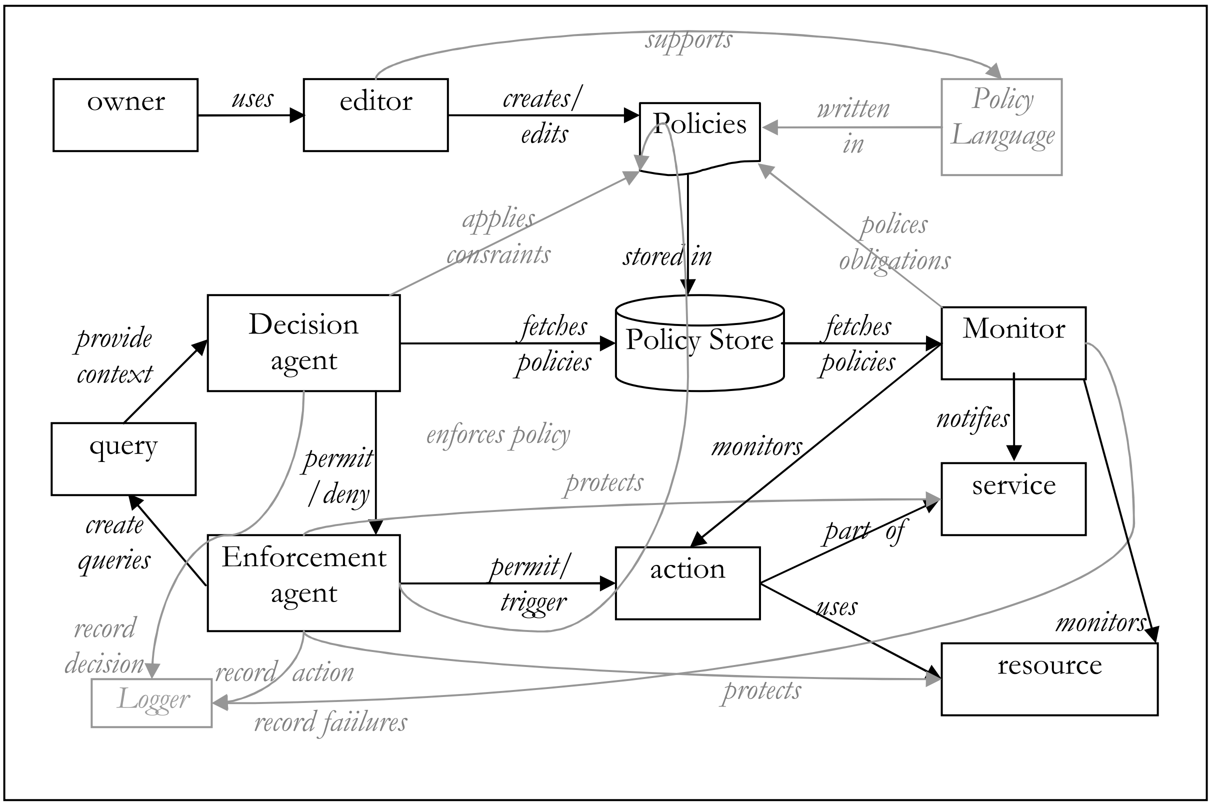
5.9. Policy Framework

There are many points within Akogrimo where decisions have to be taken by the system (Note, these may be implicit decisions, such as whether to carry out an action as requested or not.) At these decision points, guidance is required to select amongst the possible options. This guidance is provided by policy statements. In many cases, the policy is simple and static, e.g. pick the cheapest/nearest/fastest service. However, in some cases the policy is complex and context dependent. This implies the need for a specific policy "document" in which the appropriate criteria and constraints can be provided, and of course a policy language(s) in which to express the policy. Since policies need to be generated by the users and delivered to the decision point, a policy framework is needed to provide policy storage and management.

In theory each significant entity in the architecture has potentially two policies attached to it – what it can do and what others can do to it. In practice, owners responsible for a resource or community will generally only be interested in controlling how and by whom it is used. The exception is the SLA manager, who clearly also has an interesting what their service does.

In Akogrimo, the key policy types are – security, VO membership, privacy, resource usage, service usage, service delivery, etc. The scope of these policies will vary, either applying across the VO, just to a specific operational VO, to a specific resource or set of resources, etc. It should be noted that it is almost inevitable that there will be conflicts between the policies for different entities and at different scopes for the same entities.

The components of the Policy Framework are policy owner, policy description, policy editor/generator, policy store, policy-based decision agent, policy enforcement agent, the entity-action pair to which the policy is being applied and a action/resource monitor to manage obligation failures. Note, agent here is not necessarily a true software agent – it may just be an "if" statement in the interface to the entity.



Each policy has an owner, usually the owner of an entity, whose interests are being protected by the policy. (The owner may be a consortium, as in the case of a VO). The owner required a way to create policies. Normally, this is done using an editor (general text editor or specific policy editors). However, as in the case of SLAs, the policy being applied may have been generated automatically by a negotiation process.

Each Policy will require its own Policy language to describe the constraints and obligations to be applied. The Languages will be specific to the policy decision/enforcement agents, for example SAML [55] or XACML [56] for security policies, WS-Agreement [57]for SLAs, etc.

A policy store is required to hold and make available the various policies. The store must be accessible to the editor and decision agent. While it would be possible to have a single global policy store, it is more likely that policy owners would want their policy store to be local and specific to them. On the other hand, it is in the interest of the owner to have the policy known externally, so resource/time isn't wasted requesting actions that will be rejected as a matter of policy. In Akogrimo, several policy stores are envisaged:

- VO policy store – owned by the consortium and holding policies that apply to VO members (or the VO itself).
- VHE/HE store – owned by the owner of a resource or set of resources and holding resource specific policies, such as access criteria or SLA requirements. This may also apply to services, as these typically run on a resource which they have to trust (to execute them if nothing else). The alternatives for the service are to build the policy into the service (a very common mechanism) or hold the policy in a remote store (under the control of the owner).

Policies need to be applied. A distinction is made between the policy enforcement agent that stops, permits or triggers an action on a given resource or service and the policy decision agent that takes the current context and requested action and returns a decision on whether the action may be carried out or not. As this is a pure overhead on every action, the system must be designed to make this process as efficient as possible. Furthermore enforcement agents must be as close to the entity they are protecting as possible (preferable encapsulating it) to ensure they cannot be bypassed. It follows that decision points need to co-located with the enforcement points, or placed close (in terms of access time) to them. However, decision agents also need to ensure that the policies they are enforcing are up-to-date. This could be done by fetching the policy from the store at the decision time. This is potentially very inefficient, as the policy will not only have to be fetched but also parsed and transformed so the decision logic can be applied, but might be acceptable for infrequent actions where currency of the policy is critical (members joining a VO where policy constraints vary rapidly – *eg an entity joining an insurance policy backed by volatile assets such as shares, where falling share values may require the refusal of additional risk*). Alternatively, the policy store could push updates to the decision points when changes happen. The problem here is that the decision point would not know if the lack of an update was because there wasn't any, or was due to loss of the communication channel with the store. A compromise would be for the decision agent to poll frequently for updates.

There can of course be multiple policies to be applied to an entity. It is impractical to implement a decision agent that handles different policies, but it should be possible to implement one enforcement agent that queries multiple decision agents.

Obligations can't be enforced. Instead, the actions and resources are monitored and when a violation is uncovered, the appropriate service is notified to take corrective action. Services supporting an action or resource are expected to provide remedial actions for all potential policy violations (which should be possible, as the service provider needs to be aware of the behaviour required of his service by the VO/opVO. If the obligation isn't rectified, the failure is logged and the failure is escalated to higher authority. The service/resource should immediately be isolated, ie the VO/opVO policies changed to prevent the service continuing. In the case of an SLA, a violation is handled by a different service from the failing service, and this effectively rewrites the policy to permit the violation to continue (though penalties may be applied as a consequence).

5.9.1. Offerings

Members of the Universe become member of the Enterprise network by agreement to offer services according to a set of common formats and mechanisms - subject to member specific policies. Policies apply to security, privacy, resources, services, SLAs etc.(to be completed)

5.9.2. Contracts

Contracts define roles of the members of the Enterprise Networks such as x-provider, y-consumer; they define corresponding identities and policies (= (sub)sets of policy rules).

5.9.3. Lifecycle issues - initial deployment (related to VOs)

There must exist a (possibly generic/meta-)policy framework which (after definition) establishes the initial policy framework instance; this requires a policy transport/deployment and (possibly a supporting translating gateway) system - e.g. translating back and forth between a AAA/Diameter, COPS and a WSpolicy world.

5.9.4. Lifecycle issues - context awareness etc. (related to VOs)

During its life a VO has to cope with changes to single policy instances - caused by context changes. What is/are the framework/s to create and deploy these new instances eventually to the PEPs.

5.10. Business Process and Workflow

The service orchestration problem is the problem of modelling, expressing and successively enacting a set of services in order to make them cooperate in a coordinated fashion. The modelling can be done by means of a workflow, graph or structured-block based representation of the intended interactions and synchronizations between specific services. Services can publish their interfaces so that they can be used outside of any orchestration or they can interact by the means of an orchestration language. The enactment process is then carried out by an enactment engine capable of creating an instance of a workflow and successively steering and managing the run-time execution arising from the interaction and synchronization steps between services. To have a common understanding of what is written above we give a definition of some terms like business process, orchestration, workflow, services, etc. These definitions are the base for a common understanding that needs to be contextualised in the Akogrimo framework.

Hereafter we provide a list of the terms that need a common definition:

- Business process
- Orchestration
- Workflow (abstract, concrete)
- Enactment engine
- Service (abstract, concrete)
- Workflow template

5.10.1. Business Process

A set of one or more linked procedures or activities, that collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships.

5.10.2. Orchestration

Orchestration describes how services can interact with each other at the message level, including the business logic and execution order of the interactions. These interactions may span applications and/or organizations, and result in a long-lived, transactional, multi-step process model.

5.10.3. Workflow

The modelling of a business process, during which documents, information, or tasks are passed from one participant to another for action according to a set of procedural rules.

The definition given above is very general. We would like to introduce a distinction between high level workflow and low level workflow, i.e. respectively abstract workflow and concrete workflow. This distinction could help us in understanding why and for what purposes we use workflows.

> ***Abstract Workflow (AW)*** *is the description of an Abstract Services (AS) composition providing semantic information on how the workflow has been composed.*
>
> ***Concrete Workflow (CW)*** *is the description of a Concrete Services (CS) composition, therefore providing semantic and execution information both on the single CSs components and on the overall composition (e.g. dataflow bindings, control flow structures).*

5.10.4. Enactment Engine

This part is responsible for the enactment and execution of the workflow. It needs to instantiate the workflow, initialize the required information and contexts and steer the execution of the workflow (even when faults occur).

5.10.5. Service

> ***Concrete Service (CS)*** *is a Service description providing both* ***semantic*** *and* ***execution*** *information of the service.*
>
> ***Abstract Service (AS)*** *is a Service description providing only* ***semantic*** *information about a service.*

Workflow template

A workflow template contains semantic information about a specific business process and how to enact it. A workflow template should contain information on the abstract services and their relationships in order to enhance the business process.

5.10.6. Workflow manager components

The definitions given above can be used as the basis for a general orchestration architecture. These general concepts will be used to define a workflow manager component in the scope of the Akogrimo project.

The workflow manager for the Akogrimo project is in charge of the following operations:

1. Finding the most appropriate workflow
2. Monitoring the overall Services execution (handling also SLA violations and status failures)
3. Managing context changeStoring Workflow templates
5. Requesting OpVO creation/destruction

Before describing these operations in more detail, we will present the sub-components of the workflow manager and their use.

The workflow manager component is responsible for providing and managing the following sub-components:

- Workflow repository
- Workflow manager
- Workflow enactment engine

The business process (that could be identified with the application) is mapped to multiple workflows by the business process designer using a specific language. These workflows are stored in the Workflow Repository and will be used by the workflow manager at run-time.

We have decided to map the Business Process to more than one workflow for practical reasons. It is not worthwhile to instantiate one unique workflow that may require a long time to execute. In addition it could require use of a significant number of services/resources, but it is not reasonable to reserve the services in advance a long time before their use.

To summarise: we have chosen to have several workflows to leave the possibility of more flexible instantiation and execution of the business process.

The workflows will be instantiated by the workflow manager and executed by the workflow enactment engine. The workflow manager is the component that maintains the intelligence of the entire instantiation of the business process: it chooses which workflow template best fits the requirements; it decides when to instantiate the workflow; it communicates with the other components; and it maintains the logical links between the execution of different workflows (related to the same business process).

We want now to explain in detail the steps highlighted above.

1. *Find the most appropriate workflow*

The workflow manager is in charge of selecting the workflow template (from the workflow repository) that is the best match for the high-level requirements. This means that the workflow manager is responsible for interpreting the business process as defined by the business process designer and for selecting the best workflow template (related to the specific business process).

2. *Monitor the overall Services execution*

The workflow manager is responsible for retrieving information about the status of the services involved in the business process. This means that the workflow manager will subscribe to the SLA enforcement and will take into account any SLA violation.

The workflow manager will be responsible for selecting an alternative solution in the case of SLA violation or possible failures.

3. *Manage context change*

In the AkoGriMo project mobility is one of the key aspects. Mobility means that the context (of the user/service) could change during the execution of the Business Process. Every change in the context should be taken into account by the workflow manager. In fact, it is very important to optimise the workflow execution and to make it flexible. The workflow manager will subscribe to the Context Manager and it will be ready to handle any context change.

4. *Store Workflow templates*

The workflow manager is responsible for storing workflow templates in the Workflow Repository. It is responsible also for maintenance of the Workflow Repository.

5. *Request OpVO creation/destruction*

The Workflow Manager is the component that asks to the OpVo Manager to create/destruct the OpVO.

A detailed description of the Workflow Manager component is given in chapter 3.3 of deliverable document D4.4.1.

6. Selected Cross layer issues

6.1. Security

Security in Akogrimo is considered to be developed at the second cycle of Akogrimo project. However, an overview of security needs and design is given as part of the overall architecture definition of Akogrimo.

Security architecture in Akogrimo is based on layered security infrastructure that provides requirements for securing communication among components. Security mechanisms may be applied at three distinct layers:

- Network security: Akogrimo endorses network security architecture on top of each particular access network security in order to homogenously provide a strong minimum security. This will be done by the use of IPsec. IPsec will provide a secure access to the core Akogrimo network by means of encrypted IP tunnels. This is a point-to-point security solution.
- Channel security: Akogrimo provides security by securing the channel where messages are transmitted. Communication protocols at transport layer must be secured. Akogrimo may use SSL and TLS for this purpose. This provides a point-to-point security among services.
- Message security: Akogrimo endorses message security by signing and/or encrypting messages thus providing end-to-end security among services. Akogrimo needs end-to-end security since messages may pass through different intermediaries and they could be not completely trusted. Message security is provided by applying the following services:
    - Authentication: Authentication means the capability of identifying other entities. Both users and services require authentication in a secure environment.
    - Authorization: A decision must be made by referring whether an identity should be granted access for the requested service or not.
    - Message confidentiality: It means that only the intended recipients will be able to determine the contents of the confidential message
    - Message integrity: It refers to the security countermeasures for insuring that a message in transit was not altered.
    - Non repudiation: It is the concept of ensuring that a message cannot later be denied by one of the entities involved (sender and receiver).

  WS-Security provides end-to-end message security and it is used in Akogrimo for securing communication among Grid services when using SOAP messages. WS-Security defines a SOAP Security Header to contain security elements. It includes:

    - Security tokens: Akogrimo uses SAML tokens. Akogrimo components may insert SAML tokens for user authentication and authorization at Grid services.
    - Signature elements: XML-Signature is used to protect SOAP message. XML-Signature provides message integrity, message authentication and non-repudiation.
    - Encryption elements: XML-Encryption is used to encrypt the SOAP message. This provides message confidentiality.

Security in Akogrimo supports also trust, as a cross layered issue Trust relationships among partners may use PKI infrastructures for managing and validating public key certificates and Certificate Authorities. WS-Trust or another trust model like Liberty Alliance proposals may be used for establishing secure communications between Grid services, including interactions that involve third-party certification authorities.

6.1.1. Service Compositioning

Services are logical entities that are provided, managed and coordinated within a Virtual Organisation. Services can be simple or aggregated. An aggregated service is a combination of simple services linked together in a *static* way.

Service compositioning is the process by which simple services are aggregated in order to offer added value with respect to a simple service. Service compositioning can be done directly by the Service Provider who offers an aggregated service, or it can be the result of analysis performed by the business process designer

Aggregated services are exposed in the same manner as simple service; they expose a simple interface that hides their complexity. The advantage of aggregated services is that they offer a complex service in a transparent way to the service consumer.

From the point of view of the Akogrimo middleware, aggregated services can be split into simple services, which are then orchestrated by a special component of the architecture, the Workflow Manager. It is possible to say that the Workflow Manager is responsible for *dynamically* linking the aggregated service together.

The service orchestration performed by the Workflow Manager is very important because it takes into account the context in which the services must be executed (mobility, network availability, etc) and handles any SLA violation or fault detection.

6.1.2. Service Description and Service discovery

Service description and discovery offers functionality allowing a service requestor to search for a service offered by a service provider according to service properties. This enables adequate matches between the needs of the user and what the correspondent service provider offers to be made. The design of the service discovery infrastructure implies the definition and specifications of certain elements such as directory registers, service description, registry procedures, service retrieval and service invocation. All these elements are inter-related in order to build an acceptable infrastructure where services information is easy to access and easy to retrieve with an efficient management and proper maintenance. Service discovery is an obvious cross layer issue since it happens at several levels: network, middleware and application layer.

By analyzing the needs of testbeds described in D2.3.1 Test-bed Definition [33], cross layer requirements for service discovery were derived. Based on the set of requirements for service description and discovery, an overall solution for service description and discovery was proposed. An overview of technologies for service discovery can be found in the state-of-the art report (D2.2.1 vol 2).[34]

Ideally the solution would be a unified service description and service discovery system offering an API that hides multiple underlying service discovery systems from the requestor. However, it was realised that very diverse services must be handled by Akogrimo service discovery. Furthermore, to realise testbeds it will be necessary to build on existing protocols and methods for service discovery. The resulting Akogrimo design therefore consists of two decoupled systems for service discovery. The context manager handles discovery of people and “local services” in their proximity while the Service discovery subsystem covers discovery of web services/Grid services.

Traditional service discovery mechanisms only describe the syntax of a service in machine-readable form. The semantics of the service must be defined through some off-line agreement. Akogrimo will go beyond this by using a service discovery mechanism where services are described and located through specifications of service semantics. Service semantics will be expressed through onotologies. Ontologies for new application domains may be added as needed.

6.1.3. VO Management

An earlier section has described the idea of an Enterprise Network (which in turn derives from earlier work in the Trustcom project), which represents an agreement between partners and is mapped to the specific set of Grid functionality which constitutes the Virtual Organisation. In Akogrimo, we further define the Mobile Dynamic Virtual Organisation (again, defined earlier). Generally in this document, we use the simpler term Virtual Organisation to refer to the Mobile Dynamic Virtual Organisation, and VO to refer to MDVO.

In this section we outline some of the concepts associated with VOs.

An important aspect of a VO is that it crosses the existing administrative boundaries of existing conventional organisations (CO). In general, only a subset of the people and services in a conventional organisation are offered to the VO.

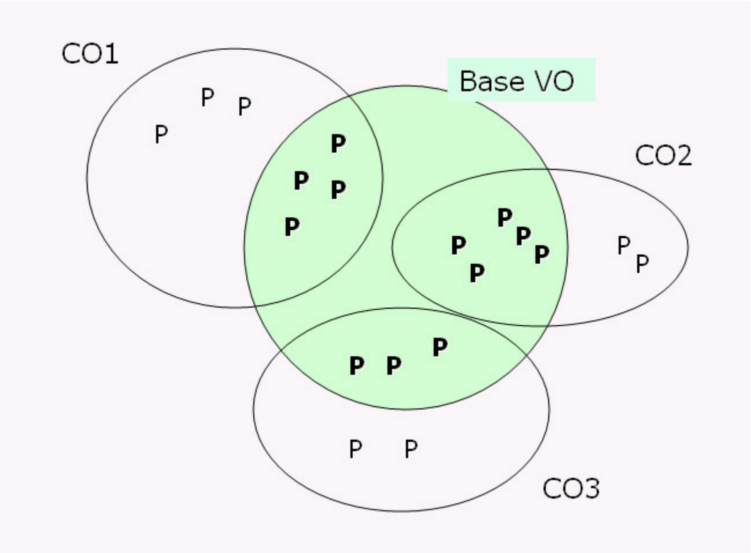


Figure 17 Conventional organisations sharing participating services in a VO

The consequence of a having a VO is that one can preassign members, policies, roles, rights and restrictions in advance. Planned automatic actions can then be initiated on the basis of, any or all of the members satisfying a certain condition.

In the e-Health test bed, the conventional organisations which offer people and services constituting a VO can include: a regional health network, a network operator, a health service provider which provides advanced medical analysis services, a heart monitoring and emergency service (HMES), first responders and police services. A patient may also become associated with a VO.

To enable workflows to be run, we introduce the Operational VO (or OpVO). In fact the purpose of an OpVO is to provide an environment within which is run one specific workflow.

To distinguish from the Operational VO, we use the term Base VO to refer to any other sort of VO – i.e. not associated with a workflow. Both concepts are VOs. The Base VO provides an environment from which Operational VOs can be created. The Base VO is typically a long term VO which may require human to human interaction to negotiate membership and associated rights and restrictions.

Generally it is expected that an OpVO can be set up quickly to respond to rapidly unfolding situations. There are several ways in which this might be done and the Akogrimo infrastructure should essentially provide mechanisms for this. Some human interaction might be required, but this should be kept to a minimum. For instance, for an emergency situation, a set of doctors who have the necessary knowledge and experience could be identified, but it may be necessary to ask them in turn, are you available for this specific responsibility. If their mobile is not answered, they are considered to be unavailable and the next in the set is called. Alternatively an indication of presence could be used.

In the e-Health case, there are several possible VOs: the HMES; a VO corresponding to each patient being monitored; or an emergency VO for a specific patient, which may require different experts when an emergency is declared.

There are several concepts relating to a VO.

- The person who initiated a VO and set its initial VO policy is regarded as the owner
- A VO has members – individual and organisations. We may also refer to them as participants.
- A member fulfils a specific role in a VO. A role determines the possible rights, responsibilities and restrictions of a member fulfilling that role. A role may have an application-specific name (e.g. heart monitoring analysis expert) which is defined in terms of generic capabilities in the VO. Some roles are predefined and fixed – e.g. the VO Owner.
- Services within a VO.
- A VO policy is a readable, processable statement which determines how certain decisions within the VO are made. Among other things, it could include a definition of the roles in a particular VO. How it works can probably rely on a general policy mechanism.

The VO mechanisms are likely to be provided within the application support layer. They would rely on mechanisms provided by lower layers such as:

- Discovery of services with respect to the VO
- A4C with respect to the VO:
- Registries of entities within the VO which in turn requires replica management

6.1.4. Signalling and Transport of Grid messages: resolution of these different mechanisms in Akogrimo

SIP is the standard protocol for session setup and management and offers potentials for current and future mobile networks with its support of mobility and localization mechanisms. SOAP on the other side supplies the framework for GRID computing and distributed services. This document aims to describe possible combinations of these two protocols and to identify their value for the Akogrimo network

SIP is the de-facto standard protocol for session setup and session management. Furthermore, the SIP architecture provides services like localization and with that, mobility management. In combination with SDP, the session description protocol, it is also possible to provide some context-awareness. The protocol is also simple and extensible to fulfill any future requirements.

SOAP as a higher level application protocol builds the framework for GRID computing as a transport of Web Services and Grid Services.

The goal of Akogrimo is to merge these two worlds, the world of mobility and the world of GRID computing (distributed computing). The usage of SIP and SOAP is a logical step to achieve this. Therefore, this document describes the two protocols in short and presents different possibilities to merge their advantages and to add missing features to each other.

6.1.4.1. SIP Overview

The Session Initiation Protocol (SIP) is already described in detail in several previous internal project documents. This section will only give a short overview about the protocol with a particular focus on the features important for the combination with SOAP.

SIP is a protocol for initiating and managing sessions between multiple entities. The protocol is text-based and similar to HTTP by reusing its message structures and error codes. As SIP does not define a "Type of Session" by default, it is usable for any kind of service needing management of sessions, like audio/video conferencing or interactive games, and/or nomadic actors, like instant messaging.

The SIP architecture provides user localization services via SIP registrars. In combination with the session management capabilities, it provides several levels of mobility; in particular user and session mobility (i.e. nomadic actors, client mobility is better managed with Mobile IP).

But SIP does only provide a "virtual" location, that means, it can provide the current access network of a user/terminal/service, but not its location in the real world (e.g. geo coordinates or "UPM Building B, office B.217").

Context-awareness in the term of terminal capabilities is managed via SDP (Session Description Protocol) which is used during the setup of sessions. If any of the required capabilities can not be provided by the mobile terminal, some re-negotiation has to take place. This could be a setup of a reduced session (e.g. simple audio instead of video communication) or the localization of a more appropriate device (e.g. the big display to view the simulation results) and further session transference to the other device.

6.1.4.2. SOAP Overview

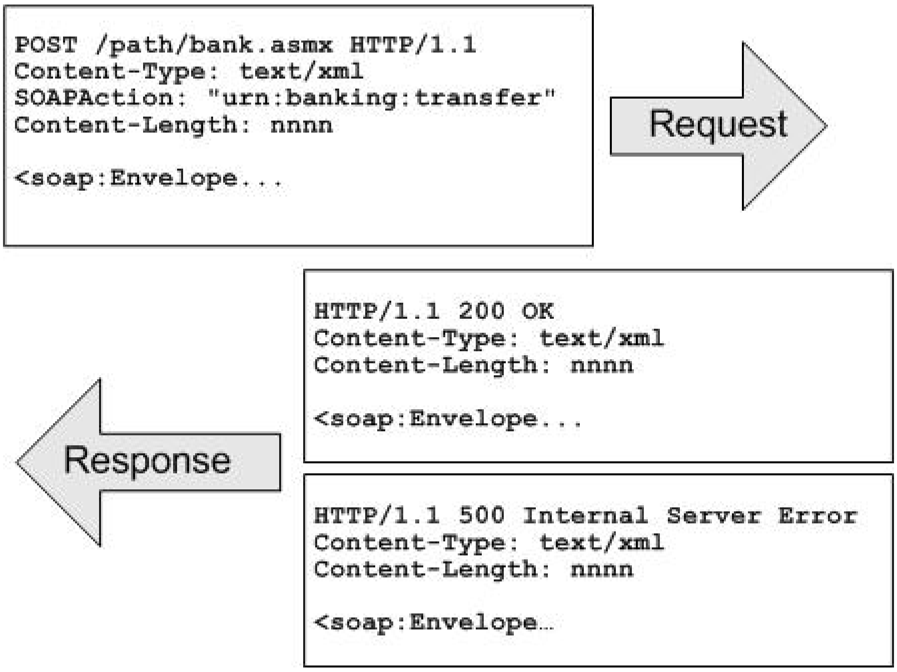
SOAP (formerly an acronym of Simple Object Access Protocol) is a light-weight protocol for exchanging messages. It was first developed to allow mechanism like RPC (CORBA, DCOM) functioning over the internet and independent of the operating system (→ object access). The current version is not limited to that, but the name remains the same. Today's official definition, found in the most recent SOAP 1.2 specification, doesn't even mention objects:

> *SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. SOAP uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics.*

This definition centralizes most of the important features of SOAP. SOAP defines a way to exchange XML encoded messages between services. It does this by providing an XML-based messaging framework that is easy extensible, independent of the underlying network/ transport protocols and independent of the programming model, language or operating system (or at least should be). The utilization of XML allows easy definition of (new) higher level application protocols.

SOAP is not bound to any transport protocol, there are implementations for the usage of TCP, HTTP, SMTP or even MSMQ (Microsoft Message Queuing), but mainly HTTP utilized. The HTTP protocol (see Figure 18) binding defines the rules for using SOAP over HTTP. SOAP request/response maps naturally to the HTTP request/response model.

The content type header for the HTTP messages must be set to text/xml or for SOAP 1.2 application/soap+xml) as defined in [RFC3902].



```
POST /path/bank.asmx HTTP/1.1
Content-Type: text/xml
SOAPAction: "urn:banking:transfer"
Content-Length: nnnn

<soap:Envelope...
```

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: nnnn

<soap:Envelope...
```

```
HTTP/1.1 500 Internal Server Error
Content-Type: text/xml
Content-Length: nnnn

<soap:Envelope…
```

Figure 18: SOAP HTTP Binding

But due to its simplicity and independence of the used transport protocol SOAP lacks support for any security, routing (localization), reliability, etc. It relies on the lower level protocols to provide these features.

Furthermore, SOAP defines a message processing model, RPC encodings and internal fault codes. For the technical description of these features, please see [46].

6.1.4.3. *Combining SIP and SOAP*

In the context of AKOGRIMO, SOAP is used for the transport and control of Grid sessions, as it is used in a traditional Grid scenario, but now actors are mobile and/or nomadic, so the traditional Grid management, designed for fixed service actors, is not adequate for this new scenario. Mobile and nomadic actors are the traditional scope of the protocols Mobile-IP (for mobility) and SIP (for nomadic users and session management). So it is needed to combine the traditional Grid management, based on SOAP, and the mobility and session management, based on SIP. To combine SIP and SOAP, there are several possibilities:

- First, the usage of SOAP and SIP in parallel. Grid applications have to be modified in order to use SIP to achieve the signaling requirements mentioned below, and then use SOAP in the same way it is used in traditional, non mobile GRIDS.
- Second, the utilization of SIP as transport protocol for SOAP messages. Applications are not aware of SIP and therefore, do not need to be modified, but the communications infrastructure has to be modified in order to transport SOAP over SIP and a control mechanism has to be included in order to use SIP.
- There is another possibility, which is using SOAP as transport for a signalling protocol which will include the session management and user mobility. This approach could be achieved by two means:
    - Applications implement an internal signalling protocol in order to manage sessions and mobility, and transport this protocol on top of SOAP, like the mobility handling included in OGSA.
    - Applications will use SIP for session and mobility management and transport it on top of SOAP.

The following sections will describe these approaches with their advantages and disadvantages.

6.1.4.3.1. SOAP for application, SIP for signalling

In this approach, applications use SIP for signaling, i.e. to manage session and mobility information, locate resources, get additional details about availability/status, etc. Once the session is established and both applications (client and server) have all the data needed to perform the service, they use SOAP to exchange service requests and responses, as they would do in a traditional, fixed GRID.

This approach has been successfully used before on similar scenarios involving signaling (telephony, Voice over IP). However, nomadic GRID services and applications must to be changed in order to add SIP-awareness to enable mobility.

Mobile Terminals can be client and/or server (e.g. the doctor in the eHealth scenario, providing his location to the GRID).

The GRID gateway is also mentioned in the general network architecture (e.g. to translate between IPv4 and IPv6).

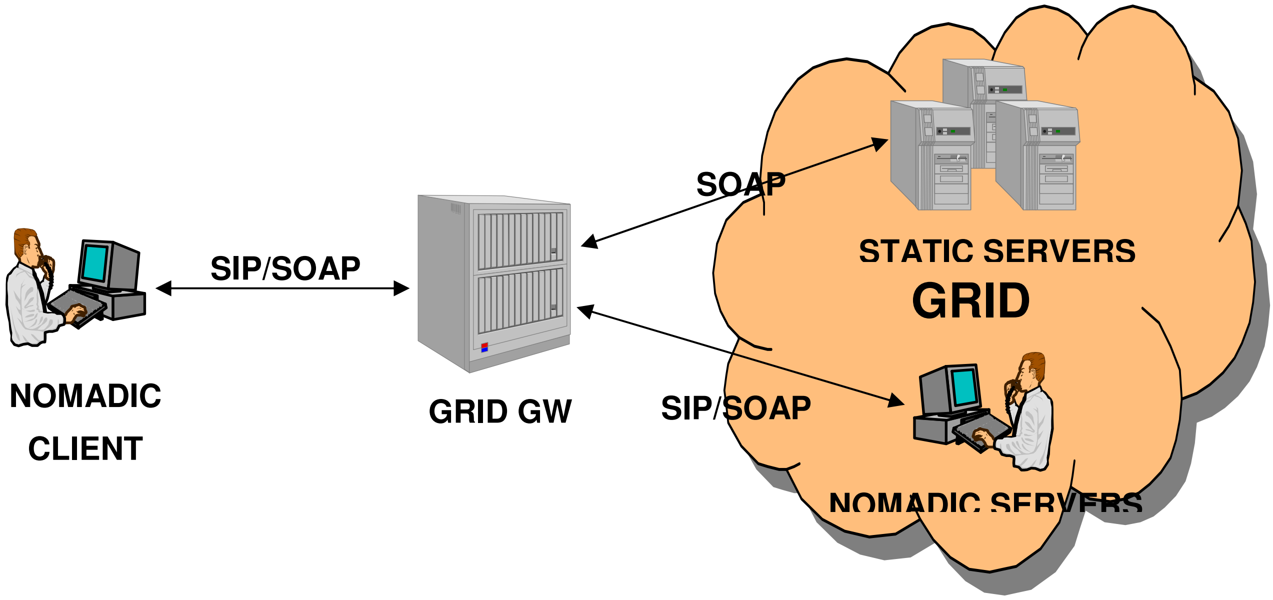


Figure 19: SIP for mobility/session handling, SOAP for application

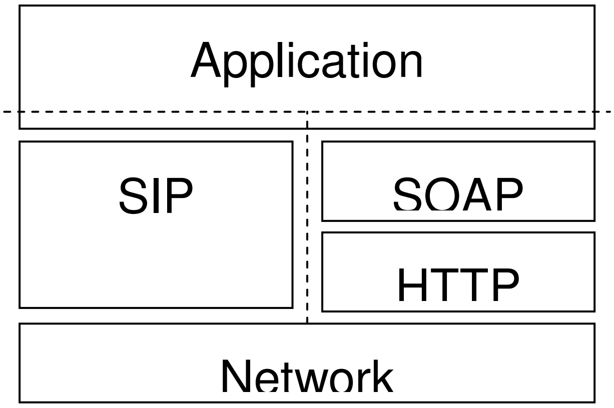


Figure 20: Proposed Protocol Stack

Requirements on SIP for the applications

In order to analyze the impact of SIP on the applications, it is very useful to characterize the different application depending on their role in the service. In this way, we have:

- Client applications could be used by any service actor (client or server), and will only emit requests and wait for answers, i.e. they do not need to listen in order to receive requests or asynchronous notifications
- Server applications could also be used by any service actor (client or server) and do need to listen in order to receive requests and/or asynchronous notifications.

Note that a service client could be composed by client applications (e.g. a web browser or a traditional Grid client) as well as server applications (e.g. to be notified of an available resource or to be provided an asynchronous simulation results). Service servers will be composed by server application and could also include optionally client applications (e.g. a doctor could have a server application in order to be contacted and also client application in order to request some results analysis to the Grid).

With this classification, we can analyse the different demands on SIP from the client part of services/applications and the server part.

Simple client applications will need SIP for a) session management and b) session mobility. For this, the application itself has to be modified in order to:

- understand changes in source and/or destination of a session (i.e. the ability to transfer a session to another node, restoring it afterwards),
- provide further session management like session save/load/restore (may be handled via SIP).
- react to context awareness in the form of terminal capabilities which will be handled via SDP (Session Description Protocol) and SIP.

For mobile and/or nomadic applications that provide some sort of service (i.e. server applications) to the GRID (or other mobile terminals), these requirements have to be extended, so that server applications have to:

- understand changes in source and/or destination of a session (i.e. the ability to transfer a session to another node, restoring it afterwards),
- provide further session management like session save/load/restore (may be handled via SIP),
- react to context awareness in the form of terminal capabilities which will be handled via SDP (Session Description Protocol) and SIP.
- keep their contact/location information updated (in the SIP Registrar), allowing clients to contact them when needed,
- provide additional information, namely presence information (availability, contact means, etc.) to the clients.

Both location and presence can be achieved via SIP. Though, the localization with SIP provides only a virtual location (the current access network of an entity) not a real-world location. Being in the same network does not automatically imply closeness!

Static server applications do not necessarily need to be SIP-aware as long as they don’t have to locate mobile nodes.

6.1.4.3.2. SIP serving as transport protocol for SOAP

The usage of SIP as transport protocol for SOAP messages combines the advantages of SIP with the flexibility of SOAP. This approach automatically adds session management mechanism, localization, security (privacy) handling to SOAP. Furthermore, GRID gains (implicit) mobility-awareness. This awareness is implicit and without extra changes for the GRID services because it is handled by the SIP infrastructure, namely the SIP registrars and SIP gateways.

SOAP messages are sent directly to a user/service via SIP, the localization is done by the SIP architecture, and the services themselves do not have to know about mobile users or their current location.

In this approach, current GRID services can be used with minimal, or even, without changes. However, deeper changes within GRID toolkits are required to use SIP instead of HTTP to transport SOAP messages.

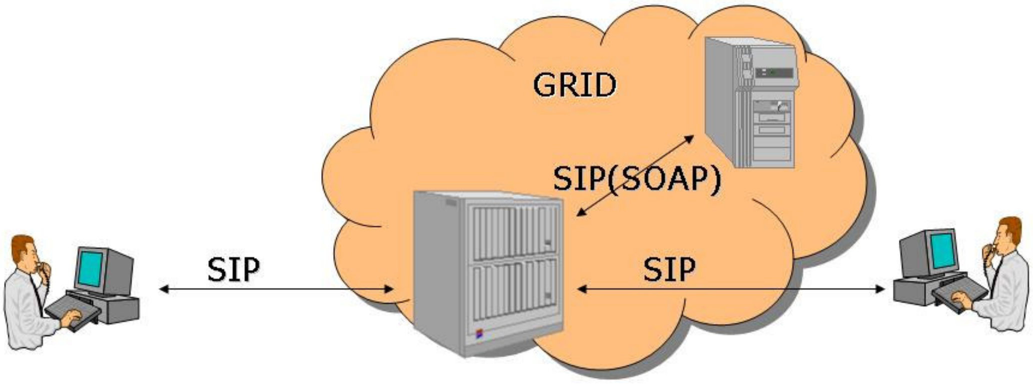


Figure 4: SIP serving as transport protocol for SOAP

6.1.4.3.3. Mobility handling with OGSA

Besides the possibility to use SIP for session mobility and localization, there are already mechanisms within OGSA [15][39] to support these features, even if they are currently named and used differently.

The GIS (GRID Information Service) is based on UDDI (Universal Description, Discovery and Information protocol) [47][48][52]. Its architecture equates more or less the SIP registry architecture.

It can not only be used to describe services (and their technical specifications) but also organizations and persons. Furthermore, it allows changes in location and description of the different entities. This could be used not only to locate services (and mobile services) but also to locate persons.

Additionally, the OGSA requires treatment of migration and relocation of services transparent to the user. The first means the change of location of a service at all; the second means a change of location even during active sessions.

Migration is already handled via GIS and UDDI; this would match user mobility and terminal mobility if these mechanisms would be used for the clients. If relocation is already transparent to the user, this could also be used to provide session mobility for mobile terminals and mobile servers.

6.1.4.4. Conclusion

All approaches have benefits and drawbacks. The first approach is less intrusive with current GRID services architecture and, although the second one adds seamless mobility and session suport to GRID services, the changes needed would likely be much more complex, and would lead to greater (if not unacceptable) development efforts in the used GRID toolkits.

Using SIP as transport protocol for SOAP would include too many changes in the framework of the GRID services. Even if this approach provides some advantages in a simple way, these changes may not be manageable during this project.

The protocol stack as shown in Figure 20 is a more sensitive solution. It will affect only the GRID services that have to be SIP-aware in some way. This includes all services that need the localization functionality provided by SIP and that require session mobility.

6.1.5. A4C (Authentication, Authorization, Accounting, Auditing and Charging)

The A4C component in Akogrimo shall provide a framework for supporting the Mobile Grid with security, accounting and charging services. The framework shall be based on the Diameter [26] protocol, which is the most recent AAA (Authentication, Authorization and Accounting) protocol proposed by IETF (Internet Engineering Task Force). The A4C component shall be based on the architecture specified by the generic AAA architecture proposed in [27]. Although other Grid projects also implement AAA mechanisms based on different and dedicated technologies, Akogrimo's A4C tries to provide a unified AAA solution, based on the mechanisms defined by IETF and enriched with auditing and charging capabilities, which can be used by network services as well as by Grid services.

The following sections will shortly describe key requirements for the main A4C tasks.

6.1.5.1. Authentication

The authentication mechanism defines a process for verifying an entity's identity. The authentication function in Akogrimo shall allow the mutual authentication of users to services and vice versa. Authentication system shall be flexible enough to support in future multiple authentication types:

- knowledge-based authentication – shared secrets, PINs, passwords
- cryptography-based authentication – digital signatures, challenge-response mechanisms
- authentication based on secure tokens
- biometric-data based authentication

An important requirement for the authentication services offered by the A4C component is the need to offer different interfaces according to the individual requirements of Akogrimo components requesting Authentication services (e.g. A Diameter application will be used for network authentication while the VO shall connect to the A4C via a Web-Service that offers authentication service). The scenarios proposed in [33] show that identity verification will be triggered not only by the network layer (as in the traditional mobile networks) but also by some upper layer components (such as the Operational VO Manager).

The heterogeneous environment of a mobile Grid requires a mechanism able to deliver authentication information between different administrative domains. Such a requirement needs a protocol capable of secure data transfer between the two domains.

6.1.5.2. Authorization

Authorization is the process of decision on an entity's allowance to perform a particular action or not. For Grid services authorization is needed also for accessing a resource or service. The vertical issue of authorization mechanisms involves access to physical resources, network services, QoS bundles, Grid services and application resources for enabling collaborative engineering and forming Virtual Organizations.

Akogrimo will use A4C authorization mechanisms for providing access control for network layer services and requesting Grid applications. After a successful request of such an application, all the further authorization decisions for performing the Grid service are taken by the VO manager using Grid-specific authorization mechanisms. Although the VO Manager performs by itself authorization decisions for accessing Grid resources, the A4C system shall be informed about these decisions in order to provide further Auditing services to the upper layers.

6.1.5.3. Accounting

The main task of the accounting system is collecting data on resource consumption from the services via metering components. This data shall be stored in dedicated databases and retrieved whenever requested by a legitimate entity. Such legitimate requests could be:

- A user wanting to see a detailed record of the services he used.
- The charging component wanting the accounting records for a certain user in order to create a charging record and subsequently a bill.

The heterogeneous aspect of Grid services require a very flexible accounting system, capable of storing multiple types of accounting records and being able to support changes in the accounting record format of an existing service. The A4C shall account for the usage of network-related resources (such as bandwidth used, data transferred) as well as for Grid-related resources (such as CPU time, memory consumed, storage size, nr. of started processes, etc.) Every component wanting to use accounting services needs to implement a meter that measures the actual resource usage and provide a bi-directional communication protocol for retrieving the metered data and configure the meter according to the current accounting policies.

6.1.5.4. Auditing

Auditing defines the process of storage and retrieval, when needed, of information on events taking place in the system, history of the service usage, SLA (Service Level Agreement) compliance, and customer charging and tariff schemes applied. The auditing component shall keep track of all events in a sequential order based on their timestamp. All Auditing records shall provide sufficient information for identifying entities involved in an event, the time the event was produced, and the outcome of that event. Its task is to support creation of trust relationships by verifying that the running processes are reasonable. Auditing mechanisms to be used by the A4C are:

- Logging of requests and their approval/denial
- Logging of session status records
- Trusted 3rd party logging for fairness
- Arbitrary checking that appropriate services are running
- Comparing of log entries from cooperating A4C servers

6.1.5.5. Charging

Charging is the task of calculating the price for a given service consumption based on accounting information. Charging maps technical values in monetary units and then applies a previously established contractual agreement between service provider and service consumer upon a tariff. One big influence in the final price paid by the user for the service but also influencing the revenue of a service provider is the charging scheme applied for a service.

Combining the Mobile World with the Grid World brings new challenges for charging the users of a mobile Grid environment. Mobile communications have well defined charging schemes that are in use for a long time and proved to be very successful [26]. These charging schemes have to be extended and adapted in order to support applicable charging schemes for a broader set of services and business models. The A4C component with its charging system shall support the definition and usage of highly-customizable charging schemes and tariff policies for allowing the service providers to model their revenue strategies with as few as possible limitations to the charging schemes they can define.

6.1.6. User Management and Identity Model

The Akogrimo project envisions personalized and identity-based services for network and Grid operators. Akogrimo use case scenarios base its methodology and efficiency on the knowledge of user's identity. How the user's identity knowledge is managed and distributed across Akogrimo components is part of this section. This chapter is based on the definition of identity in Akogrimo, section 4.1.7.

Identity management creates the framework of agreements, standards, and technologies necessary to make identity portable across different administrative domains. Related to identity management, many capabilities are involved. Akogrimo needs identity management to provide the following functionalities:

- Identity federation, account linkage or identity mapping: It is the act of two different providers that agree on a set of identifiers and/or attributes to refer to when communicating about the user. If accounts are federated, SSO can be provided across different administrative domains. In Akogrimo, many providers are working and communicating on behalf of a user. Different Grid services and A4Cs may federate their accounts for SSO purposes.
- Single Sign-On (SSO): It is the capability for the user to access different administrative domains whereas the user performs a single act of authentication. Single sign-on mechanism implies a previously bilateral (direct or indirect) trust relationship among providers. Once the user has authenticated at the identity provider, single sign-on at a specific provider can be performed through a federated account or through anonymous access. In Akogrimo, both types of SSO are possible. SSO for network and service providers like Grid services and A4C is going to be provided.
- Delegation: It is the act of transferring temporarily someone's rights to a third party to behave on someone's behalf. Identity management provides facilities to allow delegation for access rights from requestors to services. In Akogrimo, the A4C may be able to delegate to the mobile terminal or/and to the OpVOManager by providing a set of security tokens in order to act on behalf of the end user. A multiple delegation hop may also be possible. Delegation functionality is directly related to the trust framework and must be specially considered at the design of the trust model.
- Permission-based attribute sharing: This capability supports the act of sharing attributes among different administrative domains. In Akogrimo, it may be used for instance, when a service provider requires specific attributes for access control. The service provider may ask the A4C or other identity provider for some specific attributes from an end user.
- Authorization: This capability refers to the act of checking to see if a user has the proper permission to access to any type of Grid resources. The act of checking is based on predefined policies. XACML may be used in Akogrimo for representing authorization policy in XML. Policy can be represented in terms of roles, attributes of users, Grid services

and/or environment. SAML may be complementary used by providing a mechanism for transferring authentication and authorization decisions.

- Credential Lifespan and Renewal: This capability refers to the act of refreshing credentials prior to the expiration when the initially time scheduled of the credential presented at a service for a specific job is too short and it may take longer.
- Anonymity and pseudonymity support: At identity management systems, privacy of user is a capability that can be fully provided by the support of anonymous access (use of the one-time identifier for SSO purposes) or pseudonymous access. Pseudonym identifiers should be used for federated identities. They are randomly generated identifiers being unique in the context of a specific identity provider and service provider ensuring therefore user's information protection.

But all these proposed functionalities can only be provided with the support of relationships among service providers. Identity management base all its functionalities on a previously established trusted model. Trust models require business and technical agreements. Akogrimo needs a scalable and secure trust model in which dynamic and negotiated trust can be supported. Trust establishments mechanisms may be proposed for direct and indirect trust negotiations. Akogrimo may design its model based on PKI infrastructures and Liberty Alliance trust models.

Additionally to the trust model, services that are part of identity management need to dynamically exchange policy information among them. The policy information is called metadata and is used to describe what other endpoints need to know to interact with them, like capabilities, requirements, concrete network protocols and endpoint addresses. Akogrimo may specify the kind of metadata for identity management purposes (e.g. the SSO soap endpoint of a service) and it may be published within the service description component.

7. Building Blocks and their relations

Here we need to have the identified building blocks of the different layers and how they are interacting.

7.1. The Akogrimo Components

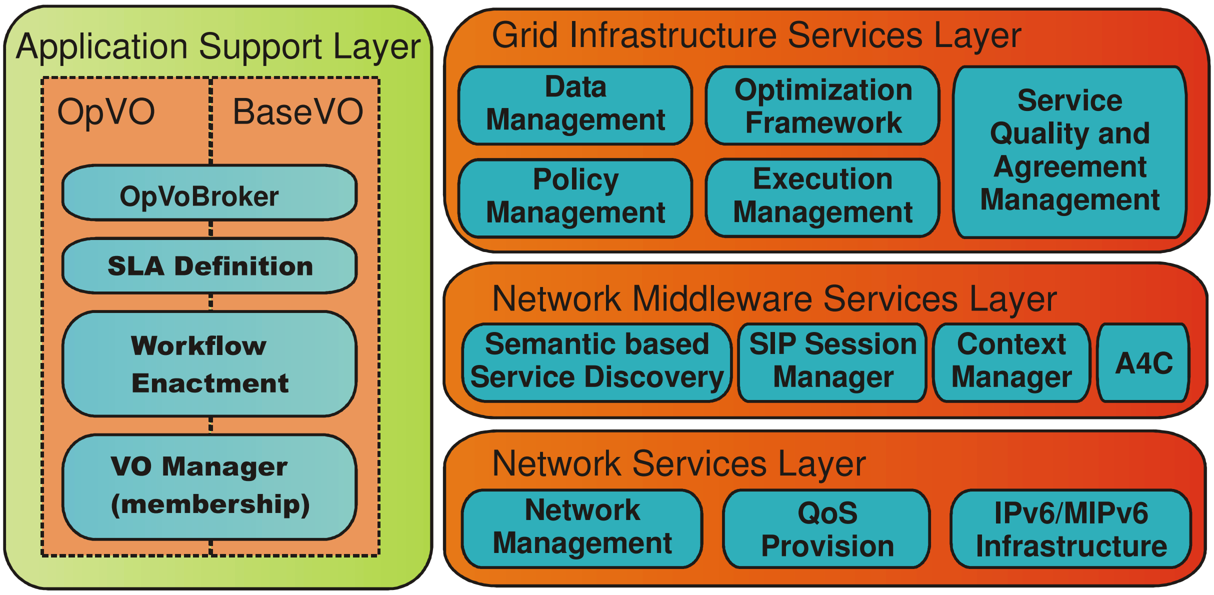


Figure 21 Akogrimo Basic Components

7.2. Network Services Layer

7.2.1. IPv6/MIPv6 infrastructure

Mobility Management

In the Akogrimo network, mobile users will be roaming across different access networks and the network must know, at any time, where a mobile terminal is in order to maintain connectivity. If multiple network providers are present, handovers among them are also be possible, assuming that they have an agreement. Additionally, the network must ensure the authentication of the mobile terminal, communication security, access to network services and communication with other network node.

Akogrimo will focus on wired and wireless LAN with Mobile IPv6 in charge of mobility management. Mobile IPv6 (MIPv6) is a network layer protocol independent of the underlying technologies used in the link layer. It is a simple and scalable solution for terminal mobility, although it has some disadvantages. The MIPv6 protocol has latency problems when handovers are performed, which limits its usefulness if real-time applications (e.g. voice, video) are being used. That shortcoming may be overcome with the use of the Fast Handover protocol.

Mobility management is highly dependent of Network Resource Management. The latter must ensure that handovers between different access networks are possible. If the network is overloaded, the handover will not be successful. Handovers between different network providers require, additionally, interaction with the foreign network provider.

Terminal mobility in Akogrimo will be achieved with the use of a terminal with Mobile IPv6 support and also a Home Agent located in the Core Network whose purpose is to forward traffic directed to the MN while it is away from home, so that it looks as though the MN is virtually at his home network. To this end, the HA must be aware of the MN's current binding.

The mobile terminal has to use an operating system which supports Mobile IPv6. The Akogrimo mobile terminal will use the Linux operating system with the Mobile IPv6 patch.

Interlayer interfaces

The Mobile Terminal's integrated presence module will communicate with the Context Manager (WP4.2) to provide enriched presence information (context information) via a SIP interface.

Network Security

The Akogrimo architecture contemplates the possibility of allowing a VO participant to freely move and actively join its VO independently of its physical position as long as someone can provide him with access to the network. The methods to access the network are not fixed and may be of any nature. A VO participant could be connected, for instance, through the supposedly secure corporation network or by means of his cellular phone using UMTS or even with a public WLAN access point. As we can see, these access technologies are diverse, and the security provided by them is of different nature and strength. In addition, the fact that a user will typically roam and use different access technologies, being this in part transparent to the user, doesn't allow to focus to provide a specially strong security in this layer. For this reason, Akogrimo endorses network security architecture on top of each particular access network security in order to homogenously provide a strong minimum security. To this end IPsec seems to be the most appropriate candidate. IPsec will provide a secure access to the core Akogrimo network by means of encrypted IP tunnels.

Access Router

A VO participant that wants to add a device to his working environment will provide the device with a certain ID and the corresponding authentication material in a way that when the user wants to finish using the device all the user specific data is not compromised and no other user can get access to it. The authentication mechanism launched involves the user's device (MN) and the access point that guards the access to the network (AR). This access authentication procedure is launched through the network independently of the access technology utilized by using the PANA protocol (Protocol for carrying Authentication for Network Access) which works over IP. Once the authentication information arrives to the AR, this information is conveyed to the A4C server, developed in the WP4.2, which works as a back-end authentication server. After the approval of the couple ID-authentication cryptographic material will be exchanged through the A4C structure, the MN and the AR being ready to create an IPsec SA. The A4C may also provide the MN with other cryptographic material such as SAML tokens in order to get him authorized to use determined services which the MN will produce upon the service's authorization request in order to proof its veracity. These tokens describe the actions that can be taken by the ID chosen by the individual and therefore are linked to it. The user may also be able to connect using different IDs or VIDs. The identity concept is considered absolute and unique to the whole Akogrimo structure.

Taking into account that a mobile user may change its point of attachment to the network and the big overhead created by the numerous authentications that take place, which has repercussions on the quality of the transparency of the handovers, mechanisms to provide fast-reauthentication and key exchanges between AR will need to be provided.

Interlayer interfaces

The Access Router will have an interface to the A4C server for authentication of users and also for sending metering information.

7.2.2. Service Provisioning

The Service Provisioning components in the Akogrimo infrastructure are intended to provide an integrated platform to provide network services to higher Grid layers. This includes a signalling framework to support all relevant user/network interactions, so from a practical point of view, this services provisioning will be supported by a SIP infrastructure.

These functional SIP infrastructures should be present in both the mobile terminal and the network side:

- In the mobile terminal side, a SIP User Agent Server and a SIP User Agent Client will be combined with some kind of control logic to orchestrate and coordinate the interoperation with the rest of the terminal components, in order to provide the adequate SIP session control required in Akogrimo. All these SIP components will be grouped in the **MT SIP module**, which will integrate also an enriched presence module (Presence User Agent and Watcher) to manage SIP based context information (WP42).
- In the network side, a SIP registrar server and a Location Database per domain will manage the logical location of the users. These entities will cooperate with one or more SIP proxy servers, which primarily will play the role of routing the requests to another entity closer to the final user. At a first stage Akogrimo will address the single domain case, so all these SIP entities in the network side will be grouped in a single node, the **SIP Server** (which contains a SIP registrar, a Location Database and a SIP proxy). It will include also a SIP Presence Server to manage SIP based context information (WP42), and will implement the external interfaces with the rest of the Akogrimo components. It could be possible to have AN SIP proxies if there are many ANs to reduce the traffic/load of the core SIP Server when the proxy does not have to request something from the SIP Server and is able to process the message directly (e.g. ending a session).

Mobile Terminal components

The main responsibility of the MT SIP module will be the provisioning of basic SIP session control (setup, maintenance, renegotiation and termination), as well as session mobility support through a SIP interface with the SIP Server. The AR, as the default gateway for IP traffic, routes SIP messages to the corresponding SIP proxy using its common IP interface. The SIP module interacts also with the QoS Module in the MT in order to assure that the required QoS will be achieved. The integrated presence module will communicate with the Context Manager (WP4.2) to provide enriched presence information (context information) via a SIP interface.

SIP Proxy

SIP entities in the network side, facilitate the communications between end point SIP entities within the Akogrimo scope. The SIP proxy receives SIP queries from and will deliver SIP responses to the SIP MT using a standard SIP interface (and through the standard Access Router

IP interface, which routes TCP or UDP packets in which SIP messages rely on). In the multidomain scenario, SIP messages will be exchanged between SIP proxies for routing purposes. Communications with the QoS Broker (to check for resources availability or to perform resources reservations) and with the PBNM (for managing purposes) are managed by both XML-CIM agents.

Interlayer interfaces

Apart from the interactions with some network layer components, the following interlayer interfaces have been identified:

- Middleware layer (WP42): the SIP server will provide SIP based enriched presence and context information to the Context Manager, which will gather context information from many different sources. This information will be provided through a SIP interface. Additionally, access to services should be authorised and authenticated, so a Diameter interface with the A4C server will be required.
- Grid layers (WP43 and WP44): by using this SOAP interface, some entities on the Grid layers could request for the establishment of a SIP session between two end-users. The SIP server will initiate the process acting as a Third Party Call Control (3PCC, RFC 3725) entity.

7.2.3. Network Resource Management

Quality of Service

Every network has limited resources. With the growing demands put on networks, ensuring adequate network resource management and quality of service aspects is becoming more important. This is particularly important when real-time applications are considered.

The Akogrimo network resource management will control not only user requests to network services, but also handovers between different access networks. It will allow end-to-end quality of service, i.e. guaranteeing a set of parameters to users, such as bandwidth, packet delay, etc. Users requesting real-time services (e.g. a video conference) will have their packet flows treated with higher priority than another user which is downloading some large file, which has no latency concerns. In this sense, metering functions will account the total number of packets and bytes exchanged between an access router and a mobile terminal, thus allowing differentiated network control. The metering information will also be useful for the charging of a user.

QoS Broker

The QoS Broker is the network component which effectively manages all network resources. It receives global network policies from the PBNM system. The status of the network traffic is also made available to the QoS Broker by the metering components in the Access Routers, to prevent the QoS Broker from allocating resources which may not be met by the network. It also exchanges information with other QoS Brokers.

Mobility is also dependent of the QoS Broker, since a user may not change Access Network if, for example, the new AN has all its resources already occupied.

QoS reservations will be possible using time-limited “QoS bundles”; fine-grained QoS parameter tuning is not possible. A QoS bundle is a set of well defined QoS services targeted at a specific type of application, such as interactive video/audio, data or a mixture of both.

Interlayer interfaces

The QoS Broker will consult the A4C Server (4.2) for user profile information, which will allow the QoS Broker to decide if a given user may or may not use a certain amount of network resources. The QoS Broker has interfaces to the SIP Server and the EMS (4.3) to allow for resource availability requests or resource reservation requests.

Policy Based Network Management

Policy Based Network Management (PBNM) is an alternative for the management of telecommunications networks that offers a way of overcoming many of the limitations of existing human resource-intensive network management techniques. In heterogeneous and distributed environments like Akogrimo, network configuration to guarantee required end-to-end QoS, or to assure certain rules for admission or congestion control could be a very complex problem to solve without automatic mechanism for configuration and control.

Network-related policies that could be applied can be grouped onto the following categories:

- Policies concerning users. Policies in this group will describe who, where, when, how and on what conditions can have access to particular services. These policies will have an impact over the A4C Server.
- Policies concerning bandwidth allocation (so related with the QoS Broker, which will enforce policies on the Access Routers). Policies in this group will control bandwidth allocation. They will define the amount of bandwidth allocated for the particular transport QoS class and conditions under which the bandwidth can be reallocated.
- Policies concerning measurements and monitoring. Policies grouped here will describe what kind of measurements are to be taken in particular situation, how often the monitoring system should measure particular value and where pass the results on.

PBNM System

The PBNM system proposed for Akogrimo is a simplified version of the IETF-based PBNM architecture. A Global Policy Server (GlobalPS) is the component responsible for interface between network operator and the PBNM system as well as for proper installation of policies in the network. For simplification purposes, in the Akogrimo first phase it will be achieved using a simple XML file. These policies are stored in the Global Policy Repository and accessed using an LDAP interface. A mapping module is responsible for mapping global policies on network element specific policies, which are delivered to the Policy Decision Points (PDPs) and Policy Enforcement Points (PEPs) using COPS interfaces. All these processes are controlled by a coordination element called Global Policy Server Engine.

Interlayer interfaces

In the Akogrimo infrastructure, the A4C server (WP42), the QoS Broker and the SIP Server (WP41) will act as IETF PDPs, while the Access Router (which receives QoS policies from the QoS Broker) will act as a PEP.

7.3. Network Middleware Services Layer

7.3.1. Context Management

Functional role:

It is not realistic that a context infrastructure can provide every representation of context information needed for any domain specific purpose. The approach is therefore that the context system gathers basic context data, processes and refines it according to common Akogrimo context requirements and allows context consumer to add extension modules to handle domain specific context inference. In the Akogrimo architecture, the component that handles context is denoted Context manger. As shown in Figure 22, the context approach in Akogrimo is based on a layered approach.

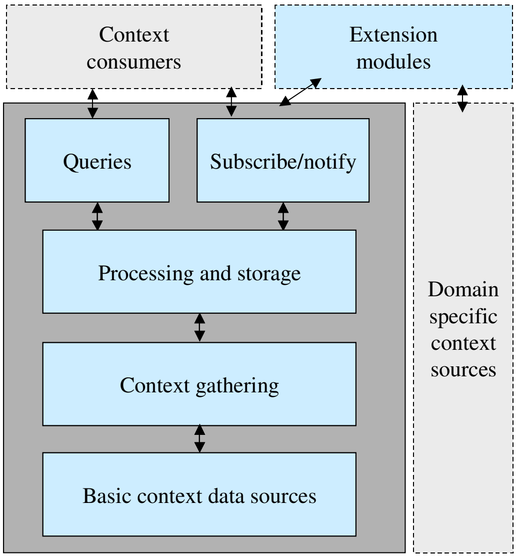


Figure 22 Context management functional layers

The context manager gathers raw context data from different context data sources across well-defined interfaces. Given the variety of possible context data sources, the context manager needs to handle multiple protocols for collecting data from different sources. The context manager has a set of rules (policies) describing the logic for context gathering (e.g. which sources provide context, at what schedule, action to take when something fails etc.). These rules are configured through an administrative interface. The context manager refines raw context data to provide standard formats (e.g. geographical coordinates, temperatures, etc.), performs processing according to common Akogrimo requirements and stores the context data in a context database.

Context consumers can either execute queries towards the context manager to get specific context information or subscribe for notifications about changes in certain context data. The context manager is responsible for distributing context data to context consumers accordingly. To handle domain specific context inference, the context manager offers interested parties interfacing with extension modules. One example is translation of location coordinates to a map of a building, such that the context will also specify which room a person is in, and what physical facilities. Context extension modules also allow context consumers to combine basic context information with additional domain specific context sources, e.g. data from specific medical sensors.

Outline of architecture:

Figure 23 shows how context data are collected. The Mobile terminal contains a SIP Presence User Agent (PUA), which sends SIP presence data (PUBLISH requests) to the context manager. The context manager in this case acts as a SIP Presence Agent (PA). A local service discovery agent in the terminal discovers devices in the vicinity using suitable protocols (Bluetooth, SLP,...). The result of this discovery, along with the capabilities of the mobile terminal itself, is then reported to the local service discovery server (which is part of the context manager). The local service discovery server will be implemented as a WEB service, accepting CC/PP documents. The context manager will register with one or more location services, to track the location of people. Location services could be based on many different location technologies. In the example here, the location service uses RFID readers to determine the location of users (which must carry RFID badges).

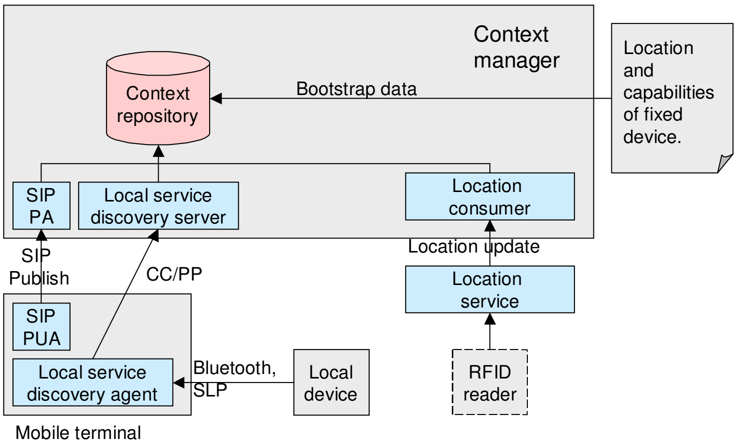


Figure 23 Collecting context data

Figure 24 shows how interested parties use context information, and how domain specific context can be inferred. A context consumer may poll for context data using a WEB-service accepting queries in a semantic language, as shown on the left-hand side of the figure. Alternatively, the consumer may subscribe to a given type of context data, and receive notifications when relevant data changes. In this case, WS-baseNotification will be used for the publish/subscribe mechanism, while queries and results will be expressed in a semantic language.

Context inference also utilizes the subscription mechanism. A context inference module will then infer further domain specific context, and submit this to the context manager, through a WEB-service accepting semantic language documents.

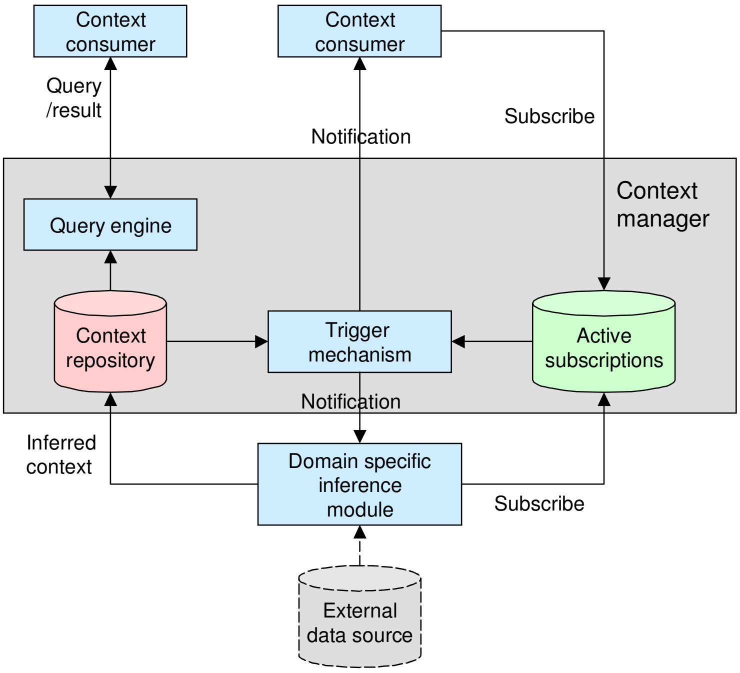


Figure 24 Using and inferring context data

__Interfaces:__

Interface Context manager – context data sources

The context manager and different context data sources must use well-define APIs to interact. The context manager will gather context data from multiple heterogeneous data sources. There will therefore be a need for several well-defined interfaces to communicate with different context sources.

- Context manager gathers user context data from a SIP presence user agent (PUA) running on the mobile terminal. An additional interface can be needed to get user context from other sources that SIP SIMPLE.
  Protocol: SIP (SIMPLE) (+ additional protocol if needed. This protocol will likely be a proprietary protocol developed in Akogrimo.)
- The Context manager collects information about services available in the nearby area of a user (not web/Grid services) from a Local Service Discovery agent running on the mobile terminal. The agent also reports terminal capabilities.
  Protocol: Web services – cc/pp
- The Local Service Discovery agent running on the mobile terminal uses local service discovery protocols to find available services.
  Used protocol: SLP, Bluetooth (OSGi is also an option)
- The context manager communicates with location service(s) to find the location/position of users.
  Protocol: the protocol used will dependent on location technology, for RFID a proprietary protocol based on web service will be developed.

Interface Context Manager – context consumers

Next, we describe the interface(s) used by external context consumers to interact with the context manager. There are at least three distinct types of interactions that are allowed across this interface (will this be one interface or three distinct ones?)

- Context queries: Context consumers use the interface for queries for specified context data
  Protocol: Semantic language (e.g. RDF, OWL) over Web services
- Context subscription: this interface allows context consumer to subscribe to notifications about changes in context info
  Protocol: Semantic language, e.g. RDF, OWL over WS-Basenotification
- Context extension: this interface allows context consumers to specify domain specific inferred context
  Protocol: Semantic language (e.g. RDF, OWL) over Web services

In addition to the interfaces presented above, the context manager needs to interact with the A4C system.

7.3.2. Service Discovery

Functional role:

The Service Discovery Service (SDS) has functionality that allows a service requestor to find an appropriate service provided by a service provider. A service provider must describe the service capabilities (Service Description) and publish the service through the Service Discovery Service. The Service Discovery Service (SDS) maintains a list of available Services and their capabilities. A service requestor (human or software agent) uses SDS to search for services that satisfies certain requirements on demand. The SDS matches the request against stored service descriptions and returns a list of matching services.

The design of the service discovery infrastructure implies the definition and specifications of certain elements such as directory registers, service description, registry procedures, and service retrieval or service invocation. All these elements are inter-related in order to build an acceptable infrastructure where services information is easy to access and easy to retrieve with an efficient management and proper maintenance.

Directory registries are repositories for information about services. Directory services provide a consistent way to name, describe, locate, access, manage and secure information about resources making relevant information accessible. The architecture of directory registers may be centralized, distributed, hierarchical, P2P structured or Ad hoc. Centralized directory registers imply to have all the information stored in one or several places (by means of using replication). This approach has some drawbacks like poor scalability or as less consistency on large registries. On the other hand, a distributed architecture would eliminate these problems. However, it requires an efficient synchronization procedure in order to not suffer of information redundancy or inconsistency. Related to this discussion it exists examples where the architecture registry has been chosen to be P2P

Service description is another important issue in SD. In order to efficiently access service information in a database, this information needs to be classified and be represented in a flexible format. The more expressive the description is the more accurate our matches will be at the time of looking for a determined service. Web Services use a XML format called WSDL in order to describe and detail their public interface. WSDL describes the protocol requirements and message formats needed to deals with the listing services for a particular Web Service.

In order to give more information weight to the description of a service we can establish relationships (schemas or ontologies) between different objects and enrich the descriptions with semantics. The use of ontologies provides a very powerful way to describe objects and their relationships to other objects. An ontology has been defined as a formal, explicit representation of a shared conceptualization. An example of a language to describe ontologies would be OWL (Ontology Web Language).

Akogrimo needs to take into account the mobile nature of services and users which must be reflected in the way services are discovered. Traditional more static focuses will have to be reviewed in order to obtain a SD system that is able to handle mobility to some extend. (E.g. due to this mobile nature, the registries will need to be updated more frequently and context information will be used to make searches more accurate.)

<u>Outline of architecture :</u>

Ideally we would like to have a unified service description and service discovery system however it is realised that the Akogrimo SD must handle very diverse services. The different types of SD protocols target different types of services. We may encounter protocols to discover devices, people, places, WSs and so forth. In order to embrace the greatest number of SD types several more or less well established protocols will be used. An API could be implemented in order to hide multiple SD systems from the requestor. It seems most realistic to have two (or more) loosely coupled systems for service discovery; discovery of "local services" (candidate solutions could be e.g. SLP, SDP, UPnP, Jini, etc.) and discovery of web services/Grid services (UDDI + semantic – not suited for mobility..).

Within the Akogrimo environment, we call General Service Discovery System (GSDS in short) to the Akogrimo subsystem in charge of supplying a particular service from a particular client request. GSDS have to deal with service of very different nature. On one hand, with the so-called local services/resources such as printers, beamers, virtual hard disks, and on the other hand with the Grid Services that can be defined as stateful web services proper to the semantic Grid.

For the sake of clarity and according to the different service nature, we assume that the discovery procedure involving both services types will be hardly decoupled (this assumption could be revised if required). Then, from now on, the system in charge of discovering local resources (services) will be called the Local Service Discovery System (LSDS), whereas Grid Service Discovery System (GrSDS) is the name for the module dealing with proper Grid Services.

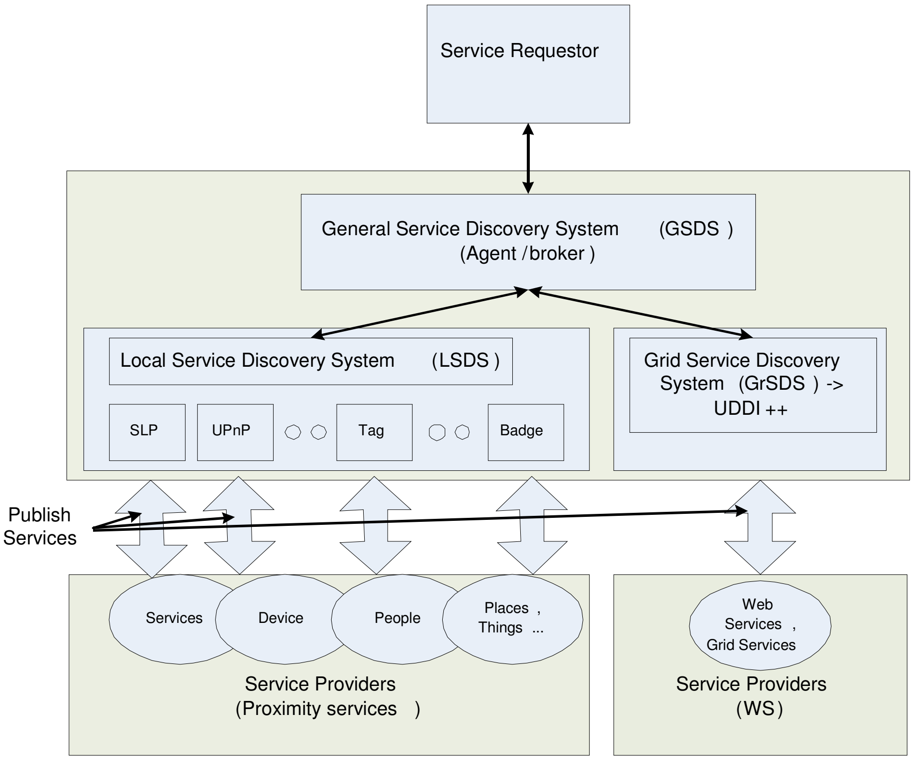


Figure 25 Concept of the Service Discovery in Akogrimo

Interfaces:

SDS - A4C:

The SDS contacts the A4C server in order to request SAML assertions to corroborate the requestor's authorization. The user may also be accounted by the use made of the SDS.

- Authorization:

The service Discovery System can be seen as a service itself. For this reason the A4C module would treat Service Discovery as any other service within Akogrimo platform. For instance, some rules relating who can use Service Discovery can be established (Authorization procedure) if required.

As in any service, the user must be authorized to use it. To this end, the A4C server has to validate the cryptographic material provided by the user to the SD mechanism. This could be any a token that designates the user as a member of a certain VO and therefore ascribe him certain rights when looking up services. For example, members of a VO won't be allowed to see services that are restricted to a different VO.

Protocol: Diameter and SAML over diameter.

- Accounting/Metering:

  The user will be accounted for the use made of the SD. Discovering basic services maybe free while the search of more sophisticated services could be charged. There is a need to meter the use made of the SD by each user and communicate it to the A4C server.

Protocol: based on Diameter.

Context Manager -> Local Service Discovery Service:

- Context Manager (Local Service Discovery Agent running on mobile terminal) - LSDS: Mobile terminals have the need of finding nearby services within the Akogrimo system. A Local Service Discovery agent on the mobile terminal collects information about available services. The available devices and services that a user has in his local environment are in fact part of its context.
  Protocol: The communication might be established though an extended SLP-protocol or similar where semantics and ontologies should be included to to enrich the service discovery.
  Protocol: Extended-SLP-type Protocol

Service requestors - GrSDS:

Service requestors will interact with the GrSDS over this interface to search for Grid services. Potential service requestors are:

- Mobile Terminal: Software or user initiated. A simple user or software agent may contact the GrSDS directly.
- EMS: The SDS is contacted by the EMS when the discovery of a service is initiated as a result of a WF initiated in the WP4.4 (application level). The EMS looks for resources to carry out the sequence of jobs that has assigned. The communication could be established through an extended SLP-type protocol (semantics included) with LSDS whereas SOAP could be used for GRSDS system.
- Workflow Manager: GrSDS is used by WP4.4 when building a composed service. When the service requested is not prone to be executed with the current device characteristics (screen size, bandwidth, and so forth) the WP4.4 has to create workflow in order to add a new device with sufficient capabilities to the current session or redirect the session to the new device: find bigger screen, etc…
- VO Manager: Design rules and rights to access services. This is not necessarily a direct interaction.

Protocol:

With GrSDS: SOAP (semantic language over Web Service)

Service provider - GrSDS:

A service provider interacts with the GrSDS to publish services they offer.

7.3.3. A4C

Functional role :

Figure 26 illustrates all identified functional components of the A4C subsystem, and their interactions with the other Akogrimo entities. The A4C system is based on the DIAMETER protocol as specified by IETF, adapted with mechanisms capable of delivering AAA services to mobile Grid services and enriched with Auditing, Non-Repudiation and Charging capabilities.

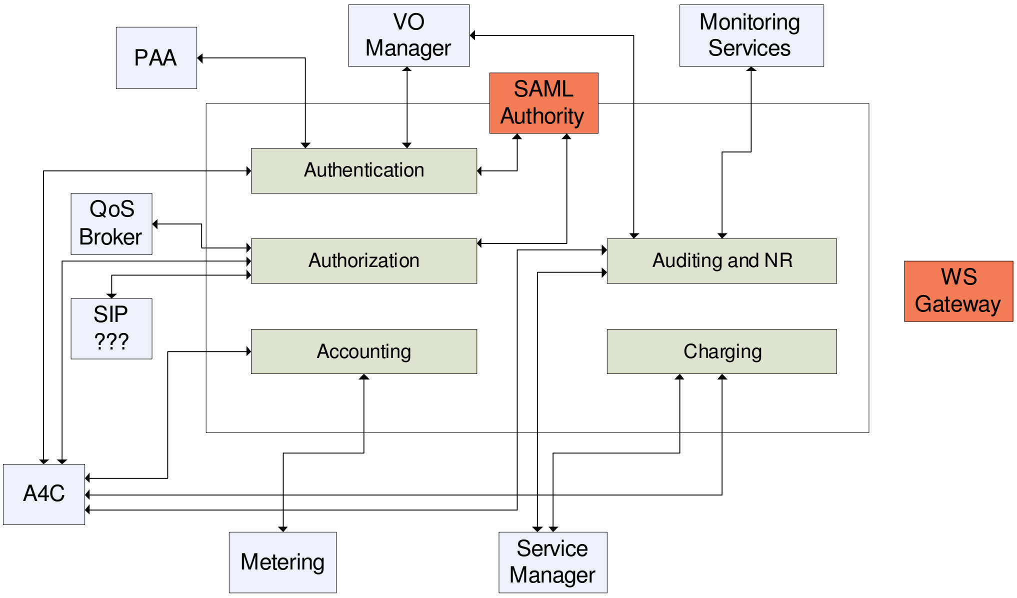


Figure 26 A4C functional overview

The authentication mechanism defines a process for verifying a user's identity. An authentication policy describes which authentication mechanism shall be used for identifying the user. In Akogrimo multiple mechanisms shall be allowed. However, the A4C system shall do the authentication itself.

Authorization is the process of deciding on an entity's allowance to perform a particular action or not. The Authorization decision depends on service specific attributes (e.g. service class for QoS service, device requirements) and user-specific attributes (e.g. name, affiliation to a certain group, age, etc.). Akogrimo will use the A4C authorization mechanisms for accessing network layer services and requesting Grid application. After a successful request of such an application, all the further authorization decisions for performing the Grid service are taken by the VO (Virtual Organization) manager using Grid-specific authorization mechanisms. Although the VO Manager performs by itself authorization decisions for accessing Grid resources, the A4C system shall be informed about these decisions in order to provide further Auditing to the upper layers.

The accounting system performs two main tasks. The first is collecting data about resource consumption while the second task is retrieving accounting data whenever this is requested by a legitimate entity. Such requests could be:

- A user wanting to see a detailed record of the services he used
- The charging component wanting the accounting records for a certain user in order to create a bill

In a mobile Grid the resources that have to be accounted for are of much more diversity than the traditional network accounting (such as CPU usage, memory consumption, storage size, accessed content).

Every service shall implement a *metering component (MC)* collecting the service usage information from a service, using a service dependent protocol, and presenting this information to the Accounting system.

Auditing defines the process of storage and retrieval when needed of information about the events taking place in the system, history of the service usage, SLA (Service Level Agreement) compliance and the customer charging and tariff schemes applied. The auditing component shall keep track of all events in sequential order based on their timestamp. All Auditing records shall provide enough information for identifying the entities involved in an event, the time the event was produced and the outcome of that event.

Charging is the task of calculating the price for a given service consumption based on accounting information. Charging maps technical values in monetary units and then applies a previous established contractual agreement between service provider and service consumer upon a tariff. One big influence in the final price paid by the user for the service but also influencing the revenue of a service provider is the charging scheme applied for a service. Multiple charging schemes exist in the mobile telecommunication and Internet providing business, but these have to be adapted in the context of Akogrimo. It is hardly probable a single charging scheme could be used for all mobile Grid services, so the charging subsystem shall be able to use different charging schemes for different services. Each service shall define its own metrics to be accounted and shall be able to define its own charging scheme to be applied on those metrics. However, a service could also use several charging schemes at the same time, for example for users with different subscription types.

Outline of architecture :

Figure 27 shows an outline of the proposed A4C architecture.

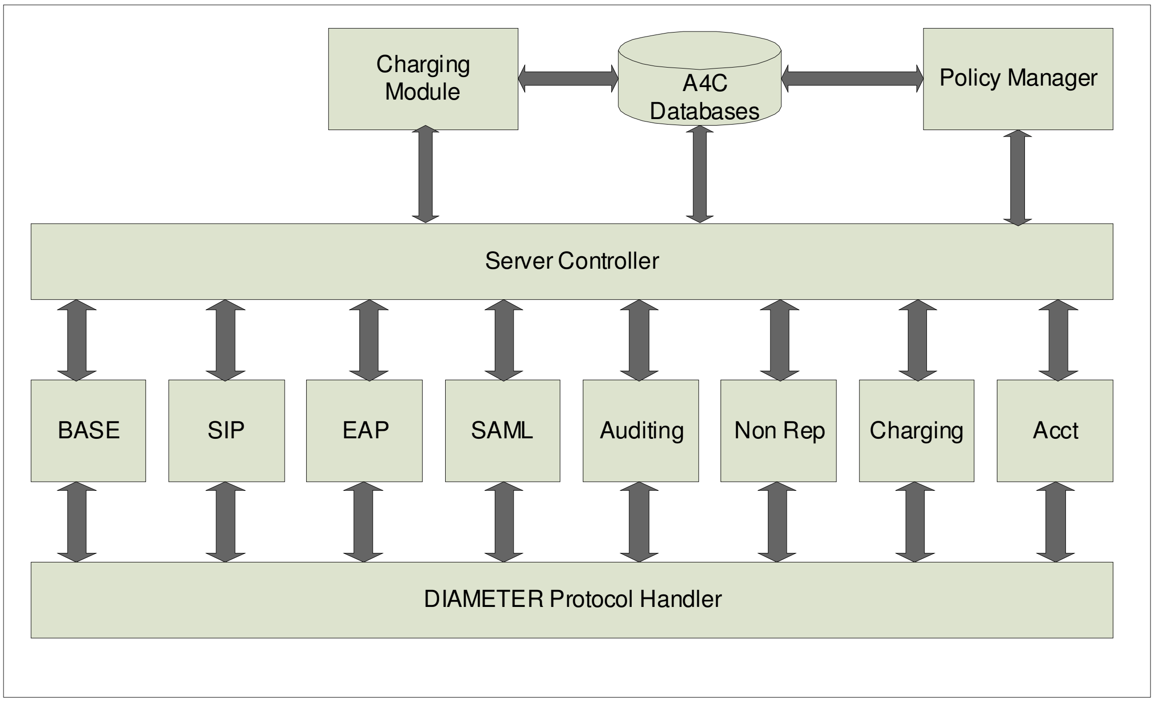


Figure 27 A4C Architecture

Interfaces:

Authentication:

- Authent.request:
  Foreign A4C – Diameter
  PAA – Diameter
  VO Manager – Diameter/WebService

- Authent.verify.SAML
  Foreign A4C – Diameter
  VO Manager – Diameter/WebService

Authorisation:

- QoS.request: QoS Broker – Diameter
- Author.request
  SIP Registrar/Proxy? (Diameter)
  Access Router – Diameter
  Foreign A4C – Diameter

Accounting:

- Acct.request
  Metering – Diameter
  Foreign A4C – Diameter
  SIP – SIP/Diameter (gateway)

Charging:

- Publish.charging.scheme
  Service "Manager" – Diameter (XML,WS)
  Foreign A4C – Diameter

7.4. Grid Infrastructure Services Layer

7.4.1. Execution Management Services

Execution Management Services (OGSA-EMS) are concerned with the problems of instantiating and managing tasks. It aims to answer questions such as: Where can a task execute? What are the locations at which it can execute because they satisfy resource restrictions such as memory, CPU and binary type, available libraries, and available licenses? Given the above, what policy restrictions are in place that may further limit the candidate set of execution locations? Where should the task execute? Once it is known where the task can execute, the question is where it should execute. Answering this question may involve different selection algorithms that optimize different objective functions or attempt to enforce different policies or service level agreements.

7.4.2. Capabilities/Functionality

- Selects the set of resources that can be used to execute a submitted job.
- Assigns jobs to resources and creates an execution plan, trying to balance the workload, optimize the performance and provide QoS.
- Handles job queues and priorities to meet SLAs or handle crisis situations.
- Replicates jobs to provide fault tolerance.
- Provides advanced resource reservation.
- Manages the job execution (deploy, start, suspend, terminate).
- Provides an API that can be used for the interaction with other components.

- Performance: the algorithms used for the functionalities mentioned above (scheduling, prioritization, replication, etc.) can be run in real time and do not cause a significant overhead to the overall system performance.
- Scalability: the system can function under heavy workload.
- Modularity: the system is composed by several components and can be easily maintained and/or modified.
- Security: the system interacts only with authenticated and authorized entities.

Job Manager

The Job Manager (JM) encapsulates all aspects of executing a job, or a set of jobs, from start to finish. A set of job instances (complex job) may be structured (e.g., a workflow or dependence graph) or unstructured (e.g., an array of non-interacting jobs). It may schedule them to resources and it may collect agreements and reservations.

Candidate Set Generator (CSG)

The CSG is in charge of finding the computational resources where the job can be executed. In order to find the possible resources where to execute the job, it is necessary to have a sort of match-making mechanism. The match-making permits to find the resources that correspond to the requirements expressed by the service consumer.

The CSG should take into account all the requirements that are static and not dynamic. The EPS should start from the match-making results provided by the CSG and combine them with the dynamic attributes, run an algorithm and find the best resources where execute. The distinction between static and dynamic attributes is strongly related to the Grid concept. To be more precise we give an example of these attributes.

Static resources attributes could be:

- Operating system
- Processors numbers
- Software available (libraries, binaries, …)
- Memory
- Disk space
- Bandwidth, etc…

Dynamic resources attributes could be:

- Free disk space
- Available CPU
- FileTransfer rate, etc…

We have listed the attributes without taking into account any distinction on the possible type of resource.

Execution Planning Services

The Execution Planning Service (EPS) is a high level scheduler that creates mappings called "schedules" between jobs and resources –the resources that have been selected by the CSG. An

EPS will typically attempt to optimize some objective function such as execution time, cost, reliability, etc – i.e. "where should the job execute", to improve the system performance, provide Quality of Service and meet the Service Level Agreements. For this reason the EPS may implement a job priority system. This can be achieved by having several job queues, each with a different priority. Additionally the EPS may implement a checkpointing or replication scheme for fault tolerance, depending on job characteristics, SLAs and the generated schedule.

Advanced reservation services (ARS)

The ARS is in charge of reserve resources for a specific period of time or permanently, depending on the type of reservation. Different types of reservation are possible:

- Computational resource reservation
- Storage resource reservation
- Network resource reservation
- Service reservation

The computational reservation guarantees that the specified resource is available at the time that a job is executed. The service consumer (SC) can reserve a certain amount of disk space, a specific percentage of CPU, etc.

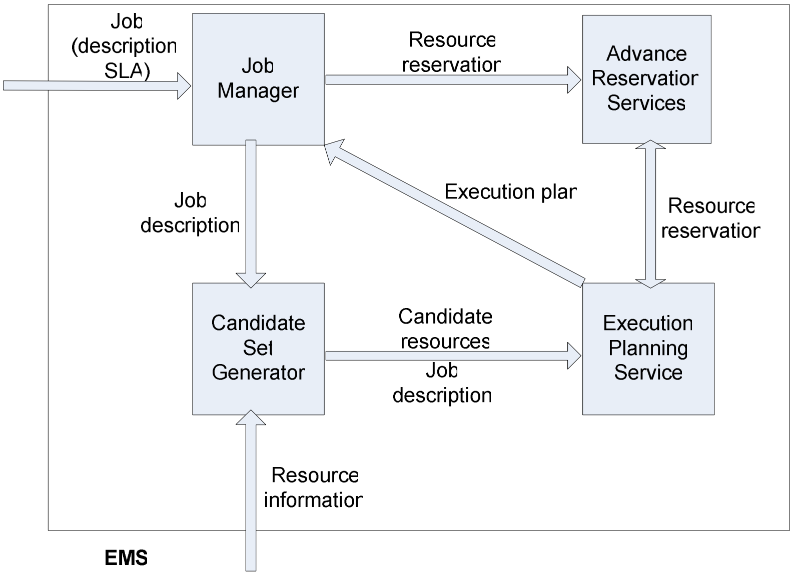


Figure 1, The EMS components and their interactions

7.4.3. Data Management services

The Akogrimo Data Management service is based on the OGSA specification. The OGSA vision offers a broadly applicable and adopted framework for distributed system integration, virtualization, and management. Successful realization of the OGSA requires the definition of a core set of interfaces, behaviours, resource models, and bindings.

This section focuses on requirements, capabilities required to support data management in Grid systems and applications and, finally, on the interactions with other components of the Grid Infrastructure Services Layer.

Efficient access to and movement of huge quantities of data, data sharing, archiving of data and data management are essential requirements in more and more science, technology and business areas.

Data services are used to move data to where it is needed, manage replicated copies, run queries and updates, and transform data into new formats.

Data services requirements include:

- *Data access.* Easy and efficient access to various types of data (such as database, files, and streams), independent of its physical location or platform, by abstracting underlying data sources is required. Mechanisms are also required for controlling access rights at different levels of granularity.
- *Data consistency.* consistency should be maintained when cached or replicated data is modified.
- *Data persistency.* Data and its association with its metadata should be maintained for their entire lifetime. It should be possible to use multiple persistency models.
- *Data integration.* mechanisms for integrating heterogeneous, federated and distributed data are required. Many different types of data must be supported (flat files, streams, DBMS, catalogues, derivations from other data, data services as data resources, etc.). It is also required to be able to search data available in various formats in a uniform way.
- *Data location management.* The required data should be made available at the requested location. It should be possible to allow for selection in various ways, such as transfer, copying, and caching, according to the nature of data.

The study of requirements introduced above results in specific functional and non-functional capabilities regarding Data Management.

7.4.4. Functionality

Data Replication Service

The primary functionality of this component is to allow users to identify a set of desired files existing in their Grid environment, to make local replicas of those data files by transferring files from one or more source locations, and to register the new replicas in a Replica Location Service.

Replica Location Service

The Replica Location Service (RLS) allows the registration and discovery of replicas. An RLS maintains and provides access to mapping information from logical names for data items to target names. These target names may represent physical locations of data items or an entry in the RLS may map to another level of logical naming for the data item.

Metadata Service

This component relates to the metadata, e.g. the data about data. The metadata service is in charge of maintaining information about the data stored, to set, get and query the metadata.

File Transfer Protocol

This protocol, despite of the name, should be dedicated to secure data transfer of huge amount of data. In addition the data to be transferred should be of any types and size.

Reliable File Transfer

The RFT is in charge of moving data from one location to another. It differs from the FTP because it provides a “job scheduler"-like functionality for data movement. Providing a list of source and destination URLs the service writes a job description into a database and then moves the files on behalf of the requestor.

Storage Element

The local storage system could be simply a file system or even a high-capacity mass storage device. The Storage Element should be placed on top of the local storage system and provide functionalities to upload data, retrieve data, replicate data, etc. The existing of a Storage Element is very important to standardise the way data are stored and accessed.

7.4.5. Monitoring services

This section describes the Monitoring subsystem of the Grid Infrastructure Services Layer. WSDM and WSRF are considered. Monitoring tasks become fundamental in every distributed computational system. Monitoring data represents an operational photograph of the system behaviour along the time axis. Such information turns to be fundamental to determine the origin of the problems or to tune different system components. For instance, fault detection and recovery mechanisms need a monitoring component to decide whether a particular subsystem or server should be restarted due to the information collected by the monitoring system.

The MUWS[30] specification of WSDM defines how the ability to manage, or, how the *manageability of*, an arbitrary *resource* can be made accessible via *Web Services*. In order to achieve this goal, MUWS is based on a number of web services specifications, mainly for messaging, description, discovery, accessing properties, and notifications, all of them included within the so-called Web Service Resource Framework (WSRF). At this point, it is very important to note that monitoring tasks must be considered as a subset of management.

The MUWS specification define the following basic entities/actors:

- Manageability endpoint (Web service endpoint): The access to the resource to be managed.
- Manageability consumer: The entity carrying out the management of the resource.
- Manageability resource: The entity being managed.

The basic concepts of management using Web services can be illustrated by the Figure 1[30]:

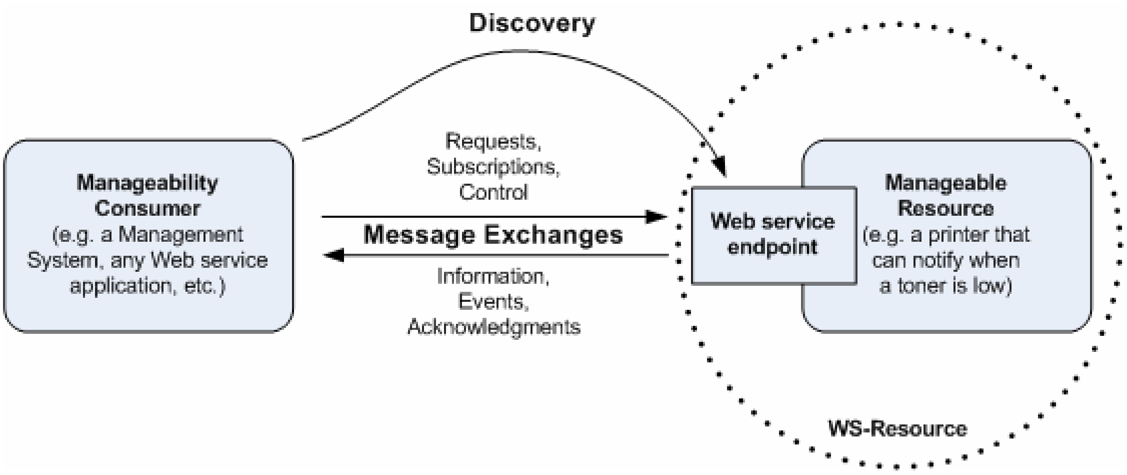

Discovery
Manageability Consumer (e.g. a Management System, any Web service application, etc.)
Requests, Subscriptions, Control
Message Exchanges
Information, Events, Acknowledgments
Web service endpoint
Manageable Resource (e.g. a printer that can notify when a toner is low)
WS-Resource

Figure 1. WSDM Concepts

The basic process of resource management can be summarized as follows:

1. Manageability consumer discovers the manageability endpoint by means of a certain discovery mechanisms that is out of the scope of the monitoring component (discovery mechanisms are the same used for standard web services).
2. Manageability consumer and the web service endpoint exchanges some messages in order to request information, subscribe to events, or, control the manageable resource associated with the endpoint.

A manageability consumer first obtains from the web service endpoint an Endpoint Reference (EPR), as defined by the WS-Addressing specification [53], and afterwards obtains any other required descriptions, including, but not limited to, a WSDL document [54], an XML Schema, or a policy document. MUWS uses the same mechanisms, for obtaining EPRs and their associated descriptions, as used by regular Web Service implementations, and their applications.

The central entity in WSDM specifications is the resource and its management. WSDM specifications involves the way a manageability entity (manageability consumer) accesses and manages certain resources. There are two different aspects related to resources: functionality and manageability. For instance, the functional aspect for printers is printing whereas the manageability aspect would be everything related to the control and management of the printer: level of the toner, whether the printer is online/offline. Every resource that is susceptible of being managed is called manageability resource. A manageable resource may support a number of capabilities. In particular, a implementation of a manageable resource should provide a number of manageability capabilities via Web Services endpoints. Consequently, the manageability consumer will access and manage manageability resources via manageability endpoints that from am implementation point of view are Web services endpoints.

The manageability capabilities implemented in the manageability endpoint can be classified in two different groups:

- General manageability capabilities: They represent those capabilities that are common to every manageability resources such as: identity or availability capability[30].
- Resource-specific manageability capabilities: Those capabilities that are proper of the particular manageability resource[31].

WSDM-MUWS specifications define the manageability capability as being composed of the following elements: properties, operations, events and metadata.

Properties of a manageability resources are exposed via a XML document called resource properties document. From an implementation point of view, the resource properties document is included within the *type* label of WSDL document [22].

The operations over a manageability capability are indicated via the label portType in the WSDL document of the corresponding web service (manageability) endpoint.

Events are offered by the manageability endpoint whenever a certain property change. Their implementation is based on WS-Notification specification [32]. Their properties will be included within the resource property document whereas the operations, such as subscription, can be found within *portType* label.

Finally metadata can be specifically defined for properties, operations and events. MUWS supplies three metadata items: mutability, modifiability and capability but also allows a different one to be defined.

7.4.6. SLA enforcement

The Service Level Agreement (SLA) subsystem offers to Akogrimo infrastructure the management of all aspects related to the fulfilment of a quality of service (QoS), that is conditions agreed between the Service Costumers (SC) and the Service Providers (SP) for using services provided by the seconds. The use of any kind of service on user's side must assure a specific level of QoS that both actors have negotiated before. SLA is in charge of managing several steps as negotiation and fulfilment of QoS and to know what recovering actions should be applied if something happens during the execution phase, it could be when a service is not available, when QoS for a specific service is not met, …

SLA must be present on several places inside Akogrimo infrastructure. Regarding WP4.4, after the negotiation phase exists a QoS agreed between a SC and a SP. As a result, a contract is made between both actors for guaranteeing the achieved agreement. This contract must contemplate the agreed QoS together all the possible situations that could happen during the use of the service. Once a service is being used, there is an agent that is responsible for the fulfillment of the conditions of the contract. Whenever a service use does not meet these conditions this agent will notify it to the SLA-Decisor that will perform the appropriate actions.

SLA-Controller: This module is in charge of supervising that all the agreements reached in a contract are respected. It also catches all violations produced on a service (in a resource) that does not fulfil the conditions that have been agreed.

SLA-Decisor: It receives and manages the notifications from the SLA-Controller and it decides, according to defined policy, which actions have to be carried out (this is show messages, to apply discounts, to destroy the service, to increase processes priorities, etc).

7.4.7. Metering services

The Metering component is responsible to keep track of usage of specific resources that are utilized within the execution phase. The parameters that are going to be metered are to be defined. However some indicative (concerning the specific purposes of WP4.3) are:

CPU-time, Wall-clock time, Storage, Bandwidth, Data transfer(s), Quality of Service (e.g. higher batch queue priority), Software usage, etc…

7.4.8. Policy management services

The Policy Manager is the component responsible for the handling of the policy that will be applied in the execution of the various jobs under this Workpackage. Examples of such policies can be: “do not transfer data between two specific databases if are greater than xGB”, “Do not meter information for specific urgency issues” etc. The policy manager must be informed by other policy managers (from WP4.4 for instance who has the overview).

Action inside the VO have to be controlled, then each participant will be subordinated to some rules in order to access resources (i.e. services) provided by other participants. In this scenario each SP can provide policies for own resources and VO manager can provide policies for all involved VO participants. The aim of Policy Manager (PM) are to provide mechanism to hold and manage policies and reply to policies requestor (PR is an abstract entity that submits policies request) at low level. In fact we will address it inside hosting environments handled by SP.

The policy manager architecture foresees the following functionality:

- DB management

  Allows to add/remove DB policies information. This functionality provides an interface used to manage policies stored in DB.

- Policies Selection
  - Searches in DB the policies that are involved in a context, described by request performed by PR.
- Uplink Request
  - Retrieves the Global policies from GPM. The request is based on the request performed by PR.
- Conflict Solving

Merges policies in order to produce the final policy. Eventual conflict will be resolved.

- Query Service
  - This functionality exposes an interface used by PR to submit query to PM System.

7.5. Application Support Services Layer

Inside Akogrimo project there is an extension definition of VO as described in paragraph above that includes also mobile terminals. Behind the application layer every entities is seen as service (in particular, Web service or WS-resource) and this layer enables applications to found their capabilities over provided services.

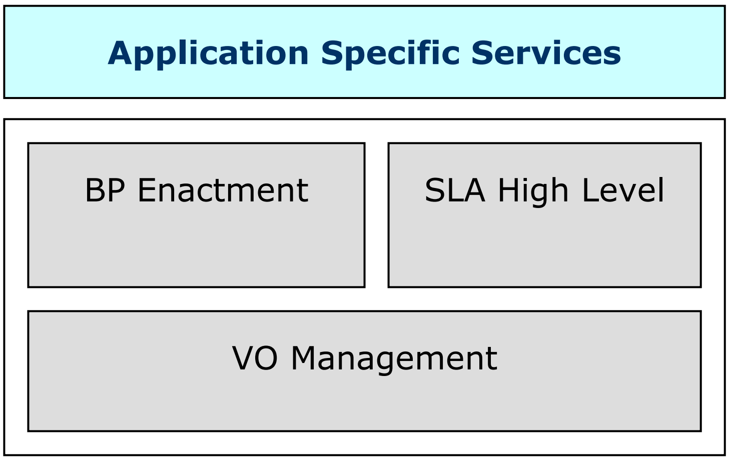


Figure 28, WP4.4 Main Components

In the figure above, three main management areas have been identified : VO Management, BP Enactment and SLA High Level. These layers provide core functionalities supporting the developing of applications foreseen in Akogrimo environment.

The Application Specific Services represent functionalities individuated for developing the application software for several devices, enabling the interactions with Akogrimo VO. Cross communication among components belonging to this layer is needed in order to gain their functionalities, e.g. communication between components within VO management and SLA High Level in order to enlist a new service provider into VO.

7.5.1. VO Management

VO management should provide capabilities to subscribe new participants (users, service providers, etc.) inside the VO, to register those participants enabling them to sign-in their presence in VO, to support publishing phase of services and to check the access rights for invoking services. These activities can be summarized:

- Subscription: anyone wants to use or to provide the VO services has to enroll to the VO, becoming VO participant (“off line” subscription or long term subscription). After subscription, any entity becomes VO member and will be related to a role (VO-role) and an entity can assume more roles in the same or other VO as well.
- Registration: a VO enrolled entity can access to its VO by sign in the VO. VO management must provide mechanisms to recognize its enlistees and to validate its accesses to VO resources for fixed time (by creating a VO session).
- Publishing services: services provider entity, after subscription and registration phase, can publish its services by asking VO management for support. These services will be available to all services requestor fulfilling all contractual services policy rules.
- VO environment management: the services purchase and using is looked over by VO management as well.

VO management supports also the “roll back” of these operations like un-subscription, unregistration and un-publishing. Therefore, taking in account the main aspects in Akogrimo such as the ability to reallocate and to replace resources, to accommodate changes in requirements or to adapt to new opportunities in the business environment, as well as to enable entities to be location bounded-less, the VO management is the service able to provide a secure managed environment for Mobile and Dynamic Virtual Organization. VO management is also responsible for instancing and governing an Operative Virtual Organization needful for execution of workflow instances by ensuring the execution environment for the involved user/s and service/s.

7.5.2. Business Process Enactment

The business process represents a set of one or more linked procedures or activities that collectively realize a business objective goal, normally within the context of an organizational structure defining functional roles and relationships. The business process is available at application layer and the final consumer wants to buy and use it. Behind the business process there is a workflow that represents the automation of the business process. The workflow coordinates and manages component services or entities involved into the automation of business process. The view of business process is a high level vision (application layer), but at low level layers we need to map it in a service[2]. In this vision we need to deploy the service inside registry service, using templates that identify their features. As service features we mean at least QoS and providers that are able to provide them. From the accounting and charging view point should be inserted also features about resource consumption and violation management.

A BP Development service is also foreseen in order to define and modelling roles inside the specific Business Process, allowing to relate actors within workflow definition with specifics capabilities and functionalities. These roles clarify the responsibilities, capabilities and credential of an entity with respect to other ones within the workflow and Grid environment. The BP roles differ from VO roles because the first ones are used only for the relations among actors involved within a specific Business Process, defining the subset of their services to be used within the context of the workflow itself.

Inside Akogrimo project there are a lot of participants that can interact together in order to provide a service. Each participant has a specific role and a specific profile, then if they cooperate to provide value added services need coordination (during the provision phase, building the operative VO and executing service). In this case choreography should be useful, because it concerns with describing the message interchanges between participants. All participants of choreography are peers, so there is no centre of control. A choreography definition can be used at design time by a participant to verify that its internal processes will enable it to participate appropriately in the choreography. It can also be used to generate a public interface that can be used to tie in internal activities to support the choreography. At run-time, the choreography definition can be used to verify that everything is proceeding according to plan. It can also be used unilaterally to detect exceptions (a message was expected but not received, or help a participant in preventing it sending messages in the wrong order or at the wrong time). In order to use choreography with web services there is a Web Service Choreography Description Language (WS-CDL).

Choreography will be also utilized in coordination with BP Development Service in order to define the rules among the services (by peer-to-peer manner) involved within a workflow. The orchestration indeed defines the services interaction among services, describing the execution order of involved entities, while choreography defines the sequence of messages between couples of services. For instance, in an execution scenario of e-Health workflow the orchestration definition can describe the point where a doctor and patient service can be involved inside the overall emergency service (i.e. emergency workflow). This step means that the doctor and the patient

[2] I would like to underline that at VO level we manage services (Web Services or WS-Resource). The Application Support services layer will interact only with services

must communicate because the doctor has to know the patient symptom. The choreography can help to describe the interaction between patient (service) and doctor (service) decreeing the patient must provide at first the symptom data, the doctor (after data analysis) must send patient its report, etc.

At Grid Application layer a Choreography service will enable user applications to fix relations between (Web) services by supporting the definition, modelling and execution of choreographies.

7.5.3. SLA High Level Services

In order to establish the QoS between service provider and service customer, the SLA negotiation and management must be controlled at this (VO) level as well. The SLA management services are in charge of monitoring the negotiation phase between service customer and service provider. At the end of negotiation phase a SLA contract is established/agreed between Service Provider and Service Customer. Inside the final contract can be inserted some policies describing the service usage, charging mechanisms and violations management.

7.5.4. Policy

Life inside the VO needs to be controlled, then each participant will be subordinated to some rules in order to access resources (i.e. services) provided by other participants, furthermore each participant has to clarify under which rules it is possible to use their own resources (i.e. services). So, when a participant/user wants to use a service is necessary to build an operative VO and the policy rules have to be taken in account. In order to guarantee the respect of the rules is necessary to have a repository where are stored information or metadata about participants profile and the rules that has to be taken in account to use his services.

7.5.5. Main Components Description

Each management area is characterized by the main components individuated. Briefly the various components functionalities can be summarized as follow.

- **VO Manager**: is the central component for the management of the VO, enabling the access of new participant entities, managing and checking their usage of VO services (e.g. enabling the publishing of services by the Service Provider participants). In order to accomplish its tasks, VO manager communicates and uses services provided by the application support layer itself and lower services (e.g. security services, A4C…).
- **OpVOManager**: is responsible for the creation and management of the Operative VO. The Operative VO represents the transient environment created for the execution of a workflow including user/s and service/s.
- **OpVOBroker**: deals with services purchase by researching and asking for their negotiation. In order to achieve this target it is in charge to check profiles and SP credentials, ranking the list of SPs that are able to provide services. After a successful negotiation, it is created a new Operative VO which will contain all negotiated services.
- **Participant Registry**: it is an information service which deals with static information about participants in the VO. Inside the OpVO it can be used to specialize the role of users that will access to it.
- **Service/User Agent**: inside the operative VO the user and the service are virtualized respectively with these two entities which should provide control on the user-service inter-

actions. They may be viewed as proxies (to user and service) used by the OpVOManager to manage these entities.

- **Workflow Manager**: deals with the interpretation and enactment of workflow. Workflow Manager mainly collaborates with OpVOBroker to complete the negotiation phase and provide the business process to final user; OpVOManager to set up the OpVO and to manage workflow execution. Workflow Manager has also to monitor the peer-to-peer communication by enacting the sequence of messages exchanged between services and dealing with the monitoring of these interactions.
- **Workflow Registry**: provides and records the description of concrete (related to workflow execution phase) and abstract workflows (related to workflow definition phase). That data will be interpreted by Workflow Manager in order to enact workflows.
- **Business Process Designer:** enables the design, updating and managing of a Business Process allowing to itemize the workflow and Business Process roles.
- **SLA Negotiator:** in charge of services selling in the VO which will be associated to an Operative VO. SLA Negotiator allows the individuation and negotiation (by means of "Pre-reservation" and "Reservation" contracts) of services.
- **SLA Translator**: provides a transparent way to access to the SLA contract and template documents, for each service provided by service providers and visible within the VO by interpreting the SLA contract. The SLA Translator provides support to SLA Negotiator during the negotiation phase as well.
- **SLA Access**: is in charge of managing the SLA documents because the other modules cannot understand them. SLA Access interacts with the SLA-Translator to achieve its purposes.
- **SLA Repository**: Repository that stores all SLA-Templates of all published services
- **SLA Contract Repository**: Repository that stores all established (live) SLA-Contracts between Service Provider(s) and Service Customer.
- **SLA Designer**: allows the definition of SLA contract for new Services joining to VO.

In order to explain the origin of individuated components at this layer, the main topics are taken into account and depicted in the following subparagraphs.

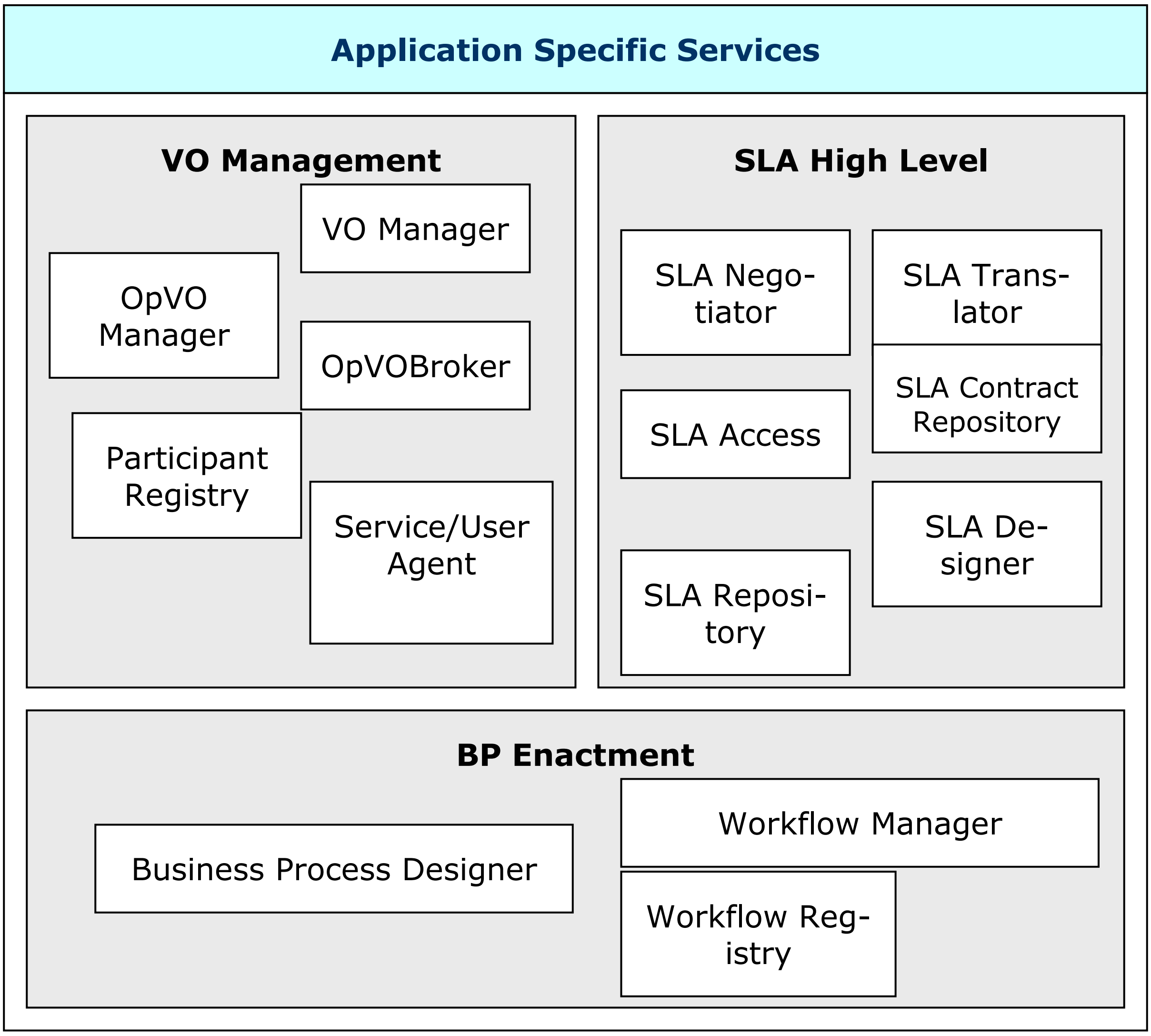


Figure 29, WP4.4 main components

8. Interface Specification

This section specifies the interfaces between the four implementation work packages WP4.1, WP4.2, WP4.3 and WP4.4. The interfaces between the components within a single implementation work package (between components of the same colour – chapter 6.2) are described in the relevant deliverables and not described here.

WP4.1 subsystem	………… WP 4.2 /4.3/4.4 …………		
	Higher layer subsystem interface	Purpose of interface	Method of connection
Mobile Terminal	EMS (and perhaps other 4.3 components)	-VO session actions	SOAP
Mobile Terminal	Context Manager (4.2)	- Context information (presence)	SIP
Access Router	A4C (4.2)	-User authentication -Network service authorization -Sending metering information	Diameter
SIP Proxy	A4C (4.2)	-User authentication -Sending user authorization requests -SIP Session Accounting	Diameter
SIP Proxy	EMS (and perhaps other 4.3 components)	-Receiving SIP session requests	SIP
QoS Broker	A4C (4.2)	-A4C user profile requests for QoS authorization	Diameter
QoS Broker	EMS (and perhaps other 4.3 components)	-Receiving resource availability requests -Receiving resource reservation requests	SOAP
PBNMS	A4C (4.2)	-Sending policy definitions -Receiving policy requests	CIM-XML

8.1. WP 4.2 Interfaces

Table 1 – WP4.2 interfaces towards higher layers

WP4.2 subsystem	Connections with WP4.4 and 4.3		
	Higher layer subsystem	Purpose	Method of connection

WP4.2 subsystem	Connections with WP4.4 and 4.3		
	Higher layer subsystem	Purpose	Method of connection
Context manager (Query engine)	BP Enactment - WP4.4	Queries for specified context data	Semantic language (e.g. RDF, OWL) over SOAP
Context manager	BP Enactment - WP4.4 Domain specific inference modules – WP4.4	Subscribe to notifications about changes in context info	Semantic language, e.g. RDF, OWL over WS-Basenotification
Context manager	BP Enactment - WP4.4	Specify domain specific context extensions	Semantic language (e.g. RDF, OWL) over SOAP
GrSDS	EMS – WP4.3 BP Enactment – WP4.4 MT - user - Software agent?	The service requestor contacts the GrSDS to find a service.	With GrSDS: SOAP (semantic language over SOAP)
GrSDS	Service Provider – WP4.4	Publish services	SOAP
A4C	VO manager – WP4.4 (A4C Client within the component)	Authentication Send user profile for Grid Service Authorization	Diameter
A4C	Metering – WP4.3	Send Metering Information	Diameter

Table 2 - WP4.2 towards lower layers

WP4.2 subsystem	Connections with WP4.1		
	Lower layer subsystem	Purpose	Method of connection
Context manager (SIP PA)	MT SIP PUA	Gather context information (precense ++).	SIP PUBLISH Additional mechanisms to collect user context information can be required.
A4C	SIP Registrar	- User Authentication - Sending user authorization requests - SIP session Accounting Authorization - Accounting	Diameter

WP4.2 subsystem	Connections with WP4.1		
	Lower layer subsystem	Purpose	Method of connection
A4C	SIP registrar/proxy?	Authorisation Accounting	Diameter

8.2. WP4.3 Interfaces

Table 3 – WP4.3 interfaces towards higher layers

WP4.3 subsystem	From WP4.4			
	Higher layer subsystem	layer	Purpose of invocation	Method of connection
Execution manager	BP Enactment	4.4	Execution request – *and execution finish?*	SOAP
SLA Enforcer	BP Enactment	4.4	To order an action according to a policy	WS-Request
SLA enforcer	SLA High Level	4.4	To access to the contract info	WS-Request

Table 4 - WP4.3 interfaces towards lower layers

WP4.3 subsystem	………… WP4.2 / 4.1 …………			
	Lower layer subsystem invoked	Layer	Purpose of invocation	Method of connection
EMS	QoS B	4.1	To use network services of WP4.1	SOAP
Monitoring	A4C	4.2	Updating the A4C info with respect to WP4.3 operations	SOAP
Monitoring	?	4.1	Update information about network availability in current moment	SOAP
Data manager	?	4.1	Request network services	?
Metering	A4C	4.2	Receive information on resource usage measurements	SOAP
SLA Enforcer	A4C	4.2	To send a notification of violation	Notification
All components	GrSDS	4.2	To discover the services that need to be used in the various execution phases	UDDI

8.3. WP4.4 Interfaces

Table 5 - WP4.4 interfaces towards lower layers

WP4.4 subsystem within (in almost all cases a Web or Grid Service)	………… WP4.3 / 4.2 / 4.1 …………			
	Lower layer subsystem	Layer	Purpose of invocation	Method of connection
VO Management (OpVOBroker)	Service Discovery Server	4.2	Search a service Note: Each service (simple and complex) should be exposed as service with its interface	WS-Request
VO Management (VOManager)	A4C	4.2	To confirm user authentication Retrieve user profile	Diameter
BP Enactment (Workflow Man-	Context Manager	4.2	This will allow context to be used to determine the circumstances in which a service would be invoked	WS Request

ager)				
BP Enactment (Workflow Manager)	Service Discovery Server	4.2	Search a complex service; based on metadata provided by the customer	WS-Request
BP Enactment (Workflow Manager)	EMS	4.3	Invoke the acquired service	WS-Request
BP Enactment (Workflow Manager)	SLA Enforcement	4.3	Subscription to violation events	WS-Request
SLA High Level	EMS	4.3	To obtain QoS parameters of a service and to specify agreed parameters to make reservation	WS Request
SLA High Level	A4C	4.2	To set the charging policies, specified in the SLA contract, which should be applied by the A4C	WS Request
SLA High Level	Policy Manager	4.3	To obtain policy metrics (mapping from high to low level parameters)	WS-Request
SLA High Level	Registry	4.2	To order storing the Contract	WS-Request

9. References

[1] Johannes Ernst and Joaquin Miller, *Requirements for a Scalable Mobility Architecture*, http://netmesh.org/papers/mobility-architecture/netmesh-mobility-architecture-requirements.pdf

[2] T. Goss-Walter, *An Analysis of the UNICORE Security Model* http://www.Gridforum.org/documents/GFD.18.pdf

[3] T. Sandholm, *The SweGrid Accounting System,* http://www.pdc.kth.se/Grid/sgas/docs/presentations/NorduGridSGAS.pdf

[4] Access Grid , http://www.accessGrid.org

[5] Intel® Mobile Application Architecture Guide, http://www.intel.com/cd/ids/developer/asmo-na/eng/61193.htm

[6] Tai M. Chung, *Design of high performance networking platform considering mobile environment*, http://www.Gridforumkorea.org/workshop/2004/2004_winter/data/Tai-MyungChung.ppt

[7] Piotr Grabowski et al., *Access from j2me-enables mobile devices to Grid services* In Mobility Conference 2004

[8] L.W. McKnight and J. Howison, *Towards protocols for wireless Grids.* In International Conference on Computer Communication and Control Technologies, volume 648, 2003

[9] OMA-RD-Identity_Management_Framework-V1_0-20050202, 02 Feb 2005, page 15

[10] Liu, Feng et al.: *WSIP – Web Service SIP Endpoint for Converged Multimedia/Multimedia Communication over IP*. IEEE ICWS '04

[11] Schulzrinne, H., IETF RFC3966, *The tel URI for Telephone Numbers*, RFC3966, December 2004.

[12] W. Saabeel and T.M. Verduijn and L. Hagdorn and K. Kumar, *A Model for Virtual Organisation: A structure and Process Perspective* in Electronic Journal of Organizational Virtualness

[13] T.J. Strader and F. Lin and M.J. Shaw, *Information Structure for Electronic Virtual Organization Management in Decision Support Systems*, volume 23, 1998, pages=75-94

[14] Kostas Petropoulos and others, *D1.3 Conceptual Model of the LAURA Prototype - Definition of Functionalities*, LAURA IST Project 2003

[15] Ian Foster and Carl Kesselmann and Steve Tuecke , *The Anatomy of the Grid: Enabling Scalable Virtual Organizations*, International Journal of Supercomputer Applications 2001

[16] T. Dimitrakos, David Golby, Paul Kearney, *Towards a Trust and Contract Management Framework for dynamic Virtual Organisations* in eAdoption and the Knowledge Economy: eChallenges 2004

[17] Stefan Wesner, Lutz Schubert, Theo Dimitrakos, *Dynamic Virtual Organizations in Engineering*, German Russian Workshop Stuttgart 2005

[18] Roman Tirler (Editor), *Next generation Grids*, European Commission, 2003

[19] D2.3.1 Testbed Description, Version 1.0

[20] D2.3.2 Validation Scenarios, Version 1.0

[21] B. Campbell et al., *Session initiation protocol (sip) extension for instant messaging*, IETF, 2004

[22] Rosenberg, J., Peterson, J. Schulzrinne, H. And G. Camarillo, A. Johnston, R.Sparks, M. Handley, E. Schooler: *SIP: Session Initiation Protocol.* RFC3261, June 2002.

[23] R. Shacham, H. Schulzrinne, Columbia University, S. Thakolsri, W. Kellerer, DoCoMo EuroLabs, February 14, 2005, *Session Initiation Protocol (SIP) Session Mobility*, Internet draft (draft-schaman-sipping-session-mobility-00), February 2005.

[24] Rosenberg, J., Peterson, J. Schulzrinne, H. And G. Camarillo, *Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)*, RFC 3725, April 2004.

[25] R. Sparks, *The Session Initiation Protocol (SIP) Refer Method*, RFC3725, April 2003.

[26] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, J. Arkko. *Diameter Base Protocol*, RFC 3588, September 2003

[27] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, D. Spence. *Generic AAA Architecture.* RFC 2903, August 2000.

[28] M. Koutsopoulou, A. Kaloxylos, A. Alonistioti, L. Merakos, K. Kawamura. *Charging, Accounting and Billing Management Schemes in Mobile Telecommunication Networks and the Internet*, IEEE COMMUNICATIONS, Vol.6/2004, pages 50-58,

[29] Modeling Stateful Resources With Web Services http://www-106.ibm.com/developerworks/library/ws-resource/ws-modelingresources.pdf

[30] Web Services Distributed Management: Management Using Web Services (MUWS 1.0) Part 1 http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part1-1.0.pdf

[31] Web Services Distributed Management: Management Using Web Services (MUWS 1.0) Part 2 http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part2-1.0.pdf

[32] Web Service Base Notification http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf

[33] Akogrimo Deliverable, D2.3.1 Test-bed definition

[34] Akogrimo Deliverable, D2.2.1 vol2. State of the Art Report

[35] M. Baker, M. Nottingham: “The "application/soap+xml" media type”, RFC3902, September 2004

[36] S. Berger, H. Schulzrinne, S. Sidiroglou, X. Wu: “Ubiquios Computing Using SIP”, NOSSDAV 03, May 2003

[37] N. Deason: “SIP and SOAP”, draft-deason-sip-soap-00, June 2000

[38] N. Deason: “SIP and SOAP White Paper”, Ubiquity Software Corporation, May 2001

[39] I. Foster, C. Kesselmann, J. M. Nick, S. Tuecke: “The Physiology of the GRID”

[40] R. Fielding, J. Gettys, J. Mogul, H. Rystyk, L. Masinter, P. Leach, T. Berners-Lee: “Hypertext transfer protocol – HTTP/1.1”, RFC2616, IETF, June 1999

[41] J. Rosenberg, H. Schulzrinne: “Internet telephony: Architecture and protocols”, IEEE Network, Vol. 13, pp. 18-23, May/June 1999

[42] J. Rosenberg, H. Schulzrinne: “Third party call control in SIP”, Internet Draft, IETF, March 2000, Work in Progress

[43] H. Schulzrinne, J. Rosenberg: “Internet telephony: Architecture and protocols – an IETF perspective”, Computer Networks and ISDN Systems, Vol. 31, pp. 237-255, Feb. 1999

[44] H. Schulzrinne, J. Rosenberg: "The session initiation protocol: Internet-centric signaling", IEEE Communications Magazine, Vol. 38, Oct. 2000

[45] H. Schulzrinne, E. Wedlund: "Application Layer Mobility Using SIP", ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 4, Issue 3, pp. 47-57, July 2000

[46] A. Skonnard: "Understanding SOAP", MSDN (http://msdn.microsoft.com/webservices/understanding/webservicebasics/default.aspx?pull=/library/en-us//dnsoap/html/understandsoap.asp), March 2003

[47] UDDI Technical Whitepaper, http://uddi.org/pubs/uddi-tech-wp.pdf, October 2004

[48] UDDI Specs, http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm

[49] W3C: SOAP 1.1 Note, http://www.w3.org/TR/2000/NOTE-SOAP-20000508/, May 2000

[50] W3C: SOAP Version 1.2 Part 1: Messaging Framework (W3C Recommendation 24 June 2003), http://www.w3.org/TR/soap12-part1/, June 2003

[51] W3C: SOAP Version 1.2 Part 2: Adjuncts (W3C Recommendation 24 June 2003), http://www.w3.org/TR/soap12-part2/, June 2003

[52] Introduction to Web Services Technologies: SOA, SOAP, WSDL and UDDI, http://www.informit.com/articles/article.asp?p=336265, September 2004

[53] Web Service Addressing, http://www.w3.org/Submission/ws-addressing/

[54] Web Service Definition Language, http://www.w3.org/TR/wsdl

[55] Security Assertion Markup Language, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[56] eXtensible Access Control Meta Language, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

[57] WS-Agreement, http://www-unix.mcs.anl.gov/~keahey/Meetings/GRAAP/WS-Agreement%20Structure.pdf

Annex A. Summary of Requirements from the Testbeds and Validation Scenarios

Within Akogrimo there are currently two validation scenarios defined. The following section gives an overview about these scenarios. The scenarios are based on Deliverable 232. More detailed information can be found there.

A.1. e-Health: Heart Monitoring and Emergency Response Process

The organizational framework where the process is executed consists of a university hospital, regional hospitals, medical specialists, general practitioners, emergency medical services and an emergency dispatch center establishing a regional health network. The health network is headed by the university hospital and provides telemedicine services to the partners and the patients attended by partners of the network. These services are focused on particular diseases and risk groups.

For providing the services the regional health network collaborates with a health service provider (HSP) and a network operator (NO). The NO hosts an infrastructure to provide telemedicine services over its network. He offers computational, network and data collection services. The HSP distributes the telemedicine equipment, provides advanced medical analysis services, configures application services specific to the health network's needs and is responsible of the patient-side accounting and billing.

One of the services provided by the regional health network addresses the patients with an increased risk on suffering a heart attack or apoplectic stroke. Patients belonging to this risk group are registered with a heart monitoring and emergency service (HMES). The main objective of this service is the early recognition of heart attacks or apoplectic strokes and a proper treatment as fast as possible. The service comprises two different cases. During the treatment process of a patient in this risk group the attending physician can recommend the application of a permanent monitoring service to observe the cardiac function that assures a more detailed diagnosis, a fast alert triggering, and a disease-specific response in the case of an emergency. In the second case the patient receives a non-permanent monitoring and emergency trigger device that is activated by the patient if he feels a problem in his cardiovascular system. In the following only the first case will be considered.

Two additional groups can be considered to be involved in order to improve the performance and quality of the heart monitoring and emergency process: first responders and the police.

There is a very simple idea behind the first responder concept. Particularly, in case of heart diseases, a treatment (e.g. defibrillation) of the patient has to be done as fast as possible. This task is to be accomplished by emergency medical services. Due to other emergency efforts of the emergency medical services and geographical distances to the emergency site a fast reaction to the emergency situation and alert can't be guaranteed. To circumvent this problem, different persons having at least a basic medical education can be involved as first responders into the emergency handling. As a first responder, for instance, firefighters, mountain rescue services, nursing services, or regular medical staff currently off-duty can be considered. First responders are alerted over the organization they are working for. The geographical location and the availability of a first responder have to be considered to decide who will be alerted. Different to regular medical staff and organizations, there exist no consistent funding models for first responders. Since the service of first responders is not regulated by law health insurance companies don't reimburse the efforts. For that reason public authorities, directly or indirectly over the organizations the first responder is engaged with (e.g. fire department) bear the costs. On the other hand the organizations mentioned above receive donations for the first responder services they provide.

The police services can be involved in several tasks. Police men usually receive a medical training and can act as a first responder, too. They are not alerted to take a first responder role, but can be involved in an emergency situation that has led to an accident. A second task could be to provide latest traffic information as an input for dynamic navigation services used by ambulances to plan the quickest route to the emergency site. Finally, depending on the national regulations, police services could be responsible for the management of several emergency resources. In this case the management task is distributed between the police and the emergency dispatch center. If necessary, the whole management capabilities could be transferred from the emergency dispatch center to a mobile police unit. The reimbursement of police service is typically done by national authorities. Only the first case, the provisioning of traffic services, is possible for utilization.

A.1.1. e-Learning: Field Trip

The scenario described here will be focused on archaeological studies of Greek civilization lived since 500 BC. A students group of a archaeological course is involved in researches, studies and results sharing on the Field Trip project in order to support their Archaeology spring exam by using the modern Information Technologies (i.e. 4th generation PDA) for accessing to an e-Learning platform. This course is characterized by some on-side studies, where a students group can fetch the real scenario of the archaeological place of today time by means of cameras, registers, electronic notes, etc.; some ubiquitous (collaborative) analysis, where students can analyze data provided onsite or gained from other sources (e.g. Web) and eventually to discuss some issues with other course students; and teacher support, where one ore more teachers can be involved in discussions or issues clarification.

For providing the Field Trip Service, there is collaboration among a Core e-learning service provider, a Grid Service provider and a Local Network provider. The Local Network provider hosts an infrastructure to provide access the other services over its network. The Grid Service Provider offers computational, network and data collection services. The Core e-learning service provider offers authentication of users, instantiation of other services, management of user profiles and groups, and search of information.

Services are billed to users by the Local Network provider and the Core e-learning service provider. Grid services are billed to the Core e-learning provider, which charges them to users.

A.1.2. Requirements

The requirements can be separated in application specific requirements and role specific requirements according to Akogrimo D232.

A.1.2.1. Application-specific Requirements of the e-health scenarios

The e-Health application will collect different types of data from a wide variety of services and/or applications and presents them in an appropriate way to users of the system; different users may have different roles. As such, it will require the availability of a number of services:

- The patient local monitoring equipment.
- The various patient's partial medical records held by multiple organisations
- A location service, which can be provided by a satellite or by triangulation of the patient's mobile phone. This service will also track participants other than the patient (e.g. the ambulance).
- A service for determining the hospital the patient should be taken to, taking into account the bed space of nearby hospitals.
- Language recognition and translation service for translating medical data of foreign patients.
- A filter which will search through the entire patient's record and choose only the information which is relevant for the pertaining emergency.
- A service for obtaining the patient's symptoms in an optimal fashion, considering the available emergency information.
- A diagnostic service which will use all the information available to reach an initial diagnosis. This information must be made available to the medical team waiting for the patient in the hospital prior to his arrival.

For the communication of the participants, the application must:

- Provide a heterogeneous mobile environment with quality of service
    - Allow the use of various kinds of devices, such as mobile phones, PDAs, portable computers and desktop computers.
    - Allow access to the application from virtually anywhere.
    - Ensure that high priority data flows will not be affected by other lower priority flows.
- Allow the sharing of different kinds of data among the participants.
- Provide an infrastructure for audio (and video) conferencing among the participants.
- Provide an organized working environment in which every participant can know his role, as well as the current status of the process he is involved with.

In addition the following requirements are typical for all kind of e-Health application:

- Reduce the usage and transport of patient data on a minimum
- Make patient data pseudonymous or anonymous whenever possible
- Assure the patient's consent when patient related data are transferred or processed
- Encrypt and sign patient data during transfer
- Assure the correctness of provided, visualized medical data and images
- Be complied to medical data exchange standards HL7 and DICOM
- Assure non-repudiability of functions executed by the HMES

A.1.2.2. Application-specific Requirements of the e-learning scenario

An e-Learning application is intended as the end global service which collects other simpler services and is provided to the user (students, teacher and experts) through web portal. These simpler services are such as:

- Service which collects user information (preferences and profiles) and provides user information about other students group with affinity to Field Trip project in order to share experiences and similar projects info.
- E-Learning service that adapts the learning process to the specific user based on collected user information.
- Information search service, which allows users get some information from a multimedia repository and from the Web.
- Simulation services to understand better the concepts and knowledge shared by all students.
- Text to speech service which allows send speech data through user device (for instance, through PDA device).
- Digital libraries consulting service that provides the students a way to fill all the gaps in their knowledge.
- Meeting services, that is, collaborative tools for students to discuss some issues with other course students, teachers, experts, etc.
- E-Learning service that evaluates the status of the learning process through direct and indirect user feedback and adapt the workflow accordingly.

With respect to the users, the e-Learning application has to provide an environment where:

- Users learn with interactive methods based on the use of multimedia technologies and Internet.
- Users are authenticated, using their preferences and profiles to adapt the learning process to each user.
- Users can exchange information and collaborate with other students with affinities to the Field Trip project.
- Users can meet via videoconference.
- Users can access the Grid system from different devices (mobile phones, PDAs, laptops and PCs) to upload, process and download multimedia data.
- Users can access the system from any place

With respect to the information, the e-Learning application has to be deployed on a platform where:

- Information can be integrated from different sources.
- Information can be indexed with semantic metadata.
- Information can be searched efficiently.
- Information can be stored in a multimedia repository
- Speech information can be recognized and translated to text.
- Information can be taken to perform simulations
- Information about sessions and context awareness is handled.

A.1.2.3. Role-specific Requirements

A.1.2.3.1. Application Provider

E-Health

From an application service provider point of view the follow requirements can be figured out:

- Configuration of the application service by orchestrating single services during design-time and run-time
- Support of different accounting and billing models (by bandwidth consumption, by time, by number of usage)
- Support of multiple security infrastructures
- A clearing system must exist to support between the service provider
- Support the life-cycle management of application services

E-Learning

The Application Provider must be able to meet the following functional requirements obtained from the analysis of the e-Learning scenario:

- Integrate information sources, enhanced with semantic information
- Use multimedia technologies and the Internet to improve the quality of learning
- Support information exchange and collaboration
- Support an interactive learning process
- Support user preferences and profiles and provide the ability to adapt the learning process to the specific user
- Support different user roles (e.g. teacher, student)
- Provide session maintenance and context awareness
- Provide a user friendly interface that allows the user to access the system from diverse devices, upload and download multimedia data and use the provided services
- Use direct and indirect user feedback to evaluate the status of the learning process and adapt the workflow accordingly

A.1.2.3.2. Platform Providers and Operators

The Platform Providers/Operators provide and run the mobile Grid platform Akogrimo or middleware (mobile network middleware, Grid infrastructure and Grid Application support layers). The Telco buys platform components from commercial vendors and integrate them into existing infrastructure. A Telco will have interest in offering functionality that is common to a wide range of services and applications in both the mass and business market. The Telco will also sell access to interfaces allowing 3rd party providers to offer both domain specific applications and value added services. Platform providers and operators provide the bridge between the mobile world and Grid world.

E-Health:

Functional requirements:

Collaboration and sharing across administrative domains

To handle the medical emergency a team of people must work closely together. This requires sharing data and services among mobile individuals belonging to different organisations who have different roles and responsibilities. Emergency operator, first responder, ambulance staff, medical staff and police officers are required to cooperate and share resources. Persons must be authenticated and granted proper access right. This implies the following requirements for Mobile Virtual Organisations.

- Creation and deletion of Virtual Organisations must be supported
- Adding and removing resources (service, user etc) to the VO must be supported
- It must be possible to grant access to specific resources in the VO (e.g. based on the role of a user, his/her task, etc.).

As soon as a Virtual Organization is in place, cooperation will be happening within a workflow of steps dealing with different parts of the task. In order to allow human resources to participate in such a workflow, flexible cooperation support tools such as web-based shared workspaces will be needed. Ideally the tasks to be performed should be delivered to the right persons at the right time with the necessary information and the application functions needed.

- The Akogrimo platform must support creation, deletion and modification of Workflow templates.
- The platform must support workflow instantiation and execution.

Discovery of services

The e-Health use cases reveal a need to find services. For instance it is necessary to locate an emergency service, a translation service, a service to compute the best route, services providing traffic information etc. Generally, this implies that the following functional requirements to the Akogrimo platform:

- The platform must support registration of service providers (service providers must be authenticated and authorised).
- The platform must support service advertisement.
- The platform must support service discovery. There should be an interface offering a flexible search for services based on their semantics.

Context management

The e-Health use cases have shown that context information is vital to deal with the medical emergency. First, context information about patient is needed to determine contact information, his location, age, sex, spoken language etc. Second, context information such as role, location and presence is required to find medical staff needed to handle the situation. The requirements identified for context information are listed below.

- The platform must gather basic context information.
- Context information might change during service provisioning. The platform must take care of changes and notify applications and services accordingly.
- The platform must allow discovery of humans based on context information (e.g. find an available medical doctor in a certain physical area).
- The platform should offer the following context information:
  - Contact information (name, address, e-mail, phone number)

- Role (medical doctor, nurse etc.)
- Age
- Sex
- Nationality
- Religion
- Availability
- Spoken languages
- Location (mapped to street address)
- Terminal capabilities
- Local services
- Network capabilities

Network services

There are several mobile actors in the e-Health use cases, e.g. patient, first responder, paramedics, police officers and medical staff. From the use cases it is recognized that the network must support real-time services such as telephony, sending urgent messages, interactive collaboration tools etc. Additionally, non-real time services such as file transfer and messaging must also be supported by the network.

- The platform must support mobile and nomadic users.
- The platform must support real-time interactive communication services (telephony, video, urgent messages etc).
- The platform must support non-real time communication services (file transfer, messaging services etc.)

Basic platform requirements

This section summarises additional functional requirements from the use cases.

- The emergency service needs to identify and authenticate the patient.
- The solution should be flexible with respects to models for billing and pricing. It could be the case that the caller is not charged for using the emergency service.
- The platform must offer services for logging communication. For instance, emergency calls needs to be logged.

Non-functional requirements:

The e-Health use cases have a level of urgency, sensitivity and privacy that implies strict non-functional requirements.

Reliability:

- The platform must have very high reliability (Telco grad) to be applicable to medical emergencies.
- The platform must comply with data protection laws, privacy regulations and other security requirements.

Performance and scalability:

- The platform must support performance required by applications. E.g. it is required that the platform never blocks emergency calls and that the Emergency applications have good performance at all times. It must support services that have real-time performance requirements such as telephony, sending urgent messages, interactive collaboration tools etc. Given the urgency of the situation even non-real time service (such as file transfer) have strict requirements on performance (throughput)
- The platform must be scalable. The performance and reliability of the platform must scale.

E-Learning

The following list shows the requirements that have been got from e-Learning scenario from operator point of view:

- Operators have to provide access to the mobile Grid from different devices such as small ones like PDAs to laptops and PCs.
- Operators have to provide access to the mobile Grid application thanks to the mobile and wireless technologies provided by network technology providers.
- Operators have to be able to select appropriate network taking into account necessary bandwidth, price, etc.
- Operators must own fixed, wireless and mobile infrastructures.
- Operators have to provide available resources according to service legal agreement, user preferences (like price) and context information.
- Operators have to integrate external services provided from other companies, such as repository Grid services, simulation services, etc.
- Operators have to provide access to multiple resource at the same time, such as simulation and collaborative sessions.
- Operators have to provide end user information about other students group with affinity to Field Trip project in order to share experiences and similar projects info.
- Operators have to support collaborative work for students to discuss some issues with other course students, teachers, experts, etc.
- Operators have to assure user mobility, that is the possibility of a student/teacher/tutor to move and to use any terminal with his own preferences and settings.
- Operators have to develop security mechanisms in general for all the applications, such as authentication /authentication mechanisms in the overall system.

Other functional requirements not derived from scenarios study, but desirable are:

- Operators have to provide session in order to allow the user to capability to continue active session during network or terminal change.
- Operators have to provide a unique and overall invoice.

A.1.2.3.3. Service Providers

In general, a Service Provider focuses on development of services that can be utilized on various platform and systems, in this case on Akogrimo platform. It provides its services to Operators and Application Providers, smaller units of the application than Application Providers who generates scenario specific applications/tools using Akogrimo technology. Application Providers integrate the services offered by service Providers.

E-Health

This section describes services needed by the e-Health scenario.

- A location/positioning service that provides location information is needed. Different types of location technology (RFID, WLAN, GSM, GPS; etc) must be deployed to provide location information both indoor and outdoor
- There is a need for a service providing maps (e.g. to plot location of patient).
- A translation service that provides translation of voice to a different language is required. A human translator could offer this service. A futuristic vision would include a software-based service that provides real-time translation.
- There is a need for basic services for processing and analyzing multimedia such as adaptation of media to terminal capabilities.
- A route planning service is needed (e.g. to find the best route to a hospital).
- The need for services for traffic (road) monitoring and management is recognized.
- Medical specific services (e.g. ECG service)
- As soon as a Virtual Organization is in place, cooperation will be happening within a workflow of steps dealing with different parts of the task. To handle the medical emergency of a patient, a team of people must work closely together. Ideally the tasks to be performed should be delivered to the right persons at the right time with the necessary information and the application functions needed. Hence, a flexible domain specific application for collaboration and cooperation is required (including instant messaging, blackboards, document sharing, geographical information system, video and phone sessions, medical records, medical diagnosis etc.). The following service could be a part of such an application:
    - Instant messaging services (buddylist) allowing members of the VO to see the context of one another and to exchange information is required.
    - Blackboards and document sharing are needed.
    - Phone / (Video) conferences services to establish communication between persons in the VO are needed. These services should consider the context of users such that sessions are established to users that are available, to the most suitable terminal etc.
    - Geographical information system for visualization relevant information is required.

There are some additional services that can be provided by specialised service providers such as:

- Access network provider
- Transport network provider
- A4C provider
- Context information provider
- Service discovery provider
- Telephony service provider

E-Learning

Service Provider must be able to meet the following functional requirements obtained from analysis of e-Learning scenario:

- To provide videoconference service.
- To provide search engine.
- To provide simulation services.
- To provide speech recognition services.
- To provide digital libraries consulting services.
- To provide multimedia repository Grid service with metadata.
- To provide authentication/authorization methods based on biometric data in order to access to the service.
- To provide methods to manage user profile.
- To provide methods to manage context information.
- To integrate existing services/app within the platform system and develop new services according to platform needs.

Other functional requirements not derived from scenarios study, but desirable are:

- To provide pricing and billing methods for charging of service usage, such as pay per usage, pay per success, etc.
- To assure privacy issues and keeping the laws concerning personal data protection.

A.1.2.3.4. Technology Providers

Technology Providers offer technology as an enabler for the Mobile Grid Application (rent and/or sold it).

Related to **Network Technology Providers** they provide network and communication services (network capabilities) and are strongly related to the Platform operator.

E-Health:

The Telco will not develop the platform themselves. Commercial platform components must be available from commercial vendors such that Telco's can integrate them into existing infrastructure. It is important that the platform offers modularity, scalability and interoperability in the context of complex and heterogeneous environments. This allows the Telco to make step-wise infrastructure investments and to cooperate with 3rd party providers. The following are typical requirements:

- The platform must offer standard external API for service development. These interfaces should be independent of underlying network technologies and hide underlying complexity from software developers.
- The platform should be based on common interfaces over standard, open, general-purpose protocols (non-proprietary).
- The platform should consist of standard platform components (HW, SW)

- The platform must be modular and extensible.
- It must be easy to configure and manage services and platform components.

E-Learning:

Next, a list of Network Technology Provider requirements got from e-Learning scenario is shown:

- To provide high speed fixed, mobile and wireless connections that can be accessed easily and enables real-time functionality.
- To provide technology that allows a network node to change its physical location with minimal consequence, changing the point of network connection (terminal mobility).