D3.1.3

The Mobile Grid Reference Architecture

WP3.1 Overall Architecture Definition and Layer Integration V2.0

Lead Editor: Jürgen Jähnert, USTUTT/RUS

10/10/2006

Status: Revised Version

SIXTH FRAMEWORK PROGRAMME

PRIORITY IST-2002-2.3.1.18

Grid for complex problem solving

Proposal/Contract no.: 004293

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

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

a. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
b. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works.
c. For the avoidance of doubt, where the work is a musical composition:
    i. **Performance Royalties Under Blanket Licenses**. Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.
    ii. **Mechanical Rights and Statutory Royalties**. Licensor waives the exclusive right to collect, whether individually or via a music rights society or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).
d. **Webcasting Rights and Statutory Royalties**. For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

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

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

a. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any credit as required by clause 4(b), as requested.
b. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or Collective Works, You must keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or (ii) if the Original Author and/or Licensor designate another party or parties (e.g. a sponsor institute, publishing entity, journal) for attribution in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; the title of the Work if supplied; and to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work. Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

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

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

7. Termination

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

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

8. Miscellaneous

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

Context

Activity 3	**Design**
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 D3.1.2 of WP3.1 in relation to the following Workpackages and their Deliverables:
	D3.1.3 is influenced by
	• WP1.1 Market and Regulations, WP2.2 Environment, WP2.3 Test-bed Definition
	  o D2.1.1
	  o D2.1.2
	  o D2.2.1
	  o D2.2.4 Report on State of the Art
	  o D2.3.1 Test-bed Definition
	  o All available Deliverables from AC4
	• D3.1.1, which can be regarded as first version of this document
	• D3.1.2, which can be regarded as second iteration of this document
	• D5.3.1 analysing the architecture based on D3.1.1
	• 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
	Further, D3.1.2 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'.
	Additionally this document will directly influence the physical instantiation to be done within AC5.
	Finally, this document provides the core for the Akorimo white paper.

List of Authors

Jürgen Jähnert	USTUTT
Patrick Mandic	USTUTT
Robert Piotter	USTUTT
Ignaz Müller	USTUTT
Ruth del Campo	USTUTT
David Lutz	USTUTT
Annalisa Terracina	Datamat
Giuseppe Laria	CRMPA
Gaeta Angelo	CRMPA
Romano Nadia	CRMPA
Gaeta Matteo	CRMPA
Julian Gallop	CCLRC
Brian Ritchie	CCLRC
Damian Mac Randal	CCLRC
Antonis Litke	NTUA
Kleopatra Konstanteli	NTUA
Brynjar Age Viken	Telenor
Per Oddvar Osland	Telenor
Eduardo Oliveros	TID
Nuno Inacio	IT-Av
Victor Villagra	UPM
Antonio Cuevas	UPM
Vicente Olmedo	UPM
Cristian Morariu	Uni ZH

Review: Stefan Wesner, USTUTT

Table of Contents

List of Figures

List of Tables

1. Introduction

This document describes the consolidated Akogrimo architecture after 2 years of project duration. Despite the fact that formally no further iterations are planned according to the project plan, it is expected, that some smaller refinements will be carried out during the last project year. One reason is especially caused by the fact that a new partner has been included in PM 24, which might either confirm the architecture or contribute new requirements to the overall Akogrimo architecture.

This document is intended as a consolidated version of the already provided architecture Deliverables D311 and D312. It should be seen as consolidated version comprising the high level architecture of Akogrimo. However, the document avoids repeating details of architectural components which might have been reported already in either the detailed architectural design Deliverables (see WP4.1, WP4.2, WP4.3, WP4.4. Within this context the goal of this document is to inform the reader about the generic architecture, the presentation of the key component and the high level interaction between these key components.

Due to the Akogrimo project plan, the goal of this document is next to the definition of a generic architecture the clear assignment of "working areas" to four different Workpackages within Akogrimo representing the Akogrimo Network, the Akogrimo Network Middleware, the Akogrimo Grid Infrastructure and the Akogrimo Grid application support. It should be mentioned, that each of this four Workpackages worked out a more detailed architecture, which is documented within the Deliverables produced by these Workpackage.

While each layer Workpackage obviously addresses its interactions with other layers, Workpackage 5.1 is concerned with the integration of these layers and will be producing its own deliverable on architecture integration comprising the physical architecture of the deployed instantiation based on the scenarios. This document therefore provides a summary of the separate single layer architecture documents and provides additional cross-layer information that will feed into the deliverable D5.1.1 Architecture Integration Report.

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.

Consequently Akogrimo assumes a pure IP-based underlying network infrastructure overcoming entirely the historically voice driven concepts and mechanisms. This leads to an architecture, which could be immediately deployed in Unlicensed Mobile Access (UMA) environment such as hot-spot infrastructures. In the mid term it is the assumption that along the convergence of networks Telecom sector eventually will come to a network technology that is similar.

The basic concepts and terms required in addressing this challenging task to build an overall architecture for such a highly dynamic environment are defined in this document. The approach chosen 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, such as different types of mobility, are described in greater detail.

A specific solution of the problem of how to allocate responsibilities to different components 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 problems around identity and security management, cross organisational accounting, the handling of context and the

interrelation of signalling as basic protocol element. This is reflected in selected key scenarios and user cases with are presented at the end of the document.

The key objective of this document is to highlight the novel features of the architecture, primarily arising from the focus on mobility of users and services. This starts in chapter 2 with the description of the full Akogrimo system including the description of key components and concepts.

Chapter 3 then focuses on selected cross layer issues, which are from an overall integration point of view worth to be mentioned. Here, service description, signalling issues and security play a central role.

Finally, Chapter 4 illustrates the interworking of the overall architecture based on selected key scenes.

In chapter 5 a conclusion is given.

2. Akogrimo Higher Layer Architecture

2.1. Architectural Goals and Constraints

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.

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.

2.1.1. 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.

The Akogrimo consortium believes that the chosen Akogrimo approach relies 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). For typical deployments of Grids the number of users is several hundred maybe some thousand users. Furthermore as rather 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

Area	Specific Challenges and Opportunities
	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. 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. 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 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.

2.2. The Generic Akogrimo Architecture

2.2.1. High level Conceptual Architecture

Figure 1 shows the Akogrimo Architecture on a very high and conceptual level.

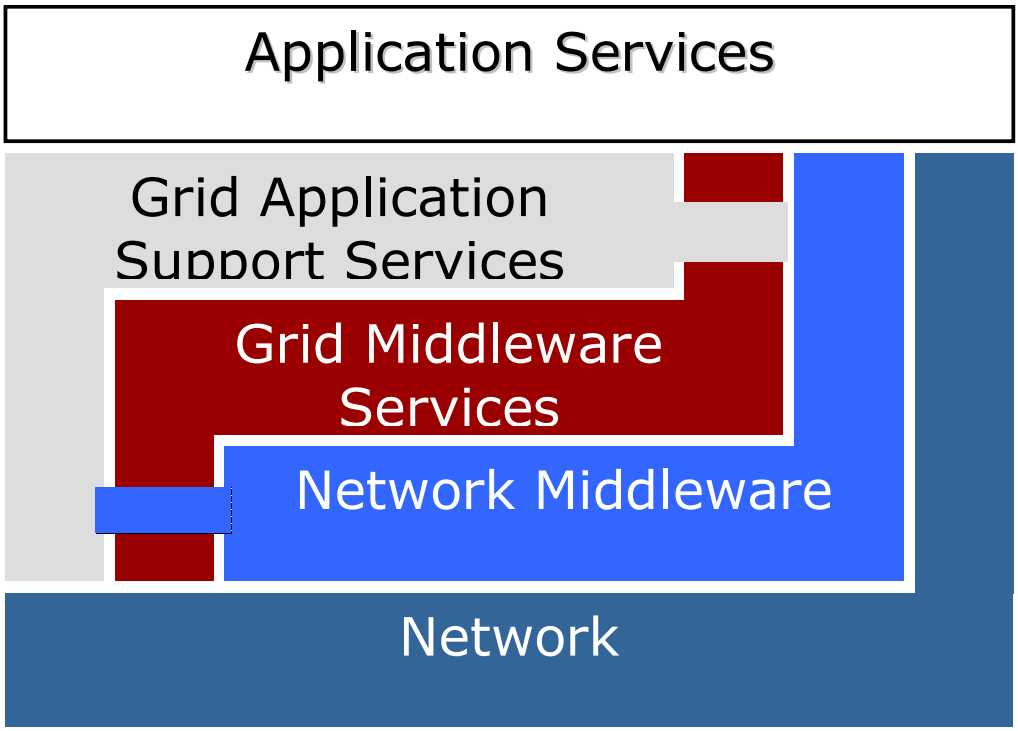

Figure 1: Akorgimo Conceptual Architecture

The figures shows basically four areas which have been named in the initial phase of the project as formal "layers", but during the evolution of the architectural design it became more and more obvious, that such a strict layered approach would not meet the requirements to the overall Akogrimo architecture.

The lowest "area", the *network* interfaces with all the other areas, but has most interaction with the *network middleware* providing network services like AAA, session control and network context in conjunction with the *network*.

Basically promoted by the traditional "Grid" community, two additional areas, the Grid Middleware Services providing a infrastructure for a service provider as well as the application support area providing VO infrastructure services complete the overall conceptual architecture supporting the applications running "on the Akogrimo platform".

Generally, in Akogrimo all services (including the traditional services offered on the network layer) are virtualised via Web Services. And the role sharing is more function based rather than a protocol stack.

A further major principle to be mentioned at this early stage is to a bi-directional exchange of all information between the layers such as identity, context, .., which will be explained in deeper details within this document.

Before the basic building blocks are presented, in the following section the key term Mobile Dynamic Virtual Organization, a very basic concept in Akogrimo, is introduced.

2.2.2. Mobile Dynamic Virtual Organization

In Akogrimo the term 'Virtual Organization' (VO) 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 and the requirements from business to business interactions in 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

2.2.1. Basic Building Blocks

The generic Akogrimo architecture has been initially introduced in D311 and refined in D312. In these deliverables, a layered approach has been presented. This section consolidates the information of the previously provided deliverables on a generic level. Here, the strict layered approach slightly disappears.

Figure 2 shows the conceptual architecture and the main building blocks. Here the 5 key building blocks are the network provider, the service provider, the Operational VO, the Base VO and the Customer domain.

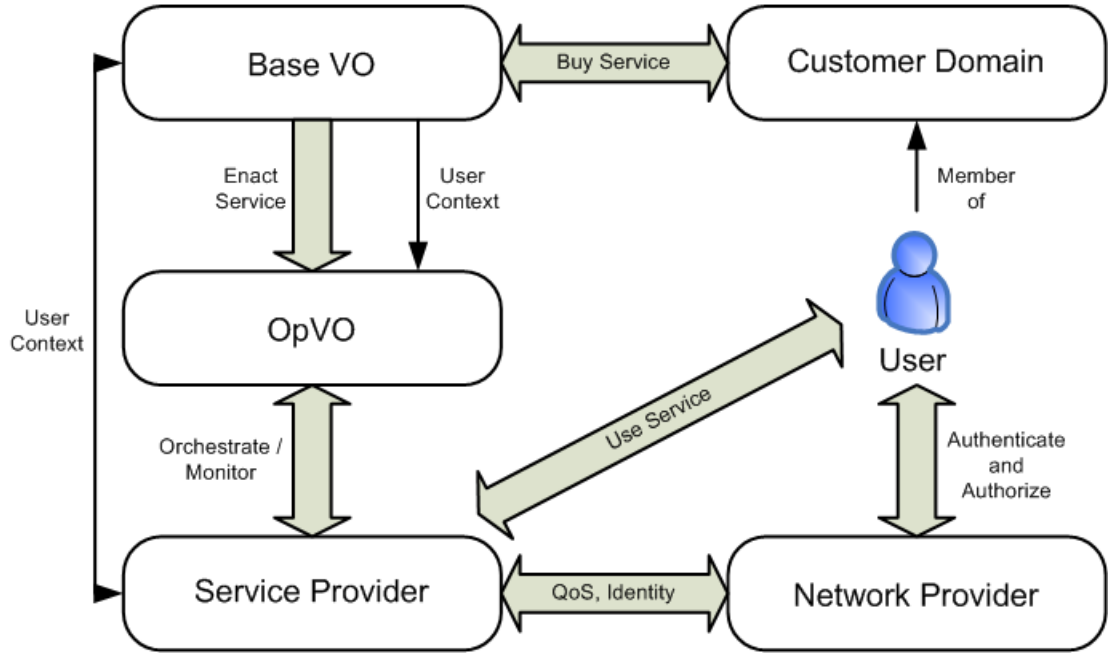

Figure 2: Akogrimo Building Blocks

The remaining sub-sections of this chapter describe what these building blocks are really dedicated to whereas in the next chapter (2.3 pp) the detailed design of selected building blocks is presented.

2.2.1.1. Base VO

The BVO is a central building block in the Akogrimo architecture and conceptually interacts with all the other building blocks introduced in Figure 2.

The interaction with those building blocks is related to the role that the different participants can have inside the BVO. Three main roles have been identified: Network Provider (NP); Customer (C); Service Provider (SP).

The following relationships are established between the BVO and the other building blocks:

- He BVO provides services to the customer domain. In detail, the user belonging to the customer domain is allowed to perform two main actions inside a BVO: searching for services and creating OpVO. Then the relation with the BVO will be oriented to manage this kind of interactions between the Customer domain and the BVO. In particular, the creation of an OpVO has relevant effects on the BVO domain, then that interaction has important security implications (strong authentication, authorization, possible attacks,…)
- The SP will be allowed to publish his services inside the BVO and their use will be acquired by the customer through the BVO itself. In this case the relation with the BVO is oriented to manage the publishing phase. This includes user context.
- The OpVO building block is a component herniated from the BVO. The BVO interacts with the OpVO creating it and configuring a dedicate subdomain for its execution.

2.2.1.2. Operational VO

The Operational VO (OpVO) is a building block that assumes an important role at execution time when the application is actually delivered to the customer. The OpVO is the "run time environment" of Akogrimo applications. In any case the OpVO lives inside a BVO and in each BVO can be instantiated several OpVOs. The OpVO can be thought as a particular instance of VO.

As in the case of the BVO, the OpVO interacts with all the building blocks identified in Figure 2, but the most of relationships have a different meaning:

- The NP provides the infrastructure to access the OpVO features via the Service Provider and the OpVO leverages on the trusted NP (members of the BVO) to authenticate the identity of the incoming requests. The relationship is similar to the one existing between NP and BVO. Anyway, the BVO has a more tightly relation because, actually, the OpVO trusts in the NP because the BVO trusts in it.
- The Service Provider will be contacted by the OpVO in order to negotiate the instantiation of a service in the SP domain to be used during the workflow execution. The success of this negotiation/orchestration/monitoring will bring to the establishment of a contract. When a service instance has been reserved, a new relation is defined between OpVO and SP that is based on the invocation of this instance from the OpVO.

2.2.1.3. Customer Domain

The customer is the entity that buys services from the Grid on behalf of a user and offers them to a user or consumes them self (in that case the customer is also the user). That can be a hospital offering aggregated medical services to patients or a mobile user who wants to access a printer.

The Customer Domain is the Home Domain of the customer; if the customer is also a service user the terms Customer Domain and Home Domain (of the user) refer to the same domain.

A trusted relationship between the Customer Domain and the BaseVO (BVO) Domain is needed as well as between the Customer Domain and the Home Domain of the user.

2.2.1.4. Service Provider

On the service provider domain are services that manage and supply resources and enforce policy and service level agreements. The tasks that are supported by the services located in the service provider domain include but are not limited to:

Execution management: This task is decomposed into multiple tasks, all related to the execution of the services requested by the client, such as preparation, initiation and managing of the execution. These tasks are addressed by the Execution Management Service (EMS), a suite of services that appears as a single service from the client's perspective. The EMS covers these tasks and involves interactions with many other services located in the other domains.

Accounting and charging: A service that supports the metering of the resources that are being consumed during the execution of a service is located in every machine that is hosting a service for sale. This service supports the accounting and charging process by sending aggregated information about the consumption of the resources.

Discovery: The service provider can advertise its services and resources. This activity is supported by the EMS that is located in the specific service provider domain. The service provider can create advertisements about the services and/or resources that are for sale and register them to the index service of the EMS that serves as a SP-domain wide index service. The EMS performs discovery by querying its index service in order to find a service that satisfies the clients' criteria.

Identity: All services located in the service provider domain are accompanied by a set of mechanisms for establishing identity and negotiating authentication.

2.2.1.5. Network Provider

On the Network provider domain there are all the related functions represented required to provide network services in a Grid-compliant manner, which makes this domain Akogrimo specific and unique. Since basic network functions – as currently provided in other architectures as well – are not described in a Grid/Web Services compliant way. Additionally, the available accounting and charging activities on this domain are ready to account for and charge for Grid services as well, which finally makes the network domain basically ready to commercially deploy grid services on their infrastructure in a way compliant to the current IETF approaches.

This comprises an identity concept which is able to offer SSO-based services comprising Grid services as well as network services, which in turn required the authorization and authentication of the device used by the user.

The Discovery of Services is a complex issue especially having the different bootstrap processes in mind and is considered within Akogrimo as one of the key research challenges, which might not completely solved within this project.

Finally, network management issues contribute to an overall policy-based management concept which of course comprises again Akogrimo "cross-layer" issues.

2.3. Base Virtual Organisation

A Virtual Organisation (VO) in Akogrimo is the dynamic collection of individuals and institutions which are required to share resources to achieve certain goals.

Starting from this generic concept two derived concepts have been defined: the Base Virtual Organization (BVO) and the Operational Virtual Organization (BVO).

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.

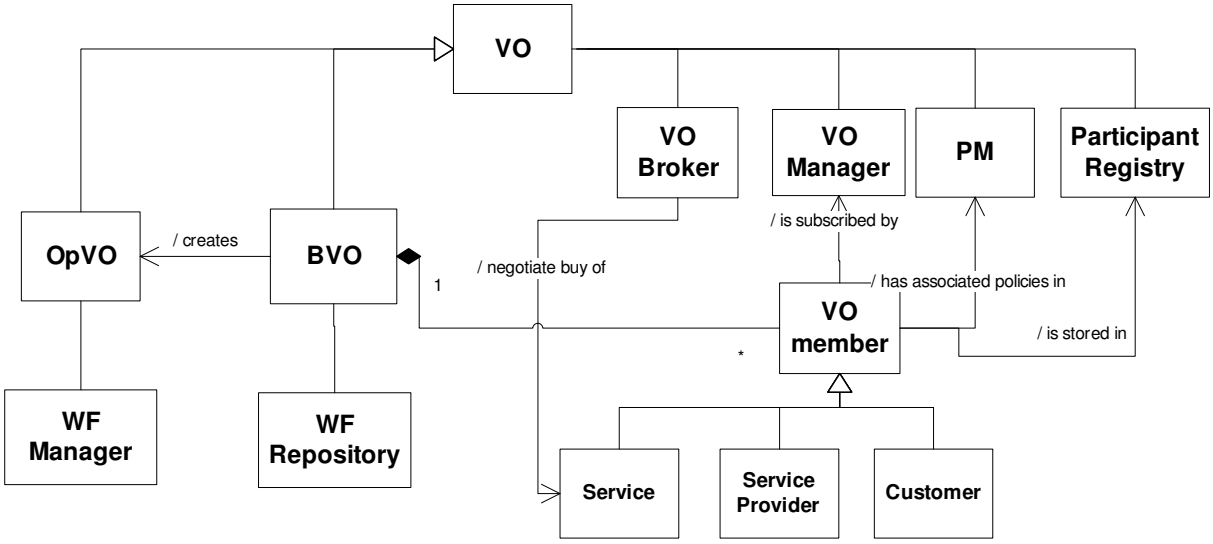

Figure 3: The Akogrimo Base VO concept

In each VO at least the following set of management services has to be present: VO Broker, VO Manager, Policy Manager and Participant Registry.

These services will be present both in the BVO and OpVO.
In particular, a BVO will use:

- A participant registry (in order to store the list of BVO members)
- A BVO manager (in order to manage subscription, authorization and some basilar BVO operation)
- A policy manager to store and distribute policies that apply to VO members.
- A VO Broker (in order to negotiation purchase of services made available from Service Providers)

A part from the above services that are common to the VOs, the BVO will include a Workflow Repository service, as well.

It is possible to state that the BVO identifies a dedicated domain that includes all the administrative services that allow managing the participants of the BVO itself. Furthermore, the BVO is composed by its members that can be: Customer, Service Provider and Services.
The BVO is a "quite static" environment in the sense that it can be thought as a sort of market place where customers demand for services that could be made available in the BVO by Service Providers that are subscribed to the BVO itself.

Summarizing in a BVO it is possible to:

- Make available services publishing them into the "yellow pages" of the BVO.
- Buy service use searching them among the ones available into the "yellow pages".

In order to use the bought services, the BVO will create an OpVO that allow interactions between buyers and providers. The OpVO bounds a dynamic environment because all the interactions taken in action during the run time execution of a workflow will be performed in the frame of an OpVO.

2.4. Operational Virtual Organisation

The purpose of the operational VO is to instantiate a business process and to manage the execution. The BVO has the role of starting the OpVO creation process. During the creation phase of an OpVO, all the dedicated management services are instantiated (the ones associated to a general VO, and they will live until the OpVO will be alive.

In this case the management services will be:

- OpVO Broker to negotiate external services to be involved in the WF execution
- A dedicated Participant Registry, Policy manager and OpVO manager to manage members of OpVO in a similar way as in the BVO.
- WF Manager: it is a management service specific of the OpVO. It manages the instantiation of a WF and its correct execution.

All these services will be destroyed when the OpVO dies.

The members of the OpVO are the management services, the User Agent (UA) and the Service Agent (SA).

UA and SA will act, respectively, on behalf of external users and services. They are the actual entities that interact with the Workflow instance.

The OpVO is the dynamic environment that allows the application execution in the Akogrimo environment.

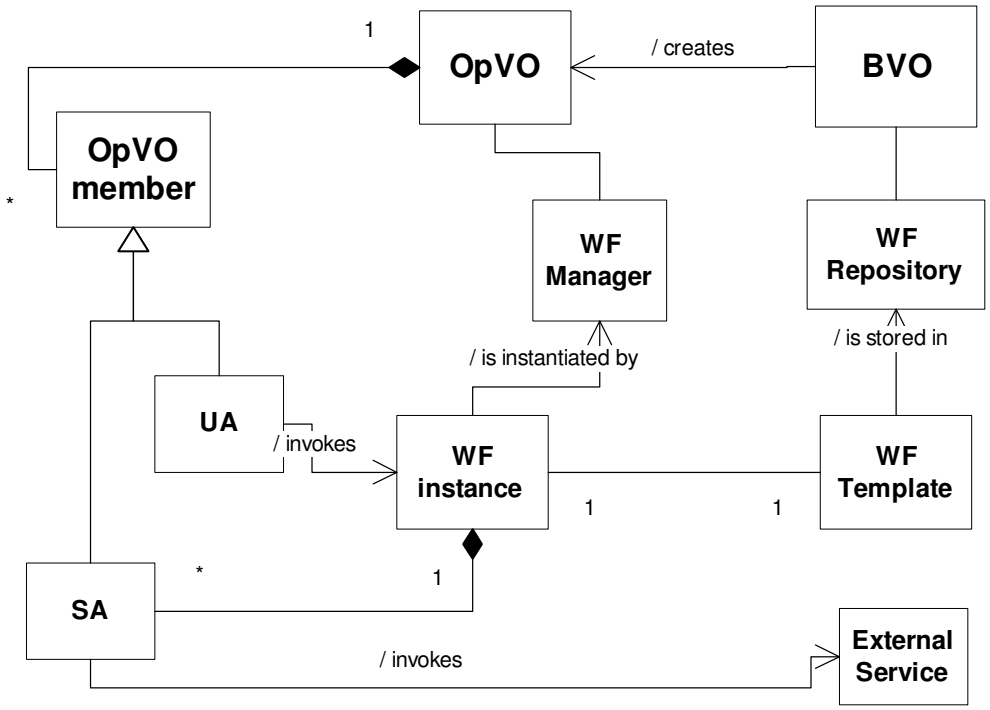

Figure 4: The Akogrimo Operational VO – Components and concepts

2.5. Business Process and Workflow

The Akogrimo architecture is intended to support business process designs that both, support and take advantage of dynamic, mobile services and users.

In this context it is important to distinguish between the terms "business process", "workflow", "choreography" and "orchestration". By a "business process" we mean a high-level description of a process in terms that are meaningful at a business level (as opposed to a computational or engineering level). Such a description should be abstracted from any particular implementation of the process, and should be more concerned with requirements and goals than specific execution approaches. At the opposite extreme, a "workflow" is a precise definition that is (or can be easily converted into) an executable form. The basic steps of a workflow are normally service invocations; though even here the workflow itself is not concerned with how the service is implemented (though it does have to depend on and make assumptions about the service's behaviour).

"Choreography" refers to a form of process control where there is no single centralised controller, but where the details of execution of a process are delegated to individual execution sites. A choreography may describe how the different parts interact, but will almost certainly not describe or control how each part performs (only that it produces the expected interactions at the expected times). In "orchestration" on the other hand, the behaviour of a process is under control of a single central agency which controls the actions of each part and mediates in (or at least knows about) all interactions between them.

To some extent, the distinction between choreography and orchestration is a matter of detail. For example, an ActiveBPEL engine running a workflow script "by itself" is acting as an orchestrator; but even so, it does not (and cannot) define or control the implementations of the services that are used in the script. In the other direction, the script may be being executed as part of a larger choreography that links other workflows. In Akogrimo, the need for a "big picture" view capable of assessing and handling context changes lead towards an orchestration solution, while the dynamic nature of most of the applications tends to support devolution of responsibility to choreographed services (ie they know what they need to achieve and just get on and deliver it)

Akogrimo intends to cater for the mobility of participants (both users/clients and services) in a business process. One consequence of this is that the business processing components must "track" users and services as they change location while retaining their identity, but must also support the ability of the process to adapt to changes in context of such mobile agents, for examples, changes in their capabilities, discovery of alternative services, and responding to situations where an agent becomes disconnected. Of course, some of this adaptation can be delegated to the services (particularly the "how" to adapt in order to still deliver in the face of a context change) but ultimately any change that may require a change in the overall business process must be handled at the orchestration level (the "what" to adapt in order to satisfy the possibly changed business objectives). For example, the availability of a high definition screen in an eHealth scenario may mean the presentation of some medical data is changed (graphs instead of text), or it may mean that the pattern of interaction with the user is changed (patient can be treated on the spot rather than being immediately rushed to hospital). One immediate consequence of this is that the business process enactment sub-system needs to have access to the context information associated with all its users/clients and services.

The final requirement generated by Akogrimo's mobile and dynamic nature is the need to build on-the-fly secure (Operational) Virtual Organizations, where the data can be shared among the dynamically changing members but prevented from falling into the hands of outsiders.

2.6. Mobility Concepts

One of the main objectives of Akogrimo is to allow mobile users to not only use the Akogrimo network, but to allow the effective use of those users devices and resources as part of the whole Akogrimo network. Mobility presents several challenges; a breakdown of the types of mobility involved follows.

2.6.1. Terminal Mobility

Terminal mobility, as the name implies, relates to the mobility of the device (or terminal) the user utilizes for accessing the network. In order to have effective terminal mobility, the communications of a given user must not be disrupted by movements of that user which might provoke network topology changes.

Topology changes can occur when users move from one network access point to a new network access point. Without terminal mobility support, the change from one access point to the other will cause the terminal to lose its connection with the old access network, acquire a new connection in the new access network and a new IP address. In practice, this causes network connections to be stopped. They then have to be restarted, with a new IP address, which causes problems for most of the software that uses the network.

A solution for this problem is the Mobile IPv6 protocol. Mobile IPv6 is further discussed in section 6.1.1.

2.6.2. User Mobility

User mobility relates to the capability of the user to access personalised services independently of the terminal device and access network he or she is using. It is provided by a user-oriented security and authentication framework. The user has to perform his/her registration in the network – and in the Grid infrastructure – before using the network services. This registration process associates the user with the terminal.

2.6.3. Session Mobility

Session mobility enables the transfer of application sessions between different devices without interruption. This is achieved with the SIP protocol. SIP can be used both by the user, and by the Grid infrastructure, to redirect communications (e.g. image display) to different devices, retaining the user association mentioned above.

2.7. Administrative Domain and Trust

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.

An administrative domain, be it a network provider or service provider (or a combination of both), is an independent organization that provides services and interfaces for accessing these services. The management (monitoring, A4C, control) of the services provided by an administrative domain is a domain-internal issue. If services within a domain are part of complex applications spreading across domains, management services can be provided to domain-external components through management interfaces.

Trust enables cooperation where a guarantee of a benevolent behaviour is not possible. Reputation systems provide information about past behaviour which can be used by potential cooperation partners to gain confidence in the future behaviour of their counterpart. If mechanisms can be found to guarantee a certain behaviour or piece of information, trust is not necessary. E.g. authentication mechanisms should provide a high confidence (if not a guarantee) in the correct identification of a person.

In Akogrimo reputation systems are not used. Instead the members of a Base VO have mutual static trust relationships. What will be used in Akogrimo is trust technology to transfer the authority to trust statements of an issuer. Especially statements about the identity of a person have to be trustworthy.

In order to set up an Akogrimo VO, there are several pre-requisites in terms of trust relationships between domains that need to be assured. These are the following:

- One domain or organization can represent several users, services or customers. It must at least represent one entity.
- All entities that want to form a VO need to be bound to at least one domain. That is, each entity has signed at least one contract with a specific organization. This organization represents the specified entity at others organizations and manages the entity's information.
- One user can have several Home Domains. However, at one time, the user can only be logged in at one Home Domain. There is no relationship between different Home Domains of the user. Their user's information is managed separately and can not be linked.
- Each domain that hosts a Grid Service of an Akogrimo VO needs to have a trust relationship established with the Base VO domain.
- The customer's domain and the Home Domain of the user need to have trust relationships established.
- The customer's domain and the Base VO domain need to have trust relationships established. The customer domain represents the user at the VO Domain.

The points explained above need to be applied before setting up the VO.

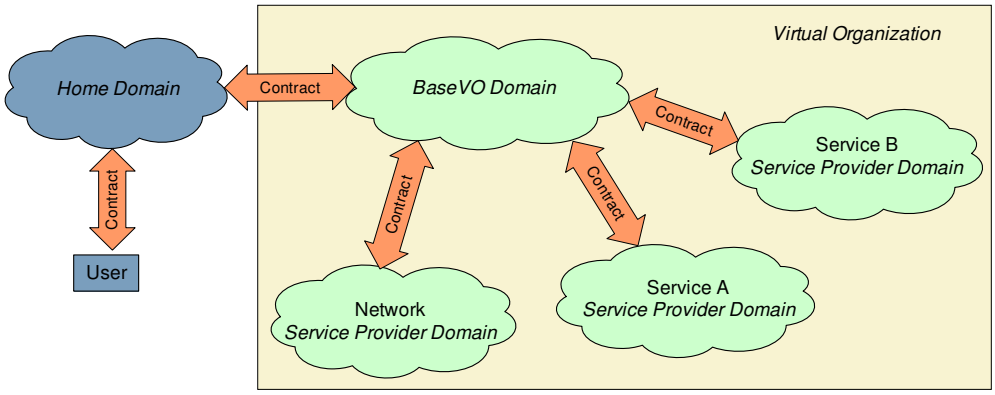

Figure 5: Echogram example of trust establishment

As shown in Figure 5, trust relationships are required between the different administrative domains in order to support this approach. The trust relationships are based on legal contracts, and from the accounting and charging point of view they assure that any service that was instantiated and consumed will be paid by the organization who requested the service in the name of the user.

Such an approach was chosen for Akogrimo for reducing the number of contracts with SPs a user has to manage. Based on the contracts between SPs, users may access services in administrative domains with which they do not have any contract. A contract and trust relation between the BVO and service providers assures that whenever a user requests a service from the SP, the SP can send the service charge fee to the BVO and expects that the BVO pays for that service session

2.8. Identity

In D311 the term of identity was described in a general and abstract term, followed by the concrete term description used by Akogrimo. This chapter intends to explain further investigated concepts related to the identity in Akogrimo which will contribute to the innovation in the Grid environment within the identity management area.

The first efforts related to the identity management topic in the Internet were in the direction of the X.509 certificates. X.509 certificates usually fulfil better the requirements of a service's identity due to its static nature – the identity of a service does not normally change. However, X.509 certificates are not flexible, they are heavy weight and they can not be used to change identity according to the user's needs. And even with the use of X.509 certificates a user has to provide more credentials than only his certificate to verify his identity. Therefore, a more flexible and interoperable solution in the Internet, the federated identity management appeared as a new initiative which intended to include the necessary flexibility and interoperability in the identity solution.

In the traditional Grid community identity has been considered as something static, represented in the form of certificates, due to the much scientific nature of grid. In this way, the user and/or the organization presents one single certificate with the same identity – identifier and user profile – to all the different organizations and they map this user's identity to a local one representing the user in that local organization. The mapping is done with a manually configured grid map-file.

Lately, there has been a closer cooperation between the grid and the federated identity world. First, Globus Toolkit released a new version which could use SAML authorization assertions. This was a step forward. But the federated identity world make use of SAML authentication and attribute assertions. With time, the Grid community realised that their certificates do not scale

well and that many organizations already offer federated identities for their users, in business and in scientific environments. Federated identities offer benefits of dynamicity, security, automatic configuration…etc. Further, the federated identity community has realised the importance of the Grid initiative and the must to provide special features not already thought as plain web-based services or stateless web services. Therefore, cooperation has been started between them in several areas and there are already research projects investigating how to merge both, like the GridShib project.

2.8.1. Personalization: User Profiles and Digital Identities

The user consists of a set of characteristics that are required to be stored either in the home domain either in a service provider domain. Some of these attributes are considered to be of a generic nature and consist of personal user information (name, sex, age, bank account ID, etc.). Another subset of attributes is related to the home domain of a user and contains information like policies to be applied after a user logs in. The personal user information and home domain specific attributes should be stored inside the home domain of a user. A third subset of attributes contains service-specific information. This set of attributes can be stored either in the home domain either in the service provider domain that provides the service. A user profile consists of the set of all characteristics of a user (personal, home domain related, service related).

An identity profile is a subset of the user profile which contains just the attributes which are relevant for a service or a group of services.

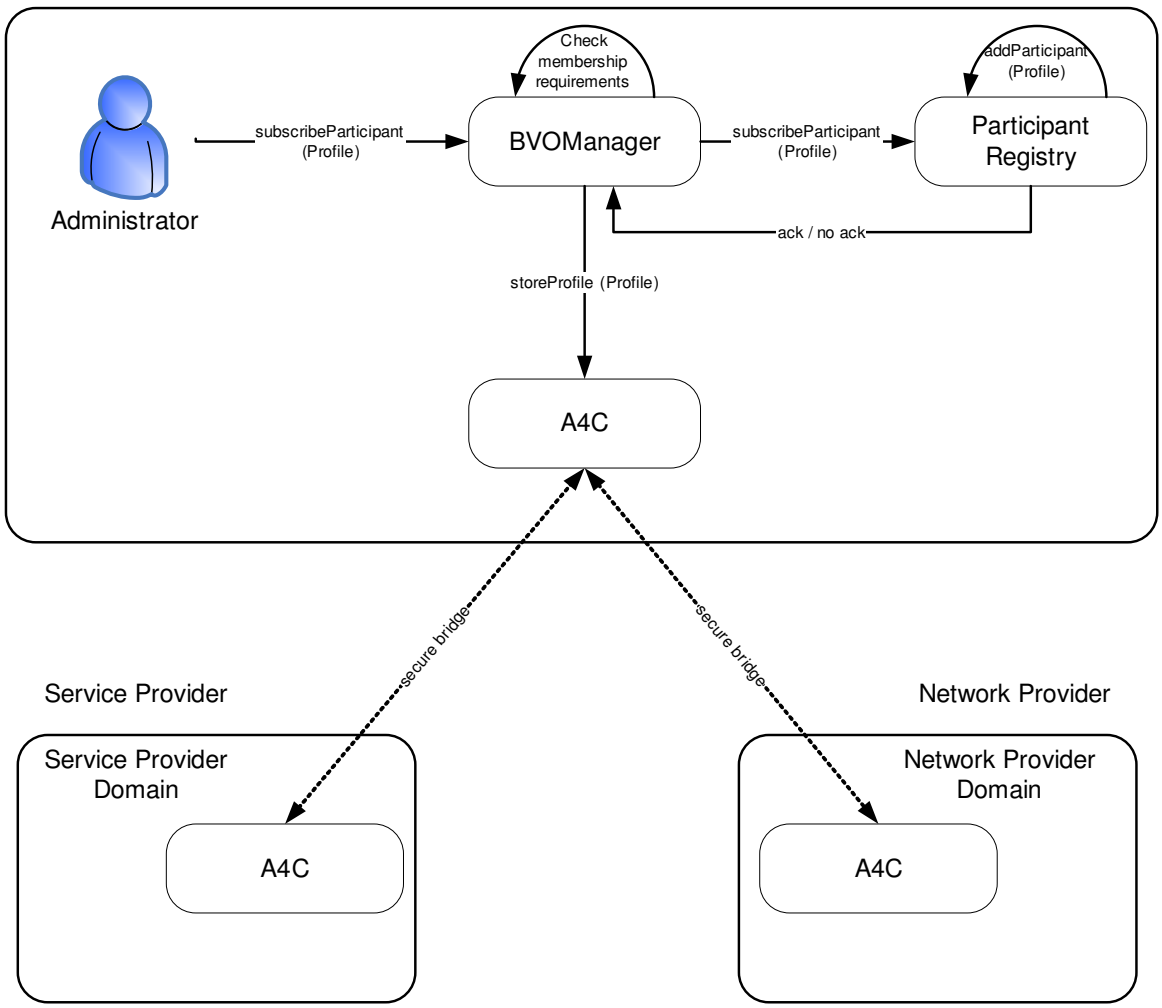

Figure 6: Identity Model

In the normal life, the user identity which is usually presented depends on the type of organization. For instance, the identity profile that an employee presents at work with attributes like social security number, Health insurance number among others- is different to the profile that a user presents, for example, at a travel agency – with attributes like interesting countries, type of trip etc.

It is desired in Akogrimo that the user can choose the attributes (excluded demanded attributes) he wants to share within a VO. Depending on several factors like the content, the purpose and the nature of the VO, he will want to customize the set of attributes that form his user profile. In Akogrimo, the personalization will be done by the user, at the BVO subscription in a very straightforward way. He will be able to select the attributes that are already stored in the home domain and include other ones at that phase. The identity profiles are stored at the Participant Registry of the VO the user subscribed to.

In this case, it is considered that the user can have several identity profiles and digital identities. It is a matter for the user if he wants to use the same identity profile for several VOs.

Figure 6 shows the Identity Management from a BaseVO perspective. When a user subscribes to a BaseVO, the BVO Manager adds user's profile in the Participant Registry of the BaseVO. Authorization requests like these from the Policy Management Systems are later sent to the Participant Registry and can be answered. Furthermore the secured bridge between all involved A4C Systems leads to the ability of exchanging profile information dynamically between Participant Registry and Home A4C Server – each time, user's profile changes during his VO-

membership either at the A4C or at the Participant Registry the other component will be informed about that event by the BVO Manager.

2.8.2. Single Sign On with multiple identities

Although the user has multiple identities, for different VOs and domains, the ability of Single Sign On is still possible with support also for anonymity or pseudonymity. During the initial authentication the user receives an IDToken, which is unique and independent of the real identity of a user. The mapping between IDTokens and the different user identities can be only performed in the A4C Server in the home domain of the user. Based on this token, the services requested by the user can verify the authentication and can request the identity of the user. The A4C Server in the home domain selects the identity to send to the service based on a predefined policy and user's choice. The detailed procedure is described in deeper detail in Deliverable D422.

2.8.3. User Identifier

Users and services need to be identified in Akogrimo. The solution to be used in Akogrimo should preferably allow a globally unique identifying scheme and not only identity unambiguousness in a single administrative domain.

The A4C uses the following format for identifying users: *username@domain* where *domain* is a FQDN. This format also corresponds to the simplified form of the SIP URI format, so the relationship between user IDs and their SIP URI will be one to one, simplifting the usage of SIP for managing sessions among the different AKOGRIMO actors. This solution gives globally unique identifiers as long as each domain makes sure that *username* is unique in the local domain. At the same time, services can also be identified using the same scheme: *service_A@domain_B*. Such an identity scheme easily allows users to act as services or services to be identified as clients of other services.

The identity scheme to be used needs to provide at least the means of pseudonymity. This allows a user to hide his real identity from a service provider. One way for achieving this is to allow a user to choose what identities will be given away and to whom.

Typically after a network login a user is presented an IDToken that can be linked to the initial authentication of the user. Based on this token, service components can verify the authenticity of the user and request his identity. The only entity that can verify the ID Token and thus the identity of the user is the A4C Server in the home domain of the user.

The solution allows users to use different identities and gives the possibility only to the A4C Server in the home domain to map different public user identities to private user identities.

2.9. Context and Device Capabilities

Context is any information that can be used to characterise the situation of an entity. So, context awareness means that devices, applications and systems have information about the circumstances under which they are operating and can adapt their behaviour to be optimal for the current situation. This increases system usability tremendous by reducing the demands on the end-user and minimizing the need for user attention.

The vision of Akogrimo involves context-aware mobile Grid services (see Figure 7), allowing the user to solve his/her tasks while devices and technology fade away into the background. Akogrimo focuses on context that describes the situation of a mobile user. While much prior Grid research has focused on batch-mode supercomputing applications, Akogrimo introduces the mobile Grid, where interactive services involving mobile and nomadic users are of key

importance. The lives of humans are much more varied and dynamic than those of computational resources, so when humans are introduced as resources in the Grid, there is a need to keep track of their context. By knowing this context, the system may easier choose the right people to participate in a workflow, and better adapt service behaviour to the situation of those people. Context has many facets:

- Presence: Is the user logged on? Is he idle or busy?
- Physical properties: User location, local time, body temperature etc.
- Device context: What terminals and other I/O units are available to the user and what are the hardware and software capabilities of those devices?
- Local services: What services are found in the proximity of the user?

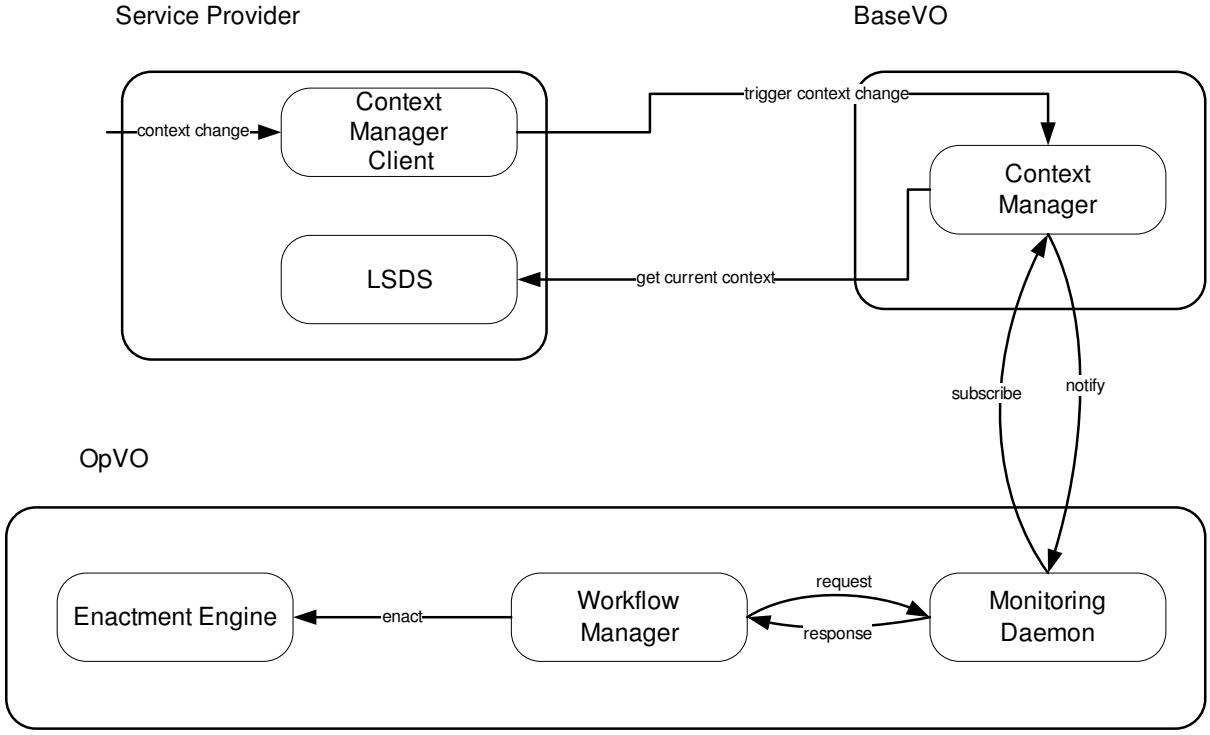

Figure 7: Context flow

The definition of context is open-ended; the set of data that should be monitored will depend on the application domain. Akogrimo shall therefore provide a context infrastructure which allows relevant parties to keep track of user context in terms of (SIP-) presence, local services in user's proximity, user location and device capabilities for user terminals.

In Akogrimo context information for a user is mainly obtained via the device(s) he/she is using, but also by relating the user to local information for the place he/she is located at. 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, hence enabling context awareness. The context includes dynamic user information such as environmental information (temperature, humidity, weather), geographical location, device capabilities (display resolution, screen size), and network capabilities (bandwidth, connection type).

2.9.1. Context Manager

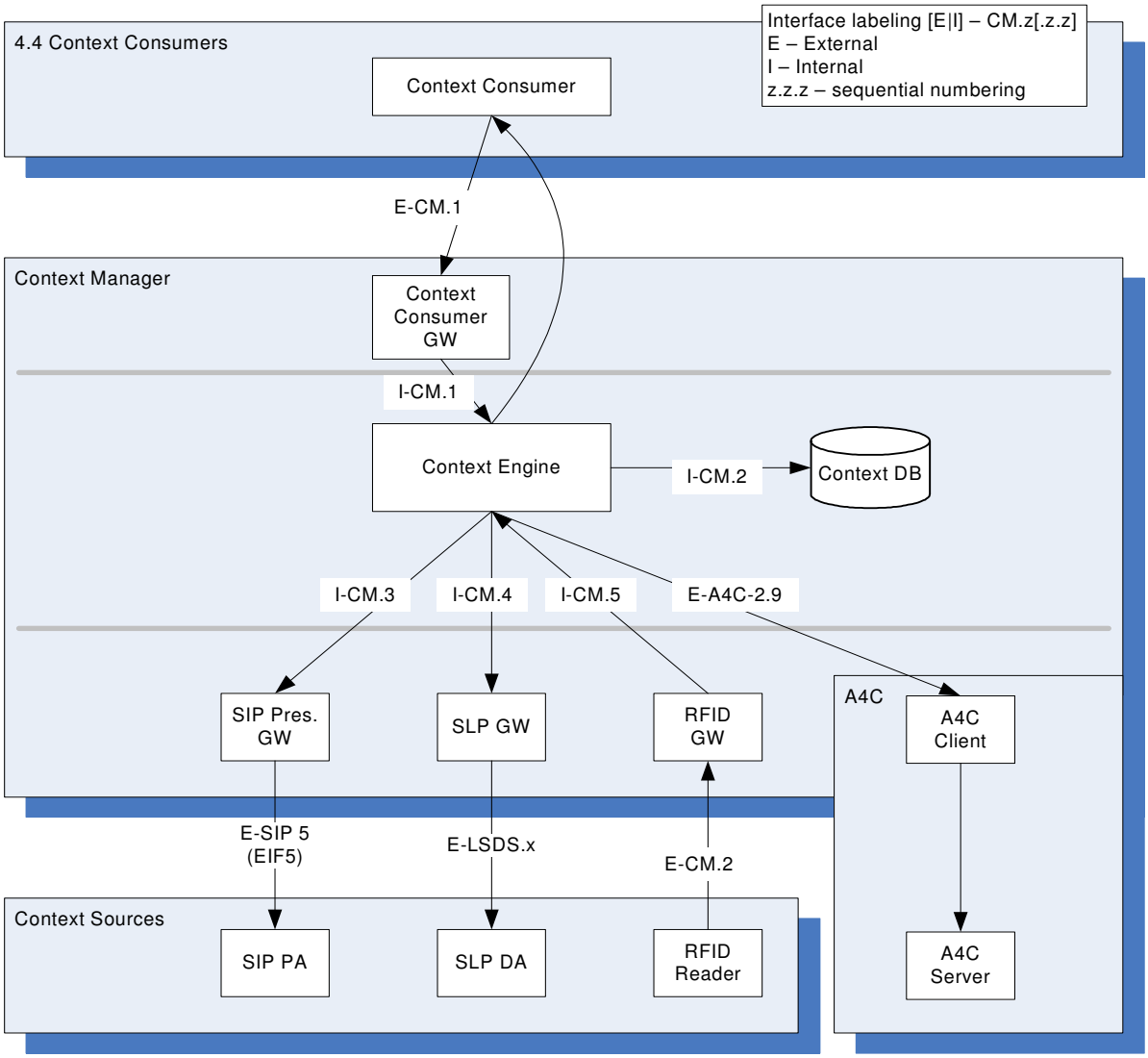

Figure 8: Structure of Akogrimo Context Awareness system

Obviously, it is not desirable that every application implements mechanisms to gather and process specific context information. A better solution is to make use of *Context Manager*, a generic context infrastructure that supports context-aware applications (aka context consumers) by gathering, processing and managing basic context information. Context Manager, being a mediator entity, is an essential component of the Akogrimo infrastructure. This element filters relevant data and converts it to a uniform format prior to delivering it to the interested destination. Additionally it has to solve the possible inconsistencies and infer high level context data from basic context information.

Figure 8 gives a more detailed view of the Context Manager design.

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 (using a suitable semantic language, e.g. RDF, OWL). The Context Manager is responsible for distributing context data to context consumers accordingly.

To handle domain specific context inference, the Context Manager should offer interested parties interfacing with extension modules. One example is the mapping 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.

More details about the Context Manager may be found in the Deliverables D4.2.2, for more comprehensive discussion of context and device capabilities, see D2.2.4, D4.2.1 and D4.4.2. Figure 8 shows the structure of the Akogimo context system.

2.10. Policy Management

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 (see Figure 9). Note, agent here is not necessarily a true software agent – it may just be an "if" statement in the interface to the entity.

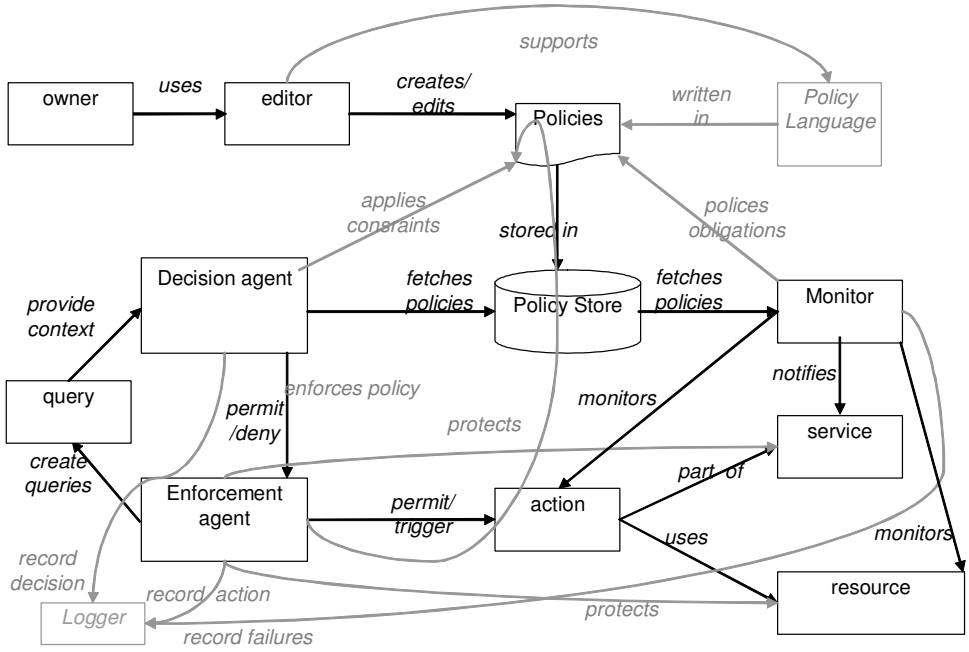

Figure 9: Policy Framework

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 requires 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. In any case, methods should be provided to detect possible policy conflicts and take the necessary actions such as preventing the policy owner. Policy conflicts may appear, for instance, when defining different policies targeting different groups (e.g. users under 18 years old and "gold profile" users) and there exist individuals belonging to both groups (a young user but still paying much and having a "gold profile").

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 for security policies, WS-Agreement for SLAs, CIM for PBNM (Policy Based Network Management), etc. However, the policy language in the enforcement agent may differ from the one used by the decision agent. Indeed, the enforcement agent may employ a policy language much closer to configuration commands. The decision agent will then need to do a policy language "translation". Akogrimo aims to offer a coherent model to policy owners to create and

edit policies. This can be easily done because the policy languages considered – at least for the decision agents - are all based in XML.

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.

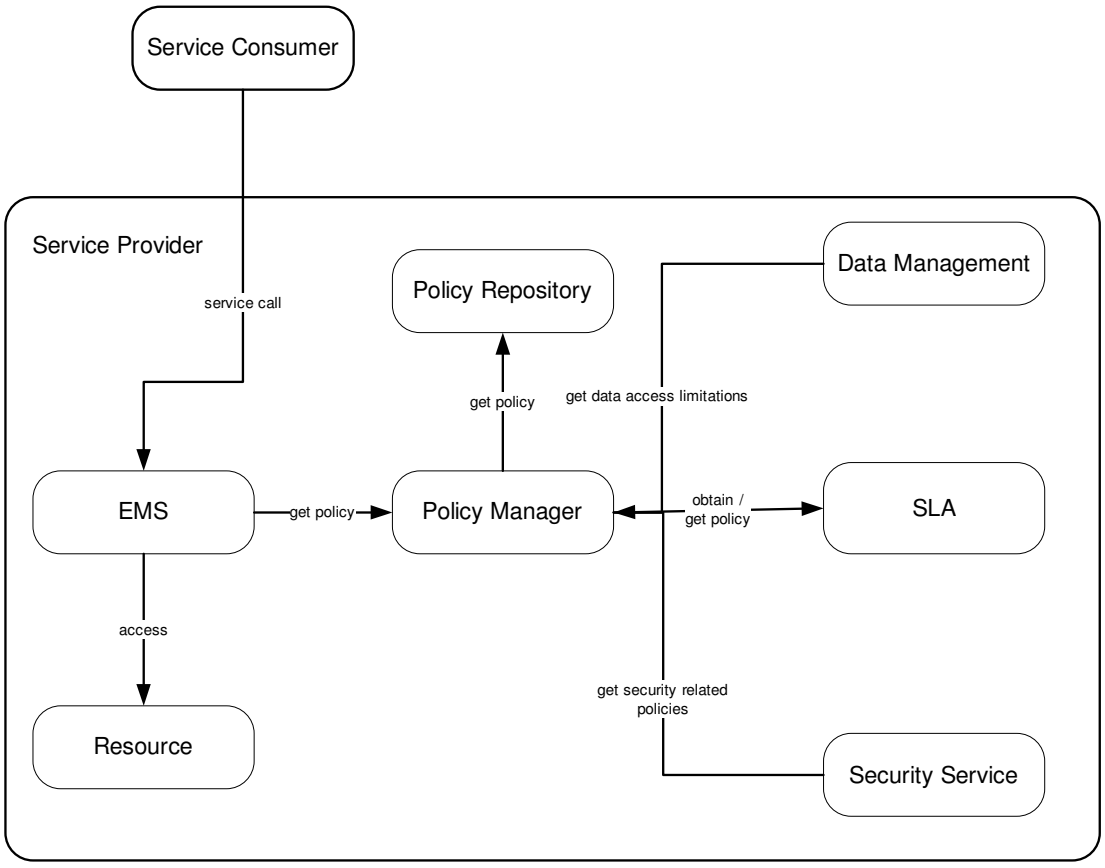

Figure 10: Policy Management – Service Provisiong detail

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. Note that the literature, especially for policies related to network management, also employs the terms Policy Enforcement Point (PEP) for the policy enforcement agent and Policy Decision Point (PDP) for the policy decision agent. As this "enforcement and decision split" 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 – e.g. 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, i.e. 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).

There are several types of policy, as shown in Figure 10 on policy management at service provider level and for Web Services or in Figure 11 on policy management at network domain level.

The policy manager in Figure 10 encompasses the functions of the policy, policy language and editor modules of Figure 9.

The policy management represented in Figure 10 could help a Video Streaming service provider to handle situations such as deciding whether to deliver or not a movie to person under 18 years old. He must obtain the user data (in particular its age), get the movie data (rating) and apply the policy "If user is under 18, do not allow him to access X rated videos, else allow him". Finally the consumer is granted access or not to the video resource.

Figure 11 is focused on network domain level policies.

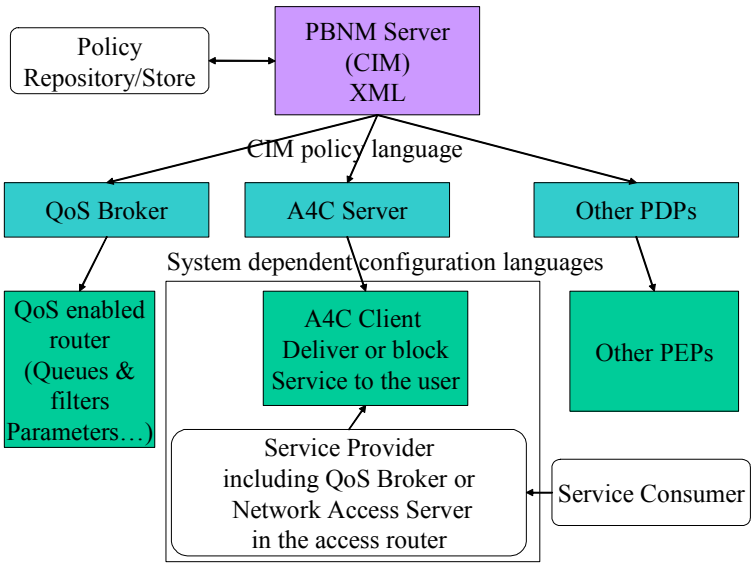

Figure 11: Policy Management – PBNM policies

Policies in Figure 11 aim to manage the relationship between different systems, for instance, and following with the previous example, when the service provider wants to obtain the user profile (user age) it will use its A4C Client to contact the A4C server and retrieve from it the user profile.

The A4C server, based on policies, will decide whether or not to deliver this user profile. Other policies that may occur within the Figure 11 framework are related to management of network resources. For example one may define using CIM a policy such as

"If the EF traffic load exceeds 70% of the allocated bandwidth, then allocate more bandwidth to the EF traffic taking it from other classes."

The QoSBroker may retrieve this policy from the Policy Based Network Management Server (PBNMS-S) at start-up. This policy can be defined using CIM languages and its QoS extensions QPIM and QDDIM. Based on this policy QoSBroker starts to monitor the traffic. When this exceeds 70% it will take a decision (it is a PDP or a decision agent) and will issue the configuration commands (for instance, using COPS protocol) to the routers so that they reduce the resources assigned to other traffic classes and increase the resources for EF traffic. Routers are the PEPs or the enforcement agents. The QoSBroker must understand the policy defined in CIM and take decisions and issue the configuration commands in a language understood by the routers, like COPS.

Although the scope of the Web Services policies and the PBNM is different, we can come to a coherent Akogrimo Policy framework, making them converge in the "high level". By high level, we mean the level closest to human natural language and farthest from device configuration commands. In this high level, the PBNM Server can be assimilated to the Policy Manager of Figure 10. It would be enough that both present a coherent editor through the policy owner and coherent high level policy management techniques such as policy conflict resolution (as explained above) to achieve a coherent policy framework in Akogrimo. Note that the policy languages for both Web Services (WSDL) and PBNM (CIM) are based on XML.

2.11. Signalling

The integration of Telecommunication infrastructures and Grid based Systems needs an integrated signalling framework which has been introduces in Akogrimo.

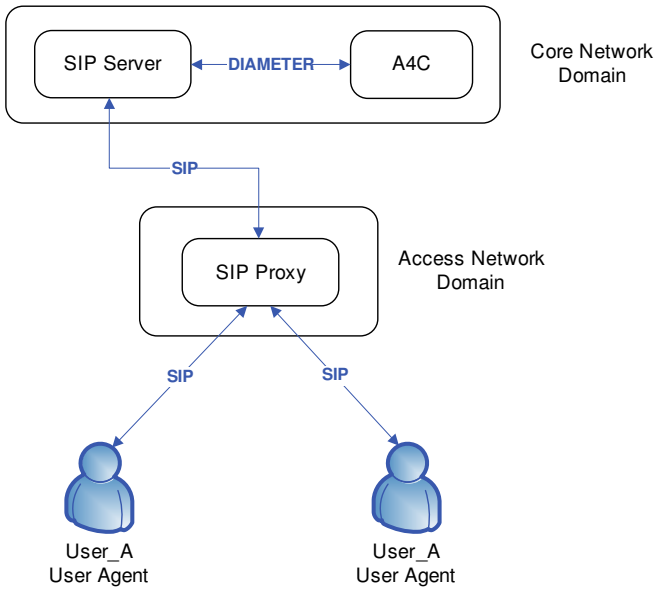

Figure 12: SIP Signalling

The Akogrimo user session makes possible to link certain events, actions and service consumptions together. An 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.

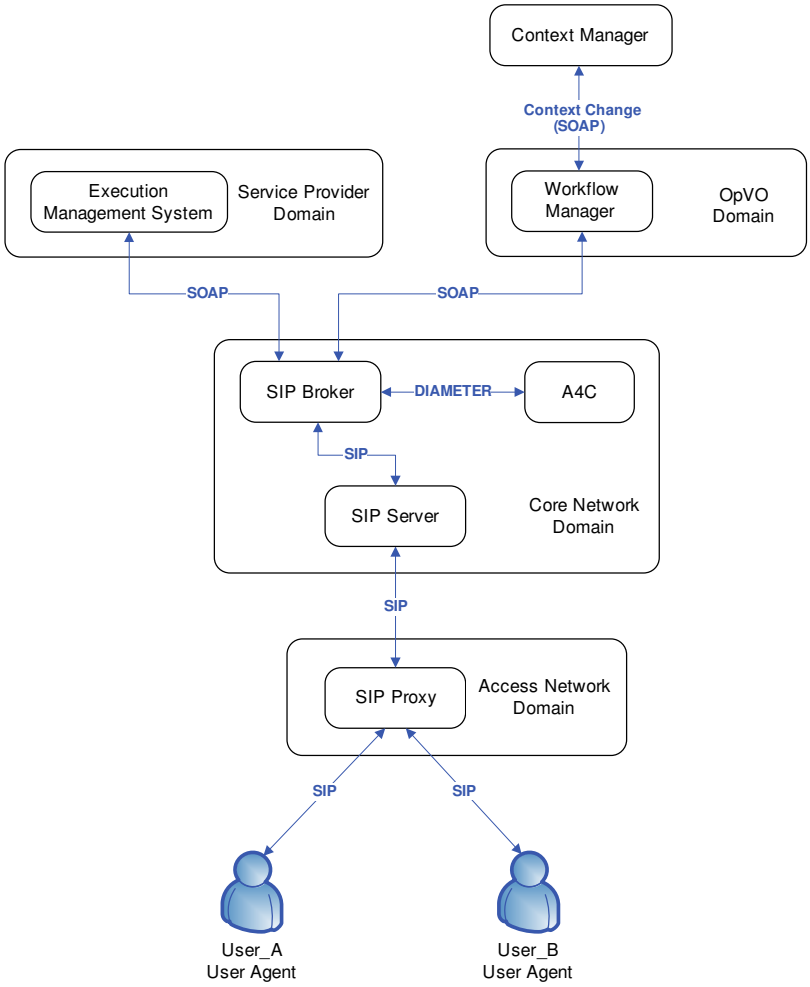

Figure 13: SIP with SOAP approach

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) 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 (as show the figure below). So, 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:

One of the most important challenges in Akogrimo is to integrate the Grid environment with the Mobile environment, where for instance the end users and service can move from one device to another one, or new device can appear/disappear. Several proposals were analysed to address this

issue: SIP over SOAP, SOAP over SIP and SIP with SOAP. Finally Akogrimo adopted the "SIP with SOAP"-Approach as shown in the following figure.

In this scenario, it is shown two SIP with SOAP interactions:

- In the first one, the Execution Management System establishes a Grid session with the users. So, it interacts using SOAP with the SIP Broker which in fact makes the real SIP session with the user on behalf of the Execution Management System.
- In the second one, a context change is detected by the Context Manager component, which notifies it to the Workflow Manager by means of a SOAP request. Once the Workflow Manager receives the context change then it decides to invoke (by means a SOAP request, as well) the transference of a previously established session to another terminal. So, it interacts using SOAP with the SIP Broker which launches the SIP signalling management for session transfer.

2.12. Discovery

Due to the importance of this subject, in this second cycle of the project the Service Discovery mechanism will be expanded to obtain a much more flexible and powerful architecture. In order to achieve this, a quite profound redesign is needed.

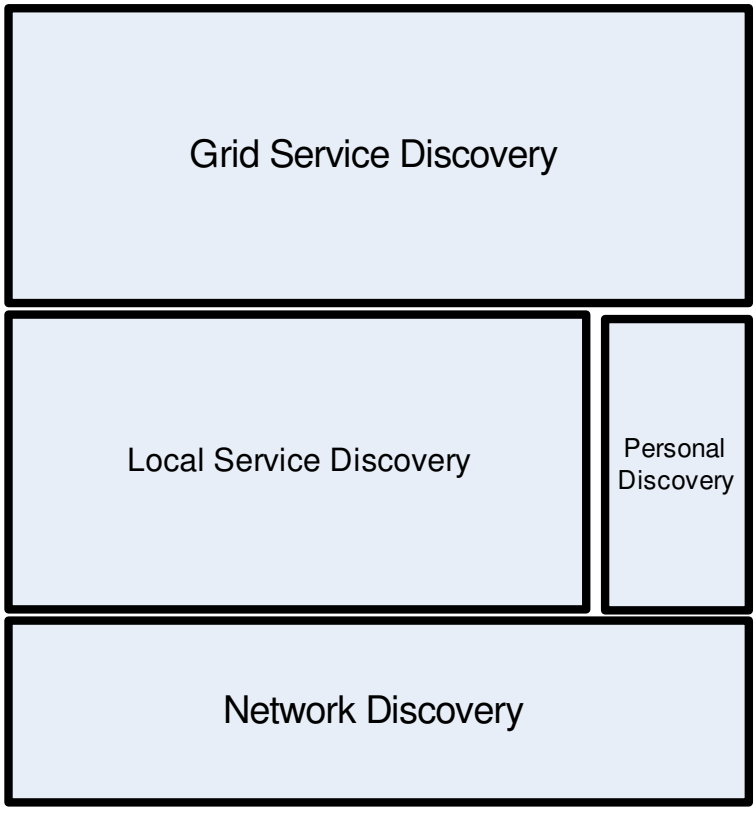

Figure 14: Network Discovery subsystems

On one hand, the challenge of providing mobility and location taking profit of this ability is a central part in the project, however, practice shows that most of the times location changes are not transparent for users in the sense that these changes require some kind of re-configuration at different levels. This reconfiguration should not suppose a burden to end users, and therefore this process needs to be partially automated. The network related SD components in Akogrimo try to overcome and solve some of these issues by providing automatic look up of resources

different types of resources. On the other hand, in order to be able to compose complicated and efficient workflows to solve complex problems, a flexible and generic enough Grid Service discovery system is required.

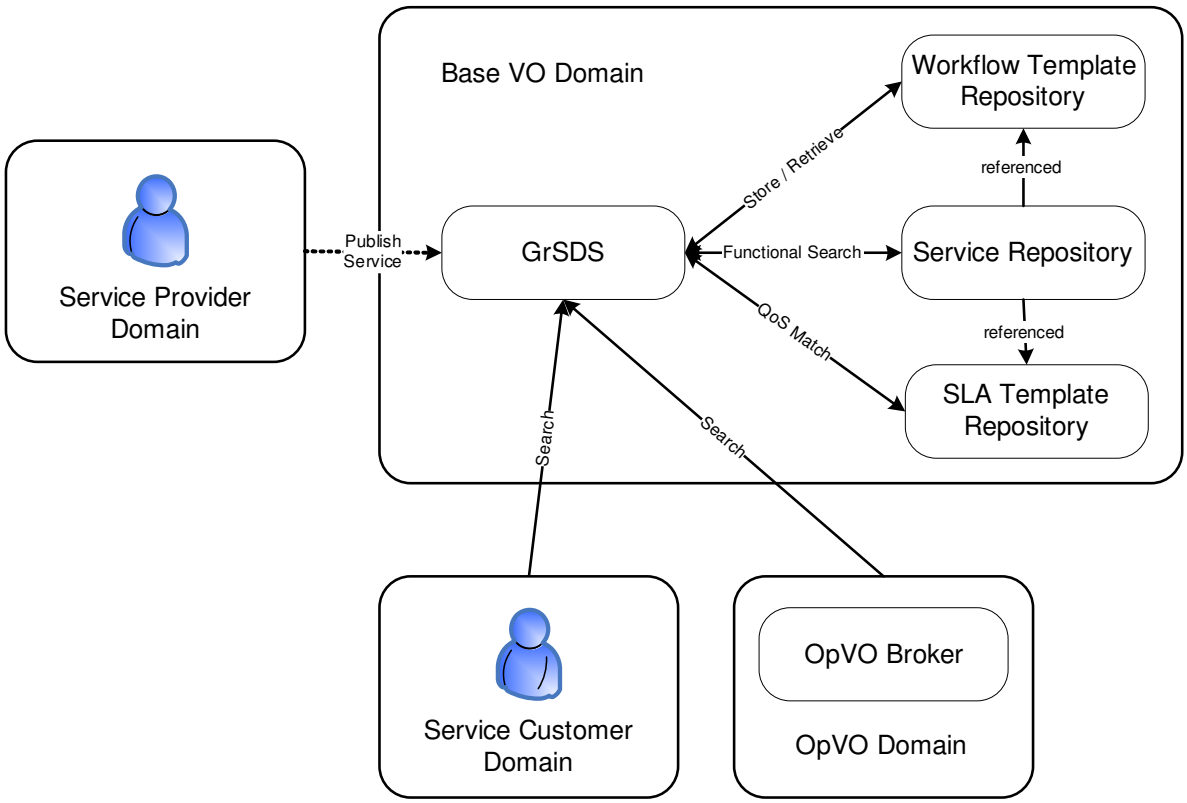

Figure 15: Grid Service Discovery Service

SD is then divided in different subsystems that execute different tasks as shown in the figure underneath. These are Network Discovery (NDS), device discovery (DDS), local service discovery (LSDS) and Grid Service discovery (GrSDS). Each one of these systems covers a different aspect in the project.

- Network discovery focuses on providing a user with network connectivity. Among all possible reachable networks accessible by all interfaces of the users' device some kind of information need to be conveyed about the network in order to choose one. This mechanism involves the Access routers of the different networks and the Mobile Terminals that perform the lookup. This network configuration problem is solved at IP level, based on a Zeroconf solution, since layers underneath L3 are of no interest in the project.
- LSDS is used to find network resources in the network being utilized by the user. Typical services rendered would be network services such as: printers, beamers, file systems, etc… that can be used accessed with standard network protocols. The LSDS is built upon the SIP infrastructure, that is, uses SIP to in order to publish and discover services reusing the SIP infrastructure which at the same time makes possible a very versatile and powerful service discovery system. One can subscribe to changes in the description of services and be notified, such as: *"notify me when new colour cartridges are available in a certain printer"* .
- Personal Discovery is used to provide the CM with some useful information about ad-hoc devices or services that the user is around of and don't belong to Akogrimo, but

necessary to compose a meaningful workflow. E.g. if the user has a certain device and the policy allows him to use a public device then the workflow shouldn't send him to the other side of the city to find an Akogrimo enabled device. This information is conveyed using the SIP protocol in the same way Presence information is propagated.

- Grid Service discovery does not discover service instances but service providers that offer a specified service. In this sense the GrSDS is like the yellow pages of the MDVO. Individual services and the associated SLA contract have to be negotiated with the discovered service provider.

Figure 15 shows the conceptual Grid services discovery part.

2.13. Authentication, Authorization, Accounting and Charging

Generally, the AAA infrastructure in combination with the identity model plays a very central role in the Akogrimo security activities. Especially the Authorisation of service usage, which can depend on several pieces of information, has been considered. 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 16 shows the relation of AAA between the components related to security.

The multi-domain service provisioning aspect highly impacts the way accounting and charging for composed services is done. A MDVO groups together several service providers and network providers that agree to share their resources for creating more complex value added services that aggregate "sub-services" of more than one organization. In order that such a service provisioning approach becomes economically feasible and efficient, appropriate accounting and charging mechanisms are needed.

Akogrimo's accounting and charging process is divided in three parts*: service accounting*, *session records aggregation* and *final bill calculation*.

The first phase, service accounting takes place inside a single administrative domain and consists of collecting details about the basic service consumptions. At the end of this phase, for each service that was instantiated and accounted, a session record containing a summary of consumed resources and associated charges is created. The session records are then sent to the A4C server of the organization who requested these services. By accepting the session record, the requesting organization accepts paying for the respective service session. It is then up to the requesting organization to recover the costs.

In the second phase, the session records received from different organizations are aggregated by the BaseVO's A4C server and the charge for the composed service is created. The result of this phase is another session record which is delivered to the home domain of a user. The home domain will be charged by the organization which created the BaseVO for the aggregated service session.

The home domain will use this final session record for the bill that will be issued to the user. Finally, the home domain recovers the costs for the BaseVO from the user who actually consumed the service.

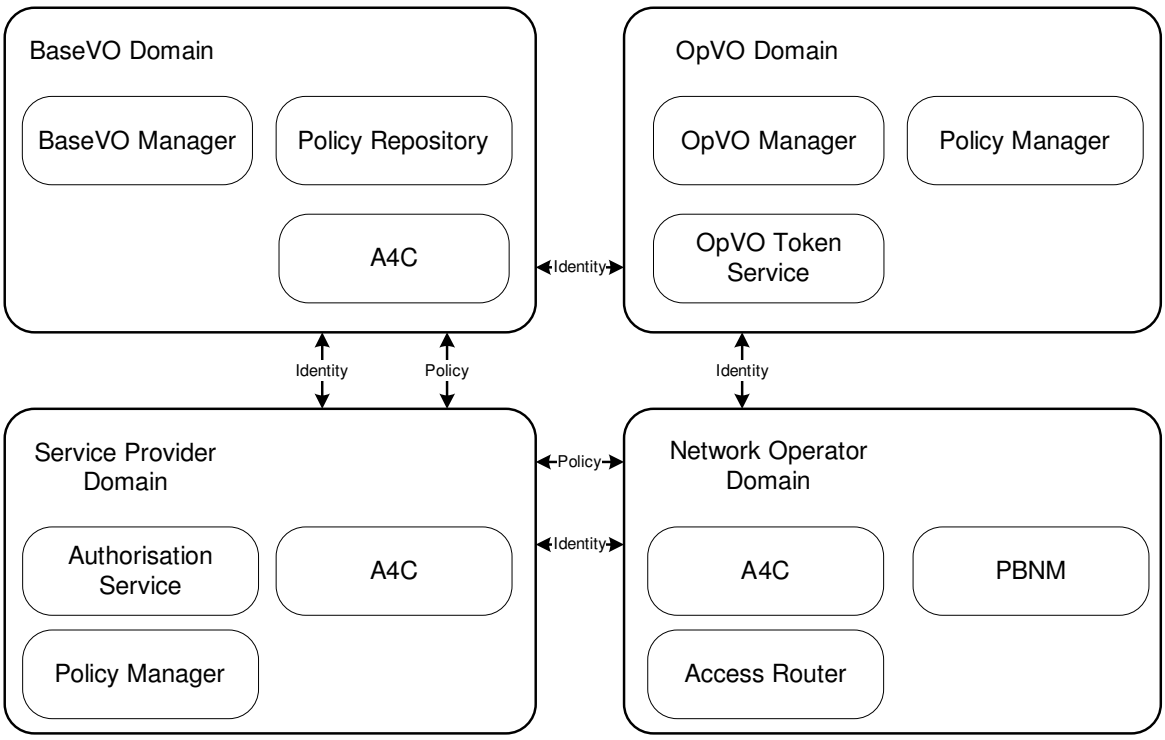

Figure 16: Security related to "AAA"

Figure 17 shows the accounting and charging data flow in the multi-domain grid environment, where network operators and service providers offer the complete service to the user. In each network operator and service provider domain there is an A4C server that receives and collects service usage data in form of accounting records from components participating in the service provisioning and performing service usage metering (e.g., the access router in the network operator domain). A4C servers in the participating domains transfer session records to the A4C server in the BaseVO domain, since the BaseVO domain is the central entity managing the service provisioning. Session records can include accounting records and charging records, depending on the business model and contract between the providers. The A4C server in the BaserVO domain sends aggregated session records to the A4C server in the user's home domain. These records enable the home domain to prepare a single bill, containing all charges for accessed services in the grid environment.

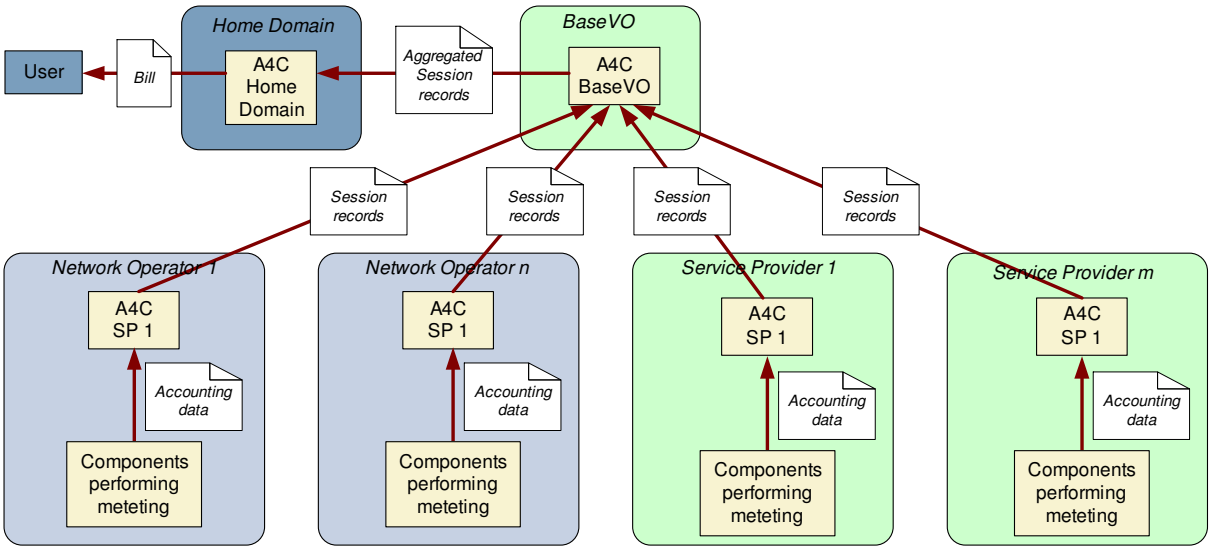

Figure 17: Accounting and charging data flow

An important aspect of this approach is the flexibility it gives to service providers with respect to the charging scheme they choose to apply for a service. For requesting a service from Service Provider 1 (SP1) (see Figure 17), BaseVO will receive from SP1 a session record containing the summary of the consumed resources and the associated charges. Further, BaseVO may choose to incorporate in the charges for the composed services only the final charges received from SP1 or it may apply some sort of charging mechanism based on the amount of resources that were consumed.

3. Overall Integration

In this chapter some selected cross layer aspects are described in greater detail. The notion "cross layer" basically relates to the technical aspects historically solved by the Grid community (Grid layer) and the network community (network layer). Cross layer integration within this context describes, how these selected issues interact with each other in order to consolidate the overall architecture. In the deep Akogrimo context, "cross layer" means technical issues with the involvement of more than two WPs of Activity 4 (Network, Network Middleware, Grid Infrastructure and Grid Application Support).

3.1. Service Level Description and Service Specification

A prerequisite to registering a service within an Akogrimo BVO is that the service provider is a member of the BVO. Members that are registered in the role Service Provider have the right to publish services into the GrSDS of the BVO. Information about BVO members and their roles inside OpVOs is stored in the Participant Registry. The Participant Registry maintains dynamic profiles that can differ in their content depending on the specific OpVO. Basic static user profiles are stored in the A4C server. In order to integrate a service into the Akogrimo platform a service provider has to take care of two parts:

- Service Description
- Service Interactions

The description is used in the finding phase. Service interactions occur in the binding and execution phase.

3.1.1.1. Service Description

The service description is used by the GrSDS to find suitable services for a customer. With the term service, in Akogrimo we mean a Web Service or WS-Resource. Their functional capabilities (services interfaces) are going to be described using standard technologies. Furthermore, in Akogrimo we need to describe non-functional capabilities, e.g. semantic information, QoS, access rights, etc.. The following description standards are going to be used:

- OWL-S mark-up of Web services will facilitate the automation of Web service tasks, including automated Web service discovery, execution, composition and interoperation
- WSDL
- WS-Agreement for defining the SLA template and contract
- WS-Policy

If the service is represented by a workflow, a workflow template has to be published into the workflow registry. The workflows will be described using:

- WS-BPEL

3.1.1.2. Service Interactions

Service related interactions occur between the OpVO and the service that belong to the hosting environment of the service provider. In order to interact with the OpVO, the hosting environment and the service have to support the following:

Interaction	Technology / Protocol	Description	Available Component (for use in the Hosting Environment)
SLA Negotiation	WSLA, WS-Agreement	The SLA Negotiator of the hosting environment talks to the SLA Negotiator of the OpVO. Once a agreement has been reached, the contract is stored in the SLA contract repository via the SLA-Access service.	SLA Negotiator
Policy Management	WS-Policy	WS-Policy is used to make statements about the policies that apply to the service or hosting environment.	Policy Manager
Registering for context information	WS-Notification for subscribing	The service can contact the Context Manager to subscribe to context information of a specific user.	
Receiving context information	WS-Notification	The service has to implement context sensitivity if desired.	
Authorisation	DIAMETER, SAML Assertions	The service has to find out if a service request should be authorized.	A4C Client
Configuring Network QoS	QoS Broker Web-service interface	The service provider has to understand the service interface of the QoS Broker.	
Receiving network metering information	WS-Notification	The service provider may use information about the network utilisation.	
Reporting SLA-Violations to the OpVO	WS-Notification	The service provider has to report SLA violations to the OpVO Manager. The service provider may use Akogrimo components for local monitoring and management.	Monitoring, SLA-Controller, SLA-Decisor, EMS

Sending Accounting information	DIAMETER	The service provider sends information about service usage to the A4C server. This includes information about SLA violations.	A4C-Client
Data Transfer	GridFTP, WS-Attachment	The service provider may use the Data Manager component if needed.	Data Manager, GridFTP Server component of GT4

Table 1: Service Interactions

3.1.2. Network Layer services

The Akogrimo project aims for an integration of the worlds of network and Grids, which presents some difficulties. The network layer typically uses a variety of protocols which have to perform under stringent time-constraints. This performance comes at the cost of interoperability. When there is a need for interaction between different systems that don't have a common language, additional effort creating some "translation layer" or similar is required. In the Grid world, however, interoperability is one of the main objectives. By making extensive use of web service based technologies, a common base is provided that makes it inherently easier to support different interactions between systems, should the need arise.

Taking into account the scenarios specified in D2.3.1, we can see that multimedia calls are important for Akogrimo. Initiating and managing multimedia calls is supported by the SIP protocol. On the other hand, for those calls to be practical, it is necessary that the network can provide the required bandwidth at all times. QoS may also be necessary for other operations requested by grid components. To this end, there will be components specifically designed to interact between network and Grid layers.

The SIP Broker is the component which will have a web service interface capable of receiving requests from GRID Entities. This component implements the integration of the GRID and Network Layers for session management. The services offered through its WS interface allow linking GRID-world sessions with Network SIP-based ones, creating true relationships between both layers from a conceptual viewpoint.

Additionally, the SIP Broker will offer other utility services, as the ones that allow GRID components to establish and transfer regular SIP sessions. To implement these services, the SIP Broker interacts with the corresponding GRID Entity and forwards the requests to the SIP Server.

The QoS Grid Gateway is the component that will handle requests from the EMS and forward them to the QoS Broker. This allows network QoS services to become part of the workflow, defined both according to the user profile (and subscribed services) and according to its current operations (e.g. emergency life support overrides contractual user capabilities). Network QoS is further detailed in Akogrimo deliverable D4.1.1 – Network Layer Architecture.

These services will be registered with the GrSDS, thus allowing any web service to discover them dynamically, along with information about their functionality and interfaces. The GrSDS is further detailed in D4.2.1 and D4.2.2.

Network QoS will, for implementation and scalability reasons, be supported by well defined QoS bundles, strongly influenced by the existing models for mobile technologies (UMTS). Each of the three defined bundles is designed for a specific usage profile, audio, video and data. A QoS

Bundle is comprised of several well defined services, which the user may choose from when using the Akogrimo network. Table 2 presents these bundles.

Signalling: This traffic is of the highest priority and time-critical. Has a low bandwidth requirement.

Interactive real-time: Time-critical traffic typically for interactive multimedia applications which are sensitive to delays and out-of-order packets.

Priority: Not time-critical, but important, such as multimedia streaming, or some grid application data exchange. Higher priority than Data Transfer, but lower bandwidth typically.

Data Transfer: Not time-critical but may be loss-sensitive.

Best Effort: As the name implies this service offers best effort. Its efficacy is highly dependant on network conditions. This is basically what today's well known Internet provides.

Type of Service	Bundle 1 Mixed audio + data [kbyte/s]	Bundle 2 High data + video [kbyte/s]	Bundle 3 Mostly voice [kbyte/s]
Interactive	10	20	10
Data	100	1000	1
Priority	1	200	1
Signalling	1	1	0
Best Effort	250	0	250

Table 2: QoS Bundles

3.1.3. Network Middleware services

On top of the basic network functionality is placed network middleware service provisioning. Here, especially network session based Signalling such as SIP-based sessions, but also AAAC based sessions are grouped.

The SIP and QoS cross-layer interfaces allow interoperability between the Grid and the network layer. This interoperability is somewhat limited as of now, but in the second phase of the project, a more consistent integration of those layers will be achieved.

The network middleware layer provides a set of basic infrastructure functions to the higher layers, including cross-layer A4C, presence and context management, and semantic service discovery. The corresponding traffic is mainly signalling sessions between components in the core network (or wired part of the access network). The QoS cross-layer interfaces offered by the network layer do not apply to core components, that are assumed to be well connected at all times. However, there are some network middleware sessions involving the mobile terminal, e.g. SIP presence publishing that can benefit from the QoS interface offered by the network layer.

On top of this network session related part Grid infrastructure related elements are added to the overall cross-layer service bundles. Here, the Grid Service Discovery System (GrSDS), provided by the Akogrimo middleware layer, is essential for interoperability and cross-layer interactions. Service providers, of e.g. the cross-layer network layer services, will register their services with the

Grid Service Discovery System. The GrSDS represents a "static" discovery registry with semantics capabilities where the static description of services (i.e. WSDL description and corresponding SLA template) is stored. When an entity needs a certain service, e.g. a service that virtualises network layer functionality, it will query the GrSDS for a list of relevant services, select and invoke the most appropriate. However, dynamic service attributes, e.g. processor load, number of instances running, queued jobs, network load etc., are not managed by the GrSDS. Details on the network middleware layer are found in D4.2.1 and D4.4.2

3.1.4. Grid Middleware services

In order to finally come from Grid resource to user-centric service, in the Akogrimo architecture the Grid functionality of the whole infrastructure is provided by the "Grid infrastructure" level. This level consists of the Web Services Resource Framework (WSRF) level and the OGSA services layers, being placed in the middle of the Akogrimo architecture participating as the "Grid glue" to the functionality of the Akogrimo system.

All Grid resources, both logical and physical, are modelled as services on the basis of Web service implementations. However, for the purposes of the resources modelling in the Grid the Web service interfaces must frequently allow for the manipulation of state, that is, data values that persist across and evolve as a result of Web service interactions. To achieve this, the Web Services Resource Framework (WSRF) defines a family of specifications for accessing stateful resources using Web services. Note that communication services are also described using this framework. The Web services layer, together with the WSRF, provides a base infrastructure for the OGSA services layer providing the overall Grid management functionality.

The basic motivation is to provide OGSA specific services implemented through the framework of the WSRF. The basic functionality that must be supported from the Grid Infrastructure Services layer, as identified by the consortium and compliant with the OGSA draft specification, can be categorized in the following:

Execution Management Services (EMS): This category of services comprises all the functionality that is concerned with the problems of instantiating and managing tasks, such as assigning jobs to resources, creating an execution plan, balancing the workload, optimising the performance, and replicating jobs to provide fault tolerance. Conceptually, these resources can be represented by the composed bundles ad indicated above.

Data Management: This category comprises all the functionality that is concerned with the access to and movement of large data sets, as well as data sharing, replicating and archiving of data.

Monitoring: This category comprises the services that are focusing on monitoring and managing of the web services within the layer.

Security: This category comprises the services that are concerned with the security issues of the specific layer. It comprises the services that will deal with the confidentiality of the communications and the authorization for execution within the system.

Policy management: This category of services comprises the functionality concerned with the management of rules and the policies which apply in the execution of services within the Akogrimo architecture.

Service Level Agreement (SLA): Services related to the enforcement of the SLA contractual terms that especially influence the execution of jobs within the layer.

The WSRF solves the problem of statefulness in the following way: it keeps the Web service and the state information completely separate. Each resource has a unique key, so whenever we want a stateful interaction with a Web service we simply have to instruct the Web service to use a particular resource. The motivation for these new specifications is that while Web service

implementations typically do not maintain state information during their interactions, their interfaces must frequently allow for the manipulation of state, that is, data values that persist across and evolve as a result of Web service interactions.

3.1.5. Grid Application Support services

At the Grid Application Support Service layer, we have to distinguish between management and business (sold by Service Providers and made available in the VO) services.

The management services will be stored in dedicated repositories and they are out of the scope of this section, while, with respect to the business services, their description has to be compatible with a discovery mechanism that could be compared with a sort of "yellow page" registry.

From the GASS layer viewpoint in order to allow the discovery of services matching the requestor requirements, it is necessary to provide a description of the service, which has to include at least the following information:

- Service Provider information that allow identifying the provider which has published the service.
- Functionalities of the service.
- List of available web methods and their technical description (e.g. using WSDL). Furthermore a semantic description of the method logic is recommended.
- Under which potential QoS conditions the service can be provided.
- A reference to the service that should be contacted in order to start the negotiation for renting the service use (the result of this negotiation will be the agreed SLA and a reference to the service instance)

These are the overall requirements to be addressed by the GrSDS with respect to the GASS layer but it is necessary to point out a particular case that should be treated as any other kind of search: a workflow is the result of the search.

In this case, a composite service (workflow) has to be instantiated then it is necessary to publish the same information required for simple service, but an additional property is required to understand the composite nature of the service. In particular, instead to publish the endpoint of the service negotiator, it will be published the identifier of the associated workflow template stored in the WF registry in order to retrieve it during the OpVO creation process. To each workflow template, a .pddIn this case, a composite service (workflow) has to be instantiated then it is necessary to publish the same information required for simple service, but an additional property is required to understand the composite nature of the service. In particular, instead to publish the endpoint of the service negotiator, it will be published the identifier of the associated workflow template stored in the WF registry in order to retrieve it during the OpVO creation process. To each workflow template, a ".pdd" file will be associated for each service to be involved in the workflow execution. This ".pdd" file contains semantic information to do another query against the GrSDS in order to find the services to be involved in the WorkFlow execution. Of course, each of these services will be negotiated with the related service provider. In this case the "provider" of the Work Flow is the VO itself.

The component that provides capabilities for searching services published attaching this kind of information will be supported by an UDDI based registry and by a SLA Template Repository.

The following figure shows a sketch of a possible high level architecture of GrSDS.

3.2. Quality of Service

In an environment with mobile users, ensuring quality of service is a complex task. The challenges that mobility imposes have to be dealt with. These are terminal mobility, user mobility, session mobility and loss of connection. This means that 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. 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.

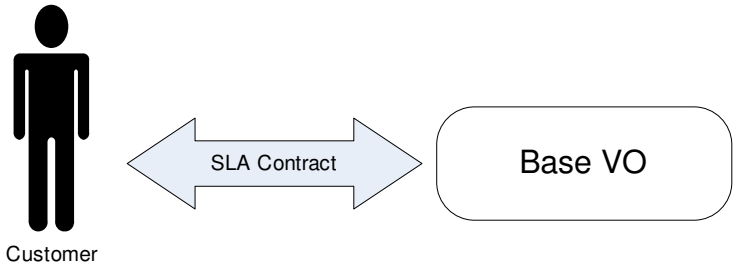

Figure 18: Buying a Service

Quality of service is the key to the overall integration of network and grid services. The service that the user buys from the Base VO is realized by combining functionality and service characteristics from different entities. The SLA contract binds the characteristics of these entities and describes the overall service features that the user can expect.

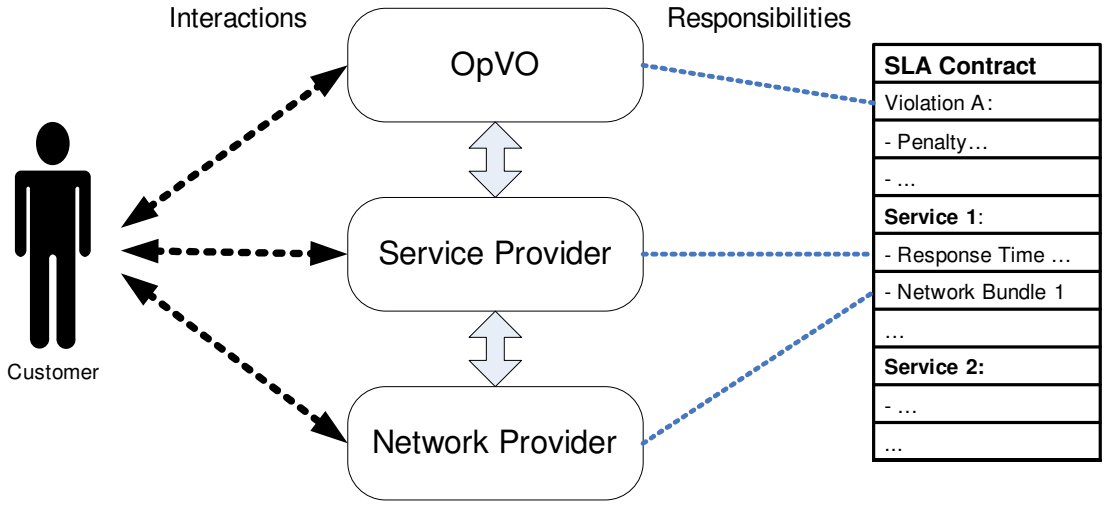

Figure 19: Using a Service

After the SLA contract has been signed it is stored in the SLA Repository of the Base VO. During the setup of the OpVO the relevant parts of the SLA contract are propagated to the individual service and network providers.

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 (SLA's) 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.

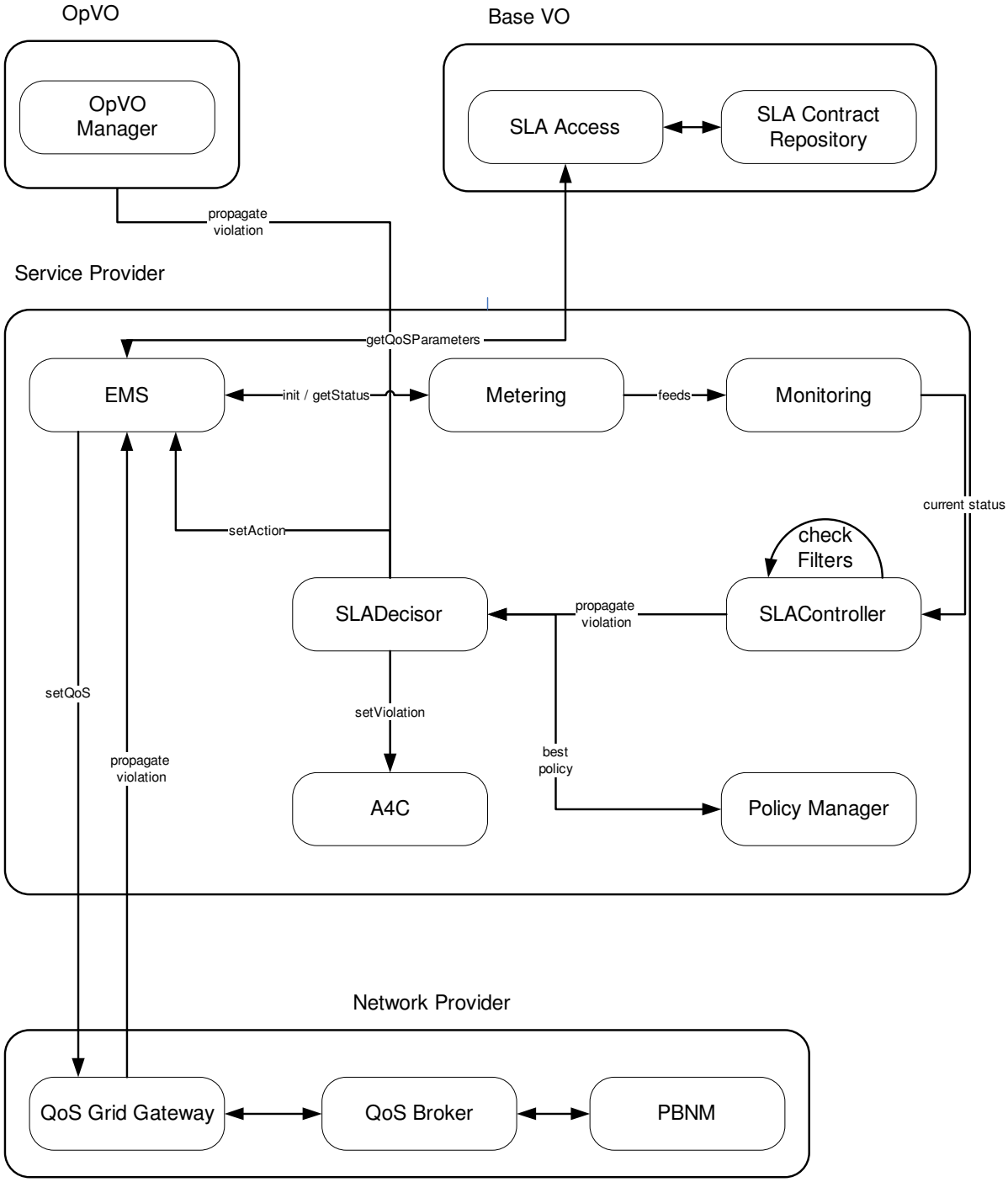

Figure 20: Quality of Service

3.3. Inter-operability of Globus GSI and WS Sec Framework

For the purpose of testing the interoperability between the GT4 and the WSRF.net security infrastructures, two WSRF-compliant secure web services have to be developed on each platform. The first implementation of these services will make use of the authentication mechanisms offered by each infrastructure but without server-side authorization. After testing

the interoperability between the authentication frameworks and having reached a plausible solution, the services will be reconfigured so as to use the authorization mechanisms that each infrastructure supports and testing will be resumed with respect to the interoperability between the different authorization frameworks. Since our goal is to test the interoperability between the different security infrastructures, the services used in the tests shall be "secure" but of very simple functionality.

3.3.1. A simple interoperability example

The WSRF web service that will be developed on GT4 platform shall expose four simple methods: *add*, *subtract*, *multiply* and *divide*. Each of these four operations shall be configured to work with a different set of authentication methods as GT4 allows configuring authentication on method level:

- The *add* method shall only be invoked using GSI Secure Conversation (WS-SecureConversation/message-level security)
- The *subtract* method shall only be invoked using GSI Secure Message (WS-Secure/message-level security)
- The *multiply* method shall be invoked using GSI Secure Conversation or GSI Secure Message.
- The *divide* method shall only be invoked using GSI Transport (TLS/transport-level security).
- The rest of the method that maybe exposed may be invoked with any of the authentication methods.

These four methods also use different levels of protection:

- The *add* method can be invoked with either encryption or digital signatures.
- The *subtract* method can only be invoked with digital signatures.
- The *multiply* method can be invoked with encryption.
- The *divide* method can be invoked with either encryption or digital signatures.
- No protection level is specified for the rest of the methods.

A WSRF web service with similar functionality shall be developed on WSRF.net platform. Our primary goal is to test the interoperability of the two security infrastructures platforms with respect to the X.509 certificates. We should then consider using a customized X.509 certificate that will meet the specific needs of the Akogrimo platform. The GT4 container that will be hosting the GT4 secure web service described above is configured to work with a specific certificate authority. The current GT4 installations that are being used are configured to trust a particular CA named SimpleCA. SimpleCA is a basic certificate authority package included with GT4. Although the quality of the X.509 certificates that the SimpleCA provides is rather low, it can be used for initial testing purposes. GT4 can be configured to use an existing full-blown CA in the future for stronger authentication.

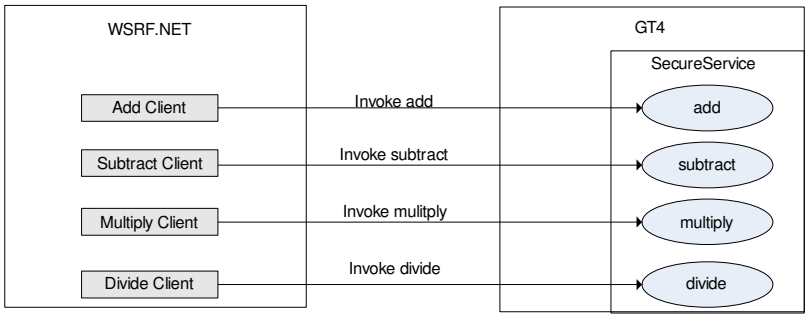

Figure 21: Invocation of GT4 secured services from .NET clients

In the first phase of the interoperability testing the focus is on testing the authentication and protection schemes of GT4. For this purpose a number of different clients developed in WSRF.net, one for each operation that the GT4 secure test service exposes should be involved. Each of these clients must be configured to use the authentication method that the corresponding operation needs.

On a later stage these tests shall be reversed, using GT4 clients and the secure web service developed on WSRF.net that will make use of the security mechanisms offered by the platform. Apart from testing the different authentication mechanisms, these tests will help us conclude on which is the most appropriate authentication method. On the last step, a secure and direct communication between the WSRF.net and the GT4 secure services shall be established using the best authentication method.

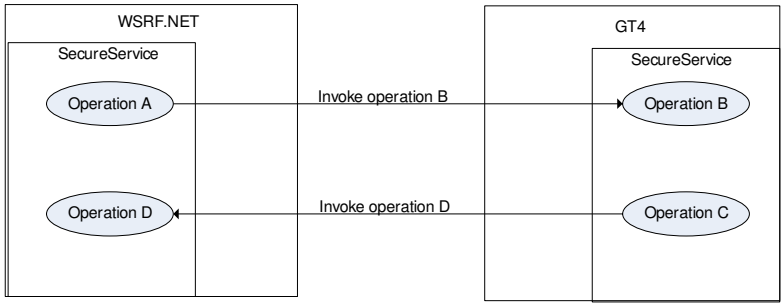

Figure 22: Operations of secured services

3.3.2. A WSRF.net client invokes GT4 Service

The WSRF.net clients that will be developed for invoking the different methods that the GT4 secure service exposes, shall take advantage of WSE functionalities for security applied to message-level. More specifically, the clients developed for invoking the add and multiply methods shall be using WS-SecureContext whereas the client invoking the subtract method shall be using WS-Security. These clients will make use of a certificate issued by the GT4 trusted CA.

As far as the divide method is concerned, although authentication is configured on method-level, using GSI Transport means that the whole GT4 container must be started with transport-level security activated. In order to define the form that the SOAP message requesting the invocation of the divide method is expected by the GT4 secure service, it is foreseen to have interception of such requests produced by the GSI Transport client and send for analysis to the WSRF.net side. It must be stressed that the focus should be on achieving interoperability on the message-level and then moving to the transport-level.

3.3.3. GT4 client invokes WSRF.Net service

The WSRF.Net secure service that shall be developed for the tests will implement the methods that use message-level (add, subtract and multiply) using WS-Policy to specify the certificate to be used for invocation.

Since transport-level security cannot be configured at method-level on WSRF.net, the SSL protocol will be used to access all methods exposed by the WSRF.net secure service when testing the transport-level security. It should be noticed that when securing the actual Akogrimo Grid wed services authentication will be configured on service level not at method level, since there is no reason for using different authentication mechanisms per method.

3.4. Security

Within Akogrimo not all security related aspects are going to be worked on. Reasons for this is that many of the technical challenges are not really specific to Akogrimo, but describe general problems which could be equally worked on in other projects and are of a more general nature.

Attacker / Target	User	Terminal SIP Client MIPL PANA	OPVO WF Manager BP User Agent,…	Base VO CM	Customer Domain Accounting ID	Network Provider SIP Broker SIP Server	Service Provider EMS SLA Negotiator Service instance
User	4/4/4	4/2/2	1/2/1	2/3/2	4/3/2	4/3/3	1/1/1 Accounting, SLA
Terminal	4/3/2	4/3/2	4/3/2	4/3/2	4/3/2	4/3/2	4/3/2
OPVO	1/1/1	1/1/1	1/2/1	1/3/1	1/2/1	4/4/4	1/3/1
Base VO	1/1/1	1/1/1	1/2/1	4/4/4	1/2/1	2/3/2	1/2/1
Customer Domain	4/1/1	2/2/2	1/3/1	1/3/1	4/3/3	4/3/2	1/3/1
Network Provider	4/3/3	4/2/2	2/3/2	2/3/2	4/4/4	4/3/2	4/4/4
Service Provider	2/2/2	3/3/3	1/1/1	1/3/1	1/3/1	1/3/1	1/3/1

Table 3: Security focus within Akogrimo

The following table presents a quantification of these issues based on the high level architecture according to Figure 2. The numbering in the table indicates first the relevance within Akogrimo, the second number indicates the likelihood whereas the third number indicates the impact always in a range between 1 (high) and 4 (low).

Within Akogrimo the areas with highest relevance are being worked on. In a further iteration it will be checked, if the areas with high relevance ("2") will be still an issue and all the other areas -

acknowledging that there are of course of high impact - are considered to be general and not specific to Akogrimo and thus not considered in the security evaluation.

The following sections present the mechanisms Akogrimo is providing against several attacks to the target areas.

3.5. Grid Application Support layer Security: Domains and Roles

The domains involved in a typical Akogrimo business flow are: BVO domain, OpVO domain, Customer domain, Network Provider domain and Service Provider domain. While the participant of the BVO and OpVO domain can play the following roles: BVO/OpVO Administrator, Customer, Service Provider, Network Provider and User.

The table below shows which domains interact between them.

	BVO	**OpVO**	**C_D**	**SP_D**	**NP_D**
BVO	X	X			X
OpVO	X	X		X	X
C_D	X	X	X		
SP_D	X			X	
NP_D					X

Table 4: Intercommunications between domains

The first column shows the domain which the message comes from, the first row the domain where the message arrives to.

In this table, the user domain is not shown because we are supposing the users are part of the customer domain. Furthermore the OpVO domain is a sub domain of the BVO domain.

3.6. Grid Infrastructure Measures taken for threat model of Service Provider

The security model that will be adopted in the Grid Infrastructure will be oriented towards the Grid Security Infrastructure of GT4, as the most components are implemented through it.

The protection that is offered involves the following security primitives:

- Authentication
- Authorization
- Confidentiality/Privacy
- Integrity

Critical components that might be exposed to external attacks (for instance Man-in-the-middle attack or Denial of service attacks) should be protected behind firewalls and intrusion detection tools that would enhance the perimeter security of these components.

3.6.1. Attacks coming from the Service Provider and to the Service Provider:

This mainly involves the last column and the last row of the security model table, and these attacks can be considered very critical for the operation of the Akogrimo platform, since these components interface and interact with a large number of different components.

Authentication: A strong authentication mechanism can be applied so as to identify constantly the answer to the question "who is who" in this interaction scheme. All parties involved must exchange authentication information in order to achieve mutual authentication. For this reason, X.509 certificates are applied and it is also foreseen to have also custom security tokens through the WS-Security specification.

Authorization: A strict authorization scheme has to be applied in order to put constraints on the functionality and interactions between the components. The interactions have to be foreseen in such a way that only components that have to communicate will be allowed to communicate.

Through GSI we have the possibility to allow for client-side and server-side authorizations. Additionally, GSI provides an infrastructure to easily plug in third party authorization mechanisms. Authorization modes involve among others:

- Gridmap: A list of 'authorized users' akin to an Access Control List (ACL) means only the users are listed in the service's gridmap may invoke the service.
- Identity authorization: A client is able to access a service if the client's identity matches a specified identity. This is like having a one-user gridmap e.g. whereas the gridmap is represented as a file in the system.
- SAML Callout authorization: Authorization decision is delegated to an OGSA Authorization-compliant (OGSA-Authz a GGF working group) authorization service. One of the main technologies used in these components is SAML (Security Assertion Markup Language).

Confidentiality / Privacy: The communication between the various components (within the Service Provider domain, but also across other columns) will be encrypted in such a way to enable the privacy of the conversations and confidentiality on the content. For this reason, GSI allows enabling security at two levels: the transport level or the message level. Using transport level security then the complete communication (all information exchanged between the client and the server) would be encrypted. At message level security then only the content of the SOAP message is encrypted, while the rest of the SOAP message is left unencrypted. Both transport-level and message-level security in GSI are based on public-key cryptography and therefore can guarantee privacy, integrity and authentication. WS-Security standard and Transport Layer Security (TLS) protocol are the main technologies for this.

Integrity: In order to have the integrity of the exchanged messages between the Service Provider components (and the components of other columns) digital signatures will be deployed. Additionally, both WS-Security and TLS for the message level and transport level security can contribute to the integrity security aspect.

From the WSRF.net perspective, the same technologies can be deployed in order to allow for a message level security and transport level security between the communicating entities.

Moreover, the management of SOAP security headers is based on Microsoft Web Service Enhancement (WSE) toolkit, which implements a subset of the WS-* specifications. WSE is based on a pipeline filters model; these filters elaborate the SOAP message body, for instance, encrypting an outbound message's body and decrypting an inbound message's body. WSE allow signing and encrypting SOAP messages using different kind of token: X.509, Kerberos ticket, custom binary token, custom xml token, SAML token and so on. Using the capabilities provided by WSE, it is possible: to publish the service security requirements, to enforce security requirements on incoming and outgoing SOAP messages, and to validate security tokens.

3.7. Network Layer Security

From the table we can see that most of 4.1 components are not considered for security in Akogrimo. This does not mean that they are secure, but rather that in Akogrimo we should deal

with threats specific to Akogrimo, not go about solving problems that are, or could be, dealt with elsewhere.

3.7.1. Attacks considered

From the table, we rule out attacks to the Terminal. We need to consider attacks to Network Provider from the BaseVO and OpVO.

We may have to deal with attacks to Customer domain and User. Feel free to add your thoughts to this document.

3.7.2. Fundamental Assumptions

- The core network is safe; the Akogrimo network will provide substantial defenses at the perimeter and less so to the components that form the basic infrastructure.
- Out of scope for Akogrimo (not specific for our project)
    - Attacks from the user, using malware, some kind of "brute force" technique or just using software exploits.
    - DoS attacks.
    - Low level attacks (e.g. someone jamming the radio frequency used by Wi-Fi); not specific to Akogrimo.
    - Exterior of Akogrimo network; not specific to Akogrimo.

The remainder of this document will consider the attacks from OpVO and BaseVO to Network provider, which are marked green in the table.

3.7.3. Affected Components

The affected components are the components which interact with the BaseVO or the OpVO, namely the SIP Broker.

3.7.3.1. SIP Broker (formerly SIP Server)

The SIP Broker allows the invocation of SIP services, such as session transfer and third party call control, from grid applications. Furthermore, it now aims to support mobile and nomadic grid services.

(Next are excerpts from the E2E security group document.)

3.7.3.2. VO security

The adopted solution to take into account the opVO security perspective takes advantage on the extensibility and backward compatibility of the SIP protocol, and consists on the definition of a new header field containing the opVO membership token, which will be included into the SIP signalling messages involved during the grid-triggered SIP call or SIP transfer. This new header will be processed at application level at the MTs, providing end-to-end security. Anyway, it will have a meaning only within the Akogrimo context; any other "non-Akogrimo" SIP entities are requested to ignore it, as SIP protocol specifies.

So using this approach, when a MT is requested to be the initiator of a SIP call, it can check if the request coming from the SIP Grid Gateway contains a valid opVO token prior to start the communication. In this way, the initiating MT can check that the call is authorized. The only requirement is, this information should be included somehow in the Workflow Manager request

to the SIP Grid Gateway, because the SIP call / transfer is sent to the MTs by the SIP Grid Gw. in response to a WS request from the WorkFlow Manager.

In the other hand, and in the case of a grid triggered SIP call, the initiator MT will include its opVO token in the request, so the receiving MT can be sure the call is authorized. MT will obtain its opVO tokens by connecting to the corresponding VO User Agent during the VO login. The User Agent is created by the workflow to represent a particular user to the workflow. The user "connects" to the User Agent from a particular MT (authentication, selection of opVO, authorization, etc). At this point, the MT can be given its opVO token.

The best aspect of this approach is that the SIP network entities do not need to be secured from the opVO security model perspective - being outside the opVO firewall they are not a security threat (other than from a denial of service). Of course, self protection for the SIP Grid Gateway and the SIP Server would be beneficial, and these could use the same mechanism (i. e. the baseVO dynamic perimeter) to ensure they only respond to valid Akogrimo opVOs and MTs.

3.7.3.3. WS interaction path

As described earlier, in principle one attacker could impersonate the identity of the Workflow Manager and make a non-authorised use of the WS exposed by the SIP Grid Gateway. Apart from this, the WS communication path can be intercepted and the exchanged messages modified by a non-authorised entity. In order to prevent this, specific WS security mechanisms are envisaged.

WS-Security core specification defines several mechanisms to provide protection against SOAP attacks. It includes also mechanisms to detect if some message has been manipulated (XML signature) and encryption mechanism as well (XML encryption).

However, the adoption of the opVO security model provides a higher level of protection, because the attacker should include a valid opVO token in the request; however, it will be rejected.

3.7.3.4. RMI path

Internal communications between the SOAP interface and the SIP Grid Gateway broker engine is made by means of an RMI interface. This means that remote clients could access the classes and resources that the RMI server (the broker engine in our case) makes accessible. So if some malicious entity knows how to invoke the corresponding methods, it could impersonate the identity of the real SIP Grid Gateway SOAP interface and could make unauthorised requests.

To prevent these kinds of attacks, we must take advantage on the Java Security Model. Using this, you can restrict the actions that remote entities can perform. By default, an RMI program does not have a security manager installed, and no restrictions are placed on remotely loaded objects. Anyway, the java.rmi package provides a default security manager implementation that can be used, which requires the specification of a security policy file containing specific permissions regarding which operations are granted, and from which locations (machines, ports). So the best way to proceed is to specify that only requests from the machine in which the SIP Grid Gateway SOAP interface is running are allowed.

The current implementation of the SIP Grid Gateway enables the configuration of the policy file to be used. No security policies are applied if this file is not specified.

3.7.3.5. SIP path

As RFC 3261 mentions, SIP is not an easy protocol to secure. Section 26 of this request for comment describes several SIP-related security considerations that as a general rule should be

considered in any SIP platform. These general considerations will be described in a separate document.

The SIP-related security problem last section describes (some malicious entity passing as the SIP Grid Gateway User Agent) can take place on the network side of the SIP path (SIP Grid Gateway – SIP Server hop) or in the user side (SIP Server – MT hop). The usual solution for the network hop is the usage of some certificate-based security mechanism, like TLS, which provides data integrity and confidentiality, preventing from MitM attacks.

Additionally, in order to avoid a malicious entity registering itself as the SIP Grid Gateway, deregistering the valid IP address binding and assuming the identity of this component, the SIP Registrar (embedded into the SIP Server) could be also configured to accept only a registration binding from a preconfigured IP address, or HTTP Digest authentication mechanism described in RFC 3261 (which is based on HTTP authentication) could be used. This approach would provide also good DoS attacks protection, since the server processing the SIP registration can remain stateless until a new request arrives, and all transactions from non-registering entities are rejected.

Finally, to avoid a direct attack to the MTs making use of the peer-to-peer nature of SIP (i.e. the attacker sends the request directly to the mobile terminal, impersonating both the SIP Server and the SIP Grid Gateway) network layer security mechanisms can be used (e.g. using an IPSec connection that refuses all IP packets not coming from the well-known SIP proxy IP address and port). Digest User-to-user authentication (RFC 3261, section 22.2) can be also used, but it implies more network traffic and a higher delay (e.g. grid call or grid transfer establishment time will be higher). Anyway, in this case the OpVO security model provides again a reasonable security level.

3.7.3.6. QoSB Grid Gateway

The QoSB Grid Gateway is not considered in the table, but since wp4.3 will adopt the GSI of GT4, we will also develop the QoSB GGw along those guidelines.

3.8. Security model applied to defend BVO and OpVO services

This section will provide an overview of the adopted model and information about the integration with the overall Akogrimo security model.

The protection that is offered involves the following security primitives: Authentication, Authorization, Confidentiality and Integrity.

These basilar security primitives could be enhanced by the introduction of firewalls and intrusion detection tools, but the design of BVO and OpVO security model has just focused on the basilar security primitives.

3.8.1. Attacks coming from the Customer or Service Provider domain and to the BVO or OpVO

These attacks are very critical because:

- The target services are exposed to the external "world"
- The impact on the BVO domain can be very high because they could start actions inside that domain that affects several components and can bring to a misbehaviour of the BVO itself.

In order to restrict the number of components involved in this kind of attacks, the architecture design has restricted the access point to the BVO through the User Agent.

Authentication

The authentication mechanism has been designed to take into account the mobility of BVO invoker. The proposed design assigns a central role to the trusted Network Providers as authentication authority.

The technologies involved in the authentication process are:

- Existing implementation of WS-Security specifications (e.g. WSE on MS platform) to manage security token included in the SOAP headers of the incoming messages.
- Security Access Markup Language (SAML) to manage a single authentication mechanism through different domains.

Authorization

The authorization will follow independent models in each independent domain and, in the BVO, the authorization focus on the message level. A basilar pull authorization model has been adopted but the architecture design can be easily modified passing to a more performance-effective push model.

Confidentiality

The communication between the external user (customer or provider) can require for encryption. Message level confidentiality is planned to be used and the current Web Service security technologies enable confidentiality of the SOAP message content. The confidentiality of SOAP message content will be based on the use of public-key cryptography that can be considered a standard approach in the Web Service world.

Integrity

To guarantee the integrity the message will be signed by the sender and also in this case Web Service security technologies provide features to sign the SOAP message. The available frameworks provide implementation also to manage SOAP messages signature. About integrity and confidentiality similar problems have been identified about the public key distribution but they have been solved following the same approach.

3.8.2. Attacks coming from the BVO and to the BVO

All the components that are involved in possible relevant attacks are listed in **¡Error! No se encuentra el origen de la referencia.** and **¡Error! No se encuentra el origen de la referencia.**. These attacks are very relevant because involve many different components inside the BVO domain and their misbehaviour can affect several aspects of the BVO management.

The security model applied for this kind of attacks is not different from the one described in section 3.8.1 (the same technologies can be applied here as well), but here intra domain communications are involved then some simplifications can be assumed that don't require trust with external authority.

Authentication: The authentication will be performed again at message level but now the authentication authority is inside the same domain. Furthermore, all the issues related to the user mobility do not arise here.

Authorization: The authorization will be decided following the same approach described above. In this case, authorization attributes can be included into the VO token or the same pull model describe above can be applied.

Confidentiality and integrity: it is managed as in the general case but now the management of public key distribution is simplified because all the involved entities into the communication are not mobile.

3.9. User Identification in Akogrimo

The Akogrimo Identity Model is very challenging due to the special requirements and needs that are specified. The following characteristics need to be considered:

- The user and the MT are decoupled. That is, a single MT can be used by different entities.
- The user only needs to authenticate himself once against his Home Domain before accessing to the Network and Different Grid Services.
- The user must be able to decide which attributes are given to each Grid Service represented at an OpVO.
- The user can have different identities at different OpVOs.
- The user can be anonymous and/or pseudonymous.
- The user can have more than one Home Domain.
- The user can link its Home Domains.
- The user can delegate its identity to other entity.

In order to fulfil these characteristics, the following requirements and assumptions are necessary:

- The Home Domain of the user and the Customer Home Domain of an OpVO need trust relationships.
- The Customer Home Domain and the OpVO Home Domain need trust relationships.

3.9.1. A4C deployment and process

Figure 23 shows the A4C infrastructure. Every domain has its own A4C server. The figure shows only one A4C server per domain, but more than one A4C server can be deployed in a domain for e.g. performance or redundancy reasons. The SAML authority in the home domain supports the authentication process and manages identity tokens. The A4C clients in the service provider domains represent network components that require authentication or provide accounting information.

The A4C servers maintain trust relations with other servers as shown in Figure 5. The trust relation means that the servers trust each other and accept messages from each other, e.g. in case of authentication the authentication decision is accepted from the partner A4C server. A trust relation between A4C servers exists between

- Home and customer domain
- Customer and BaseVO domain
- BaseVO domain and service provider domains in the virtual organization

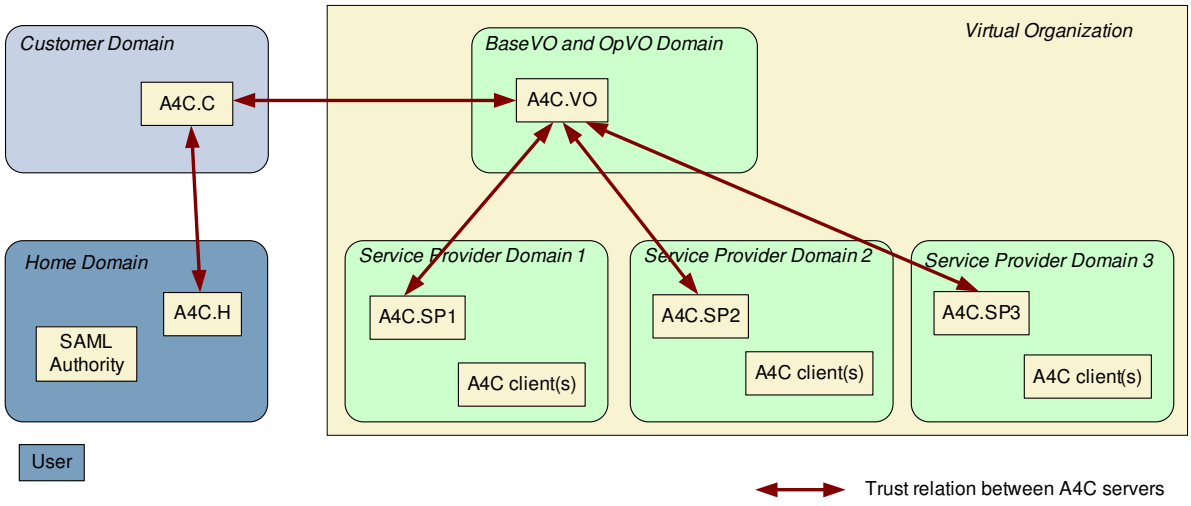

Figure 23: A4C infrastructure

3.9.2. ID Token and SAML

Akogrimo uses the Security Assertion Markup Language (SAML) to send security information in the form of authentication and attribute assertions to the Akogrimo components. SAML provides an additional security block concerning high confidential information (like authentication and attribute information of a user) in the Akogrimo architecture. SAML is a secure interoperable language used to share user's information from the A4C Server to the other components in order to provide Single Sign-On –SSO for short- capability to the user and to offer attribute sharing of the user to other components. In order to provide SAML messages, a SAML Engine is needed in Akogrimo. This is the SAML Authority. The SAML Authority is part of the security infrastructure in Akogrimo. It generates XML messages based on the SAML standard to send authentication and attribute information. The SAML Authority is an internal subcomponent of the A4C Server. It aims at supplying IDTokens and SAML assertions to the A4C Server. The A4C Server contacts the SAML Authority when it requires to generate IDTokens and to verify these tokens presented by different components.

3.9.2.1. IDToken generation

The IDToken within the Akogrimo context comes from the idea of the SAML Artefact defined in the SAML protocol. The use of a pointer to a SAML protocol message (in other words, a SAML assertion) to be sent to a Service Provider instead of sending the direct SAML assertion seemed a very promising idea for Akogrimo. The Mobile Terminal could receive it the first time the user would authenticate against his home A4C Server. Then, it could be used from the MT to authenticate against a service provider, without implying too much overhead at the MT. However, the definition of the SAML Artefact implies the one-time use restriction. This constraint means that each time the user (with the MT) wants to authenticate to a service, he should request to the A4C Server to create a new SAML Artefact for that service. Since this restriction is not desired for Akogrimo because it generates too much traffic between A4C Server and Mobile Terminal, Akogrimo takes the idea of the Daidalos project in the sense that they modify the design of the SAML Artefact, to be able to be used by the MT several times without exposing an additional security risk. The major benefit of the SAML Artefact with reuse capability -namely ID-Token in Akogrimo context - is that it enables the generation of different and personalized SAML assertions to different service providers without needing major overhead or additional traffic flow.

The SAML Authority - as part of the A4C Server- is in charge of the generation of the IDToken. The IDToken as a string contains the following elements:

- SAML Artifact: This parameter is a random-generated number. It is the pointer of the SAML assertions in the SAML Authority. This number remains the same at each usage of the token.
- Serial Number: This parameter is a counter. It will be increased by 1 each time the IDToken is used, in order to avoid replay attacks.
- Random Number: This parameter is a random-generated number and it is changed each time the MT uses the token. It is used to avoid security attacks.
- Signature: The signature of the issuer of the token. The IDToken will be signed by the SAML Authority of the A4C Server, the first time it issues the IDToken and sends it to the MT.

3.9.2.2. IDToken usage

The IDToken, as explained in the previous section, is created from the SAML Authority within the authentication process of the user and it is sent back to the MT when the authentication succeeds, as part of the response of the authentication.

The MT then stores the IDToken since will be needed in the following message exchanges with other components. When the MT wants to send an access request to a component, it must first update the IDToken, in order to make it secure. This update must be done each time the user wants to reuse it. The update consists of increasing a unit in the Serial Number and generating a new random number. It then needs to sign the new IDToken, so that the IDToken is completely updated. At this moment, the MT can already use this updated IDToken.

When a component receives an access request from the MT, it receives appended the updated IDToken. In order to know if the user is authenticated, it requests the A4C Server for information of the token. The A4C Server then, contacts the SAML Authority to validate the token and obtain the authentication information in the form of a SAML assertion. The validation of the IDToken consists of verification of the sequence number and signature of the IDToken. When the token is valid, a SAML assertion is issued in order to provide the authentication information of the user. The SAML assertion will contain the name of the user and the authentication method proceeded.

3.9.2.3. SAML Authority

The SAML Authority is responsible for the creation of SAML-based authentication and attribute-assertions and provides the functionality of creating, storing and managing SAML assertions and ID-Tokens. The following picture represents the internal architecture of the SAML Authority Engine and the communication with the A4C Server:

The components depicted in the SAML Authority are the following:

- SAML Engine- This is the main element in the SAML Authority. It is an application server that receives requests in the form of SOAP messages from the SAML Authority Client at the A4C Server. It understands the requests, resolves internally with the other components the result, and generates the SOAP Response with the result included at the body.
- IDToken Generator- This element creates the IDToken. It is initiated from the SAML Engine, when it receives the request of an issuance of a token for a user.
- Authentication Assertion Generator- This element generates the authentication assertion when it is invoked from the SAML Engine.

- Attribute Assertion Generator- This element generates the attribute assertion when it is invoked from the SAML eEgine.
- Database Interface- This is an interface used by the SAML Engine and also by the generators to access the database of the SAML Authority.
- SAML Authority DB- This element stores the IDToken and assertions of the users when requested by the SAML Engine or the generator entities.
- SOAP Library- This library is used for the understanding of SOAP Requests and the generation of the SOAP response. It is used by the SAML Engine.
- SAML Library- This library is used by the SAML Engine and the generator entities, in order to create SAML messages.

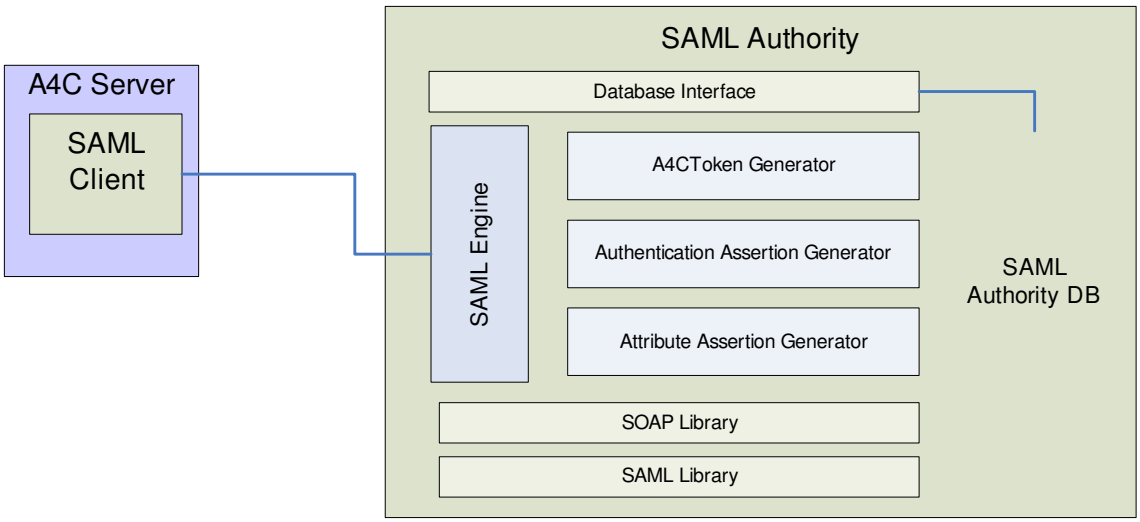

Figure 24: SAML Authority component view

3.9.2.4. Authorization and Centralized Policy Management - Security Policy Case

An Akogrimo system consists of one Base VO, a set of Service Providers, a set of Network Providers and a dynamic set of OpVOs, created on request and dissolving when complete, usually addressing a single application domain. Each if these can be considered as a separate administrative domain, i.e. for the purposes of this section, a separate security domain. Each domain will have assets to be protected, and will achieve this by restricting the actions that can be performed on these assets to authorized actions by authenticated actors. Deciding whether an action is authorized requires a decision based on the requesting actor, the action and the target asset/actor. The criteria for making these decisions are normally expressed as a security policy. Each domain will therefore have a Policy Management Framework (see PMF section). It also needs a mechanism to identify and authenticate actors and targets. Each domain therefore has a membership list and an Authentication Service.

There are two basic units of operation within Akogrimo, the BaseVO and the OpVOs. Both exploit dynamic distributed security perimeters to protect themselves, their assets and their members (see DDSP section). In essence, each member of the VO has its own identity token (key pair) and is given a membership token confirming its membership of the VO (VO and member identities, signed by the VO). VO assets, and other members, only respond to requests from other members, thereby isolating the VO from the outside world. By including a role, identity or pseudo-identity (identity w.r.t. the VO) in the membership token, fine-grained control over the internal operations is possible, thereby helping protect against insider attacks. The

membership token is signed by, or on behalf of, the VO administrator. Messages can be signed and/or encrypted using the membership token, providing VO-level integrity and confidentiality (everything in the VO can see/change anything), or, if the administrator disseminates them, by role or (pseudo)identifier tokens, providing end-to-end confidentiality (only the sender and receiver can see the message content.

3.10. Authentication Key Management Infrastructure and Certification Authority

With the use of encryption methods provided by public-key cryptography a communication between partners is more secure. But this encryption method is not protected against misuse of an intruder whether steeling private keys or act as man in the middle. To make sure that the holder of a key pair is the entity (user or service) he claims to be, Akogrimo should provide a PKI (Public Key Infrastructure).

3.10.1. Architecture

The PKI consists of at least one Registration Authority (RA) and one Certification Authority (CA). Whenever an entity needs a certificate (X.509) that claims its identity, a certificate request has to be sent to the RA. The RA verifies the identity of the requester by given credentials and forwards the request after a successful verification to the CA. The CA issues the certificate and signs it – it is expected that each member trusts the CA, thus the CA-signature means doubtlessness. To guarantee this doubtlessness it has to be assured that

- Each member trusts the CA
- A policy guarantees a reliable way of validating identities
- A policy guarantees a reliable way of issuing certificates
- The certificates can be removed in case of misuse

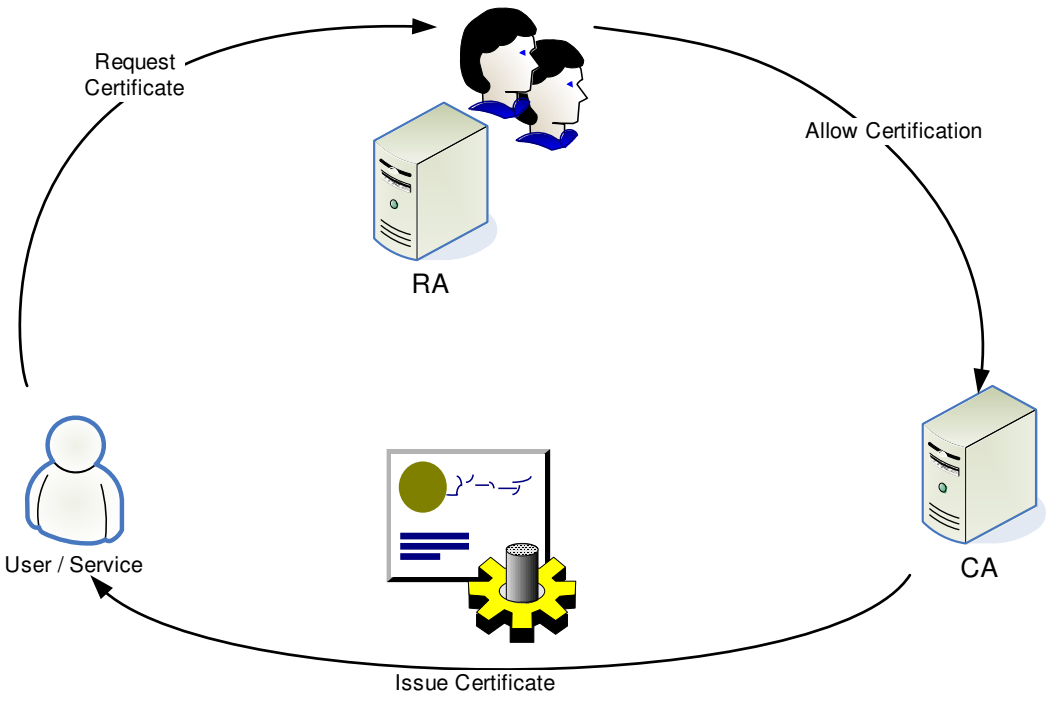

Figure 25: Certificate Issuing

For the fist issue it is considered to provide a root CA that is certified by a worldwide accepted CA (or self signed when each member has trust in this Authority without a certificate from a trusted 3rd party). This Akogrimo-Root-CA will handle certificate requests for entities and especially for other CAs that are intended to be used in Akogrimo. E.g. when each domain owner wants to use his own CA, these CAs have to be certified by the Akogrimo-Root-CA, so that all entities from other domains have the ability to trust the certificates of that CA too. In that way the trust to each CA used in Akogrimo is ensured by inheritance:

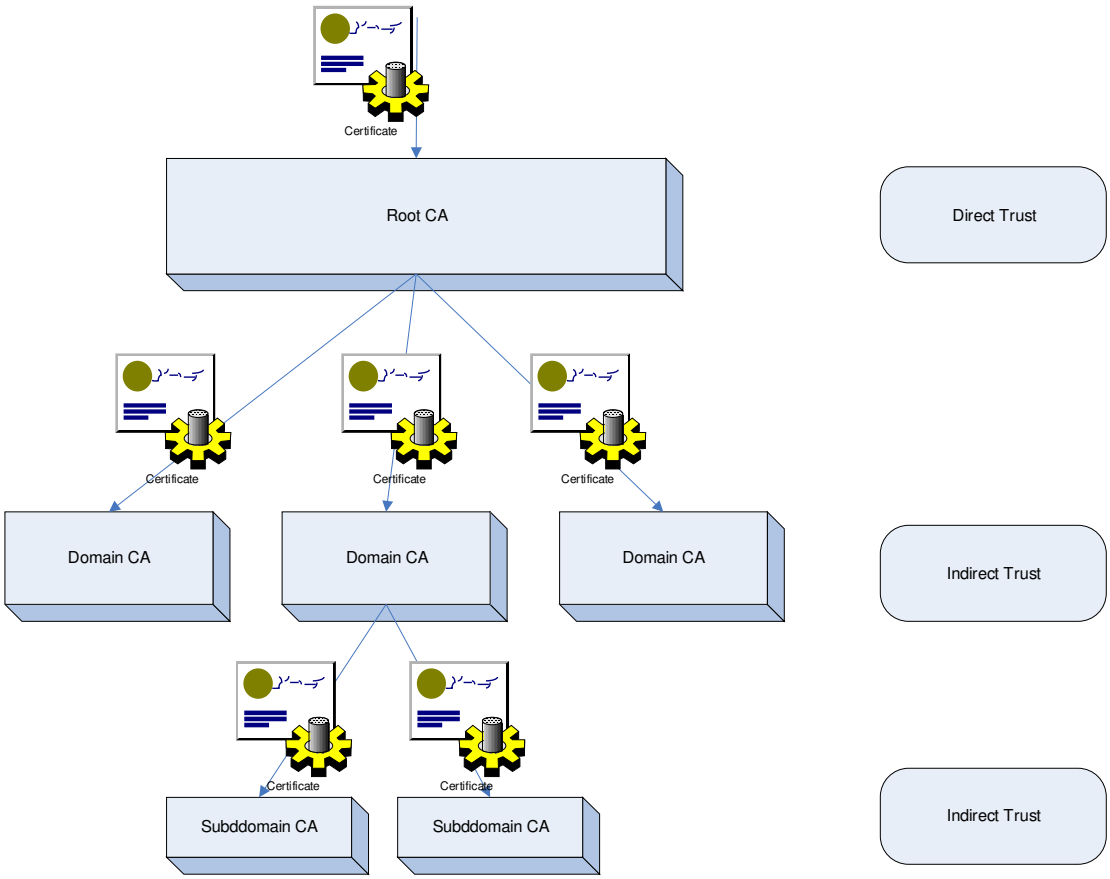

Figure 26: CA Hierarchy

If it is not feasible to establish a Root-CA that is trusted by each member, all CAs in different domains (or in a larger scale if possible) must have certificates from other trusted CAs.

A revocation list is used to verify the status of certificates. When a user reports misuse (or it is detected by other sources) the certificate is revoked and this revocation is issued. Due to constant updates of the revocation list all Akogrimo members are aware that this certificate is not valid any longer.

It is thought that the user certificate is stored on a commercial device for that purpose (USB-Stick, Smartcard, etc.) to fulfil the requirements regarding user's mobility. In this case it shall be possible to use the same certificate on more than one device simultaneously to offer the user the possibility to use more than one device at the same time. The certificates of non-mobile components are stored at their machines. Since it is possible for each user to have different identities in different VOs, he must have possibly more than one certificate.

The sender presents his certificate to the receiver when he wants to communicate using key encryption. The receiver in turn sends back his certificate to the sender to guarantee that each partner holds the information that is needed for secure communication.

To ensure that a certificate is not revoked, each partner can contact the CA (or every other entity which is administrating the revocation list) to check the transmitted certificate.

3.10.2. Usage

3.10.2.1. Certificate Issuing (Offline Process)

Figure 27 shows the standard way of certificate issuing. The requester generates a key pair, consisting of a private and a public key. He sends a certificate request to the responsible Registration Authority from where the request is forwarded to the Certificate Authority after a successful credential check. The Certificate Authority generates and signs then the certificate and informs the requester how to obtain it

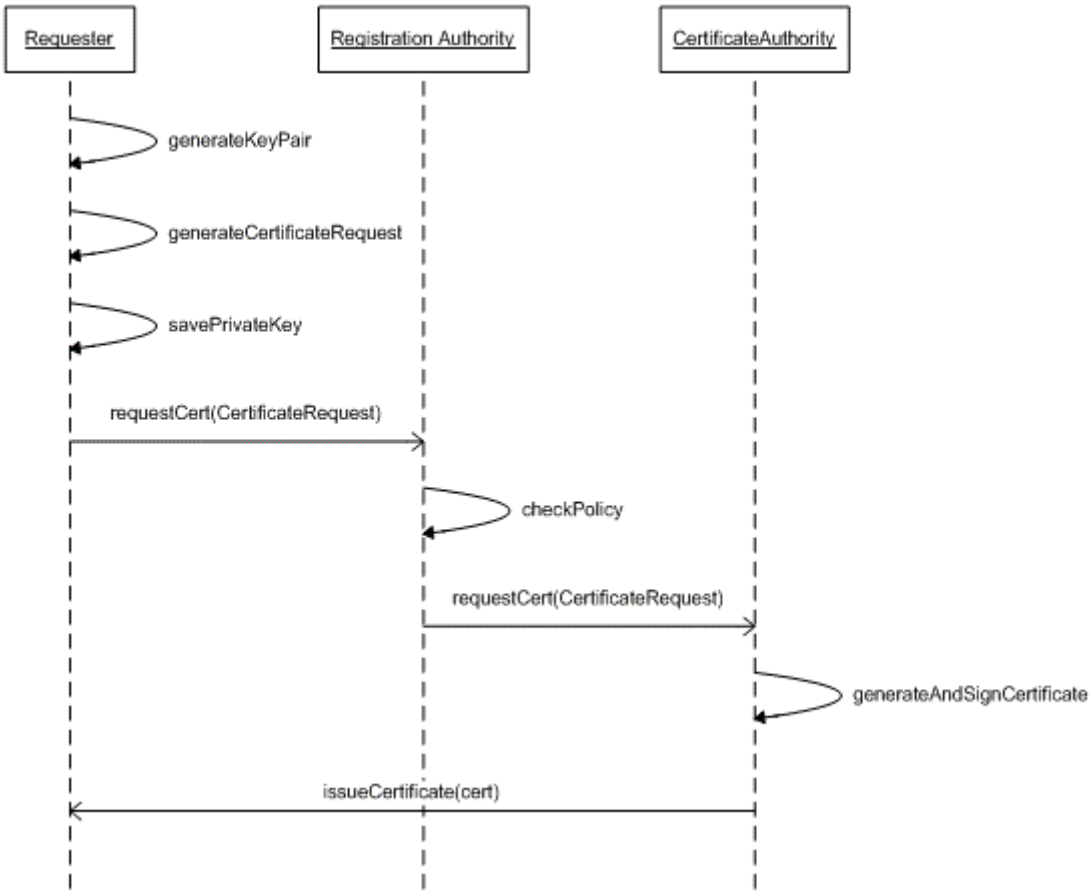

Figure 27: Certificate Issuing

3.10.2.2. Certificate Use

Figure 28 shows the two ways in which a certificate is usually used. When a sender wants to sign a message, he sends his certificate to the receiver that claims that he is the owner of the private key that corresponds to the public key in the certificate. The receiver can be sure that the message was signed by the entity that claims to have done it after obtaining the message and checking the validity of the certificate.

When a receiver wants to obtain an encrypted message from a specific sender, he sends his certificate to him. The sender is now able – after checking the validity of the certificate – to encrypt the message with the public key of the receiver and can be sure that only the wanted receiver is able to decrypt the message.

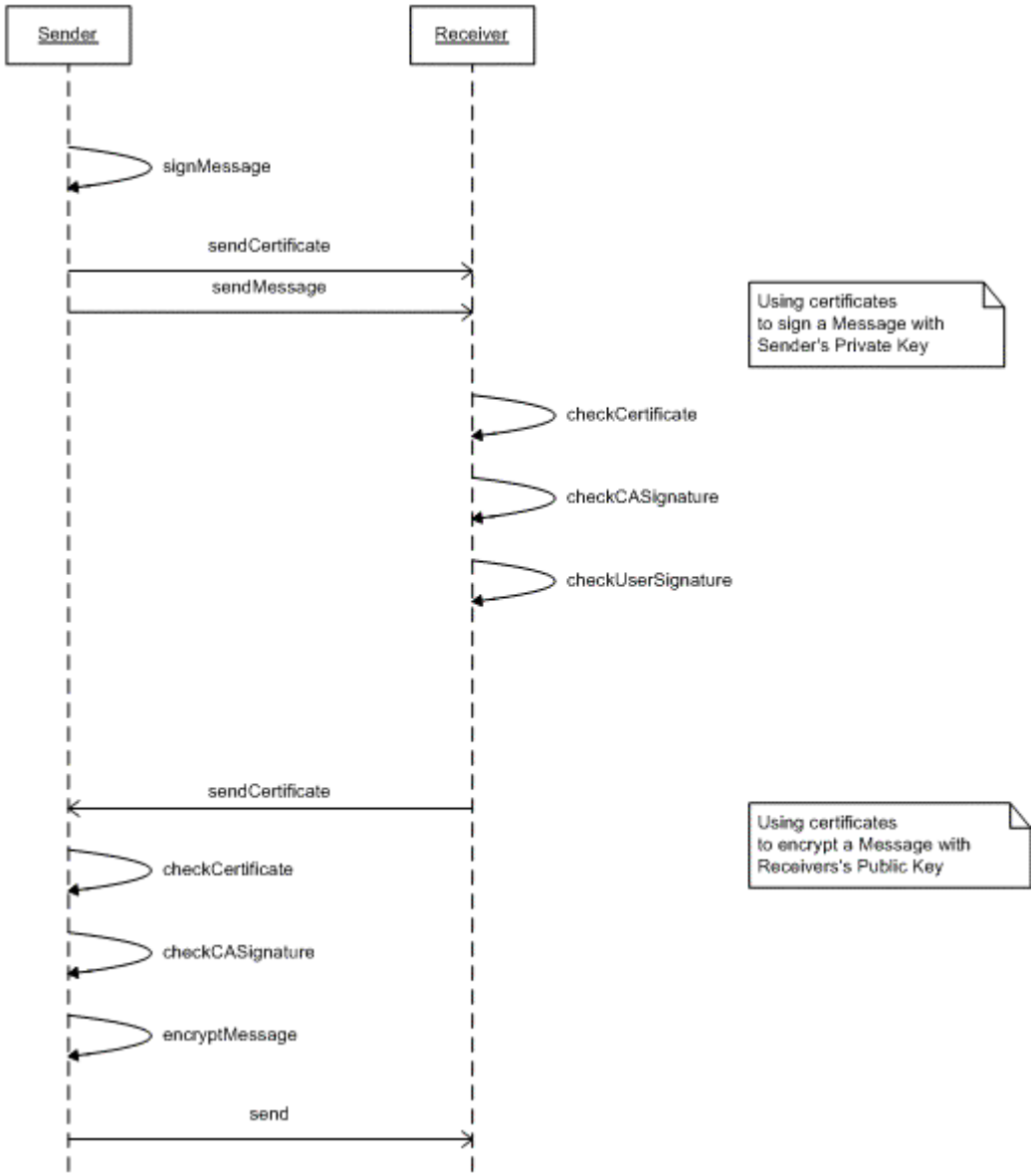

Figure 28: Certificate use

3.10.3. Involved Components

3.10.3.1. Certificate issuing: CA and RA

Akogrimo does not need a full equipped CA, because most of the CA-jobs are done offline and depend on the way the provider implements it. To guarantee some basic functionality like certificate issuing a CA will be build up. An LDAP-Server contains the Revocation-List and provides the opportunity to update the cached revocation lists in each component. (In another way we can offer the chance to validate a certificate each time it is sent by contacting the LDAP-Server)

Since all tasks related to the RA are done by offline processes and the implementation of RAs is done by each provider in a maybe different way, there is no need to implement a RA in Akogrimo.

3.11. Authorization and Centralized Policy Management - Security Policy Case

3.11.1. Inter-domain relationships

An OpVO, by its very nature, involves services that belong in multiple domains, the BVO for its core services, various SP's and NP's domains for its application services. The OpVO therefore need to grant membership to entities in different security domains. To do that, it needs to be able to authenticate itself to the other domains and in turn be able to authenticate them. This is achieved by making the BVO a Trusted Third Party. Service and Network Providers must a priori register with the BVO, and will be given BVO membership tokens. The OpVO is created by the BVO, and at creation is given a BVO membership token. This token is then used to bring new services into the OpVO – the OpVO contacts an Instantiator service (provided by all SPs), requesting a service instance be allocated to it and supplying the OpVO token to be passed to the new service instance.

Of course, the distribution of the BVO tokens must also be protected. As for OpVOs, this could be done using an external (to Akogrimo) TTP. However, since OpVOs are relying on, and paying for, SPs to meet their obligations to provide the services, an explicit contract should be accepted/signed before a new SP is allowed to join the BVO. Since this essentially off-line, the token distribution can be incorporated into this process. At this point, for Service discovery purposes, the SP can provide details about the services they offer and any policies that they wish to apply to the use of these services. This can allow Service Discovery to take SP policies into account. The VO keeps a registry of the member name, public key and location.

3.11.2. Authorization

Every action in Akogrimo (other than those internal to Service or Network providers) happens within the context of a VO. Therefore the owners of the VO, sender and receiver all have an interest in the action being carried out, and may want to constrain that action according to a pre-defined policy. When a message is received, the receiver must check that the action requested does not contravene anything in the VO or the receiver's policies. It does this by implementing a PEP, which extracts from the message the action (and possibly input data) and asks an Authorization Service (PDP) for permission to proceed. It will also have to provide the identities of the requestor, the VO and the target resource (and possibly other context information, if the policies are likely to need that information to make the decision, e.g. CPU load for a "best endeavours" policy). The response will be either "allow" or "deny", but there may also be obligations associated with this permission. Obligations will probably be requests to update or modify the target or context before or after the request has been executed (e.g. record the time or log the results). The PEP can be implemented as part of the service, or as part of the message delivery chain. The advantage of the latter is that it will be easier to protect as it is completely independent from the resources/ services being accessed. However, this may cause problems if the target's private key is required to decrypt the message in order to extract request data for the authorization.

The PDP must be part of the same domain as the PEP, since there must be complete trust between the two. The PEP must be able to authenticate the PDP (PEP authentication is less important, as the only consequence of a spoof PEP attack would be to release details of the policy. The PDP needs access to the relevant policies in order to evaluate the requests received from the PEP. Local policies, i.e. the Service Provider's policies, can be retrieved directly from a filestore or database. Retrieving the policy for the VO requires accessing the appropriate Policy Manager. For OpVOs, is the BaseVO PolicyManager, in which the owners of the =pVOs can

store the relevant policies. Access to this store could be restricted to OpVO members (i.e., those with a valid OpVO token, but this is probably not necessary. Requiring a BaseVO token should ensure only BaseVO members, i.e. registered Service and Network Providers, have access.

3.11.3. VO operation

On creation, the first action an OpVO has to do is to bring into the OpVO the necessary core members services. The OpVO contacts the relevant factory (or equivalent) within the BaseVO, providing its BaseVO membership token to demonstrate it is authorized to obtain an instance of the service. The factory creates an instance and is given its public key. The factory then provides the new instance with a BaseVO membership token. The public key of the instance is returned to the OpVO, which creates an OpVO membership token and passes it to the instance. The instance thereby becomes a member of the OpVO. A similar mechanism is used to bring SP services into the OpVO, but interacting with the factory or equivalent in the SP domain. This factory will not be a member of the BaseVO (it will be a member of the SP domain), so it need a mechanism to authenticate the requestor and authorize the request. The SP must provide such a mechanism, using the BaseVO token(s) it received when the SP registered with the Base VO.

Messages within a VO can use WS-Security and WS-SecureConversation mechanisms to protect the communication end-to-end. Messages must include the senders VO token, and should be signed to ensure integrity. Confidentiality can be ensured by encryption. For one-off and small messages, the sender encrypts the message using the receiver's public key, obtainable from the VO. For multiple or large messages, efficiency can be improved by exchanging a symmetric encryption key – the key is signed by the sender, encrypted using the receiver's public key and sent to the receiver.

On receipt of a message, the receiver must ensure it contains a valid VO token and is properly signed by the sender. The next step is to ensure the request is properly authorized. The message body will have to be decrypted using the OpVO membership token (or the target's private key).

This mechanism also applies to interactions with the user's Mobile Terminal (MT). During the logon sequence, the MT is issued with an OpVO membership token. If the MT participates in multiple OpVOs, the MT will have to ensure that responses are returned to the correct OpVO by including the membership token (and encrypting the message?). This applies even to cases where the MT is to map a SIP call to another MT – the receiving MT should ensure that the SIP call initiation message has an OpVO token, and let the user know that this is an authorized call (or at least distinguish it from an unauthorized call).

The above gives reasonable protection against most threats to the BVO and OpVO operations, apart from denial-of-service attacks. The main (minor) vulnerability is during the token distribution to new members – requires trust in the factory and the factory environment. One area rather outside our control is the security of the SP domains – we can provide them with the necessary tokens, but cannot ensure they are used correctly. To protect against rogue SPs, intrusion detection is required (i.e. monitoring the messages being exchanged and ensuring only those expected by the workflow are happening – any non-expected messages should result in the sender being thrown out of the OpVO.

3.12. Context Management Information Handling in Security Policies

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 pas 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.

3.13. End to End Security

3.13.1. Attacks and Thread Model

This chapter shows some threat and attacks examples that should be taken into account.

3.13.1.1. Access Router

The AR is the first line of defence of the Akogrimo network. Its role is to prevent unauthorised users from accessing the Akogrimo network, and to forward valid packets from authorised to their destination.

In this section we make a short explanation of how the AR works internally, and then proceed to analyse what attacks can be made.

General characteristics of the AR are:

- It only accepts packets from authenticated users. All other packets are dropped.
- When a packet from a new flow arrives, it checks with the QoS Broker if it's allowed
- When a packet from an already known flow arrives, filters are already in-place and the packet is dealt with without further requests to the QoS Broker.

Attacks are divided in two types, external and internal. External refers to the AR's interface to the access network and internal refers to the interface to the core network.

3.13.1.2. External attacks

- DoS (Denial of Service)
- There is no getting around these kind of attacks
- If the terminal is only allowed to use "well-behaved" Akogrimo software, then DoS attacks are not possible
- Unauthenticated users get their packets dropped. However, a big number of simultaneous attackers can be effective at DoS'ing the AR.
- Various possibilities
- Low level radio signal jamming
- Unauthenticated user flooding the wireless network with bogus packets
- Unauthenticated user sending the AR valid packets that must be analysed
- These types of attack are out of scope for the Akogrimo project ?
- Impersonation
- The AR works in layer 3. An attacker can spoof packets with a valid user's ip address. Authentication+authorisation+token should stop this from happening. This probably has more to do with the MT than the AR.

3.13.1.3. Internal Attacks

- Malicious QoS Broker sends wrong rules, for benefiting or hurting a user(s)
- Malicious A4C server can send fake user information
- Malicious QoS Broker can send fake rules; can benefit a certain user, giving him QoS level which he would not get normally
- Any component can cause DoS attacks

3.13.2. QoS Broker

The QoS Broker receives requests from the Access Routers and the EMS Web Service. With user profile information it gets from the A4C Server, and also information about the current network status, it decides if a user is allowed a network flow at a given QoS level.

3.13.2.1. External attacks

- Authenticated user(s) causing repeated requests to the QoS Broker (DoS)
- Since the AR gets rules installed when a new flow is requested, this will only cause problems if the user starts a large number of different flows. With typical Akogrimo software this is not possible, so the user's MT would have to have been compromised and special software installed on it.
- Several users can cooperate, maximizing the attack

3.13.2.2. Internal attacks

- Malicious A4C server can send fake user profile
- Malicious AR can
- use data collected from innocent users to benefit or harm a certain user
- Malicious EMS can also benefit a certain user; can cause repeated QoS requests

Any component can cause DoS attacks

3.13.3. SIP Infrastructure

This section describes possible attacks against Akogrimo SIP infrastructures, i.e. SIP Broker, SIP Server, AN SIP proxies, and SIP MTs.

3.13.3.1. SIP Registration Hijacking

The SIP registration mechanism allows a user agent to identify itself to a registrar as a device at which a user is located, in Akogrimo project the SIP address are as *user@akogrimo.org*. A registrar assesses the identity asserted in the "From" header field of a REGISTER message to determine whether this request can modify the contact addresses associated with the address-of-record in the "To" header field. While these two fields are frequently the same, there are many valid deployments in which a third-party may register contacts on a user's behalf.

The "From" header field of a SIP request, however, can be modified arbitrarily by the owner of a UA, and this opens the door to malicious registrations. An attacker that successfully impersonates a party authorized to change contacts associated with an address-of-record could, for example, de-register all existing contacts for a URI and then register their own device as the appropriate contact address, thereby directing all requests for the affected user to the attacker's device.

This threat type belongs to a family of threats that rely on the absence of cryptographic assurance of a request's originator. Any SIP UAS that represents a valuable service, for instance the SIP Broker, might want to control the access to its resources by authenticating requests that it receives (more detailed [1]). Even an end-user (MT) is interested in ascertaining the identities of originator of request (for more details see [1])

This threat demonstrates the need for security services that enable SIP entities to authenticate the originators of requests.

3.13.3.2. Impersonating a SIP Server

The domain to which a request is destined is generally specified in the Request-URI. UAs commonly contact a server in this domain directly in order to deliver a request. However, there is always a possibility that an attacker could impersonate the remote server, and that the UA's request could be intercepted by some other party.
In Akogrimo project this attack type is especially important. For instance, a malicious entity could impersonate the SIP Server, allowing it to obtain confidential data or to perform DoS attacks. In a similar way, the impersonation of the SIP Broker could allow malicious interactions with the GRID components which could be the basis of several attacks.

3.13.3.3. Tampering with SIP Message Bodies

Consider a UA that is using SIP message bodies to communicate session encryption keys for a media session. Although it trusts the proxy server of the domain it is contacting to deliver signalling properly, it may not want the administrators of that domain to be capable of decrypting any subsequent media session. Worse yet, if the proxy server were actively malicious, it could modify the session key, either acting as a Man-in-the-Middle, or perhaps changing the security characteristics requested by the originating UA.

This family of threats applies not only to session keys, but to most conceivable forms of content carried end-to-end in SIP. These might include MIME bodies that should be rendered to the user, SDP, or encapsulated telephony signals, among others. Attackers might attempt to modify SDP bodies, for example, in order to point RTP media streams to a wiretapping device in order to eavesdrop on subsequent voice communications.
For these reasons, the UA might want to secure SIP message bodies, and in some limited cases header fields, end-to-end. However, since many header fields are legitimately inspected or altered by proxy servers as a request is routed, not all header fields could be secured end-to-end.

The security services required for bodies include confidentiality, integrity, and authentication. These end-to-end services should be independent of the means used to secure interactions with intermediaries such as proxy servers.

3.13.3.4. Tearing Down SIP Sessions

Once a session has been established by initial messaging, subsequent requests can be sent that modify the state of the dialog and/or session. It is critical that principal participants in a session can be certain that such requests are not forged by attackers.

Consider a case in which a third-party attacker captures some initial messages in a dialog shared by two parties in order to learn the parameters of the session ("To" header, "From" header, and so on) and then inserts a BYE request into the session. The attacker could opt to forge the request such that it seemed to come from either participant. Once the BYE is received by its target, the session will be torn down prematurely.

Similar mid-session threats include the transmission of forged re-INVITEs that alter the session

The most effective countermeasure to this threat is the authentication of the sender of the BYE. In this instance, the recipient needs only know that the BYE came from the same party with whom the corresponding dialog was established. Also, if the attacker is unable to learn the parameters of the session due to confidentiality, it would not be possible to forge the BYE. However, some intermediaries (like proxy servers) will need to inspect those parameters as the session is established.

3.13.3.5. Denial of Service and Amplification

Denial-of-service attacks focus on rendering a particular network element unavailable, usually by directing an excessive amount of network traffic at its interfaces. A distributed denial-of-service attack allows one network user to cause multiple network hosts to flood a target host with a large amount of network traffic.

SIP creates a number of potential opportunities for distributed denial-of-service attacks that must be recognized and addressed by the implementers of SIP systems.

Attackers can create bogus requests that contain a falsified source IP address and a corresponding "Via" header field that identify a targeted host as the originator of the request and then send this request to a large number of SIP network elements, thereby using hapless SIP UAs or proxies to generate denial-of-service traffic aimed at the target.

Similarly, attackers might use falsified "Route" header field values in a request that identify the target host and then send such messages to forking proxies that will amplify messaging sent to the target.
"Record-Route" could be used to similar effect when the attacker is certain that the SIP dialog initiated by the request will result in numerous transactions originating in the backwards direction.

A number of denial-of-service attacks open up if REGISTER requests are not properly authenticated and authorized by registrars.
Attackers could de-register some or all users in an administrative domain, thereby preventing these users from being invited to new sessions. An attacker could also register a large number of contacts designating the same host for a given address-of-record in order to use the registrar and any associated proxy servers as amplifiers in a denial-of-service attack. Attackers might also attempt to deplete available memory and disk resources of a registrar by registering huge numbers of bindings.

The use of multicast to transmit SIP requests can greatly increase the potential for denial-of-service attacks.

These problems demonstrate a general need to define architectures that minimize the risks of denial-of-service, and the need to be mindful in recommendations for security mechanisms of this class of attacks.

3.13.4. The SIP/SOAP Security Problem

The SIP with SOAP interaction in Akogrimo takes place in the component called "SIP Broker", which implements the conceptual joint between the GRID and Network layers, interacting with the needed GRID Entities (the EMS in most of the cases).

Additional functionality is also placed in the SIP Broker. To be precise, in the prototype version two functionalities are offered:

- Grid-initiated SIP Session, which enables the possibility to arrange a GRID-triggered session between two SIP users.
- Grid-initiated SIP Transfer, which enables the possibility to inform a user he/she can move an ongoing session to a most suitable device.

Other functionalities (e. g. a WS interface to expose the SIP SD facilities) are being considered to be included in next versions. So we can consider that the SIP with SOAP approach in Akogrimo mainly relays on the capabilities that this component offers.

The internal structure of the SIP Broker consists mainly on three differentiated main submodules:

- The **SOAP interface**, which accepts incoming WS requests from the "grid world".
- The **Broker Engine**. This component is the core of the SIP Broker. Its main task is to convert the WS request parameters (typically the Akogrimo IDs of the user to be put in contact, or the device URI the user is allowed to move to) into a suitable format that the SIP UA can understand. This is made through an A4CServer's user profile request (using an embedded A4C client), that returns – apart from other information - the SIP AoR of the user. For efficiency purposes, and considering that only one SIP UA and one A4C client are needed, it has been implemented as a separate Java process that can be accessed through Java RMI.
- The **SIP UA**, which handles the SIP process itself. It has been implemented as a Java library that other components can use.

In order to avoid that some non-authorised entity can make use of the facilities that the SIP Broker can provide, security mechanisms must be implemented to protect each possible access path to this component.

Apart from attacks, another crucial consideration must be taken into account. The currently implemented interactions take place within the context of a certain VO, i.e. some entity decides that two members of an OpVO should have a conversation, or one of the members should be requested to transfer an ongoing session it has with another OpVO component to a most suitable device. The distributed perimeter security model adopted by the OpVO means that interactions between entities can only occur in the context of a specific OpVO instance. This issue is considered in section 2.2.

3.13.4.1. SIP Broker threat model

Figure 29 show both the authorised way to use the SIP Broker and the potential attacks we have identified.

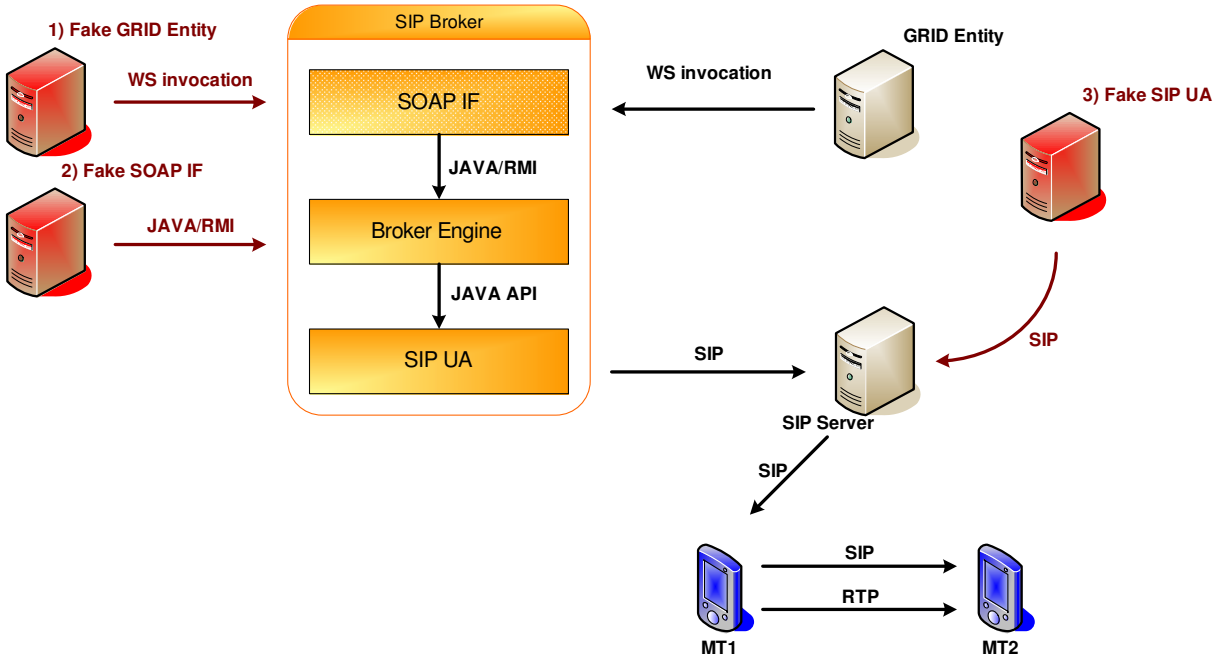

Figure 29: "SIP with SOAP" possible attacks

During a normal (authorised) operation, GRID Entities will request for a SIP process using the interface that the SOAP interface subcomponent provides; then it invokes the corresponding method of the Broker Engine using Java RMI, which will invoke the Java API provided by the SIP UA to start the corresponding SIP mechanism (obtain details about a concrete resource or get its status). These interactions have been depicted with black arrows in the figure.

We have identified three types of potential attacks, depending on the faked entity. Non-authorised element and interactions have been depicted in red:

If the SOAP interface do not check the identity of the invoking element (i. e. the EMS or the Workflow Manager), it could be possible that a non-authorised element makes use of the exposed WS interface to trigger bogus SIP interactions. So in order to avoid this kind of attacks, this WS interface should implement authorisation mechanisms. This is a pure WS interface security problem.

The broker engine is exposed as a RMI daemon, so it could be invoked from a remote machine. So if the identity of the calling entity is not guaranteed, some entity which knows the RMI URL could fake the identity of the SOAP interface and make non-authorised requests; to solve this, the JAVA RMI interface must guarantee that only the real SOAP interface is allowed to use the broker engine. This is a pure Java RMI security problem.

Finally, some malicious SIP entity could pose as the whole SIP Broker (to be precise, posing as the SIP UA) to send false SIP requests to end user's MTs. If neither the SIP Server nor the end user terminal checks for the request authenticity, this non-authorised entity could send a request to a MT to initiate a session or to perform a transfer. Anyway, this is a pure SIP security problem.

So all possible attacks in which some entity poses as the SIP Broker (of any of its components) can be reduced to a technology-related security problem (security with WS, RMI or SIP interactions). Chapter 3.15.3 describes these different security mechanisms.

3.13.4.2. OpVO's security model in Akogrimo

Essentially, any user/service participating in an OpVO will be given an OpVO token to prove their membership of that OpVO instance. Distribution of these tokens will be secured using the

user/service's identity token and the BaseVO token. Subsequently, the user/service should only accept instructions/data from an entity that also possesses that token. This token can be used to provide end-to-end security for the communications.

The VO security mechanism operates as follows: all participants in an OpVO are given membership tokens signed by the OpVO and containing their identity. Each participant only responds to messages that contain this token. The message content can be protected using the token and message sequencing mechanisms. Thus the OpVO members communications within this OpVO are secured against external attach. This basic mechanism can be extended to provide security against insider attacks by enforcing role-based authorisation, or even checking with an authorisation engine.

For example, we create an eHealth OpVO, O1, with doctor D, patient P, and two services S1 and S2. Now all of these are provided with a membership token for O1, containing their name. O1 (or rather its workflow instance) will instruct S1 to invoke S2. S1 checks the message came from O1 and has an O1 membership token (if not, the message is ignored). It then invokes S2, passing its own membership token. S2 checks the message is from S1 and that S1 has a membership token. It then accepts the invocation.

In the context of the SIP with SOAP interactions being considered in Akogrimo, it would be beneficial if the Mobile Terminals could process the tokens, signatures, encryption, message sequencing, etc. If it can't, then the SIP Broker will have to take over the responsibility of operating the "distributed perimeter". On receiving an instruction to a Mobile Terminal, it will have to check that the message has been authorized before acting on it, i. e. that the Mobile Terminal belong to the same OpVO as the sender. It will have to access the Participant's Registry to check what OpVOs the MT belongs to, so it needs to be a member of the same BaseVO as the OpVO. It may also have to check with the OpVO that this particular interaction is authorized (the insider attach protection mentioned above)

In the example, if the services were doctor and patient, a session between them should only be possible if the doctor is treating the patient, i. e. in the context of an OpVO. When the patient receives the request for a new session, he/she can be sure that the caller is his/her own doctor, because the call is known to belong to the OpVO.

The inclusion of this mechanism in the SIP with SOAP security model is described in section 3.15.3.

3.14. Business Flow Security Methodology

Within a Mobile Dynamic Virtual Organization (MDVO), various business flows take place. Flows can take place either within an administrative domain (intra-domain) or between domains (inter-domain). For both possibilities, relevant security issues have to be determined and the respective measures for compliance with the desired level of security have to be defined and implemented.

3.14.1. Business Flow Types

Irrespective of the specific environment of an MDVO, various business flow types are identified. The most important business flow types consist of the following:

- **Workflows**: Workflows are perceived as the operationalized steps of a business process. Worflows document the structured way in that tasks are executed, thus indicating who is responsible for processing a piece of work within a given time-frame, and how different tasks inter-relate. Workflows can be modelled by means of Petri-nets or workflow diagrams.

- **Information Flows**: Information flows depict the way that information pieces take through an organization. In other words, information flows are a supporting process for workflows so that they document which information out-flow is generated by a workflow entity based on a certain in-flow. Information flows are best modelled in the form of message sequence diagrams.
- **Financial Flows**: Financial flows are tightly coupled with product and service flows, since financial compensations form the natural counterpart for commercially offered (electronic) products and services. These compensations can be both monetary, i.e. they are charged in terms of an agreed currency, as well as non-monetary. Financial flows reflect sources of revenue for providers, whereas they reflect sourcing cost elements for consumers, so that financial flows become an important tool for financial planning and thus also for business modelling. However, there is no standardized modelling tool available.

With regard to security, those presented business flows outline the respective to be secured steps of processing (workflows) and exchanging (information flows) information in and between administrative domains (intra- oder inter-domain), as well as the compensation (financial flows) for consumed electronic products and services, whereas inter-domain financial flows are of much higher relevance from a security point of view than intra-domain financial flows.

3.14.2. Business Flows in an MDVO

MDVOs are composed by legally independent organizations. The single organizational entities, individuals or organizations of larger extent, form administrative domains, each disposing of resources and services. Resources (infrastructure, knowledge, human labor etc.) are offered to the MDVO through well-defined interfaces, defining services which in turn encapsulate the economic potential of available resources in terms of value adding workflow steps, requiring input information in the service request and providing a result in a pre-determined form.

Besides multi-domain service provisioning issues, resulting in a distributed value chain, MDVOs are characterized by a high level of dynamics with respect to organizational composition and to adaptiveness of workflows, mainly due to support of a variety of mobility aspects (device, user, session mobility), which also implies considering device and user context information. All aspects, inter-domain service provisioning, mobility support, and dynamics lead increase complexity with regard to security:

- **Trust**: In an MDVO, trust-building mechanisms are not trivial to be implemented. Service provisioning over a distributed value chain on one hand allows the various VO members to focus on their respective core competencies, on the other hand the overall organization consists of separately administered domains, in which potentially different practices and security standards apply. Furthermore, sensitive data may be exchanged electronically among domains. Otherwise expressed, the span of control in an MDVO is shorter than in non-virtual organizations, and security verification as well as audits become more complicated. If the organizational alignment of an MDVO changes dynamically, these concerns are even amplified.
- **Mobility**: If devices and users are mobile, assumptions with respect to environmental influences are transient. This does not automatically imply that security is tampered, but it makes threat and risk assessment more difficult than in a static environment. For instance, a wider range of contact points to non-members of the MDVO has to be assumed, whereas outsiders are regarded to be untrusted. This leads to the conclusion that the risk for abuse by non-authorized users is higher than in a well-known, static scenario. From a technical viewpoint, shared wireless media might be even more vulnerable to exploits than wired access technologies.

- **Role Model**: Different role holders in an MDVO hold different information pieces, so that e.g. a grid application service provider by aggregating basic services integrates more business logic than other VO entities such as a single grid service provider. This fact puts a different weight on the various roles in the overall service delivery. Some actors, implementing one or more critical business roles for instance with respect to financial clearing may form a highly prestigious target for attacks. Therefore, MDVO role models have to be analysed with regard to a differentiation in security levels to be achieved, finally providing answers to what interests and contents have to be protected from what threats and by what measures. Service provisioning in a distributed value chain on one hand might have negative implications on trust-building (cf. Trust paragraph), on the other hand distribution of security-relevant information across administrative domains potentially shows positive effects on security concerns for a specific role holder that has been compromised. In addition, the use of virtual, time-wise limited public identities as a means for federated identity management represents a tool with a similar effect than information distribution across domains.

3.14.3. MDVO business flows in practice

In the following, the concepts introduced in the previous sections will be applied in practice describing the approach that it will be followed in Akogrimo to manage the business flows in the Akogrimo multi-domain environment.

The following sections will provide information about definition of a possible approach to authenticate and authorize Akogrimo users through the different domains

In Figure 30 some basilar assumptions are defined. Their reading is a pre-requirement to understand the following sections.

The NLD domain is the underlying domain, in order to do any thing you have to pass through a NLD, this means that before accessing a SLD you have to pass through a NLD.

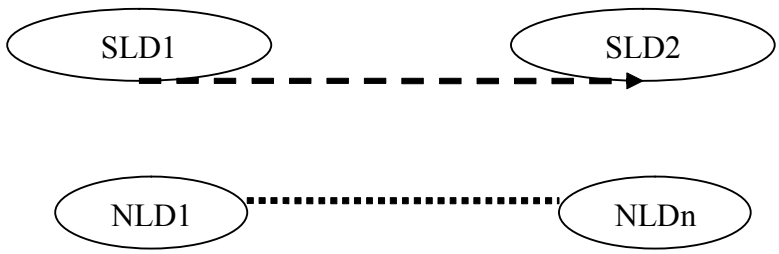

Figure 30: Akogrimo basilar business flow between domains

We could summarize that in order to pass from SLD1 to SLD2, we have to pass at least through one NLD.

In this section, some basilar Akogrimo business scenario will be described.

A member of the BVO in order to act in the Akogrimo needs to pass through several logic domains. This means that we should have several authentication and authorization passing through them. We should have a unified approach to face this challenge reducing as much as possible the need for presenting every time own credentials (Single Sign On).

Typical Akogrimo scenarios are:

- A customer asks for a service of the BVO (e.g. creation of an OpVO). See Figure 31.

A user tries to use an OpVO (see Figure 31).

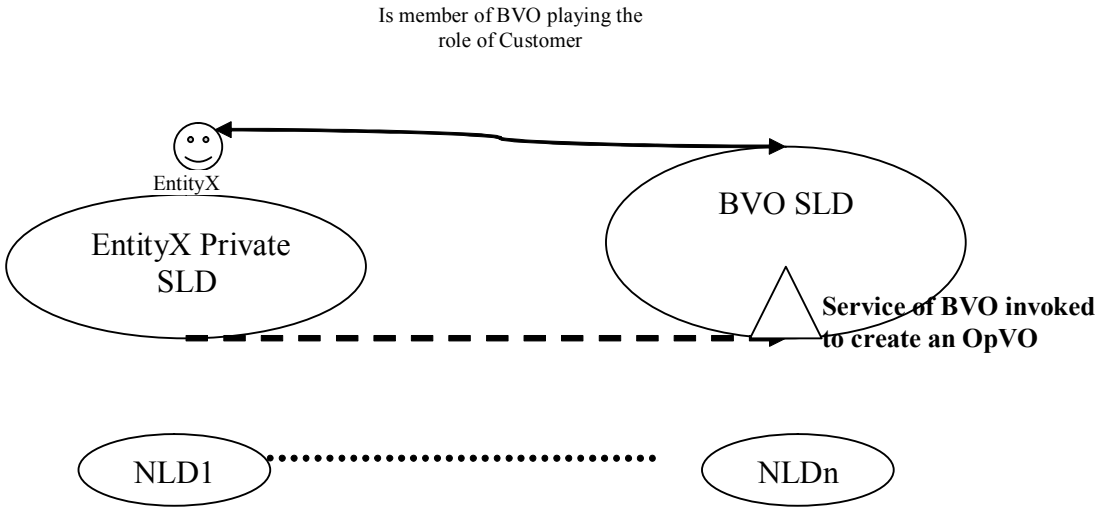

Figure 31: A customer asks for creating an OpVO

In Figure 32 it is depicted a typical Akogrimo scenario where we have the stated problem. EntityX (member of BVO as Customer) asks for creating an OpVO. This means that an invocation starts from the EntityX private SLD. It has to use one (or more) NLD but this can not be identified in advance (the customer could be mobile/nomadic). The request arrives into the BVO SLD.

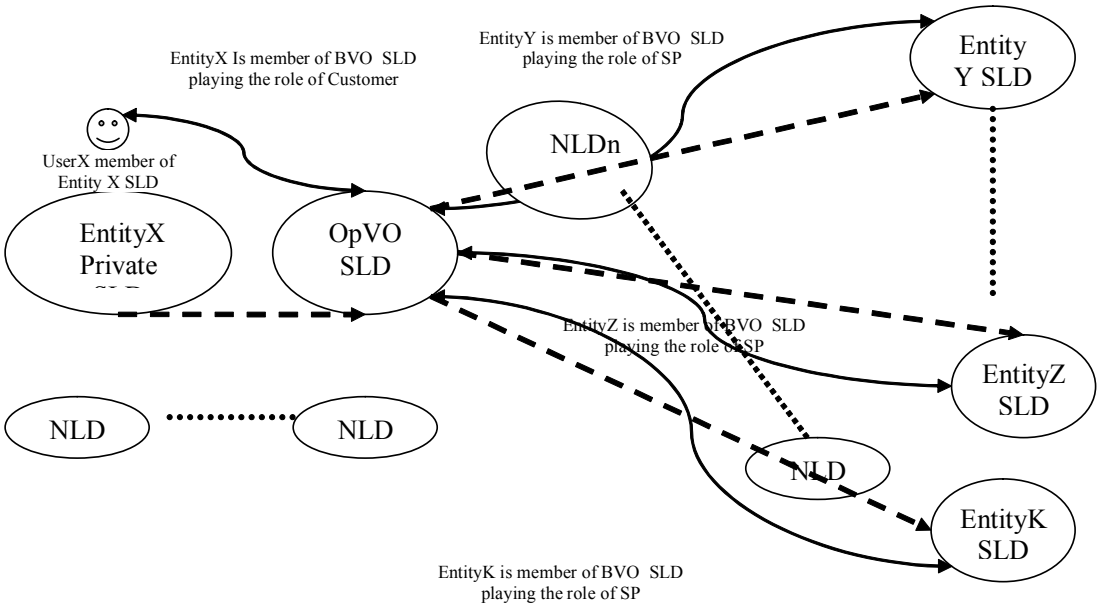

Figure 32: A user uses an OpVO

it is clear as the general problem stated at the beginning of this section becomes quite complex in this situation.

In this case, there is user of an OpVO that is member of an EntityX logic domain. This user has to access a service of the OpVO (e.g. the ehealth monitoring service) then an invocation starts from the EntityX private SLD. The user will pass through one (or more) NLD but this can not be identified in advance (the user could be mobile/nomadic). The request arrives into the OpVO

SLD. Now from this last SLD several invocations have to start in order to use services in the SLDs of the SPs involved in the OpVO execution. Of course also these last invocations must pass through several NLDs that cannot be known in advance because the invoked services could be on mobile devices.

In order to address the stated problem, solutions should be found to make aware the private domains of the entities involved as members of the BVO/OpVO that they could be contacted by members of external domain. The following sections propose some possible approaches, in particular, they focus on the setup of members' domains that should be performed during the subscription of a member to the BVO

3.14.3.1. Approach

This section enters into the detail describing a possible approach to manage the issues previously introduced.

The following use cases have been taken into account:

- BVO subscription
- OpVO creation
- OpVO use

For each use case, it is possible to assume several approaches. Here, a network leaded approach has been assumed, but for each use case an alternative approach is described (see Annex 2)

3.14.3.1.1. A member subscribes to the BVO

In this case, the NP is assumed to have a basilar role in the subscription process. This description envisages a subscription phase that is mainly managed off-line. Of course, this simplifies the overall process.

The basilar assumption is that the NP is the first entity to become member of the BVO: this means if we don't have NPs we cannot start to create a BVO.

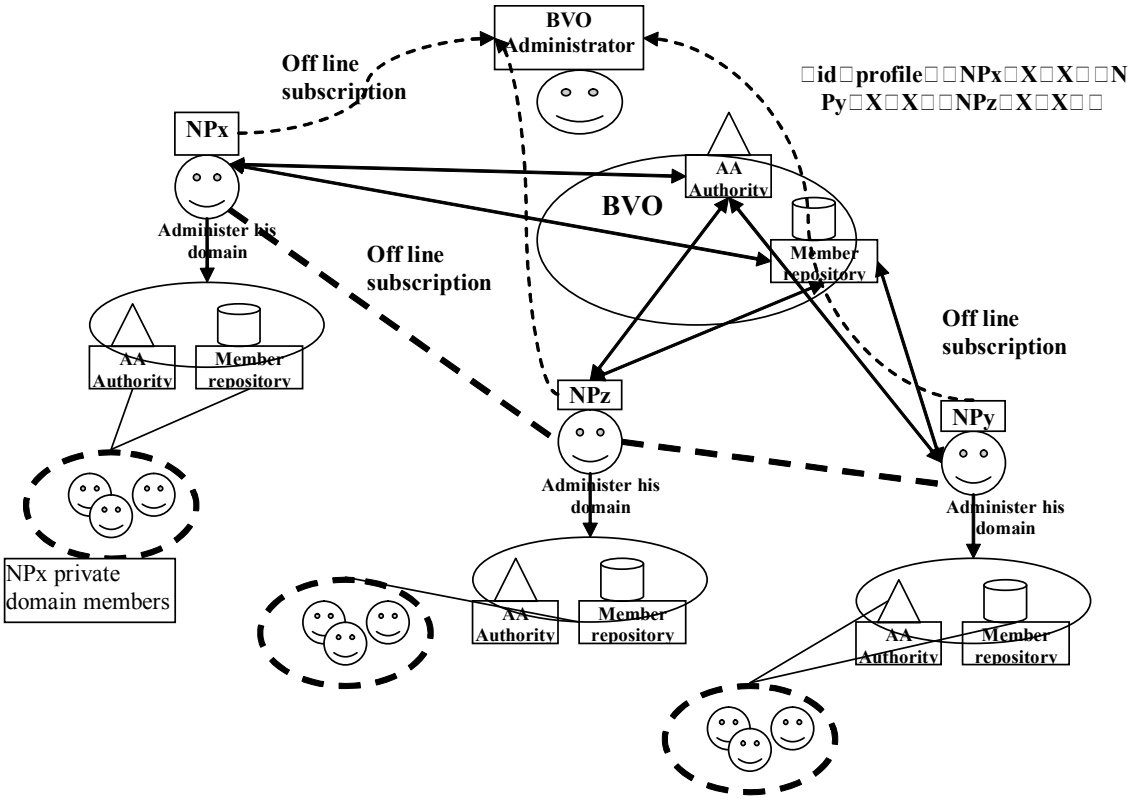

Figure 33: NP as basilar members of the BVO

Different NPs will join the BVO, that is, the BVO administrator will assign to the NP a unique identifier (e.g. NPx.BVO) inside the BVO, a credential to authenticate it (pwd, certificate,…) and a profile that allows to retrieve the associated policies to be applied (see).

We assume that the NPs subscribed to the same BVO will establish a trusted network between them (see A4C description in D4.2.2). Furthermore we are assuming that as AA authority and member repository, the NPs have been using an A4C infrastructure.

In this infrastructure they will store their own members (members of the NP private domain). The members of a NP that is member of a BVO have some privileges, in particular, they can ask for joining the BVO.

This subscription is supposed to be done off line, as well. Among the rights of a NP as member of a BVO there is the capability to present her private members as subscriber of the BVO. Of course the subscription process will lead to have a new member of the BVO that could play the role of customer.

The user1 starts (offline) a negotiation with the BVO administrator and they reach an agreement. The BVO Administrator stores the id of the user1 in BVO member repository and the related policies to be applied by the authorization authority of the BVO. The authentication of the user1 is delegated to the authentication authority of the NP1 domain (there is a trust with it because the NP1 is part of the BVO).

A UserAgent instance is created that will act on behalf of the customer inside the BVO. This UA is a BVO management service and it will be provided on creation with a credential that allows it to play its role inside the BVO.

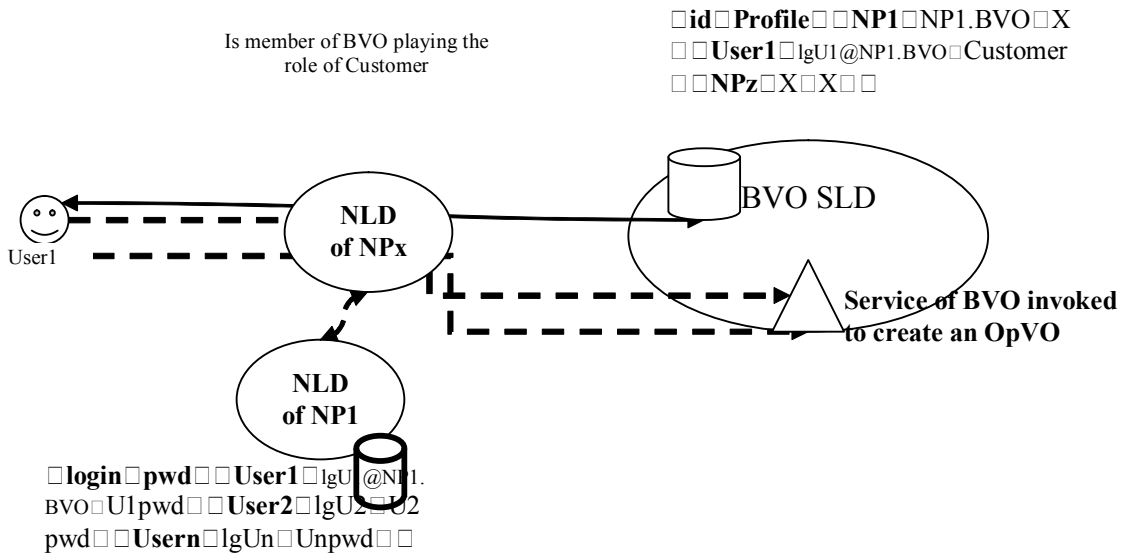

Figure 34: BVO member asks for creating an OpVO

The user1 is a member of NP1 domain. As member of this domain she will have a username and pwd to be authenticated inside the NP1 domain. Otherwise the NP1 is member of the BVO then it will have credentials to access it (let us assume this credential is a certificate). The user1 is aware that his NP is a member of BVO and asks the administrator (offline) to have a credential to subscribe to the BVO. The NP1 delegates its certificate allowing its user1 to contact (offline) the Administrator of BVO to subscribe the BVO assuming the role of customer.

3.14.4. The OpVO creation phase

The User1 that has the role of Customer inside the BVO wants to create an OpVO. In order to reach the BVO SLD, she will use the NP1 (she is member of this NLD) or she can connect through another NPx passing through his NLD. NP1 and NPx are part of the same trusted network.

The goal of this section is to show how the assumptions done until now can help to have a single way to authenticate and authorize User1 during his cross domain transit.

Preconditions:

- User1 has subscribed to the BVO that is he has an User Agent to contact in the BVO SLD.
- User 1 can connect using his own NP or another one that is member of the BVO.

1. User 1 login the network using usrname and pwd. The authentication and authorization process will follow the one described in D4.2.2. Then if it will use the local network (NP1) or foreign network (NPx), it will be the same.
2. The User1 is provided with an IDToken that is a pointer to a SAML token where it is stated that User1 has been authenticated by A4C of NP1.
3. User1 is connected then he can contact his UA to ask for creating an OpVO. The UA will receive the IDToken signed by the A4C server and it will pass it to the authentication/authorization (AA) Authority of the BVO.

4. The AA authority will retrieve the SAML token from the trusted SAML authority and it will check that User1 has been authenticated.
5. The shared identity used by User1 (lg1@NP1.BVO) will allow the BVO AA authority to find this member in the PR and the related profile, through the profile it will be possible to retrieve the policies and the invocation will be authorized or not.
6. The UA can start the OpVO creation process.

The UA will communicate with other services inside the same SLD then we assume the communication can be secure.

Points to be well defined:

- How during this communication will each service usage be assigned to the UA and then to the customer?
- How is the network access allowed and guaranteed in these communications?

As result of the OpVO creation process we propose to create an instance of many BVO management services (this is a slight difference with the current OpVO creation phase), in particular, we will have:

- OpVO Mng instance
- WF Mng instance
- OpVO Broker instance
- OpVO negotiator instance
- OpVO PR instance

All these instances will have some of the rights assigned to their fathers but, in particular, they will be allowed to communicate with services associated to the same OpVO.

Then the first step of the OpVO creation implies the creation of the Mng service instances that will manage the OpVO during its lifetime. In this way the OpVO will define an SLD that will include the above Mng services and the workflow instance of the application. Furthermore, several UA and SA agent instance will be created as well. The UA and SA assume an important role from the security view point because they are the component that should allow the cross and multi domain communication, in fact, they guarantee the secure communication from the external domain to the OpVO (external user of the OpVO) and from the OpVO towards the external domains (SP SLD) that host the orchestrated service instances.

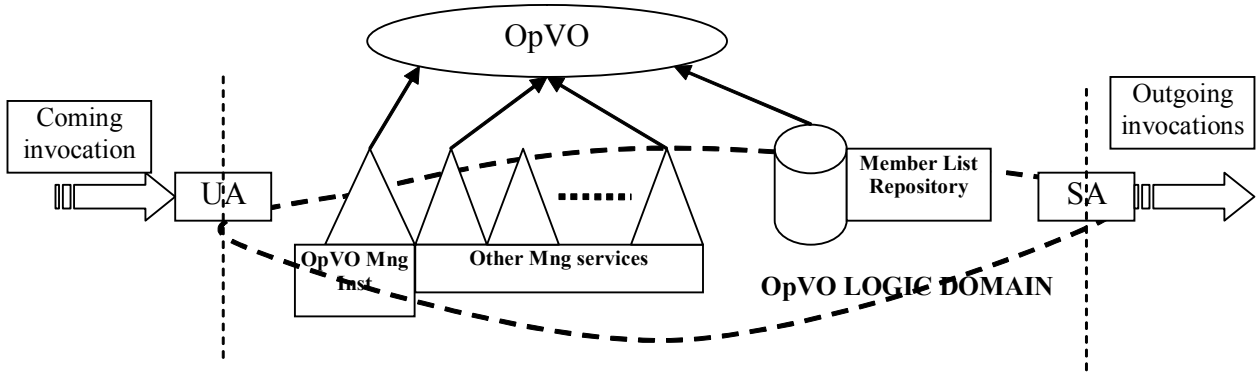

Figure 35: OpVO Creation phase

3.14.5. Adding user to the OpVO

Adding a user to an OpVO can be done by whom has created it. In Akogrimo, the customer can create an OpVO then they can add user to own OpVO.

Adding a user means to instantiate a UA that will accept invocation from an authorized user and that will be authorized to invoke services inside the specific OpVO.

A member of a NP[1] (member of the BVO) will ask for becoming of an OpVO as user.

The User1 will ask (off line) the C1 (they belong to the same NP1 NPL) to become member of the OpVOx.

The C1 as human administrator of the OpVO SLD will use the OpVO Mng inst to create an instance of an OpVO. This will update the OpVO member information introducing this new user (using the same id of used in the NP1 domain). At the same time the OpVO Mng inst will update the information about communication inside the OpVO to allow the UA to contact OpVO services (e.g. the workflow instance).

The User1 will be authenticated by NP1 following a similar approach as the one described in section 3.3.2

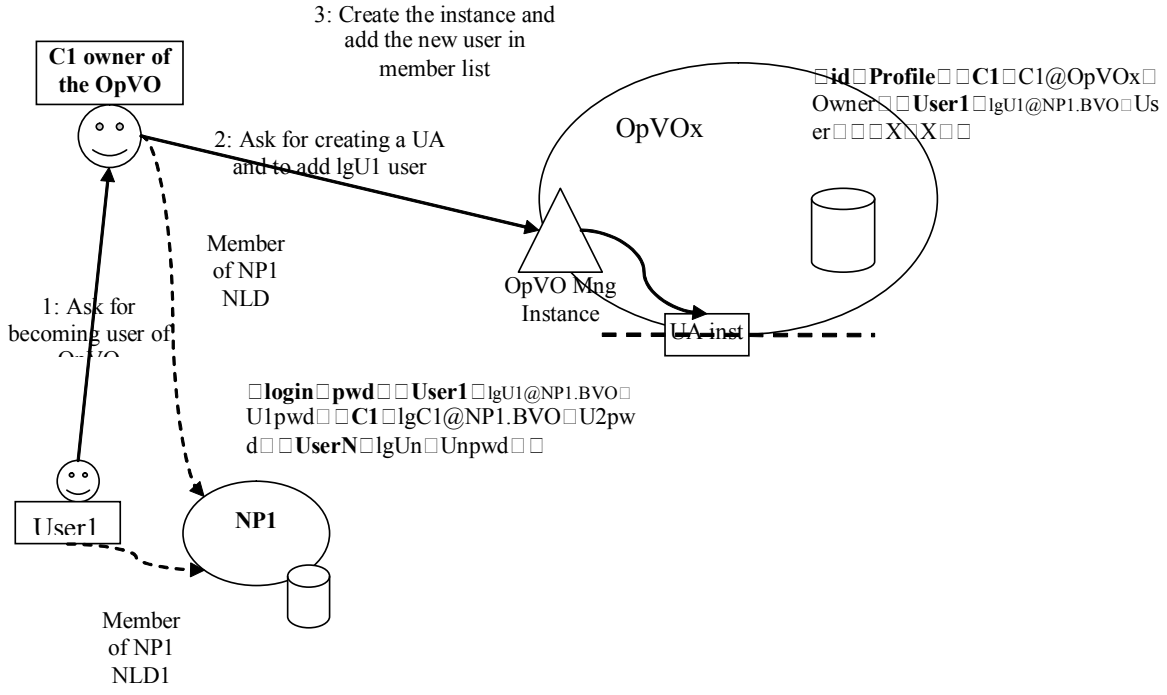

Figure 36: Adding a user to the OpVO

3.14.6. Use of the OpVO

An OpVO is created to make available to Akogrimo users a specific application. The delivery of the application capabilities involves interactions and collaboration between all service instances running in the OpVO.

These service instances can live in different administrative domains and, in this section, we show how using the security and authorization design, described in the previous sections, it is possible manage multi-domain authentication and authorization issues.

3.14.7. Service Compositioning

The composition of several services (hosted in different SP domains) using a workflow is performed in Akogrimo through an OpVO that will host in its domain the workflow management components necessary to address the requirements of a typical Akogrimo business process.

These components are:

- Workflow manager
- Monitoring Daemon
- Business Process Engine
- WF template registry

All these components are part of the Business Process Enactment subsystem of the Grid Application Support Service layer in the Akogrimo architecture.

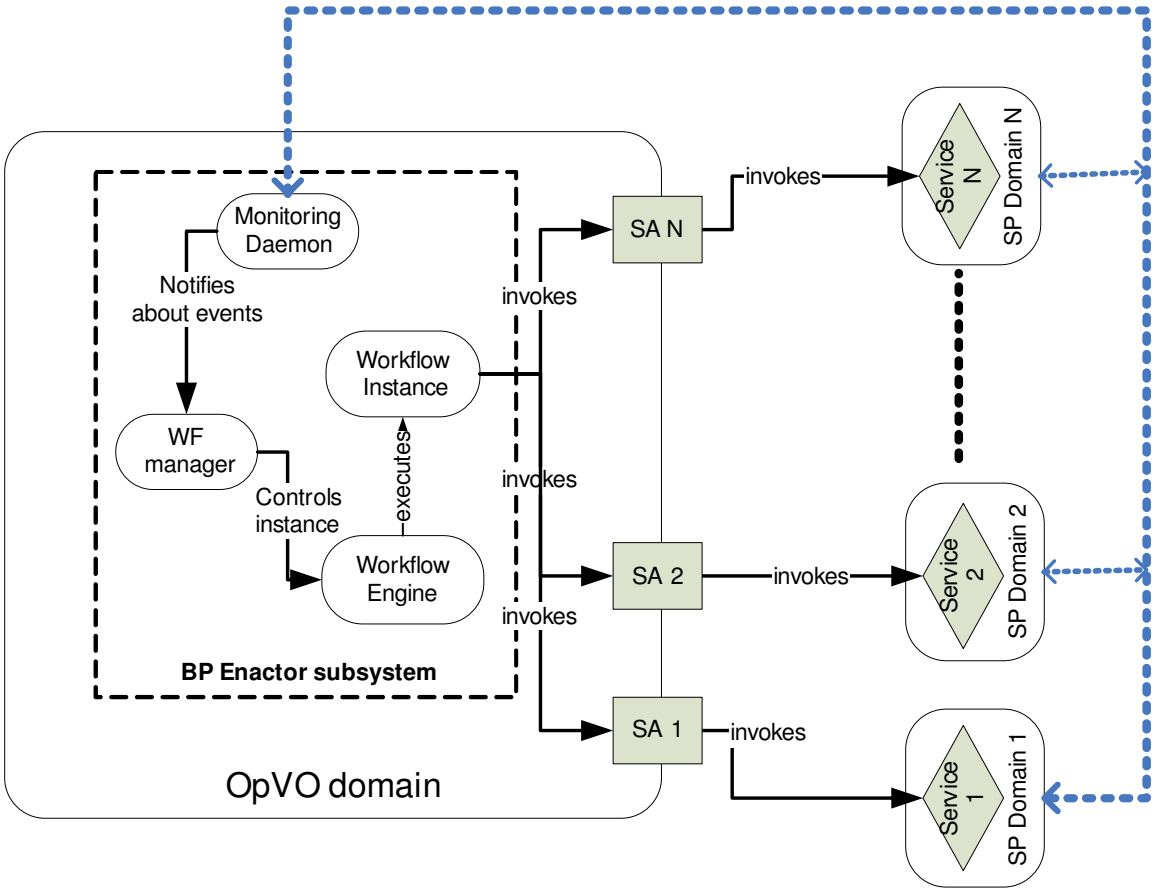

Figure 37: Service Composition

In Figure 37 it is sketched how composition of services is performed in Akogrimo. During the OpVO creation phase the external services to be invoked during the workflow execution are identified and the overall OpVO environment is set up.

This means that the WF manager instructs the WF engine (a BPEL based engine) to instantiate a specific workflow then when an external trigger arrives the workflow instance starts to perform some actions.

During its activities the WF instantiate will invoke external services; actually, it invokes the Service Agents acting on behalf of the external services in the OpVO domain. (SAs are a sort of bridge between the OpVO domains and the SP domains).

The WF instance execution is controlled by WF manager and Monitoring Daemon (MD). The latter is the component that is notified about external events (coming from the SP domain, e.g. context change). If the MD knows has to manage the event it will perform the associated actions else it will notify the WF Manager that in its turn will take a decision about the specific WF instance execution.

3.15. End-to-End security model in Akogrimo

The security problem in Akogrimo means all layers, so means all components of each layer. At this chapter will be analysed the security in Access Network and the security in the Core Network (concerning to SIP entities, SIP Server and SIP Broker).

3.15.1. AR security model

The AR uses a given set of rules determining whether or not to drop (or block) a packet. The information base is provided by the QoS Broker.

A rule identifies a network flow and has information on what to do with it.. Each network flow is identified by protocol, source and destination addresses, their respective port numbers and DSCP code.

Since the AR needs to see the packet headers, end-to-end encryption, or IPSec in tunnel mode, is not possible.

However, IPSec in transport mode is possible. Packets will be encrypted from the MT to the AR, where they are decrypted, allowing the AR to perform its operations. Then, after traversing the core network and reaching the AR that serves the destination MT, they are once again encrypted and finally delivered to that MT.

3.15.2. SIP security model

The following use case scenarios detail the security framework to be applied in Akogrimo project with relation to SIP.

3.15.2.1. Registration

During SIP registration, network authorisation (through the usage of the A4C token, as explained in section 7.2.1 of [6]) takes place. SIP REGISTER request does not contain a body in Akogrimo, so no content integrity protection is required. In this case, the usage of HTTP Digest for authentication seems to be redundant. However, it would be beneficial to add some security mechanism able to provide MitM protection in order to avoid registration hijacking attacks. This kind of protection will avoid also the interception of the A4C token from a malicious entity. TLS and IPSec could be in principle good candidates, but TLS has the inconvenience of run only over TCP, heavier than UDP thus less appropriate in a mobile (i.e. wireless) environment. So the usage of an IPSec connection is recommended.

3.15.2.2. Presence Publication

It wasn't implemented any security kind until now, related presence publication scenario.

3.15.2.3. Session Handling

The same than before (registration process), the security implemented until now during a session setup process is as explained in point 7.2.2 ([6]).

When one user_A wants to start a multimedia session with another user_B would open his appropriate application and start the call. The user_A's terminal would first of all check if he is authorized by the QoS Broker; the session may be denied if network load does not allow it or if the user has exhausted his resources (bandwidth, total data used, etc.).

Then, it sends a SIP request to the SIP proxy, which first of all would have to be authorized. Supposing it was indeed authorized, the SIP proxy would then forward the request to the appropriate user_B's terminal. After that message exchange, the SIP proxy communicates with the destination user_B's terminal and starts the normal SIP message exchange. When the SIP initiation process is complete, the communication is direct between the two users' terminals.

3.15.3. SIP with SOAP security model

In this section security mechanisms to protect the different paths in which "SIP with SOAP" attacks can take place, as well as how to include the OpVO security perspective.

3.15.3.1. VO security

The adopted solution to take into account the OpVO security perspective takes advantage on the extensibility and backward compatibility of the SIP protocol, and consists on the definition of a new header field containing the OpVO membership token, which will be included into the SIP signalling messages which form a part of the GRID-triggered SIP interactions. This new header will be processed at application level at the MTs, providing end-to-end security. Anyway, it will have a meaning only within the Akogrimo context; any other "non-Akogrimo" SIP entities are requested to ignore it, as SIP protocol specifies.

So using this approach, when a MT is requested to provide its connection details (which will allow it to be contacted by the GRID entities), it can check if the request coming from the SIP Broker contains a valid OpVO token before sending a response. In this way, the MT can check that the interaction is authorized. The only requirement is, this information should be included somehow in the EMS request to the SIP Broker, as the SIP messages will be sent to the MTs by the SIP Broker in response to a WS request from the EMS.

In the case of additional services as GRID-triggered session establishment, if two MTs are to be put in contact, the OpVO token will be included in the request reaching the initiator MT. When producing the needed signalling messages, the initiator MT will include its OpVO token in the request, so the receiving MT can be sure the session is authorized. The MTs will obtain their OpVO tokens by connecting to the corresponding VO User Agent during the VO login. The User Agent is created by the workflow to represent a particular user to the workflow. The user "connects" to the User Agent from a particular MT (authentication, selection of OpVO, authorization, etc). At this point, each MT can be given its OpVO token.

The best aspect of this approach is that the SIP network entities do not need to be secured from the OpVO security model perspective - being outside the OpVO firewall they are not a security threat (other than from a denial of service). Of course, self protection for the SIP Broker and the

SIP Server would be beneficial, and these could use the same mechanism (i. e. the baseVO dynamic perimeter) to ensure they only respond to valid Akogrimo OpVOs and MTs.

3.15.3.2. WS interaction path

As described earlier, in principle one attacker could impersonate the identity of the EMS or the Workflow Manager and make a non-authorised use of the WS exposed by the SIP Broker. Apart from this, the WS communication path can be intercepted and the exchanged messages modified by a non-authorised entity. In order to prevent this, specific WS security mechanisms are envisaged.

WS-Security core specification defines several mechanisms to provide protection against SOAP attacks. It includes also mechanisms to detect if some message has been manipulated (XML signature) and encryption mechanism as well (XML encryption).

However, the adoption of the OpVO security model provides a higher level of protection, because the attacker should include a valid OpVO token in the request; however, it will be rejected.

3.15.3.3. RMI path

Internal communications between the SIP Broker's SOAP interface and the broker engine is made by means of an RMI interface. This means that remote clients could access the classes and resources that the RMI server (the broker engine in our case) makes accessible. So if some malicious entity knows how to invoke the corresponding methods, it could impersonate the identity of the real SIP Broker's SOAP interface and could make unauthorised requests.

To prevent these kinds of attacks, we must take advantage on the Java Security Model. Using this, you can restrict the actions that remote entities can perform. By default, an RMI program does not have a security manager installed, and no restrictions are placed on remotely loaded objects. Anyway, the java.rmi package provides a default security manager implementation that can be used, which requires the specification of a security policy file containing specific permissions regarding which operations are granted, and from which locations (machines, ports). So the best way to proceed is to specify that only requests from the machine in which the SIP Broker's SOAP interface is running are allowed.

The current implementation of the SIP Broker enables the configuration of the policy file to be used. No security policies are applied if this file is not specified.

3.15.3.4. SIP path

The SIP-related security problem last section describes (some malicious entity posing as the SIP Broker User Agent) can take place on the network side of the SIP path (SIP Broker – SIP Server hop) or in the user side (SIP Server – MT hop). The usual solution for the network hop is the usage of some certificate-based security mechanism, like TLS, which provides data integrity and confidentiality, preventing from MitM attacks.

Additionally, in order to avoid a malicious entity registering itself as the SIP Broker, deregistering the valid IP address binding and assuming the identity of this component, the SIP Registrar (embedded into the SIP Server) could be also configured to accept only a registration binding from a preconfigured IP address, or HTTP Digest authentication mechanism described in RFC 3261 (which is based on HTTP authentication [6]) could be used. This approach would provide also good DoS attack protection, since the server processing the SIP registration can remain stateless until a new request arrives, and all transactions from non-registering entities are rejected.

Finally, to avoid a direct attack to the MTs making use of the peer-to-peer nature of SIP (i.e. the attacker sends the request directly to the mobile terminal, impersonating both the SIP Server and the SIP Broker) network layer security mechanisms can be used (e.g. using an IPSec connection that refuses all IP packets not coming from the well-known SIP proxy IP address and port). Digest User-to-user authentication (RFC 3261, section 22.2) can be also used, but it implies more network traffic and a higher delay (e.g. grid call or grid transfer establishment time will be higher). Anyway, in this case the OpVO security model provides again a reasonable security level.

3.16. SOAP with SIP: SIP-based GRID session management

One the most important technical challenges addressed by AKOGRIMO is the integration of the GRID environment in a ubiquitous and mobile context where the different actors (users and services) can move to a different terminal (with a different network address), or appear/disappear suddenly.

All these issues have been the focus of well-know network protocols like the Session Initiation Protocol (SIP), which establishes an architecture for establishing and managing sessions in a mobile and ubiquitous environment.

The current usage of SIP has been focused to the management of multimedia sessions, with Voice over IP as "killer application" of the SIP usage, since it is a paradigm of the mobile and ubiquitous scenario where SIP is applied. But there are very few applications of SIP beyond this multimedia session management.

In the scenarios defined for AKOGRIMO, there are a variety of application sessions, from multimedia session with videoconferencing between mobile users to grid-related sessions where a user is accessing a service located in a mobile and/or ubiquitous terminal. So it is needed a framework for addressing this mobility and ubiquity feature in non-multimedia sessions like Grid sessions, problem also known as the ***"SIP with SOAP interaction"*** where the main protocols (SIP for ubiquity and SOAP for Web Services) have to interact.

Deliverable D3.1.1 included an introduction to this problem, a description of the SIP and SOAP architectures, and some alternatives for addressing the combination of SIP and SOAP. These were the following:

- Signalling over SOAP: Using SOAP as a transport protocol for a signalling protocol which will include session management and user mobility. So, the applications have to use a signalling protocol to locate ubiquitous users and manage the sessions with them. This signalling protocol will be implemented as a web service, using SOAP to transport it. One example would be using SIP over SOAP or to extend some of the current proposals for service migration and relocation based on Grid Information Services and UDDI.

- SOAP over SIP: this approach will hide all the mobility and ubiquity problems to the Grid applications, since SOAP will be transported in SIP messages which will manage sessions and mobility/ubiquity in a transparent way for the services. There were some proposals for this approach, proposing a new method in SIP for transporting SOAP messages or using the MESSAGE SIP method for transporting them [101]. However, all these approaches have failed because of two main reasons:

    - SIP has not been designed as a transport protocol but as a control/signalling protocol. Although it is being used for transporting short text messages (MESSAGE method used in Instant Messaging), using SIP as a transport for SOAP can lead to problems with large messages or lots of messages because of the potential for SIP to use UDP as an underlying transport protocol.

    - This approach will imply the modification of all the current Grid platforms which are based on mapping SOAP over HTTP.

- SOAP for application, SIP for signalling. This approach combines both worlds, so applications will use directly SIP for signalling, 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 will not impose any change in the underlying GRID platforms, but it will require that the GRID applications have to be SIP-aware.

So, the final decision was the combination of both worlds (SIP for signalling, SOAP for application transport) and make this as another clear example of a cross-layer issue between the network and grid worlds, where in this case, the GRID sessions can take advantage of the SIP protocol for managing GRID session in a mobile and ubiquitous scenario.

So, with this approach GRID application would be able to:

- Establish a GRID session with a mobile/ubiquitous service without a prior knowledge of its current location.
- Use the SIP mechanisms to manage sessions, like transfer a session, or control a GRID session by a 3rd party.
- Using the presence and context management mechanisms established by the SIP SIMPLE infrastructure.

But this approach will impose some modifications to current Grid applications in order to use SIP for session management:

- At start up phase, mobile/ubiquitous services will be registered in the SIP registrar with the SIP REGISTER method.
- The applications, instead of using EPRs (End Point References, as stated in WS-Addressing) for identifying resources, should use in the establishment phase SIP-URIs for identifying mobile/ubiquitous services.
- Then, they will send a SIP invite to the service URI.
- And in the SIP answer, by means of a specific "Grid Session Description Protocol", the application will get the needed Grid session information to start the Grid invocations (i.e. the current service EPR).

Also, Grid applications could include the adequate logic for:

- use further session management features provided by SIP like session save/load/restore, useful also for session transfer. There are already some proposal for this feature (Web Services Continuity), like the approach presented in [102]
- react to context awareness changes which are notified with SIP.

There are already some specific proposals at an architecture level for implementing this, but this has to be validated in the specific development working groups. The initial idea is the following:

- At startup, services will register in the SIP Server (1) and publish their presence data (2). When the EMS needs information about the status of the services, it will subscribe to its presence via WSRF Subscriptions (3). This will trigger a SIP Subscription to the correspondent SIP Presentities by the SIP Broker (4). From now on, the SIP Server will inform the SIP Broker about every change in the status of the services (5), which will in turn inform the EMS (6). In the figure, the EMS is informed about the availability of the services Alice and Bob.

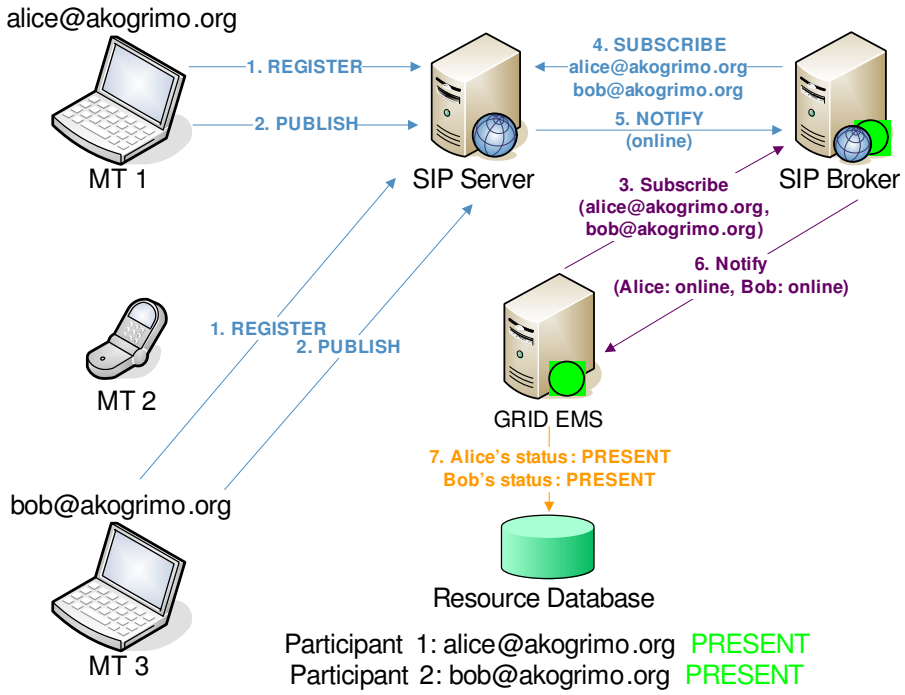

Figure 38: Presence using SIP

- When the EMS needs to contact a service, it will refer to the SIP Broker to obtain its connection details (1). The SIP Broker will then establish a SIP session with the ubiquitous service (2-5). In the answer, the contacted service will include a special session description document containing its connection details, i.e. its IP address or EPR (5). With the details, the EMS is able to continue with the management of the Virtual Organization (6).

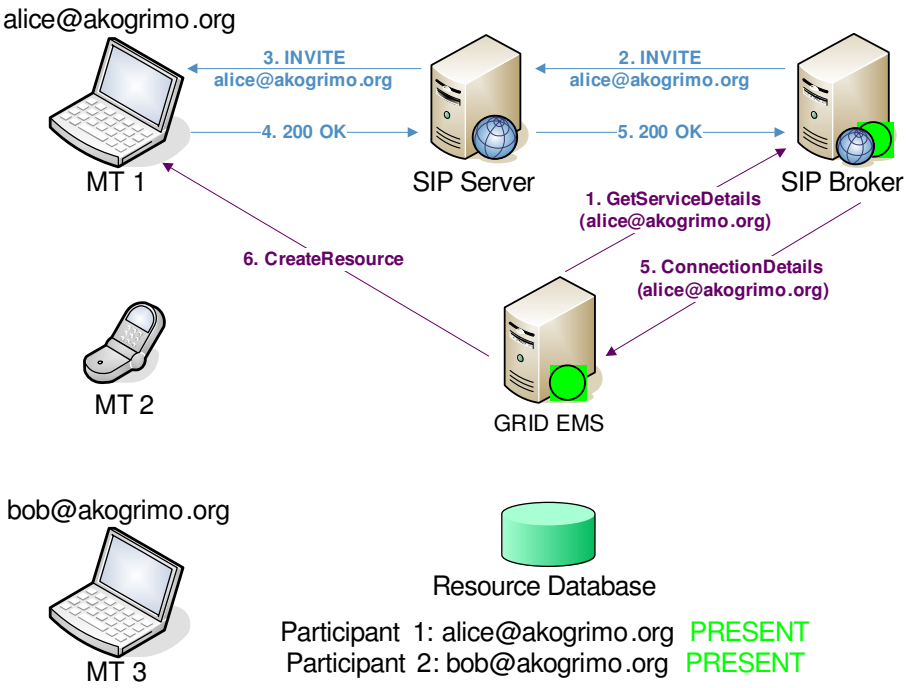

Figure 39: Resource Creation using SIP

- If, at any time, the previously established session terminates, either voluntarily (the terminal is switched off – A steps) or accidentally (the connection drops – B1 step), the change in the status of the service is communicated to the EMS (C1).

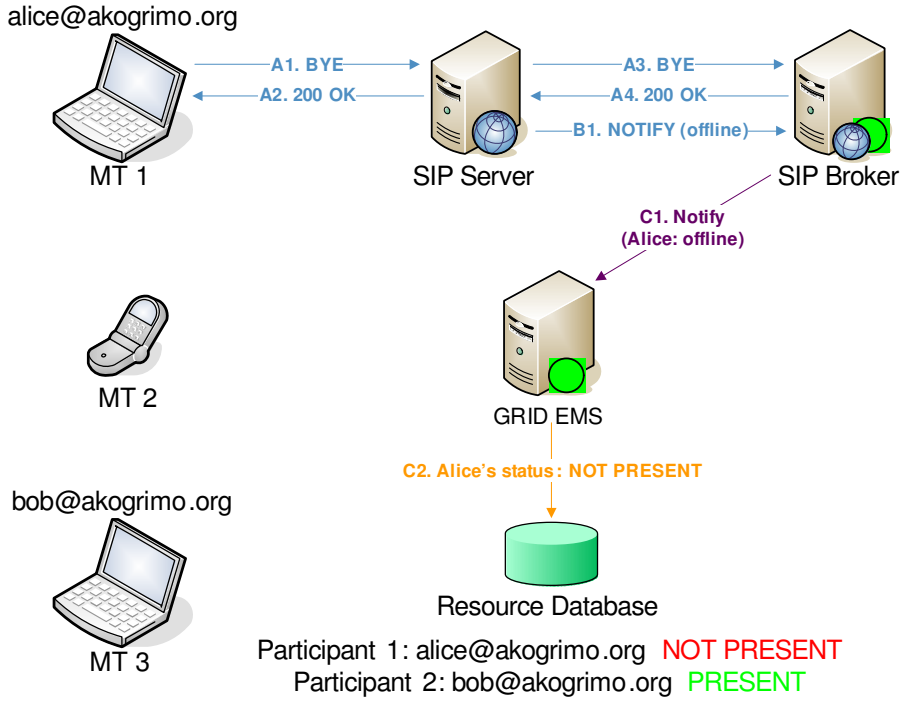

Figure 40: De-registration/Notification

- If the service reappears in another Mobile Terminal (1, 2), the EMS will again be informed (4), for it to be able to include the service again in the VO management process.

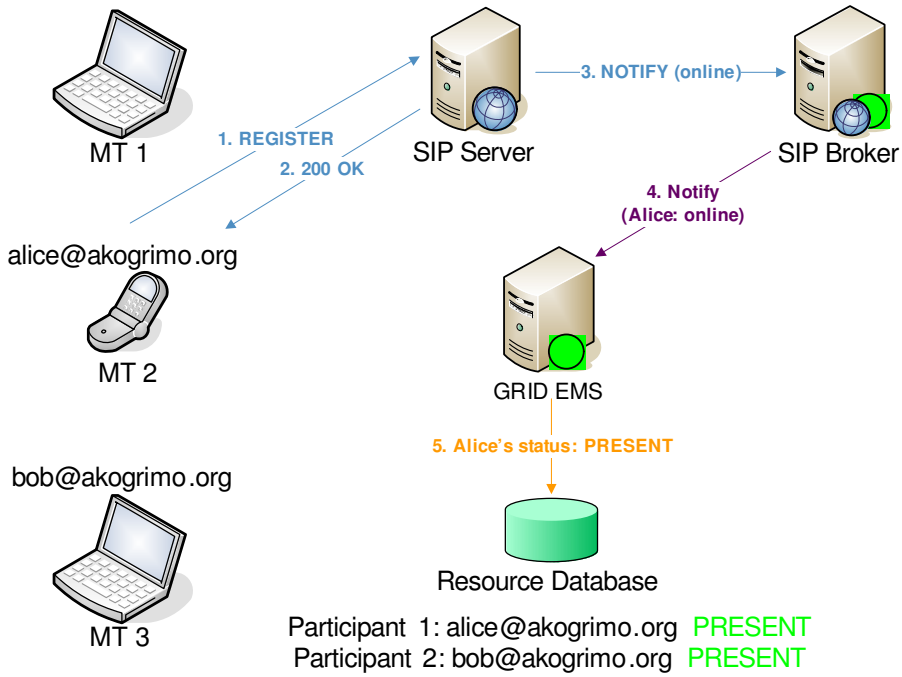

Figure 41: Present Status Change

3.17. Business process concepts and Service Provider composition of services in Akogrimo

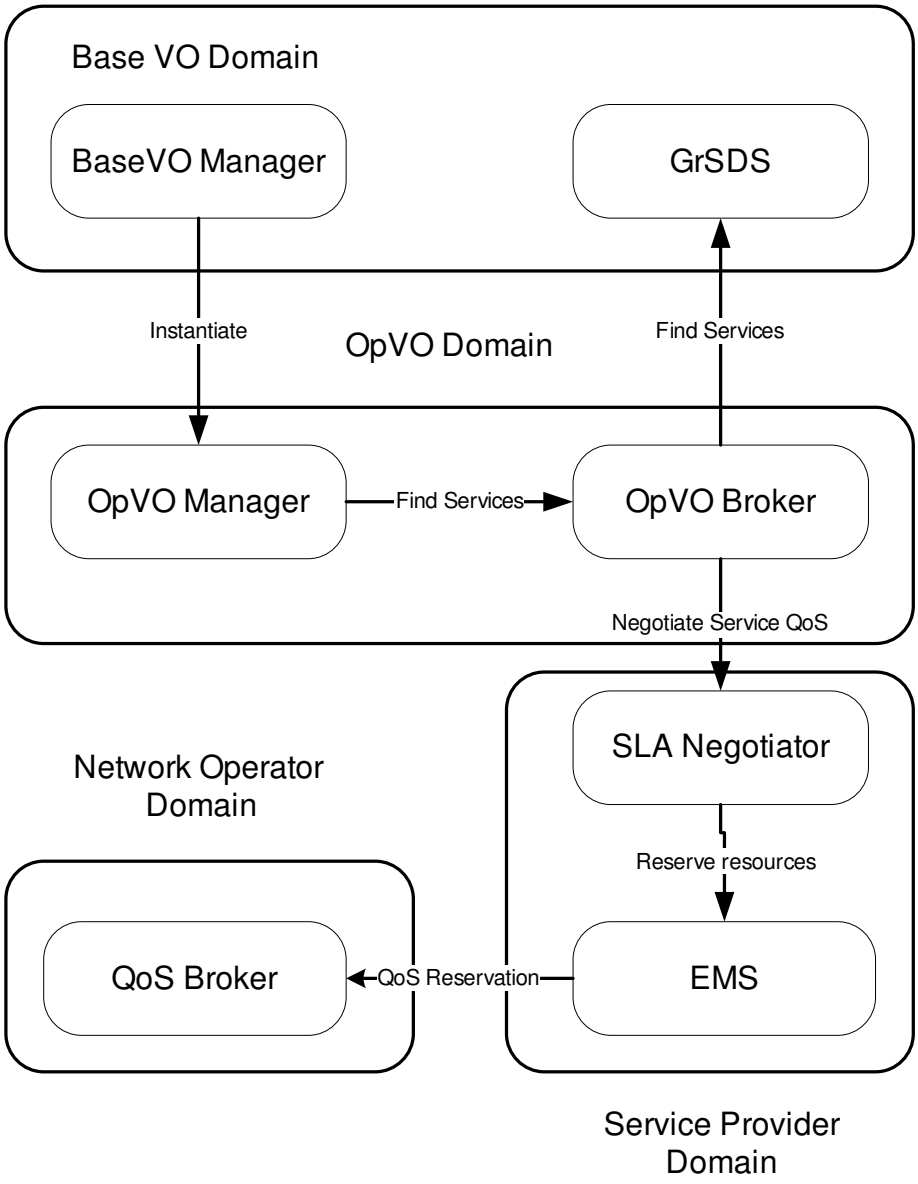

Figure 42: Akogrimo Service Composition

The Akogrimo architecture is intended to support business process designs that both support and take advantage of dynamic, mobile services and users.

It is important to distinguish between the terms "business process", "workflow", "choreography" and "orchestration". By a "business process" we mean a high-level description of a process in terms that are meaningful at a business level (as opposed to a computational or engineering level). Such a description should be abstracted from any particular implementation of the process, and should be more concerned with requirements and goals than specific execution approaches. At the opposite extreme, a "workflow" is a precise definition that is (or can be easily converted into) an executable form. The basic steps of a workflow are normally service invocations; though even here the workflow itself is not concerned with how the service is implemented (though it does have to depend on and make assumptions about the service's behaviour).

"Choreography" refers to a form of process control where there is no single centralised controller, but where the details of execution of a process are delegated to individual execution sites. A choreography may describe how the different parts interact, but will almost certainly not describe or control how each part performs (only that it produces the expected interactions at the expected times). In "orchestration" on the other hand, the behaviour of a process is under control of a single central agency which controls the actions of each part and mediates in (or at least knows about) all interactions between them.

To some extent, the distinction between choreography and orchestration is a matter of detail. For example, an ActiveBPEL engine running a workflow script "by itself" is acting as an orchestrator; but even so, it does not (and cannot) define or control the implementations of the services that are used in the script. In the other direction, the script may be being executed as part of a larger choreography that links other workflows. In Akogrimo, the need for a "big picture" view capable of assessing and handling context changes lead towards an orchestration solution, while the dynamic nature of most of the applications tends to support devolution of responsibility to choreographed services (ie they know what they need to achieve and just get on and deliver it)

Akogrimo intends to cater for the mobility of participants (both users/clients and services) in a business process. One consequence of this is that the business processing components must "track" users and services as they change location while retaining their identity, but must also support the ability of the process to adapt to changes in context of such mobile agents, for examples, changes in their capabilities, discovery of alternative services, and responding to situations where an agent becomes disconnected. Of course, some of this adaptation can be delegated to the services (particularly the "how" to adapt in order to still deliver in the face of a context change) but ultimately any change that may require a change in the overall business process must be handled at the orchestration level (the "what" to adapt in order to satisfy the possibly changed business objectives). For example, the availability of a high definition screen in an eHealth scenario may mean the presentation of some medical data is changed (graphs instead of text), or it may mean that the pattern of interaction with the user is changed (patient can be treated on the spot rather than being immediately rushed to hospital). One immediate consequence of this is that the business process enactment sub-system needs to have access to the context information associated with all its users/clients and services.

The final requirement generated by Akogrimo's mobile and dynamic nature is the need to build on-the-fly secure Virtual Organizations, where the data can be shared among the dynamically changing members but prevented from falling into the hands of outsiders. The security is implemented using security tokens valid inside the VO boundary and validating the identity of the external invoker through an interaction with the Akogrimo A4C servers hosted in the NP domains. The VO security stops at the User and Service Agents, which act as proxies for the user and service; the security of the links beyond these agents is the responsibility of the respective user/service provider. Within the VO, security is the responsibility of the VO manager, which can dynamically alter the internal security to reflect the state and requirements of the workflows.

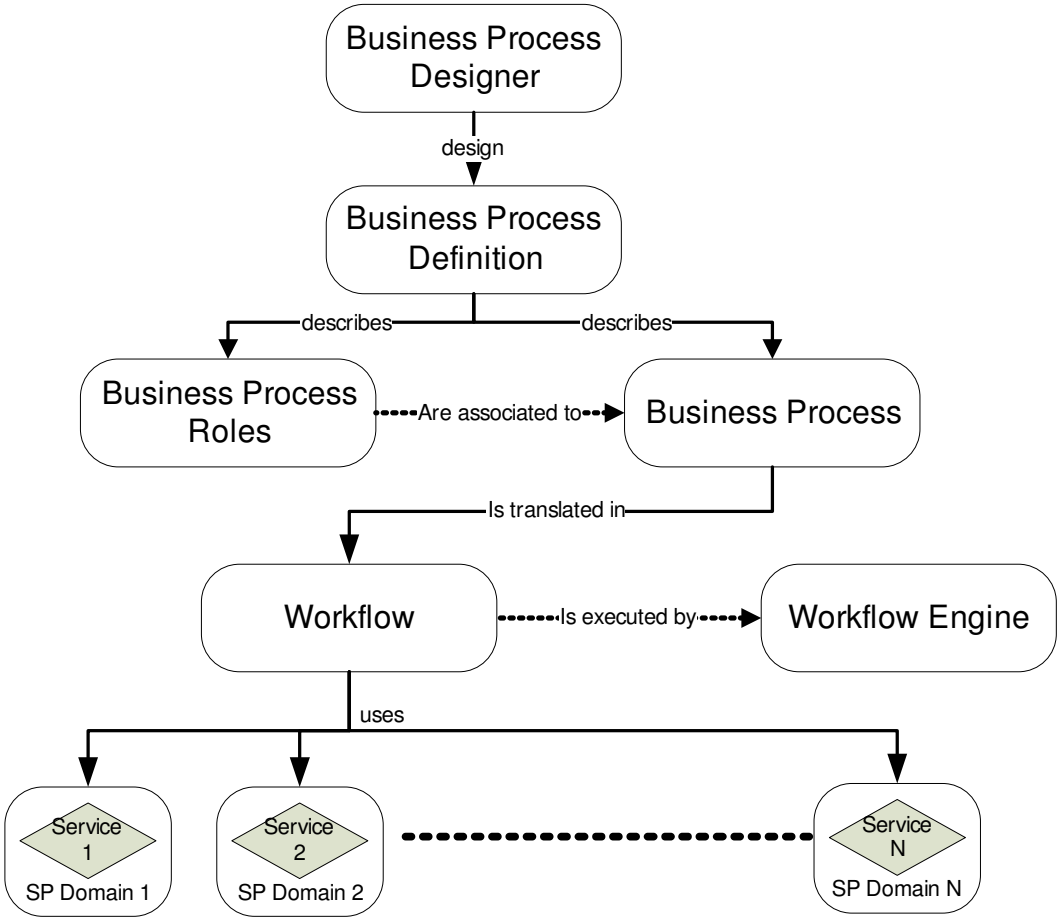

Figure 43: Business Process concepts

A service provider manages one or more services hosted in their private domains. 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.

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 Business Process Enactment (BPE). It is possible to say that the BPE is responsible for managing *dynamically* the aggregated service together.

The service orchestration performed by the BPE is very important because it takes into account the context in which the services must be executed (mobility, network availability, etc) and handles any event that has to be managed at workflow level.

4. Selected key Scenarios

In this chapter we present the following scenarios:

- Akogrimo Log-In
- Service Invocation and Charging
- Base VO Registration
- OpVO Creation
- Context Sensitive Services

4.1. Akogrimo Log-In

The Akogrimo Log-In scenario demonstrates four important features of the Akogrimo architecture:

- Authentication & Authorisation across organisational boundaries
- Network QoS setup
- SIP presence announcement
- Returning a Base VO reference to the user's terminal
- Returning a public customer ID to the user's terminal

Trust relationships between different network operators enable the information exchanged between A4C servers in different domain, thereby enabling cross-organisational authentication, authorisation and accounting.

Right from the start the quality of service parameters are adjusted in compliance to specified network policies.

As soon as the network connection has been established the presence of the user is communicated to a SIP server. This presence announcement makes the user available as service consumer and service provider. The system can react on the user's presence status e.g. by delivering waiting messages or resuming pending workflow actions.

The Base VO reference that is transmitted to the user's terminal during the login procedure is the user's link to their service delivery platform, the virtual organisation. The public customer ID is the identifier of the user in the domain of the customer. Customer's domain and Base VO domain have a trust relationship to enable the link between the Base VO domain and the user's domain.

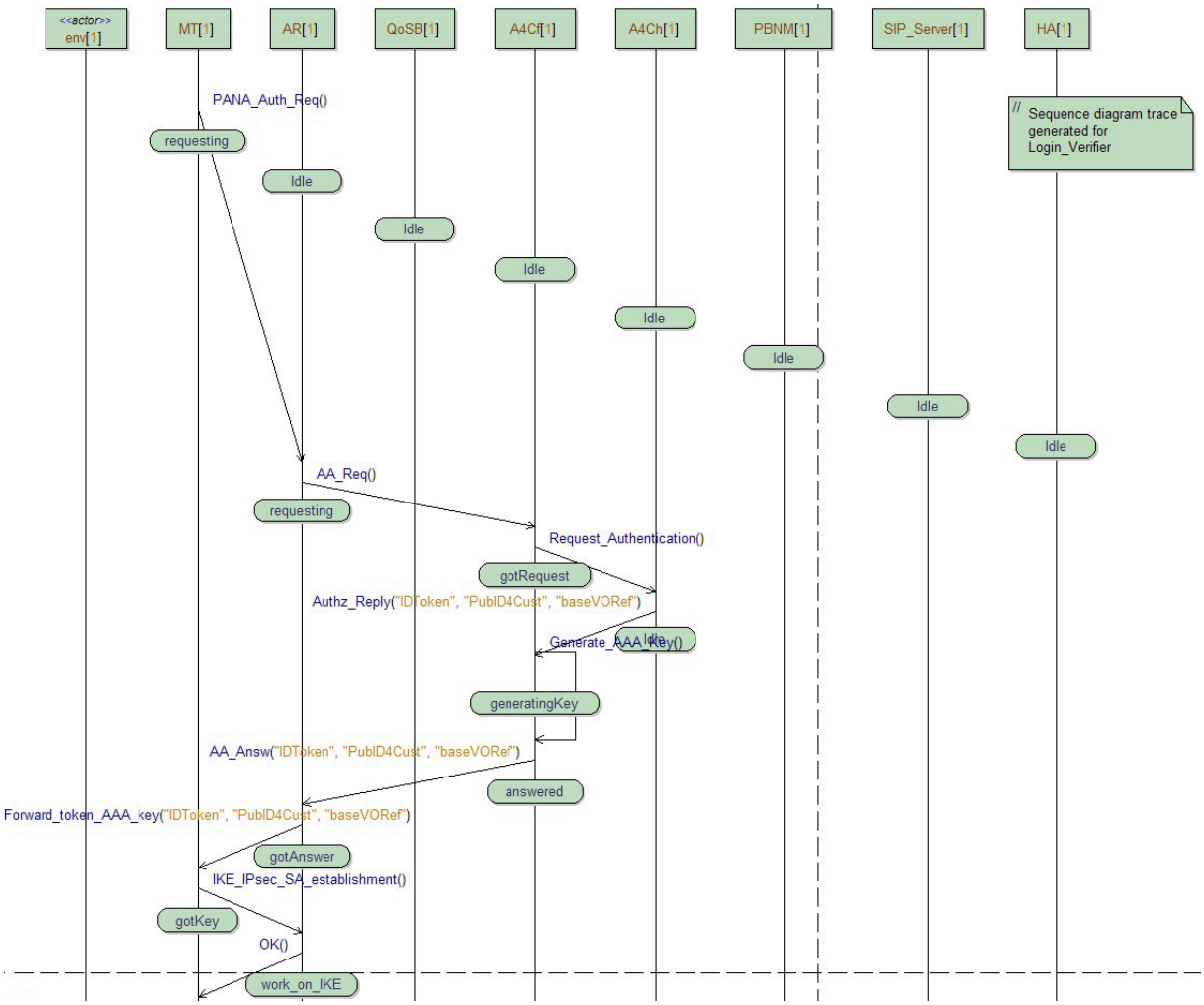

Figure 44: Login Sequence (1)

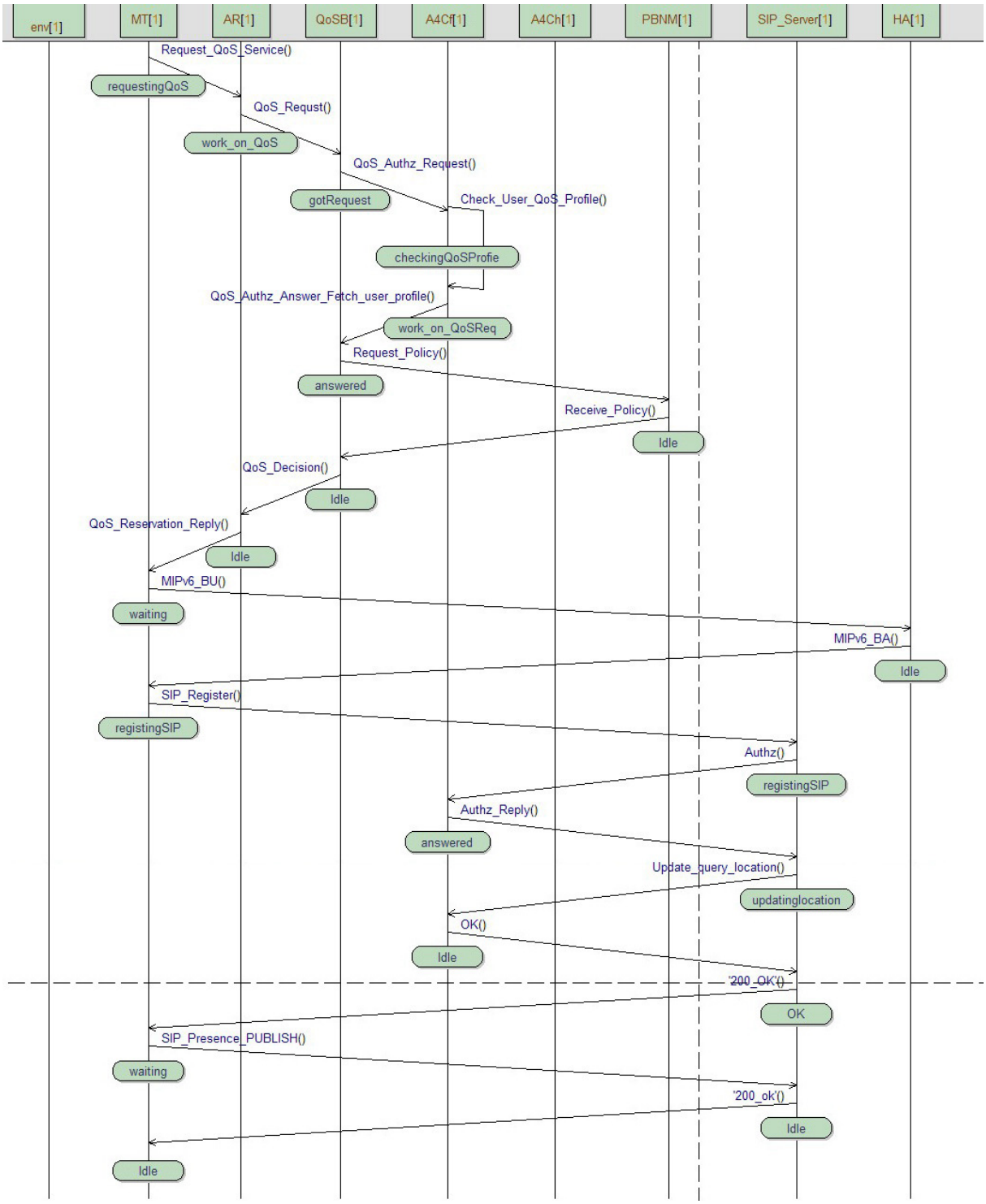

Figure 45: Login Sequence (2)

4.2. Service Invocation and Charging

The following sequence diagram demonstrates the invocation of a service by the user and the charging of the service by the A4C.

The preconditions are:

- The user is already a member of the Base VO and has an identifier (public ID) in the customer's domain.

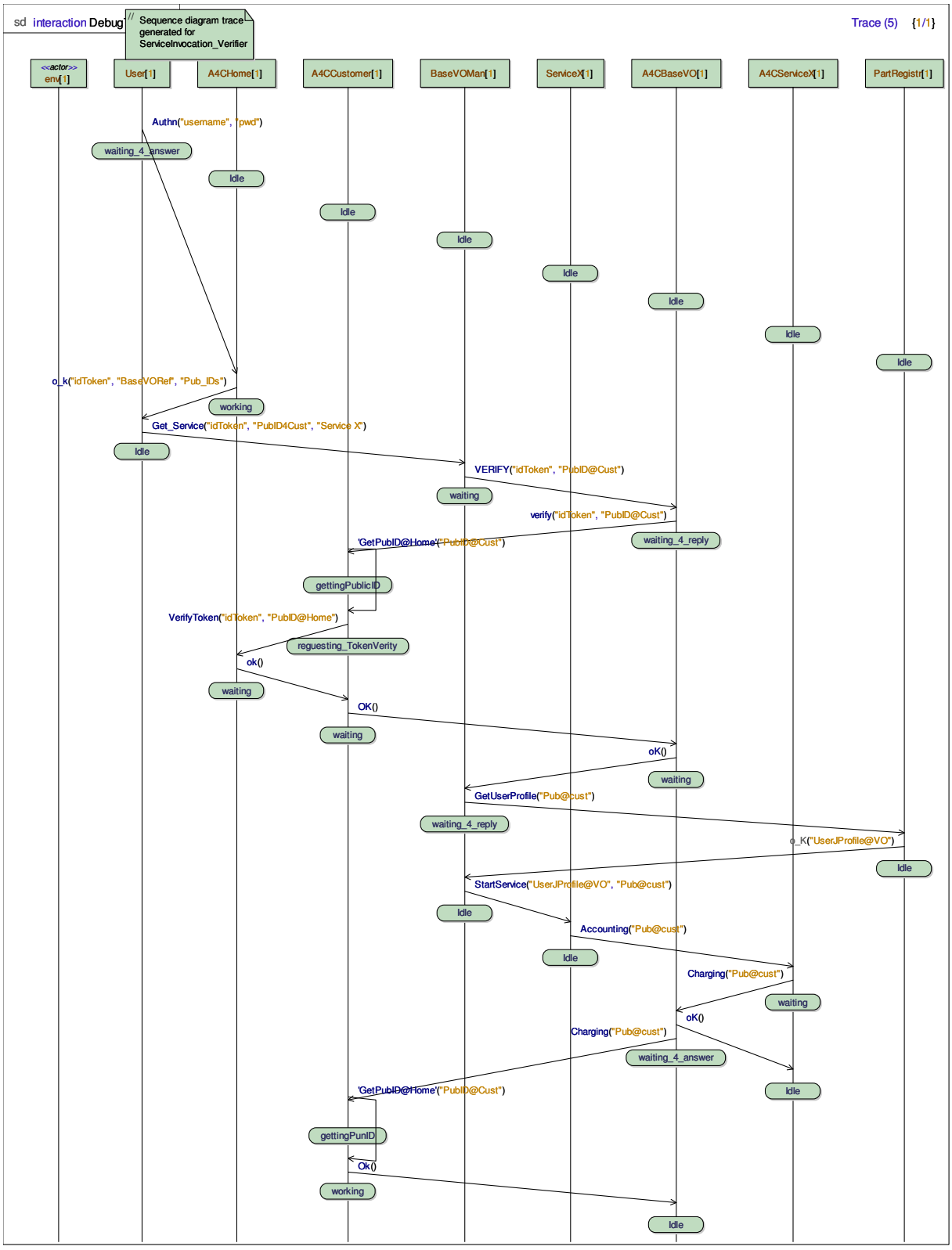

Figure 46: Service Invocation and Charging

- The user and the customer belong to different domains but a trust relationship has been established between their domains.
- The customer and the Base VO domain have enabled the trust relationships

In this scenario the following actions are shown:

- The user logs-in

- Invocation of the service through the Base VO
- Verification of the IDToken (cross-verification)
- Retrieval of the user profile
- Charging of the service (cross-charging)

4.3. Base VO Registration

The Base VO Registration demonstrates the registration of the user by the customer. The preconditions are:

- The user and the customer are at the same domain (they may come from different domains, but for simplicity this is not shown in the diagram)
- The customer and the Base VO domain have enabled trust relationships

The phases at the sequence diagrams are:

- Generation of the new identifier "publicID" for the user at the Base VO
- Verification of the IDTokens from the user and customer
- Provision of the specific user profile to the Base VO,
- The user profile is approved under the conditions of the Policy Manager

The post conditions are:

- The user is a member at the Base VO

A new pair (PublicID, profile) is stored at the Participant Registry of the BaseVO

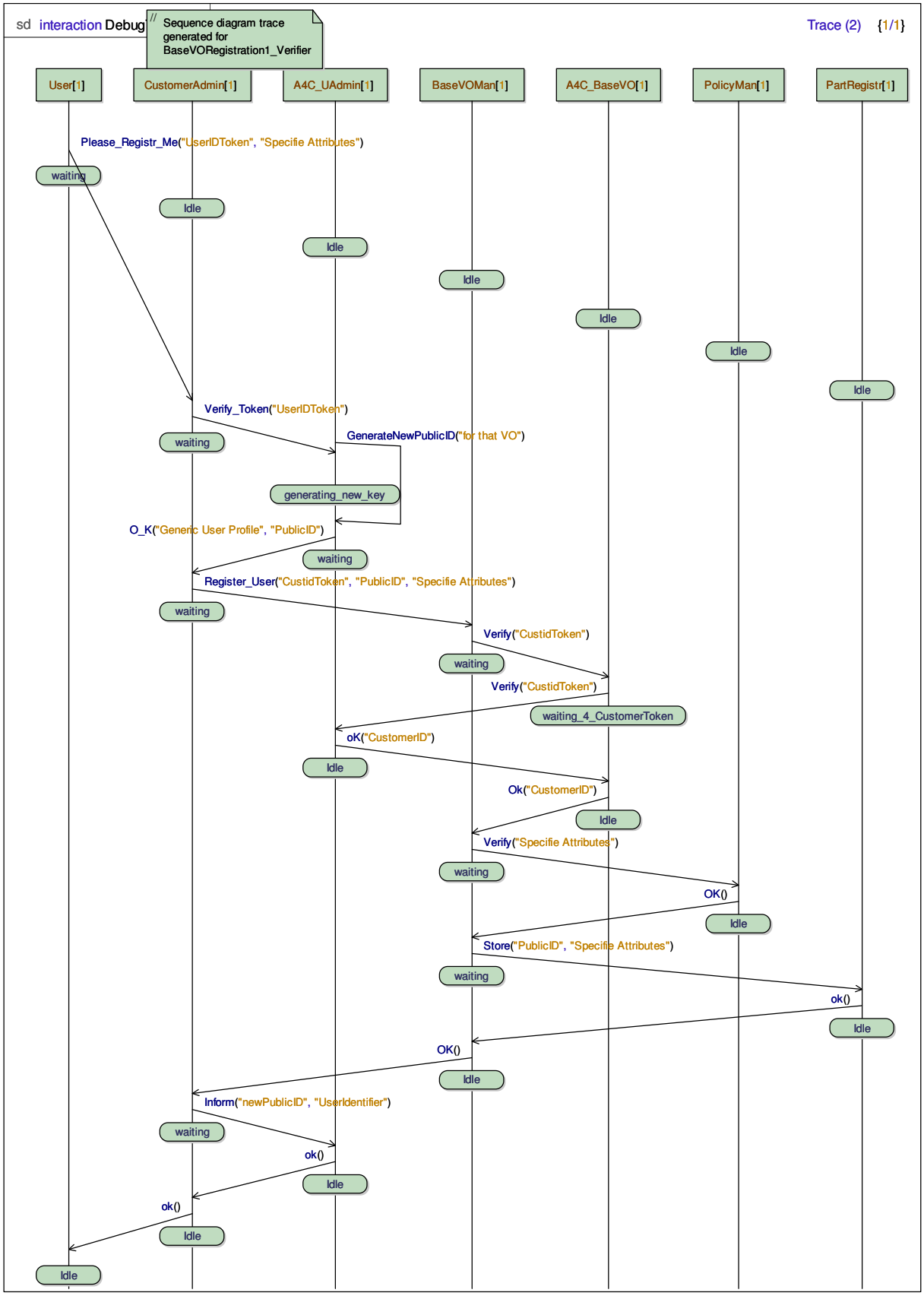

Figure 47: Base VO Registration

4.4. OpVO Creation

An OpVO is the operational environment for service delivery in Akogrimo. The creation phase involves a sophisticated search and negotiation procedure to find and bind the required services according to the customer's request. In the scenario, the requested service is implemented by a workflow. This scenario demonstrates:

- Service selection by the user
- Negotiation of the service quality
- Contract establishment
- Service deployment

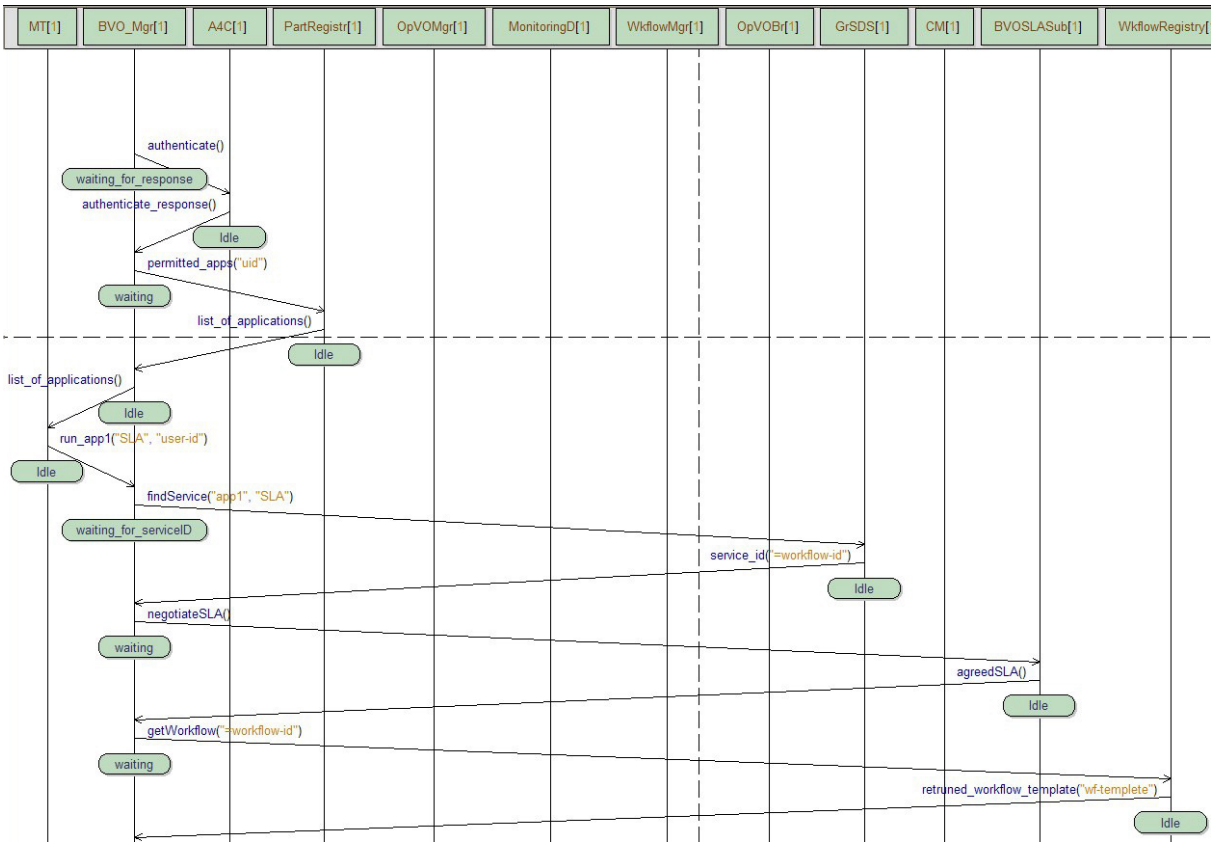

Figure 48: OpVO Creation (1)

The Base VO Manager offers a list of applications, e.g. patient monitoring services. The details of these applications and the quality of service can be specified by the customer. This customer created SLA is used in the service discovery to find the most suitable services in the Grid Service Discovery Server (GrSDS). The GrSDS has to be able to understand the SLA parameters and match them with parameters of registered services. In case a service is represented by a workflow, the SLA-Subsystem of the Base VO negotiates high level service parameters with the customer.

The OpVO Manager is created after the workflow service is selected. The OpVO Manager handles the actual instantiation of the OpVO. The OpVO Broker is responsible to find services that fit into the workflow description from functional and non-functional point of view. Also the involved human actors, e.g. a teacher who is part of an eLearning business process, establish a contract with the OpVO by means of their SLA component. After all services and actors that are involved in the business process have been found, the workflow is deployed into the Enactment Engine.

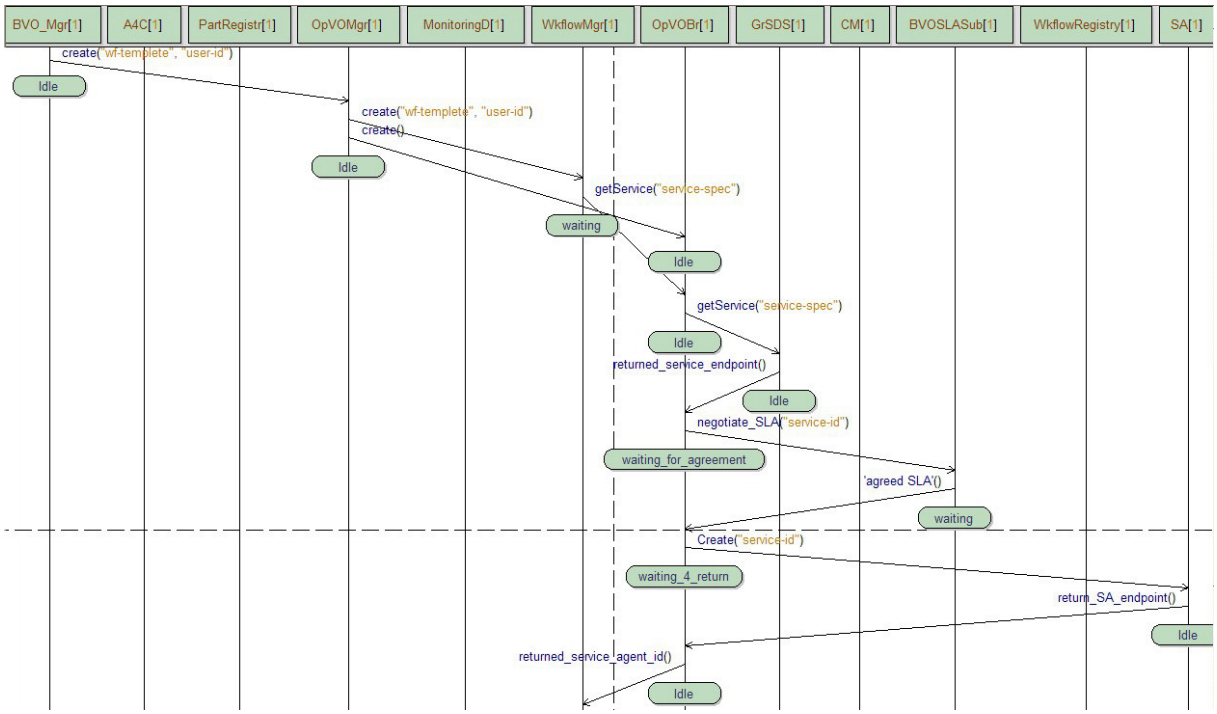

Figure 49: OpVO Creation (2)

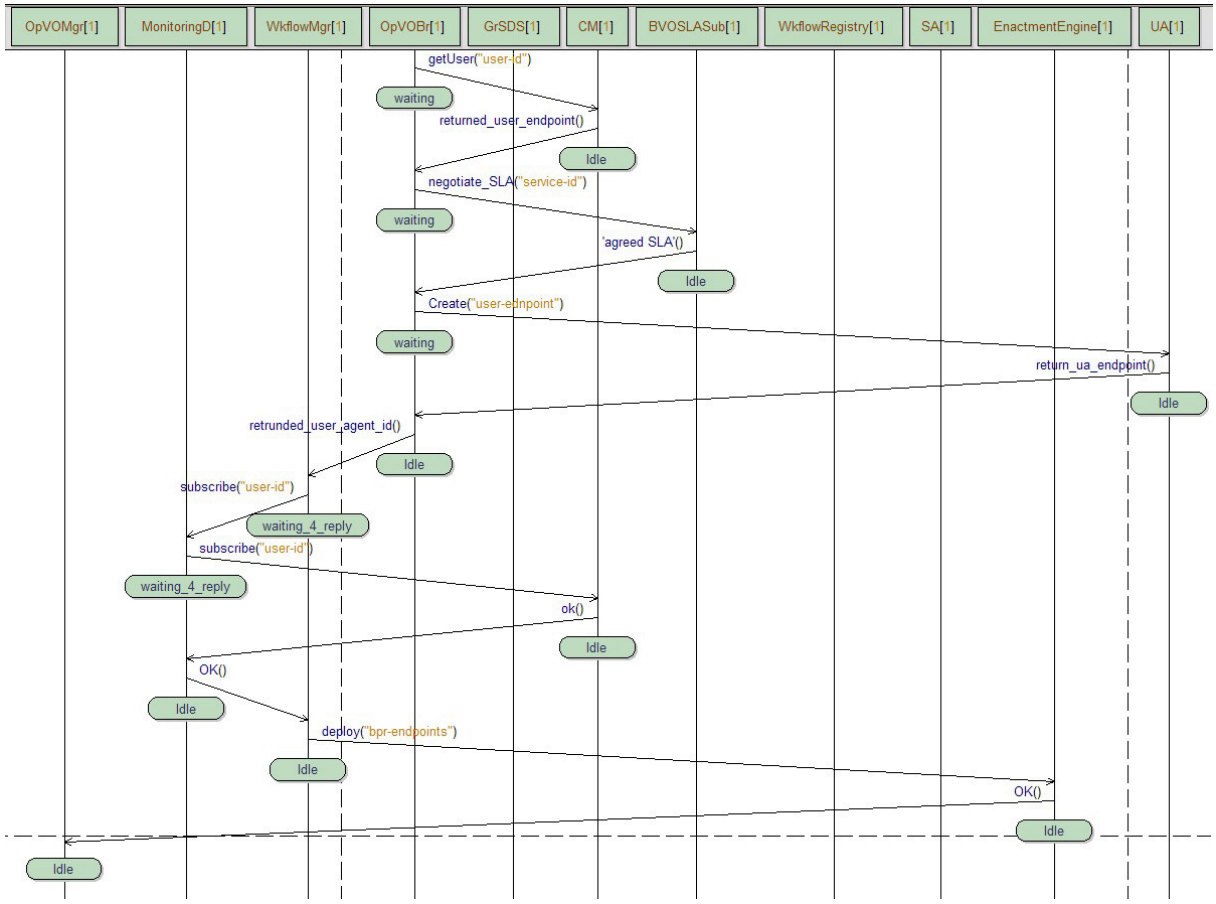

Figure 50: OpVO Creation (3)

4.5. Context Sensitive Services

One of the key features of Akogrimo is that information can not only flow from the service provider to the consumer, but also from the service consumer to the service provider, thereby enabling context aware services that are tailored to the user's needs. Adaptation of service qualities can be done at different levels. E.g. a video service might directly query the context of the user and adapt the delivered video stream to the user's device. More complex adaptations can be achieved by context sensitive workflows. The following scenario demonstrates how this can be achieved with the Akogrimo architecture.

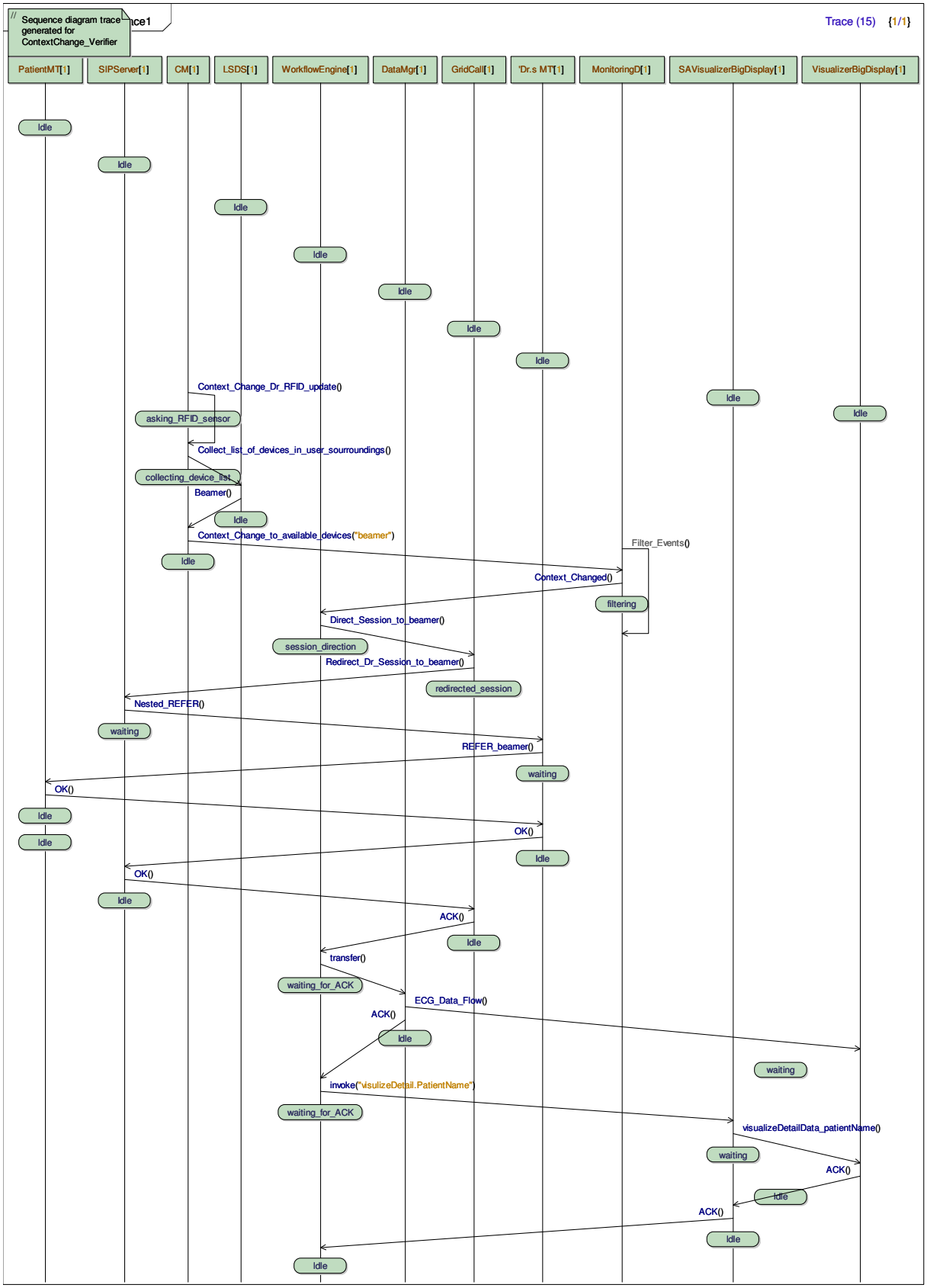

Figure 51: Context Sensitive Workflow

A context change is signalled to the Context Manager by sensors on the doctor's terminal or by sensors that are placed in the environment. The Context Manager then gathers additional information about the user's environment, in our case, by asking the Local Service Discovery Server (LSDS) to provide a list of local services. The context information is sent to the Monitoring Daemon. The Monitoring Daemon has the ability to filter context information and can trigger appropriate actions in the workflow engine. In our case the doctor's audio/video communication with a patient is transferred to a bigger display and the Data Manager is informed to send the ECG data to a visualizing service on that big display.

4.6. Service Deployment

One of our primary goals is to create a dynamically managed enabled Grid. In order to achieve this goal, the Akogrimo platform must be able to support dynamic deployment of web services build on the WSRF specifications on every machine that belongs in its underlying Grid. By doing so the Akogrimo Grid becomes more adoptable and flexible, as it supports the creation of dynamically extensible virtual organizations.

Dynamic service deployment is a task far from being trivial. The shutting down and starting up of the service containers must be avoided if another service is being executed at that time. Restarting the container also results in the loss of all in-memory resources, thus it is of great importance that a machine that receives a dynamic deployment request should be able to fulfill it without interfering with the execution of other services. Any request for another service received during execution of a dynamic deployment procedure should be handled properly. Once a service has been dynamically deployed in the machine, it must become automatically available.

In Akogrimo, the support of dynamic service deployment is intended to be provided through the use of HAND [HAND06], a highly available dynamic deployment infrastructure, based on the Java Web Services Core of Globus Toolkit 4. What HAND claims to provide is capability, availability, and extensibility for dynamic deployment of Java Web Services in dynamic Grid environment. This infrastructure involves two different approaches to dynamic deployment:

Service-level deployment (HAND-S), in which we deactivate one or more existing services, install new services, and re-activate those services — without reloading the whole container.

Container-level deployment (HAND-C), in which the installation of any new service involves reloading (reinitializing and reconfiguring) the whole container.

The importance and the usefulness of each of these approaches can be viewed through different scenarios of their use. Container-level deployment is a good choice when a new machine is added to the Akogrimo Grid, while service-level deployment is required when the deployment of a new service is needed on a machine that already existed in the Akogrimo Grid.

Up to this very moment, the implementation of HAND-C has been completed, while HAND-S is still a prototype, not yet suitable for production use, but still available for experimental use. HAND extensions are not included in the GT4 latest release but will be in future releases (GT4.1+). Nevertheless, a stable implementation of the HAND-C has been merged into the GT code repository and is available to use and experiment upon.

Although from the client's perspective the HAND appears to be a single service, HAND is in actuality a suite of services, each responsible for a particular task in the dynamic service deployment procedure:

Deployer is in charge of invoking the actual deployment actions.

Undeployer is in charge of the undeployment actions.

Reloader is responsible for reloading the container without executing any deployment actions.

Validater is responsible for checking the correctness of the GAR file that is being deployed.

Logging is used to record a detailed log of the execution of the reloading actions.

Restorer is a simple backup mechanism used to restore the container to its previous working state, in case that an error occurs during the deployment.

In the Akogrimo system, the interactions taking place and the steps being followed in order to achieve dynamic service deployment are depicted in the next diagram and will be the following:

The EMS retrieves the GAR file of the service by invoking retrieve operation on the Data Manager.

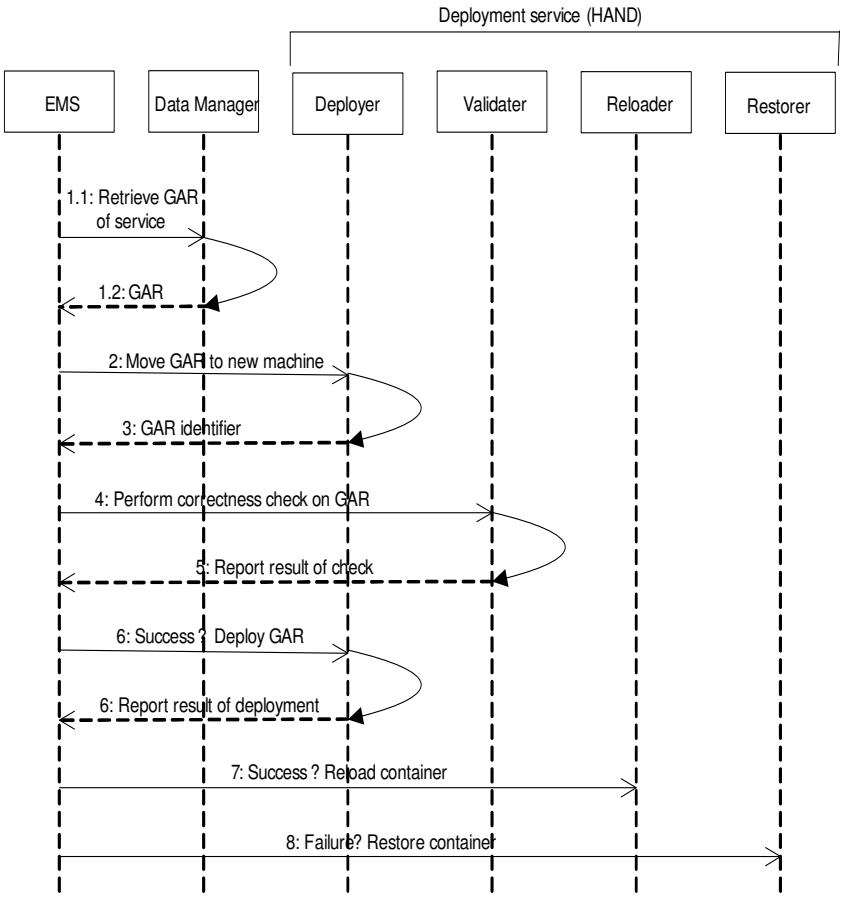

Figure 52: Sequence diagram of Service Deployment procedure

The GAR file is transferred to the new machine either via SOAP using the upload function or via RFT using the download() function of the Deployer.

After the GAR has been successfully transferred to the new machine, the Deployer returns an identifier for the GAR to the EMS.

The EMS asks the Validater service to perform a correctness check on the GAR.

The Validater reports the result of this check.

If the GAR file is valid, the EMS deploys it by calling the deploy function with the identifier obtained at Step 3.

Once the GAR has been deployed, the EMS reloads the entire container on the new machine by invoking the reload operation of the Reloader service.

If an error occurs during the deployment of the GAR, then the EMS restores the container to its previous working state by invoking the Restorer.

4.7. EMS, Service Migration, Monitoring

The Akogrimo system must be able to reallocate resources on the fly to meet the client's needs as expressed in the SLA contracts. In order to address this requirement the cooperation and coordination of various services is needed including the services responsible for the monitoring of the execution as well as the deployment service. The service responsible for the coordination of these services is the EMS. In the following figure, the services involved in this process – called service migration process - as well as the interactions among them are depicted. In this figure the following scenario is illustrated: the Metering service informs the Monitoring service whenever a significant change in the performance parameters of the business service resource that is being executed occurs. The Monitoring service filters this information and communicates it to the SLA-Controller resource that is bound to the specific business service execution. When a violation of the terms expressed in the corresponding SLA contract is detected, the SLA-Controller informs the SLA-Decisor service, which in turn decides on what is the appropriate action to be taken according to the related policy. At this point the EMS checkpoints the properties of the business resource so that the service execution will be resumed and not restarted in case service reallocation is needed. If the execution of the service should be restarted on a different location, the SLA-Decisor contacts the EMS and the EMS in turn contacts the Discovery service asking for suitable service resources. If there is no match but there is a machine that meets the client's needs, the EMS asks the Deployment service to deploy the Business service on the new machine. After the successful deployment of the service, the EMS triggers the Business service to create a business resource on the new machine. As a result the Business service returns the EPR of the new resource to the EMS and the latter sets the state of the new business resource to the last known state of the previous business resource, which it right after destroys. Then the EMS is ready to start the execution of the service on the new machine.

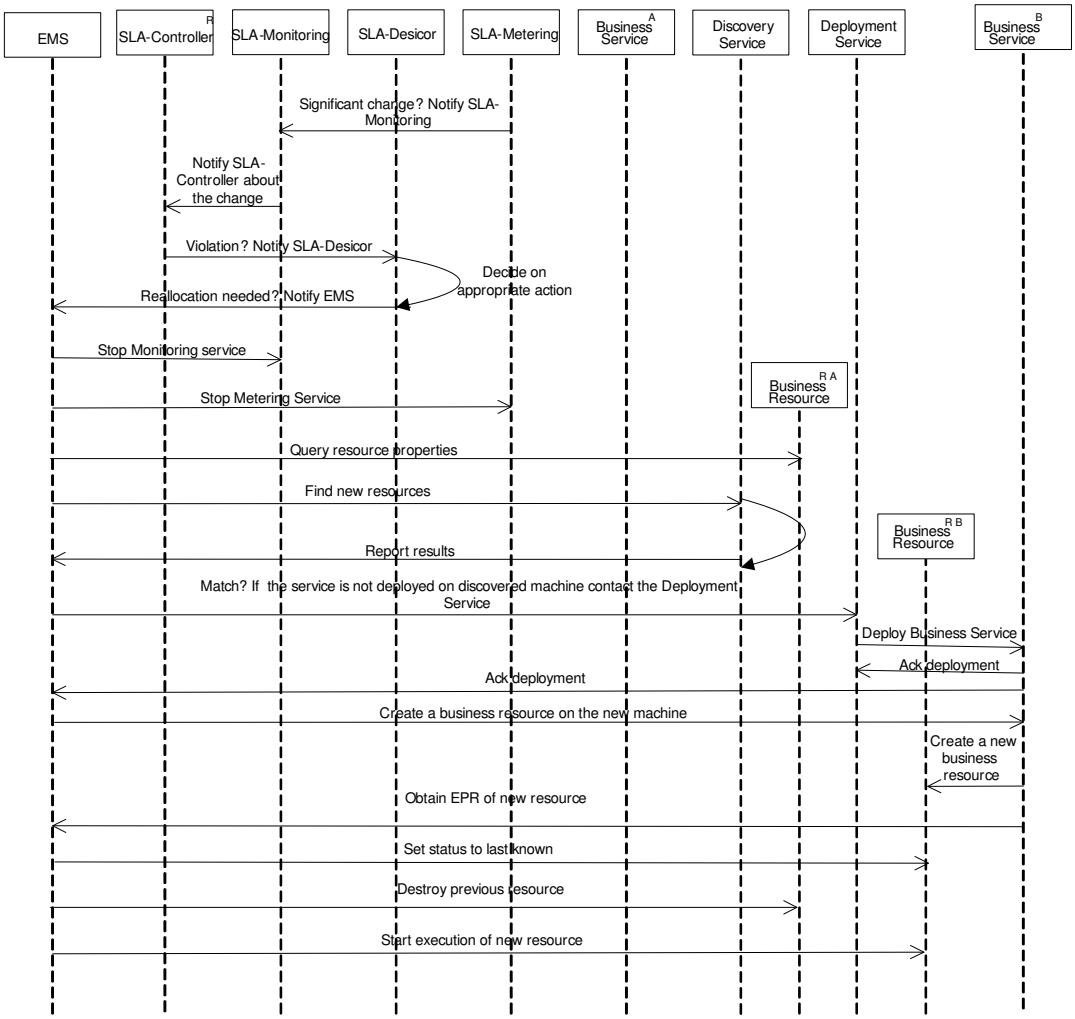

Figure 53: Sequence diagram of Service Migration procedure

4.8. Multi-domain operations

When one user wishes to obtain a service from Akogrimo architecture, it's necessary crosses several different domains, such as: access network domain, core network domain, Service Provider, VO domain, and so on.
The Interdomain Grid Call is a scenario example to explain how Akogrimo project manages the multidomain issue.
In this scenario a patient under treatment needs to be under control and needs to connect an ECG to monitor any possible anomaly while he is not sleeping. He wakes up, connects the sensors to his heart and turns on his MT (e.g. cell phone). The heart beats are constantly analyzed after this. Suddenly the system recognizes a potential hazard and the patient is put in contact by video conference with a Dr who observes the data on his laptop simultaneously. This mechanism is the Grid Call, that is, a call/session establishment between two users (Dr and the patient who belong to two different networks), invoked from Workflow Manager (Grid Part).

The following figure shows the Interdomain Grid Call scenario:

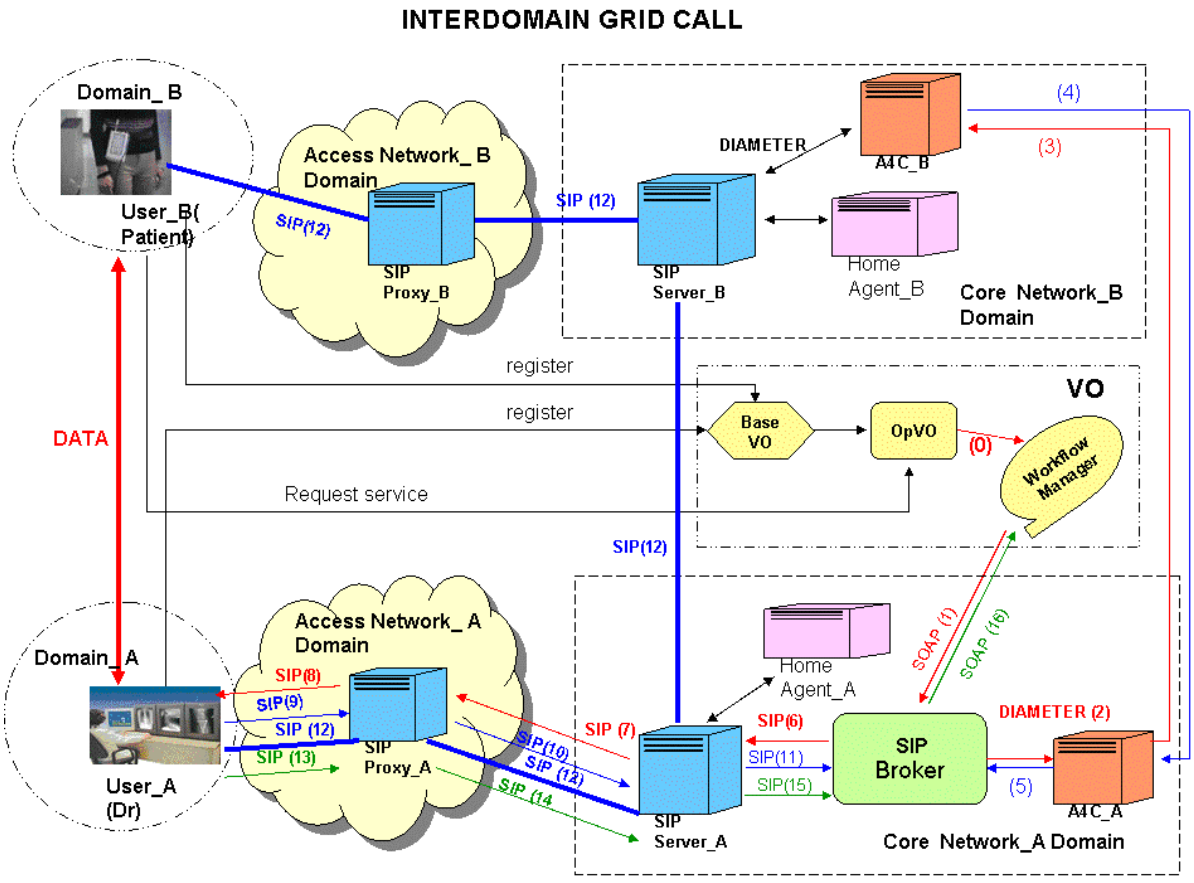

Figure 54: Interdomain Grid Call

The following preconditions are considered, in this figure:

- Both patient and doctor should be registered in the BaseVO
- SIP User Agent (of the SIP Broker) and (at least) one of the user terminals involved support SIP REFER. So this is a requirement for Akogrimo SIP terminals initiating grid calls.
- User_A (Dr) MT and User_B (patient) MT successfully registered (which means successfully authorised to make use of Akogrimo SIP infrastructures).
- When the User_B (patient) requests service on the OpVO, we are assuming that before the User_B has connected to the network and then he has been authenticated and authorized (from the NP viewpoint). When he has network access he will request the service, then in order to authenticate the invoker, the OpVO will contact the NP domain. Of course, if the OpVO receives a positive response it will pass to authorize the invocation with respect to the internal policy of the OpVO
- The User Agent is created by the Workflow to represent a particular user to the workflow. The user "connects" to the User Agent from a particular MT (authentication, selection of OpVO, authorization, so on). At this point, the user MT can be given its `OpVO_token`. It can then use this in all outgoing P2P communications, putting it in the "init call" request, so the receiver can be sure the call is authorized.
- The Workflow Manager will include the OpVO_token in all requests. This `OpVO_token` is passed through to User_A MT (Dr), then it can check that the call is authorized
- Patient UA triggers the Workflow Manager.

Taking in account the preconditions before, the scenario step-by-step is as follows:

1. The SIP Broker receives one SOAP(1) query from the Worflow Manager in order to put in contact two Akogrimo users (Doctor and the patient)
2. The SIP Broker retrieves the default SIP URIs using DIAMETER protocol with the A4C_A. When A4C_A receive the request it checks that one address requested (patient address) belong to another network domain, so, the A4C_A should contact with A4C_B, in order to obtain the default SIP URIs of the patient. (3-5)
3. Once the SIP Broker has obtained the SIP URIs, it sends a SIP REFER message(6) to User_A (Dr), indicating that a session with User_B (patient) should be initiated. The SIP Server_A maps the default SIP URIs, to the current SIP URIs, and redirects the REFER message to User_A (Dr) MT (7-8).
4. If User_A (Dr) accept the REFER, then, it sends a 202 Accepted message to the SIP Broker (9-11)
5. The sequence (12) is a general SIP *session setup between two users (Dr and patient) in different networks (network_A and network_B).*

   Once the User_A(Dr) receives the request to initiate one SIP session with other User_B (patient), the first he has to do is checking the `OpVO_token` included in the SIP request. If this OpVO_token is valid then he initiates the SIP call setup, since both users (Dr and Patient) belong to the same BVO. Otherwise, User_A, rejects the SIP request, and does not initiate the SIP call.

   If the `OpVO_token` is valid then User_A (Dr) starts the general interdomain SIP session setup. He requests resources to the QoS Broker in domain_A and if there is enough available resource, then it sends the INVITE message to its SIP Server_A

   When the SIP Server_A has validated the credentials presented by the User_A(Dr) in the INVITE message, it should inspect the "Request-URI" header of the INVITE message in order to determine how the message should be routed. In this instance the "Request-URI" designates a remote domain (Domain_B).

   The SIP Server_A at Domain_A should then establish a TLS connection with the remote proxy server SIP Server_B at Domain_B. Since both of the participants in this TLS connection are servers that possess site certificates, mutual TLS authentication should occur. Each side of the connection should verify and inspect the certificate of the other, noting the domain name that appears in the certificate for comparison with the header fields of SIP messages. Once it has done so, and TLS negotiation has completed, resulting in a secure channel between the two proxies, the SIP Server_A can forward the INVITE request to Domain_B, in order to carry on with the session setup process.

   On the other hand, the receiver user (that is User_B, the patient), also can check the `OpVO_token` included in the SIP request. So, if he detects the OpVO_token is not right he rejects the SIP call initiation request.
6. Once the session is established, User_A(Dr) notifies the successful of the SIP session setup to the SIP Broker, and this one to the Workflow Manager (13-16).

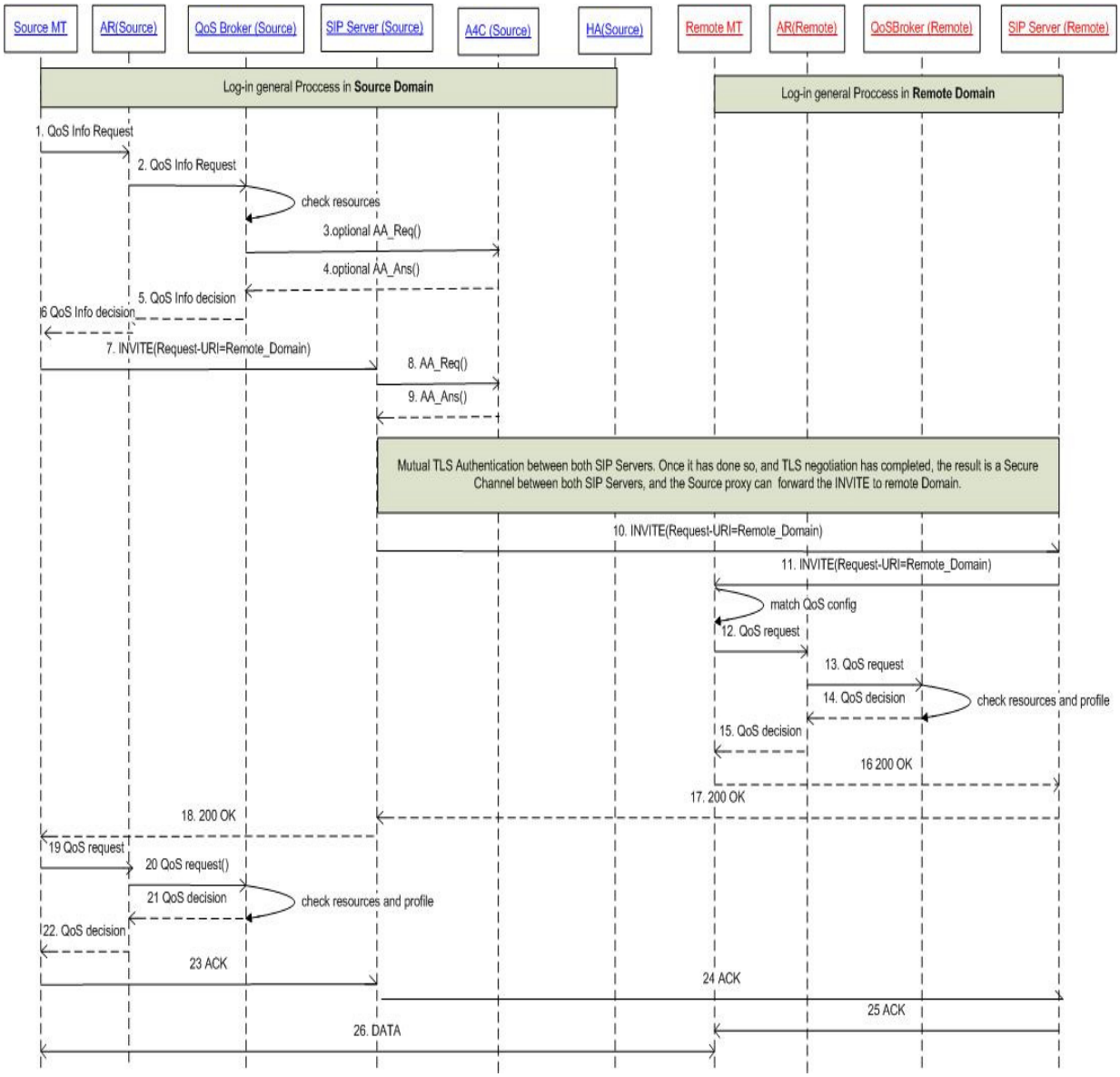

Figure 55: Multi-domain Scenario

5. Conclusion and Outlook

This Deliverable should be regarded as a baseline document for the whole Akogrimo project, with the assumption that the presented architecture is for the time being final, but will be enhanced in subsequent activities. The Deliverable summarises the work during the first 24 months of the project and includes a description of generic operational scenarios with the key goal to visualize based on selected scenarios the interaction of the components generically described in chapter 2 and chapter 3. These examples are of course not considered to be complete and could be easily extended upon special request. The examples are – as currently described – considered by the Akogrimo consortium as the key sequences helping the reader to understand the "cross-layer" interaction, which basically means how Grid concepts and technologies interact with the network layer. .

The Akogrimo Architecture has three goals which are orthogonal: First, the Internet Protocol is regarded as convergence layer for a mobile Grid infrastructure, which relies on Internet-like AAAC concepts for commercialising the composed services. This requires a lower layer technology independent solution with the goal of providing seamless mobility compared to that of existing networks in a way grid based resource management entities can seamlessly incorporate such a network in their service compositioning process. So key goal here was not to design a net Grid layer, but to converge the network layer towards "grid-compliance". Second, Akogrimo will manage virtualised resources of the network layer efficiently. Third, the whole Akogrimo concept will be targeted towards a commercial environment and all the required mechanisms will be added around the IETF AAA concept. This instead required an adaptation of traditional grid accounting principles towards the IETF AAA concepts.

This Document describes the third iteration loop of the Akogrimo Deliverable and is based on Akogrimo Deliverable D311 and D312, but with refinements represented in chapter 2 and 3 and some extensions in chapter 4.

6. REFERENCES

[Dey01] A. K. Dey, Understanding and Using context, Personal and Ubiquitous Computing Journal, Volume 5 (1), 2001, pp. 4-7.

[D2.2.4] J. Gallop et al: "Report on the state of the art", Akogrimo Deliverable D2.2.4, v1.1.

[D4.2.1] J. Wedvik et al: "Overall Network Middleware Requirements Report". Akogrimo Deliverable D4.2.1, v 1.1. URL: http://www.akogrimo.org/modules.php?name=UpDownload&req=getit&lid=39

[D4.2.2] P. Osland et al: "Final Integrated Services Design and Implementation Report", Akogrimo Deliverable D4.2.2, v1.0, URL: http://www.akogrimo.org/modules.php?name=UpDownload&req=getit&lid=42