The Akogrimo Mobile Grid Reference Architecture - Overview

Editors
Jürgen M. Jähnert[1], Stefan Wesner[2] , Víctor A. Villagrá[3]
[1]*Rechenzentrum Universität Stuttgart*
Email: jaehnert@rus.uni-stuttgart.de
[2] *Höchstleistungsrechenzentrum Stuttgart*
Email: wesner@hlrs.de
[3] *Univ. Politécnica de Madrid*
Email: villagra@dit.upm.es

This document is a summary of the Akogrimo Overall Architecture and is based on the public deliverable D3.1.3 The Mobile Grid Reference Architecture which is also available for download from the Akogrimo website at http://www.mobilegrids.org.

This complex and long document has been summarized with the help of all partners in WP3.1 to this shorter version aiming to provide to technical experts an initial understanding of the basic concepts and architecture decisions taken in the Akogrimo project. For a more application, business or fundamental overview the reader is directed to the other white papers.

This document outlines consequently the fundamental concept of a Mobile Grid as understood by Akogrimo, introduces the fundamental concept such as the Base and Operative Virtual Organizations and outlines the major building blocks. A special focus is given on security considerations and the document ends with a list of documents recommended for further reading.

SIXTH FRAMEWORK PROGRAMME

PRIORITY IST-2002-2.3.1.18

Grid for complex problem solving

1. Introduction

The Akogrimo project is driven by the basic idea that Next Generation Grids (NGG) should be integrated with Next Generation Networks. An Akogrimo NGG must be able to address the needs of a volatile environment with fast changing context such as bandwidth, device capabilities, location, a variety of competing access network providers and global and local service providers.

Consequently Akogrimo assumes a pure IP-based underlying network infrastructure. As a result the Akogrimo architecture can be immediately deployed in Unlicensed Mobile Access (UMA) environments 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 similar network technology.

The basic concepts and terms required in addressing this challenging task are summarised in this document. The major identified research challenges are identity and security management, cross organisational accounting, the handling of context and the interrelation of signalling as basic protocol element.

Akogrimo does not consider the network as a pure transport mechanism for data or control messages but aims on one hand to avoid the duplication of functionality but more important to enable a cooperation between applications, grid and network middleware and the network. The goal is to reach a platform combining the functionality of next generation networks, service and knowledge oriented services that are able to provide a general purpose added value services provisioning platform not limited to a certain area such as conversational services. We see a general purpose and *open* platform enabling dynamic and ad-hoc collaboration addressing beside audio and video communication at the same level also the provision of data, knowledge and computational services as the key element of success. The Akogrimo platform aims to provide business opportunities for different stakeholders and enable the dynamic composition of complex services.

So in summary the key differentiators of Akogrimo compared to existing Grid solutions are:

- Support for Dynamic Virtual Organisations (VO) that may be even built ad-hoc e.g. by including local service providers
- Enable adaptive and context aware applications reacting on the changing conditions of the VO members such as bandwidth, device capabilities or e.g. RFID based events
- Provide a smooth integration of multimedia communication with application services
- Prevent the duplication of functionality in the Grid middleware layer by leveraging the functionality anticipated in the next generation of networks such as Identity Management or Cross-Organisational accounting.

2. The Generic Akogrimo Architecture

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

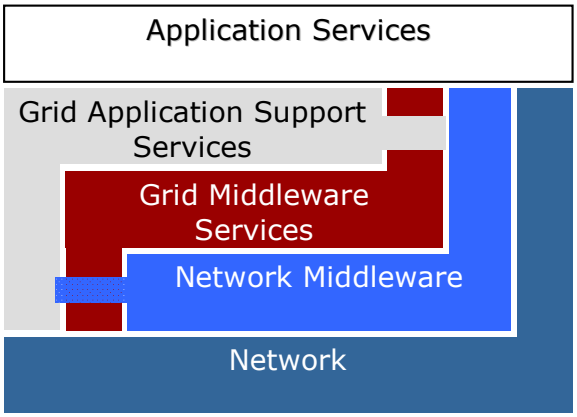

Figure 1: Akogrimo Conceptual Architecture

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 such a strict layered approach was considered as inappropriate to describe the approach and has been kept as organisational "area".

The lowest "area", the *network* interfaces with all the other areas, but has most interaction with the *network middleware* providing network services like Authentication, Authorization, Accounting, Auditing and Charging (A4C), 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 Virtual Organisation (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 as a (web) service. And the role sharing is more function based rather than a protocol stack.

A further major principle is the bi-directional exchange of all information between these "areas" such as identity, context, etc…, which will be explained in more detail within this document.

2.1. Basic Building Blocks

The generic Akogrimo architecture has been initially introduced in D3.1.1 and refined in D3.1.2[1]. In these deliverables, a layered approach has been presented. This section consolidates the information of the previously provided deliverables on a generic level. Here and in the final version of the architecture D3.1.3 The Mobile Grid Reference Architecture, the strict layered approach has been given up.

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 Operative VO (OpVO), the Base VO (BVO) and the Customer domain.

[1] These documents including the most recent version D3.1.3 The Mobile Grid Reference Architecture are available from the Akogrimo website http://www.mobilegrids.org

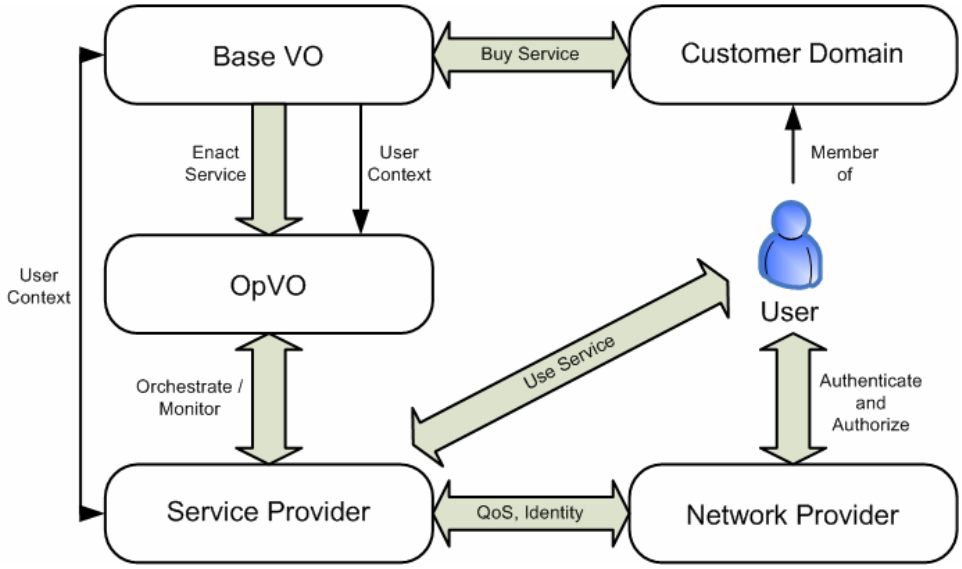

Figure 2: Akogrimo Building Blocks

2.1.1.Base VO

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

The interactions with those building blocks are 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:

- The 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 due to its 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.
- The OpVO building block is a component derived from the BVO. The BVO interacts with the OpVO creating it and configuring a dedicate subdomain for its execution.

2.1.2.Operative VO

The Operative Virtual Organisation (OpVO) is a building block that fulfils an important role at execution time when the application is actually delivered to the customer. The OpVO is the "run time environment" for Akogrimo applications. The OpVO is built from a subset of members of a BVO. Each BVO can create several, even overlapping, OpVOs. The OpVO can be seen as a particular instance of a VO.

Similar to the BVO, the OpVO interacts with all the building blocks identified in Figure 2

- 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 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 result in the agreement 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.

2.1.3.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 herself (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.

2.1.4.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.

2.1.5.Network Provider

On the Network provider domain there are all functions represented required to provide network services in a Grid-compliant manner. This makes this domain unique to Akogrimo as basic network functions – as currently provided in other architectures as well – are so far not described in a Grid/Web Services compliant way. The available accounting and charging functionality in this domain is used to account and charge for Grid services in a similar and integrated way as done for the usage of network resources. This 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 Single Sign On (SSO)-based services for Grid services **and** network services assuming 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.

Finally, network management issues contribute to overall policy-based management concepts covering all "areas".

3. Key Akogrimo Concepts

3.1. 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. 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. 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 Operative Virtual Organization (OpVO).

3.1.1.Base Virtual Organisation

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 Operative VO when a business process is instantiated and executed.

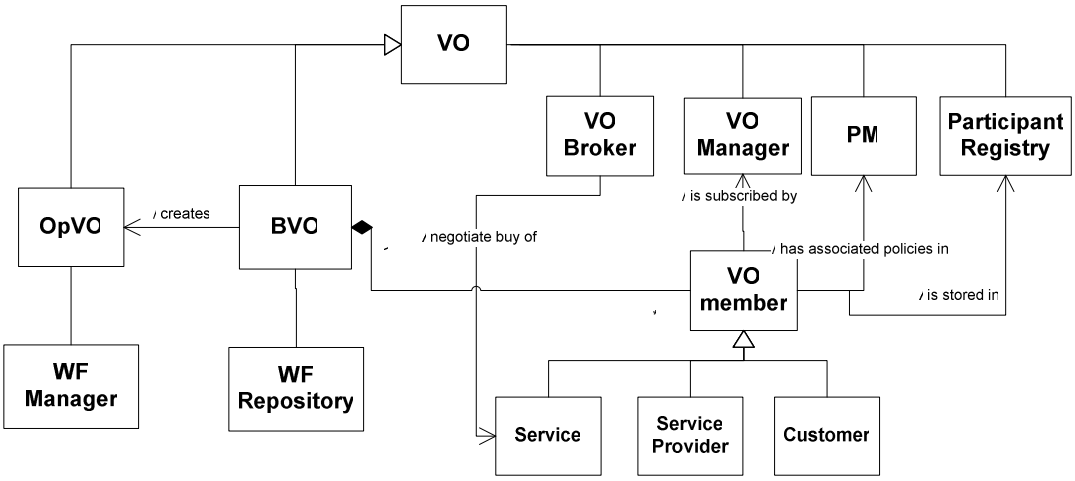

Figure 3: The Akogrimo Base VO concept

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)

Apart 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:

- Publish available services in "yellow pages" registry of the BVO.
- Buy services by searching them among the ones available in the "yellow pages".

In order to allow the usage of the bought services, the BVO will create an OpVO that allow interactions between buyers and providers.

3.1.2.Operative Virtual Organisation

The purpose of the Operative VO is to instantiate a business process and to manage its execution. The BVO has the role of initiating the OpVO creation process. During the creation phase of an OpVO, all the dedicated management services are instantiated (the ones associ-

ated to a general VO), and they will live until the OpVO is terminated.

This management services are:

- 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.
- The WF Manager is a management service specific for OpVO. It manages the instantiation of a WF and its correct execution.

All these services will be destroyed when the OpVO is terminated.

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

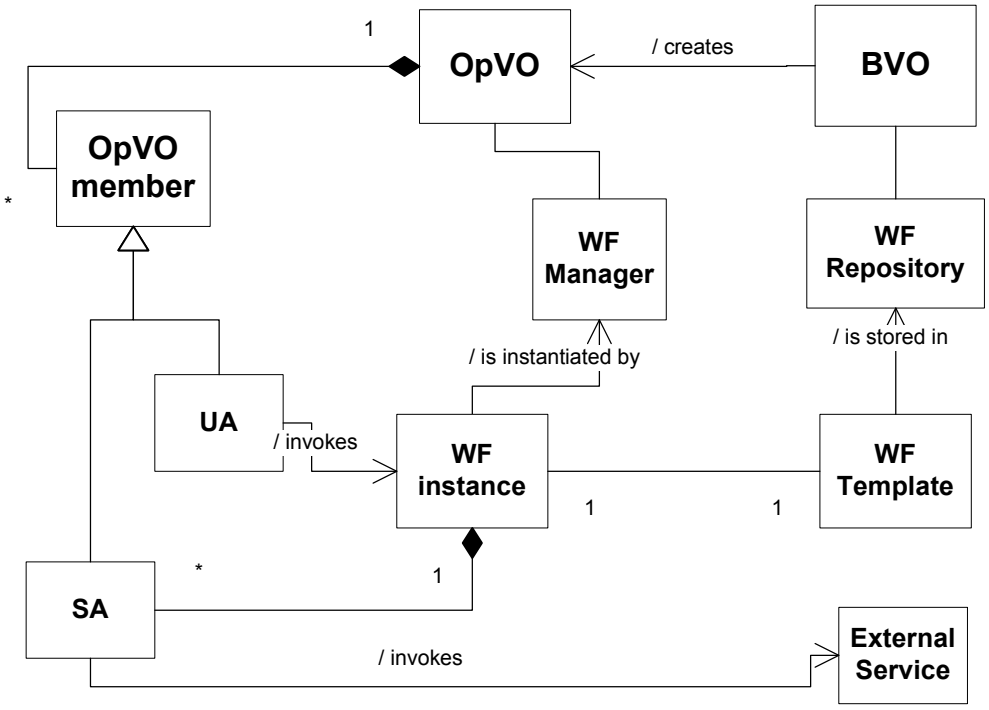

Figure 4: The Akogrimo Operative VO – Components and concepts

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

3.2. Business Process and Workflow

The Akogrimo architecture is designed for business processes supporting and taking 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 Business Process Execution Language (BPEL) 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 (they know what they need to achieve and just deliver it)

Akogrimo intends to support 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.

3.3. 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.

Terminal Mobility, as the name implies, relates to the mobility of the device (or terminal) the user utilizes for accessing the network.

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.

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.

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.

3.4. 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.

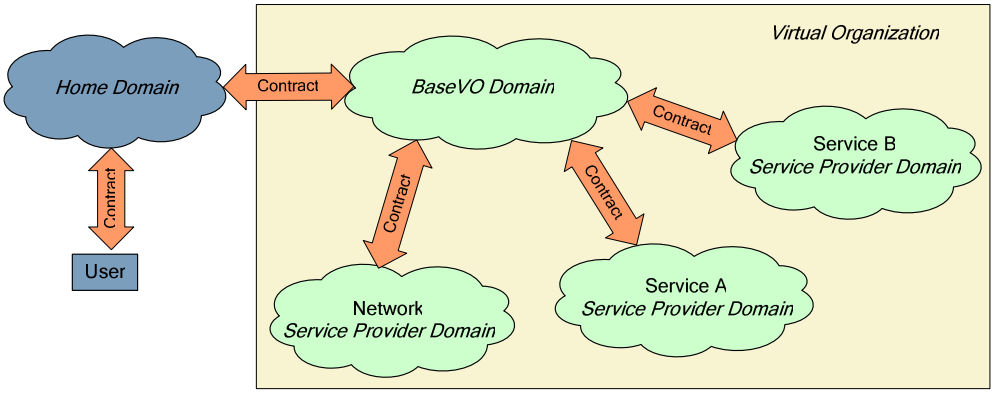

Figure 5: Example of trust establishment

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

3.5. The Akogrimo Identity Concept

The first efforts related to the identity management topic in the Internet were in the direction of the X.509 certificates. But they are not flexible, heavy weighted and they can not be used to change identity according to the user's needs. 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. In this way, the user and/or the organization present one single certificate with the same identity 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.

With time, the Grid community realised that their certificates do not scale well and that many organizations already offer federated identities for their users. 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.

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. The personal user information and home domain specific attributes should be stored inside the home domain of a user. The subset of attributes containing service-specific information 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. The identity profiles (the user can have several identity profiles for one or more VOs) are stored at the Participant Registry of the VO the user subscribed to.

It is desired in Akogrimo that the user can choose the attributes (excluded demanded attributes) he wants to share within a VO. The personalization will be done by the user, at the BVO subscription in a very straightforward way.

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. Attribute 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 ex-

changing 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.

Although the user has multiple identities, for different VOs and domains, the ability of Single Sign On is still given. 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.

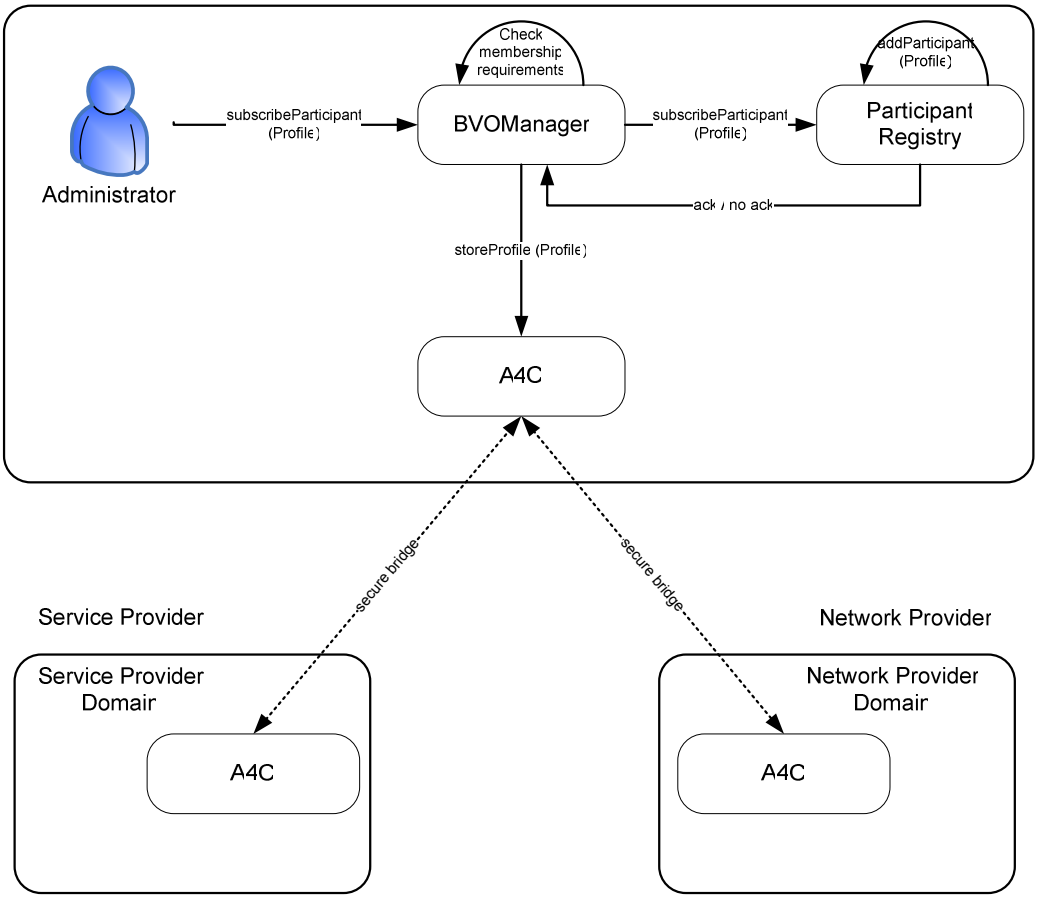

Figure 6: Identity Model

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, simplifying 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.

3.6. Context awareness

Context is any information that can be used to characterise the situation of an entity, primarily a user. Context awareness means that devices, applications and systems have information about the circumstances under which users are operating and can adapt behaviour to be optimal for the current situation. In this way context-aware services will be an important enabler for pervasive and ubiquitous computing; allowing the user to solve his/her tasks while devices and technology fade away into the background.

The vision of Akogrimo involves context-aware mobile Grid services. 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.

The definition of context is open-ended; the set of data that should be monitored will depend on the application domain. Common scenarios in Akogrimo include eHealth, eLearning, and Crisis Management. For these scenarios we are interested in keeping track of user context in terms of

- Presence: Is the user logged on? Is he idle or busy?
- Physical properties: User location, local time, body temperature etc.
- Environmental information: Temperature, humidity, weather
- 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? What are the display capabilities (display resolution, screen size), and network capabilities (bandwidth, connection type).
- Local services: What services are found in the proximity of the user?

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. A general framework for context handling is shown in Figure 7, where Context Manager (CM - part of the Akogrimo Network Middleware) is the central component. The following actions take place:

1. Applications subscribe to context information for a specific user. These applications are located in the Generic Application Services Layer or within Domain and Application Specific Services.
2. CM will request raw context data from context sources, which may be the users themselves, the user terminal, sensors, or location technologies.
3. CM will refine, filter and store current context for each user which has a pending subscription.
4. Updates are distributed to the requesting applications. Distribution may take place

when the context for a person changes, or in appropriate intervals.

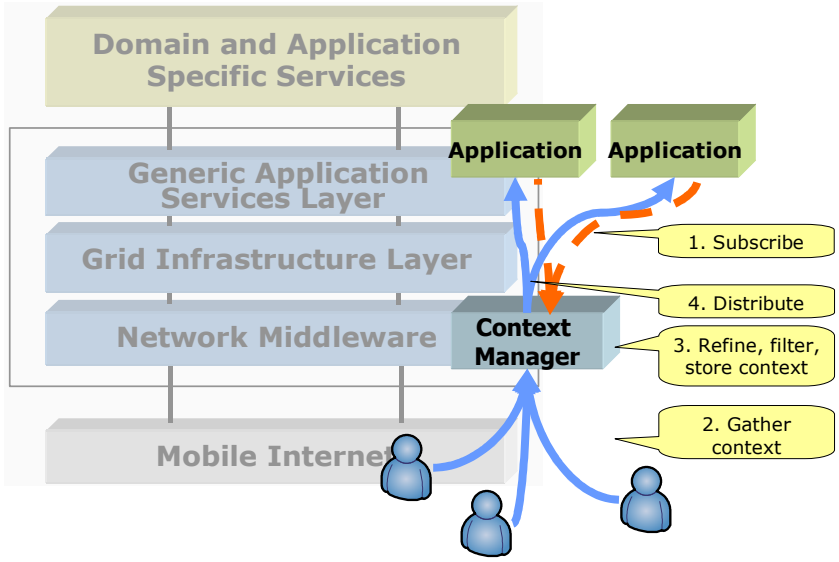

Figure 7 Context Manager in the Akogrimo Architecture

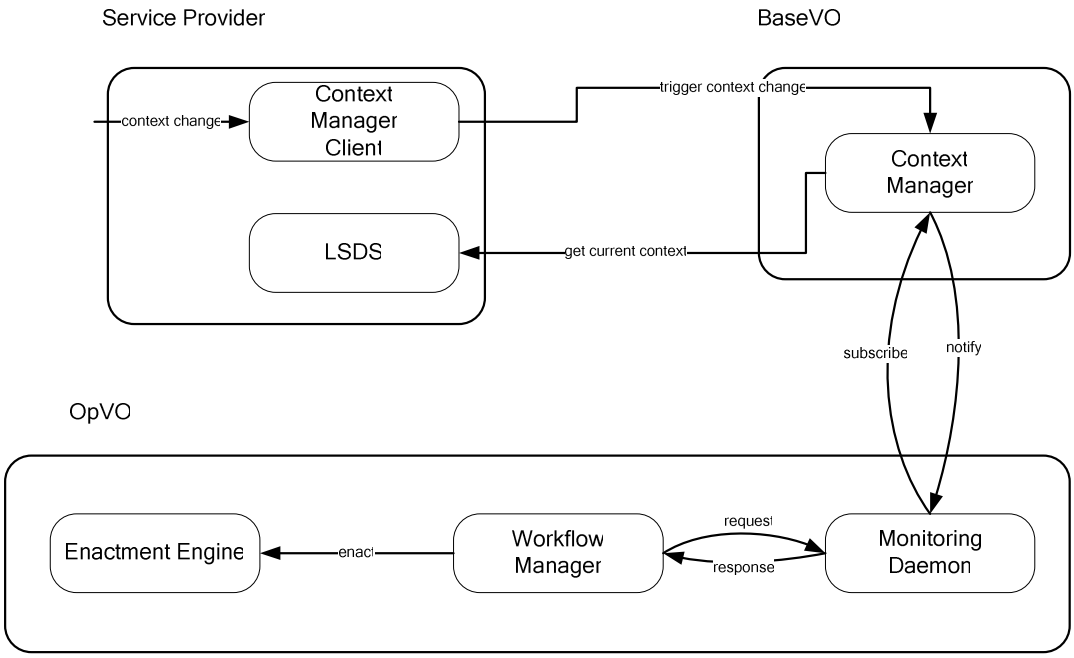

Figure 8: Context flow

In Figure 8 we show the location of the various components within the Service Provider domain, Base Virtual Organization (VO) and Operative VO. LSDS – local service discovery server – contains overview of services and devices that may be related to the user's location, and thereby is part of his/her context. 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.

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[2].

3.7. Policy Management

Within a Web Service framework and in Network Management, policies can be described in the form of a policy language and may be applies to diverse domains, including VOs, SLAs, and A4C. A policy description can be created by software or by a person, updated, communicated, interpreted, used as a basis for making decisions and enforced. Policies are a means of specifying and influencing management behaviour within a distributed system without coding the behaviour into the constituent components.

Policies need to be applied and a distinction is made between:

- a policy enforcement agent (or Policy Enforcement Point, PEP) that is required to stop, permit or trigger an action on a given resource or service in a specific context; enforcement may result in certain operations being prevented by a VO, depending on policy; or for QoS it may result in configuration commands to routers so that the balance resources offered to traffic classes are shifted.
- a policy decision agent (or Policy Decision Point, PDP) that takes the context and requested action and returns a decision on whether the action may be carried out or not;
- and the Policy Manager which is responsible for the creation and updating of a policy repository and for communicating policies to the PDP on its request.

This distinction allows the possibility of each being in different places in the distributed system or in the same place if the system design requires it. As the split between enforcement and decision is potentially a pure overhead on every action, the system must be designed to make this process as efficient as possible.

Although there is a generic language for describing policy (WS-Policy), current practice is that languages actually used are specific to policy domains and this is followed in Akogrimo, for instance:

- SAML for security policies,
- WS-Agreement for SLAs,
- CIM for PBNM, including A4C and QoS, for example *"If the EF traffic load exceeds 70% of the allocated bandwidth, then allocate more bandwidth to the EF traffic taking it from other classes."*;
- WS-Policy for several aspects of Grid middleware, for example "*do not transfer data between two specific databases if the quantity to be transferred is greater than xGB*".

In all cases the policy languages are based in XML, thus providing a framework for transforming where required, and Akogrimo employs a Policy Manager to cover aspects of Grid middleware that adopt WS-Policy. At this level, it may be possible to move towards a coherent Mobile Grid policy framework, making them

[2] As mentioned before check http://www.mobilegrids.org for public versions of the Akogrimo deliverables.

converge. An implication would be the joining of a PBNM Server to a WS Policy Manager. However at present, the path followed is coexistence, rather than convergence.

Policy in the field of Web Services has developed separately from developments in network management (PBNM), but nonetheless the ideas of PDPs and PEPs are similar in both fields. In a mobile grid, the relationship of SLA and QoS presents an example where WS policy and PBNM do need to interact.

While the policy language used for the PDP may be close or identical to that used by the policy repository, the PEP may employ a language closer to configuration commands.

Policy violations generally require corrective actions. Services supporting an action or resource are expected to provide remedial actions for all potential policy violations. If the obligation isn't rectified, the failure is logged and the failure is escalated to higher authority. In the case of an SLA, a violation is handled by a different service from the failing one, and this effectively rewrites the policy to permit the violation to continue, possibly with the application of penalties.

More details can be found about policy frameworks in the full document on the Akogrimo architecture (D3.1.3), on the use of policy in Grid middleware (D4.3.2) and on the use of PBNM in network management (D4.1.2).

3.8. Discovery

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 reconfiguration 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.

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.

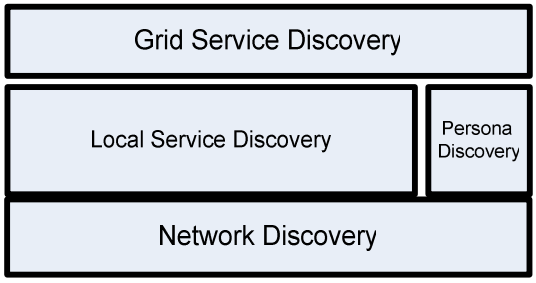

Figure 9: Network Discovery subsystems

- 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 network configuration problem is solved at IP level, based on a Zeroconf solution, since layers underneath Layer three 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 going to be 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. E.g. 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.

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

Generally, the A4C infrastructure in combination with the identity model plays a very central role in the Akogrimo security activities. Especially the authorization of service usage, which can depend on several pieces of information, has been considered. E.g. user's identity attributes may include a company identifier, provided by the A4C, and the current location, provided by the context. The authorization 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 authorization 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 10 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.

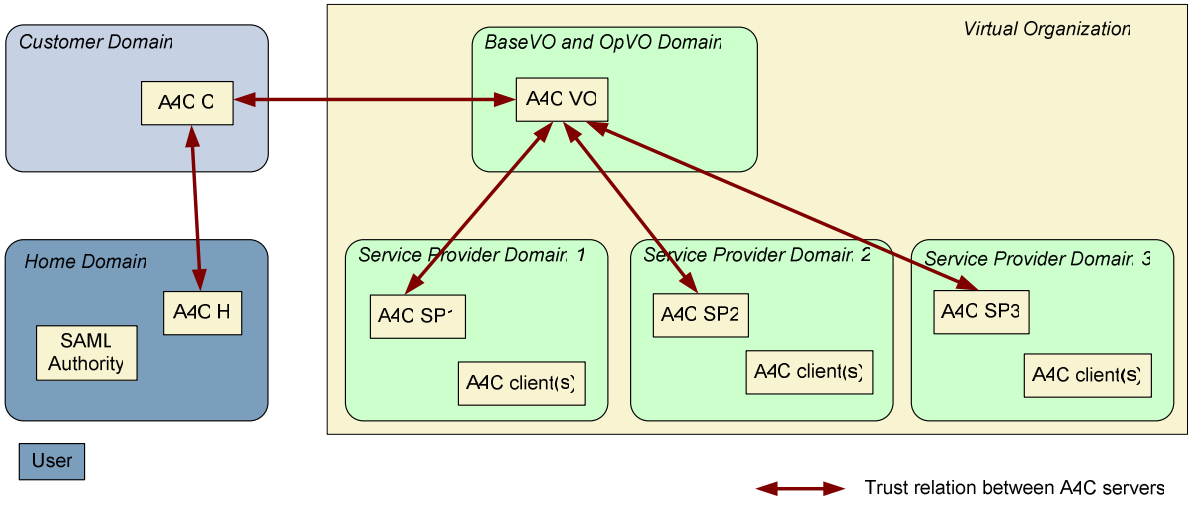

Figure 10: A4C infrastructure

The A4C servers maintain trust relations with other servers as shown in Figure 10. 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

The multi-domain service provisioning aspect highly impacts the way accounting and charging for composed services is done. A MDVO groups together several providers that agree to share their resources for creating more complex, value added services. In order for such a service provisioning approach to become 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.

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 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 11 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/service provider domain an A4C server collects service usage data in form of accounting records and generates session records. Session records can include accounting records and charging records, depending on the business model and contract between the providers.

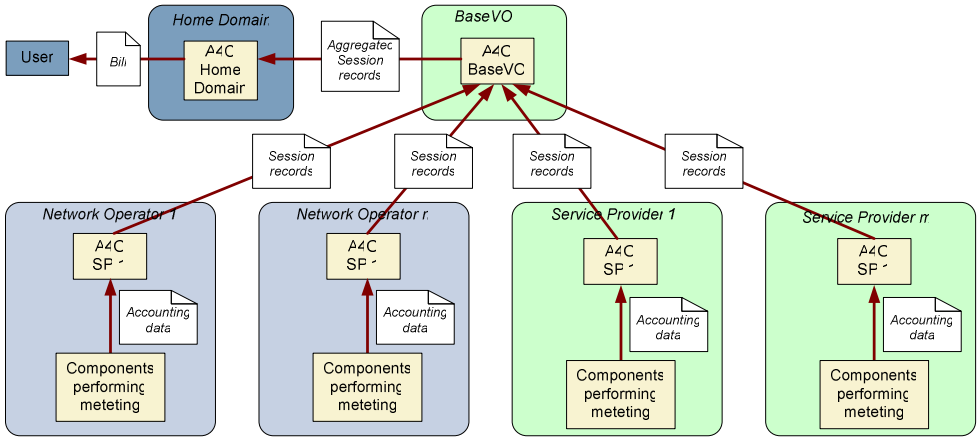

Figure 11: Accounting and charging data flow

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 11), 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.

4. 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 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).

4.1. 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.

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

4.1.1.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 1 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 1: QoS Bundles

4.1.2. 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 A4C 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

4.1.3. 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 us-

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

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.

4.1.4.Grid Application Support services

At the Grid Application Support Service (GASS) 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 semantic description will be put in a metadata file. 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 metadata file will be associated for each service to be involved in the workflow execution. This metadata 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.

4.2. Quality of Service Provisioning

In an environment with mobile users, ensuring quality of service is a complex task. 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.

QoS 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. After the SLA contract (see Figure 12) 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.

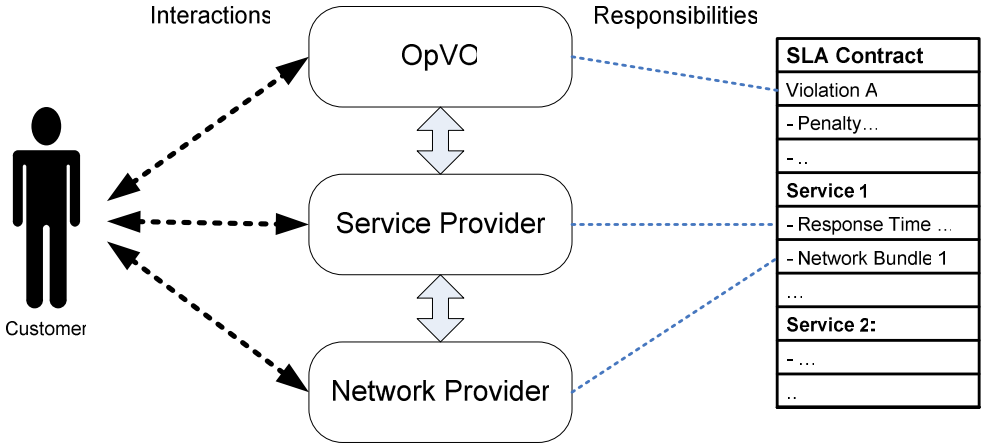

Figure 12: Using a Service

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. 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 Operative VO level. The situation is then handled according to the business process definition.

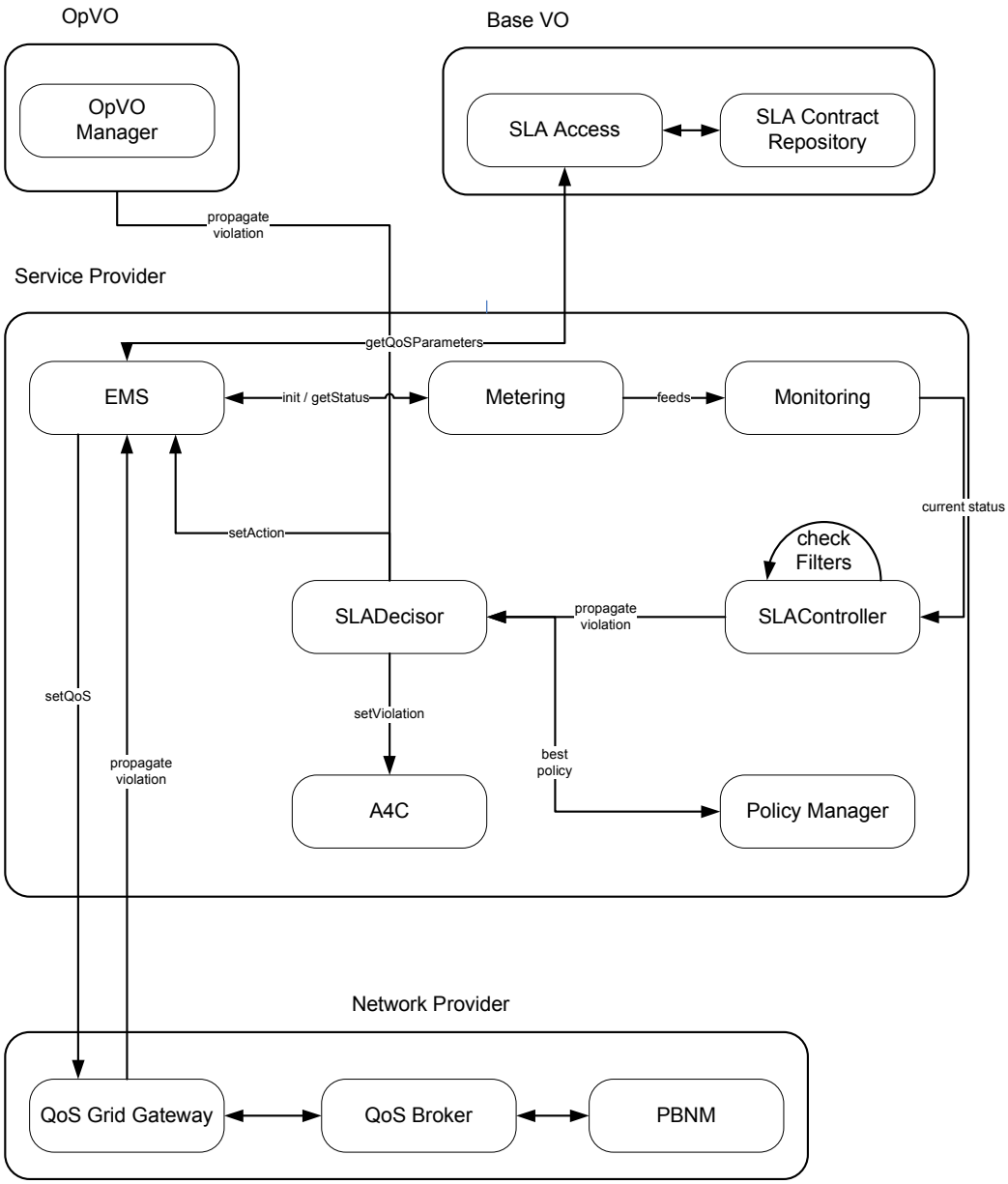

Figure 13: Quality of Service in Akogrimo

4.3. Session Management: SOAP with SIP interaction.

The integration of Telecommunication infrastructures and Grid based Systems needs an integrated signalling framework which has been introduced in Akogrimo, based on the session concept. The session concept in Akogrimo makes possible to link certain events, actions and service consumptions together, providing also some facilities related to sessions, like pausing/resuming, session establishment with ubiquitous users, session transfer, etc. There are two types of sessions in Akogrimo:

- Grid-sessions, which are maintained by the Execution Management System (EMS) with the resources of the OpVO during the OpVO lifetime.
- User-sessions which are maintained by the users with the specific services they are using.

Akogrimo has adopted a unified signalling framework based on the Session Initiation Protocol (SIP) network protocol, which has been traditionally used for the establishment and management of multimedia sessions. It works by enabling Internet endpoints to discover one another and to agree on a characterization of a session they would like to share. Each endpoint should send to an adequate SIP Server a *SIP Register* for registering in a SIP Server their current location/available features and a later *SIP Invite* for locating prospective session participants and forwarding them the session invitation.

One of the most important challenges in Akogrimo is to integrate the Grid environment with the Mobile environment, where end users and services can move from one device to another one, or new devices can appear/disappear. For multimedia data sessions between user and services this will be done in the traditional way using SIP. But for the integration of the Grid-session with this signalling framework, several proposals were analysed: SIP over SOAP, SOAP over SIP and SIP with SOAP. Finally Akogrimo adopted the "SIP with SOAP" approach.

In this approach, the EMS is going to create an OpVO by contacting a number of ubiquitous services, which have been previously registered in a SIP-server. So it sends a SIP-invite to the services URI, and the resources are answering including in the SIP answer payload the details needed by the EMS in order to build the EPR for it to create the services. In order to simplify the EMS design, it will use SOAP to contact a broker module (SIP Broker) which is responsible of sending the SIP invites on behalf of the EMS.

Additionally to establish Grid Sessions, the SIP Broker is offering some advanced functionality to the Grid modules:

- Use the SIP mechanisms to invoke the establishment of a specified session between two actors.
- Use the SIP mechanisms to invoke the transfer of a previously established session between two actors.
- Using the presence and context management mechanisms established by the SIP SIMPLE infrastructure.

The following figure shows the "SIP with SOAP" approach:

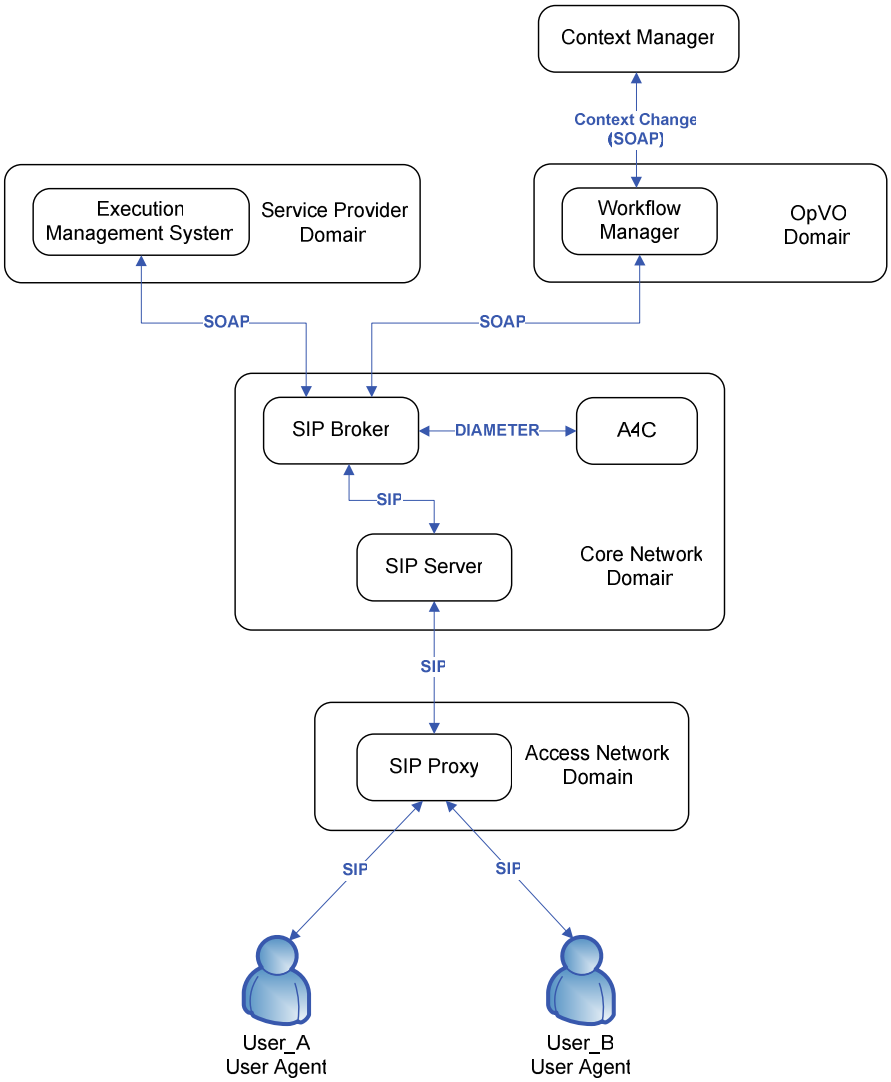

Figure 14: SIP with SOAP approach

In this scenario, two main interactions are shown:

- 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. Once the Workflow Manager receives the context change then it decides to invoke 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..

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 resources identification, should use - in the establishment phase - SIP-URIs to identify mobile/ubiquitous services.

- Then, they will send a SIP invite to the service URI. In the received SIP answer, by means of a specific "Grid Session Description Protocol", the application will get the needed Grid session information (i.e. the current service EPR).

Also, Grid applications could include the adequate logic to use advanced session management features provided by SIP like session save/load/restore, useful also for session transfer and to react to context awareness changes which are notified with SIP.

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

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

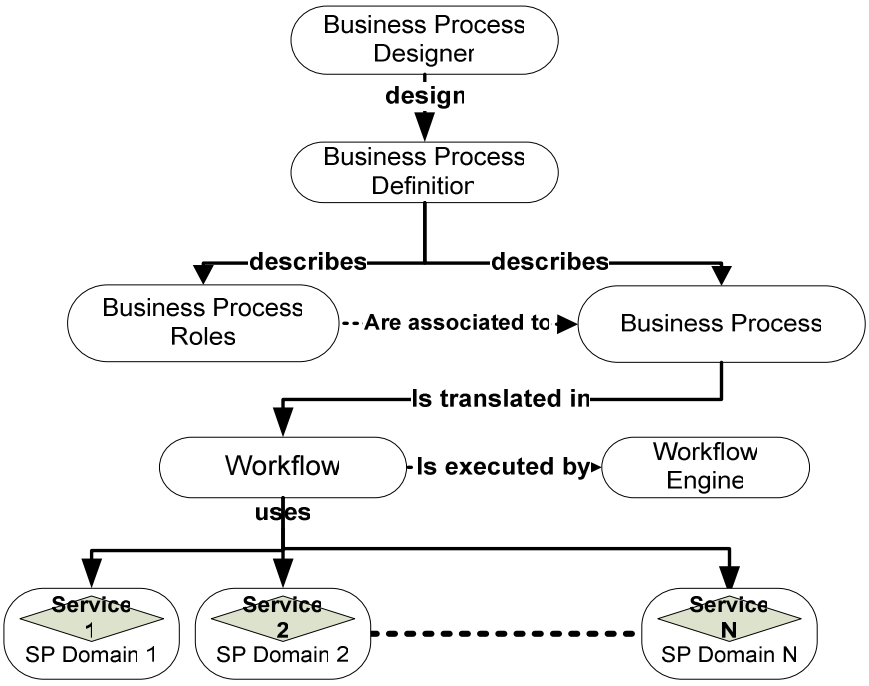

Figure 15 - Business Process Concepts

Figure 15 shows how the business process concepts are related among them. A business process designer designs the Business Process Definition that describes the Business Process and the associated roles. From the business level description the Business Process is translated in a workflow that will be executed by a Workflow Engine. During its execution the workflow uses different services hosted in different administrative domain. It is worth mentioning that in this context the term service refers to Web Services.

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 parties. Of course, some of this adaptation can be delegated to the services but ultimately any change that may require a change in the overall business process must be handled at the orchestration level. One immediate consequence of this is that the subsystem managing the workflow execution 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.

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 composition is the process by which simple services are aggregated in order to offer added value with respect to a simple service. Service composition 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.

5. The Akogrimo Security Framework

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.

The following table presents a quantification of these issues based on the high level architecture. Here, the fields are indication which security issues are addressed by Akogrimo at all. In the following subsection a more detailed description of selected areas is given.

Attacker / Target	User	Terminal SIP Client MIPL PANA	OPVO WF Manager BP User Agent,…	Base VO CM	Customer Domain Accountin g ID	Network Provider SIP Broker SIP Server	Service Provider EMS SLA Negotiator Service instance
User			X				X
Terminal							
OPVO	X	X	X	X	X		X
Base VO	X	X	X		X		X
Customer Domain			X	X			X
Network Provider							
Service Provider			X	X	X	X	X

Table 2: Security focus within Akogrimo

5.1. 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) capability to the user and to offer attribute sharing of the user to other components. The SAML Authority (Akogrimo's SAML Engine) 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.

5.2. 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 provides a PKI (Public Key Infrastructure).

The PKI consists of at least one Registration Authority (RA) and one Certification Authority (CA). Whenever an entity needs a (X.509) certificate 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.

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 (Sub-) 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.

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. 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.

5.3. Authorization and Centralized Policy Management – Security Policy Case

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 is 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.

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 OpVOs 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.

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 fac-

tory 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.

5.4. 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.

5.5. 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 Grid and Network sessions, 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 services 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 that he/she can move an ongoing session to a most suitable device.

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

- The SOAP interface, which accepts incoming WS requests from Grid entities.
- The Broker Engine, which is translating 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 the corresponding SIP URI. This is made through an A4CServer's user profile request that returns the SIP AoR of the user. 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 non-authorised entities making use of the facilities that the SIP Broker provides, security mechanisms must be implemented to protect each possible access path to this component. Figure 16 shows both the authorised way to use the SIP Broker and the potential attacks that have been identified.

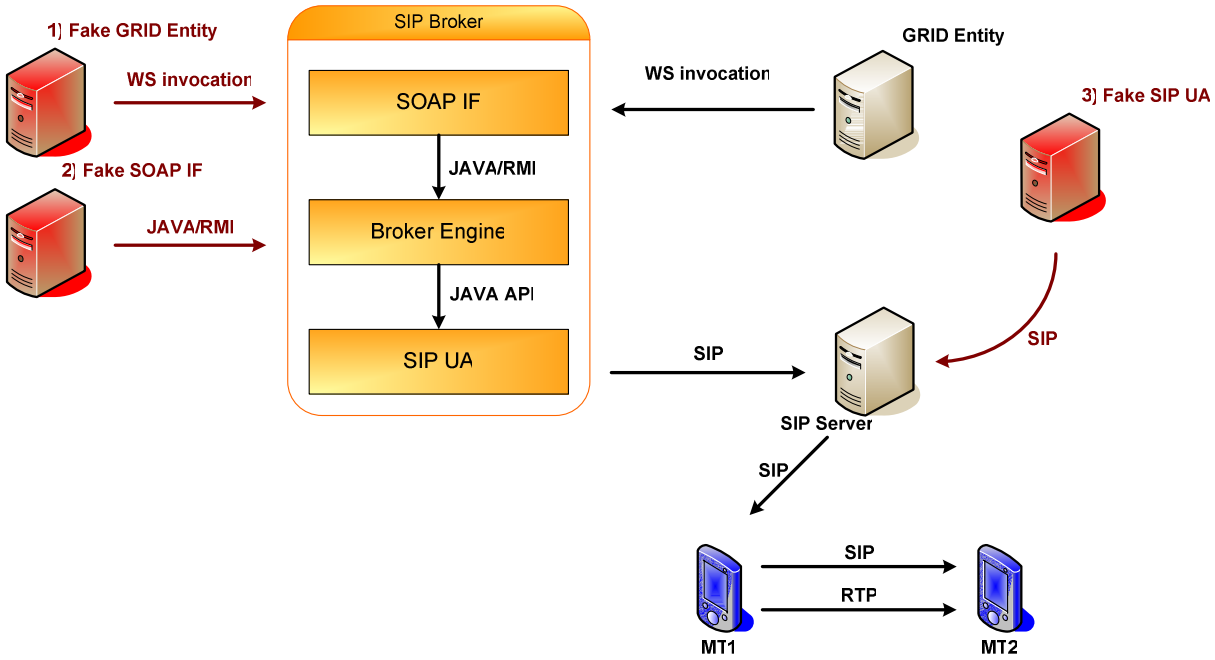

Figure 16: "SIP with SOAP" possible attacks

Normal operation will start with Grid Entities requesting a SIP process using the interface provided by the SOAP IF subcomponent; 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. These interactions have been depicted with black arrows in the figure.

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

1. If the SOAP interface does not check the identity of the invoking element (i. e. the EMS or the Workflow Manager), it could be possible for a non-authorised element to make use of the exposed WS interface to trigger bogus SIP interactions. In order to avoid this kind of attacks, the SOAP IF component must implement authorisation mechanisms. This is solved using WS-security functionalities.
2. The broker engine is exposed as a RMI daemon, so it could be invoked from a remote machine. If the identity of the calling entity is not guaranteed, some entity knowing 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 solved by means of Java RMI security functionalities.
3. 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 users' MTs. If neither the SIP Server nor the end users terminals check for the request authenticity,

this non-authorised entity could initiate a session or perform a transfer. This is solved using SIP security functionalities.

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)

5.6. 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.

5.7. 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). The most important business flow types consist of the following:

- **Workflows**: Worflows document the structured way in that tasks of a business process are executed, and they 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. 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. Financial flows are 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.

MDVOs are composed by legally independent organizations. The single organizational entities form administrative domains, each disposing of resources and services. Resources are offered to the MDVO through well-defined interfaces, defining services which in turn encapsulate the economic potential of available resources.

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 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.
- **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.
- **Role Model**: Different role holders in an MDVO hold different information pieces. 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. 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.

6. Conclusion and Further Reading

This document summarises the high level Akogrimo architecture and can be regarded as a baseline document for the whole Akogrimo project. After a rather strategic positioning of the architecture the generic architecture has been introduced showing the key architectural building blocks.

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 A4C 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 the key goal here was a convergence of Network and Grids. 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 A4C concept. This instead required an adaptation of traditional grid accounting principles towards the IETF A4C concepts.

As this document does intentionally does not cover all aspects worked out in the architecture workpackage or even in the detailed design work and does not discuss the findings in the business related workpackages find below a recommended incomplete list of deliverables recommended for further reading

Area	Deliverable Name	Content
Architecture	D3.1.3 Mobile Grid Reference Architecture	The more elaborated and detailed version of this document
	D5.1.2 Integrated Prototype	In this document the physical architecture of the prototype is described.
Design	D4.1.2 Consolidated Network Service Provisioning Concept	This document describes the different points of interaction between the network infrastructure and the Grid running on top of it. The architecture of this network infrastructure, its requirements, and components were detailed in D4.1.1. The

Area	Deliverable Name	Content
		current document takes this architecture as a starting point and emphasizes the symbiosis between the network infrastructure and the Grid from the point of view of the network side while tackling other innovative aspects not necessarily inherent to the classical network architecture approach, such as the handling of "user location".
	D4.2.2 Integrated Services Design and Implementation Report	The document describes the functionality implemented as part of the network middleware layer of the Akogrimo architecture. The network middleware layer provides a set of funtions to the upper layers, allowing Akogrimo to present several enhancements to the standard Grid architecture (i.e. OGSA). Specifically, the network middleware layer offers cross layer A4C, service registration and discovery, and presence and context management.
	D4.3.2 Prototype Implementation of the Infrastructure Services Layer	This document describes in short the services prototypes that have been developed under the workpackage WP4.3 Grid Infrastructure Services Layer of the Akogrimo project. The services presented in this document have been based on the architecture that has been designed in the deliverable D4.3.1 "Architecture of the Infrastructure Services Layer V1".
	D4.4.2 Prototype Implementation of the Grid Application Support Service Layer	In this document, the functionality and implementation status of the first prototype of Grid Application Support Service layer is presented. The different components, their interfaces and implementation technologies are documented.
Business	D3.2.1 The Akogrimo Consolidated Value Chain	This document addresses economic opportunities that are available to the different participants in a value chain for solutions based on Mobile Grid Services.
	D3.2.2 Business Modelling Framework	Mobile Grid Services only seem to be justified if there are economic advantages for enterprises. This means that enterprises need to be able to generate competitive advantages by using a Mobile Grid infrastructure. For this reason it is examined in the last consequence of this deliverable which business strategies can be followed by today's telecommunication or Grid companies in order to maximize their benefits from offering services in the Mobile Grid field. Marginally, a particular business strategy for consulting enterprises and academic institutions is mentioned.

7. References

[1] The Akogrimo project, website http://www.mobilegrids.org