# D2.2.1 Vol 2

## WP2.2

## Version 1.0


## WP2.2 D2.2.1 Volume 2 – The State of the Art

## Dissemination Level: Public

Lead Editor: Julian Gallop, CCLRC

14 April 2005

Status: Final

**SIXTH FRAMEWORK PROGRAMME**

**PRIORITY IST-2002-2.3.1.18**

*Grid for complex problem solving*

*Proposal/Contract no.: 004293*

**License**

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

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

**1. Definitions**

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

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

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

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

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

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

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

a. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any reference to such Licensor or the Original Author, as requested.
b. You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation. The exchange of the Work for other copyrighted works by means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.
c. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work, You must keep intact all copyright notices for the Work and give the Original Author credit reasonable to the medium or means You are utilizing by conveying the name (or pseudonym if applicable) of the Original Author if supplied; the title of the Work if supplied; and to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work. Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.
d. For the avoidance of doubt, where the Work is a musical composition:
    i. **Performance Royalties Under Blanket Licenses**. Licensor reserves the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital

performance (e.g. webcast) of the Work if that performance is primarily intended for or directed toward commercial advantage or private monetary compensation.

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

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

**5. Representations, Warranties and Disclaimer**

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

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

**7. Termination**

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

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

under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

**8. Miscellaneous**

a. Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
b. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
c. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

**Context**

| | |
|---|---|
| **Activity 2** | Environment & Requirements definition |
| **WP2.2** | Environment & Project Context |
| **Task 2.2.2** | **State of the Art** |
| **Dependencies** | **Does not depend on any within the Project**<br>**Needed as input for Activities 3 and 4** |

**Contributors:** Jan Wedvik (Telenor), Stefan Wesner (HLRS), Theo Dimitrakos (CCLRC), Giuseppe Laria (CRMPA), Christian Morariu (UniZH), Nuno Inacio (IT-Aveiro), Patrick Mandic (USTUTT), Robert Piotter (HLRS), Ignaz Mueller (HLRS), Brynjar Viken (Telenor), Sara Frances Gonzales (TID), Ruth del Campo (USTUTT)

**Reviewer:** Juergen Jahnert (USTUTT)

**Approved by: QM**

| Version | Date | Authors | Sections Affected |
|---|---|---|---|
| 0.1 | 4/1/05 | Julian Gallop (ed) | Created from latest version of the internal report ID2.2.1 Volume 2 |
| 0.98 | 28/3/05 | Julian Gallop (ed) | References now distributed to the associated chapters.<br>Fuller set of cross-layer themes – service discovery (from document edited by Brynjar Viken); VO Management (Sara Frances Gonzales, TID); user management and identity model (Patrick Mandic - USTUTT). More complete chapter 5 (Ignaz Mueller - USTUTT). Many other subsections completed, correspondences between different subsections resolved resulting in movement of text to better places and writing introductory sections (Julian Gallop - CCLRC).<br>Invitations to add to the lists of conclusions and challenges in chapter 1 and to the abbreviations. |
| 1.0 | 19/4/05 | Julian Gallop (ed) | Merged new material on Liberty and Shibboleth from Ruth del Campo (USTUTT); improvements based on comments from Robert Piotter (USTUTT); other improvements including filling some missing term definitions. |

# Table of main sections



# Detailed Table of Contents

# List of Figures

# List of Tables

# 1. Summary and conclusions

The State of the Art document reports on technologies which are relevant to design and implementation in Akogrimo. It brings together the knowledge of experts in the, until now, separate fields of mobile networks, Web Services, Grids architecture and semantic web.

Through this work and other analysis work taking place in the early phase of the Project's design-oriented working packages, a number of technologies and specifications are becoming preferred within their field:

- OGSA - the key definition for Grids as foreseen at present, but adapted as necessary to offer and use context information when users and services are mobile;
- Web Services, as a concept being adopted widely for loosely coupled, adaptive distributed computing for business purposes;
- SOAP as an underlying messaging protocol for Web and Grid Services;
- SIP as a protocol for signalling;
- WS Security as a collection of specifications for security at the Web Service level, together with experience from Grid projects which have developed prototypes;
- Mobile IPv6 - which provides a necessary and probably sufficient basis for handing over a mobile node from one connection point to another;
- The use of specifications such as OWL-S for a basis for knowledge-based discovery of Grid Services.

However there remain challenges to be met by subsequent work in Akogrimo. There are challenges that are of a general nature:

- Identifying the gaps between mobile networks and Grids and bridging them.
- Adapting concepts which have previously been defined for non-mobile Web Services
- Ensuring a clear and well-understood relationship from specific technologies to present future requirements
- Identifying the requirements for future standardisation

There are challenges that relate to specific technologies:

- Making user context available to Grid interfaces;
- Ensuring that security/authorisation works when a user changes their connection method (for instance, a secure connection being replaced by an insecure one);
- Ensuring that a Quality of Service agreement is viable when the connection method changes;
- Integrating grid services with the accounting infrastructure from the network layer.

# 2. Introduction and structure of the report

Deliverable D2.2.1 of the Akogrimo Project consists of 2 volumes. This is volume 2, Report on the State of the Art. Its purpose is to describe and assess the current state of the art of technology and standards relevant to Akogrimo.

Chapters 3 to 6 describe the technologies which correspond to 4 layers as defined in the Project's Description of Work (DoW) and also the separate workpackages WP4.1 to WP4.4 for design and implementation. Starting with the highest layer at the top, these chapters are as follows:

- Chapter 6: Mobile Grid Application Support Services Layer: This chapter describes support services which go beyond Grid foundations and package Grid Services in a convenient way for application domains – within the Project, the testbeds in particular.
- Chapters 4 and 5: These correspond to the middle 2 layers. These two layers have been the subject of reallocation within the Project and, although the same titles are preserved, the chapter allocations are slightly different from the ones described in the DoW:
  - Chapter 5: Mobile Grid Infrastructure Layer – This chapter focuses on specifications which provide the foundation for Grids based on Web Services; the addressing of and access to resources; and manageability of services.
  - Chapter 4: Mobile Network Middleware Layer – This chapter describes a range of technologies which can provide support for Mobile Grids, including security, charging, pervasiveness and knowledge discovery and collaboration.
- Chapter 3: Mobile Network Layer: This describes the technologies necessary for supporting mobile networks.

Some topics are represented at multiple layers. For certain topics, the descriptions in the layer chapters are complemented by a section in chapter 7, which brings together these cross layer topics.

To take security as an example, it is described in the layer chapters as follows:

| sections | security topics |
|---|---|
| 6.3 | specifications to support security in Web Services, including WS Security;<br>adaptive security for VOs with dynamic constitutions |
| 4.2 | security specifications to support the infrastructure, including Diameter, Radius, Kerberos, PKI, X.509 |
| 3.5 and 3.6 | network access security, including IPSec;<br>end to end security, including OpenPGP |

**Table 1 - Description of security at each layer**

Correspondingly in chapter 7, there is a section on AAA (7.4), which includes a description of a possible Akogrimo security architecture for Virtual Organisations, the relationship of an AAA Client and Server and how this works on a mobile node. There is also a section on models for user management and identity (section 7.1), which is an important foundation for authentication.

Following that chapter, there are chapters on updating this document, and some selected abbreviations and terms.

# 3. Mobile Network Layer

This chapter describes the main technologies that can be used to deploy a mobile network. Mobility is approached in different ways depending on the kind of mobility addressed, whether terminal mobility, user mobility, session mobility, inter-domain mobility or inter-technology mobility.

Some concepts cut across multiple layers and are presented in this way in the chapter on Analysis of cross layer themes,. In the case of Service Discovery, the aspects relevant to the mobile network layer are not discussed here at all, but are postponed to the corresponding section (7.3) in chapter 7.

## 3.1. Access technologies

In the Akogrimo framework, mobility plays a very important role that must be studied at every level. In reference to the types of access technologies, and in order to provide the maximum degree of mobility, some wireless technologies are presented together with other wired technologies. It is important to remark that all these access technologies can support the use of IP as network layer protocol.

**GSM/GPRS** [1][27][23] GSM is one of the second-generation (2G) mobile phone technologies and currently the most popular one. It is mainly optimized for voice communications, is based in circuit-switching and the maximum transmission speed that can achieve is of 14.4 kbit/s. General Packet Radio Service (GPRS) extends GSM in order to provide a packet-switched based communication instead of circuit-switched, making it more suitable for applications like web, instant messaging, and so forth. GPRS permits data rates of up to around 140kbits/s.

**UMTS** is one of the third-generation (3G) mobile phone technologies and is considered the successor of GSM. W-CDMA, standardized by the 3GPP, is used by UMTS as the underlying standard. It permits packet-switched data transmission and permits rates of up to almost 2 Mbit/s. GSM-only mobile stations connect to the network using the GSM air radio interface (Um), while UMTS stations connect using the UMTS air radio interface (Uu). UMTS stations which are outside the service area connect using the Um.

**WiFi [22]** is a set of standards based on the IEEE 802.11 specifications to build local area networks (LAN) that allows wireless communications in an unlicensed part of the radio spectrum. Its communication range may go up to a 100m and the transmission rate can be of up to 54Mbit/s (802.11g).

**WiMax[21][26]** refers to the IEEE 802.16 group. The aim is to provide wireless access to metropolitan areas (MAN) of up to a radius of 50km with a data rate of up to 70Mbit/s.

**Bluetooth** is a technology that permits wireless connections within a small range of around 10m (personal area networks or PAN). It operates in a free-license bandwidth and can achieve rates of about 700 kbit/s.

**Wired networks**. Mobility does not necessarily imply wireless. A user or mobile device may connect from different points of the network even if these are wired accesses. Examples of these networks can be the extensively used Ethernet, or FDDI (optical fiber).

Change of point of attachment to the network (hand-over) at this level, are usually done at the hardware layer, providing fast, efficient and seamless movement.

## 3.2. Mobile IPv6

A network which is supposed to be ubiquitous and allow any kind of heterogeneous nodes can not be linked to one single access technology. It is worth mentioning that seamless mobility between different access technologies has been achieved in the Mobydick [25] project and is being more deeply studied in the Daidalos [20] project. The base, to permit this link-layer independent mobility, is given by Mobile IPv6, which is a network layer mobility solution based on IPv6.

### 3.2.1. Basic functionality

Due to the fast evolution and constant growth of the Internet, there has been a need to redesign IPv4 in order to accommodate present and future demands. To this end, IPv6 was developed as a replacement and brings many improvements, among others, extended addressing capabilities, header simplification, extensibility, and so on.

However, for both IPv4 and IPv6, the IP address topology is designed in a way that determined addresses belong to determined networks or sub-networks, and thus, the routing of packets is performed according to this structure. If a node disconnects from its point of attachment to the network and connects to the network at any other place, a reconfiguration of a new IP address, net-mask and routers is necessary.

Mobile IPv6 extends IPv6 so that mobile nodes are able to change their point of attachment to the network with minimal disruption. When a mobile node roams across different networks, Mobile IPv6 deploys a mechanism, restricted to the network layer and so transparent to higher layers. Therefore, already established connections (for example a TCP connections) are not dropped when a node changes its position from one network to another and needs to reconfigure its network characteristics. In fact, these connections are not even aware of this change. Since Mobile IPv6's scope is restricted to the network layer, it does not impose any requirements on the lower layers. This means that a mobile node is able to roam across networks independently of the access technology. For example, a hand-over may take place between two Ethernet accesses or between an Ethernet and a WLAN access without any concern.

The main idea of how mobility has been implemented in MIPv6 revolves around the use of two addresses for a node to be exhaustively addressable at any point. These addresses are:

- Home Address (HoA): As any other regular node, a mobile node has a permanent address, which is valid inside its home network. This address always remains the same, independently of in which network the mobile node is located, i.e. whether at home or roaming. This is the only address that the transport or higher layers are aware of.
- Care-of Address (CoA): For as long as the mobile node is located in a network other than its home network (i.e. a foreign network) it makes use of the CoA, which is a suitable address for the foreign network in which the mobile node is located at that moment. The relation between the HoA and the CoA of a mobile node is called a binding.

The new entities that the Mobile IPv6 protocol adds to IPv6 are described as follows:

- Mobile node (MN): A node with the ability to change its point of attachment to the network keeping connections alive.
- Home agent (HA): A node that resides at the MN's home network, which is in charge of forwarding traffic directed to the MN while it is away from home so that it looks as though the MN is virtually at his home network. To this end, the HA must be aware of the MN's current binding.
- Correspondent node (CN): Any peer node which an MN is communicating with. A CN does not necessarily have to implement mobility i.e., a CN may be either mobile or stationary.

When the MN is away from its home network, the HA will act on its behalf, forwarding all the packets addressed to the MN at its HoA to its current CoA by means of a tunnel. The CN is only aware that the MN is roaming if it supports Mobile IPv6. If this is the case, the CN will realize that its traffic is being tunnelled by the HA and may perform a route optimization by directly communicating with the MN by means of the MN's CoA instead of using the tunnel provided by the HA. This should be the standard mechanism because it notably increases the performance. In order for the MN to communicate its position to the HA and all the CNs it is communicating with, a Binding Update (BU) packet is sent with the address currently used and a Binding Acknowledgement (BA) packet is received by the MN as a confirmation. These packets must be protected, since several attacks can be applied if this is not done, such as man-in-the-middle, impersonation, Denial of Service and so on). The standard way to protect these packets is IPSec, although there are other alternatives such as using cryptographically protected CoAs.

### 3.2.2. Fast Handover

The goal of Mobile IPv6 is to achieve seamless mobility of devices. Due to its movement, the MN may have to change its point of attachment to the network by attaching to a new access router that is closer to its current position. This change of point of attachment to the network is called handover and involves acquiring a new CoA and communicating the HA and CNs its new location before it can be fully operative again and the communication can be re-established. However some applications, such as voice, for example, are very sensitive to delays and need the handover to be fast enough for it to be seamless. Fast Handover [24] reduces handover latency by anticipating the handover using the link layer. It allows the MN to send packets as soon as it detects a new network link, and also allows the MN to receive packets as soon as its attachment to the new link is detected.

## 3.3. Session-based mobility

In the previous sections, two different methods have been described in order to support mobility. The first one works at link layer and permits very fast hand-overs, the second one is handled at software layer and is link-layer independent. The kind of mobility provided by these methods is however limited to the terminal that the user is currently utilizing (i.e. terminal mobility). In order to achieve a concept of mobility that is more user oriented, based on the mobility of the user independently of the terminal or terminals utilized by him, session mobility is the solution. Session mobility permits the user to maintain independent sessions using different

devices, change the device used in the middle of a session, reconnect from a different location, and so forth. In the Akogrimo context this can be used to establish MDVO sessions.

In the following, the main technologies that are able to provide session mobility are shown:

**H.323 [19]** is an ITU-T recommendation designed to provide multimedia communication services over packet-based networks, which is extensively used in Voice over IP (VoIP). It does not set any restriction regarding the kind of transport protocol to use. The H.323 recommendation encompasses wide variety of fields: point-to-point and multipoint conferencing, inter-network interoperability, heterogeneous client capabilities (video and data support are optional), audio and video codecs, management and accounting support, security (authentication, integrity, privacy, non-repudiation) and supplementary services like call forwarding or call transfer. This shows the great complexity of H.323.

**SIP [18]** stands for Session Initiation Protocol. SIP is being standardized by the IETF. It is a protocol oriented to establish multimedia communication services over IP networks. It is based on other existing popular protocols like HTTP or SMTP and is, by far, less complicated than H.322. SIP allows users to call each other independently of their location. By carrying another independent protocol called SDP (Session Description Protocol), the session to be established is described. However, SIP is independent of the protocol used to describe the sessions.

## 3.4. Interdomain mobility

A new dimension in the concept of mobility is the free roaming of users across different administration domains. It is a fact that this has been a great success in the cellular telephony world; therefore in a future IP-based architecture for mobile telephony, similar mechanisms in order to authenticate, authorize, account, audit and charge users (A4C) [17] need to be developed. This would provide a user with the capacity of freely roam across different administration domains and providers holding a unique contract with its "home" provider. In terms of authenticating a user in order to get access in different administration domains there are several technologies that should be taken into account.

**Diameter [16]** is an AAA protocol designed by the IETF that allows a user to roam across different administration domains. When a user enters a foreign administration network the user's authentication information is conveyed from this foreign domain to its home domain in order to prove its veracity so that the user is authorized to access the new network. Diameter can also understand mobile IP and therefore convey mobile IP information to the user's home domain before he is authenticated and authorized to use the network, in order to make the hand-over across different domains less heavy and therefore seamless to the upper layers. In addition, the IETF is putting effort into standardizing how SIP and Diameter have to work together.

**PANA [15].** The Protocol for carrying Authentication for Network Access was designed by the IETF in order for a client to authenticate against other networks independently of the type of access network utilized by utilizing IP-based protocols. PANA allows a client to get authenticated via a back-end AAA infrastructure so that the client does not need to understand any specific AAA protocol. PANA carries EAP payloads in order to perform the authentication.

**EAP [14].** The Extensible Authentication Protocol is defined by the IETF in order to carry different authentication methods depending on the requested necessities. These methods may be based on shared secrets, certificates or any other authentication fundamentals. EAP defines just four types of packets, which are: request, response, success and failure. The authentication

procedure takes a different number of round trips depending on the type of authentication method chosen. In the following the most outstanding ones are mentioned.

- EAP-MD5 is based on a shared secret method. The authenticator sends a challenge to the client, who hashes the challenge together with the shared secret and sends the result back to the authenticator.
- EAP-Archie is another method based on shared secrets. Apart from authentication the client, Archie has the advantage over MD5 authentication that mutual authentication can be performed so that the man-in-the-middle attack can be avoid. In addition, once the authentication is performed, the two peers involved are able to derive a session key with the cryptographic material exchanged.
- EAP-TLS is based on digital certificates and therefore in this case a PKI is needed. It is based on TLS, which is a mechanism that is extensively used to secure web pages. Mutual authentication can be performed.
- EAP-TTLS. This method establishes a TLS secure "tunnel" through which a TLS authentication is performed. The major advantage of this method is that the identity of the user remains hidden since it is sent when a secure tunnel is already established.
- EAP-PEAP. This method also establishes a TLS channel before authenticating. It is independent of the authentication method chosen.
- EAP-AKA is based on a shared secret method. This method is used in UMTS to perform authentication. It requires a SIM (smart) card or a virtual operative unit to emulate it. This method notably reduces the number of round trips required in comparison with the methods based on TLS and does not need to use any kind of PKI infrastructure. EAP-AKA is, therefore, a very suitable for keeping the transparency of the handovers.

## 3.5. Network access security

In a mobile network where heterogeneous access methods are used and especially wireless type networks, it is extremely important to provide a secured connection between a user and the network. In order to achieve this goal, most access technologies offer their own encryption methods to provide message privacy.

- **GSM/GPRS/UMTS.** In this kind of network, a user authentication is performed by means of a shared secret embedded in a smart card. UMTS in addition to authentication the user to the network provides with the capability for the user to authenticate the network, in order to avoid fallacious base stations. User traffic and signalling are both encrypted over the radio link.
- **WLAN.** In the area of wireless LANs, there are three technologies that should be commented:
  - WEP: The Wired Equivalent Privacy is part of the 802.11 standard. It is known to be very vulnerable and its use is not suggested.
  - WPA: The Wi-Fi Protected Access appeared as a response of the vulnerabilities of WEP. It provides data encryption, user authentication (with an EAP based solution) and message integrity.

- 802.11i (WPA2): Based on WPA. Includes support for roaming and pre-authentication and provides stronger cryptography.

As it can be seen, there exist different access technologies with different levels of security. A highly heterogeneous network will probably make use of many different access technologies in order to provide access to different kinds of users. It is desired that no matter what the access technology utilized is, a certain degree of security can be assured. By means of a VPN (virtual private network), users can access a network securing their access independently of the access technology utilized. In order to do this, several protocols can be used:

- **IPSec [13]** is an abbreviation of IP security. It is the standard method to secure IP packets designed by the IETF. IPSec can provide integrity, authenticity and confidentiality of IP packets by means of two different protocols: authentication header (AH) and encapsulated security payload (ESP). In addition, two modes of communication are supported: transport mode and tunnel mode. In transport mode the upper-layer protocols are protected whereas in tunnel mode the entire IP datagram is protected by encapsulating it in a new IP datagram using the IPSec. IPSec is often used to create VPNs.
- **PPTP [12].** The Point-to-point tunnelling protocol was developed by Microsoft and is considered insecure. Its use is not advisable.
- **OpenVPN [11]** is a SSL based VPN solution. It makes use of a single IP port to forward all the tunnelled datagrams. It is able to create tunnels at level 2 (Ethernet) or level 3 (IP). The authentication can be performed by means of pre-shared secret keys, certificates or passwords.

## 3.6. End-to-end security

Regarding end-to-end security there are several technologies that can be used:

**IPSec** (also mentioned above) can be used in transport mode to provide end-to-end security. The IP datagrams are totally secured except for the fields that may be changed by the routers (TOS, TTL…). It provides a secure connection between communicating hosts, not between users. This is totally transparent to applications.

**SSL/TLS [10]** enable end-to-end security. They run on a layer underneath application protocols and above the TCP protocol. In order to authenticate the end points, public key certificates are used. The main drawback of this mechanism is that it does not support the use of applications that utilize UDP as transport protocol, and therefore is not as absolute a solution as like IPSec is.

**OpenPGP [9]** and **S/MIME.** OpenPGP is an IETF internet standard which is used to protect data in general. It is well known for its use in e-mail security. It provides user-to-user protection by using private-public key pairs. On the other side, in a similar fashion to OpenPGP, S/MIME provides a way to secure MIME and is also used for e-mail security. S/MIME is based on X.509 certificates, and therefore uses a hierarchal system for certification of public keys whereas OpenPGP is based in the so-called Web of Trust (unordered structure). These two protocols can be utilized to provide end-to-end security to protocols like SIP.

## 3.7. Policy-based Network Management (PBNM)

Networks that encompass a great number of users and services need to make use of some kind of network management that helps to abstract and ease their operation. This fact needs to be addressed specially taking into account the mobile nature of the networks we are dealing with. Nodes may appear and disappear and the configuration of the network needs to be accordingly changed. This can be achieved with the so-called Policy-based Network Management (PBNM), which uses a set of rules to react to the eventualities that may occur. The structure of PBNM typically involves a policy server and some clients that enforce the policy decisions. In order to make this possible, a protocol for this communication is necessary.

**COPS.** The Common Open Policy Service (COPS) protocol (IETF [28]) is a client/server policy exchange protocol. The clients (Policy Enforcement Point or PEP) send requests to the server (Policy Decision Point or PDP). The PDP then makes a decision and sends it to the PEP. The COPS protocol is extensible; it can recognize new self-identifying objects and supports client specific information without changes to the protocol. It uses TCP for transport, provides message level security for authentication, replay protection and message integrity. It can also use IPSec or TLS for authorization and securing communications between the PDP and PEP.

**CIM/XML.** The Common Information Model (CIM) [8] is an object-oriented common data model for representing systems management information in enterprise computer systems. The exchange of CIM information is done using CIM-XML.

## 3.8. Quality of Service (QoS) in a mobile network

Some types of network traffic have certain requirements in order achieve a correct functionality. For example: IP telephony requires a limited jitter and delay, multimedia requires a certain throughput. Quality of Service (QoS) [7] refers to the capability of a network to provide some guarantees to the different kinds of traffic according to their nature and requirements and/or according to a Service Level Agreement (SLA). The mechanisms utilized to provide QoS may involve mechanisms such as giving different priorities to traffic flows, traffic shaping, resource reservation, etc.

Providing QoS in a shared mobile environment is a matter that has to be studied with special attention. The solutions to provide QoS in mobile networks based on IP may be similar to the ones used in static networks. However, assuring a mobile user with a determined QoS implies new issues, which are linked to the user's mobility, to the shortage in bandwidth resources, and to the high loss of information, which is typical of the wireless connections. The resources of a wireless connection may suddenly change, due to physical obstacles between the user and the access point, due to the weather conditions. They may also change because a user moves to a cell that cannot assign any resources to him/her. The issue of wireless networks is therefore the possibility of anticipating which cell will be occupied by the user, as well as designing adequate mechanisms that can react to the user's movements. Mobility is effected by means of a handover (section 3.2) and the QoS can be affected during and after this.

**IntServ** and **DiffServ**: To facilitate true end-to-end QoS on an IP-network, the Internet Engineering Task Force (IETF) defined two models: **Integrated Services** (**IntServ,** [6] and IETF RFC 1633 [29]) and **Differentiated Services** (**DiffServ,** [6] and IETF RFC 2475 [30]). IntServ follows the signaled-QoS model, where the end-hosts signal their QoS need to the network. It is a framework that provides individualized QoS guarantees to individual application sessions. DiffServ works on the provisioned-QoS model where network elements are set up to service multiple classes of traffic, with varying QoS requirements. Both models can be driven off a policy base, using COPS (see 3.7above).

The **Intserv** architecture has two important features – a router is required to know the resources it has already reserved for other sessions, and any session that requires QoS guarantees must first reserve enough resources at each router on the session path to destination, thus making sure that its end-to-end requirements are met.

In order to set up a session, that session must first declare its QoS requirements and characterize its traffic so that the routers can determine if they have sufficient resources to meet the session's requirements. The QoS requirements are defined in the RSpec (Resource Specification) and the traffic requirements are defined in the TSpec (Traffic Specification). The session's TSpec and RSpec are then sent to the routers at which the resources will be reserved using the RSVP protocol. After receiving the characterization of a session, the router determines whether it can admit the session.

The Intserv architecture defines two classes of service: guaranteed service and controlled-load service. Guaranteed service provides mathematically provable bounds on queuing delays. Controlled-load service provides a quality of service resembling an unloaded network – small percentage of dropped packets and close to zero queuing delays. The controlled-load service is particularly suited for real-time

The Intserv framework allows guaranteed QoS for individual flows, but it has some limitations, such as:

- Scalability – the routers need to maintain per-flow state for all the flows that pass through the router. In large enough networks, this will imply a significant overhead.
- Flexible service differentiation – Intserv has a small number of service classes.

**Diffserv** was designed in order to provide scalable and flexible service differentiation. Diffserv implements only simple functionality at the core of the network; more complex operations are done at the edge of the network. This way, the hardware requirements for routers are much lower than for Intserv.

The edge routers mark arriving packets according to their traffic class. The different classes will then receive different service inside the core network. After marking the packet, the router can forward it immediately, delay it or discard it. The core routers simply forward arriving packets according to their markings and the per-hop behaviour of the packet's class. Packets with the same traffic class are treated the same by core routers, independently of any other properties (source, destination, etc.), and thus the core routers needn't keep the state for all the flows.

**802.1p [5].** The 802.1p signalling technique allows grouping packets into different traffic classes. 802.1p establishes eight levels of priority. Traffic is simply classified and sent to the destination; no bandwidth reservations are made. It can be used in conjunction with IP Precedence – using layer 3 switches, 802.1p prioritization can be mapped into IP Precedence before forwarding it to routers.

**802.11e [4]** is an enhancement that adds QoS features to 802.11a and 802.11b, enabling the prioritization of data, voice and video communications. The original 802.11 MAC featured two modes of operation: Distributed Coordination Function (DCF) and Point Coordination Function (PCF). The DCF only supports best effort services, and provides no guarantees of packet delay, bandwidth or jitter. It also degrades significantly under heavy load. PCF is a central polling scheme which can guarantee maximum latency, but is inefficient.

The original MAC provided no means of differentiating traffic. 802.11e features two new modes of operation: Enhanced DCF (EDCF), which is basically DCF with support for traffic classes, and Hybrid Coordination Function (HCF), which extends the PCF, but makes a much more efficient use of the medium than PCF. These two new modes support eight priority classes, which can be mapped directly to other protocol priority levels.

**802.16 [3].** 802.11 is a successful wireless technology. However, since it was designed as a LAN technology, it is not well suited for outdoors usage. The 802.16 specification is a MAN technology and targets mainly the "last mile" wireless communications. It can provide wireless broadband access to "hot-zones", which may encompass business, residential or rural areas, WiFi hotspots or even mobile users. Unlike WiFi which has ranges of up to a hundred meters, WiMax has ranges in excess of 10 miles.

802.16 MAC supports higher layer protocols such as ATM, Ethernet or IP, and was designed to easily accommodate future protocols. It delivers ATM compatible QoS: UGS, rtPS, nrtPS and Best Effort. Terminals can be assigned uplink and downlink burst profiles according to their link conditions. It uses a variable length PDU, as well as other mechanisms that increase its efficiency.

## 3.9. References for Mobile Network Layer


[1] Beaujean, Ch., Chaher, N., et al., Implementation and Evaluation of an End-to-End IP QoS Architecture for Networks Beyond 3rd Generation, http://www.it.uc3m.es/cgarcia/articulos/IST_Mobile_Summit_2003.pdf

[2] An Architecture for a Secure Service Discovery Service http://iceberg.cs.berkeley.edu/papers/Czerwin-Mobicom99/sds-mobicom.pdf

[3] WiMAX forum. http://www.wimaxforum.org/home

[4] Garg P. et all. Using IEEE 802.11e MAC for QoS over Wireless.

[5] http://www.xilinx.com/esp/wired/optical/net_tech/ieee8021p.htm

[6] Computer Networking, James F. Kurose, Keith W. Ross, 2nd Ed., Addison Wesley

[7] Beaujean, Ch., Chaher, N., et al., Implementation and Evaluation of an End-to-End IP QoS Architecture for Networks Beyond 3rd Generation, http://www.it.uc3m.es/cgarcia/articulos/IST_Mobile_Summit_2003.pdf

[8] DMTF. http://www.dmtf.org/education/

[9] Callas J. et all. OpenPGP Message Format. RFC 2440.

[10] Dierks T. and Allen C. The TLS protocol Version 1.0. RFC 2246

[11] OpenVPN. http://openvpn.sourceforge.net

[12] Hamzeh K. et all. Point-to-Point Tunneling Protocol (PPTP). RFC 2637.

[13] Kent S. and Atkinson R. Security Architecture for the Internet Protocol. RFC 2401.

[14] Aboba B. et all. Extensible Authentication Protocol (EAP). RFC 3748.

[15] Forsberg D. et all. Protocol for Carrying Authentication for Network Access (PANA). Internet draft - Work in Progress.

[16] Calhoun F. et all. Diameter Base Protocol. RFC 3588.

[17] Cuevas, A., Constantin, Ch., et al., AAAC Design, http://www.iwi.uni-hannover.de/lv/seminar_ss04/www/Norman_Behrens/bibliography/D0401.pdf

[18] Handley M. et all. SIP: Session Initiation Protocol. RFC 2543.

[19] Karim A. H.323 and Associated Protocols

[20] Daidalos. http://www.ist-daidalos.org/

[21] The IEEE 802.16 Working Group on Broadband Wireless Access Standards. http://www.irit.fr/~Ralph.Sobek/wifi/

[22] Wireless Fidelity – Specifications. http://www.irit.fr/~Ralph.Sobek/wifi/

[23] Bettstetter C. et all. GSM Phase 2+ General Packet Radio Service GPRS: Architecture, Protocols, and Air Interface.

[24] Fast Handovers for Mobile IPv6, http://www.ietf.org/internet-drafts/draft-ietf-mipshop-fast-mipv6-02.txt

[25] Moby Dick Project. http://www.ist-mobydick.org/

[26] Intel. IEEE 802.16 and WiMax. Broadband Wireless Access for Everyone.

[27] A Brief Overview of GSM. http://kbs.cs.tu-berlin.de/~jutta/gsm/js-intro.html

[28] Durham D. et all. The COPS (Common Open Policy Service) Protocol. RFC 2748 http://www.ietf.org/rfc/rfc2748.txt

[29] http://www.ietf.org/rfc/rfc1633.txt

[30] http://www.ietf.org/rfc/rfc2475.txt

# 4. Mobile Network Middleware Layer

This section provides a state of the art survey for the *Mobile Network Middleware Layer*. The term *mobile network middleware* comprises a range of technologies intended to enhance the Akogrimo infrastructure in terms of mobility support and cross-layer integration. Specifically, this section will address:

- Integrated service platform: Giving the application developer a unified view of the variety of technologies comprising the Akogrimo platform.
- Security, charging and pricing: To make the system user friendly and to make the incurred usage costs transparent, there should be an A4C infrastructure covering all layers, allowing e.g. single sign-on, and flexible composition of value chains.
- Pervasiveness: Akogrimo services should be omnipresent and universally useful. They should be able to serve the user optimally in his or her present situation and preferences, making optimal use of whichever services, networks and devices that are available.
- Collaboration support: The architecture should support collaboration between humans, seeing them and their knowledge as resources in the Grid.

Mobile network middleware is not an established term, and the concept is as such work in progress, attempting to bridge the gap between Grids and mobile networks.

## 4.1. Integrated service platform and middleware

Nowadays the highest demand for GRID applications is in scientific field although the amount of potential users is wider. In particular, mobile users might be the future users of this new technology. In fact wireless devices (laptops and PDAs), with currently limited resources, would benefit from the opportunity of using a considerable amount of resources made available by all the other devices connected to the network. Of course, the need for managing the transparency of the service and the mobility of users requires a big initial effort in the creation of a suitable integrated service platform and middleware.

The major goal within this topic is the design and implementation of a user-centred and programmer-friendly service platform for Akogrimo, where interoperability with heterogeneous networks is supported through the usage of the state-of-the-art middleware technology. This platform will provide:

- An interface to services implemented by the platform components and integrate these services into a coherent service model and APIs.
- Provide these services to users through a middleware layer.

The Network Middleware Layer constitutes the Service Provisioning Platform (SPP) towards the Grid Infrastructure Layer. Provided services are network signalling, security, confidentiality, non repudiation, authentication, authorisation, accounting, auditing and charging (A4C). These activities can be considered to be network centric and shall be linked to higher layer resources through the introduction of the concept of GRID based sessions and transactions (user/service

centric), which involves the development of a common signalling framework to provide GRID based services.

Network Middleware architecture will be designed and implemented under the WP4.2 scope, which addresses relevant concepts of networking and service provisioning from the operator's perspective. Main concepts are overall network management, QoS control security, SA4C and service provisioning, with special focus on the heterogeneous mobile landscape of operators and considering interconnection topics. Additionally, an integrated access point to enhanced GRID services (personalisation / adaptation, context-awareness, knowledge discovery and human collaboration support) through operator's mobile network will be developed. Considering that relevant technologies regarding some of these key concepts will be listed in further sections, this one will address the following:

- Network interoperability
- Signalling
- Network management
- QoS
- Middleware for Next Generation GRID (NGG)

### 4.1.1. Network interoperability

Akogrimo intends to be as global and generalised as possible, so agreements and collaboration between network exploiters seems to be crucial. We distinguish to levels of interoperability:

- Between different network exploiters (administrative domains roaming)
- Between access technologies (inter-technology roaming).

In some technologies are currently necessary roaming agreements, for instance, while subscribers can use the same Global System for Mobile Communications (GSM) phone in almost any country in the world, GSM's faster, data-oriented cousin, GPRS, doesn't travel so well. Roaming agreements allow the customers of one operator to use the networks of other operators when traveling away from their home country or city.

Regarding to 3G mobiles it would be convenient several roaming agreements since it is not a thought out idea. Network sharing can involve varying degrees of co-operation between operators of different mobile networks. There are several degrees of co-operation which depend on the amount of infrastructure shared between the parties.

The first level of network sharing involves the shared use of sites ranging from individual mast sites to sharing of the grid. It also may include site support infrastructure, such as site support cabinets. The second level of network sharing involves base stations, antennas and radio network controllers, also known as radio access network sharing, i.e. the sharing of initial transmission equipment. The third level of network sharing involves the core network, including mobile switching centres (MSCs) and various databases, also referred to as the "intelligent part" of the network. The last level of network sharing involves the sharing of radio frequencies.

In the last months the vision of agreements among enterprises has come up [31] seeing the need of producing a set of specifications that will allow grid products from various suppliers to

interoperate, and to produce test suites that will help customers identify products that are interoperable.

One common way to guarantee network interoperability between different entities is by defining a **Service Level Agreement (SLA**) between them. It is a contract between two or more entities which specifies, usually in a measurable way, the terms of service that an outsourcer will provide.

In this way is interestiong to consider the work of **Liberty Alliance Proyect Consortium** [32] which is committed to develop an open standard for federated network identity that supports all current and emerging network devices.

### 4.1.2. Signalling

One of the goals of the Network Middleware Layer is the design of a common signalling framework to provide Grid-based services. In this context "Common" does not mean that a unique signalling technique will be used, because the diversity of signalling task makes it almost impossible to define a single strategy that performs optimally all required tasks. There is a complex variety of signalling protocols, specialised in several network control topics.

The **Session Initiation Protocol** (**SIP**, IETF RFC 3261 [33]) is an application layer protocol created with the aim of creating, modifying, and terminating voice, video, and multimedia sessions independently from the underlying transport protocol used and from the kind of session instantiated. A goal for SIP was to provide a superset of the call processing functions and features present in the public switched telephone network (PSTN). SIP works in concert with several other protocols and is only involved in the signaling portion of a communication session. SIP acts as a carrier for the Session Description Protocol (SDP, IETF RFC 2327 [34]), which describes the media content of the session, e.g. what IP ports to use, the codec being used etc. SDP is carried as an opaque body in SIP messages, to describe multimedia sessions. Thanks to SDP, the parties involved in a SIP session can negotiate their receiving capabilities and communicate which media streams they are able to process; the information is carried in a human-readable text format. Additionally, some kind of mobility support is provided. **SDPng** extends SDP features on multimedia sessions description and capabilities exchange.

In typical use, SIP "sessions" are simply packet streams of the Real Time Transport Protocol (RTP). RTP is the carrier for the actual voice or video content itself.

The relationship between SIP and Grids has never been investigated and seems to be a promising candidate for supporting Grids based sessions and transactions signalling.

The ITU alternative to SIP protocol (commonly used in VoIP environments) is **ITU H323, Packet-based multimedia communications systems.** It comprises a suite of protocols for enabling audio/video conferencing capabilities. Current version is H323v5 [35].

**Real Time Streaming Protocol** (**RTSP**, IETF RFC 2326 [36]) is the IETF-standardized protocol for controlling streaming media servers. RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data. RTSP and SIP share a number of common functions. For example, both establish sessions and use SDP to describe them.

**Real-time Transport Protocol** (**RTP,** IETF RFC 3550 [37]) provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services. RTP does not address resource reservation and does not guarantee quality-of-service for real-time services. The data transport is augmented by a control protocol (**RTCP**) to allow monitoring of the data delivery in a manner

scalable to large multicast networks, and to provide minimal control and identification functionality.

The **Hypertext Transfer Protocol** (**HTTP**, IETF RFC 2616 [38]) is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol which can be used for many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through extension of its request methods, error codes and headers.

**Extensible Markup Language** (**XML** [39]) is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.

Regarding Mobile Dynamic Virtual Organisations Management, **Mobile Agent Technology** (MA) [40], [41], seems a good solution to provide some kind of support on this topic, specially in the Mobility and Managements areas. MA is considered to be an enabling technology for automated, flexible and customized service provision in a highly distributed way as network nodes become active and take part in the computation of applications and provision of customized services. This distribution adds another technical advantage, namely scalability, while at the same time bottlenecks of centralized approaches such as reduced network availability and malfunctioning are avoided.

### 4.1.3. Network Management

There are several meanings for *Network management*. In some cases, it involves a solitary network consultant monitoring network activity with an outdated protocol analyzer. In other cases, it involves a distributed database, auto polling of network devices, and high-end workstations generating real-time graphical views of network topology changes and traffic. In general, network management is a service that employs a variety of tools, applications, and devices to assist human network managers in monitoring and maintaining networks.

The **Simple Network Management Protocol (SNMP)** is an application layer protocol that facilitates the exchange of management information between network devices. It is part of the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite. SNMP enables network administrators to manage network performance, find and solve network problems, and plan for network growth.

Three versions of SNMP exist: SNMP version 1 (**SNMPv1,** IETF RFC 1157 [42]), SNMP version 2 (**SNMPv2,** IETF RFC 1902 [43] to RFC 1908) and SMNP version 3 (**SMNPv3,** IETF RFC 3410 [44] to IETF RFC 3418). All versions have a number of features in common, but SNMPv2 offers enhancements with relation to SNMPv1, such as additional protocol operations. SNMPv3 introduces improvements on security.

**Remote Monitoring (RMON** and **RMONv2)** are SNMP applications used for monitoring network elements.

**Common Management Information Protocol (CMIP)**, an OSI protocol used with the **Common Management Information Services (CMIS)**, supports information exchange between network management applications and management agents. CMIS defines a system of network management information services. CMIP supplies an interface that provides functions which maybe used to support both ISO and user-defined management protocols. The CMIP

specification for TCP/IP networks is called **CMOT** (CMIP Over TCP) [45] and the version for IEEE 802 LAN's is called **CMOL** (CMIP Over LLC). CMIP/CMIS are proposed as competing protocols to the Simple Network Management Protocol (SNMP) in the TCP/IP suite, but most vendors jumped on the SNMP bandwagon instead.

In **Web-based management (WBM**) [46], Web servers are embedded in the network devices. One can use a Web browser to browse the configuration and status of the device and also to configure it.

Web servers are more intelligent than SNMP agents and can be more sophisticated. WBM uses a centralized model and suits for small offices, whose management requirements are not that big. By using management applications you can enhance the WBM and, for example, filter out unwanted information. If there are devices on the network that use SNMP, then a Web agent can be used as a proxy.

**Web-based Enterprise Management (WBEM)** [47] is an initiative by Desktop Management Task Force (DMTF) trying to integrate all these standards (SNMP, CMIP and WBM). This makes the protocol used transparent to the management applications.

### 4.1.4. Quality of Service (QoS) in mobile network middleware

Quality of Service (QoS) determines the effect of service performance which determines the degree of satisfaction of a user of the service. QoS represents one of the most crucial issues as it involves many different aspects and directly impacts the user satisfaction. Some aspects of QoS, IntServ and DiffServ, are dealt with in section 3.8, under "Mobile Network Layer", but others need description here.

**MPLS, Multiprotocol Label Switching** (IETF RFC 3031 [48]) is an IETF initiative that integrates Layer 2 information about network links (bandwidth, latency, utilization) into Layer 3 (IP) within a particular autonomous system--or ISP--in order to simplify and improve IP-packet exchange.

**RSVP (Resource Reservation Protocol)** (IETF RFC 2205 [49]) an its extensions was designed to provides end-to-end QoS signaling services for application data streams. By original design, RSVP fits well into the framework of the Integrated Services (IntServ) (IETF RFC 2210 [50]) with certain modularity and scalability.

The IETF Next Step in Signaling (NSIS) working group is addressing the problem of signaling QoS, by defining the requirements, framework and protocols necessary to obtain such a goal. Existing QoS signaling protocols for IP networks has been analysed in an internet draf ([51]), and are presented as follows.

**YESSIR (YEt another Sender Session Internet Reservations)** is a resource reservation protocol that seeks to simplify the process of establishing reserved flows while preserving many unique features introduced in RSVP. Simplicity is measured in terms of control message processing, data packet processing, and user-level flexibility. Features such as robustness, advertising network service availability and resource sharing among multiple senders are also supported in the proposal.

**Boomerang** is a another resource reservation protocol for IP networks. The protocol has only one message type and a single signaling loop for reservation set-up and tear-down, has no

requirements on the far end node, but, instead, concentrates the intelligence in the Initiating Node (IN).

**INSIGNIA** is proposed as a very simple signaling mechanism for supporting QoS in mobile ad-hoc networks. It avoids the need for separate signaling by carrying the QoS signaling data along with the normal data in IP packets using IP packet header options. This approach, known as "in-band signaling" is proposed as more suitable in the rapidly changing environment of mobile networks since the signaled QoS information is not tied to a particular path.

**Border Gateway Reservation Protocol (BGRP**) is a signaling protocol for inter-domain aggregated resource reservation for unicast traffic.

**SICAP (Shared-segment Inter-domain Control Aggregation protocol)** is an inter-domain signaling solution that performs shared-segment aggregation on the Autonomous System (AS) level with the purpose of reducing state required at Boundary Routers (BRs).

**Dynamic Aggregation of Reservations for Internet Services (DARIS)** defines an inter-domain aggregation scheme for resource reservations.

Currently IETF NSIS WG (Next Steps In Signalling Working Group) is considering protocols for signalling information about a data flow along its path in the network. Based on existing work on signalling requirements, proposes an architectural framework [52] which protocol suite is structured in two layers: NTLP (NSIS Transport Layer Protocol) and NSLP (NSIS Signalling Layer Protocol).

**Generic Internet Messaging Protocol for Signalling (GIMPS)** [53], is a concrete solution for the generic NTLP defined in the framework. It specifies protocol stacks for the routing and transport of per-flow signalling messages along the path taken by that flow through the network. The solution uses existing transport and security protocols under a common messaging layer, which provides a universal service for diverse signalling applications. The **NSIS Signalling Layer Protocol (NSLP)** [54] is being designed for signalling QoS reservations in the Internet. It is in accordance with the framework and requirements developed in NSIS. Together with the NTLP, it provides functionality similar to RSVP and extends it. The QoS-NSLP is independent of the underlying QoS specification or architecture and provides support for different reservation models.

Another proposal for QoS specification is **End-to-end negotiation protocol (E2ENP)** [55] which enables negotiating and coordinating QoS on an end-to-end basis both at application and network layer. Based on a flexible extensible markup language (XML) model and extending SDPng concepts, the protocol enables the negotiation of system capabilities and allows provider-services to effectively influence the negotiation process. The aim of the E2ENP design is to optimize the efficiency of multimedia call setup and reduce the time for QoS renegotiations, whenever vertical handovers or spontaneous network reconfigurations occur.

### 4.1.5. Network Middleware

The major goal of WP4.2 which this section addresses is the design and implementation of a user-centred and programmer-friendly service platform for Akogrimo, where interoperability with heterogeneous networks is supported through the usage of the state-of-the-art middleware technology. This platform will provide user-centred abstractions of services for programming the NGG, in forms of APIs and SDKs, so current middleware technology state-of-the art is presented in this subsection.

Middleware solutions can be classified into several groups:

- Message oriented middleware (MOM) offers a basic set of commands with which to communicate over the network, often as few as SEND and RECEIVE. Application developers then create application-specific functions or routines built on top of these basic functions.
- RPC based middleware, which are procedure or function oriented. Developers define functions using an interface description language (IDL), and then compile that function into client and server stubs that actually do the networking.
- Database access middleware products offer data-oriented APIs to access databases.
- Distributed Transaction Processing (DTP) monitor middleware solutions enable handling transaction semantics over a network, by adding commands like BEGIN and END transaction regarding messages offered by MOM products.
- Object oriented based middleware enables object interactions, so it is the easiest middleware to employ because it manages directly objects created in the tool across the network.

Message and object oriented middleware seems to be the most suitable choice for the Akogrimo proposals.

#### 4.1.5.1. Middleware and Mobility

Mobility imposes several characteristics in the middleware regarding conventional middleware for fixed networks, like fixed /mobile networks integration, multiple and heterogeneous wireless devices supporting, continuous wireless access to contents/application provisioning (periodic unavailability – disconnections, adaptative bandwidth), E2E security, QoS and dependability from handled devices to application servers, which implies dynamic re-configurability (reflective middleware, self repairing systems)… In this sense IEEE Network has identified some interesting topics [56]:

- Application designers with a higher level of abstraction to achieve distribution transparency.
- Middleware support for micro/macro/multi-domain mobility
- Real-time and reflective middleware
- Middleware for peer-to-peer platforms
- Support of hierarchical heterogeneous environments (different protocols, resource discovery mechanisms, etc.)
- Distributed objects in reconfigurable pervasive applications
- Specification and enforcement of Quality of Service (QoS)
- Design of CORBA, .NET, and J2EE-based broker applications for mobile/fixed networks
- Management and programmability of distributed object systems
- Middleware for distributed and mobile agents
- Middleware security, including authorization and authentication
- OMG Model Driven Architecture (MDA) and its application to network middleware
- Patterns for distributed object design

- Middleware for network processors
- Programmable control plane and data path techniques
- Reliable and fault tolerant middleware
- Integration of distributed object and Web Service technologies, including SOAP interoperability and service discovery.

### 4.1.5.2. Middleware Standardization Forums

This section tries to provide an overview about active middleware standardization forums:

The **Object Management Group** (**OMG**, [57]) is an open membership, not-for-profit consortium that produces and maintains computer industry specifications for interoperable enterprise applications. It involves virtually every large company in the computer industry, and hundreds of smaller ones.

The **Open Mobile Alliance** (**OMA**, [58]) is focussed on the development of mobile service enabler specifications, which support the creation of interoperable end-to-end mobile services. OMA drives service enabler architectures and open enabler interfaces that are independent of the underlying wireless networks and platforms.

The **World Wide Web Consortium** (**W3C**) [59] develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential. W3C is a forum for information, commerce, communication, and collective understanding.

The **Open Service Gateway Initiative** (**OSGi**) [60] intends to specify, create, advance, and promote an open service platform for the delivery and management of multiple applications and services to all types of networked devices in home, vehicle, mobile and other environments.

The **Digital Living Network Alliance** (**DLNA**) [61] is focused on delivering an interoperability framework of design guidelines based on open industry standards to guarantee interoperability of wired and wireless networks of Personal Computers (PC), Consumer Electronics (CE) and mobile devices in the home enabling a seamless environment for sharing and growing new digital media and content services.

### 4.1.5.3. State of the Art

This subsection describes most common technologies and solutions regarding this topic.

**CORBA (Common Object Request Broker Architecture)** [62], a specification from the Object Management Group emerged as a promising middleware architecture for object communication in potentially heterogeneous and distributed environments. As the architecture developed and established itself as an industrial standard, a number of various implementations appear. Using the standard protocol IIOP, a CORBA-based program from any vendor, on almost any computer, operating system, programming language, and network, can interoperate with a CORBA-based program from the same or another vendor, on almost any other computer, operating system, programming language, and network.

Wireless CORBA [63] is an OMG specification that intends to provide wireless access and terminal mobility using CORBA.

The **MDA - OMG's Model Driven Architecture** [64]- is a new way of writing specifications and developing applications, based on a platform-independent model (PIM). A complete MDA specification consists of a definitive platform-independent base UML model, plus one or more platform-specific models (PSM) and interface definition sets, each describing how the base model is implemented on a different middleware platform.

The **Distributed Component Object Model** (**DCOM**) [65] is a distributed object technology from Microsoft that evolved from its Object Linking and Embedding (OLE) and Component Object Model (COM). DCOM's distributed object abstraction is augmented by other Microsoft technologies, including Microsoft Transaction Server and Active Directory. DCOM provides heterogeneity across language but not across operating system or tool vendor.

**SOAP** (**Simple Object Access Protocol**) [66] is a way to communicate programs running in the same or different operating systems by using HTTP and XML as the mechanisms for information exchange. Its specification is public, and it provides heterogeneity across both language and vendor. It has been proposed as a standard interface to the IETF (RFC 3288 [67]).

The **Microsoft .NET Framework** [68] is an integral component for building and running software applications and Extensible Markup Language (XML) Web Services. These components facilitates integration by sharing data and functionality over the network through standard, platform-independent protocols such as XML, SOAP, and HTTP. Microsoft's distributed object framework .NET also has heterogeneity across language and vendor among its stated goals.

**Java Remote Method Invocation** (**RMI**) [69] is a mechanism that allows methods of remote Java objects can be invoked from other Java virtual machines, possibly on different hosts. Java RMI has recently been evolving toward becoming more compatible with CORBA. In particular, there is now a form of RMI called RMI/IIOP ("RMI over IIOP") that uses the Internet Inter-ORB Protocol (IIOP) of CORBA as the underlying protocol for RMI communication.

**Java 2 Platform Enterprise Edition** (**J2EE**) [70] is a platform-independent, Java-centric environment from Sun for developing, building and deploying Web-based enterprise applications online. The J2EE platform consists of a set of services, APIs, and protocols that provide the functionality for developing multitiered, Web-based applications. It is a basic piece of the SUN Java 2 platform, which comprises as well as mentioned J2EE, **Java 2 Standard Edition** (**J2SE)** [71] for desktop applications and **Java 2 Micro Edition** (**J2ME**) [72] for consumer and embedded devices.

**Jini Networks Technology** [73] is a Java based technology that enables the creation of highly-adaptative network-centric services. It provides an environment for creating dynamically networked components, applications, and services that scale from the device to the enterprise. It comprises technologies such JavaSpaces Technology and Jini extensible remote invocation (Jini ERI).

**JXTA** technology [74] is a set of open, generalized peer-to-peer protocols that allow any connected device (cell phone, to PDA, PC to server) on the network to communicate and collaborate. It intends to provide interoperability across different P2P systems, platform independence (SOs, programming languages…) and ubiquity.

**IBM WebSphere** [75] provides middleware to set up, operate and integrate e-business applications across multiple computing platforms using web technology. It has been constructed using open standards such as the Java 2 Platform, Enterprise Edition (J2EE), XML and Web Services standards.

**Microsoft Message Queuing** (**MSMQ**) technology [76] enables applications running at different times to communicate across heterogeneous networks and systems that may be

temporarily offline. MSMQ provides guaranteed message delivery, efficient routing, security, and priority-based messaging, and it can be used for both synchronous and asynchronous scenarios.

**BEA MessageQ** [77] is a middleware solution for distributed enterprise applications that allows the reliable exchange of guaranteed application messages across heterogeneous platforms,like UNIX, Windows, Win NT, OpenVMS, IBM, and Unisys mainframes.

**TIBCO ActiveEnterprise** [78] provides an e-business infrastructure for integrating applications, databases and other sources of information. Applications can select from several qualities of service including reliable, certified and transactional, as appropriate for each interaction. It constitutes a robust and fully .NET managed implementation of the **Java Message Service** (**JMS**) [79]. Messages are self-describing and platform independent, with a user-extensible type system that provides support for data formats such as XML.

**JORAM** [80] is an ObjectWeb (open-source software community focussed on middleware [81]) open source solution based on a pure Java implementation of JMS . Provides support to local, TCP and SOAP (HTTP/XML) client-server communication protocols and allows J2ME applications to access the JORAM platform in a standard JMS way.

**Open Source Message Queue** (**OSMQ**) [82] is an advanced, pure Java, asynchronous message middleware framework developed by Boston Systems Group. BSG has chosen to release the product as open source, using the GNU public license, with an interface designed to be less complex than JMS.

**XmlBlaster** [83] is an open source, multiplatform, multiprotocol, language neutral and XML-based message oriented middleware.

## 4.2. Security

For ensuring a healthy infrastructure for Grid Services every entity that it is using it must obey a set of rules. One of the main tasks of the Network Middleware layer is to define such rules, strictly monitor if they are respected and take necessary measures when abuse is detected. Defining the good and bad behaviour and its monitoring and accountability is the task of the AAA subsystem.

### 4.2.1. Security Challenges for a Mobile Grid Architecture

Security requirements within the Grid environment are driven by the need to support scalable, dynamic, distributed virtual organizations. The main challenges for a mobile grid architecture can be grouped in the following categories:

- *Heterogeneous distributed environment* - in a heterogeneous environment of different types of devices running different software, participation in the grid necessitates management of transparent access.
- *Multiple security mechanisms* – a platform that spreads across several physical and logical network domains having different access technologies should provide the necessary mechanisms to facilitate the interoperability of the different security architectures

- *End-to-end security* - cooperating systems with different security policies and protocols will have to negotiate trust arrangements in order to provide end-to-end security (identification, authentication and authorization)
- *Dynamic creation of services* – users must be able to create new services and resources dynamically without the intervention of an administrator. These services must be coordinated and must interact securely with other services.

### 4.2.2. Security Requirements

In order to provide quality security services at the Network Middleware Layer, the following requirements should be addressed:

*Authentication* – for enabling interoperability the platform should provide plug points for multiple authentication mechanisms

*Authorization* –access *to* grid services must be controlled based on authorization policies attached to each service. It should accommodate various access control models and implementations

*Delegation* – establishment of dynamic trust domains requires facilities to allow for delegation of access rights from requestors to services.

*Single sign-on* – participants in a grid environment often need to coordinate multiple resources to accomplish a single task. Security mechanisms have to ensure that once a successfully authentication is performed no need for re-authentication is required.

*Secure logging* – facilities for time stamping and mechanisms for securely logging any kind of operational information or event should be provided. The term *securely* in this context means reliably and accurately so that this information cannot be altered by inappropriate agents.

*Privacy* – both service requester and service provider must be allowed to define and enforce privacy policies.

*Manageability* – security management in Grids is needed, such as: identity management, policy management, key management.

### 4.2.3. Current Grid Security Technologies

Addressing those security considerations outlined above, current approaches in the grid technology arena are investigated. The following set of existing approaches encompass:

- *GSI - Grid Security Infrastructure* The Grid Security Infrastructure (GSI) offers secure authentication and communication over an open network for the Globus Toolkit. It provides a number of useful services for Grids, including mutual authentication and single sign-on. It is based on public key encryption, X.509 certificates and the Secure Sockets Layer communication protocol.
- *GT2 (Globus Toolkit 2) Security model* – GT2 provides services for Grid Resource Allocation and Management (GRAM), Monitoring and Discovery (MDS), and data movement (GridFTP). They all use a GSI infrastructure to provide security functionality.

- *GT3 OGSA Security model* - is an OGSA based security model defined and implemented within Globus Toolkit 3 [85]. It uses Web Services security policies to address the grid security requirements.

### 4.2.4. State of the Art

**AAA (Authentication, Authorization, Accounting)** [86] is a framework for controlling the access to computer resources, enforcing policies, audit usage and providing the necessary information for billing the usage of services

**RADIUS (Remote Authentication Dial-In User Service** [87] [88]) is an Authentication, Authorization, and Accounting (AAA) protocol for applications such as network access or IP mobility. It enables remote access servers to communicate with a central server to authenticate dial-in users and authorize their access to the requested system or service. It is intended to work in both local and roaming situations. RADIUS was originally developed by Livingston Enterprises for their PortMaster series of Network Access Servers, but later (1997) published as RFC 2138 [87] and RFC 2139 [88]. Now, several commercial and open-source RADIUS implementations exist. Features can vary, but most can look up users in text files, LDAP servers, or various databases. Accounting tickets can be written to text files, various databases, or forwarded to external servers. SNMP (section 4.1.3) is often used for remote monitoring. RADIUS Proxy servers are used for centralized administration and can rewrite RADIUS packets on the fly (for security reasons, or to convert between vendor dialects). RADIUS is extensible; most vendors of RADIUS hardware and software implement their own dialects. .

**DIAMETER** [89] evolves from RADIUS scheme by adding new functionalities such as the ability to ask for additional logon information beyond the basic authentication. It also supports user roaming in a MobileIP environment. It consists of a Base Protocol defined by RFC 3588 and a set of additional Diameter applications. The Base Protocol complies with the AAA Transport Profile (RFC 3519). Examples of Diameter applications currently in preparation are:

- Mobile IPv4 Application (MobileIP)
- Network Access Server Application (NASREQ)
- Extensible Authentication Protocol Application (EAP)
- Credit-Control Application (CC)
- Session Initiation Protocol Application (SIP)
- various applications in the 3GPP IP Multimedia Subsystem (IMS)

**TACACS (Terminal Access Controller Access Control System)** [90] is a security protocol that provides a centralized system for controlling the access to network resources. In the context of Akogrimo could be used as a mechanism to interconnect legacy systems. It is defined in RFC 1462.

**Kerberos** [91] is a protocol that provides strong authentication for client/server applications by using secret-key cryptography in a networking environment. The Kerberos protocol uses strong cryptography so that a client can prove its identity to a server (and vice versa) across an insecure network connection. After a client and server has used Kerberos to prove their identity, they can also encrypt all of their communications to assure privacy and data integrity as they go about their business.

**LDAP (Lightweight Directory Access Protocol) [92]** can be used to retrieve directory-based information such as email addresses or public keys in centralized, corporate-wide directories.

**PKI (Public Key Infrastructure)** is a combination of several software, encryption technologies, and services that enable enterprises to secure their communications and business transactions on the Internet. PKIs integrate digital certificates, public-key cryptography, and certificate authorities into an enterprise-wide network security architecture.

**Digital certificates** are attachments to electronic messages used for security purposes. The most common use of a digital certificate is to verify that a user sending a message is who he or she claims to be, and to provide the receiver with the means to encode a reply.

**X.509** [93] is a ITU-T standard for Public Key Infrastructure (PKI). It specifies standard formats for public key certificates and a certification path validation algorithm. It assumes a strict hierarchical system of Certification Authorities (CAs) for issuing certificates instead of a web-of-trust model (like PGP), where anyone can sign and attest the validity of other's key certificate. The term, **X.509 certificate** usually refers to the IETF's profile of the X.509 v3 certificate standard, as specified in RFC 3280 [93].

**XKMS (XML Key Management Specification) [94]** defines a Web Services interface to a public key infrastructure. This makes it easy for applications to interface with key-related services, like registration and revocation, and location and validation. XKMS is a foundational specification for secure Web Services, enabling Web Services to register and manage cryptographic keys used for digital signatures and encryption.

When combined with WS-Security, XKMS makes it easier than ever for developers to deploy enterprise applications in the form of secure Web Services available to business partners beyond the firewall.

### 4.2.5. Recommendations

Security considerations shall be taken in all of the four layers in the Akogrimo platform. The Network Middleware shall be responsible of secure authentication of users, offering access to different services based on user credentials and keeping track of all the events occurred, for accounting and billing as well as for security purposes.

The AAA module should be based on the DIAMETER protocol, since it has a strong support for mobility and it is very easily scalable for any other future requirements for the AAA module. Several gateways should be provided for interaction with legacy AAA systems like RADIUS and TACACS which are largely used by network operators. The overall AAA architecture shall be layer-independent with extension modules capable of offering layer-specific functions in each of the four layers.

## 4.3. Charging and Pricing

Based on the necessity of having an un-contradictory terminology, the following definitions with respect to charging and pricing in Akogrimo are utilized in this section. These definitions are based on previous work performed and found in [95] [100] and others.

- ***Metering***
  *Meters are needed for capturing data about resource consumption in the network (e.g., transmitted volume). Applied to services and Grid services, resource consumption may require the metering of service-specific parameters as well.*

- ***Accounting***
  *Accounting describes the collection of data about resource consumption and service usage. This includes the fully decentralized control of data gathering (via metering), transport, and storage of accounting data.*

- ***Charging***
  *Charging derives monetary values for accounting data sets based on service and customer-specific tariff parameters.*

- ***Billing***
  *Billing translates all costs calculated by Charging into a unit of billing (billing record) and generates a final bill for the customer, which may be delivered electronically or in traditional terms on paper in various pre-defined periods, such as once per day, per week, or per month.*

- ***Pricing***
  *Pricing defines the process of determine a monetary value for a good or service to be charged on a per-item, per-usage, or per-time basis. The tariff applied to a good or service usage needs to be pre-determined.*

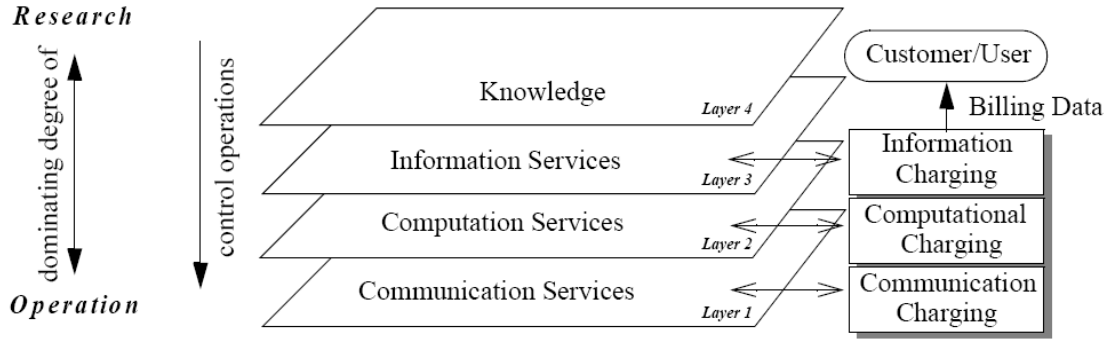


**Figure 1 - Charging Interfaces for a Computational Grid Architecture**

## 4.3.1. Resource Charging

As proposed in [95], a 4 layer Grid Charging Architecture allows for a differentiation between all influencing factors of the charging process. These layers can be seen in Figure 1 and are shortly explained in the following:

- *"Communication Services"* layer deals strictly with communication aspects of the grid and has its own interface for communication charging process. Resources that could be charged for at this layer might include volume of traffic, network usage in time, number of packets, etc.

- *"Computation Services"* layer deals with computational issues and applies the charging function for services like CPU time, storage, memory consumption, etc. This layer also has an interface for accessing the charging information specific for this layer

- *"Information Services"* layer integrates application specific information and services and provides a wholesale type of grid-layer. Depending on the application, this layer may offer services to maintain distributed databases , support distributed simulation tools, enable distributed simulation programs, operate new learning technology scenarios or provide collaborative work spaces. At this layer, communication with local accounting tools is required to allow the calculation charges for a full end-to-end service of the Grid [95].
- *"Knowledge"* layer includes mainly user, customer or application-oriented tasks that will be mostly defined by users, while applying services from the information and computation layers explicitly. A charging interface is not needed at this layer.

## 4.3.2. Internet Pricing Schemes

Currently there are several charging and pricing schemes for mobile communications and the Internet. Most of them can be grouped in the following three categories:

- Differentiated Service Charging
- Usage based charging
- Dynamic charging

### *4.3.2.1. Brief summary of pricing and charging schemes*

In [96], an evaluation of some of the current pricing and charging schemes used in IP networks is included. Key criteria include: compliance with existing technologies, measurement requirements, support for traffic management, provision of individual QoS, degree of network efficiency, degree of economic efficiency, impact on social fairness, and pricing time frame.

The following text contains a brief summary of the review. For a more detailed evaluation the reader is referred to paper [96].

- **Flat Pricing** [97][98] is a pricing scheme that charges the user a fixed amount per time unit (e.g. month), irrespective of usage. It is a simple and convenient pricing scheme especially for charging network access. It makes no assumptions on the networking technology that is already deployed. Since charging is not related to usage, no measurements are required for accounting and billing. It also does not explicitly support individual QoS guarantees to the individual user.
- **Paris-Metro Pricing** [99] is a scheme where the network is divided in several logical networks, each of them having a different associated cost. Users can choose one of these logical networks and thus define the service level expected.
- **Priority Pricing** [99] requires a priority field in every packet header. Based on this field a user can indicate the value of their traffic by selecting a priority level, thus selecting the price level of the service. The IP protocol already provides the priority field, so compatibility with the current technologies is assured. Measurements are required for keeping track of the priority level of each transmitted packet. Priority pricing raises the economic efficiency of the network.
- **Smart-Market Pricing** [102] focuses on the issues of capacity expansion and social cost imposed on other users [96]. Besides the fixed charge for covering the connection cost and a charge for each packet transmitted, this scheme introduces a usage cost when the network is

congested. The user associates a price for each packet it transmits and places this value in the packets' header. This information contains a user's willingness to pay for a transmission. The network makes an auction for underlying services prices and chooses the best option for a packet. The introduction of the auction mechanism requires some important technology changes to protocols and networking hardware.

- **Edge pricing** [103] uses locally computed charges based on simple expected values of congestion or route. An example could be the charging based on the time of the day usage. Such a scheme is very simple and permits the charging to be applied on the edge of the network.
- **Cumulus Pricing Scheme** [103][104] proposes a solution for charging based on DiffServ technology. It tries to out pass the problems of other Internet charging models which assume a single-service best-effort network that provides a similar service to all its users. Prices in this scheme are based on flat fees, but the scheme is flexible enough to allow network management according to the actual forces that steer the market. CPS defines a clear relation between different time-scales of accounting periods, measurement periods and charging periods.

#### *4.3.2.2. Mobile Operators Charging and Billing Models*

Since the current mobile communication technologies seem to converge to a technical platform based on a TCP/IP network, it is expected that the mobile operators will embrace some of the pricing schemes from the Internet community.

### 4.3.3. Akogrimo approach

The Akogrimo Project aims to unify two different worlds: Mobile World and the Grid World. A common approach for a charging model will not be easy to define, since at the moment grids are used more for scientific progress than big revenues. Several existing models should be taken into account and possibly combined for a consistent charging model used in both worlds.

Another important issue is the ability to offer a unified bill for mobile grid services. A Grid Service requested by a user will probably consume resources from several different providers. Each of those providers expects revenue for the usage of its resources. The players involved in the service as well as the different costs for delivering the service should be transparent for the end-user. Although some services may be paid for separately, it is in most cases desirable that a single bill shall be issued to the end-user containing the costs for the entire service that was provided to him.

Issues relating specifically to Grid accounting are covered in section 7.5.

## 4.4. Pervasive computing

The goal of pervasive computing is to enable services that are optimally useful for any user in any situation. To achieve this universal utility, pervasive services should have particular characteristics:

- **Mobility**: The service should be continuously available even if the terminal (and its owner) is moving from place to place (terminal mobility). The service should be available to the user on any terminal instance and type (user mobility).
- **Context awareness**: The service should adapt its behavior to be optimal for the current context of the user. The service should also be able to respond dynamically to changes in that context. Context is a very broad term, encompassing issues such as the current location of the user, the geography of that location, network infrastructure and I/O devices available, the activity in which the user is engaged, the user mood and so on.
- **Personalization**: The behavior of the service should reflect the individual preferences of the users. These preferences may be specified explicitly by the user, or may be derived from analyzing the user's habits.

### 4.4.1. Pervasive computing reference model

The figure below shows a reference model for pervasive applications. Context is initially determined from sensor input (e.g. location, time and wireless network signal strength) and explicit context indications from the user, e.g. setting the mobile in "meeting" mode. Further information on the current context is extracted from a variety of sources of contextual data. These include directory services and geographical information systems (GISs).

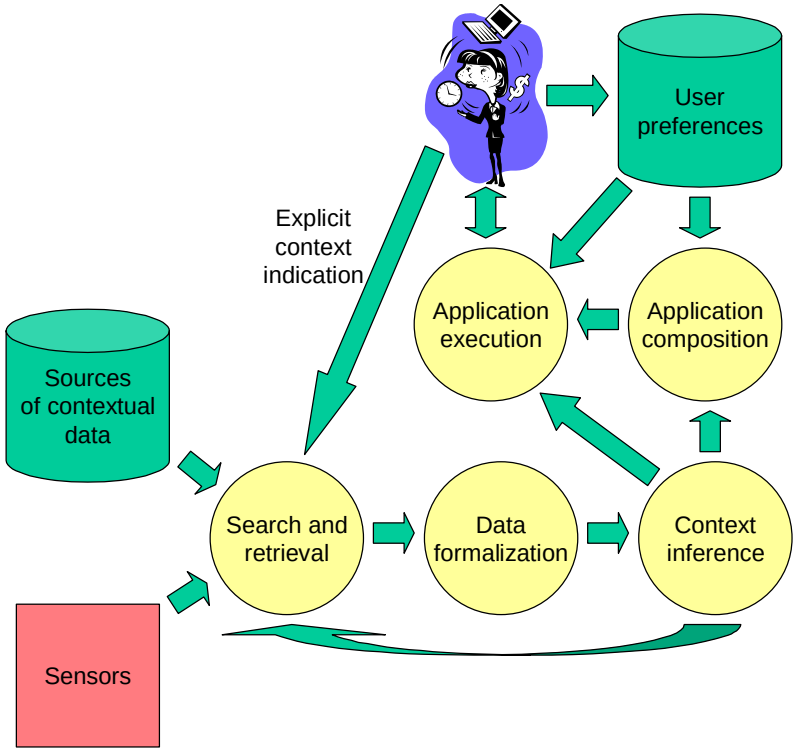


**Figure 2 - pervasive computing reference model**

The context data are formalized for further analysis through methods such as ontologies or standardized data models. From these formalized data, further knowledge about the user's context can be deduced, and this can be fed into an iterative search for further context data (e.g. knowing that the user stands close to an Italian restaurant could trigger a search for their menu and information on the nutritional value of Italian food). When knowing the context, an application can be composed of the most suitable services (network services and higher level services). The context is also fed into the application execution, such that the application can behave in the optimal way in the current context. The entire process is dynamic, such that the system can continuously adapt itself to context changes. There are functions for managing explicit and implicit user preferences. These also serve as input for application composition (e.g. use the cheap but low-bandwidth connection) and application execution (e.g. download low-resolution images).

### 4.4.2. Context-awareness

A context-aware service is a service that is able to adjust its behavior to the current context of the user in order to provide maximal utility within that context. There are many aspects of context:

- **Available networks:** What networks are available and what kind services do they offer in terms of bandwidth, QoS guarantees etc. What tariffs apply?
- **Terminal capabilities:** What are the I/O and computational capabilities of the terminals that are currently available to the user?
- **Location and geography:** Where is the user located and what are the conditions at this location? Which infrastructure and what services are available there, and which threats and annoyances?
- **Activity:** What is the user doing?
- **Physiological and mental state:** What is the bodily state of the user and what is his mood?
- **Time and date.**

#### *4.4.2.1. Positioning technology*

Positioning is the process of measuring where a terminal is located (and hence the position of the user using that terminal). Several technologies may be employed for determining terminal position. Some key technical attributes of such technologies are:

- **Accuracy:** What is the deviation between the position given by the positioning system and the actual position? What is the deviation between repeated measurements for the same (stationary) terminal? What is the deviation between two terminals in the same location?
- **Coverage:** What geographical area does the system cover? Can it be used indoors? Does it require line-of-sight between the terminal and base stations or satellites?
- **Speed:** How long must a terminal be in a given position for that position to be measured?

Positioning systems of interest for pervasive applications can be classified as either satellite navigation systems, wireless communication networks, or smart tags.

Two satellite positioning systems are available for general use, the US GPS (Global Positioning System) (cf. [109]) and the Russian Glonass. The European Galileo system should be operational by 2008. The GPS allows positions to be determined with an accuracy of around 10 meters for civilian receivers. If a correction signal from a fixed nearby transmitter is available, the accuracy is about 3 meters. This arrangement is know as DGPS (Differential GPS). DGPS is available in most populated areas in developed countries. The Galileo system promises an accuracy of around 5 meter for the Open Service (free of charge), and around 1 meter for the Commercial Service (for a fee). Combined receivers, utilizing both GPS and Galileo signals, should improve accuracy and coverage even further. Both systems have global coverage, but require line-of-sight between the terminal and a number of satellites. For that reason, they will have limited value indoors and may also have reduced accuracy in difficult terrain such as between high-rise buildings. Galileo attempts to remedy these problems by placing the satellites in higher orbits (where they are less likely to be obstructed by obstacles) and by encoding signals in a way requiring lower signal strength (improving their usefulness indoors).

Wireless communication networks are systems that were designed for voice and/or data transport rather than positioning. Such a network consists of fixed base stations and mobile terminals. To perform their transport service, most such network needs to have some notion of terminal positions. This information can then be used for positioning applications. For efficient routing and bandwidth usage, the network needs to know which base station that has the best radio connection to the terminal. If the base station uses directional antennae, the network must also know which sector around the base station that provides the best connection to each terminal. Some multiplexing schemes, such as CDMA (Code Division Multiple Access) and TDMA (Time Division Multiple Access) also require the round trip time between the terminal and the base station to be known. So, depending on the network technology, a given terminal may be placed in the vicinity of a specific terminal, within a sector around that base station, and/or at a specific distance from that terminal.

In the Global System for Mobile communication (GSM) the maximal range of a base station is 35 kilometres[1]. In dense urban areas, the radius served by a single base station may be a kilometre or less, and this circle may be split in a number of sectors (typically 2-6) using directional antennae. Each such sector constitutes a GSM cell. The GSM radio interface specification also requires the base station to know the round-trip time to each terminal (and hence the distance). The SS7 standard specifies mechanisms for accessing positioning data. Depending on the implementation, that position may merely be the centre of the cell, or it may also reflect the measured distance between the base station and the terminal. For a GSM system that implements the Open Systems Access (OSA) location APIs, these may also be used for accessing location data.

In wireless LANs based on the 802.11x standards, each base station has a range of up to about 400 meters, depending on the particular standard being used, and the topology of the site. Base stations generally do not use directional antennae, which would anyway be of limited use indoors due to reflections. A better estimate of terminal distance can be derived form measuring signal strength. Under optimal conditions and after manual calibration at multiple points in the target area, a precision of 1-3 meters can be obtained. There is no standardized way of making such

[1] There are solutions with extended range at the expense of lower spectral efficiency for rural areas..

measurements, and current solutions rely on proprietary client software. Actual results will depend on the number of base stations that are within range, and the topology of the target area.

Bluetooth radio interfaces has a range of about 10 meters and can thus potion a terminal within a 10-meter sphere. There is no standardized way of obtaining measurements of signal strength or round-trip time to get a better position estimate. By having several fixed bluetooth devices at the same site, a terminal can be positioned within the intersection of several such 10-meter spheres, to obtain better accuracy, as demonstrated by [105].

Smart tags or RFID tags are intended to replace bar codes for tagging objects with a machine-readable label. RFID is a concept rather than a standard, and there are several incompatible competing technologies. The maximum distance between a tag and a reader can from less than a meter and up to a few meters, depending on the implementation.

| **Technology** | **Coverage** | **Accuracy** | **Indoor use** |
|---|---|---|---|
| Satellite positioning | Global | 1-10 meters | Limited |
| GSM | Populated areas in most developed and some developing countries | 1-35 km (>400 m with triangulation, cf. [110]) | Yes |
| UMTS | Urban areas in some developed countries | A few hundred meters (>30 with triangulation) | Yes |
| Wireless LAN | Up to 400 meters from base station | 100s of meters (1-3m with multiple base stations and signal strength measurements) | Yes |
| Bluetooth | 10 meters from reference device | 10 meters. | Yes |
| RFID | Up to a few meters from each reader | From less than one to a few meters. | Yes |

**Table 2 - Characteristics of available positioning technologies**

#### 4.4.2.2. Storage and retrieval of temporal and geo-spatial data

Location based services need mechanisms for efficiently accessing and analyzing geographical data in order to determine the context on the user's location. Geographical data could be mapped to primitive data types such as floating point numbers for coordinates and stored as such in a relational database. But searching in such schemes, using one index per coordinate, is inefficient.

For that reason, the database is augmented with spatial data types, such as 3D points and polygons. Specialized index types, R-tree, quadtree, and B-tree with grid indexing, allow efficient searches in spatial data (cf. [111]). A database supporting spatial data types can also offer spatial functions to determine containment and overlap between objects, permitting queries such as: "Find all gas stations in area X". Commercial database products with spatial data features are available from Oracle, IBM, and others. Spatial databases may also be used for modelling temporal aspects of data, by mapping time to a fourth dimension.

#### 4.4.2.3. Determining terminal capabilities

By having some mechanism for negotiating which media formats to use in a multimedia session, there is better chance of finding media formats that both parties can (and will) use.

The W3C has defined the Composite Capabilities/Preference Profiles (CC/PP) framework (cf. [112]). The typical use of CC/PP is to adapt a web page to a handheld terminal with limited resources. CC/PP is based on the Resource Description Framework (RDF), which is a metadata modelling language. CC/PP can specify such data as screen size, number of colors, supported Java VM versions etc. The model is extensible, allowing new classes and attributes to be added. An HTTP request may contain an URI referring to a CC/PP profile stored on an arbitrary WEB server, eliminating the need for transferring the profile over the wireless link. If the profile changes, the client may send only the differences since the last request

The UAProf standard, defined by the Open Mobile Alliance, is a subset of CC/PP, and is specifically aimed at WAP sessions.

The Synchronized Multimedia Integration Language (SMIL) 2.0 specification defined by the W3C is an XML-based language for defining interactive multimedia presentations (cf. [114]), typically consisting of audio and video streams to be played in parallel or sequence. SMIL has also been adopted by the 3GPP (which defines UMTS standards) for the *Transparent end-to-end Packet.switched Streaming Service* (PSS). The *Content Control Module* of SMIL defines a syntax for describing alternative (e.g. versions of the same video stream with different bit rates) and conditional media streams, such that the client may choose the most appropriate set of media streams for that terminal.

The Session Initiation Protocol (SIP) is a signalling protocol for IP telephony and other types of multimedia sessions. SIP normally uses the Session Description Protocol (SDP) to negotiate media formats (cf. [115]). SDP was originally developed to be used with the Session Announcement Protocol (SAP) for announcing multicast sessions. In that arrangement, the media formats would be chosen unilaterally by the broadcasting entity, and not negotiated. SDP hence lacks mechanisms for specifying choices, alternatives and preferences.

### 4.4.3. Application personalization

Different users will use applications in different ways. If an application can adapt its behaviour to the preferences of the individual user, it will be more useful for him or her. The user can explicitly specify such preferences, or they can be derived form analyzing the user's past behaviour.

#### 4.4.3.1. Managing explicit user preferences

Personalized applications will need some mechanism for storing and accessing user preferences. Several standards and proposals exist. Some distinguishing features of these are:

- **Data model:** XML schema, RDF (Semantic WEB model), others.
- **Data location:** Profiles could be stored in the terminal, in specialized profile servers or both. The profile may be stored centrally or distributed across different systems.
- **Access control mechanism:** How to control who can access which data?

The CC/PP and UAProf standards described in section 4.4.2.3 are also designed to convey data describing user preferences in the same way as terminal capabilities. These standards define no access control mechanism, but specify that some external mechanism must perform access control.

The Generic User Profile (GUP) is under standardization by the 3GPP (cf. [113]). Preferences are described in a flexible and extensible format based on XML Schema. GUP defines access control mechanisms to restrict who can access which parts of the profile. A GUP profile may be distributed across different servers in different organizations (linked together through URIs).

The Lightweight Directory Access Protocol (LDAP) is a protocol for accessing X.500 directory services[2] (cf. [116]). LDAP is a simplification of the original X.500 directory access protocol, using TCP instead of the OSI stack, and omitting some of the more complex mechanisms of the original protocol. X.500 is a distributed, hierarchical directory mechanism for storing data about any type of entity. Each entry belongs to a class with typed attributes. The scheme is extensible, such that new classes and subclasses may be defined. LDAP supports encryption and authentication through X.509 and SSL, and access control list may be defined to restrict access to specific objects or attirbutes.

### 4.4.4. Pervasive terminals

Pervasive services should be available to the user in any situation, and must thus be able to utilize whatever terminal is there. This could mean either using mobile terminal that the user brings with him or stationary terminals in the users current location that the service can discover and press into service.

#### 4.4.4.1. Portable and wearable devices

Mobile terminals have limited resources compared to stationary equipment such as workstations or terminals for nomadic use such as laptops. The mobile devices typically have less network bandwidth, smaller screens, simpler input devices, less storage capacity, and less processing power. For that reason, applications and runtime environments aimed at stationary devices will

[2] Or indeed any directory service supporting the X.500 features required by LDAP.

often perform poorly on mobile devices. Several initiatives have thus attempted to create application environments suitable for mobile devices.

The Wireless Application Protocol (WAP) stack, which is standardized by the Open Mobile Alliance (cf. [117]), defines protocols for mobile, networked applications. The Wireless Profiled Transport Control Protocol (WS-TCP) optimises TCP for wireless links. A gateway (WAP proxy) translates between WS-TCP and regular TCP. The WAP stack also includes XHTML MP (Mobile Profile), which is a simplified hypertext mark-up language for making WEB pages for mobile devices.

The Symbian operating system (cf. [118]), defined by the Open Mobile Architecture Alliance (cf. ), was designed specifically for mobile phones. Symbian incorporates the WAP stack and APIs for hardware typically found on mobile phones such as Bluetooth, infrared interfaces, and keypad or stylus input devices. Symbian also has mechanisms for speech and handwriting recognition. The system has small resource requirements compared to operating systems for stationary terminals.

#### 4.4.4.2. Transcoding of content

Transcoding denotes the process where content is translated from one format to another. There are several types of such translations:

- Between different standards for representing the same media, e.g. from MPEG2 to MPEG1.
- To a different media, e.g. text to speech.
- To a lower fidelity, e.g. reducing video resolution or lossy compression of images.

Pervasive services use transcoding to adapt content to device capabilities, network bandwidth or user preferences. The success of the world wide web and the proliferation of handheld terminals has lead to a particular interest in techniques for transcoding WEB pages for such terminals. HTML transcoding methods can be classified as syntactic or semantic.

Syntactic methods apply a set of rules that transforms the page according to its syntax. For example, a page containing headings followed by body text can be split into a master page containing only headings and a subordinate page for each paragraph.

Better transformations can be made if the semantics of the page can also be taken into account. This method requires some form of semantic annotation to be present, and is thus less general.

The eXtended Stylesheet Language (XSL) of the W3C is syntax for declaring transformations of XML documents into other representations, including (X)HTML. A document must then be authored as an XML where tags identify different semantic elements in the document, and different style sheets will translate this semantic representation into HTML suitable for different devices and preferences.

IBM (cf. [124]) suggests an approach using separate semantic annotation documents, based on the RDF language. The original HTML document will then be transformed into a new document based on the annotations and rules from e.g. a CC/PP profile. The annotation document could for example have a rule stating that the rightmost HTML frame in a page contains advertisements. This method can be applied to exiting web pages without re-authoring them, but the annotation document must be updated when the web page changes.

### 4.4.5. Service discovery

Pervasive applications will typically be used in a dynamic setting where resources will come and go or change, and where the optimal composition of an application will depend on a context that is also dynamic. For those reasons, an application cannot be statically composed. Instead there must be mechanisms for dynamically discovering services (resources) and composing (or re-composing) the application from those services.

For run-time service composition to be possible, there must be a mechanism for services to announce their existence and properties, and clients wishing to compose a service must be able discover the existence of those services. Such publish/discovery schemes can be classified according to criteria such as:

- **Local or global:** Is the scheme aimed at finding resources close to the user, or globally?
- **Peer-to-peer or centralized:** Do clients and services find each through direct communication, or is there a central directory serving as a repository of available services?
- **Support for notifications:** There may or may not be a mechanism for notifying a client when resources of interest become available or change properties.

Microsoft's UPnP protocol (cf. [106]) is typically used for finding peripherals such as printers, file servers or smart home appliances close to the user. UPnP operates only in peer-to-peer mode; there is no directory. Services publish their existence through IP broadcast messages, and clients also use broadcast requests to find available services. Services are described by XML documents that specify the service type and all supported methods and parameters. Clients can subscribe to notifications of changes in properties of specific services.

Universal Description Discovery and Integration (UDDI) defines a directory mechanism for locating Web Services (cf. [107]). UDDI aims to create an open, global market for Web Services. Clients search for services using the URIs of XML services descriptions, as well as other metadata. Services are described as Web Services Description Language (WSDL) documents, which identify the type, operations and parameters of the service.

Bluetooth is a personal area network standard, for wireless interfacing between devices such as mobile phones, headsets and laptops. The Bluetooth Service Discovery Protocol (SDP) is used for finding available services (cf. [108]). Bluetooth devices can search for services using type identifier and other metadata. The Bluetooth standard specifies a set of device types (such as those listed above) and the methods they should support. Third parties may define additional types.

## 4.5. Knowledge discovery and collaboration support

The introduction of mobile grids will inevitably bring about the issue of representing and supporting human resources in the grid infrastructure. Mobile resources in a mobile grid include mainly human users (humans are mobile), while grids traditionally have focused on computing resources. The difference will be in coping with the spontaneous nature of human cooperation and allowing human beings to participate in problem solving and grid processes. For instance, in

our disaster handling scenario, it is the human participant with their mobile devices that are the main users of the infrastructure. Grid technologies provide a flexible infrastructure for incorporating human users in systematic and ad hoc ways. In this view, a virtual organization contains computing resources in seamless cooperation with human resources. This chapter outlines a number of technologies that are of importance for supporting the human-human cooperation that will happen in the mobile grid.

The two types of cooperation that will be covered in this chapter are:

- *Ad hoc cooperation*: In this case cooperation is not planned in advance and happens in an ad hoc way following spontaneous encounters among human resources in the mobile grid. For allowing this ad hoc cooperation to happen, the mobile grid will employ a presence and availability model for the human resources (to be streamlined with that of computing resources) where criteria for "seeing" other human resources will be connected to the task that the user of the mobile grid wants to perform. This presence model will be augmented with an expertise discovery tool to allow the discovery of "the right experts" to participate as human resources.
- *Planned cooperation*: As soon as a virtual organization is in place, cooperation will be happening within a workflow of steps initiating and performing different parts of the task. In order to allow human resources to participate in such a workflow, flexible cooperation support tools such as web-based shared workspaces will be needed. The role of a shared workspace is 1) to allow human resources see and modify the current status of the task, and 2) to allow them to communicate with other human resources.

Discovery affects multiple layers and is also described in section 7.3.

### 4.5.1. Tools for ad hoc cooperation

Ad hoc cooperation is characterized by the fact that the cooperation is not planned but happens in a spontaneous way. Examples mostly referred to are colleagues meeting in the office building corridors and lunch rooms, and exchanging information that will initiate further cooperation. This form of ad hoc and informal cooperation is assumed to constitute the major part of all human-human cooperation [119] and is supported by a number of highly popular tools:

- *Presence and availability tools*: These tools allow people to see the status of other people. In this way the basis for ad hoc cooperation is put in place (i.e. by seeing his/her status one can judge the availability of the person). In addition, it is shown that seeing people will by itself initiate cooperation (therefore the proverb "out of sight, out of mind"). The most known examples are various types of the popular "messengers" [120], called so because they often support text-based messaging in combination with presence and availability.
- *Instant communication tools*: These tools are mainly the known communication tools (i.e. text messages, phone calls, video calls) used in combination with presence and availability tools. They are called instant because the initiator of communication knows *a priori* whether the communication will be successful. For instance, when calling a person the caller is not sure whether the person is available, while using a tool such as Skype [121] the user can have some information about whether the call will be successful. The same applies for instant messaging (IM) and chat tools where the communicating parts can "see" each other before initiating the communication.

- *Expertise discovery tools*: Although there exist extensive work in the area of resource modelling and discovery, the mobile grid will need to expand this work with that of discovery of human expertise and experience. Human resource discovery is fundamentally different from computing resource discovery because of many reasons, e.g. human expertise is often not modelled in a formal way, human expertise is constantly evolving, human expertise tracking and discovery involves ethical issues related to e.g. privacy.

### 4.5.2. Tools for planned cooperation

Although there is no distinct border between ad hoc and planned cooperation we separate them here because of pragmatic reasons. Any practical solution has to support not only both types of cooperation, but also seamless transitions from one type to another.

Planned cooperation has been the main focus of grid research. The planning is done in form of workflow models where one defines a priori which resources will be involved and how cooperation among resources will happen. The modelling is often based on a timeline principle with steps of cooperation happening in predefined order. This type of planning is often suitable when the cooperation at hand is among computing resources. In this case, one can rely on the properties of the resource (e.g. the resource will always get input A and produce output B) and partly on its availability and quality of service. In cases when a resource is not available in a specific step of the workflow, another instance of that resource (with the same resource description and QoS) is chosen to replace the missing resource.

Involving human resources in a grid problem solving workflow introduces a great amount of uncertainty into the workflow. One can also argue that involving human resources is in fact an indication that there is uncertainty in the workflow in the first place, and human resources are introduced for coping with that uncertainty (e.g. through fuzzy decision making or through scanning the situation). For this reason, tools for supporting human activity in the grid workflow will be necessary.

There are a number of technologies for coping with the "human aspects" of planned cooperation. Much of this research has its origins from the field of CSCW (computer supported cooperative work). Two of these technologies with most relevance for Akogrimo are discussed here:

- *Flexible workflows*: To cope with uncertainty in workflows, research has been done in the area of flexible workflows. A flexible workflow allows on-the-fly modifications to its course of action. For example, if a resource is not available in an instance of a workflow, a workaround can be modelled for that instance on the fly. In the most extreme case tools for *cooperative planning* [122] allow the whole planning happen ad hoc. One property of mobile grids is that resources making up a virtual organization are discovered dynamically and might disappear without warning (see the disaster management scenario for examples). In this case changes to the workflow might become the norm and not the exception.
- *Shared workspaces*: In many cases where the level of uncertainty is so high that planning becomes practically impossible (see again the disaster management scenario) another relevant technology is that of shared workspaces [123]. A shared workspace can be seen as a snapshot of an ongoing process. A shared workspace provides a picture of the ongoing activity, where the picture contains elements that are useful for cooperation and decision making. BSCW (Basic Support for Cooperative Work) is a well-known example of a shared workspace (also used as a support tool in the Akogrimo project). A shared workspace in BSCW contains

information about digital resources (e.g. documents, URLs, calendars) and their status, human resources who participate in the activity[3], meta information about the resources, and other artefacts that introduce some structure (planning) to the workspace. Examples of the latter are the folder hierarchies in BSCW.

## 4.6. References for Mobile Network Middleware Layer


[31] http://www.computerweekly.com/Article130068.htm

[32] http://www.projectliberty.org/

[33] http://www.ietf.org/rfc/rfc3261.txt

[34] http://www.ietf.org/rfc/rfc2327.txt

[35] http://ftp3.itu.int/av-arch/avc-site/2001-2004/0305_Gen/h323V5consented.zip

[36] http://www.ietf.org/rfc/rfc2326.txt

[37] http://www.ietf.org/rfc/rfc3550.txt

[38] http://www.ietf.org/rfc/rfc2616.txt

[39] http://www.w3.org/TR/2004/REC-xml-20040204/

[40] http://attend.it.uts.edu.au/icita05/CDROM-ICITA04/papers/92a-13.pdf

[41] http://honiden-lab.ex.nii.ac.jp/survey-data/term6/paper/4-Tei.pdf

[42] http://www.ietf.org/rfc/rfc1157.txt

[43] http://www.ietf.org/rfc/rfc1902.txt

[44] http://www.ietf.org/rfc/rfc3410.txt

[45] http://www.ietf.org/rfc/rfc1189.txt

[46] http://jpmf.home.cern.ch/jpmf/talks/bell_labs_20000315.pdf

[47] http://www.dmtf.org/standards/wbem/

[48] http://www.ietf.org/rfc/rfc3031.txt

[49] http://www.ietf.org/rfc/rfc2205.txt

[50] http://www.ietf.org/rfc/rfc2210.txt

[51] http://ietfreport.isoc.org/ids/draft-ietf-nsis-signalling-analysis-04.txt

[52] http://ietfreport.isoc.org/ids/draft-ietf-nsis-fw-06.txt

[53] http://ietfreport.isoc.org/ids/draft-ietf-nsis-ntlp-03.txt

[54] http://ietfreport.isoc.org/ids/draft-ietf-nsis-qos-nslp-04.txt


[3] BSCW does not however show much information about the current status of a human resource nore his/her current expertise.

[55] http://staffx.webstore.ntu.edu.sg/personal/ceandreas/Shared%20Documents/papers/ICC_2004.pdf

[56] http://www.comsoc.org/pubs/net/ntwrk/cfpnetwork0903.pdf

[57] http://www.omg.org

[58] http://www.openmobilealliance.org/

[59] http://www.w3.org/

[60] http://www.osgi.org/

[61] http://www.dlna.org/home

[62] http://www.omg.org/gettingstarted/specintro.htm#CORB

[63] http://www.omg.org/docs/formal/04-04-02.pdf

[64] http://www.omg.org/mda/

[65] http://www.microsoft.com/com/tech/DCOM.asp

[66] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsoap/html/understandsoap.asp

[67] http://www.ietf.org/rfc/rfc3288.txt

[68] http://msdn.microsoft.com/netframework/

[69] http://java.sun.com/products/jdk/rmi/

[70] http://java.sun.com/j2ee/

[71] http://java.sun.com/j2se/index.jsp

[72] http://java.sun.com/j2me/

[73] http://java.sun.com/developer/products/jini/index.jsp

[74] http://www.jxta.org/

[75] http://www-306.ibm.com/software/websphere/

[76] http://www.microsoft.com/windows2000/technologies/communications/msmq/default.asp

[77] http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/more/messageq

[78] http://www.tibco.com/software/enterprise_backbone/enterprisemessageservice.jsp

[79] http://java.sun.com/products/jms/overview.html

[80] http://joram.objectweb.org/

[81] http://consortium.objectweb.org/

[82] http://www.osmq.org/

[83] http://www.xmlblaster.org/

[84] Nagaratnam, N., Janson, P., Dayka, J., Nadalin, A., Siebenlist, F., Welch, V., Foster, I., and Tuecke, S. July 2002. *The security architecture for Open Grid Services. GGF5,*. July 2002, http://www.globus.org/ogsa/Security/

[85] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, and S. Tuecke. Security for Grid Services. *The 12th IEEE International Symposium on High Performance Distributed Computing (HPDC)*, pages 48-61, June 2003.

[86] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, D. Spence, Generic AAA Architecture, RFC 2903, August 2000, http://www.ietf.org/rfc/rfc2903.txt

[87] C. Rigney, A. Rubens, W. Simpson, S. Willens, Remote Authentication Dial In User Service (RADIUS), RFC 2138, April 1997, http://www.ietf.org/rfc/rfc2138.txt

[88] RFC2139, http://www.ietf.org/rfc/rfc2139.txt

[89] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, J. Arkko, Diameter Base Protocol, RFC 3588, September 2003, http://www.ietf.org/rfc/rfc3588.txt

[90] C. Finseth, *An Access Control Protocol, Sometimes Called TACACS*, RFC 1492, July 1993 http://www.ietf.org/rfc/rfc1492.2.txt

[91] Kerberos website, http://web.mit.edu/kerberos/www/ , accessed in November 2004

[92] W. Yeong, T. Howes, S. Kille, *Lightweight Directory Access Protocol*, RFC 1777 March 1995, http://www.ietf.org/rfc/rfc1777.txt

[93] R. Housley, W. Polk, W. Ford, D. Solo, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, RFC3280, April 2002, http://www.ietf.org/rfc/rfc3280.txt

[94] Warwick Ford, Phillip Hallam-Baker, Barbara Fox, Blair Dillaway, Brian LaMacchia, Jeremy Epstein, Joe Lapp, XML Key Management Specification (XKMS), March 2001, http://www.w3.org/TR/xkms/

[95] B.Stiller, J.Gerke, P.Flury, P.Reichl, Hasan, Charging Distributed Services of a Computational Grid Architecture, Proceedings of the 1st International Symposium on Cluster Computing and the Grid, IEEE Computer Society (2001), p596, ISBN:0-7695-1010-8. Abstract at http://portal.acm.org/citation.cfm?id=560889.792365 , paper at http://www.tik.ee.ethz.ch/~reichl/publications/IQ01.pdf

[96] M. Falkner et.al., An Overview of Pricing Concepts for Broadband IP Networks

[97] L. Anania and R. J. Solomon, "Flat: The Minimalist Price," Internet Economics, L. W. McKnight and J. P. Bailey, Eds., Cambridge, Massachusetts, 1997, MIT Press, pp. 91–118.

[98] P. C. Fishburn and A. M. Odlyzko, "Dynamic Behavior of Differential Pricing and Quality of Service Options for the Internet," June 1998, available from URL http://www.research.att.com/~amo

[99] A. M. Odlyzko, "A Modest Proposal for Preventing Internet Congestion," Sept. 1997, available from URL http://www.research.att.com/~amo

[100] RFC3334, Policy-based Accounting, http://www.faqs.org/rfcs/rfc3334.html (2002)

[101] RFC2722, Traffic Flow Measurement Architecture, http://www.faqs.org/rfcs/rfc2722.html (1999)

[102] J. K. Mackie-Mason and H. R. Varian, "Pricing the Internet", Int'l Conf. Telecommunication Systems Modelling, Nashville, TN, USA, March 1994, available from URL http://www.spp.umich.edu/papers/listing.html , pp. 378–93.

[103] S. Shenker et al., "Pricing in Computer Networks: Reshaping the Research Agenda", ACM Computer Communication Review, 1996, pp. 19–43.

[104] B. Stiller, J. Gerke, P. Reichl, P. Flury: "The Cumulus Pricing Scheme and its Integration into a Generic and Modular Internet Charging System for Differentiated Services" Journal on Network and Systems Management, Vol. 3, No. 9, September 2001, pp 293-325.

[105] Josef Hallberg et al., "Bluetooth Positioning", Luleå University of Technology, 2001, http://media.sm.luth.se/publications/2002/hallberg02bluetooth.pdf.

[106] Microsoft Corporation, "Understanding universal plug and play", 2000, http://www.upnp.org/download/UPNP_UnderstandingUPNP.doc

[107] UDDI.org, " UDDI technical white paper", 2000, http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf

[108] Eugene A. Gryazin, "Service Discovery in Bluetooth", Helsinki Univerity of Technology, undated, http://www.cs.hut.fi/Opinnot/Tik-86.174/SD_in_Bluetooth.pdf.

[109] Corvallis Microtechnology, Inc, "Introduction to the Global Positioning System for GIS and Traverse", 1996, http://www.cmtinc.com/gpsbook/

[110] Motorola Inc., "Overview of 2G LCS Technologies and Standards", 2001, http://www.3gpp.org/ftp/workshop/Archive/0101LCS/Docs/PDF/LCS-010019.pdf

[111] Jonathan W. Lowo, "Special Handling of Spatio-Temporal Data", 2003, http://www.geospatial-online.com/geospatialsolutions/article/articleDetail.jsp?id=76577&&pageID=2

[112] W3C, "CC/PP Information page", http://www.w3.org/Mobile/CCPP/

[113] 3GPP, "3GPP Generic User Profile – Data Description Framework", 2005, http://www.3gpp.org/ftp/tsg_t/WG2_Capability/SWG2/SWG2_Joint_Meetings/0208_SWG2_SA5_Tampere/S5-023004%20T2C-020035%20T2-020683%20%20TS%2023.241%20GUP%20DDF%20(v0.3.1)%20(T2C-020003).doc

[114] Jeff Ayars et al., "Synchronized Multimedia Integration Language (SMIL 2.0)", W3C Recommendation, 2001, http://www.w3.org/TR/smil20/.

[115] M. Handley, V. Jacobson, "RFO2327 - SDP: Session Description Protocol", IETF, 1998, http://www.ietf.org/rfc/rfc2327.txt

[116] J. Hodges and R. Morgan, "RFC 3377 - Lightweight Directory Access Protocol (v3) Technical Specification", IETF, 2002, http://www.ietf.org/rfc/rfc3377.txt

[117] WAP Forum, "Wireless Application Protocol – WAP 2.0 – Technical White Paper", 2002, http://www.wapforum.org/what/WAPWhite_Paper1.pdf.

[118] Kevin Dixon, "Symbian OS Version 7.0s - functional description", Symbian Inc., 2003, http://www.symbian.com/technology/symbos-v7s-det.html.

[119] Kraut, R.E., Fish, R.S., Root, R.W. and Chalfonte, B.L. Informal communication in organizations: form, function, and technology. in Oskamp, S. and Spacapan, S. eds. People's reactions to technology in factories, offices, and aerospace, Sage Publications, 1990, 145--199.

[120] See for instance http://www.jabber.org.

[121] http://www.skype.com.

[122] Carlsen, S., Krogstie, J., Sølvberg, A. and Lindland, O.I. Evaluating Flexible Workflow Systems. in Proceedings of the Thirtieth Annual Hawaii International Conference on System Science, 1997.

[123] Gutwin, C., Greenberg, S. and Roseman, M. Workspace Awareness in Real--Time Distributed Groupware: Framework, Widgets, and Evaluation. in Sasse, R.J.a.C., A. and Winder, R. ed. Proceedings of the HCI'96: People and Computers XI, London, U.K., Berlin, 1996, 281--298.

[124] Masahiro Hori, Goh Kondoh, Kouichi Ono, Shin-ichi Hirose, and Sandeep Singhal, "Annotation-Based Web Content Transcoding", IBM, undated, http://www9.org/w9cdrom/169/169.html.

# 5. Mobile Grid Infrastructure Services Layer

## 5.1. Foundation

This section presents fundamental specifications for Web and Grid Services.

### 5.1.1. Messaging

#### *5.1.1.1. WS-Addressing*

WS-Addressing [135] [142] defines two constructs *endpoint reference* and an associated *message information header*. These constructs are typical elements of messaging protocols. By defining endpoint references and message headers, WS-Addressing effectively moves message addressing information from the transport protocol to the SOAP layer. WS-Addressing provides a well-defined way to do asynchronous one-way messaging, with the ability to correlate messages. Many other specifications as well as WSRF build on and leverage endpoint references and message information headers.

#### *5.1.1.2. WS-Notification*

WS-Notification [136] is a family of documents that define a topic-based publish/subscribe pattern. The WS-Notification family of documents includes: a white paper "Publish-Subscribe Notification for Web Services" and three normative specifications: WS-BaseNotification, WS-BrokeredNotification, and WS-Topics.

##### *WS-BaseNotification*

WS-BaseNotification defines Web Services interfaces for NotificationProducers, NotificationConsumers and SubscriptionManager. The specification depends on the WS-ResourceProperties and WS-ResourceLifetime that are part of the Web Services Resource Framework. WS-BaseNotification defines a basic publish/subscribe pattern for asynchronous notification.

##### *WS-Topics*

WS-Topics supplements the publish/subscribe pattern defined by WS-BaseNotification to that effect that it allows to define topics to subscribe to. Furthermore it specifies an XML model for describing metadata associated with topics and allows building topic hierarchies.

##### *WS-BrokeredNotification*

WS-BrokeredNotification defines interfaces for a NotificationBroker and a PublisherRegistrationManager. A NotificationBroker is an intermediary, which among other things, allows publication of messages from entities that are not themselves service providers. A

subscriber can subscribe for one or more topics and is notified if any of the NotificationProducers have generated a message for the topic of interest. The PublisherRegistrationManager is responsible for maintaining the association of producers and topics.

#### 5.1.1.3. WS-Eventing

In a Web Services 5.4 oriented relationship it's often important to know if certain events occur in other services or applications. Instead of requesting all the time the status of a service it would be much better for the consuming service if there could be a mechanism for registering interest at this service. The WS-Eventing specification defines a protocol that allows Web Services (called an "event sink") to subscribe (called a "subscription") to another Web Service (called an "event source") in receiving messages about events (called "notifications").

The specification aims to meets the following requirements5.4:

- Defines message format for creating renewing and deleting event subscriptions. Subscriptions can be defined to automatically expire after a certain period of time or at a particular date-time.
- Event Sink (i.e. the subscriber) can determine which subscriptions it is receiving the notification from
- Defines how one event sink can subscribe on behalf of another one
- The event sink can request that all notifications sent to it be marked with specific reference properties.
- Provides extensibility to support sophisticated or unanticipated scenario subscriptions
- Allows to specify filters, which identify the notifications to be sent. Note that the source must support the dialect in which the filter is specified. Otherwise the request must be denied.
- Leverages other Web Service specifications for secure, reliable, transacted message delivery
- Supports both SOAP v1.1 and v1.2

Through the composable nature of Web Service specifications, WS-Eventing leverages the other Web Service specifications for secure, reliable, transacted message delivery.

#### 5.1.1.4. WS-ReliableMessaging

WS-ReliableMessaging specification – reliable messaging (RM) for Web Services - describes a protocol that allows messages to be delivered reliably between distributed applications in the presence of software component, system, or network failures. The defined messaging protocol is able to identify, track, and manage the reliable delivery of messages between exactly two parties, an RM Source and an RM Destination. The specification also defines a SOAP binding required for interoperability and integrates with and complements the WS-Security, WS-Policy, and other Web Services specifications.

The specification defines four levels of delivery assurances that the endpoints can provide[130]:

**AtMostOnce:** Messages are delivered at most once without duplication or an error is raised on at least one endpoint. Some messages in a sequence may not be delivered.

**AtLeastOnce:** Messages are delivered or an error is raised on at least one endpoint. Some messages may be delivered more than once.

**ExactlyOnce:** Every message sent is delivered without duplication or an error will be raised on at least one endpoint.

**InOrder**: Messages are delivered in the order that they were sent. This delivery assurance can be combined with any of the above delivery assurances. It requires that the sequence observed by the ultimate receiver be non-decreasing. It does not say anything about duplications or omissions.

In essence, sequenced messages are sent from one WS-ReliableMessaging enabled endpoint to another. It is the responsibility of the end points, the RM Source and RM Destination to fulfil the delivery assurances, or raise an error in case of failures.

#### 5.1.1.5. WS-Enumeration

WS-Enumeration [137] describes a Web Services messaging protocol. The specification was published in September 2004 and is a normative prerequisite for WS-Management. WS-Enumeration uses a session abstraction called an *enumeration context*. The problem that is addressed by WS-Enumeration is accessing data that is too large to fit in a single SOAP message. The "solution" is to read the data in chunks. A data source can decide request-by-request whether the consumer or the source is responsible for maintaining the state of the progress of reading data.

### 5.1.2. State & Resource Provision

#### 5.1.2.1. WS-Resource Framework (WSRF)

WSRF [131] is a set of specifications that defines necessary means to provide standardised access to and management of resources that are exposed via Web Services. This includes mechanism for getting and setting values of one or more properties, as well as querying across these values. It's possible that a resource can be destroyed on request or after it has exceeded a certain lifetime (which may be extended anytime). By changes towards the resources interested parties may be informed by a notification event.

Additional the specifications allow for joining Web Services with the WS-Resource into groups, similar to a registry. Such groups can be restricted as to what services are allowed to join, meaning whether certain Web Service interfaces exist and/or certain (resource) parameters are exposed. A service group can act and communicate on behalf of its members, i.e. the services may be addressed via the service group they belong to.

The Web Services Resource Framework comprises the following specifications:

**[WS-ResourceProperties]** which standardises the operations of a WS-Resource, as well as the structure of the resource properties document, which represents a view on the WS-Resource's state. Furthermore subscription to notification events using [WS-Notification] is described.

**[WS-ResourceLifetime]** Management of a WS-Resource in terms of immediate and delayed destruction, respectively extension of lifetime is described within the specifications. Notably, no message exchange with respect to the creation of a WS-Resource is specified.

**[WS-ServiceGroup]** specifies the standard resource properties defining contents of a service group and how to access details of an entry. Note that it is not an objective to represent the function of a member in a group. The constraints for membership are expressed by intension using a classification mechanism.

**[WS-BaseFaults]** The purpose of these specifications is to define a common way for error messages. Each Base Fault specifies a reference to where the fault was generated, and a timestamp when the fault occurred.

**[WS-RenewableReferences]** The WS-RenewableReferences specification defines the mechanisms to renew a WS-Resource endpoint reference that becomes invalid.

Besides these, WSRF relies on [WS-Notification] for event notification and [WS-Addressing] for referencing a WS-Resource and the appropriate resource properties document.

#### *5.1.2.2. WS-Transfer*

WS-Transfer [132] specification defines a mechanism for accessing XML representations of Web Service-based resources. It enables state transfer over SOAP-based protocol by defining how to invoke simple verbs like Get, Post, Put and Delete. The specification defines two entities: resources, which are entities addressable by an endpoint reference; and resource factories, which are Web Services that create a new resource from an XML representation. It also defines two operations for sending and receiving the representation of a given resource and two operations for creating and deleting a resource and its corresponding representation.

## 5.2. Manageability

Software systems are ever becoming more complex and so is their management. Failures in one component can easily influence the whole system. In traditional monolithic applications methods and tools to track down problems are well known and established, not so in distributed application environments. With the growing need to monitor and manage distributed applications several approaches to the problem have been made. In this document we will outline and evaluate the Grid Monitoring Architecture (GMA), proposed by the GGF; the Web Services Management Framework (WSMF) developed by HP; and the Management Using Web Services (MUWS) as well as the Management Of Web Services (MOWS) that are being standardized by the OASIS Web Services Distributed Management (WSDM) Task Force.

### 5.2.1. A Grid Monitoring Architecture (GMA)

In this paragraph we give a short summary of the Grid Monitoring Architecture (GMA) as specified by the GGF in the Performance Working Group document [139].

The GMA describes a high level service oriented architecture that can be applied to monitor components of a distributed IT system. It was actively developed by the GGF Performance Work Group until mid 2002. The motivation and intention is given as follows:

| |
|---|
| The goal of this paper is to describe the major components of a Grid monitoring architecture and their essential interactions. By adopting standard terminology and describing the minimal specification to support required functionality, we hope to encourage the development of interoperable high quality performance tools for the Grid. |

**Table 3 - GMA Motivation**

The Grid Monitoring Architecture consists of three types of components:

- Directory Service: supports information publication and discovery
- Producer: makes performance data available (performance event source)
- Consumer: receives performance data (performance event sink)

An example is shown in Figure 3.

For these components, basic interactions are defined, like adding an entry to the directory service, querying it for a consumer or producer, or subscribing/unsubscribing to an event producer. Furthermore general guidelines for implementation are given.

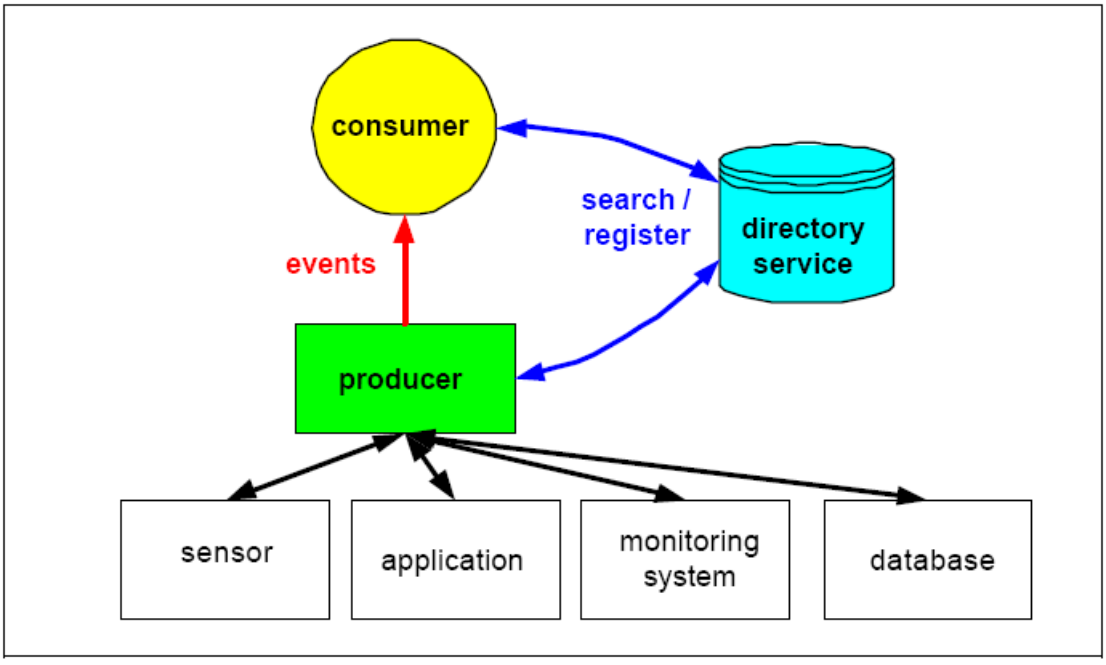


**Figure 3 - GMA Components**

Regarding the implementation, recommendations like the following are made and detailed:

- All system components must scale.
- The data management system must adapt to changing performance conditions.
- Monitoring data must be managed in a distributed fashion.
- Security standards are useful.

Figure 4 shows an example from the GGF document [139]:

- "Event data is collected on the two hosts and at the network routers between them. The host and network sensors are the sources of the measurement data, which is managed by a producer. The producer registers the availability of the host and network events in the directory service. A real-time monitoring consumer subscribes to all available event data for real-time visualization and performance analysis. The producer is capable of computing summaries of network throughput and latency data based on parameters provided by a "network-aware" client. This client uses the summarized network information to optimally set its TCP buffer size. The producer's event data is also sent to an archive." [139]

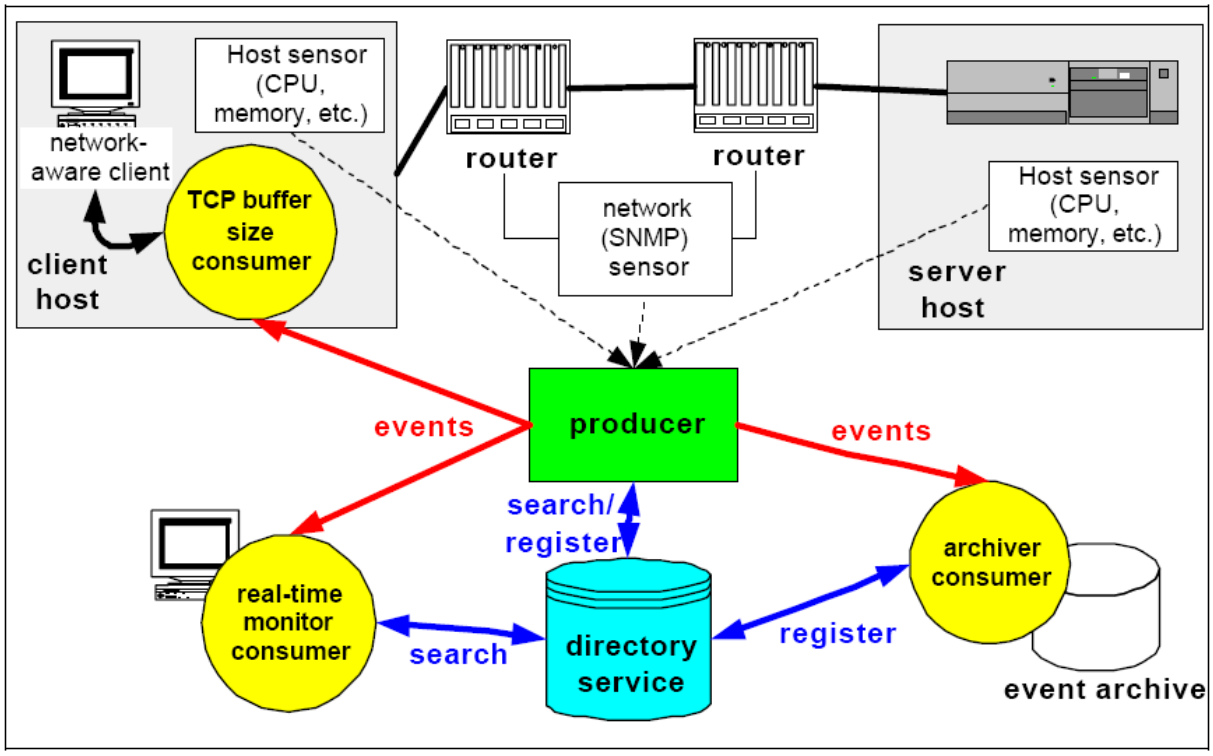


**Figure 4 - GMA sample use**

## 5.2.2. Web Services Management Framework

The Web Services Management Framework (WSMF) is developed by HP. The reference implementation by HP handles managed objects in Linux (Java) and .NET (C#) environments. Part of the work done by HP is submitted to OASIS for standardisation. Two specifications that

originate from the WSMF are Management Using Web Services (MUWS) (section 5.2.3) and Management Of Web Services (MOWS) (section 5.2.4).

In a presentation [140] given by Bryan Murray et al the GlobusWORLD 2004 a sample multi-media delivery system implementation is described (see Figure 5 - Demonstration using WSMF). To get an impression of what has been done in the WSMF context, a slide from the presentation [140] is shown in Figure 5.

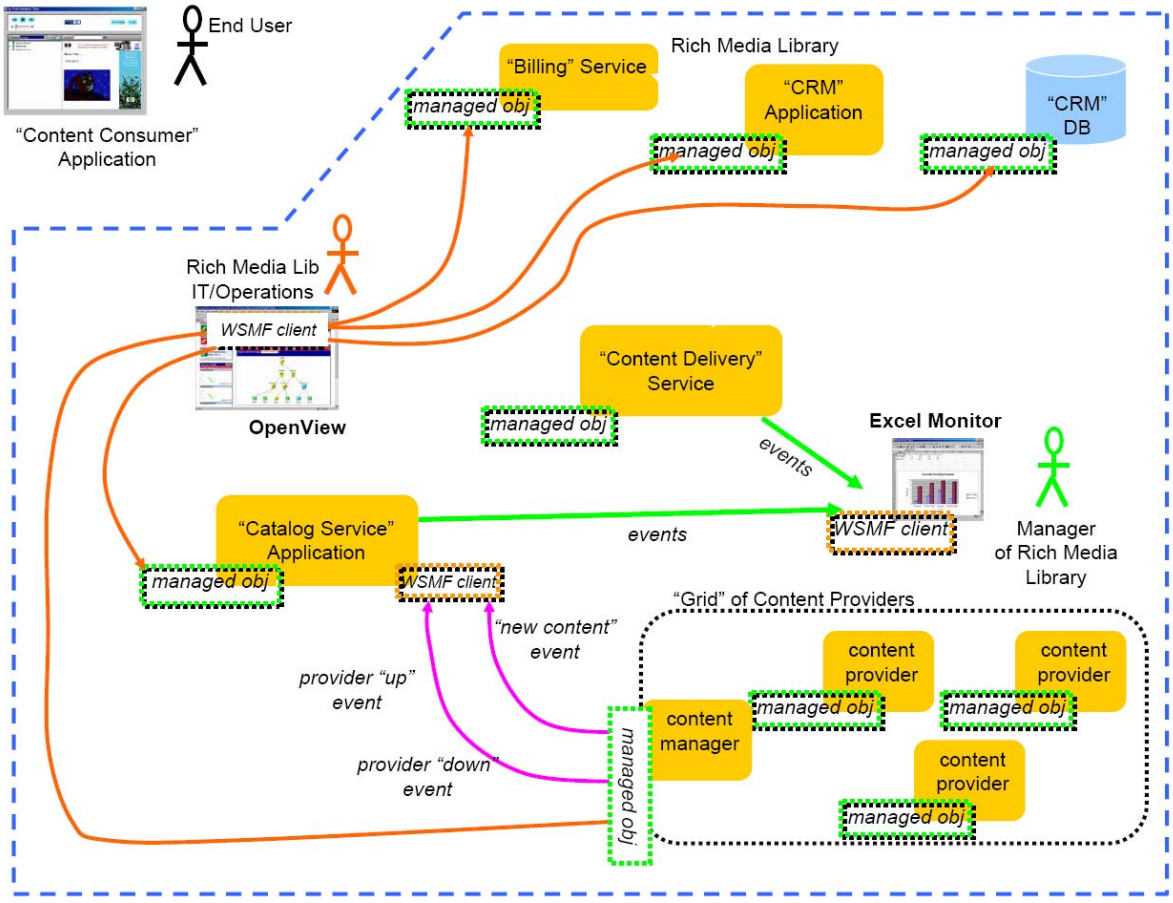


**Figure 5 - Demonstration using WSMF**

Among the technologies used in the HP demo are the Globus Toolkit GT3.0, WS-Events notifications tied to an OpenView Service Navigator and .NET applications.

The authors [140] see the future work regarding standardisation as follows:

OASIS/WSDM:

- Develop MOWS/MUWS proof-of-concept demonstration
- Complete MOWS information model
- Continue developing MUWS architecture

GGF/CMM

- Map WSDM MOWS information model to Grid model

- Complete gap analysis between WSDM requirements and Grid management requirements

### 5.2.3. Management Using Web Services (MUWS)

This current (December 2004) Management Using Web Services (MUWS) [133] specification is only a working draft and there is no guarantee that any part of its content will appear in the final release specification. However this specification is identified as relevant that has to be investigated and mentioned.

The MUWS specification deals with managing distributed resources using Web Services technologies. Therefore MUWS is based on a number of Web Services specifications like messaging, description, discovery, accessing properties and notifications (WS-Notification). Conceptually, management using Web Services could be described as follows (Figure 6).

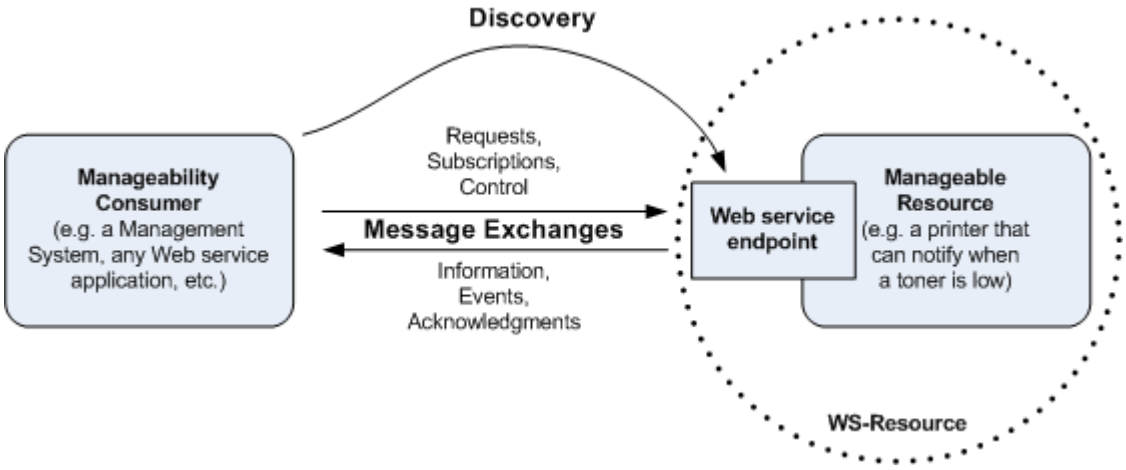


**Figure 6 - MUWS Concept**

The MUWS specification [133] deals with managing distributed resources using Web Services technologies. Examples of manageability functions that can be performed via MUWS include:

- monitoring quality of services
- enforcing service level agreements
- controlling tasks
- managing resource life-cycles

The MUWS specification defines following entities:

- Manageability endpoint: The WS representation of the resource to be managed.
- Manageability provider: E.g. a collection of managed endpoints.
- Manageability consumer: The application doing the management.

A manageability consumer, that could be a management system, an automation process, or simply any Web Service application, discovers a Web Service endpoint in order to exchange messages like request information, subscribe to events or to control the manageable resource. The manageability consumer discovers the Web Service endpoint by using the WS-Addressing

Endpoint Reference (EPR). The Web Service EPR provides access to a manageable resource that could be a magnetic storage disk, which can report its internal temperature reading, etc. Together the Web Service EPR and the manageable resource are a WS-Resource as defined by the WS-Resource specification framework (WSRF). MUWS architecture has a focus on manageable resources and on providing access to them and its manageability capability. A manageable resource is able to provide a set of manageability capabilities via Web Service endpoints. Every manageability capability has following definition:

- Unique identity
- Defined distinct semantics
- Defined resource-specific properties, operations, events (notification) and metadata (including policies)

Of course, every manageability capability itself is extensible and the MUWS specification provides mechanisms for defining new capabilities as well as mechanisms to discover, identify and use of capabilities.

It is the duty of the manageability provider to publish the manageability endpoints. A manageability consumer is able to use any and all of WSDM MUWS, SNMP or CIM/WBEM to access manageability information for an exposed resource.

The relation of manageable endpoint, resource and manageability endpoint is given in Figure 7, which is taken from the MOWS 0.5 specification (see [141]).

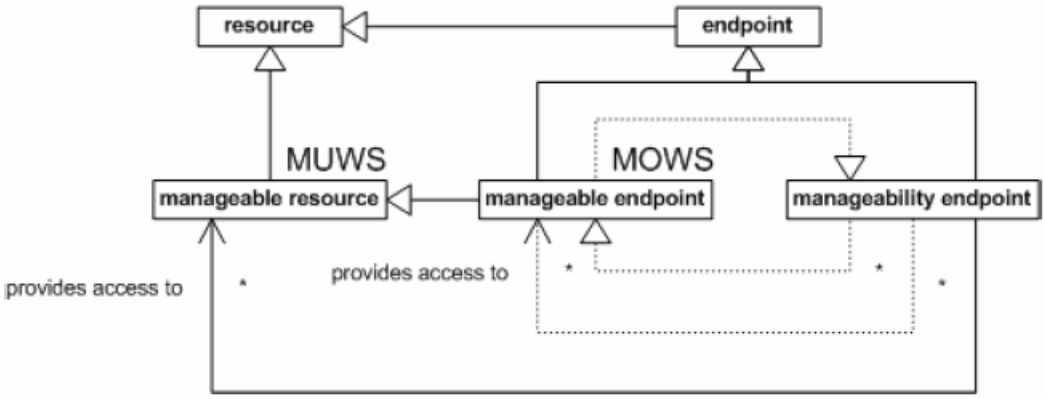


**Figure 7 - MOWS locus of implementation**

MUWS defines a state model for managed resources, as shown in Figure 8 - MUWS Resource State Model.

### 5.2.4. Management Of Web Services (MOWS)

The WSDM MOWS specification [138] is an application of MUWS to Web Services. i.e. MOWS treats Web Services as resources that are to be managed. One example given is counting the number of messages that the managed Web Service receives. Concepts defined are: *Identity* of an endpoint (from MUWS), *version*, metrics *classes* (as defined by MUWS), *operational state* (related to WSLC, i.e. Web Service Life Cycle), *operational status* (from MUWS), *request processing state*. None of these concepts is really new or unique. MOWS merely offers the possibility of addressing them in

a general Web Services management fashion. For example,. request processing state (*received*, *processing*, *completed*, *failed*) is also very well addressed in the context of Web Services choreography.

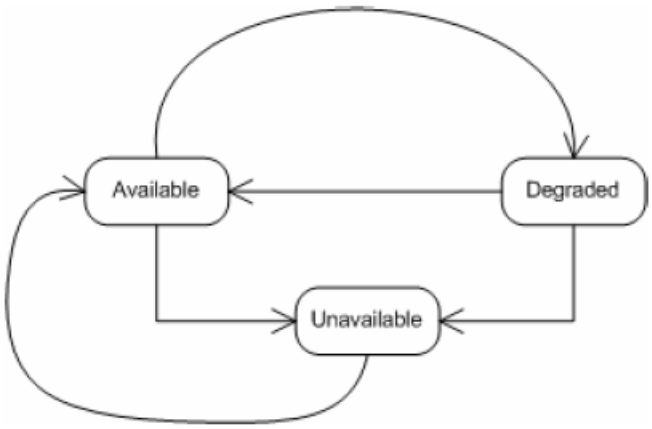


**Figure 8 - MUWS Resource State Model**

Metrics that are of concern to the manageability consumer are to be made accessible by the endpoint may include the metrics like the *number of failed request* or the *average service time.* Semantics of metrics classes are to be identified by the class name.

The identification capability is used to support the management of the endpoint and may be used to determine if two manageability providers manage the same resource or not.

The Web Service lifecycle (WSLC) states defined by the W3C Web Services Architecture Management Task Force can be mapped to the MUWS state model.

### 5.2.5. WS-Management

The recently (October 2004) published WS-Management [134] specification by Microsoft aims to standardise the way in which managed resources can be accessed in order to perform management tasks, like e.g. monitoring resource-parameters.

A WS-Management Resource is structured in the following way (Figure 9 below).

An agent offers management capabilities for a system by exposing a set of Web Services, so-called "Resource Services" that provide access to the actual resource(s). All resources covered by one resource service share the same operations and representation schemas, i.e. can be accessed in the same way. Resources in this sense are e.g. disk drives that are part of a PC ("System").

The respective information (locations) is provided using the WS-Addressing EPR that contains an identifier of the resource to be accessed.

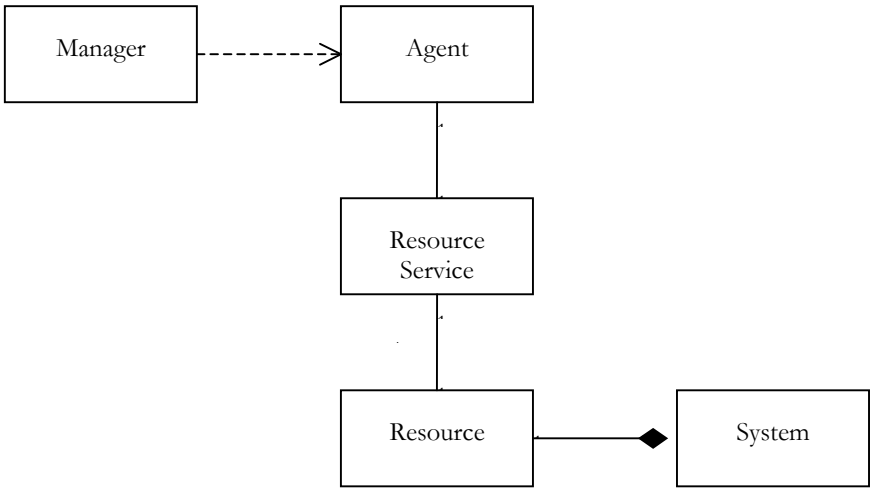


**Figure 9 -** Main concepts of WS-Management

## State

Using WS-Management an administrator (Manager) may access and manipulate the state of a Resource using the WS-Transfer specifications as described below, thus allowing reading, updating and even creating & deleting of system parameters if so supported by the Resource.

## Events

Furthermore, an administrator may want to receive events created by the Resource, respectively by the System that e.g. provides updates on parameter-changes and the like. To achieve this WS-Management relies on the WS-Eventing specifications (described in section 5.1.1.3). The specifications propose to use subscription managers for handling subscription to events, which have to be referenced in the EPR.

WS-Management extends the eventing capabilities by three additional event delivery modes that provide event messages in specific ways sensible for management tasks:

(a) Batched delivery mode, in which multiple event messages are bundled into one SOAP envelope

(b) In pull delivery mode event messages are logically queued and enumerated according to WS-Enumeration (section 5.1.1.5) - compare WS-Transfer (section 5.1.2.2)

(c) Trap delivery mode allows using UDP multicast for sending event messages to many customers

Additionally, the specifications foresee handling resubscription to specific events, in which case the subscriber may want to catch up with event messages since last unsubscribing. This obviously requires either the Resource (event source) or the Subscription Manager to keep a log of all messages.

### 5.2.6. Resumé of GMA, MUWS and MOWS

Like most complex system also software systems need to adjust their functioning to a changing environment. Like a power plant adopts its output to the current need, a complex distributed software system may adjust to a change of network bandwidth, or CPU load, or it might automatically replace faulty nodes. In order to realize this, a sort of self awareness must be established. Auto-adjustment depends on following elements:

- Sensors, to detect the current state of the system
- A model of the intended working of the system
- A means to adjust the current state

The GMA best fits in the sensor category, as it is concerned with monitoring of the resources in a Grid system. WSDM MUWS and MOWS provide for monitoring capabilities as well as the means to adjust the current system state. Automatic corrective actions would have to be performed by the management application that is not further detailed in any of the presented documents.

| Framework | Relation to WSRF | Comment |
|---|---|---|
| GMA | works together with it | GMA can be implemented using any technology that can realize a basic service oriented architecture |
| WSMF | works together with it | WSMF claims to be model-neutral. The MUWS/MOWS specifications that originate from it build upon the WSRF. |
| MUWS/MOWS | depends on it | Also depends on WS-Addressing [135] [142] |

**Table 4 - Relationship of frameworks to WSRF**

## 5.3. Further OGSA capabilities

This chapter on mobile grid infrastructure has covered several aspects of OGSA. We also need to describe the following aspects of OGSA:

- Data services
- Execution Management Services

### 5.3.1. Data services

In OGSA [125] [127], the requirements of data services are stated as: easy and efficient *access* to data, independent of physical location or platform; maintenance of *consistency* in the presence of replicas and caches; *persistence* as required; *uniform* access to integration of heterogeneous, federated and distributed data; and availability at the required *location*.

To some degree these requirements are satisfied by the downloadable OGSA-DAI toolkit [126]. This work is intended to closely mirror the work of the Database Access and Integration Services Working Group (DAIS WG) within GGF, to the extent that that is possible when also providing a toolkit compatible with currently available grid software.

At a lower level, Globus Toolkit 4.0 [128] is planned to provide Reliable File Transfer (RFT) and the pre-WSRF toolkit GridFTP.

In Akogrimo, a mobile service could be required, as part or all its functionality, to offer a Grid-based data service. This could include the metadata that may be required from other sources in order to interpret the data. It could be a mobile sensor which may act as one part of a distributed data storage or which may return its acquired data to a managed repository.

### 5.3.2. Execution Management Services

In OGSA [125], the Execution Management Services (EMS) are designed to solve problems concerned with the instantiating, managing and terminating items of work. The slightly generic phrase "item of work" is intended to convey that this need not correspond to a Grid Service. It could refer to a legacy application, the details of which are shielded by a Service interface, or to portions of a Service.

The same release of Globus, as is mentioned above (section 5.3.1), will include WS GRAM which is a reimplementation of the previous Grid Resource Allocation Manager (GRAM) but modified to be compatible with WS-RF. Other components for this release of Globus are to be the Community Scheduler Framework and Workspace Management.

In Akogrimo, a mobile device could run a service. If it is known that it holds a large up to date set of sensor data, but it is not time to transmit it, but nonetheless a Service Requestor requires some aggregate (the mean of all data satisfying some condition), it could make sense for the mobile device to perform that calculation. The alternative may be to ship the dataset somewhere else and calculate it faster there. Clearly there are resource tradeoffs.

## 5.4. References for Mobile Grid Infrastructure Layer


[125] Towards Open Grid Services Architecture (OGSA), GGF, http://www.globus.org/ogsa/

[126] The Open Grid Services Architecture Data Access and Integration (OGSA-DAI), http://www.ogsadai.org.uk/

[127] I.Foster, C.Kesselman, J.M.Nick, S.Tuecke, The physiology of the grid: An open grid services architecture for distributed systems integration, http://www.globus.org/research/papers/ogsa.pdf

[128] Status and Plans for the Globus Toolkit 4.0 (GT4), http://www-unix.globus.org/toolkit/docs/development/4.0-drafts/GT4Facts/index.html

[129] WS-Eventing, available from:

http://msdn.microsoft.com/webservices/understanding/specs/default.aspx?pull=/library/en-us/dnglobspec/html/ws-eventing.asp

[130] Web Services Reliable Messaging Protocol (WS-ReliableMessaging), available from: http://msdn.microsoft.com/ws/2004/03/ws-reliablemessaging/

[131] WS-Resource Framework, available from: http://www.globus.org/wsrf/

[132] WS-Transfer, available from:

http://msdn.microsoft.com/library/en-us/dnglobspec/html/ws-transfer.pdf

[133] Management Using Web Services (MUWS), available from:

http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm

[134] WS-Management
http://www.intel.com/technology/manage/downloads/ws_management.pdf

[135] The WS-Addressing specification can be found at:
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-addressing.asp

[136] The WS-Notification documents and specifications can be found at: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn

[137] The WS-Enumeration specification can be found at:
http://msdn.microsoft.com/library/en-us/dnglobspec/html/ws-enumeration.pdf

[138] The MOWS specification is part of the OASIS WSDM initiative and can be found at:
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm

[139] GGF Performance Working-Group, *A Grid Monitoring Architecture*, March 2000, Revised 27 August 2002, http://www-didc.lbl.gov/GGF-PERF/GMA-WG/papers/GWD-GP-16-3.pdf

[140] Bryan Murray, Latha Srinivasan, Jem Treadwell, *Managing the Grid with WSMF*, GlobusWORLD 2004, http://www.globusworld.org/program/slides/2c.pdf

[141] OASIS documents, http://www.oasis-open.org/committees/documents.php

[142] Don Box, et al., Web Services Addressing, March 2003,
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-addressing.asp

# 6. Mobile Grid Application Support Services Layer

## 6.1. Introduction

### 6.1.1. Introduction to this layer

The Application Support Services Layer will provide high level services to simplify the use of the GRID and to allow integration between applications and the basic Akogrimo Grid middleware.

This layer should serve as an application umbrella providing high-level functionality supporting a complex composition of services. This complements the work of the domain-oriented test beds definition that are primarily focused on a single application domain, by looking at the integration and composition of multiple needs in the context of a spread of applications (limited to those investigated in Akogrimo).

On the basis of this premise, this layer could cover different aspects such as: advanced collaborative tools, grid portals, composition of existing services, VO policies management, SLA definition, billing services, security management etc.

In particular, we are going to focus on VO management, SLA billing, portals and service composition, then the technologies listed in this section will be related to this topic, which we are focusing on.

Akogrimo applications will use Web Services to communicate and it makes sense to consider Web Services composition. Furthermore, lower layers may expose some of their functionality to higher layers, allowing parts of the network to be used through Web Services.

Some concepts cut across multiple layers and are presented in this way in the chapter on Analysis of cross layer themes,. In the case of Service Discovery, the aspects relevant to the mobile network layer are not discussed here at all, but are postponed to the corresponding section (7.3) in chapter 7. The section on Authentication and Authorisation and Accounting (AAA) (7.4) complements the section in this chapter on "Security: Web Services approach to Authentication & Authorization" (6.3).

### 6.1.2. The Grid Application Support Services layer role with respect to existing frameworks for supporting Grid Applications

Today, grid application support frameworks have the aim of providing a layer of abstraction on top of the underlying grid infrastructure and of providing an easy, comprehensible programming model with minimal possible extensions. They should make it possible to manage easy tasks such as file transfers, finding, and picking up the right resources on the basis of QoS parameters and job execution.

The state of the art offers different solutions for supporting Grid Applications: GridLab project [145], GridBlocks [146], Grid Application Framework for Java [144], GAF [147], Cactus [148], GRASP [149].

Each of them offers different capabilities and follows different approach to support Grid Application, however, with the emerging of OGSA, in our opinion, fully exploiting the grid means taking advantage of the virtualized grid infrastructure to accelerate processing time or to increase collaboration. In an OGSA compliant GRID environment it will mean that the application can run as a Web Service in such environment, while optionally taking advantage of the various services provided by the grid infrastructure.

The Akogrimo middleware is planned to be OGSA compliant and then it is supposed that applications interacting with it will be designed using components exposed as Web Service or they will have Web Service artefact (a wrapper, an interface, a façade, a veneer, or whatever you want to call it) that can invoke the existing code.

We envisage a Grid application support services layer implemented strictly within the vision of the Web Services Architecture (according with the last evolution in the frame of Grid research that led to the introduction of WSRF). Such an application framework is built on Web Services related specifications, can coexist with other Web Services specifications, and hopefully it should leverage on existing tools for Web Service development. Due to these considerations, the following sections mainly will describe technologies and specifications coming from the Web Service world, in particular, as we have stated before will focus on aspects related to VO management, SLA definition, service composition and security

## 6.2. Service composition and workflow management

One of the revolutionary concepts made possible by the Internet is "electronic value chains" that can be carried out through Service composition. It is the problem of composing autonomous services to achieve new functionality, that has the potential to reduce development time and effort for new applications. The service composition assumes particular relevance if it has applied to “electronic value chains” because it can provide several points of added value:

- Firstly, increasing numbers of interesting services are moving online and the web is fast transforming from a collection of static pages to a provider of numerous useful services.
- Furthermore, Web Services conform to the standard HTTP protocol which makes it (relatively) easier to integrate them into a common framework.
- Finally, because, potentially, the web has several independent service providers providing related services, there is an inherent need for composing complementary services provided by independent providers to achieve the end-user's needs.

Service Composition [151] allows different autonomous services to be combined in such a way that a new service with a different functionality is created. Service composition allows highly modular and independent services to be developed and then use all of them to create a much larger and more complex service.

Service composition is intimately linked to service description, as well as service discovery. In order to have an automatic composition of services, it is necessary to know beforehand what services are available and their functionality. User preferences, rules set by providers should also be taken into account. The user’s location and the technological means available to him at that moment will also influence service composition. This is illustrated in Figure 10.

A pervasive and mobile environment [154] such as Akogrimo will add new challenges to service composition. A user may request a service from a device, and then move to another device which has more limited functionality and is not able to display the service's results. Also, the user may move between different access points, so if the service requested is location dependent, its result will be useless. Therefore, service composition must handle possible errors, eventually restarting the service composition process with the new parameters, so that the result will be useful for the user.

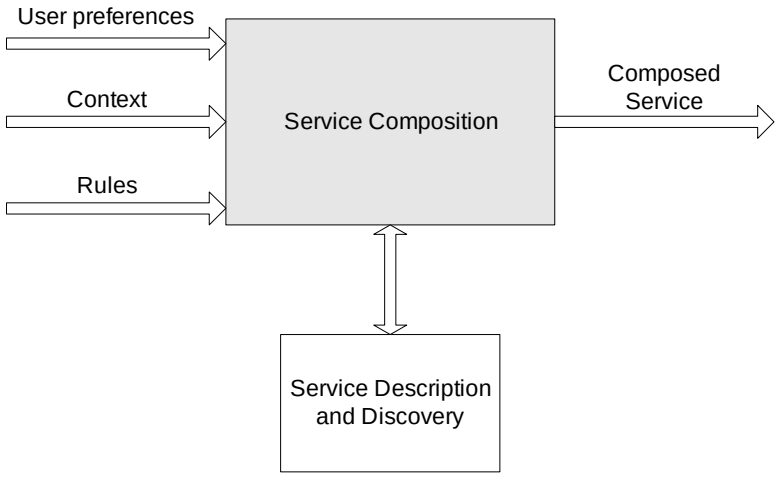


**Figure 10 - Service composition inputs and outputs**

On the more general subject of workflow, we are assuming that at the Grid Application Support Services layer workflow will mean Web Service workflow and this has all potentials to successfully integrate value chains and opens the door to dynamic e-business, e-health, e-government is a daunting challenge when integrating different value chains from different corporations.

This section introduces some specifications and standards for service composition and Web Service workflow management for meeting real world business demand.

## 6.2.1. Orchestration

### *6.2.1.1. Web Service Orchestration*

Often, Web Services need to be put together using pre-defined scripts or orchestration [152][153] scripts, containing messages, branching logic and invocation sequences. Web Services orchestration is about providing an open, standards-based approach for connecting Web Services together to create higher-level business processes. Standards such as BPEL4WS and BPML are designed to reduce the complexity required to orchestrate Web Services, thereby reducing time-to-market and costs, and increasing the overall efficiency and accuracy of business processes. Without a common set of standards, each organization is left to build their own set of proprietary business protocols, leaving little flexibility for true Web Services collaboration.

#### 6.2.1.2. BPEL4WS

The Business Process Execution Language for Web Services is an initiative of the industry leaders BEA Systems, IBM, Microsoft, SAP AG, Siebel Systems to drive and ensure interoperability for the description and communication of business processes based on Web Services.

BPEL4WS defines a notation for specifying business process behaviour based on Web Services. Processes in BPEL4WS export and import functionality by using Web Service interfaces exclusively.

BPEL4WS provides a language for the formal specification of business processes and business interaction protocols. By doing so, it extends the Web Services interaction model and enables it to support business transactions. BPEL4WS defines an interoperable integration model that should facilitate the expansion of automated process integration in both the intra-corporate and the business-to-business spaces.

BPEL4WS represents a convergence of the ideas in the XLANG and WSFL specifications. Both XLANG and WSFL are superseded by the BPEL4WS specification

The specification provides an XML-based grammar for describing the control logic required to coordinate Web Services participating in a process flow. This grammar can then be interpreted and executed by an orchestration engine, which is controlled by one of the participating parties. The engine coordinates the various activities in the process, and compensates the system when errors occur. BPEL4WS is essentially a layer on top of WSDL, with WSDL defining the specific operations allowed and BPEL4WS defining how the operations can be sequenced. A BPEL document leverages WSDL in three ways:

- Every BPEL process is exposed as a Web Service using WSDL. The WSDL describes the public entry and exit points for the process.
- WSDL data types are used within a BPEL process to describe the information that passes between requests.
- WSDL might be used to reference external services required by the process.

BPEL4WS provides support for both *executable* and *abstract* business processes. An executable process models the behaviour of participants in a specific business interaction, essentially modelling a private workflow. Abstract processes, modelled as business protocols in BPEL4WS, specify the public message exchanges between parties. Business protocols are not executable and do not convey the internal details of a process flow. Essentially, executable processes provide the orchestration support described earlier while the business protocols focus more on the choreography of the services.

#### 6.2.1.3. BPML

The Business Process Management Language (BPML) is a meta-language for describing business processes. The BPML specification provides an abstract model for expressing business processes and supporting entities. BPML defines a formal model for expressing abstract and executable processes that address all aspects of enterprise business processes, including activities of varying complexity, transactions and their compensation, data management, concurrency, exception handling and operational semantics. BPML also provides a grammar in the form of an XML Schema for enabling the persistence and interchange of definitions across heterogeneous systems

and modeling tools. BPML itself does not define any application semantics such as particular processes or application of processes in a specific domain; rather it defines an abstract model and grammar for expressing generic processes. This allows BPML to be used for a variety of purposes that include, but are not limited to, the definition of enterprise business processes, the definition of complex Web Services, and the definition of multi-party collaborations.

By leveraging the WSCI specification, BPML enables the modeling of end-to-end processes that can be translated into collections of private implementations executed as BPML processes and public interfaces defined using WSCI. Together, they provide an end-to-end view that depicts the role of each individual business process in the overall choreography, and the business activities performed by each role. Both BPML and WSCI share the same underlying process execution model, as well as similar syntaxes.

#### 6.2.1.4. Brief comparison of BPML and BPEL4WS

The BPML specification can also be loosely compared to BPEL4WS, providing similar process flow constructs and activities. The features supported by BPML include persistence, instance correlation, and roles. The language was designed to manage long-lived processes, with persistence supported in a transparent manner. XML exchanges occur between the various participants, with roles and partner components similar to the BPEL constructs. Additionally, BPML supports recursive decomposition, the ability to compose sub-processes into a larger business process. Furthermore, BPML includes a robust exception handling mechanism. Finally, BPML provides the ability to nest processes and transactions, a feature that BPEL currently does not provide.

### 6.2.2. Choreography

#### 6.2.2.1. WS-CHOREOGRAPHY

This is defined in the frame of W3C. It provides support to specify interactions of Web Services with their users (the sequence and conditions for message exchanging).

WS-Choreography is different from Orchestration because it takes more of an "inside-out" perspective, describing an executable process from the perspective of one of the partners, while choreography takes more of a collaborative and choreographed approach.

#### 6.2.2.2. WSCI

The Web Service Choreography Interface (WSCI) is an XML-based interface description language that describes the flow of messages exchanged by a Web Service participating in choreographed interactions with other services.

WSCI describes the dynamic interface of the Web Service participating in a given message exchange by means of reusing the operations defined for a static interface. WSCI works in conjunction with the Web Service Description Language (WSDL), the basis for the W3C Web Services Description Working Group; it can, also, work with another service definition language that exhibits the same characteristics as WSDL.

WSCI describes the observable behaviour of a Web Service. This is expressed in terms of temporal and logical dependencies among the exchanged messages, featuring sequencing rules, correlation, exception handling, and transactions. WSCI also describes the collective message exchange among interacting Web Services, thus providing a global, message-oriented view of the interactions.

WSCI does not address the definition and the implementation of the internal processes that actually drive the message exchange. Rather, the goal of WSCI is to describe the observable behaviour of a Web Service by means of a message-flow oriented interface. This description enables developers, architects and tools to describe and compose a global view of the dynamic of the message exchange by understanding the interactions with the Web Service.

## 6.2.3. Coordination and Transactions

### *6.2.3.1. WS-COORDINATION*

WS-Coordination describes an extensible framework for providing protocols that coordinate the actions of distributed applications. Such coordination protocols are used to support a number of applications, including those that need to reach consistent agreement on the outcome of distributed activities.

The framework defined in the specification enables an application service to create a context needed to propagate an activity to other services and to register for coordination protocols. The framework enables existing transaction processing, workflow, and other systems for coordination to hide their proprietary protocols and to operate in a heterogeneous environment.

Additionally the specification describes a definition of the structure of context and the requirements for propagating context between cooperating services.

The WS-Coordination specification talks in terms of activities, which are distributed units of work involving one or more parties (which may be services, components, or even objects). At this level, an activity is minimally specified and is simply created, made to run, and then completed.

WS-Coordination looks set to become the adopted standard for activity coordination on the Web. WS-Coordination provides only activity and registration services, and is extended through protocol plug-ins that provide domain-specific coordination facilities. In addition to its generic nature, the WS-Coordination model also scales efficiently via interposed coordination, which allows arbitrary collections of Web Services to coordinate their operation in a straightforward and scalable manner.

### *6.2.3.2. WS-Atomic Transaction & WS-Business activity*

The WS-Atomic Transaction & WS-Business Activity are two distinct models proposed in the frame of WS-Transaction specification that supports the notion of the service and participant as distinct roles, making the distinction between a transaction-aware service and the participants that act on behalf of the service during a transaction. A transaction-aware service encapsulates the business logic or work that is required to be conducted within the scope of a transaction. This work cannot be confirmed by the application unless the transaction also commits and so control

is ultimately removed from the application and placed into the transaction's domain. The participant is the entity that, under the dictates of the transaction coordinator, controls the outcome of the work performed by the transaction-aware Web Service.

WS-AtomicTransaction (private WS-* specification) and WS-BusinessActivity (private WS-* specification) are two specifications respectively provide the definition of the atomic transaction coordination type and business activity coordination type which are to be used with the extensible coordination framework described in WS-Coordination. WS-Coordination (private WS-* specification) describes an extensible framework for providing protocols that coordinate the actions of distributed applications.

#### *6.2.3.3. WS-CAF*

In parallel to the above standards, there is also another competing specification: WS-CAF (private WS-* specification), which supports information sharing and transaction processing of Web Services. WS-CAF itself is the collection of three Web Services specifications: WS-CTX, WS-CF, and WS-TXM (all private WS-* specifications). WS-CTX provides an open, common, interoperable runtime mechanism to manage, share, and access context information among related Web Services. WS-CF defines a software agent to handle context management to ensure that messages and results are correctly communicated. WS-TXM defines three distinct transaction protocols that can be plugged into the coordination framework for interoperability across existing transaction managers.

#### *6.2.3.4. Brief comparison*

Relatively, the first Web Services transaction and coordination framework (WS-AtomicTransaction + WS-BusinessActivity + WS-Coordination) is a bit more widely acknowledged than WS-CAF, though the both are expected to co-exist for some time.

## 6.3. Security: Web Services approach to Authentication & Authorization

### 6.3.1. Web Services security specifications

Web Services provide a standardized framework for interoperable, secure, reliable and transacted messaging, and constitute the ideal underlying service-oriented messaging framework for collaboration within dynamic VOs across enterprises. This section covers in detail the Web Services Security (in the remainder of this section referred to as WSS) specifications, following the architecture and roadmap proposed by IBM and Microsoft in April 2002 [150].

The WSS specifications provide a comprehensive and composable security framework for SOAP-based Web Services, supporting and integrating various security models, mechanisms, and technologies in a way that enables a variety of systems to securely interoperate in a platform- and language-neutral manner. The WSS roadmap consists of different, flexible and extensible specifications, each addressing specific parts of the security framework, including WS-Security,

WS-SecureConversation, WS-Trust, WS-SecurityPolicy, WS-Federation, WS-Authorization, and WS-Privacy.

The leading standard of the Web Services security protocol layer is WSS (OASIS), which currently consists of three subset protocols: WSS SOAP Message Security (WS-Security), WSS UsernameToken Profile, and WSS X.509 Certificate Token Profile. The WSS SOAP Message Security (WS-Security) supersedes the former WS-Security (private WS-*specification) as well as WS-Security Addendum (private WS-*specification), and proposes a standard set of SOAP extensions that can be used when building secure Web Services to implement message content integrity and confidentiality. WSS UsernameToken Profile and WSS X.509 Certificate Token Profile respectively describe how to use the UsernameToken and X.509 authentication framework with the WSS SOAP Message Security (WS-Security). In the near future, another specification: WSS SAML Token Profile will also be added to the WSS protocol set.

#### 6.3.1.1. WS-Policy

Provides a framework that allows Web Services to describe and communicate (publish) their policies to Web Service requestors. WS-Policy provides a general-purpose model and corresponding syntax to describe and communicate the policies of a Web Service. WS-Policy defines a base set of constructs that can be used and extended by other Web Services specifications to describe a broad range of service requirements, preferences, and capabilities. WS-Policy by itself does not provide a negotiation solution for Web Services.

#### 6.3.1.2. WS-Federation

WS-Federation is a specification produced by IBM and Microsoft to provide identity federation. It defines mechanisms that are used to enable identity, account, attribute, authentication, and authorization federation across different trust realms. It is part of the overall Web Services security framework (WS-Security).

The WS-Federation specification particularly describes models and a framework for federation of identity, attribute, authentication, and authorization information. The key driving requirements are to enable appropriate sharing of identity, authentication, and authorization data, brokering of trust and security token exchange, not requiring local identities at target services, and optionally hiding of identity information and other attributes. The models build upon the foundations specified in WS-Security, WS-Policy, and WS-Trust. WS-Trust is extended to allow attributes and pseudonyms to be integrated into the token issuance mechanism to provide federated identity mapping mechanisms.

WS-Federation provides federation of meta-data. Web Services may want to indicate where requestors can obtain security tokens in order to satisfy the services claims requirements. The mechanisms defined in WS-PolicyAssertions are therefore extended with a <wsse:RelatedService> assertion, allowing a trust domain to indicate where to find its identity provider (wsse:ServiceIP), its security token service (wsse:ServiceSTS), its attribute service (wsse:ServiceAS), and its pseudonym service (wsse:ServicePS).

WS-Federation provides a mechanism for cleaning up any cached state and security tokens that may exist within the federation. A requester may send a <wsse:SignOut> message to its security token service. This message is then federated/forwarded to security token services in other

domains where the requester is "logged in". The precise implication of a sign-out on currently active transactions is undefined and is resource-specific.

WS-Federation foresees an Attribute Service, with the possibility to expose attribute stores as UDDI endpoints.

WS-Federation specifies a Pseudonym Service through which parties (via a request/response protocol) can get, set, or delete pseudonyms. This is essentially a way a mapping an identity in one domain to another identity in another domain.

It provides part of the functionality of Liberty alliance (section 7.1.4.2).

#### 6.3.1.3. WS-Authorization and WS-Privacy

WS-Privacy will describe a model for how Web Services and requesters state privacy preferences and organizational privacy practice statements, while WS-Authorization will describe how to manage authorization data and authorization policies.

#### 6.3.1.4. WS-Attributed Based Access (WS-ABA)

WS-ABA is fine-grained negotiation-based access control model for Web Services. The goal of the model is to express, validate and enforce access control policies without assuming pre-established trust in the users of Web Services, while at the same time being in line with recent developments on identity management.

**WS_ABA** is an access control model for Web Services characterized by capabilities for negotiating service parameters. It is intended to be used within the SOAP standard.

The model allows express, validating and enforcing access control policies without assuming pre-established trust in the users of Web Services.

Access conditions are expressed in terms of identity attributes of the requesters. Moreover, in order to support a fine-tuning of access control, access conditions also take into account the parameters characterizing Web Services.

Once trusted, users can change dynamically their access requests in order to obtain authorizations. Other key features of WS-ABA are the efficient support of digital signatures and a mechanism supporting composed delegation. The model, the underlying architecture and the implementation are under development.

### 6.3.2. Grid security frameworks

Use of Grids places further demands on security solutions and these are discussed in section. 7.2.3.6 witihin the section on VO Management.

## 6.3.3. Adaptive and agile security

Virtual Organisations are highly dynamic as both members of the VO and infrastructure services may change dynamically throughout the lifetime of the VO. Furthermore, the security of the VO depends on the security enforced in each of its constituent members and the VO must change dynamically in order to react to security relevant events. Thus, there is a need to adapt the security infrastructure in response to events and to provide automatic methods for reliably establishing trust, detecting intrusions, adapting security policies to new situations and recovering from critical security conditions.

Adaptive Security, also known as Agile Security aims to provide these needs by establishing intelligent techniques to adapt the security needs of the dynamic environment. The main purpose of the Adaptive Security is to identify the problems and automatically correct them. Traditional access control mechanisms are often based on static access control configurations and do not provide adaptive security. In Adaptive Security, security does not rely solely on a set of static system configurations defined by a human administrator, but an ongoing adaptive process in which policy based techniques are used to provide automated configurations to dynamically handle security events. The security management follows the same feedback loop encountered in network and systems management, which includes monitoring, diagnostic/analysis and applying corrective actions (response)[4]. Monitoring involves instrument and detecting change in the environment where the VO is deployed or in the VO itself. The analysis phase includes both diagnostic and identification of the changes which have led to the events as well as deciding upon the corrective actions to be performed according to pre-defined policies. Response phase includes enforcing the configuration changes dictated by policy or decided during the analysis phase actions directed by the analyser.

Although advocated by many, there is relatively little work in the area of *adaptive* or *agile security*. The following sections present a few of the frameworks proposed. Most of them focus on the integration of existing security mechanisms or products e.g., intrusion detection in order to provide adaptive behaviour. A more reliable and flexible model would involve mechanisms for establishing and reasoning about trust, and dynamic negotiation of policies and security parameters.

### *6.3.3.1. Tivoli Risk Manager*

In large organisation, it is hard to predict the normal traffic of the network, and often intrusion detection tools can generate thousands of events a day with many often being false alarms. Often administrators do not know if the alarm is a malicious event. Firewall, intrusion detection tools, access control and Web Services all have different security functions, and often do not interoperate with each other. These independent vendor products have no interaction between them, and each have separate consoles are managed and administrated individually. A more effective solution would integrate these solutions to minimise security threats.

**Tivoli Risk Manager** developed by IBM aims to manage security threats, malicious users and other vulnerabilities across an enterprise security checkpoints by correlating security information

[4] Workshop on Logical Foundations of an Adaptive Security Infrastructure: http://www.aero.org/wolfasi/

and alerts from a set of devices including intrusion detection systems, firewalls, routers and networks. Tivoli Risk Manager comes with an intelligent engine, which correlates all the information from multiple sensors and presents a single alarm for each attack.

Tivoli Risk Manager tries to achieve the following main objectives:

- Provides a single centralised control point (a single web-based security console) to monitor and manage security alerts across the enterprise.
- Integrates products such as security applications (firewalls, anti-virus tools), networks and operating systems to provide a comprehensive security management environment.
- Helps system administrators to identify types of threats, pattern of intrusions and attacks accurately using advanced correlation techniques, aggregation and summarisation to speed the respond time and to reduce the false alerts.
- Provides a variety of pre-defined respond tasks (automatic tasks) to resolve urgent or severe security attacks such as denial of service attacks, policy violations or unauthorised access. These responds include disabling user accounts and reconfiguring firewall policies.
- Provides analytical historical reports to assess business risks and to support decision-making.

The Risk Manager architecture contains the following set of components[5]:

- ***Event Generating Components:*** A range of Risk Manager adopters and sensors collecting information about possible intrusions and passing this information as sensor events to the Risk Manager Correlation Engine.
- ***Real-Time Alerting Components:*** These components (including the Risk Manager Correlation Engine) are responsible for performing correlation and writing events to the event repository. The correlation is based on normalising the information, aggregation rules and situation analysis to compute the severity and of the situation. If there is an attack, the Correlation Engine sends situation alarms to administrators at a centralised console. The components include servers, which can respond to or modify the events automatically.
- ***Historical Reporting Components:*** These components consist of data mining and analysis mechanisms for decision support, and other components for risk management.

#### 6.3.3.2. Intelligent Security Infrastructure Management Systems (ISMS)

The Intelligent Security Infrastructure Management Systems (iSIMS) technology platform developed by Symbiot, has been recently released[6]. Similar to Tivoli Risk Manager, iSIMS interoperates with other security infrastructures such as firewalls, intrusion detection/presentation systems and virtual private networks to accumulate security events in real-time, and uses the security event data to build a risk model. The risk model provides information

[5] Recommended Practices for Risk Management with Tivoli Risk Manager. RedBook, published by IBM, 2002: http://publib-b.boulder.ibm.com/Redbooks.nsf/RedbookAbstracts/redp0202.html?Open

[6] Symbiot Security Web Site: http://www.symbiot.com/isimstechnology.html

about the *significance* (how threatening is an attacker) and the *impact* (the cost of the attack if it succeeds). The system measures cost in dollar terms.

The measures of threats are defined using a risk score. The risk metric used is similar to a credit score provided by the credit reporting bureaus. Risk scores are used throughout the iSIMS platform and the Symbiot.NET repository to provide accountability, consistency and standardisation. The Symbiot.NET is the central repository containing attacker attribution profiles based on cooperative surveillance and reconnaissance collected by network participants. The knowledge base of iSIMS is updated using the data from the Symbiot.NET. This data is used to identify attackers, what they can do, evaluates their methods and intentions and recommend appropriate countermeasures.

The *Expected Value of Risk,* EV(R) is a function of the following terms integrated over time[7]:

$$EV(R) = \int \left[ P_{THREAT}(t) \cdot P_{VULNERABILITY}(t) \cdot Value(t) \right] dt$$

The probability of threat $P_{TREAT}(t)$, is an estimator derived from the cumulative alerts generated by existing security components and can be self-generated by the ISIMS platform based on anomaly, behaviour and signature analysis of the live network traffic. The probability of vulnerability, $P_{VULNERABILITY}(t)$, is estimated based on an evaluation of how the network will respond to threats. Value is determined using both the asset cost and the Net Present Value (NPV) of the revenues associated with the exposed infrastructure.

Symbiot.NET provides a model of *graduated response* against intrusion events, from simple techniques such as blocking traffics to more aggressive operations. The iSIMS platform and Symbiot.NET attacker knowledge base use a range of rules to establish recommendations for countermeasures to be enacted including blocking traffic, rate-limiting (adjusting the bandwidth available), diverting traffic and quarantine (redirecting into a special area for analysing the characteristics).

### 6.3.3.3. Adaptive Security Policies

The work presented in paper[8] was one of the first to advocate the use of adaptive security policies in highly secure environments. It argues computer security policies must be adaptive to react to changes in the security environment. The paper compares various methods for implementing security policies by separating the definition of the policy in a security server from the enforcement, which is done by the microkernel. A prototype operating system, the Distributed Trusted Operating System (DTOS) is used to evaluate policies. The DTOS design consists of a microkernel and a collection of security servers. Security servers define the policies enforced by the microkernel. When a request for a service is made to the kernel, the kernel submits the request with various information such as security context of the subject and object,

[7] Enterprise Specific Event Significance: http://www.symbiot.com/es2.html

[8] M. Carney and B. Loe, A Comparison of Methods for Implementing Adaptive Security Policies. 7th USENIX Security Symposium, 1998, San Antonio, Texas, USA.

to the security server to determine if the access is permitted. The security policies are similar to firewall rules.

This paper presents and compares a number of approaches for changing the security policies to adapt to dynamic security environmentthat are further developed in[8]. However, these approaches require either reloading of the policy or changing the algorithm, which the security server uses to make its security computation. None of the proposed methods provide a concrete solution and they have weakpoints.

More recent work[9] argues that the policy implementation defined in the DTOS is not effective and scalable, and proposes an authorisation framework to specify and enforce security policies, which assist in detecting and responding to security attacks and misuse. In[9], security policies accommodate changes in the security requirements and assist in detecting and responding to intrusion and of abuse of user privileges. Authentication policies can require more information from a user when suspicious activity has been detected. For example, a policy can be specified to contain the followings[10]:

Alice can run a process on host doc.ic.ac.uk. If the request fails, a notification must be sent to a system administrator. The process must not consume more than 10% of the CPU. An audit record about the completed process must be generated.

Here Alice, 10%, notification and audit are conditions, run is the access right and doc.ic.ac.uk is the target object. The policies are defined using a range of conditions that provide run-time adaptation in the event of possible attacks or misuse. Conditions include access identity, authentication mechanisms, time (periods for which access is permitted), and location of the user, payment, system threat level and notification. Failure of conditions may indicate distrustful behaviour. The conditions are classified as *pre-conditions* (conditions which must be satisfied before the execution), *request-result conditions* (conditions which must be activated when the authorisation is granted or denied), *mid-conditions* (conditions which must be true during the execution) and *post-conditions* (conditions which must be satisfied after execution). Some of these conditions may trigger simple responses such as limiting the resource computation.

To enforce the policies, they adopt a three-phase policy enforcement scheme; access control phase, execution control phase and post-execution actions phase. They make use of Generic Authorization and Access-control API (GAA-API)[9], which provides a general-purpose execution environment in which policies are evaluated. The GAA-API returns the status values for each phase to describe policy enforcement process.

#### 6.3.3.4. Security Agility for Dynamic Execution Environments

Security agility[11] is a technique, which extends the functionality of software components to make them aware of their dynamic security environment and adapt to policy changes, and respond to

[9] T. Ryutov and C. Neuman, The Specification and Enforcement of Advanced Security Policies. 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY'02), Monterey, California.

[10] T. Ryutov, The Specification and Enforcement of Advanced Security Policies. Presentation slides: www.policy-workshop.org/2002/Files/ Slides/AdvancedPolicies.pdf

[11] M. Petkac, L. Badger and W. Morrison, Security Agility for Dynamic Execution Environments. Proceedings of the DARPA Information Survivability Conference & Exposition Volume I of II, Hilton Head, South Carolina.

intrusion detection. These components are able to enforce their part of the global policy, and contain internal mechanisms to automatically adapt when security policies change. Figure 11 shows an abstract view of the strategy. Each component has a rule set which provide components with built-in knowledge of security policies, models, and mechanisms to adapt to changes. These components also interact with the Agility Authority to receive policy updates. The Agility Authority transmits authorised security policy change requests to agile security components, which then dynamically behave according to the new security policy requirements. In response to a policy change, a component may take a number of actions, for example, terminating connections, tighten access to resources, changing cryptographic algorithms or accepting new security enforcement responsibilities.

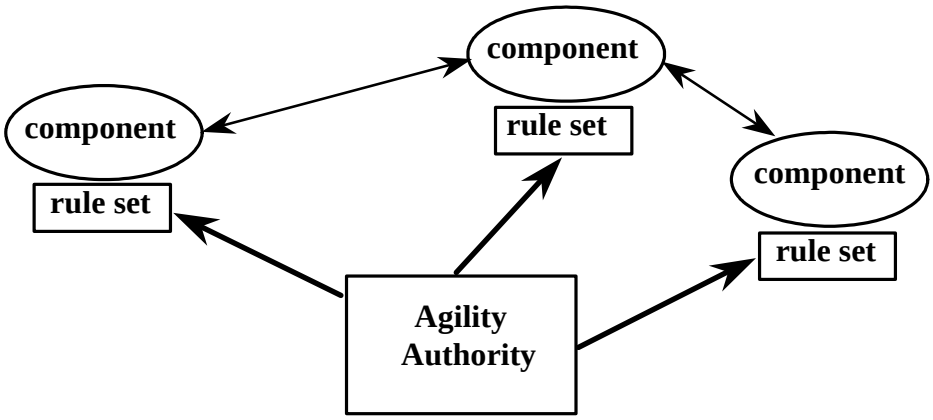


**Figure 11 - Security Agility Solution Strategy[12]**

Figure 12 shows the general architecture of a security *agile* component and indicates the interaction of the key elements of the security agility toolkit. The component-specific code implements the component's non-security responsibilities. The security-specific functionality is carried out by the agility subsystem. The component-specific code includes a number of Control Transition Points (CTPs), which are interfaces that conditionally transfer control to the security agility subsystem. The CTPs provide the agility subsystem with an outline of the component's behaviour. The subsystem uses this information to provide the appropriate security services (e.g., cryptography) on behalf of the component.

The security-agility subsystem is able to carry some reconfiguration internally (e.g., closing files) but it may require some help from the component-specific code. The security-agility subsystem will invoke callback functions to achieve such tasks. The agile system component receives notification of dynamic changes to the system security policy and information to modify the application-level security policy from the Agility Authority.

[12] Project Profile, *Security Agility for Dynamic Execution Environments*, NAI Labs security Agility Research, 2001

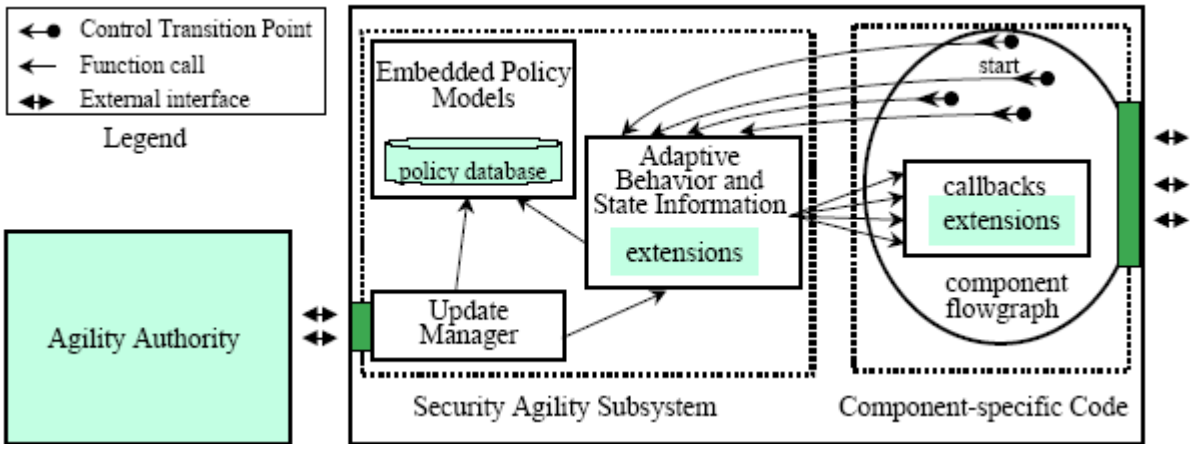


**Figure 12 - Security Agile Component Architecture[86]**

# 6.4. Service Level Agreement (SLA)

## 6.4.1. SLA definition languages

A Service Level Agreement is a formal negotiated agreement between a service provider and its customer, the service requestor. When a customer orders a service from a provider, an SLA is negotiated and a contract is drawn up. The contract involves parameters relating to the service to be provided. A parameter may relate to a desired Quality of Service (QoS) which may include aspects of networking, the execution of which are described in sections 3.8and 4.1.4. The service provider must perform SLA monitoring in order to verify whether the QoS parameters specified in the SLA contract are respected. The SLA monitoring involves monitoring the performance status of the offered service and provides relevant information to the service level management system. Then, the management system could assess the provider's commitments and applies penalties if those commitments weren't met.

Service level management may apply to a single or a group of Web Services within the same domain or business field. It consists of the monitoring of operations of the Web Service and the control of the Web Service to meet the guaranteed service and QoS.

In order to define the mechanisms for such commitments, several specifications have been recently carried out defining several XML based languages, enabled to describe the contract between the service provider, its customer and a possible third party. These languages were defined closely to a common language allowing a common understanding of the service provider commitments to perform a service according to agreed guarantees. Several languages have been defined (WSLA specification language [13,14], etc), and all of them are a complement to the service

[13] Ludwig, H., Keller, A., Dan, A., King, R.P., Franck, R.: Web Service Level Agreement (WSLA) Language Specification, Version 1.0, Revision wsla-2003/01/28. International Business Machines Corporation (IBM). On-line at:http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf (2003)

[14] Keller, A., Ludwig, H.: The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. Journal of Network and Systems Management, Vol. 11, No 1 (Mar. 2003) Plenum Publishing (2003)

description implemented by WSDL [15]. In general, such languages are used within a framework allowing the management of Web Services and their compositions.

In the following we present an overview of recent works most relevant to the GRASP project dealing with Service Level Agreement in WS context, and we also present the OGSI-Agreement specifications (WS-Agreement).

#### 6.4.1.1. Web Service Level Agreement (WSLA)

The WSLA Language, developed by IBM[13,14], defines a type system for various SLA artifacts and it is based on the XML schema. The process of reaching a Service Level Agreement must be possible to be automated and, in order to facilitate this, an SLA in the WSLA contains three parties:

- Parties' section: identifies the actors involved in the contract agreement. It distinguishes the signatory parties (signing the contract) and the supporting parties (sponsoring parties), which provide services to the signatory parties, such as measuring the service's parameters. This section contains the contact information related to each party (representative, address, …).
- Service Description Section: defines the SLA parameters of the service. These parameters are defined by Metrics, which define the way to measure and evaluate them (measurement directives). This description also specifies how to aggregate metrics to form a composite metric.
- Obligations Section: this last section of an SLA defines the obligations and actions guaranteed in case of violation of Service Level Objectives (SLO), respective to the state of the related SLA parameter. The SLO contains the guaranteed condition of a service for example sending a notification on violation events.

##### 6.4.1.1.1. Standard extension

The WSLA language provides a mechanism, defined as abstract, to extend its various types in the core WSLA language. This abstraction allows defining new specific domains from existing language elements, using XML schema derivation such as business topics. This extension provides to the authors of an agreement the capability to facilitate relating the WSLA to a WSDL-defined service and a WSDL-defined management actions, defining common metrics in the context of Web Services and a set of standard predicates to define the contractual guarantees.

##### 6.4.1.1.2. Runtime Architecture

The following figure represents the WSLA monitoring framework building blocks (Figure 13).

[15] http://www.w3.org/TR/wsdl

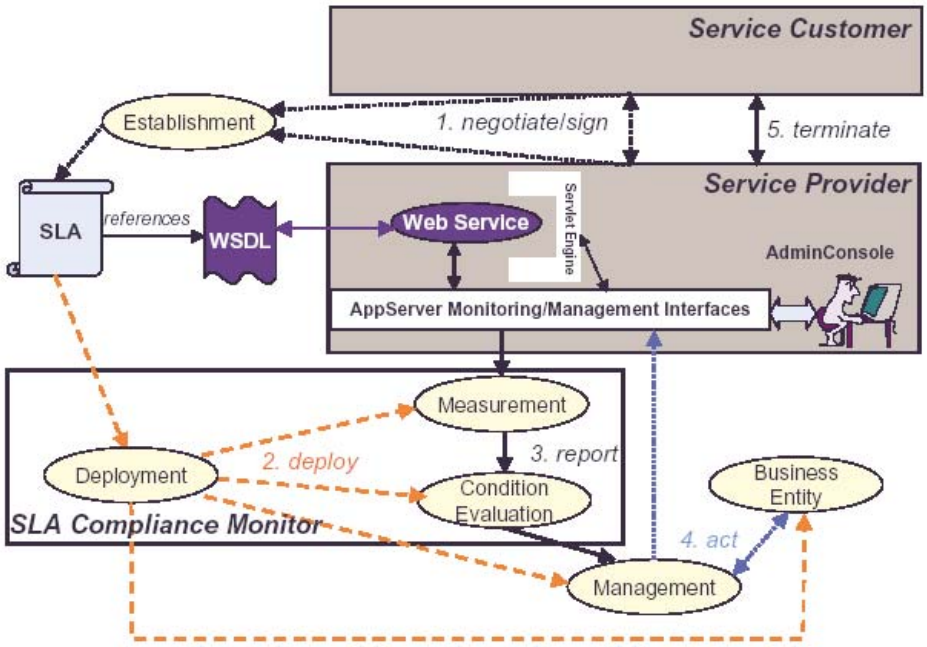


**Figure 13 - Runtime Architecture**

The interface of a Web Service is defined by an XML document in the WSDL [15]. The SLA references the WSDL document and extends the service definition with SLA management information. The SLA management lifecycle consists of five distinct phases:

Phase 1: the negotiation:

This phase consists of the drawing up of an agreement between the service provider and the customer for an offered service. In this phase the two parties establish different aspects of the SLA: QoS parameters, price, third parties and their roles, and so on. An SLA document is produced containing all elements of the contract making it available for deployment.

Phase 2: the deployment:

This service allows the WSLA interpretation and the set up of the corresponding components managing the SLA. Each party (Service Consumer and Provider) is responsible for the deployment of its functions and the set up of its supporting parties if necessary. This information is passed in a standard format from one party to another.

Phase 3: Measurement and Condition Evaluation Service:

This phase deals with controlling during the runtime system whether the QoS objectives agreed by the provider are reached in monitoring the SLA parameters and evaluating the violation of the SLO.

The measurement Service collects runtime information on the metrics related to SLA parameters. It notifies to the Condition Evaluation Service the results of the monitoring. Multiple measurement services may simultaneously measure the same metrics.

The goal of the Condition Evaluation Service is to determinate the probable violation of guaranteed QoS regarding the collected metrics. This can be done each time a new value is available, or periodically.

Phase 4: Corrective Management Action

Once a SLO parameter has been violated, corrective actions contained in the SLA need to be carried out. This functionality apply a two-pronged services:

*Management service*: Following a receipt of violation of SLO notification, this service must carry out the corrective actions described in SLA document. Before acting, it requests the business entity in order to verify if the proposed actions are allowable.

*Business entity*: It contains the business policy of the service provider and verify if the corrective actions specified in a SLA, some time ago, are still in accordance with the business targets.

Phase 5: SLA Termination:

The conditions of termination of the service must be specified in the SLA document. These conditions must describe the penalties in case of breaking one or more SLA clauses. This could be negotiated between service provider and consumer in the Negotiation phase.

The "SLA compliance Monitor", tool has been developed for the WSLA language by IBM and it is included in the IBM Web Services Toolkit.

The WSLA framework enables specification of detailed SLA parameters, management information of the system, price/penalties for Web Services, but the monitoring resource is not already well-specified. Another WSLA design goal, is to address the "wide variety of SLAs" by providing an SLA template document which includes several automatically processed fields, limited to a small set of variant services using the same kind of SLA parameters.

The integration with existing resource management is work still in progress and special attention is paid on Common Information Model (CIM), see section 3.7. The WSLA infrastructure is powerful but very complex to perform.

#### 6.4.1.2. SLA notification generation (SLAng)

SLAng, SLA notation generator[16,17,18], is an XML-based language designed to produce a formal language with a well defined syntax and semantics, for describing Service Level Specifications (QoS parameters) (SLS) in the domain of distributed systems and the context of e-Business. This

[16] Lamanna, D.D., Skene, J., Emmerich, W.: SLAng: A Language for Defining Service Level Agreements. In Proc. of the 9th IEEE Workshop on Future Trends in Distributed Computing Systems - FTDCS 2003 (Puerto Rico, May 2003). IEEE-CS Press (2003) 100-106

[17] Skene, J., Lamanna, D.D., Emmerich, W.: Precise Service Level Agreements. International Conference on Software Engineering (ICSE) 2004

[18] Lamanna, D.D., Skene, J., Emmerich, W.: Specification language for Service Level Agreements. Document submitted to EU IST Project 34069 TAPAS. March 2003.

work has been carried out by the TAPAS project[19]. The approach of SLAng is to define an SLS at different levels: the application level as well as the application service (ASP), systems resources and so forth.

#### 6.4.1.2.1. SLAng model

SLAng introduces a reference model for inter-organisational service provision at storage, network, middleware and application level for a distributed component architecture.

The following figure shows the depicted model:

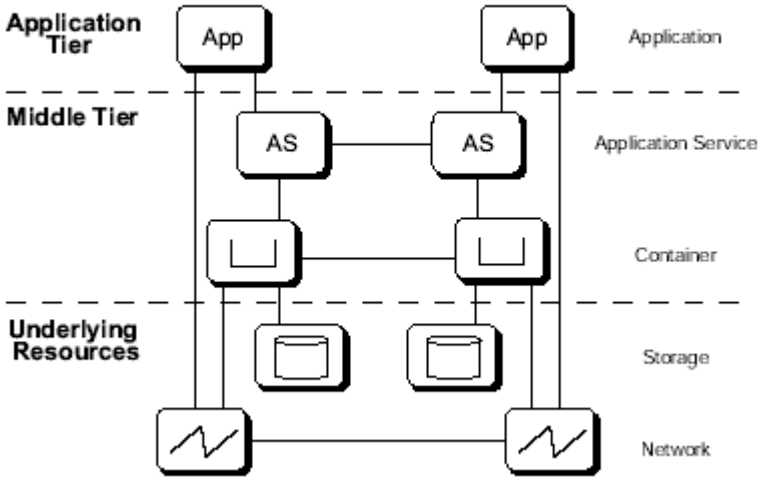


**Figure 14 - Service Provision Reference Model**

This traditional layered architecture points out that service provisioning could occur at any level of the architecture and different parties could provide or consume services.

SLAng defines six different types of SLAs: three vertical and three horizontal. These types regulate the type of agreement occurring between the different type of parties of this architecture.

The vertical ones are:

- Hosting: between service provider and host
- Persistance: between a host and storage service provider
- Communication: between container and network service provider

The horizontal are:

- Service: between an Application or service and ASP
- Container: between containers providers

[19] http://www.newcastle.research.ec.org/tapas/

- Networking: between network providers

This cross-layered architecture aims at taking into account the whole interactions cases between the actors taking a share in an e-business model.

#### *6.4.1.2.2. SLAng Language and semantic*

As previous languages, SLAng syntax is based on the XML schema. This language has been modelled by the use of Unified Model Language (UML) and Object Constraint Language (OCL) in order to precisely define the meaning of service level agreement.

The description of an SLA in SLAng has the following structure according the different types of SLA below:

- End point description of the contractors containing information on provider-customer,
- Contractual statement defining the agreement itself: kind of service, duration of the agreement, charging clause, violation clauses, and so on
- Service Level Specification describing the QoS parameters related to the service and the metrics associated.

The notion of a mutual responsibility between the customer and the provider of a service is defined in each SLA in order to take into account the bilateral aspect of the contract.

This approach enables SLA management in distributed systems in an e-Business context and focuses not only on Web Services issues but also on different types of SLA. It has a broader scope compared to the IBM and HP approach. But the detailed mechanism to create a Service Level Agreement is not described by SLAng. Another aspect of SLAng, the definition of the QoS metrics, is defined into the SLAng schema, so only predefined SLA formats could be used.

SLAng seems less flexible than the other approaches.

The TAPAS project supplements the SlAng approach by implementing the QoS functionalities of the architecture using J2EE technologies, in particular with JBOSS and JONAS.

### *6.4.1.3. Web Service Offering Language (WSOL)*

WSOL (Web Service Offering Language) is a language for specification of constraints and classes of service for Web Services[20,21,22]. The syntax of WSOL is defined by using XML schema. WSOL is fully compatible extension of WSDL.

---

[20] Tosic, V., Ma, W., Pagurek, B., Esfandiari, B.: Web Service Offerings Infrastructure(WSOI) – A Management Infrastructure for XML Web Services. Submitted for conferencepublication. Also published as: Research Report SCE-03-19, Department of Systems andComputer Engineering, Carleton University, Ottawa, Canada, Aug. 2003. On-line at:http://www.sce.carleton.ca/netmanage/papers/TosicEtAlResRepAug2003.pdf (2003)

[21] Tosic, V., Pagurek, B., Patel, B. Esfandiari, B., Ma, W.: Management Applications of the Web Service Offerings Language (WSOL). In Proc. of the 15th Conference On Advanced Information Systems Engineering - CAiSE'03 (Velden, Austria, June 2003). Lecture Notes in Computer Science (LNCS), No. 2681. Springer-Verlag (2003) 468-484.

This language allows to enable a Web Service to offer several different "Classes of service". To consumers that means a Web Service could provide different service levels defined by several classes of service. Classes of service can differ in usage privileges, service priorities, response time guaranteed to consumers, etc. The classes of service are defined at the level of Web Service and not QoS constraints.

WSOL defines the following constructs:

Constraint:: three types of constraint are defined:

- Functional constraints are specific to the execution to a Web Service operation to be functionally correct. WSOL enables specification of several functional constraints such as pre-, post- and future conditions, as well as invariants[22]. A third party can evaluate functional constraint.
- QoS constraints (non functional) are specific to an operation invocation such as performance, reliability, and so on. Non-functional constraints check whether the monitored QoS metrics are within specified limits. These constraints are described by an external ontology of QoS metrics and measurement units. These external ontologies definitions can be reused for different Web Services.
- The Access rights constraint specifies conditions under which any consumer has the right to use any invoked operation in the service offering.

Management statement:: this enables the specification of important information about the represented class of service offered. Three main XML schemas for types of management statement are defined:

- Price statement: cost of the use of a particular operation of the offering service.
- Monetary penalties statement: amount of money the service provider have to pay to the consumer whether it could not fulfil the constraints in the service offering.
- Responsibility statement specifies the management responsibility of a management entity. A management entity could be the service provider, the service consumer or a third party.

Service offering (SO): A WSOL service offering contains the formal representation of various constraints and management statements that determine the corresponding class of offering. WSOL service offering can be viewed as one simple contract or one SLA between the service provider, consumer and eventually a management third party trusted by supplier and consumer.

---

[22] Tosic, V., Patel. K., Pagurek, B.: Reusability Constructs in the Web Service Offerings Language(WSOL). In Proc. of the Workshop on Web Services, e-Business, and the SemanticWeb (WES) at CAiSE'03 (Velden, Austria, June 2003). LNCS, Springer-Verlag. Extendedversion published as: Res. Rep. SCE-03-21, Department of Systems and Computer Engineering,Carleton University, Ottawa, Canada, Sep. 2003. On-line at:http://www.sce.carleton.ca/netmanage/papers/TosicEtAlRepSeptember2003.pdf (2003)

Reusability elements: WSOL gives the possibility to reuse constraint and management statement constructs enabling easier derivation of a new service offering from existing service offerings of the same Web Service or other Web Services by using inheritance, inclusion or template instantiation. WSOL defines several special reusability elements [22].

Service offering dynamic relationship (SODR): SODR states what service offering could be replaced by another whether constraints from the used service offering cannot be fulfilled. These relationships are specified in a file outside of the WSOL file in order to avoid frequent modifications of the service offering definitions.

A Web Service Offerings Infrastructure (WSOI) also integrates, on top of Apache Axis, solutions for monitoring WSOL service offerings. In WSOI, monitoring and accounting activities are developed through a specific handler [21].

This infrastructure needs to be easily adaptable thanks to the reusability constructs, which can offer a comparison between different Web Services, and especially to reusability for SLAs. This confers a means of comparing Web Services. But, this approach reduces any possibilities to enable instantiation of SLA on-demand, the concept of reusability involves a static definition of the SLA (predefined) for a service given. However, it provides solutions that are relatively simple to use and implement and lightweight in terms of run-time overhead.

### 6.4.2. Existing SLA management approaches in Grid environment

Here are some existing approaches for SLA management in the field of grid computing.

- Service Negotiation Acquisition Protocol (SNAP)
- GRASP (GRid Application Service Provider)

### 6.4.3. SLA monitoring

Here we describe possible models to enable the monitoring of execution compliance with the Service Level Agreement

#### *6.4.3.1. WS-Agreement*

In June 2003 the GGF released Version 0 of the OGSI-Agreement specification [143], which proposed a general agreement based management of Grid Service instances. This initial version suggested the usage of Agreement Grid Services. Beside the fact that OGSI-Agreement, Version 0, naturally was quite general and lacked concrete implementation approaches, Agreement creation was done by invocation of Factory::createService with appropriate arguments, leading to a fault or the creation of a new service. Thus Negotiation and Instantiation were logically coupled.

Recently there has been a significant evolution of OGSI-Agreement that culminated in the release of Version 1.0, now called WS-Agreement[23]. In the following we will give a short overview on the key concepts of WS-Agreement. In general, WS-Agreement aims to define a language for negotiation and monitoring of agreements between a service client and a service provider. The assumed key requirements are:

- Description of agreements about services independent of the domain
- Creation of agreements about single services as well as collaborating services
- Support for different condition languages used to define service level objectives or constraints
- Independence of the negotiation model
- Combination with any WS-* (WS-Policy, WS-Addressing, …) specification

As a consequence of the extensibility requirements above, specific condition expressions, service descriptions and metric definitions languages are outside the scope of WS-Agreement. Figure 15 shows the structure of an Agreement, which is built upon the following elements:

- Context – contains the participants and other information such as the termination time of the agreement
- Service Description – contains domain specific service descriptions and necessary information to interact with the service instance
- Guarantee Terms - includes the condition collection under which a service will be executed, often referred to as Service Level Objectives
- Negotiability Constraints – may describe rules for the negotiation phase

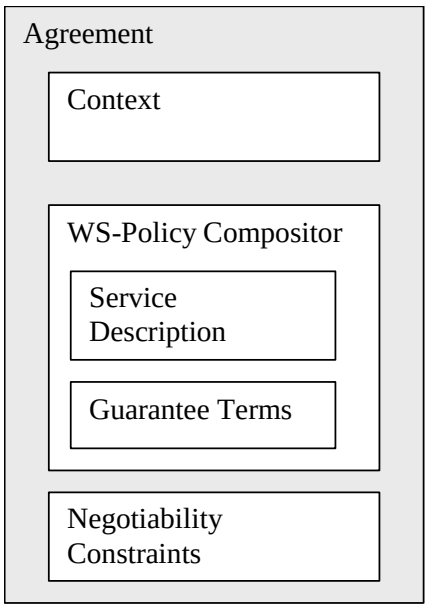


**Figure 15 - Agreement Structure**

[23] https://forge.gridforum.org/projects/graap-wg/document/WS-AgreementSpecification/en/2

The *service description* is what a service provider agrees to provide, specified in a domain-specific way using a domain specific description language.

The *guarantee terms* specify the actual service levels that the parties have agreed on, in other words the values of the service attributes given in the service description. A guarantee term consists of three parts: qualifying condition, service level objective and business value. The qualifying condition may be based on service attributes or external factors such time. A service level objective is a set of the values of the service parameters that have to be met by the service provider. The business value is specifying a priority on the objective either by defining an *importance* value on the objective, or by specifying a penalty as the consequence of violation of the service level objective. The management system uses the guarantee terms to monitor the service and enforce the agreement.

The negotiability constraint can be given by a party to restrict the number of offers to be exchanged between him and other parties during the negotiation phase of the agreement. It is not by any means interpreted as a promise by the party but the values that it generally is willing to accept.

In the version 1.1 of the WS-agreement specification dated 2004-04-26, the conceptual model is based on two layers, the service and the agreement layer, as shown in the Figure 16, below.

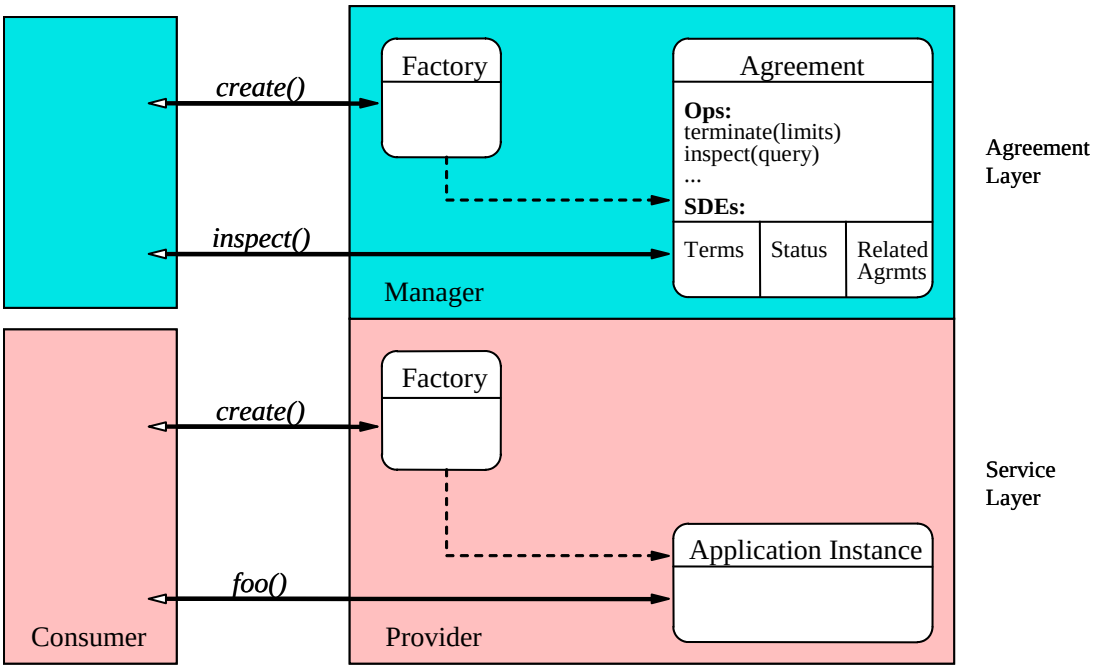


**Figure 16 - WS-Agreement Conceptual Layered Service Model**

The service layer is simply the application-specific layer of business services being provided. These services may or may not be specified by Web Service interfaces.

The agreement layer is a Web Service-based interface that can be used to represent and monitor agreements with respect to provisioning of services implemented in the service layer.

## 6.5. Portals

### 6.5.1. Web Service Remote Portlet

The Web Services for Remote Portlets specification defines a Web Service interface for accessing and interacting with interactive presentation-oriented Web Services. It has been produced through the joint efforts of the Web Services for Interactive Applications (WSIA) and Web Services for Remote Portals (WSRP) OASIS Technical Committees.

After the introduction of Web Services as the means for integration of business logic via the Internet, the Web Services-based WSRP standard is the means for integration of Web Services based presentation components. WSRP provides a standard that enables all content and application providers to provide their services in a manner where they can easily be discovered and plugged into all compliant portals without programming effort on the portal's side. Portal administrators can find and integrate the WSRP services they need with just a few mouse clicks, typically by using their portal's administration user interface to browse a registry for WSRP services and then select some for automatic integration into the portal. Portals act as intermediaries between end users and WSRP services (v1.0) and aggregate services from many different content providers. They often offload significant traffic from content providers by caching content, thereby enabling content providers to serve a huge number of users with little IT infrastructure.

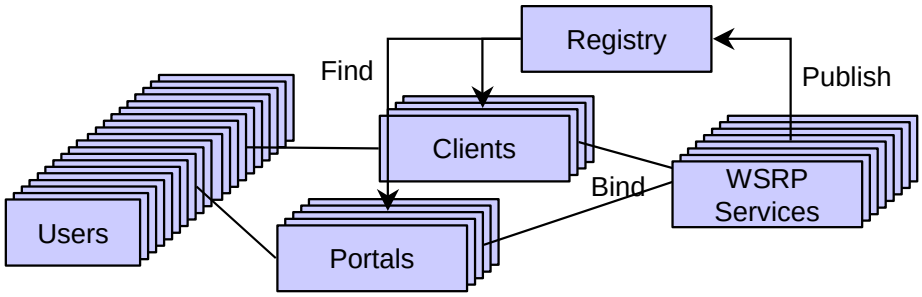


**Figure 17 - A Portal using Remote Portlet Web Services**

By providing WSRP services, content and application providers can leverage portals as multiplying intermediaries to reach a number of end users that they never could have reached otherwise.

Portlets are presentation-oriented, interactive web application components designed to be aggregated and communicate with users through the hosting portal server. Typically, portal servers maintain a catalog of available portlets from which end users can select portlets for placement on portal pages. Portlets may be local or remote to a portal server. Local portlets are usually tightly integrated with portal servers and typically run on the same physical server or cluster of servers. WSRP services run on remote servers at other places in the intranet or the Internet and are loosely coupled to the portal server.

Today, many content providers publish their content live on the Internet using HTTP or FTP servers or they provide client software that replicates and caches content via proprietary protocols. In each case, integrating content into a portal is a difficult task. In order to allow for easy integration of their content in portals, content providers can use WSRP to surface their content as remote portlets and publish them as WSRP services in public registries.

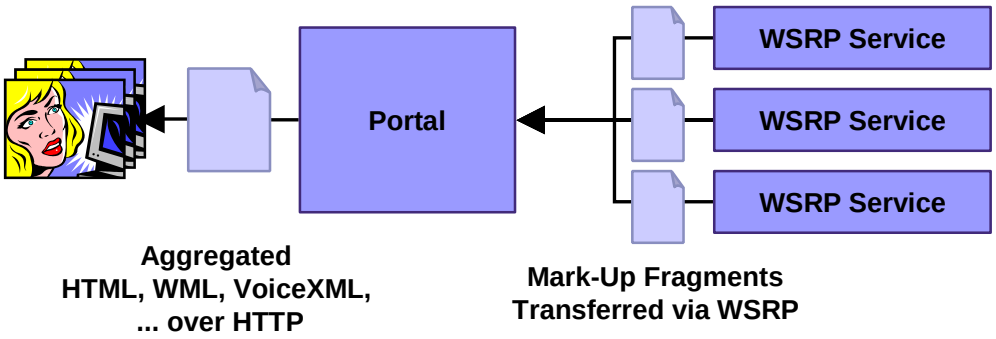


**Figure 18 - Content/Application Providers providing WSRP services**

In order to provide this value add to subscribers, the content provider serves remote portlets via the desired bindings in addition to any classical content server. Once the content provider has published a WSRP service in registry, it can be added by administrators of portals wishing to use the content provider by simply looking up the content provider's business entry in the registry and binding to the desired WSRP service. The portlets on the content provider's server become available immediately without any programming or installation effort and are now available to the portal's users.

WSRP fits into the greater context of the Web Services standards stack. It uses WSDL to formally describe the WSRP service interfaces and requires that at least SOAP binding be available for invocations of WSRP services.

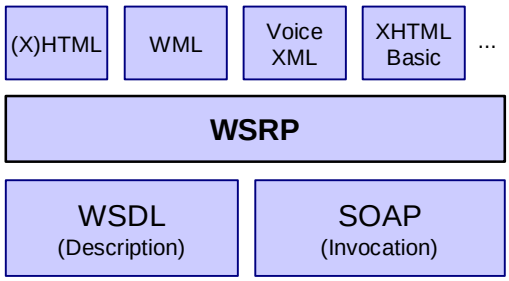


**Figure 19 – WSRP and related Standards**

WSRP defines the notion of valid fragments of mark-up based on the existing mark-up languages such as HTML, XHTML, VoiceXML, cHTML, etc. For mark-up languages that support CSS style definitions, WSRP also defines a set of standard CSS class names to allow portlets to generate mark-up using styles that are provided by WSRP compliant portals such that the mark-up assumes the look and feel of the consuming portal.

A goal of the WSRP standard is to make it easy to implement simple services (e.g. just provide mark-up fragments), but also allow for more complex services that require consumer registration, support complex user interaction and operate based on transient and persistent state. WSRP will enable interoperability of portals by allowing them to consume remote portlets provided by other portals or content / application providers as well as the sharing of local portlets for remote access by other portals in a standardized manner.

### 6.5.2. Application in Akogrimo

## 6.6. References for Grid Application Support Services Layer


[143] OGSI-Agreement Specification Version 0, http://www.globus.org/research/papers/OGSI_Agreement_2003_06_12.pdf

[144] http://www.icewalkers.com/Linux/Software/517920/Grid-Application-Framework-for-Java.html

[145] http://www.gridlab.org/

[146] http://gridblocks.sourceforge.net/Portal.htm

[147] http://www.neresc.ac.uk/projects/gaf

[148] http://www.cactuscode.org/

[149] www.eu-grasp.net

[150] IBM and Microsoft. Security in a Web Services World: A Proposed Architecture and Roadmap. April 2002. http://msdn.microsoft.com/webservices/?pull=/library/en-us/dnwssecur/html/securitywhitepaper.asp

[151] Service Composition, http://www.serviceoriented.org/service_composition.html

[152] Web Service Orchestration, http://www.serviceoriented.org/web_service_orchestration.html

[153] Chris Peltz, Web Services Orchestration, http://devresource.hp.com/drc/technical_white_papers/WSOrch/WSOrchestration.pdf

[154] Dipanjan Chakraborty, Filip Perich, Anupam Joshi, Tim Finin, and Yelena Yesha, A Reactive Service Composition Architecture for Pervasive Computing Environments, http://ebiquity.umbc.edu/v2.1/paper/html/id/56/

# 7. Analysis of cross layer themes

For some issues, subdividing into layers loses some coherence. We regard some themes as having a cross-layer significance and discuss them here.

## 7.1. User management and identity model

### 7.1.1. Overview

The goal of this section is to provide a state of the art description of user management and identity systems in Grids.

It is worth mentioning that Grids, due to the great computing power they are able to provide, have from the beginning been aimed to scientific environments, in which security and other tangential issues were secondary or not needed at all. However, today a new tendency has arisen that pushes Grids to a totally new field for this kind of technologies, which is the commercialization of the role of the Grids. The fact of introducing Grids in a commercial environment introduces new requirements that were not needed before. A secure framework has to be provided together with AAA services (authentication, authorization, accounting). These requirements can not be provided or do not make sense without the introduction of the concept of user/identity management. If there is a need to authenticate someone/something, it is obviously also mandatory to clarify "who" that someone/something is. Identity management becomes necessary in order to impel a commercial structure, in which services are offered to certain customers as a result of a trade contract.

### 7.1.2. Identity

A description of what identity is is necessary in order to fully understand the use of this concept in Grids.

Identity is defined as:

*"The collective aspect of the set of characteristics by which a thing is definitively recognizable or known"* [197]

The set of characteristics that describes an individual can be of a vast variety of types and depend on the context. For example, a credit card identifies one as a customer a certain bank, a passport as citizen of a country, the DNA identifies a person physically. In addition an individual can be further described by its environment and preferences such as: medical history, entertaining preferences, education history. And going more in detail, more ephemeral characteristics may be added such as mood, availability, location…

In a commercial environment, the highest and most important degree of identity comes from the trade contract established between a customer and the entity that offers a determined service. The concept of identity is not necessary linked to a physical person but to a group of people, an entity, a service…

### 7.1.3. Requirements

In order to build a structure on top of which the management of identities contributes to the successfully creation of a commercial Grid, the following aspects must be taken into account:

- AAA and Identity: In order to authenticate, authorize and account a user there is a need to map its identity with all the possible material correspondent to the AAA field such as, cryptographic material to authenticate the user, information about what the user is allowed to do, etc…
- Federation: In a highly heterogeneous network such as the one enabled by Grids, it is not possible to keep a centralized, standard way of keeping user information. In addition, due to the commercialization of the Grid, a large number of administration domains and services take part of the of the whole structure, existing different contracts between customers and service providers, which implies that the user has different identities, depending on the provider he is facing at that moment. This also means independent AAA services and therefore, different authentication material, logins, etc… In order to decrease the complexity of the system from the user point of view, and to provide new functionalities to ease the expansion of the commercialization, identity federation is mandatory. With identity federation, several entities with which a user interacts with can exchange identity information in order to provide different added values:
    - Collaboration: allow services from different providers to be able to fetch identity information. This can be used, among other things, to synchronize list of contacts, the calendar, etc…
    - Single account: allow users to make use of different services using a single account and not having to open a new account every time a new service from a different service provider is used.
    - Bill aggregation: allow users to aggregate bills from different accounts and from services from different services provider in a unique bill.
    - Single Sign-on and Log-out: allow the user to use different services from different service providers without having to sign-on each time by having the service providers interchange the authentication information related to that identity. The same concept is used in order to log-out: a user can log-out from each account by performing a single log-out.
    - Anonymity: a user can choose to remain anonym when using a service. This is done by having the user sign-on to a service making use of a "virtual" identity, which the service will authenticate with an identity provider, which the user has a contract with.

### 7.1.4. Existing Work

There exist some projects related to the development of identity structures. Some of the most significant ones are presented.

#### 7.1.4.1. SAML

The Security Assertion Markup Language (SAML) is a proposed standard, developed by the OASIS XML-Based Security Technical Committee (SSTC), which aims to provide a means in order to transmit user information from an entity to the other. SAML makes use of widely known technologies such as XML and Simple Object Access Protocol (SOAP) to build a basis upon which the user related information can be suitably exchanged. The information that SAML permits to share is conveyed between entities by means of data structures called assertions. Different kinds of assertions can be distinguished:

- Authentication assertions: declarations about a user's identity.
- Attribute assertions: contain particular details about a user.
- Authorization decision assertions: dictate what a user is authorized to do at a particular site.

SAML is designed to be independent from the underlying transport protocol.

OASIS released in March 2005 the final documents for SAML v2.0 [207]. The SAML v2.0 specifications have been fed by the OASIS committee, the Liberty Alliance and the Shibboleth project. SAML v2.0 provides a unified framework in which different forms of federation are supported.

Currently, all SAML implementations (Netegrity, Oblix, Entrust…) were developed for SAML 1.1 specifications.

#### 7.1.4.2. Liberty Alliance

The Liberty alliance project [198] [199] is an alliance of over 150 companies which was formed in September 2001. The consortium aims to develop open standards for network identity management without providing a specific product or service. Starting up from the basis of SAML, this project builds upon it to enable a consistent federated identity framework.

The Liberty alliance project is divided in three phases in which new features are added to the basis.

The first phase is the Identity Federated Framework (ID-FF) and gives support for the following main features:

- Simplified Single Sign On and Global Logout: These features are already discussed in section 7.1.3.
- Opt-in Account Linkage or Account Federation and Federation termination Notification: It allows the user to link and unlink their accounts of different service providers to provide Simplified Single Sign On and Global Log Out among its accounts.
- Identity Provider Introduction: It refers to the ability by which a service discovers to which identity provider the user is related, to allow the service to ask for information of the user.

The second phase is the Web Services Framework (ID-WSF) which defines the infrastructure for creating, discovering and consuming identity services. This framework enables users to manage the sharing of their personal information among services in a secure environment.

The most important features of ID-WSF are:

- Permission-based attribute sharing: It allows the user to dictate what attributes can be released to different services.
- Interaction service: It allows an identity service to interact with a user to allow the identity service to share the user's data with other service.
- Identity services templates: These define how to query and modify data stored in identity services.

The last phase is called Identity Services Interface Specification (ID-SIS). This phase is on top of the ID-WSF and defines personal information services whose purpose is to address methods for identity-dependent services, such as an e-wallet service, calendar and geo-location.

Currently, specifications are based on SAML v1.1 but they are expected to be based on SAML v2.0.

There are several implementations of these specifications, from which it is worth mentioning the following:

- SourceID liberty: is an open source implementation Liberty Alliance whose current state is confined to the ID-FF specification.
- Sun Java System Access Manager: is a Sun implementation of Liberty Alliance. It supports all phases of Liberty Alliance. This is a commercial product from SUN.

#### *7.1.4.3. Shibboleth*

Shibboleth is an Internet2/MACE project whose goal is to develop an open, standards-SAML based solution in order for different entities to exchange information about their users in a secure fashion. Shibboleth's main goal is to enable the resource sharing among academic institutions, allowing students and professors at one university to access resources from a different one. Attributes in Shibboleth are the key for the access control decision to any specific resource. Shibboleth allows a user's attributes to be retrieved from the home site, and for these to be used in authorisation decisions at remote sites, thus providing a Single Sign On solution for accessing web services. A user defines which sets of attributes to share with any of the institutions or resources of the federation.

The Shibboleth open source implementation has built its federated identity environment on the base of an open Java and C++ implementation of SAML version 1.1 called OpenSAML. In addition many extensions have been done in order to provide extra features not supported in the core SAML standard such as anonymity or privacy protection. There are currently some formal federations that already use Shibboleth software such as SWITCH (from the Swiss Education and Research Network) or InCommon (from the Internet2 university members). The InQueue Federation, operated by Internet2, was designed for testing purposes for organizations that are becoming familiar with the Shibboleth software package and the federated trust model.

#### 7.1.4.4. PERMIS

The PERMIS system is the outcome of the EU project of the same name, or in full, Privilege and Role Management Infrastructure Standards Validation [159], which ended in 2002. It provides a framework for creating and managing a distributed authorisation infrastructure using X.509 standard attribute certificates (ACs) to hold users' roles. The Privilege Management Infrastructure (PMI) used by PERMIS defines a complete set of processes required to provide an authorisation service. It defines an authorisation API in Java, a SAML protocol interface, and a Policy XML DTD for constructing authorisation policies.

Further work on PERMIS is being carried out in other projects. The FAME-PERMIS project will design and develop middleware extensions to facilitate authentication strength linked to flexible, intelligent and fine-grained access control. It will support the use of a wide range of authentication methods to achieve specific authentication strength, or Level of Assurance (LoA). which is then fed into the authorisation decision engine, such as PERMIS, to decide the users' privilege rights. The SIPS project aims to seamless integrate PERMIS and Shibboleth. The DyCoM project is to contribute to enabling collaborations within dynamically evolving, scalable Virtual Organisations serving different project teams, by building on top of the GRASP Security Infrastructure, for distributed security enforcement, and PERMIS. These projects, all due to end in 2006, are introduced in [160].

The PERMIS software is freely available for research and education purposes as part of the US National Science Foundation's Middleware Initiative (NMI) software release, which also includes OpenSAML.

#### 7.1.4.5. WS-Federation

WS-Federation is a specification by IBM and Microsoft to provide identity federation and is discussed in section 6.3.1.2.

## 7.2. VO Management

Development of ideas on the Grid led to the idea of a Virtual Organisation (VO), which brings together entities from distinct, physical, conventional organisations and enables sharing and dynamic change. The services and characteristics of a VO are made available by the Grid Application Support Services Layer, but at lower layers makes use of technologies to manage resources and makes use of authentication and authorisation technologies to validate membership and access.

### 7.2.1. Definition of VO/MDVO

A *Virtual Organization, VO,* is a network of organizations and/or individuals, with a commonality of purpose or interest, which collectively make up an identifiable and coherent entity. In the beginning of the Grid technology, VOs generally consisted of supercomputing facilities with the aim of enhancing the computing power in order to perform very complex calculations in scientific environments. In the course of time, this situation has evolved, being possible today to

connect different equipment in real time, according to the necessities of the applications and the resources available. Besides, it is possible to reallocate and to replace resources, to accommodate changes in requirements or to adapt to new opportunities in the business environment, that is, make up a *Dynamic Virtual Organisation, DVO.*

The "dynamic nature" implies that the entire set up of a virtual organization may change in response to the market place. In this sense, virtual organizations of this type are temporary as to their ability to react quickly as regards the membership, the structure, the objectives, etc. Its vague/fluid boundaries and opportunism, as well as equity of partners and shared leadership mainly characterize a dynamic virtual organization.

One of the central aspects of Akogrimo is mobility, and here emerges the concept of *Mobile Dynamic Virtual Organisation, MDVO,* which is a Dynamic Virtual Organisation with at least one essential entity (in Akogrimo's case typically an Application User or more significantly a Provider) that is not bound to a location but can move, so that mobility aspects like context and personalization become important.

Then, *Virtual Organization Management* could be defined as the operation of maintaining the membership lists and properties of users, roles, organisations, resources and services belonging to a particular VO, their group memberships and roles in that VO, and a mapping of that information onto the local site authentication and authorization systems for the services and computing resources that the VO is entitled to use.

### 7.2.2. Lifecycle

There are two concepts which can be confused with each other: networked organization and virtual organization. The difference between a networked organization and a virtual organization is not in the structure of the organizational relationships, but in the behaviour of the organization. Network organizations tend to be stable and are resistant to change. The virtual organization, in contrast does show organizational processes to enable change. If change is the important aspect that distinguishes virtual from other organizations, a way of describing beginning, change and ending is needed.

A VO is created to run a specific set of tasks and may include multiple specially created Grid Services. A VO is created on the basis of a business agreement between participating organisations and individuals each of which contribute specific resources (computers, services, people, etc.). The agreement defines all resources and services available to VO members and conditions on which these resources and services are provided and used. A VO, like a real organisation, may contain all basic services required to run typical organisation, but these services "physically" and administratively may be run by member organisations on behalf of the VO.

There are several phases in the lifecycle of a VO (this uses the phases described in TrustCoM State of the Art report [158]).

- **Identification:** opportunity identification, evaluation and selection.
- **Formation:** partners identification, evaluation, selection and partnership formation.
- **Operation and Evolution:** Controlled integration of the services and resources offered by VO partners.
- **Dissolution:** when the VO is no longer required, the SLA and resources are released in order to become available again to be used by other VOs.

## 7.2.3. Evaluation of current technologies for VOs

The current technologies will be grouped according to their functionality within the VO Management. These provide facilities to manage resources, users, groups and services.

### *7.2.3.1. Technologies for Resource Identification*

One of the main distinguishing characteristics of the Grid application domain is the need to share resources (e.g., data, instruments, computers, humans, etc.) between organisations. For a resource to be shared it must be uniquely identified and, in many cases, to have associated metadata that describes it. Once the decision has been made to globally identify a resource, it is important that the resource is given a unique name with which it can be identified throughout the Grid application domain.

There is, as yet, no common standard for describing Grid resources. Different Grid middleware systems have had to create ad hoc methods of resource description and it is not yet known how well these can interoperate. So, there have been proposals for naming and uniformly providing access to resources, like the REpresentational State Transfer (REST) model. However, since REST depends on HTTP it is an specific protocol and hence unsuitable for heterogeneous systems like the Grid. Components in a REST system obey the following constraints:

- resources are identified through URIs
- resources manipulated through representations

Other possibility is the idea of URNs, which are globally unique, persistent, and accessible over the network. The Uniform Resource Name (URN) proposal is a technology-agnostic scheme for identifying resources. The Grid Resource specification utilises URNs as follows: a Grid Resource Identifier (GRI) is a URN that globally, uniquely, and everlastingly identifies a resource. A resource cannot have more than one GRI.

In addition to this, there is another kind of naming resources, LSID. A LSID conforms to the URN standards defined by the IETF. Every LSID consists of up to five parts: the Network Identifier (NID); the root DNS name of the issuing authority; the namespace chosen by the issuing authority; the object id unique in that namespace; and finally an optional revision id for storing versioning information. Each part is separated by a colon to make LSIDs easy to parse [190].

Both GRI and LSID are unique and everlasting, they can be stored in databases in the safe knowledge that they could be used, at any time in the future, to locate any services offering that resource.

Also, resources have associated Grid Resource Metadata (GRM). A GRM provides an extensible mechanism for the addition of application-specific metadata, and so this specification provides only a minimal set of metadata information elements. Often in the Grid it is desirable to know about the lifetime of a shared resource. Also, consumers wishing to access or operate on a resource may need to locate services that may be aware of this resource. For example, services may allow a data resource to be returned in a message or may provide the means to extract information from a resource (e.g. through a query). Consequently, the lifetime of a resource and the service endpoints, that may be aware of it, can be part of that resource's GRM document.

### 7.2.3.2. Technologies for Resource Allocation

Allocating resources is a significant problem for participants in the grid. The Grid architecture is a very distributed and rather dynamic collection of resources and services. Grid users, administrators, and the Grid services themselves need directories to keep track of these entities and to maintain relationships between them. These technologies are linked to resource identification technologies since it is necessary to know the information of a resource in order to allocate it. Particular applications selecting resources from a very large collection according to criteria such as connectivity, cost, security and reliability.

Co-allocation, scheduling, and brokering services allow VO participants to request the allocation of one or more resources for a specific purpose and the scheduling of tasks of the appropriate resources.

The HTTP-based Grid Resource Access and Management (GRAM) protocol is used for allocation of computational resources and for monitoring and control of computation on those resources. Ever since Globus Toolkit 2 appeared GRAM was one of its bases, but with Globus Tookit 3 and later versions, GRAM uses Web Services technologies to addresses the acquisition of resources (UDDI, SOAP, WSDL).

### 7.2.3.3. Technologies for Resource Registration

Grid Resource Registration Protocol (GRRP), an associated soft-state resource registration protocol, which is used to register resources with Grid Index Information Servers, GIIS. GRRP complements GRIP by defining a notification mechanism that one service component can use to "push" simple information about its existence to another element of the information services architecture. For example, GRRP is used by an information provider to notify a aggregate directory of its availability for indexing, or by an aggregate directory to invite an information provider to join a VO. It is a soft-state protocol, meaning in our context that state established at a remote location by a notification (e.g., an index entry for an information provider) may eventually be discarded unless refreshed by a stream of subsequent notifications.

Regarding user registration an example of registration at a Internet portal is shown [195] where the possible new user inserts his/her data and the information supplied will be forwarded to the VO administration and resource providers for validation before the registration process is completed.

The Virtual Organization Membership Service eXtension project (VOX) [196] is an example of this functionality. The Virtual Organization Membership Registration Service (VOMRS) is a major component of the VOX project. VOMRS is a service that provides the means for registering members of a VO, and coordinating this process among the various VO and grid administrators. It consists of a database to maintain user registration and institutional information, a server to handle members' notification and synchronization with various interfaces, Web Services and a web user interface for the input of data into the database and manipulation of that data. The VOX project also includes a component for Site AuthoriZation (SAZ), which allows security authorities at a site to control access to site resources and a component for the Local Resource Administration (LRAS), which associates the VO member with the local account and local resources on a grid cluster.

### 7.2.3.4. Technologies for Resource Discovery within VOs

Mechanisms are required that allow consumers and producers to discover each other and that subsequently allow information to flow between producers and consumers as Figure 20 - Grid Information Service architecture [194] depicts.

Directory services allow VO participants to discover the existence and/or properties of VO resources. A directory service may allow its users to query for resources by name and/or by attributes such as type, availability, or load. Directory services such as X.500 [191], LDAP [193], and UDDI [192] do not address explicitly the dynamic addition and deletion of information sources; in the case of X.500 and LDAP, there is also an assumption of a well-defined hierarchical organization, and current implementations tend not to support dynamic data well. Metadirectory services permit coupling multiple information sources. LDAP (Lightweight Directory Access Protocol) is based on X.500 directories, has a fairly flexible schema system, and is optimized for the sorts of access that membership and preferences needs (high read rate, low write rate).

As Figure 20 - Grid Information Service architectureshows, using the GRid Information protocol (GRIP), users can query aggregate directory services to discover relevant entities, and/or query information providers to obtain information about individual entities. Description services are normally hosted by a Grid entity directly, or by a front-end gateway serving the entity

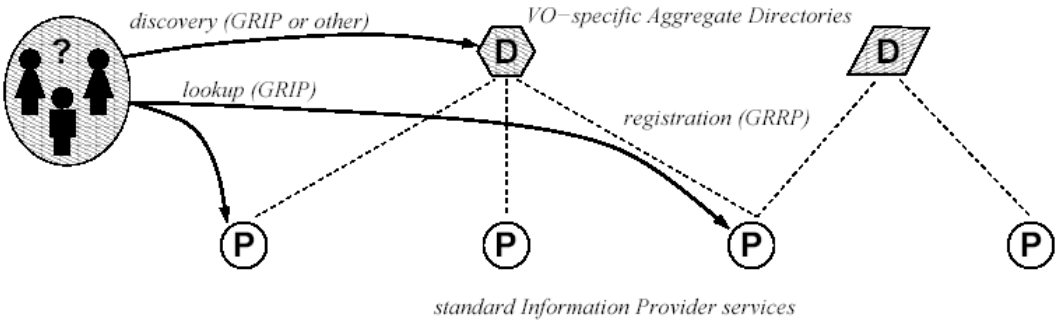


**Figure 20 - Grid Information Service architecture**

The Grid Information Service Architecture comprises two fundamental entities: highly distributed information providers and specialized aggregate directory services. An information provider is defined as a service that speaks two basic protocols. The GRid Information Protocol (GRIP) is used to access information about entities, while the GRid Registration Protocol (GRRP) is used to notify aggregate directory services of the availability of this information. We define an aggregate directory as a service that uses GRRP and GRIP to obtain information (from a set of information providers) about a set of entities, and then replies to queries concerning those entities.

A user, or more frequently an aggregate directory or other program, uses GRIP to obtain information from an information provider about the entity(s) on which the provider possesses information. It would be adequate to adopt the standard Lightweight Directory Access Protocol (LDAP) as the protocol for GRIP.

In Globus Toolkit there are two classes of Information Servers:

- Resource Description Services that supplies information about a specific resource (e.g. Globus 1.1.3 GRIS).
- Aggregate Directory Services that supplies collection of information which was gathered from multiple GRIS servers (e.g. Globus 1.1.3 GIIS) and offer customized naming and indexing.

#### 7.2.3.5. Technologies for User/Group Management

Within a single Virtual Organization (VO), thousands of users will potentially need access to hundreds of computing sites, and the traditional model where users register for local accounts at each site will present significant scaling problems. However, computing sites must maintain control over access to the site and site policies generally require individual local accounts for every user or VO member.

The management of a VO is done by assigning roles to registered users. Whenever users log in to a portal they obtain access to a restricted set of functionalities based on the role assigned to them by the VO administrator. This is done by first authenticating and then authorising users using their digital certificates.

First users have to register with the VO. Then once their request is approved by the VO manager, it is forwarded to the managers of the resources. Finally, users can view the resources they have been approved to access and accounting information about their resource usage. The resource manager's responsibility is to approve user's access to the resources they manage by assigning account names to them. He/She also needs to install clients for resource usage logging and gridmap file management on their resources. A VO Manager is responsible for enrolling users and resources (along with assigning managers to them) into the VO. He/She also allocates users to the resources and view the overall resource usage of the VO.

A VO member can have one or more roles. The role defines actions that a VO Member can perform within the VO. An example of some roles that can exist in a VO are:

- Visitor: A person who posses a valid certificate from the Certificate Authority approved by VO.
- Applicant: An experimenter who belongs to one of the VO institutions and possesses a certificate from one of the VO-approved Certificate Authorities. An applicant has submitted a VO registration form but has not yet been approved.
- Member: An applicant who has been approved. A member can submit jobs to the Grid. By default a member is assigned to an experiment wide group.
- VO administrator: A designated VO member who is in charge of registration and has access to all information collected by the VO. He is responsible for assigning administrative roles.
- Institutional VO representative: Vouches for the identity of an applicant.
  - Upon registration a member can select a representative from the list of known representatives. The selected representative does not necessarily belong to the member's institution.
- Grid site administrator:

- Assigns/revokes the role of System Administrator or Local Resource Provider to/from the VO members affiliated with the site
- Administers authorization of VO member to the site. The details are site specific and depends on regulations and policies of each particular site.

- Local resource provider: Administers authorization a member to use the grid resource (this could include addition of this member to the gridmapfile, mapping member to local account, etc)
- Group owner:
  - Creates groups and subgroups within the experiment.
  - Assigns/revokes group manager/owner role to a member of the VO.
  - A Group owner is a Group manager as well.
  - A Group owner owns the group if he owns any of ancestor group.
- Group managers: Assigns/removes members to/from the group he manages.

#### 7.2.3.6. Technologies for Authorization and Authentication within VOs

Enrolment into a VO is split into two phases: authentication and authorization. Authentication, proof of your identity, is encapsulated within a Globus compatible X.509 public key certificate. A Certification Authority is needed in the security module. Authorization to join a VO is down to the policy of that VO.

The Virtual Organization Membership Service (VOMS) was developed within the European DataGrid project [200] [205]. Its architecture uses the authentication and delegation mechanisms provided by the Globus Toolkit Grid Security Infrastructure (GSI). It addresses shortcomings with previous work, such as lack of flexibility and scalability.

Authorization and Access Control security service is a key part of the managed security in an open service oriented environment. Authorisation is typically associated with a service provide or resource owner, who control access to a resource based on provided by requestor credentials or attributes that define requestor's privileges or roles bound to requestor's identity. Separation of Authentication and Authorization services allows dynamic role based access control management and virtual association between interacting entities, and provides a basis for privacy in an open environment.

Authentication solutions for VO environments should have the following characteristics:

- Single Sign on: Users must be able to authenticate just once to access to multiple grid resources.
- Delegation: Users must be able to endow a program with the ability to run on his/her behalf.
- Integration with local security solutions: Interoperate with various local solutions.
- User-based trust relationships: Each of the resource providers must not interact with each other to configure security environment.

In Globus Toolkit 3 is introduced the Grid Resource Identity Mapper (GRIM), which provides a setuid program to generate proxy credentials for transient services. GRIM acts as a resource facility for transient services to obtain the Grid credentials of a resource.

Cardea was developed as a part of the NASA Information Power Grid [201]. It uses XACML to a policy for resource access control and SAML (7.1.4.1) to make assertions about a user or resource. XACML (eXtensible Access Control Markup Language) is an XML specification for defining access control policies in a wide variety over the Internet [202] and is being seeing as complementary to SAML.

The GRASP Security Infrastructure was developed within the European Project GRASP ([183]). In GRASP services and resources can be created on demand and shared between administrative domains. A security perimeter protects communities of users, services and resources across these domains. The GRASP Security Infrastructure [203] enables the dynamic formation and self-management of these perimeters.

The VOM Portal (VO Membership Portal [157] [204]) provides a secure web-based user registration functionality. Users register with the VO by using their grid certificate, embedded within a standard desktop web browser. This mechanism is used to prove their identity during all interactions with the portal. Once the user is registered, it is up to the administrator of that particular VO to approve or reject the user's registration request based on the information provided. Note, the VOM here (M for Membership) is not to be confused with VOM (M for Management) described earlier.

For resource usage authorization, it is useful to look at both the Community Authorization Service (CAS) being developed by the Globus project and the Virtual Organization Membership Service (VOMS).

More detail on Authentication and Authorization is provided in sections 6.3 and 7.4. In addition, more detail on security solutions for VOs can be found in [158].

## 7.3. Service discovery

Services need to be described, published and discovered to enable them to be composed with other services. This section focuses on discovery at multiple layers and the pros and cons of different technologies will be addressed.

### 7.3.1. Service Discovery at the Mobile Network layer

The ever growing number of computational resources and services connected to a network together with the rising use of mobile devices implies that the user of the mobile devices encounters different topologies and services depending on the network he is currently using. A change of network implies a reconfiguration in the mobile device in order to match the services it is going to use in the new network. To this end, mechanisms are needed in order to locate a particular network service or device amongst a large number of devices of accessible services and devices. These mechanisms are the Service Discovery protocols.

As an example, suppose that a user is using his laptop in network he does not know and wants to print a document in any nearby printer. The user issues a service request and the network responds informing of available printers and their network address and capabilities. The replies

are collected by the user's laptop and are showed to the user so that he can select the most convenient one according to his needs. This may be implemented in several ways, such as having the user send a broadcast message and receiving a response from every printer available, (which could overload the network in the case there are too many) or keeping a centralized register of the services in hierarchically disposed servers.

The Service Discovery Service (SDS), for example, provides a directory of services available in a network, much like yellow pages, where companies advertise their services and contacts and where customers, on the other side, search for providers that offer services with the whished characteristics. Compared to other known directory technologies, such as DNS (Domain Name Service), the SDS offers additional flexibility for the service provider to describe the nature of its service and the customers to query for the existence and location of services with specific capabilities.

In general, one needs mechanisms for determining available network interfaces, their QoS properties, tariffs, end-to-end QoS, etc. Discovery of end-to-end network transport services suitable for higher layer services ultimately depends on the physical location of the end-systems involved. That is, what are the properties of the different access and transport networks that can be used for network transport. Hence, after higher layer services that have specific QoS requirements have been discovered it will be necessary to initiate a negotiation of end-to-end network services. These issues are considered out-of-scope.

In the following, technologies for service discovery at the network layer are briefly evaluated.

#### *7.3.1.1. Discovery of basic network connectivity*

##### *DHCP*

The Dynamic Host Configuration Protocol (DHCP) [163] is a client-server protocol which provides its clients with network configuration parameters. DHCP is composed by two components: a mechanism for allocation of network addresses to hosts (clients) and a protocol for delivering configuration parameters from the DHCP server to its clients.

There are three different ways of allocating IP addresses:

- Automatic allocation – the client is assigned a permanent IP address
- Dynamic allocation – the client is assigned an IP address for a limited period of time
- Manual allocation – the client is assigned an IP address manually configured by the network administrator

##### *DNS*

The Domain Name System (DNS) [165] stores information about host names and domain names in a distributed database on the Internet. IP addresses aren't easily read or remembered by humans, so we use host and domain names when typing URLs or an email address. DNS translates host or domain names to IP addresses and vice-versa.

The DNS is highly scalable. Its design is decentralized and hierarchical; a domain has an authoritative DNS, it may have several sub domains, each with its own DNS server. On top of the hierarchy stand the root servers.

#### *IPv6 autoconfiguration and relationship with DHCP*

IPv6 enabled devices have the capability of automatically configuring themselves (stateless autoconfiguration). This process may require some configuration at routers, but no DHCP server is needed in the network. The IP address of a device is configured using a combination of their MAC address and information they received from routers. They can also use DHCP (stateful autoconfiguration) for receiving additional parameters after being assigned an automatic IPv6 address. That address is valid for a limited lifetime; when that lifetime is expired the address becomes invalid.

### 7.3.1.2. Discovery of network connection type

#### *SLP*

The Service Location Protocol (SLP) [164] allows network clients to discover available services on a network. Users need not know the name of a host which provides a particular service, they only need to know the type of service they want. In addition, SLP can also provide clients with information about the configuration of those services.

#### *WPAD <needs reference>*

The Web Proxy Auto Discover (WPAD) [166] protocol allows software (typically web clients) to automatically configure themselves to use a local proxy, instead of requiring manual intervention from users or network administrators. It does not define any new standard. Instead, it uses pre-existing discovery mechanisms.

WPAD allows the use of several Discovery Mechanisms: DHCP, SLP and DNS. A WPAD-enabled client first uses DHCP to find a proxy (or cache) server. If unsuccessful it uses SLP and if the cache server is still not found, it then searches DNS records for the proxy server.

### 7.3.1.3. Discovery services for reservation of capacity in access network

#### *CARD*

The Candidate Access Router Discovery (CARD) enables a mobile node to obtain information about candidate access routers for the mobile node's next handover. The information the mobile node obtains, which includes the IP addresses and capabilities of the routers, allows it to make decisions about the handover.

#### 7.3.1.4. Discovery of ad-hoc and hot-spot services

##### Hot-spot

In a wireless LAN, there is typically one or more Access Points (AP) which may be connected by wired Ethernet or using another wireless channel. The AP periodically sends beacons which are used for synchronization. Wireless stations can then join the network using the parameters found in the beacons sent by the AP. The wireless stations and AP form what is known as a Basic Service Set (BSS).

##### Ad-hoc

In an ad-hoc network, there are no Access Points, therefore there is no wireless infrastructure. Instead, the first ad-hoc station establishes an Independent Basic Service Set (IBSS) and starts sending beacons to maintain synchronization among the stations. Other ad-hoc stations may join the network using the IBSS parameters found in the beacon frame.

### 7.3.2. Discovery of devices, people and places

#### 7.3.2.1. Discovery of devices

##### Currently used discovery protocols

**Service Location Protocol (SLP),** set forth by Internet Engineering Task Force (IETF), uses multicast and limiting its scalability in larger networks. SLP only addresses service location discovery, leaving other aspects of service discovery to the application.

**Jini**, Sun's Java-based service discovery protocol requires Java Virtual Machine (JVM) in a client and that consumes processing power and memory. Jini uses Remote Method Invocation (RMI) to invoke a service and that consumes resources. The use of multicast makes it difficult for Jini to scale up for Internet.

**UPnP** (Universal Plug and Play), Microsoft's service discovery technology, is an architectural framework for self-configuring, self-describing devices. Its main objective is to support zero-configuration networking, and automatic discovery for UPnP enabled devices. A device can join a network, obtain an IP address, announce its own capabilities and detect the presence of others devices in the network, as well as their capabilities. It also allows a device to leave the network automatically without leaving unwanted state behind. It was designed to enable multiple vendor inter-operation. There are a number of vendors that sell routers (usually with a built-in firewall) with UPnP support, for home or small offices. In this particular case, UPnP provides methods for e.g. learning public IP addresses, enumerating existing port mappings and adding or removing port mappings. UPnP uses Simple Service Discovery Protocol (SSDP) and existing Internet protocols, such as IP, TCP, UDP, HTTP and XML for its operation and depends heavily on multicast. These characteristics make UPnP inefficient or sometimes unsuitable for smaller devices and low-bandwidth networks.

The **Salutation Consortium** consists of several peripheral device manufactures such as HP, IBM and Cannon. The Salutation Manager Protocol (SMP) is used for Salutation Manager

communication between other Managers, clients and servers. It is based on Remote Procedure Call (RPC), which is so resource demanding that it does not scale well to small devices and low bandwidth networks used in mobile pervasive environments. Salutation Lite is reduced implementation of Salutation.

**Bluetooth SDP** is service discovery protocol found in Bluetooth protocol stack. It performs only service discovery and it is limited to single hop networks only.

Table 5 (below) summarises the properties of some protocols or methods of organizing and locating resources (such as printers, disk drives, databases, e-mail directories, and schedulers) in a network.

| **Feature** | **SLP** | **Jini** | **UPnP** | **Salutation** |
|---|---|---|---|---|
| Directory | Yes | Yes | Yes | Yes |
| Service utilization | Yes | No | Yes | Yes |
| Service description | Service templates with attributes | Attribute/value pairs | XML based | Name value pairs of functional units |
| OS independence | Yes | Yes | Yes | Yes |
| Transport protocol independence | No | No | No | Yes |
| Programming language independence | Yes | No | Yes | Yes |
| Scalability | Enterprise | Enterprise | Small networks | Enterprise |
| Security | IP dependent | Java based | IP based | Authentication |
| Service property filtering | Yes | Yes | Yes | No |
| Prerequisites | TCP/IP | Java VM and Java, RMI, IP multicasting | TCP/IP RPC, HTTP, Unicat, multicast | RPC |
| Implementation example | Open SLP Novel Netware | Macromedia's application server – Jrun | UpnP Digital media | IBMs NuOffice |
| Developer | IETF | Sun Microsystems | Microsoft | Salutation |

**Table 5 - Service discovery protocols**

Technical challenges for these current service discovery protocols are devices with limited computer power, restricted memory, varying network bandwidth and time to time loss of traffic. Another difficulty is that they don't scale up to larger mobile and wireless networks in terms of number of devices and network size in hops. They are more suitable for environments, where there are low mobility of client devices and stationary devices serving these clients. Access points are used to connect client devices to the network, where devices are homogeneous consisting mostly of portable personal computers, portable desktop assistants (PDAs).

#### 7.3.2.2. Discovery of real world objects- people, things and places <needs some words on places>

Discovering other real world objects has been an outcome of projects such as Sentient Computing [168], which uses sensors and resource status data to maintain a model of the world, and Cooltown [167], which regards people, places and things to be part of the discoverable, connected world.

A person can be discovered by the following approaches:

- Represented by a device where she/he is logged on to – user profile etc.
- Using a tag, active badge, or RFID that allows a person (or other physical object) to be discovered. The physical object has virtual counterparts (e.g. Web Service) that represent the physical object.

### 7.3.3. Discovery of services and knowledge in a Grid environment

#### 7.3.3.1. Introduction

As asserted by the OGSA specifications, "an information service might allow producers and consumers to discover each other by making detailed descriptions about themselves available for querying. A special distributed registry or point-to-point scenario could be used for that purpose. The descriptions could include, for example, the type of producer or consumer, what information they produce or consume, and their endpoint URLs."

Of course, also in the Akogrimo vision, at the grid application layer, discovery and composition of dynamic services are required capabilities.

In the simplest scenario we may have a user that needs to locate a service description that meets some desired functional and other criteria. A possible solution for this is to query a central directory service where service descriptions are published, and receives an answer. The directory owner decides who can publish information into the directory and who is authorized to query it. UDDI is an example legacy directory service. OGSA directory services could be based on WSRF service group concepts.

OGSA defines a set of functions for a discovery mechanism:

- Define a naming scheme
- Discovery capabilities themselves

Some of these functions are now briefly described as they are intended by OGSA documents. These considerations are at the base of the protocol presentation and comparison of the chapter 2 of this draft document.

### 7.3.3.2. Naming scheme

Traditional distributed systems usually support a two- or three-layer naming scheme. In OGSA, the naming service, ***OGSA-naming***, uses a three-level convention. Every named OGSA entity is associated with: an optional ***human-oriented name***, an ***abstract name***, and an ***address***.

The human-oriented name is usually human-readable and may belong to a name space. Name spaces are usually hierarchic and usually have syntactic restrictions. Hierarchic name spaces permit each part of a name to map to a particular context. For example, in the Unix file "`node1:/var/log/error`," the context for `var` is the root directory of a machine named "`node1`," the context for "`log`" directory is "`var`," and the context for "`error`" file is "`log`." OGSA-naming does not require human-oriented names to be unique. Many different naming schemes exist and this document does not attempt to prescribe the set of all supported schemes.

The abstract name is a persistent name that does not specify a particular location. The abstract name may not be globally unique. A WSRF endpoint reference with renewable references and the Legion Object Identifier are examples of abstract names. A mechanism, outside the scope of this document, is required to map human-oriented names to abstract names.

The address is a concrete name that contains the location of an entity. The combination of the endpoint address and reference properties in a WSRF endpoint reference, a memory address, and an IP address/port pair are examples of addresses. A mechanism, outside the scope of this document, is required to bind abstract names to addresses.

In OGSA the existence of a resource handle is assumed. This is an abstract name of a resource and its associated state (if any).

### 7.3.3.3. Discovery

OGSA defines the discovery for services and resources. A ***directory*** (or registry) is an obvious solution, but not the only one. A directory is distinguished from other possible solutions in that it has persistent storage for the "latest" information and is optimized for searches. Low latency response to a high volume of queries is required. The directory may be replicated for scalability.

Alternatively, a compilation or guide of information can be stored in an ***index*** (such as Google). Unlike a registry, which tends to be centrally controlled, anyone can create an index.

Another alternative is ***peer-to-peer discovery***, where a Web Service is a node in a network of peers and dynamically queries its neighbours in search of a suitable match. The query propagates through the network from one node to another until a match is found, a particular hop count is reached, or some other termination criterion is satisfied. Yet another alternative for a discovery service is a general GMA-based service, that is a discovery service based on the Grid Monitoring Architecture capabilities stated by OGSA [162].

Furthermore, a general OGSA service that provides a ***combination of the above capabilities*** can provide more flexibility to the end user. For Grid resources in general (including services and

applications), the amount of available information about resources could be large, dispersed across the network, and updated frequently. Searches in this space may have unacceptable latencies. In order to manage such information in a controllable way, it is important to separate information source discovery from information delivery. Searches should only be used to locate information sources or sinks. A special-purpose directory is used to hold metadata about the resources. In this case the directory must cope with high rates of updates that are expected in the dynamic OGSA environment. Individual producer/consumer pairs can limit the amount of data flowing between them to that satisfying the consumer query. This model differs from a message broker that combines the mechanisms for finding sources and sinks of information and its delivery into a single searchable channel. The merits of this approach are described in the already mentioned OGSA GMA document [162].

A user of such a general service should be able to put any information, irrespective of its intended use (e.g., discovery, monitoring), into it without needing to understand the complexity of the system. He first must specify what information is to be made available in OGSA, where the type and structure of that information should be well-defined. The user might also wish to specify certain properties and policies. This could include how the information is held, retention period, guarantee of delivery, persistency, and access control. A consumer could filter information of interest using a subscription topic, for example, and it could support a query to further refine the events delivered to it through predicates defined in the query expression.

More advanced users may require a deeper understanding of the internal workings of the service. Following a request for information in a general (GMA-based) service, expected behaviour is that a "mediator" capability is used to perform a registry/schema lookup and locate suitable sources of information. For long-term queries the mediator ensures that, as sources are dropped or new relevant sources come online, the subscribed consumers are updated. Mediation is also concerned with planning the distributed queries to the relevant producers, as well as merging the results.

#### 7.3.3.4. Standards and solutions

In this chapter some specific technologies emerging to face services discovery and coordination are described and analysed. The aim of this work is to trace a state of the art in terms of discovery protocols and standards that can be a starting point for the choices needed by the Akogrimo architecture.

Major attention is given to the Web Services related protocols, such as UDDI, WS-Discovery and the Web Services Resource Framework set of standards (WSRF).

##### *UDDI*

The Universal Description, Discovery, and Integration (UDDI) protocol is a central element of the group of related standards that comprise the Web Services stack. The UDDI specification defines a standard method for publishing and discovering the network-based software components of a service-oriented architecture (SOA). Its development is led by the OASIS consortium of enterprise software vendors and customers.

A UDDI registry's functional purpose is the representation of data and metadata about Web Services. A registry, either for use on a public network or within an organization's internal infrastructure, offers a standards-based mechanism to classify, catalog, and manage Web Services,

so that they can be discovered and consumed by other applications. As part of a generalized strategy of indirection among services based applications, UDDI offers several benefits to IT managers at both design-time and run-time, including increasing code re-use and improving infrastructure management by:

Publishing information about Web Services and categorization rules specific to an organization;

Finding Web Services (within an organization or across organizational boundaries) that meet given criteria;

Determining the security and transport protocols supported by a given Web Service and the parameters necessary to invoke the service;

Providing a means to insulate applications (and providing fail-over and intelligent routing) from failures or changes in invoked services.

| | |
|---|---|
| Naming schema type: | Well organized, hierarchic tree of names, both human readable addresses and URLs;<br>Service descriptions and binding with technical models to be used for services invocation and data exchange;<br>Services are usually grouped by organizations that provide them, facilitating cooperation. |
| Supported Discovery Mechanism: | Directory Registry with a specific API for queries and updates. |
| Pros: | Stable and widely used;<br>UDDI version 3.0 supports Registries affiliation (this improves scalability and cooperation between different organizations to selectively coordinate their service oriented applications) |
| Cons: | UDDI is basically a directory services, without peer to peer communication or multicast discovery mechanism.<br>Mobility is not supported. |

**Table 6 - UDDI main features**

### *ebXML registry service <what's a CPP? – term used below>*

The discovery of a business that provides a desired Web Service (and the downloading of its Collaboration Protocol Profile (CPP) can be done via the ebXML Registry Service [169]. However, one may be confused whether to use UDDI or the ebXML Registry Service to look for a Web Service. UDDI is used to publish and discover Web Services; an ebXML Registry Service, on the other hand, not only publishes and discovers Web Services, but also provides information about, for instance, business processes, business documents and business profiles. However, both systems can complement each other; organizations can continue to use UDDI to inquire about businesses in the global UDDI Registry. Those entries can then be used in referring to ebXML Web Services in the ebXML Registry.

### *Semantic Web <needs checking>*

Web Services publish statements describing their intended or normative behaviour. These statements should be given common, machine processable, extensible semantics that support judgment of:

- Whether a service can perform a given task;
- The relative ranking of a set of services with respect to basic QoS criteria.
- And then, using reasoning, to match service descriptions against requirements

We need to use a language for semantics such as RDF(S) or OWL (Web Ontology Language) and a language for Web Service description such as the DARPA Agent Markup Language (DAML-S), OWL-S, or Web Service Modeling Ontology (WSMO). OWL-S is a W3C specification (currently a W3C Member Submission) [170],[188] which enables the use of the Semantic Web to describe, advertise and discover Web Services. It is based on OWL and supplies a core set of markup language constructs for describing the properties and capabilities of Web Services in unambiguous, computer-intepretable form. OWL-S markup of Web Services will facilitate fuller automation of Web Service tasks, such as Web Service discovery, execution, composition and interoperation.

We then need to consider publication and discovery standards: Semantic UDDI Registry. OWL-S does not dictate any form of registry for services. This also applies to WSMO, where no explicit registry architecture for any of its elements is defined. If UDDI is used for discovery, the currently most likely option seems to be extending or embedding OWL-S or WSMO in UDDI, such as suggested in [185] and [186].

Related initiatives:

- WSDF: Web Services Discovery Framework
- Web Services Mediation Framework

### *WS-Discovery*

WS-Discovery is a multicast discovery protocol to locate services. The primary mode of discovery is a client searching for one or more target services. To find a target service by its type, a scope in which the target service resides, or both, a client sends a probe message to a multicast group; target services that match the probe send a response directly to the client. To locate a target service by name, a client sends a resolution request message to the same multicast group, and again, the target service that matches sends a response directly to the client.

To minimize the need for polling, when a target service joins the network, it sends an announcement message to the same multicast group. By listening to this multicast group, clients can detect newly-available target services without repeated probing.

To scale to a large number of endpoints, this specification defines multicast suppression behavior if a discovery proxy is available on the network. Specifically, when a discovery proxy detects a probe or resolution request sent by multicast, the discovery proxy sends an announcement for itself. By listening for these announcements, clients detect discovery proxies and switch to use a

discovery proxy-specific protocol. However, if a discovery proxy is unresponsive, clients revert to use the protocol described herein.

To support networks with explicit network management services like DHCP, DNS, domain controllers, directories, etc., this specification acknowledges that clients and/or target services may be configured to behave differently than defined herein. For example, another specification may define a well-known DHCP record containing the address of a discovery proxy, and compliance with that specification may require endpoints to send messages to this discovery proxy rather than to a multicast group. While the specific means of such configuration is beyond the scope of this specification, it is expected that any such configuration would allow clients and/or target services to migrate smoothly between carefully-managed and ad hoc networks.

| | |
|---|---|
| Naming scheme type: | URL, endpoints, human readable names and attributes; |
| Supported Discovery Mechanism: | Multicast discovery protocol;<br>Network announcement support. Services that dynamically join a network can announce their presence, reducing network traffic and improving overall performance;<br>Scalability is assured by a discovery proxy mechanism. If a discovery proxy services is located, multicast protocol is suppressed. This capability allows scalability over Internet and wide range networks. |
| Pros: | Scalability<br>Multicast discovery, proxy discovery and announcements support.<br>Interoperability with other rich network services (DHCP, DNS, Active directory domains…)<br>Multicast mechanism can be used to support mobility of users and services. |
| Cons: | It's still a young protocol;<br>Few implementations available. |

**Table 7 - WS-Discovery main features**

### *WSRF Discovery support and specification*

WSRF specification includes a suite of modular protocols and standards:

| | |
|---|---|
| WS-ResourceLifetime | Mechanisms for WS-Resource destruction, including message exchanges that allow a requestor to destroy a WS-Resource, either immediately or by using a time-based scheduled resource termination mechanism. |
| WS-ResourceProperties | Definition of a WS-Resource, and mechanisms for retrieving, |

| | |
|---|---|
| | changing, and deleting WS-Resource properties. |
| WSRenewableReferences | A conventional decoration of a WS-Addressing endpoint reference with policy information needed to retrieve an updated version of an endpoint reference when it becomes invalid. |
| WS-ServiceGroup | An interface to heterogeneous by-reference collections of Web Services. |
| WS-BaseFaults | A base fault XML type for use when returning faults in a Web Services message exchange. |

In the field of services discovery, the WS-ServiceGroup specification must define a means of representing and managing heterogeneous by-reference collections of Web Services. This specification can be used to organize collections of WS-Resources, for example to build registries, or to build services that can perform collective operations on a collection of WSResources.

The ServiceGroup specification can express ServiceGroup membership rules, membership constraints, and classifications using the resource property model from WS-ResourceProperties. Groups can be defined as a collection of members that meet the constraints of the group as expressed through resource properties. The ServiceGroup specification should also define interfaces for managing the membership of a ServiceGroup.

The interfaces defined by WS-ServiceGroup are expected to be composed with other Web Services interfaces, which define more specialized interaction with the service group and/or with the services that are members of the ServiceGroup. For example, specialized interfaces may other means of querying the contents of the ServiceGroup, and for performing collective operations across members of the ServiceGroup.

This WS-ServiceGroup specification defines a means by which Web Services and WS-Resources can be aggregated or grouped together for a domain specific purpose. In order for requestors to form meaningful queries against the contents of the ServiceGroup, membership in the group must be constrained in some fashion. The constraints for membership are expressed by intension using a classification mechanism. Further, the members of each intension must share a common set of information over which queries can be expressed.

In this specification, the ServiceGroup membership rules, membership constraints and classifications are expressed using the resource property model of the WS-ResourceProperties specifications. Groups are defined as a collection of members that meet the constraints of the group. The ServiceGroupRegistration interface extends the basic ServiceGroup capabilities with message exchanges for managing the membership of a ServiceGroup.

The ServiceGroup and ServiceGroupRegistration interfaces defined in this document are commonly expected to be composed with other application domain specific interfaces, which define more specialized interaction with the service group and/or with the services that are members of the service group. For example, specialized interfaces may offer means of querying the contents of the ServiceGroup, and for performing collective operations across members of the ServiceGroup.

WS-ServiceGroup is inspired by a portion of the Global Grid Forum's "Open Grid Services Infrastructure (OGSI) Version 1.0"

This specification intends to satisfy the following requirements:

- Define the standard resource properties by which a requestor can query and retrieve contents of a service group.
- Define the standard resource properties by which a requestor can query and retrieve details of an entry in the service group.
- Define standard message exchanges and resource properties by which a requestor can add new entries for a member in a service group.

The main features of WS-ServiceGroup are shown below:

| | |
|---|---|
| Naming scheme type: | URL, endpoints, human readable names and attributes;<br>Uses WS-ResourceProperties specification. |
| Supported Discovery Mechanism: | It only allows directory creation with grouped services references |
| Pros: | It's integrated with other WSRF standards and protocols;<br>Consider both Web Services and resources as suggested by the OGSA specification for grid services. |
| Cons: | No multicast and discovery protocols support<br>Can be used to create services directory, but it isn't a directory standard |

**Table 8 - WS-Service Group main features**

### 7.3.4. Discover context information <needs more>

Context includes location, geography, network, devices and is linked with personalisation. More detail about context, including discovery, is in section 4.4.

## 7.4. Authentication and Authorisation and Accounting (AAA)

Here we focus on the authentication and authorisation aspects of AAA and defer discussion of accounting to section 7.5. For brevity, we will nonetheless use the term AAA.

The nomadic capabilities and mobility support that Akogrimo middleware is to realize are a brand new challenge for AAA. The identity management, for instance, and the related authentication mechanisms must support users' mobility (i.e. dynamic switching of a user from a network domain to another) and the possibility of a user to access the Grid (the MDVO he belongs to) from different devices.

### 7.4.1. Goals

The goals AAA work should reach include:

- **Dynamic perimeter Security checking and enforcement**: authentication and authorization should continue to work properly even if users or services are temporary disconnected (because of local network limitations, or during the transfer from a network domain to another). The "identities" must be recognized even if the connection point to the network change from time to time so as from a particular device to another. Also in this case it is necessary to allow multiple authorization frameworks to work independently (within administration domains) and together (to manage Virtual Organizations). Commonly known authorization systems that may be used in each different network domain and many of them are described in the related sections across this document. For example SAML (Security Assertion Markup Language – see 7.1.4.1) and XACML (eXtensible Access Control Markup Language) should be taken in consideration as standard formats to communicate policy assertions. Besides, **WS-Secure Conversation** protocol may be useful to exchange authorization policies statement across domains and different AAA solutions to realize services grouping management.

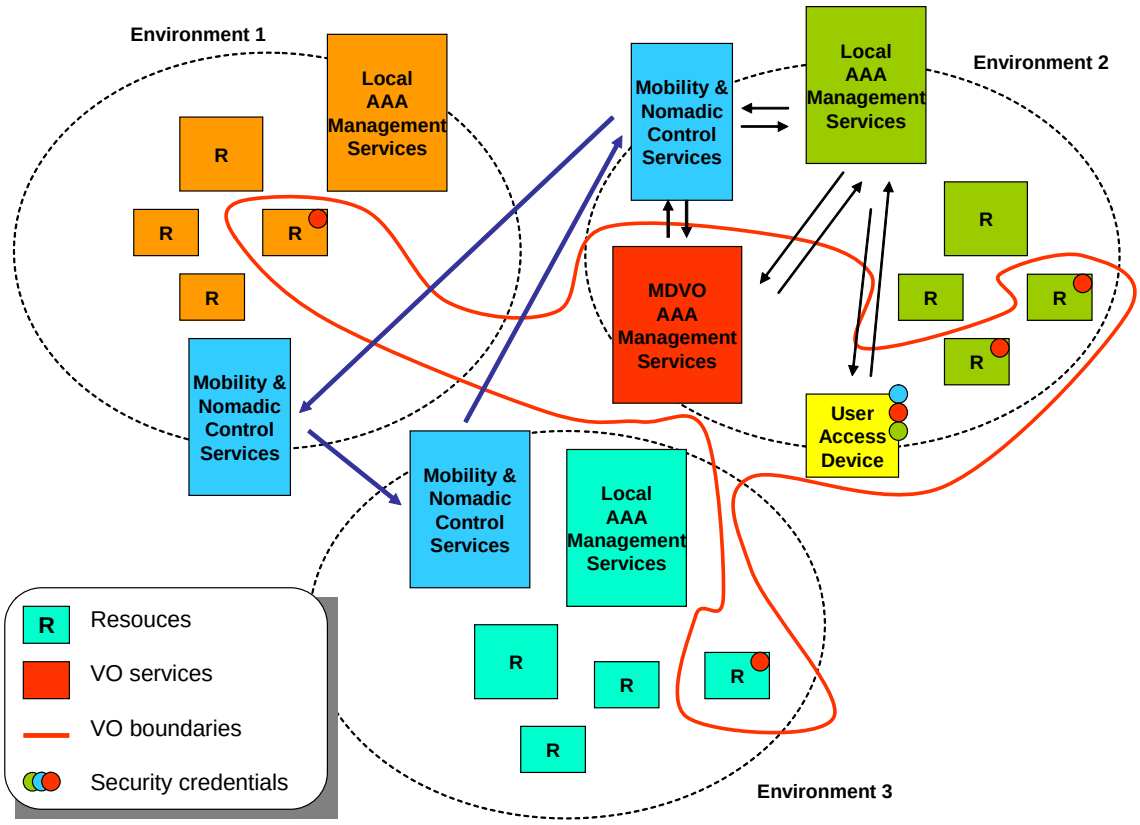


**Figure 21 – Akogrimo security architecture, roles and responsibilities**

### 7.4.2. Security architecture, roles and responsibilities

In the figure above, there is depicted a possible subdivision of roles and responsibilities form a security point of view:

- Each network domain should have local security management services to discovery, locate and authorize local resources and users interactions. These local services probably are strictly based on network protocols and related AAA mechanisms. Here also "local" has a time related meaning: local security management services must control every resource or user that is in the respective network at any given time.
- At the application layer each Virtual Organization has its own security management services (registries, policy enforcement points…) These VO services can interoperate with the locals ones to share and verify rights, permission, policy assertions, identity credentials and so on…
- In each network domain there are also some mobility and nomadic control services that realize the information exchange between network domains and VO services to manage users and resources migration from an environment to another. These services should allow the identity recognition (and registration) of the same user even if connected from a "foreign" network.

### 7.4.3. AAA server and client at the network layer

From the network layer viewpoint, a traditional AAA Architecture (AAAArch) has two main entities: an AAA Server and an AAA Client (cf. [206]). The AAA Server maintains a set of internal databases and respective records about user identities, service policies, service descriptions. The AAA Server performs all domain-relevant AAA tasks.

The AAA Client is usually a (mobile) device (however, it could also be an application residing on a device) that requests Authentication, Authorization, and Accounting services from the AAA Server.

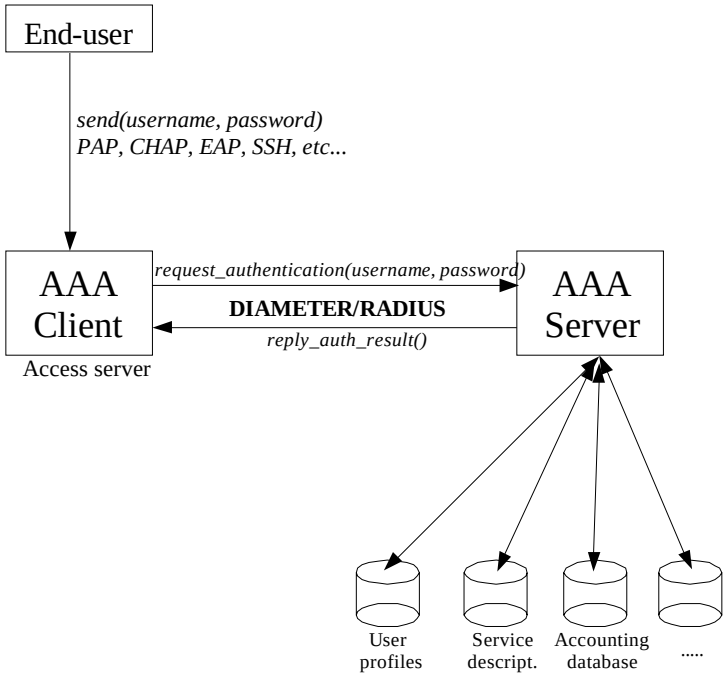


**Figure 22 - AAA Basic Architecture**

Note that every AAA Client has a home domain, he is associated with, and every AAA Server determines a domain. Multiple physical AAA Servers may exist within a single domain for performance reasons. As a prerequisite of an application of the AAAArch, identities for devices, applications, or users are required. However, this type of identity model is not inherently part of AAAArch.

- **Authentication**: let us assume a simple example of a user who wants to access the Internet from his home using a dial-up connection to his ISP (Internet Service Provider). For using the service provided by the ISP, the user has to authenticate first, so he will send his authentication information (*e.g.*, username/password) to the Access Server he is connected to. The Access Server itself might not have the authority of authenticating users, but it knows the domain-internal address of the AAA Server being responsible for this authentication task in his network. The request is forwarded to the AAA Server selected, which is based on the user records, and it decides if the authentication is granted or not. In case the user asking for access is not part of this AAA Server's domain, AAA Servers do communicate with AAA Servers in other domains to achieve a reply. The final authentication decision is transmitted to the Access Server who will allow or deny the user to use its services, based on this decision.

- **Authorization**: In a second step let us suppose that the user was allowed to use the access server as its Internet Gateway and he wants to start an RTSP (Real-time Streaming Protocol) session with an audio streaming server located in his ISP's network. Based on company's policies, not all users are allowed to use this server. When the streaming server receives a request for a connection from the user, it will request an authorization from his AAA Server. This requires a message to be sent, which will contain at least the User ID (UID) and the Service ID (SID) requested for by the user. Of course, this service was previously registered with the AAA Server, and certain policies were established by the service provider. The AAA Server will check its local database of service policies and based on its information stored previously it will grant or deny the user's access to the

requested for service. In this case, the streaming server plays the role of the AAA Client and it requires the authorization service from the AAA Server.

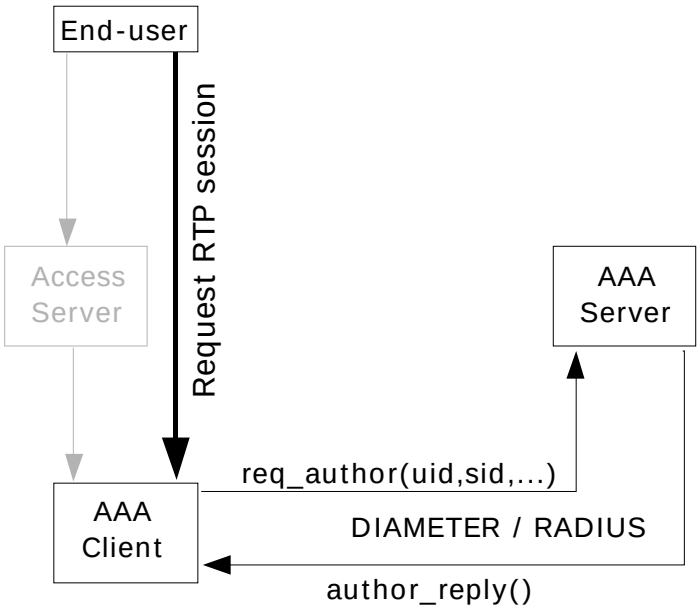


**Figure 23 - Authorization Process**

### 7.4.4. AAA for a mobile node

In the case of mobility, a mobile node (MN) that belongs to a "Home Network" (HN) (its identification information is stored in the AAA Server of the HN) may want to access services in a "Foreign Network" (FN). Let us assume a simple scenario where a user wants to use an Access Point (AP) in a foreign network. The AP in this scenario will play the role of the AAA Client. It will request an authentication and authorization service for the MN from its AAA Server.

The MN tries to register with the AP using its *username@realm* information. The AAA Client sends this request to the AAA Server for Authentication. The AAA Server in the FN recognizes that the user belongs to the other *realm* and forwards this request to the AAA Server of the Mobile Node's Home Network. This operation requires previously established agreements between the FN and HN on roaming policies. The AAA Server performs the requested for operations and replies its answer to the AAA Server of the FN, who will forward it to the AAA Client (AP in the case discussed). Authorization and accounting messages have to follow the same path.

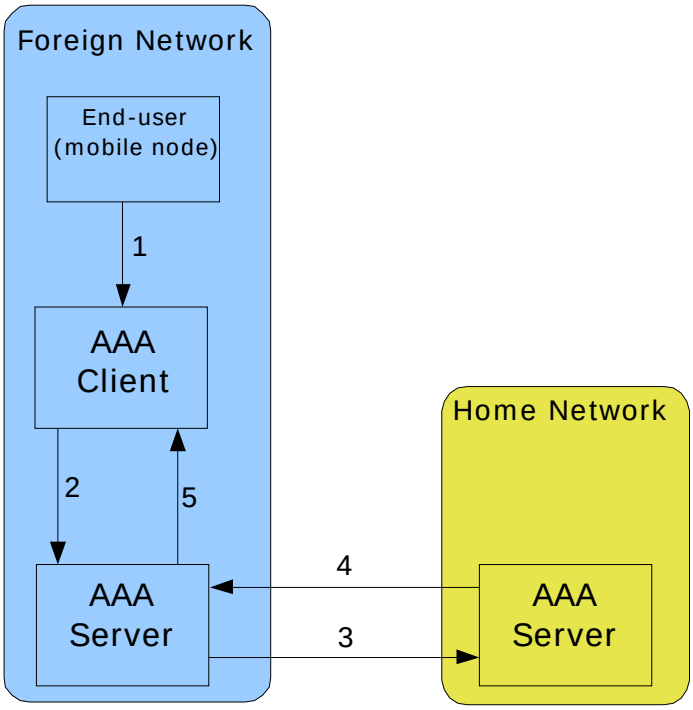


**Figure 24 - AAA Messages in a Roaming Scenario**

Of course the above scenario will be specified during the design activities but in the following we can list some different technologies (across different layers that could be used in order to establish the above).

## 7.4.5. Technology list

Akogrimo infrastructure should implement an *open grid*, relying on the Internet. Application level communication protocols adopted by the project must guarantee integrity of the exchanged messages and reliability of each transmission. The heterogeneous nature of the network layer required by Akogrimo (classical wired networks, wireless network, GSM, GPRS or UMTS cells…) implies, even if security features at lower layer of the stack can be used, a general management of the AAA issues at the application layer.

Of course, users and services must be identified and authenticated in order to access grid resources with the proper rights. Each VO should implement application layer authentication services to achieve this goal. These last services should be able to cooperate between them and with security mechanisms used by each organizational domain participating in the VO. Even if the Akogrimo framework leaves free every domain administrator to choose the authentication and authorization schemas he consider best, standard protocols for the realization of ad hoc Public Key Infrastructures should be preferred. X.509 certificate and their attached attributes may be suitable for identity management and right assignment to users and services.

### *7.4.5.1. Web Service-related security technologies*

At the application layer we expect mainly to face and solve the security issues above the SOAP Messages layer and then we'll leverage on Web Service related security technologies, such as:

- X.509 certificates are used to identify services belonging to the various groups (secure perimeters);
- the WS-Security extension to the SOAP protocol could used to sign and add security tokens to the exchanged soap messages between services. Several WS-Security implementation are available on different platforms (e.g. Microsoft WSE 2.0,…)
- Custom security token could be used in order to allow group creation and management and could Akogrimo custom information if required.
- WS-Security to assure authenticity, integrity, and confidentiality of SOAP messages;
- WS-Trust (over WS-Security) to realize trusted relationship between services and users within the same VO.

More details on WS security specifications can be found in section 6.3.1.

#### *7.4.5.2. Network layer protocol technologies for AAA*

These Web Service capabilities must be combined with the (local) network layer protocol used to authenticate and authorize users to access resources and services. Some relevant technologies are described in other parts of this document:

- Radius and Diameter in section 4.2.4
- SNMP in section 4.1.3
- SIP in section 4.1.2

## 7.5. Accounting, charging and pricing

### 7.5.1. Overview

The aim of this section is to provide a state of the art description for Grid accounting systems. It should be underlined that until recently, Grid systems aimed to provide the means to the scientific community for their demands in high performance and distributed computing capabilities [155]. There was no need for having commercial orientation in the development of the Grid middleware. Today, we can see the migration towards the exploitation of the capabilities of a Grid infrastructure in a commercial manner in virtual marketplaces. For this reason an accounting system incorporated in Grid environments forming Virtual Organizations seems to be mandatory. Grid accounting technology is not mature enough yet to provide a variety of widely accepted Grid accounting systems. However so as to make available the most widely accepted systems, there is an effort from the Global Grid Forum to provide the means for the concrete definition and standardization of these activities. Additionally, there are already some projects that have handled the specific issue for their own needs, providing fruitful results and contribution to the scientific community.

In what follows we will provide a general discussion on Grid Accounting and describe some existing work on it.

The accounting procedures described in this section include entries for low-level network resources, which are themselves described in more detail in section 4.3.

#### 7.5.1.1. Requirements

In any Grid environment, resource providers (i.e. resources) and resource consumers (i.e. users) are the main entities interacting with each other. Users request the services provided by resources. This makes users and resources a good starting point for making a list of the minimal functionality required from the accounting system [156] [161].

From a resource's perspective, the accounting system needs some means to:

- Provide cost information to users
- Grant or deny users access based on account balance (i.e. authorization is based on someone's ability to pay)
- Get a guarantee that user funds will be available to pay for resource usage
- Track resource usage
- Charge the user for the resource usage
- Get resource usage information for each submitted job (e.g. for evaluation purposes)

From a user's perspective (or some entity working on behalf of the user), the accounting system needs some means to:

- Specify which VO/project should be charged for resource usage
- Get the cost associated with using a resource
- Get account balance
- Get information on completed transactions
- Get usage information on completed jobs

#### 7.5.1.2. Properties

The accounting system possesses several important properties, like scalability, user transparency, consistency, fault tolerance, trust and security.

#### 7.5.1.3. Design Issues

There are several ways of realizing an accounting system. The proposed design can be referred to as an online bank approach. In this approach, a user is granted/denied access to resources based on the amount of available funds on his/her account. If a user submits a job without having sufficient funds available, the accounting system may not allow the job to run. Other solutions, with less restrictive consistency demands, are also conceivable. An offline bank (where accounting information is cached locally at resources and then periodically synchronized with a bank) and a gather/scatter approach (where allocations are spread across resources and resource consumption information is gathered from all resources when needed) would offer alternative solutions.

#### 7.5.1.4. Types of Resource Usage

What types of resource usage should the user be charged for? This, of course, depends on what resource usage information can be acquired, as well as the needs of the resource provider. All resource providers will not charge their users for all resource types. The following is a list of resource types that might be of interest:

- CPU-time
- Wall-clock time
- Number of processors
- Storage
- Bandwidth
- Data transfer(s)
- Memory
- Quality of Service (e.g. higher task priority)
- Software usage

Schemes for charging and pricing usage of mobile communications are discussed in details in section 4.3

#### 7.5.1.5. The currency

An interesting issue is what "currency" should be used to charge a user. At least two alternatives exist:

- Charge for usage of each different resource
- Convert different types of resource usage into a single currency unit (such as grid credits). This approach generally requires some function to map a set of resource usage records into a single currency.

#### 7.5.1.6. Resource Valuation

Should resources somehow be valued (e.g. by performance)? One could argue that it should be less expensive to run a 2 hour job on a Pentium III than on a 486. Furthermore, in the case of all resource usage being converted into a single currency, the different resources need to be given appropriate weights. A more mature accounting system would allow resource providers to set cost weights on their resources and make those prices available to users, creating a kind of "grid market place".

#### 7.5.1.7. Job-cost Information

If different resources are valued differently, a user must be able to obtain correct cost information in order to make an informed decision on the choice of resources. Some potential solutions are:

- Use static prices throughout the system.
- Query the resource.
- Make cost information available through some information system (such as MDS).

#### 7.5.1.8. When to Charge the User

When should the user be charged for the resource usage? At least three alternatives exist:

- pre-payment: pay before usage
- continuous-payment: pay for used resources as job executes
- post-payment: pay after usage ("credit card model")

The pre-payment approach has the apparent problem of estimating an exact run time prior to job submission. The continuous-payment approach will probably place a heavy load on the accounting system, making it unscalable. In the third case the resource must be guaranteed that sufficient funds will be available after the job has finished.

#### 7.5.1.9. Failures

What happens in case of failures? E.g., if a resource provider crashes, the network fails, a user cancels a (queued or executing) job, etc. In such cases policies are needed to govern what actions need to be taken.

#### 7.5.1.10. Logging and Resource Usage Tracking

It is likely that resource providers, VOs, allocation committees and users are interested in detailed information regarding job execution and resource usage. This information could, e.g., be used for evaluating resource usage or presenting information to the allocations committee, funding agencies, etc. Hence, some sort of logging service, capable of holding detailed information on resource usage might be needed. Moreover, from a user perspective, some sort of "job tracking" functionality could prove very useful. I.e., users would not only have access to information about the resources consumed by their jobs, but also detailed information about the status of the jobs (whether the job finished, failed, cancelled, etc), job input data and job output data.

## 7.5.2. Existing Work

There exist a number of related research projects developing or specifying technology relevant to a large-scale, heterogeneous, and distributed accounting system. In the following, some of the more significant contributions are presented.

### *7.5.2.1. GGF*

There are four research groups within GGF [156] currently working on accounting related standards. They are all developed within the framework of the Open Grid Services Architecture (OGSA) [172] and in compliance with Open Grid Services Infrastructure (OGSI) [173].

#### *7.5.2.1.1. Grid Resource Allocation Agreement Protocol (GRAAP)*

The GRAAP working group [161] is currently specifying a protocol based on Service Level Agreement (SLA), this being called Agreement-based Service Management (WS-Agreement). OGSI-Agreement defines the interactions involved in negotiating and reserving a resource usage contract between agreement providers and agreement initiators. The contract in turn specifies the Quality of Service (QoS) of a delivered service that the service consumer can expect. Further, a contract state machine is defined as well as a term-language framework. Some standard terms are defined but most terms are expected to be standardized in a domain specific context, e.g. for a particular job description language. In fact, unifying the plethora of job description languages used in the Grid today was one of the motivating factors behind the OGSI-Agreement effort, and one of its first applications.

#### *7.5.2.1.2. Resource Usage Service (RUS), and Usage Record (UR)*

RUS [172] is a service that can be used to publish and query resource usage data. It relies heavily on the Usage Record format, which is a standard XML document composed of the various usage properties, like CPU-time, wall-clock time, and disk space, which Grid resources may record. RUS is hence intended to be used both by resources and by various brokers and users interested in usage information, e.g., banking services, work load managers and resource funding agencies [172] [174].

#### *7.5.2.1.3. Grid Economic Services Architecture (GESA)*

GESA [175] essentially extends the OGSI Grid service model into an economic service model, where you can charge consumers for service usage. In order to achieve this, GESA defines an architecture based on OGSI-Agreement contracts and Resource Usage services. This combines all accounting related efforts within GGF into a common model. Two new components are also defined; a Grid banking service and a chargeable Grid service, the latter being a direct extension of the OGSI Grid service.

### 7.5.2.2. European Grid Projects

There are a number of Grid Projects within the EU that are considering various accounting systems to fit their needs [171]. An inventory recently made by the GridStart projects [176] in the EU, however, showed that no complete solutions are yet in production use. The most promising work on a more generic Grid resource accounting model has been developed by the DataGrid project, and is called the Data Grid Accounting System (DGAS) [184].

#### 7.5.2.2.1. DGAS

The DGAS model envisions a whole new economic Grid market, where supply and demand of Grid resources work in unison to strive towards equilibrium where all resources are fully utilized to the lowest possible price. This equilibrium is achieved by a built-in, self-adapting feedback mechanism, where GridCredits can be earned by offering resources, and spent by using resources, and thus mimicking the monetary market. Although visionary and elaborate this model has some shortcomings, an obvious one being that some resource providers would want to trade their earned Grid credits against something else than resource usage. One of the nicest features of DGAS is that it is fully distributed and thus true to the decentralized cornerstone idea of the Grid. Each user has an account in a local bank called the Home Location Registry (HLR). When a job is submitted by the user, the resource broker receiving the request contacts a pricing authority at various resources and the local bank to check whether there are sufficient funds to run the job. If the bank grants the transaction the request is passed to the job controller which sends it to an available resource that matched the user requirements. A resource monitoring service then tracks the job status and the resource usage and sends periodic reports to the HLR. When the job completes the total cost is calculated and possible holds on amounts in the HLR are unlocked and the credits spent are withdrawn from the user HLR and deposited into an account in the resource HLR. However, a concern about DGAS is that it is not based on OGSA technologies. However, an evaluation of the implementation might still be valuable to gain understanding of the problem domain.

#### 7.5.2.2.2. SweGrid Accounting System

The SweGrid Accounting [171] system contains the following main entities:

- Users
- Virtual Organizations (VOs)
- Projects
- Resources
- Bank services
- Log services

In SweGrid the bank accounts will be assigned to projects and not to the individual project members, even though the members have individual user accounts on the Grid. Within each VO, a user might participate in multiple projects (with different bank accounts). Furthermore, there is nothing stopping a user from being a member of projects in several VOs. Each VO has one associated bank service to manage the accounts of the VO projects. However, one bank might service several different VOs. I.e., there is a one-to-many mapping from a VO to its associated

bank. This one-bank-per-VO limit might be too restrictive in a general grid (accounting) environment. However, to date, we have not come up with a scenario where a VO would actually need the services of more than one bank. One such scenario might be if a VO grows too large. In that case a second bank could be introduced to offer some degree of load-balancing. Another such scenario is that of an allocation provider, such as SNAC, which does not trust the VO bank and would like to deploy its own bank. In such a case, it should probably set up its own VO rather than deploying a second bank for the existing VO. The use of more than one bank for a VO needs to be investigated further. The log service holds detailed information about the resources consumed (Usage Records - URs), job status, input data, output data, etc. about jobs submitted by the VO users. The log is a conceptually simple service as it is a write-once/read-many kind of service.

#### *7.5.2.2.3. Other related efforts*

An inventory made by the GridStart projects in the EU [176] showed that no complete solutions are yet in production use. GridStart, AVO [177], EGSO [178], and FlowGrid [179] all have similar accounting needs to SweGrid's (based on scientific funding, and usage tracking focused), whereas EuroGrid [180], GRIA [181], GEMSS [182], and GRASP [183] take more of a business model approach similar to the GESA chargeable service model with an emphasis on the billing process [175].

## 7.6. References for cross-layer themes


[155] Foster, I., Kesselman, C., Nick, J., and Tuecke, S., The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, Globus Project, 2002, http://www.globus.org/research/papers/ogsa.pdf

[156] The Global Grid Forum. http://www.ggf.org

[157] Asif Saleem, Marko Krznari´c, Jeremy Cohen, Steven Newhouse, John Darlington, Using the VOM portal to manage policy within Globus Toolkit, Community Authorisation Service & ICENI resources, UK All Hands Meeting 2004 http://www.allhands.org.uk/2004/proceedings/papers/164.pdf

[158] TrustCoM State of the Art Evaluation – Phase 1, EU FP6 Project Trustcom IST

[159] Privilege and Role Management Infrastructure Standards Validation: http://sec.isi.salford.ac.uk/permis/

[160] Current Projects in ISI at the University of Salford, UK, http://sec.isi.salford.ac.uk/Projects.htm

[161] Grid Resource Allocation Protocol. GGF Working Group. https://forge.gridforum.org/projects/graap-wg

[162] GGF Performance Working-Group, *A Grid Monitoring Architecture*, March 2000, Revised 27 August 2002, http://www-didc.lbl.gov/GGF-PERF/GMA-WG/papers/GWD-GP-16-3.pdf

[163] Resources for Dynamic Host Configuration Protocol (DHCP), http://www.dhcp.org/

[164] SLP: Service Location Protocol RFC2165, Javvin Company, http://www.javvin.com/protocolSLP.html

[165] About the DNS (Domain Name System), DNS Resources Directory, http://www.dns.net/dnsrd/

[166] Web Proxy AutoDiscovery (WPAD), http://www.webopedia.com/TERM/W/WPAD.html

[167] What is Cooltown?, Hewlett-Packard Laboratories, http://www.cooltown.com/cooltown/index.asp

[168] Sentient Computing, http://www.uk.research.att.com/spirit/

[169] ebXML Position Paper, http://java.sun.com/dev/evangcentral/totallytech/ebXML.html

[170] OWL-S: Semantic Markup for Web Services, W3C Member Submission 22 November 2004, http://www.w3.org/Submission/OWL-S/

[171] Elmroth,E, Gardfjäll,P, Mulmo, O, Sandgren, Å, Sandholm,T, "A Coordinated Accounting Solution for SweGrid" Technical Report, October 2003, available at: www.pdc.kth.se/~sandholm/docs/reports/SGAS-0.1.3.pdf

[172] OGSA Resource Usage Service. GGF Working Group. https://forge.gridforum.org/projects/rus-wg

[173] Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C., Maquire, T., Sandholm, T., Snelling, D., and Vanderbilt, P. Open Grid Services Infrastructure (OGSI) Version 1.0, GGF, 2003

[174] Usage Record. GGF Working Group. https://forge.gridforum.org/projects/ur-wg/

[175] Grid Economic Services Architecture. GGF Working Group. https://forge.gridforum.org/projects/gesa-wg

[176] Gagliardi, F. et al. GRIDSTART Project – IST Grid Projects Inventory and Roadmap. GRIDSTART-IR-D2.21.2-V0.5. EU 2003. http://www.gridstart.org/GRIDSTART-IR-D2.2.1.2-V0.5.doc

[177] AVO, http://www.euro-vo.org

[178] EGSO, http://www.mssl.ucl.ac.uk/grid/egso/egso_top.html

[179] FlowGrid (FLOW simulations on-demand using GRID computing), http://www.unizar.es/flowgrid/

[180] Eurogrid, http://www.eurogrid.org

[181] GRIA, http://www.gria.org

[182] GEMSS (Grid-Enabled Medical Simulation Services), http://www.ccrl-nece.de/gemss/

[183] GRASP (GRid based Application Service Provision) http://eu-grasp.net/

[184] Guarise, A., Piro, R., and Werbrouck, A. DataGrid Accounting System – Architecture – 24v1.0. EU DataGrid 2003. http://server11.infn.it/workload-grid/docs/DataGrid-01-TED-0126-1_0.pdf

[185] Herzog, R., Lausen, H., Roman, D., Zugmann, P. (eds.): WSMO Registry. WSMO working draft, available at http://www.wsmo.org/2004/d10/v0.1/ , 2004

[186] Paolucci, M., Kawamura, T., Payne, T. R., Sycara, K.: Importing the Semantic Web in UDDI, Proceedings of Web Services, E-business and Semantic Web Workshop, 2002

[187] <need to fix this reference> http://www.w3.org/Submission/2004/07/

[188] OWL-S specification, http://www.daml.org/services/owl-s/

[189] http://www.lesc.ic.ac.uk/iceni/downloads/materials/AHM2003/vom.pdf

[190] http://www-124.ibm.com/developerworks/oss/lsid/

[191] Recommendation X.500, Information technology – Open System Interconnection – The directory : Overview of concepts models, and services. *ITU-T*, November 1995

[192] Universal description discovery and integration (UDDI). http:// www.uddi.org, 2001

[193] What's LDAP? *ClickMail Central Directory Gracion Software* http://www.gracion.com/server/whatldap.html

[194] Grid Information Services for Distributed Resource Sharing, *Carl Kesselman – Ian Foster* http://www.globus.org/research/papers/MDS-HPDC.pdf

[195] https://lcg-registrar.cern.ch/virtual_organization.html

[196] Virtual Organization Membership Service eXtension project (VOX) computing.fnal.gov/docdb/documents/ 0002/000221/001/VOX_status_report.ppt

[197] Yourdictionary, www.yourdictionary.com

[198] Liberty alliance, www.projectliberty.org

[199] Liberty Alliance Project White Paper. Liberty Alliance & WS-Federations: A Comparative overview. October 2003.

[200] European DataGrid Project, http://www.eu-datagrid.org

[201] The Information Power Grid - http://www.ipg.nasa.gov

[202] eXtensible Access Control Markup Language (XACML) Version 1.0, OASIS Standard.

[203] Djordjevic I., Dimitrakos T., Phillips C. "An Architecture for Dynamic Security Perimeters of Virtual Collaborative Networks" IFIP/IEEE NOMS2004.

[204] A. Saleem, M. Krznaric, S. Newhouse, and J. Darlington. ICENI Virtual Organisation Management. In *UK e-Science All Hands Meeting, p. 117–120, Nottingham, UK*, 2003.

[205] D.Kelsey, EU DataGrid Security, presented at the Grid Security Workshop, held at the UK National e-Science Centre, December 2002

[206] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, D. Spence, Generic AAA Architecture, RFC 2903, August 2000, http://www.ietf.org/rfc/rfc2903.txt

[207] OASIS, The OASIS Security Assertion Markup Language (SAML) V2.0, available via http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

# 8. Updating the State of the Art

It is unlikely that the State of the Art will stand still during the Project. For example, there are competing or overlapping Web Service specifications; OGSA requires more precise specification; and further specifications will be developed in relation to mobility and semantic issues. In all areas the status of implementations will change.

Experts in specific topics in Akogrimo will be keeping themselves up to date with these and other external developments, as required by the design and implementation phases of this Project and as required by their organisations. Nonetheless it will still be required to make the knowledge more widely available within the Project.

Instead of reissuing this entire document frequently which would be somewhat heavyweight, there will be a more lightweight process involving the use of update documents. At a certain frequency to be determined, technical experts will be requested to contribute updates. It will also be appropriate to ensure that an update is made in advance of the 2nd cycle of the Project.

# 9. Abbreviations and terms

## 9.1. Abbreviations

| | |
|---|---|
| **2G** | Second-generation wireless telephone technology. |
| **3G** | Third-generation mobile telephone technology |
| **AAA** | Authentication, Authorisation and Accounting |
| **Akogrimo** | Access To Knowledge through the Grid in a Mobile World |
| **CIM** | Common Information Model |
| **CN** | Correspondent Node |
| **CoA** | Care-of Address |
| **COPS** | Common Open Policy Service |
| **DCF** | Distributed Control Function |
| **DMTF** | Distributed Management Task Force |
| **EAP** | Extensible Authentication Protocol |
| **EDCF** | Enhanced Distributed Control Function |
| **GPRS** | General Packet Radio Service |
| **GSM** | Global System for Mobile Communications |
| **HA** | Home Agent |
| **HoA** | Home Address |
| **HCF** | Hybrid Control Function |
| **IPSec** | IP Security |
| **LAN** | Local Area Network |
| **MAN** | Metropolitan Area Network |
| **MDVO** | Mobile Dynamic Virtual Organisation |
| **MIPv6** | Mobile IP version 6 |
| **MN** | Mobile Node |

| | |
|---|---|
| **MOWS** | Management Of Web Services |
| **MUWS** | Management Using Web Services |
| **nrtPS** | Non-real-time Polling Service |
| **OASIS** | Organization for the Advancement of Structured Information Standards |
| **OGSA** | Open Grid Service Architecture |
| **PAN** | Personal Area Network |
| **PANA** | Protocol for carrying Authentication for Network Access |
| **PBNM** | Policy-based Network Management |
| **PCF** | Point Control Function |
| **PDP** | Policy Decision Point |
| **PEP** | Policy Enforcement Point |
| **PPTP** | Point-to-point tunnelling Protocol |
| **rtPS** | Real-time Polling Service |
| **SDS** | Service Discovery Service |
| **SIP** | Session Initiation Protocol |
| **SNMP** | Simple Network Management Protocol |
| **SSL** | Secure Socket Layer |
| **TLS** | Transport Layer Security |
| **UGS** | Unsolicited Grant Service |
| **UMTS** | Universal Mobile Telecommunications System |
| **VoIP** | Voice over IP |
| **VPN** | Virtual Private Network |
| **WEP** | Wired Equivalent Privacy |
| **WiFi** | Wireless Fidelity |
| **WiMAX** | Worldwide Interoperability for Microwave Access |
| **WLAN** | Wireless Local Area Network |

| | |
|---|---|
| **WPA** | WiFi Protected Access |
| **WSDL** | Web Service Description Language |
| **WSRF** | Web Service Resource Framework |

## 9.2. Terms

This section provides definitions of major terms as used in this document.

Sources used for definitions of some terms include:

- Webopedia: Online Computer Dictionary for Computer and Internet Terms and Definitions [208]
- W3C Web Services Glossary [209]
- Terms and Definitions Database Interactive (TEDDI).[210]
- Yourdictionary.com [211]
- RFCs, which (using the example RFC3334), are accessed using a URL of the form http://www.faqs.org/rfcs/rfc3334.html

If consulting this document online, this letter list can be used to navigate alphabetically: [A] [B] [C] [D] [E] [F] [G] [H] [I] [J] [K] [L] [M] [N] [O] [P] [Q] [R] [S] [T] [U] [V] [W] [X] [Y] [Z]

[A]

| | |
|---|---|
| Term: | Accounting |
| Definition: | Accounting describes the collection of data about resource consumption. This includes the control of data gathering (via metering), transport and storage of such data |
| Comment: | This definition is used in a networking context, which need not preclude its use in other contexts. |
| Source: | RFC 3334 [100] |

| | |
|---|---|
| Term: | Authentication, Authorisation and Accounting (AAA) |
| Definition: | Mechanism for identifying users, authorising or denying their access to specific resources and accounting for their usage. Explained in sections 4.2.4, 7.4 and 7.5. |
| Comment: | |

| | |
|---|---|
| Source: | [86] |

[B]

| | |
|---|---|
| Term: | Billing |
| Definition: | Billing translates costs calculated by a charging scheme into monetary units and generates a final bill for the customer. |
| Comment: | This definition is used in a networking context, which need not preclude its use in other contexts. |
| Source: | RFC 3334 [100] |

| | | |
|---|---|---|
| Term: | Business Process Execution Language for Web Services (BPEL4WS) | |
| Definition: | "BPEL4WS explicitly allows the use of nondeterministic data values to make it possible to capture the essence of public behaviour while hiding private aspects. […] It is also possible to use BPEL4WS to define an executable business process. The logic and state of the process determine the nature and sequence of the Web Service interactions conducted at each business partner, and thus the interaction protocols." | |
| Comment: | This definition is used by the official development project consortium of BPEL4WS consisting of BEA, SAP, Microsoft, IBM, Siebel; | |
| Source: | http://www.oasis-open.org/committees/download.php/4413/wsbpel-specification-draft-Nov2303.htm | |

[C]

| | |
|---|---|
| Term: | Charging |
| Definition: | Charging derives non-monetary costs for accounting data sets based on service and customer specific tariff parameters. A charging scheme is an instruction for calculating a charge and is usually represented by a formula that consists of charging variables (e.g. volume, time, reserved peak rate) and charging coefficients (e.g. price per time unit). The charging variables are usually filled by information from accounting data. Charging policies define the tariffs and parameters which are applied. |
| Comment: | This definition is used in a networking context, which need not preclude its use in other contexts. |

| | |
|---|---|
| Source: | RFC 3334 [100] |

| | |
|---|---|
| Term: | Common Information Model (CIM) |
| Definition: | |
| Comment: | The CIM is a conceptual information model for describing managed entities, their composition and their relationships. |
| Source: | [8] |

[D]

| | |
|---|---|
| Term: | Distributed Management Task Force (DMTF) |
| Definition: | The Distributed Management Task Force, Inc. (DMTF) is the industry organization leading the development of management standards and integration technology for enterprise and Internet environments. DMTF standards provide common management infrastructure components for instrumentation, control and communication in a platform-independent and technology neutral way. DMTF technologies include information models (CIM), communication/control protocols (WBEM), and core management services/utilities. |
| Comment: | |
| Source: | http://www.dmtf.org/about |

| | |
|---|---|
| Term: | Dynamic Virtual Organisation (DVO) |
| Definition: | The "dynamic nature" implies that the entire set up of a virtual organization may change in response to the market place. In this sense, virtual organizations of this type are temporary as to their ability to react quickly as regards the membership, the structure, the objectives etc. Its fluid boundaries and opportunism, as well as equity of partners and shared leadership mainly characterize a dynamic virtual organization |
| Comment: | |
| Source: | Literature http://www.vtt.fi/ |

[E]

| Term: | End-to-End security |
|---|---|
| Definition: | "Securing the data exchanged between two communicating nodes in a way such that the data is not exposed to the intermediate entities that forward the traffic from one point to the other" |
| Comment: | |
| Source: | |

[F]

[G]

| Term: | Grid Service |
|---|---|
| Definition: | A Grid Service is a Web Service that follows specific conventions to do with address discovery, dynamic service creation, lifetime management, notification and manageability. |
| Comment: | |
| Source: | [155] |

[H]

[I]

| Term: | Identity |
|---|---|
| Definition: | "The collective aspect of the set of characteristics by which a thing is definitively recognizable or known" |
| Comment: | |
| Source: | Yourdictionary [211] |

| Term: | IPv6 |
|---|---|
| Definition: | "IP version 6 is the internet protocol which is intended to replace the previous standard IPv4. The major changes involved are the following: expanded addressing capabilities, header format simplification, improved support for extensions and options, flow labelling capability, and authentication and privacy capabilities." |
| Comment: | |
| Source: | Based on RFC 2460 |

| Term: | Inter-domain Mobility |
|---|---|
| Definition: | "The ability to transparently use resources of different administration domains while holding a single contract with only one of them." |
| Comment: | |
| Source: | |

[J]

[K]

[L]

| Term: | Local Area Network (LAN) |
|---|---|
| Definition: | "A network that spans a relatively small area." |
| Comment: | |
| Source: | |

[M]

| Term: | Management Of Web Services (MOWS) |
|---|---|

| | |
|---|---|
| Definition: | MOWS defines the manageability model for managing Web services as a resource and how to describe and access that manageability using MUWS. |
| Comment: | |
| Source: | An overview of the work of the OASIS Web Services Distributed Management Technical Committee, which includes MOWS [138] |

| | |
|---|---|
| Term: | Management Using Web Services (MUWS) |
| Definition: | MUWS defines how to represent and access the manageability interfaces of resources as Web services. It is the foundation of enabling management applications to be built using Web services and allows resources to be managed by many managers with one set of instrumentation. |
| Comment: | |
| Source: | An overview of the work of the OASIS Web Services Distributed Management Technical Committee, which includes MUWS [138] |

| | |
|---|---|
| Term: | Meter, metering |
| Definition: | In a network, a meter is responsible for measuring traffic; it observes packets as they pass by a single point on their way through a network and classifies them into certain groups, this process being referred to as metering. |
| Comment: | This definition is used in a networking context, which need not preclude its use in other contexts. |
| Source: | Based on RFC3334 [100] and RFC 2722 [101]. |

| | |
|---|---|
| Term: | Metropolitan Area Network (MAN) |
| Definition: | "A large network usually spanning a campus or a city." |
| Comment: | |
| Source: | Based on www.wikipedia.com |

| | |
|---|---|
| Term: | MIPv6 |

| | |
|---|---|
| Definition: | "Is a protocol which allows nodes to remain reachable while moving around in the IPv6 Internet." |
| Comment: | |
| Source: | RFC 3775 |

| | |
|---|---|
| Term: | Mobile, Mobility |
| Definition: | In common usage, mobile or mobility refers to the ability to move. A mobile phone, by contrast with a domestic phone, can be used wherever its bearer is located, subject to restrictions of technology and social obligation. In Akogrimo, the idea of mobility is extended to any device that enables a user to stay connected while on the move, to services on a Grid, and to the Grid notion of Virtual Organisation (VO). It also includes the idea that a user can be mobile by moving from one device to another, while retaining the continuity of engagement with ongoing services. |
| Comment: | |
| Source: | |

| | |
|---|---|
| Term: | Mobile Dynamic Virtual Organisation (MDVO) |
| Definition: | "A Dynamic Virtual Organisation with at least one essential entity (in Akogrimo's case typically an Application User) that is not bound to a location but can move so that mobility aspects like contextuality and personalization become important". In addition it is a goal of Akogrimo that mobility will extend to the services not only the users. |
| Comment: | |
| Source: | Akogrimo via University of Zurich |

[N]

| | |
|---|---|
| Term: | Nomadic |

| | |
|---|---|
| Definition: | A more limited form of mobility where the user or service may disengage from the network for a period and re-engage when a suitable point of access is available, possibly at a different location. Continuity of service is generally required. |
| Comment: | |
| Source: | |

[O]

| | |
|---|---|
| Term: | Open Grid Services Architecture (OGSA) |
| Definition: | Historically, the notion of a networked Grid began with an idea which was initially defined in terms of a specific implementation, but matured into a publicly defined architecture, the Open Grid Services Architecture (OGSA). |
| Comment: | |
| Source: | A summary of the OGSA activity in the GGF, this also provides a link to the current OGSA document itself [125] |

| | |
|---|---|
| Term: | Open Grid Services Infrastructure (OGSI) |
| Definition: | OGSI refers to the first base infrastructure on which OGSA has been built, now being replaced by WSRF. It defines the standard interfaces and behaviors of a Grid service, building on a Web services base. |
| Comment: | |
| Source: | A summary of the OGSA activity in the GGF, this also provides a link to the current OGSI document itself [125] |

| | |
|---|---|
| Term: | Operator |
| Definition: | Within a discussion concerned with mobile access to a computer network, an Operator is in generally the provider of a mobile network and there is a responsibility and motivation for mobile networks to interwork. |
| Comment: | |

| | |
|---|---|
| Source: | |

| | |
|---|---|
| Term: | Organisation for the Advancement of Structured Information Standards (OASIS) |
| Definition: | It is a not-for-profit, international consortium that drives the development, convergence, and adoption of e-business standards. |
| Comment: | OASIS is responsible for the development of many Web Service standards. |
| Source: | The OASIS "about" web page http://www.oasis-open.org/who/ |

[P]

| | |
|---|---|
| Term: | Policy Based Network Management (PBNM) |
| Definition: | "The management of complex networks by means of a set of rules which are followed by the devices controlling the network configuration." |
| Comment: | |
| Source: | |

[Q]

| | |
|---|---|
| Term: | Quality-of-Service (QoS) |
| Definition: | "Quality of Service is defined in CCITT Recommendation E.800 as follows: "The collective effect of service performances which determine the degree of satisfaction of a user of the service."<br>For a given service, QoS is a statement of the performance of the service as offered or specified to the customer. It is defined and measured in terms of parameters which are stated in user understandable language appropriate to the particular service concerned, and which are user verifiable. These parameters will depend upon the service definition, and upon the point at which the service is accessed by the user" |
| Comment: | |
| Source: | Literature TEDDI |

[R]

[S]

| Term: | Service |
|---|---|
| Definition: | (1) In the context of a network layer, this is the provision of a specific function to a customer connected to the network.<br>(2) In a Grid context, it is specifically a Grid Service or a Web Service. |
| Comment: | |
| Source: | |

| Term: | Service Discovery Service (SDS) |
|---|---|
| Definition: | Provides a directory of services available in a network |
| Comment: | |
| Source: | |

| Term: | Service Level Agreement (SLA) |
|---|---|
| Definition: | An SLA is an agreement between the provider and consumer of a service. In the Web and Grid Service world this may result from a negotiation between a provider and consumer and subsequently needs to be monitored for enforcement and accounting purposes. |
| Comment: | |
| Source: | |

| Term: | Session Initiation Protocol (SIP) |
|---|---|
| Definition: | Session Initiation Protocol is being standardized by the IETF and is a protocol oriented to establish multimedia communication services over IP networks. SIP allows users to call each other independently of their location. |
| Comment: | |

| | |
|---|---|
| Source: | [18] |

[T]

| | |
|---|---|
| Term: | Terminal Mobility |
| Definition: | "The ability of a terminal to freely change its location while maintaining alive the communications already established with other entities." |
| Comment: | |
| Source: | |

[U]

[V]

| | |
|---|---|
| Term: | Virtual Organisation (VO) |
| Definition: | « A network of organisations and/or individuals, with a commonality of purpose or interest, which collectively make up an identifiable, coherent, entity » |
| Comment: | |
| Source: | Now commonly used within Grids, an early use of this is in [127]. |

| | |
|---|---|
| Term: | Virtual Private Network (VPN) |
| Definition: | "A VPN is a private network which communicates over a public network. Secure VPNs use cryptographic tunnelling protocols to provide confidentiality, sender authentication and message integrity." |
| Comment: | |
| Source: | Based on www.wikipedia.com |

[W]

| Term: | Web Service |
|---|---|
| Definition: | The World Wide Web is more and more used for communication between applications. The programmatic interfaces made available are referred to as *Web services*.<br>A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. |
| Comment: | Note that the preceding the definition in the Web Services Architecture document, it says: For the purpose of this Working Group and this architecture, and without prejudice toward other definitions, we will use the following definition. |
| Source: | Web Services Architecture document at W3C http://www.w3.org/TR/ws-arch/ . |

| Term: | Web Service Description Language (WSDL) |
|---|---|
| Definition: | WSDL is a language for describing Web Services |
| Comment: | Each Web Service has a description associated with it using the WSDL XML-based language. WSDL is machine-processable and human-readable. One of the aspects of a Web Service described in WSDL is the set of message types accepted and produced by the Web Service |
| Source: | Web Services Architecture document at W3C http://www.w3.org/TR/ws-arch/ |

| Term: | Web Service Resource Framework (WSRF) |
|---|---|
| Definition: | The WSRF is a set of Web Service specifications that define an approach to modelling and managing state in a Web Service context. It treats the persistent state as a resource. This is a long term successor to OGSI as an supporting infrastructure for the Open Grid Services Architecture (OGSA) |
| Comment: | |
| Source: | Web page at GGF introducing the WS-Resource Framework [131] |

[X]

[Y]

[Z]

## 9.3. References for Terms

[208] Webopedia: Online Computer Dictionary for Computer and Internet Terms and Definitions, http://www.webopedia.com/

[209] W3C Web Services Glossary, http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/

[210] Terms and Definitions Database Interactive (TEDDI). http://webapp.etsi.org/Teddi/ , access: 2004-09-06

[211] Yourdictionary, www.yourdictionary.com