Instant Messaging based on OMA- IMPS: Interworking Guidelines

1.1

18 February 2008

This is a non-binding permanent reference document of the GSM Association.

Security Classification Category (see next page)
UNRESTRICTED

Security Classification: Unrestricted

Copyright Notice

Document History

Version	Date	Brief Description
0.0	20/12/06	
0.1	02/01/07	Revisions by Vodafone and Telenor
0.2	12/01/07	Revisions by R Mangtani, GSMA
0.3	18/01/07	Addition of IPIAG document
0.4	05/03/07	Revisions by R Mangtani, GSMA following conditional approval by Packet 28.
0.5	31/3/07	Revisions as requested by DAG 36, removal of SDO diagram, references to wireless village, update section 3, annex 6 remove emoticons and ext:ui table.
0.6	18/02/2008	Addition of test cases from Peter Dawes, Vodafone as Annex B
1.0		Approved

Table of Contents

1 GLOSSARY

Terms	Definitions
CSP	Client to Server Protocol
DNS	Domain Name System
ENUM	E.164 number and DNS
HLR	Home Location Register
HPLMN	Home Public Land Mobile Network
IM	Instant Messaging
IMPS	Instant Messaging and Presence Service
NAPTR	Naming Authority Pointer DNS Resource Record
OMA	Open Mobile Alliance
PDP	Packet Data Protocol
PDP	Policy Decision Point
PDU	Protocol Data Unit
QoS	Quality of Service
SIP	Session Initiation Protocol
SSP	Server to Server Protocol
UE	User Equipment
URI	Uniform Resource Identifier
URL	Universal Resource Locator
VPLMN	Visited Public Land Mobile Network

2 SCOPE

The goal of this PRD is to ensure that interworking is handled correctly with Instant Messaging Service using the interworking protocol OMA-IMPS. Use of any other IM protocol such as SIP/SIMPLE is out of the scope of this document and should be addressed in separate documents.

This PRD concentrates on the first phase of IM introduction, i.e. what can be done with the IMPS protocol by operators within a relatively short timeframe. The goal of the document is not to describe each and every aspect of IMPS in detail, but rather concentrate on the most important issues the first implementations of IM using OMA-IMPS will face when interworking.

This PRD introduces guidelines for usage of inter-PLMN connections in an OMA-IMPS environment and the requirements OMA-IMPS has for inter-PLMN IP backbone network. Other issues discussed here are e.g. addressing and routing implications of OMA-IMPS.

Note that this PRD does not aim to give a comprehensive introduction. (See OMA-ERELD-IMPS-V1_2-20050801-A.pdf [3] document for that purpose) to OMA-IMPS, however, chapter 3 has a short introduction

This PRD concentrates on network layer interworking; therefore higher layer issues like service interconnection are not discussed in detail. Radio interface, QoS details, GPRS backbone, interworking with PSTN as well as connections between IMS network elements and terminals/applications are not within the scope of this document. Similarly connections to private networks such as corporate networks are also out of scope.

3 BACKGROUND

Background issues are limited here in the interests of clarity and brevity, readers needing further introduction to OMA-IMPS should refer to the OMA specifications relevant to OMA-IMPS given under reference [3].

3.1 Introduction

Instant Messaging and Presence services have become widely available in the Internet during the last decade. PC users with Internet connections have been enjoying different services providing ever-increasing number of features for a number of years. However, none of the service providers agreed to work together to provide an interoperable solution. Despite this lack of interoperability, several of the services were successful. Most of the IM communities built so far are based on proprietary solutions (MSN, AOL, etc.) with not many interworking agreements in place. Despite that, every community has a lot of subscribers and the amount of traffic interchanged could be seen as considerable. Mobile devices were completely shut out of the Internet IM systems for a long time the closest thing to IM in the mobile world was the widespread use of SMS (Short Message Service) messages. The perceptions of the mobile devices have since changed from simple mobile phones that could only be used to make phone calls, to powerful minicomputers. Applications unimaginable only five years ago are now standard on many smart phones.

There are various manufacturers of devices, proprietary software technologies and access technologies etc. that create problems when instant messaging and presence services are enabled. IM services in mobile operators are required to have to full interoperability from the outset. The enabling protocols, like IMPS, have been agreed between main mobile environment providers and operators in order to guarantee that a fully interoperable service is deployed.

Additionally, in order to guarantee the success of the mobile IM service, it is likely that interoperability with currently deployed IM Internet communities will be offered to mobile IM users. Of course, this interoperability will be based on bilateral agreements between each mobile operator and e.g. MSN, AOL or other IM community, this requirement is out of the scope of global requirements to the whole mobile community, but dealing with these two different types of traffic raises important issues when proposing network interworking.

The Open Mobile Alliance (OMA) was formed in 2002 for resolving technical issues related to mobile services. OMA defines open specifications for the mobile industry, helping to create interoperable services. In summary, the core principle of OMA is to make products and services not dependent on proprietary technologies but instead create open global standards, protocols and interfaces. To solve the standardization problem in the instant messaging and presence area, the OMA IMPS specification v1.1 was published in November 2002. Based on Wireless Village v1.1 specifications, OMA IMPS v1.2 [3] was published in January 2005.

Nowadays most major mobile phone manufacturers include OMA IMPS compliant application software in their terminals by default.

3.2 IM Interworking scenarios

Although the service is generically noted as Instant Messaging, it is composed of two features presence and messaging interactions. Messaging is the procedure whereby a user sends text

information to another user on a one-to-one basis. Presence refers to the possibility of displaying and changing the users on line status i.e. the ability/willingness to send/receive IM. These definitions are related to the first step of IM service deployment. In the following phases, messaging and presence become the basis of more complex services involving the exchange of any type of near real/non-real time information and much more detailed user related information. The simpler scenario is shown in the following figure:

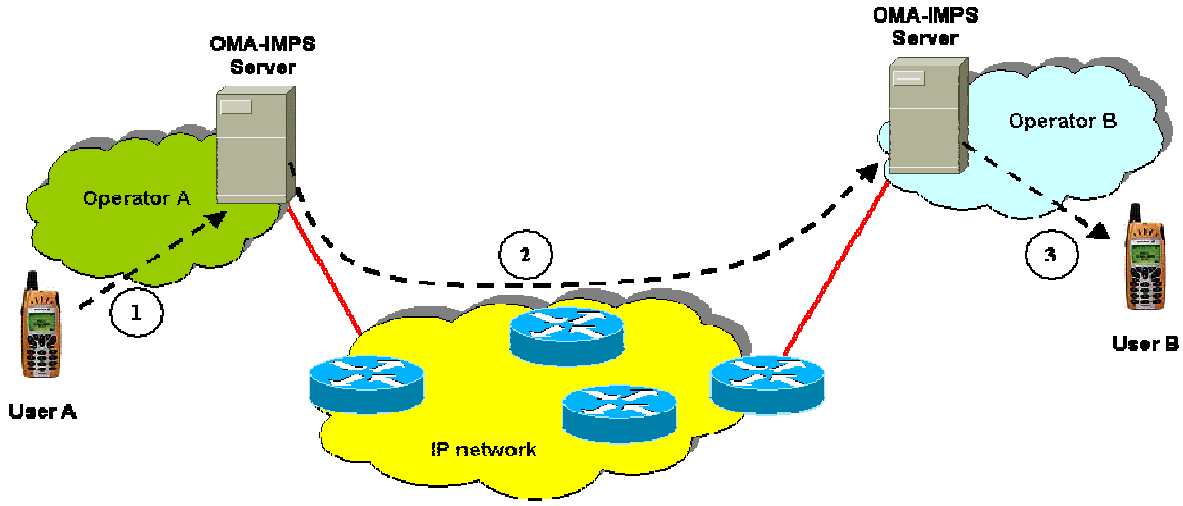

Figure 1: IM Interworking Case

Clarification of Figure 1 (customer of Operator A sends an IM to a customer of Operator B):

1. User A sends an IM with User B's IMPS address (wv:user@domain)
2. Operator A's IMPS server based on receiver domain notes that this IM is addressed to Operator B. It resolves the IP address of Operator B's IMPS server and sends the IM over the common IP network
3. Operator B's IMPS server sends the IM directly to its User B, assuming the user is available and/or willing to receive IM.

The typical Presence interaction is summarised below in Figure 2:

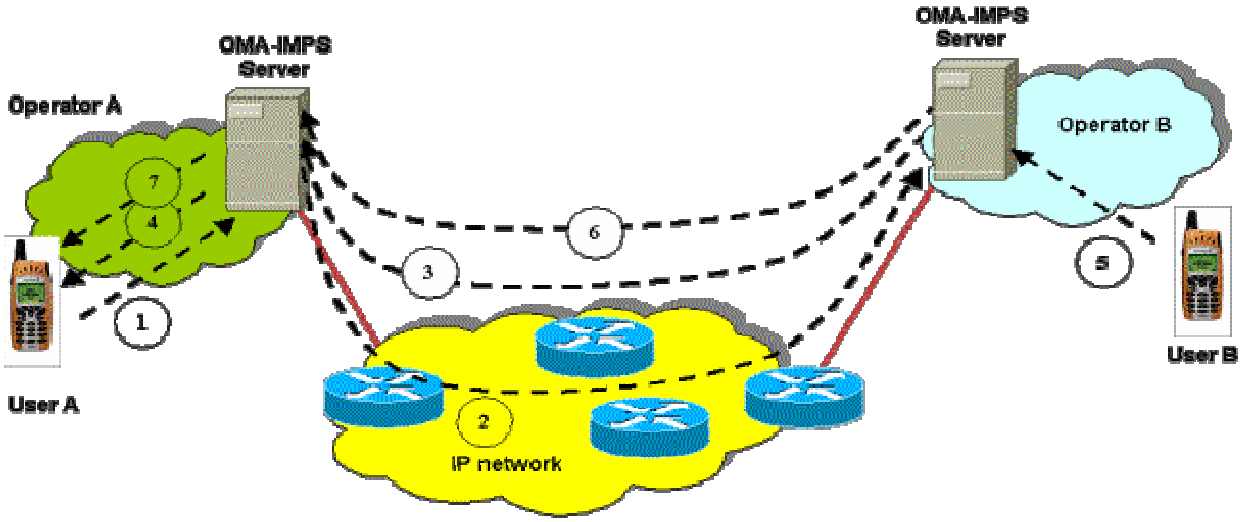

Figure 2: Basic Presence Interworking Case

Clarification of Figure 2 (A customer of Operator A subscribes to the presence of a customer of Operator B):

1. User A sends an Presence subscribe request to User B's using its IMPS address (wv:user@domain)

2. Operator A's IMPS server based on receiver domain notes that the request is addressed to Operator B. It resolves the IP address of Operator B's IMPS server and sends the request over common IP network. The IMPS server of operator B subscribes User A to the presence of its user, User B (subject to user B expressed preferences).

3. The current state of User B's presence (attributes) is sent back to the User A via IMPS Server A

4. User A receives User B's presence update from IMPS server A.

5. Eventually, User B changes its presence information (e.g. moves from online→offline)

6. IMPS Server B also sends such updated information to any user who subscribes to User B's presence, in this case, User A. Thereafter the presence update is sent back towards IMPS Server A.

7. User A receives update of User B's presence from IMPS server A.

3.3 Architecture.

Unlike some popular Instant Messaging systems (e.g. Skype) that utilise peer-to-peer technology, the IMPS, solution is purely a client-server-based solution. Direct peer to peer (client-to-client) communications do not exist under any circumstances nor are any protocols defined in the OMA IMPS specification for direct client-to-client communication.

The main elements of the system are:

- IMPS Servers
- IMPS Clients: embedded and CLI clients
- Mobile Core Network
- Proprietary Gateway and proprietary server(s)

The IMPS Servers are the core of the solution. The majority of the system functionality resides in the servers. Servers connect to OMA IMPS-compliant clients using the CSP and CLP protocols, connect to other OMA IMPS servers using the SSP protocol and to elements of the mobile core network utilizing the SMCNP protocol. A logical element within the IMPS Servers, Service Access Point, serves as the connection point to the other elements.

The IMPS servers are inter-connected with the SSP protocol. This protocol is used for both intra-domain and inter-domain communication, across different service providers (such as different mobile operators). It is also used for the communication between the Proprietary Gateway shown in figure 3 and the IMPS Servers. Similarly to the CSP protocol, the SSP protocol is XML-based.

The SSP connections run over HTTP (port 80) or HTTPS (Port 443) and between two IMPS servers, at least two TCP connections are established, one for outgoing requests and the second for incoming requests. The exchange of messages between two IMPS servers typically takes place via an in intermediate IP network to the other IMPS server. However, the servers also support direct routing of the messages according to business and service agreements. There is no automatic discovery of the IMPS servers; the configuration (connections and any routing arrangements) must be configured manually.

4 INTERWORKING AND ROAMING

Roaming is not addressed in this document because it doesn't make sense to define "IM roaming". The roaming is managed as normal at the GPRS data layer and as such is out of the scope for this document.

In this context the meaning of term IM interworking based on using OMA-IMPS, is that the home network's packet resources are used in order to connect to the home network OMA-IMPS core resources. These are used to set up an IM session with a customer of another OMA-IMPS capable network, independently of whether the customer is roaming or not. In other words, IMPS interworking means that two different IMPS servers are connected through an inter-PLMN IP network to enable interchange of IM and Presence information.

Although it has been said in section 3 above that OMA-IMPS connections and routing agreements required for interworking should be configured manually, because OMA-IMPS is intended to be used over a direct link (it might involve an intermediate IP network, but it is just a logical direct link each end point knows the IP address of the other end point). This approach is not feasible where there are multiple mobile operator networks therefore it is suggested that OMA-IMPS servers should be integrated in mobile network operator infrastructures in both local and interworking environments. In the latter case one, that means integration with the inter-PLMN Backbone network, GRX/IPX, which will see the IM connection as a new service e.g. MMS, the traffic of which it has to deal with. Details of that integration will be described in chapter 6. In this chapter, only what OMA-IMPS requires for interworking is addressed.

Interworking means, in the OMA-IMPS environment, to implement the procedures and rules defined by OMA-IMPS SSP standards. OMA-IMPS SSP Protocol Stack looks like:

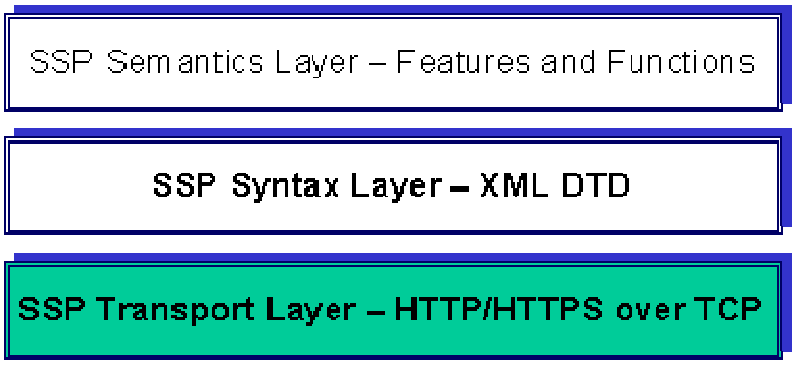

Figure 3: OMA-IMPS SSP Protocol Stack

OMA-IMPS SSP Protocol is implemented by Request – Responses messages sent between both servers. A pair of Request - Response messages is defined as an SSP Transaction. Any SSP message, Request or Response, is implemented by SSP Primitives, which are XML documents.

The SSP Primitives are pipelined onto Transport Layer units, which are in this case HTTP/HTTPS messages over TCP connections. HTTP is a pure Client-Server asymmetrical, protocol. The HTTP method used for carrying the SSP Primitives is POST, which is intended to convey large amount of user data, like user formularies, to be processed by Web Servers. In OMA-IMPS, both Request and Responses are meant to carry a large amount of information. Therefore, for all the reason exposed, OMA-IMPS requires two persistent TCP connections, each of them carrying messages in one direction only. A pair of TCP connection is the minimum set required, but operators can agree to build additional pairs for the sake of redundancy and load balancing.

In addition to the two transport connections, interconnecting two IMPS servers requires building two logical connections at the IMPS or Application level. These two logical connections are named sessions and are, more or less, the IMPS realization of the Service Level Agreements achieved between the operators willing to interoperate. Sessions are built between operators using Login IMPS procedure, whereby both IMPS servers authenticate each other and agree to offer IM related services to the counterpart, refer to section 5.2 for details. It is down to each server to build and keep alive one of the sessions, this one is seen as the service Provider.

Therefore, only two Sessions are required and it is down to the operators how to map those sessions to the TCP connection pair(s). 1-to-1 mapping and connection reuse mapping are allowed. The following figure 5 summarizes what has been said, describing 1-to-1 mapping.

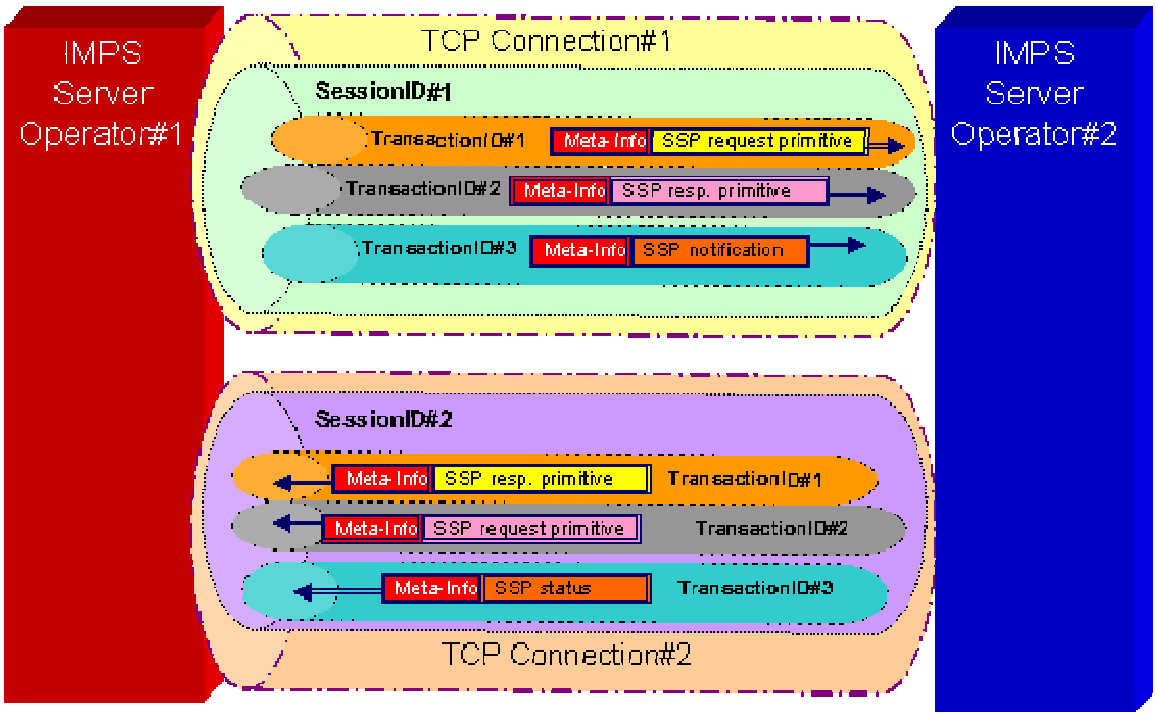

Figure 4: Session-TCP connections 1-to-1 mapping.

As shown in the figure, any OMA-IMPS SSP transaction, such as a Request-Response pair, is carried in different TCP connections and might be carried in each different session,

depending on the mapping. Transaction ID is the tag used to correlate both messages and it is up to the IMPS Servers to do this correlation.

The above outlines how IMPS is intended to be used in both a one-to-one direct connection and routing. Both servers need to make a direct mutual login in order to interwork and they must store both TCP connection and session connection information and mapping in order to find the other domain and send the messages to it using the right session and connection parameters.

The requirement that IMPS has, because of the transport protocol used (HTTP), the method chosen (POST) and the version of the protocol required (HTTP 1.1) is to maintain those two persistent TCP connections between both servers. Despite that, in practical terms, this is actually a loose requirement. As it can be seen in the picture, any TCP connection is carrying such a method of communication. Therefore, every IMPS server must discover the port in which its counterpart is listening for its traffic. That could complicate routing table provisioning in the IMPS server, but, fortunately, destination port is always the HTTP(S) well-known ports 80 or 443.

Despite that, it must be said that this solution has only a relatively short timescale for implementation, because provision of routing tables for direct links hardly scales in such a global service. The Integration of this service in the Inter-PLMN backbone network is a challenging task and is addressed in the following chapters.

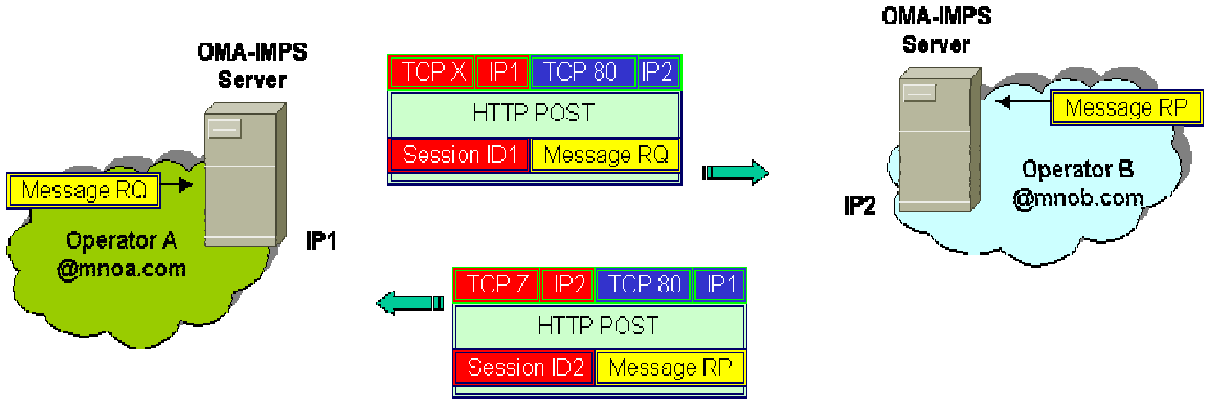

Figure 5: IMPS traffic.

5 REQUIREMENTS FOR THE INTER-PLMN NETWORK

5.1 General Issues

General requirements for the inter-PLMN backbone are applied from PRD IR.34 “Inter-PLMN backbone guidelines” [1].

From a technical point of view the required IP network between IM services from different mobile operators might be e.g. public Internet (with VPN) or direct leased line such as Frame relay or ATM. Another solution, which in many cases could be considered to be the recommended one, is to utilize an existing, proven and reliable inter-operator IP network, i.e. GRX/IPX, as specified in IR.34 [1].

Using GRX/IPX networks to carry IM traffic is less onerous than building direct connections between each and every IM network in the world. The DNS system will resolve the destination domain to an IP address that will be routed over the appropriate link. The IP connection is straightforward (see next figure). Transport and logical interconnectivity are addressed in chapter 6.

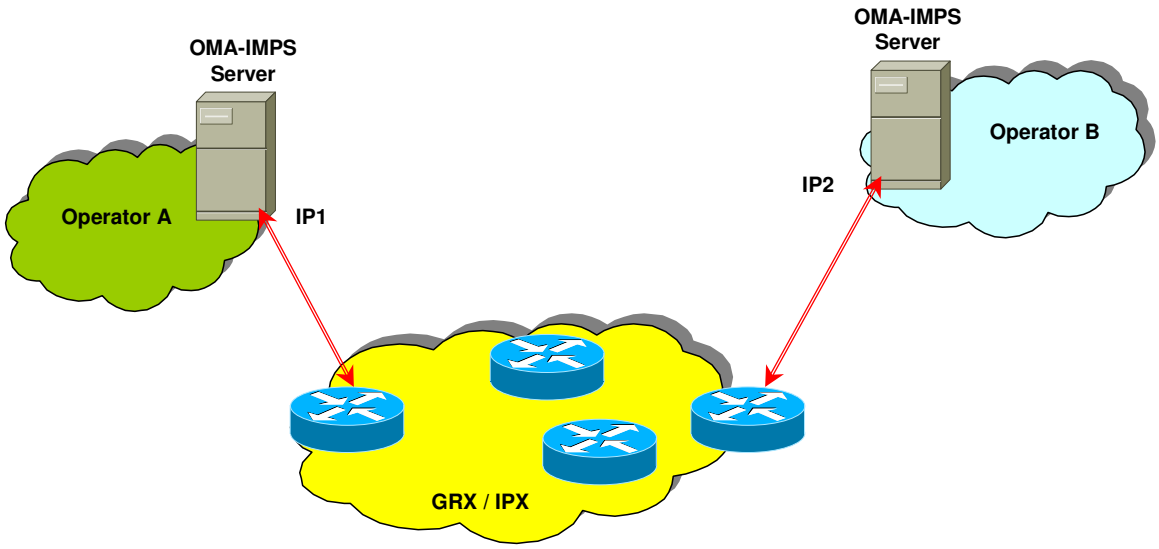

Figure 6: GRX/IPX IP connectivity.

Existing networks can be reused as much as possible instead of building separate networks for each and every new service. The preferred inter-PLMN network for IM is GRX/IPX, as it is already preferred network in e.g. GPRS roaming, MMS interworking and WLAN roaming. There is no need for new separate VPNs inside GRX/IPX for IM traffic, since IM traffic can co-exist with current GRX/IPX protocols, such as GTP and SMTP.

5.2 Security Issues

In order to maintain proper level of security within the inter-PLMN backbone certain requirements for IM operators and Inter-PLMN backbone providers should be taken into account. The same security aspects shall be applied as described in IR.34 [1].

OMA IMPS does not require any specific transport layer security mechanism, such as data integrity and confidentiality. However, current implementations using an IP network such as the Internet- are using security in the underlying transport layer in order to ensure the security of transmission in the underlying layers. The deployment of security is negotiated between the service providers through an offline configuration agreement .

There is another level of security that is implemented at the IMPS level. This security involves the mutual authentication between two IMPS servers using the IMPS Login procedure. This figure outlines the procedure. Refer to 7.4.6 for details:

Figure 7: Login procedure.

5.3 Service Related Issues

OMA-IMPS traffic doesn't have specific requirements in terms of QoS that is required to be provided by the Inter-PLMN backbone network, at least for first deployment. In addition some IM mechanisms implemented by OMA-IMPS, like content indirection (e.g. instead of including a very heavy payload in the IM, refer to an URL where the actual content is), should help to keep this requirement at a low level of priority.

DNS services as provided by the Inter-PLMN backbone network will be used by any IM service and OMA-IMPS in particular. Additionally, some other database services related to MSISDN storage, (MNP/ENUM) will be required by some use cases related to IM services [2]. See section 6.5 for details.

6 ADDRESSING AND ROUTING

6.1 User Addressing

Every OMA-IMPS user has one OMA-IMPS user identity that allows them to run IM services (send/receive IM, Presence related services, etc.). The standardized form of OMA-IMPS user address is:

Global-User-ID = "wv:" User-ID "@" Domain.

The User-ID either refers to the Internet style address or to a mobile number of the user. If the User-ID refers to the mobile number of the user, the user name always starts either with a digit or with a '+' sign. A user name referring to Internet-type address MAY not start with a '+' sign or digit.

The syntax of the User-ID is defined as follows:

User-ID = Mobile-Identity | Internet-Identity
Internet-Identity = *alpha
Mobile-Identity = (digit | "+") *digit
digit = "0" | "1" | "2" |"3" |"4" |"5" |"6" |"7" |"8" | "9"
alpha = Any non-control ASCII character (decimal 32 – 126, inclusive) except specials
specials = "/" | "@" | "+" | " " | TAB

When the User-ID refers to the mobile number address, the User-ID preceded with '+' sign refers to the international numbering in The International Public Telecommunication Numbering Plan.

Examples:
wv:Jonh.Smith@imps.com
wv:+1234567890@imps.com
wv:4567890@imps.com

- There isn't any restriction or requirement in the domain part of the user address, so each operator might use whatever naming scheme they wish and it is likely that the chosen schema is a human-friendly name like vodafone.com, sonera.fi, etc.

6.2 Service Addressing

The Service-ID in SSP is equivalent in the semantic role to the User-ID in CSP. The Service-ID in SSP uniquely identifies a Server. The syntax of Service-ID is defined as follows.

Service-ID = "wv:"@ Domain

Domain is a set of the IMPS entities that have the same Domain part in their IMPS addresses. The Domain is associated with one IMPS server (the unique access point) to which the IMPS service requests must be delivered if the addressed network entities refer to this Domain.

The Service-ID is used in the session establishment and other SSP management functions.

An example of the Service-ID is:
Service-ID: wv:@imps.com

6.3 Direct Interconnection scheme.

As a first approach, as defined in IMPS standards, interconnection should be managed in a one-to-one basis. As has been described in point 4, it is necessary to build at least two TCP connections and two logical Sessions between two IM operator domains in order to allow interoperability between them. Therefore, in a direct link scenario using GRX/IPX as the Inter-PLMN IP network all the routing information will be available to the OMA-IMPS servers on a one-to-one basis. Figure 9 shows how the routing information is recorded in a connection reuse mapping between TCP connections and IMPS sessions (e.g. a session can use both TCP connections):

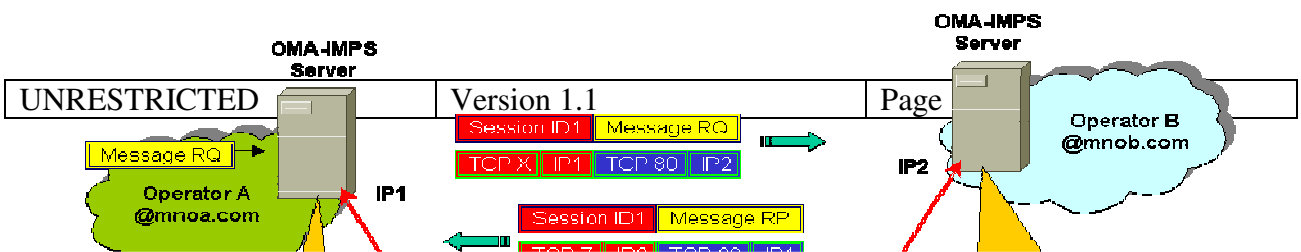

Figure 8: IMPS interconnectivity using GRX/IPX only for transport (No Hub). Case of TCP connections reuse.

Using GRX/IPX in this way means to give up the advantages that this network can provide to an IM service. In fact, in this implementation all the routing intelligence is kept at the OMA-IMPS servers, which have the drawback of the lack of scalability when the number of interconnections increases since the routing information is stored in each Server. Due to the protocol requirements, every OMA-IMPS server must keep two types of records, one related to the pair of Sessions built with the other domain and another one resolving IP address to reach that domain.

6.4 IMPS and the Domain Name Service-DNS.

As has been described earlier in the document, the direct interconnection model that IMPS demands is not viable in the longer term for a widely deployed IM service. Such static configuration in domain names to IP addresses mapping in the IMPS servers, leads to a lack of scalability and, flexibility e.g. when operators decide to reconfigure their IP address range assigned to services like IM.

A more flexible and scaleable solution involves the use of DNS records. The DNS solution doesn't mean that GRX/IPX DNS services are involved immediately, as it will be shown later in the document, but it is made clear that those DNS services are indeed the long-term solution.

In IMPS DNS services will be used by the IMPS server to find out the IP address of the point of contact, i.e. another IMPS server, associated to a target IM domain name. In detail, what this DNS should be able to resolve are the following items:

- Distinguishing the outbound traffic to a user belonging to another IMPS mobile domain, from IM traffic addressed to IM Internet fixed users (MSN, AOL, etc.).
- Distinguishing IM traffic towards a target domain, from other IP traffic addressed to the same domain.
- Resolving to the right IP address for each type of traffic.

Note that the storage of Session-ID in the IMPS server has been put aside of the requirements list, this is because it is out of the scope of DNS functionalities. This requirement will be addressed later in this document.

Interoperability solutions involving DNS services do not need to move immediately to such complex solutions in order to deal with GRX/IPX DNS constraints. It is better to describe an incremental solution based on the complex scenario.

6.4.1. Early/Rapid deployment.

Instant messaging interoperability will be based on bilateral agreements between mobile operators. Therefore, it is assumed that in the first stages of the IM service deployment, each operator will have a limited number of interoperability agreements in place. Under this scenario, the IMPS Direct Interconnection scheme is capable of taking on the system data requirements that must be provisioned to allow interconnection traffic.

However noting that it is worth putting in place a solution based on DNS from scratch, because the transition to the long-term solution based on GRX/IPX DNS will be much more straightforward.

The proposed solution for early deployment does not use GRX/IPX DNS services. Instead of that, a “private operator DNS server” is recommended. Following figure 10 outlines the solution;

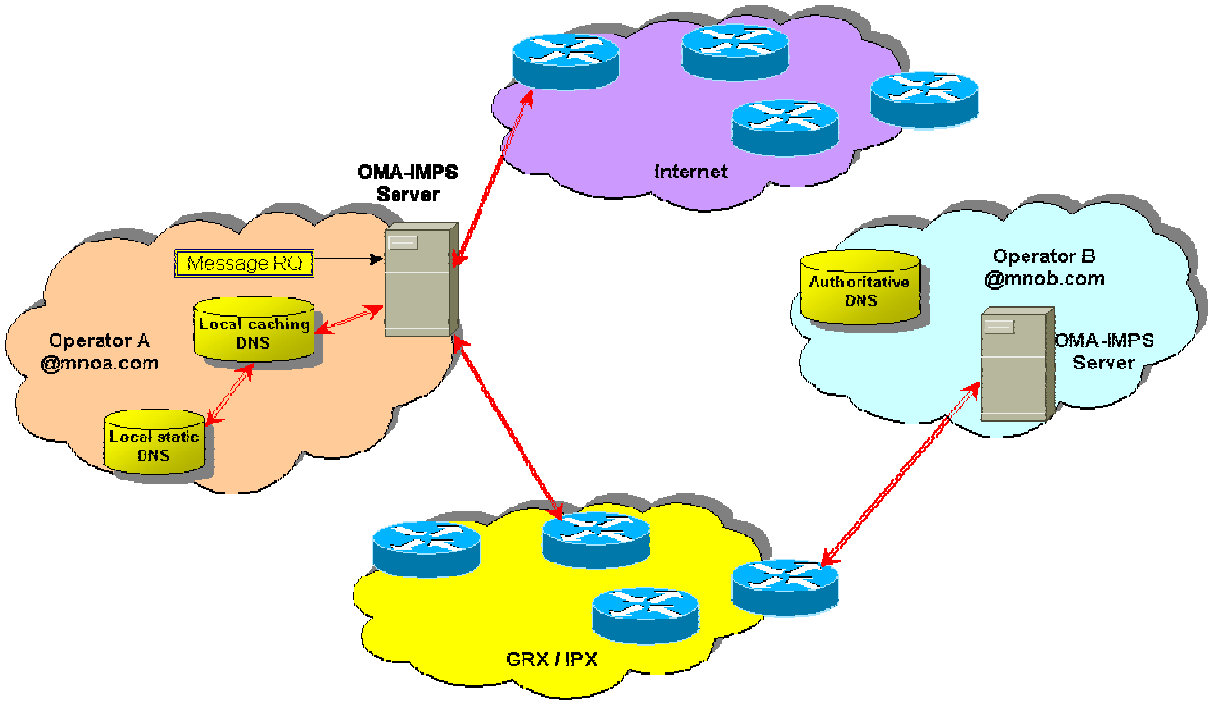

Figure 9: Interconnection architecture for rapid deployment.

It is assumed that mobile-to-mobile IM traffic will be routed to the Inter-PLMN backbone network, for the reasons outlined earlier and others that are described later. It is assumed that

in the mobile IM environment, IMPS protocol will be used. To be clear, interworking traffic to the Internet IM communities should be routed towards Internet backbone.

The private operator DNS server solution works as follows. Let's assume that the Operator A IMPS server is going to receive user request for IM interactions towards other users belonging to another IM domain. As described earlier in the document, the identity of the targeted user will look like:

wv:peter.smith@domainame

IMPS server now relies on its DNS server to resolve the target domain name. Note that there isn't any restriction regarding *domainame* syntax (operator.com, operator.fi, etc.).

The IMPS server first checks its local caching DNS, which is caching answers to previous DNS queries issued by the Resolver (in this case, the IMPS Server+Local DNS caching). If the required information is there, it is used.

If the required information is not in the local caching DNS, a query to the Local Static DNS server must be made. There are two possible methods for obtaining an answer to this query:

6.4.1.1. Option 1 – Terminal 'A' records

6.4.1.1.1. Description

This solution means that the required IP address of the target IMPS server is delivered, as shown in the example:

```
domainame.          IN  A    101.1.2.3
```

6.4.1.1.2. Advantages and Disadvantages

This solution reproduces the direct interconnection schemes outlined by IMPS. Therefore, it has the same lack of scalability and flexibility that the original scheme had. Besides, it requires manual labour in configuring this static information. It is, however, the simplest one.

It is assumed that distinguishing each type of traffic is done by bilateral agreement that provides unique domain names able to differentiate IM traffic from other traffic addressed to the same operator, as well as uniquely identify domain names belonging to Internet IM operators like MSN, AOL, etc.

6.4.1.2. Option 2 – Redirection NS records

6.4.1.2.1. Description

This is the solution that implements an actual "private operator DNS network". The answer received is an indicator that points to the target network. This redirection is realized using a single NS record. This NS record redirects the DNS Resolver to the right network by returning a new DNS server to query, as shown in the example:

```
domainame.          IN NS    dns1.domainame
dns1.domainame.       A    101.1.2.3
```

6.4.1.2.2. Advantages and Disadvantages

As has been shown in the example, the answer breaks the circular dependency and introduces the fact that the DNS server is under the domain being resolved. That means that, by some means, the IP addresses of those DNS servers must be available to the Local DNS Static server. That involves static configuration or access to special databases like e.g. IR 21 where those addresses would be stored. That makes this solution not as flexible as it could be, in terms of adaptation to those circumstances requiring IP address reconfiguration, but the amount of manual updates required, could be seen as reasonable.

It is assumed that distinguishing each type of traffic is done by bilateral agreement that provides unique domain names able to differentiate IM traffic from other traffic addressed to the same operator, as well as uniquely identify domain names belonging to Internet IM operators like MSN, AOL, etc.

6.4.1.3. Other considerations.

In the Option1 and Option 2 solutions above, it is down to the operators to provide themselves with the unique domain names that allow DNS to differentiate IM traffic from other traffic addressed to the same operator or IM traffic to others fixed Internet users. That means that the end user should be provisioned with an address like this:

wv:peter.smith@im.domainame

This solution seems much simpler, but its main drawback is that a "service-customized" domain name (using im) is created for each service the operator is offering, and the user provisioned with it. That is not best practice, but could be acceptable for early deployment.

Some IMPS vendors are arguing for a more "radical" approach in the user addressing for IM, for the sake of SMS continuity and fast provisioning of the service, they claim that only MSISDN should be used as the user identity for IM (only MSISDN, not wv:MSISDN@domain, which is an standard and accepted identity). If this were widely accepted, it would not require any DNS involvement at all and only require an ENUM/MNP solution as discussed in section 6.5. At this point in time such a solution cannot be recommended. It is out of the IMPS standard and will require agreement between all operators to be feasible.

6.4.2. Future Phases.

The scalability of the "private DNS server network" is limited. So, as the IM service interconnection traffic grows, a more dynamic and hierarchical solution is required. That leads to the GRX/IPX DNS network architecture. The problems to face for that solution are the same that have been described before, so, the solutions are more or less similar. However, there are some considerations that must be taken into account in order to fit the requirement that GRX/IPX network has regarding domain names.

There is an incompatibility of user and domain addressing schemes. GRX/IPX DNS cannot handle *domainame* because it is intended to be completely separate of Internet access. This problem is the same that has been identified and addressed in IR.65 [4] and the solutions given here are borrowed from that document.

The recommended solution from that document is rewriting the domain name utilizing NAPTR domain rewrite rules in the local operator DNS. This NAPTR can be merged with the one used in the Option 3 outlined above. This NAPTR could look like this:

```
wv.domainame.
;;     order  pref  flags  service    regexp  replacement
IN NAPTR 100   50   "a"    "http+I2R"   "im.domainame.mnc123.mcc456.3gppnetwork.org
```

Assuming that NAPTR, the solution looks as follows:

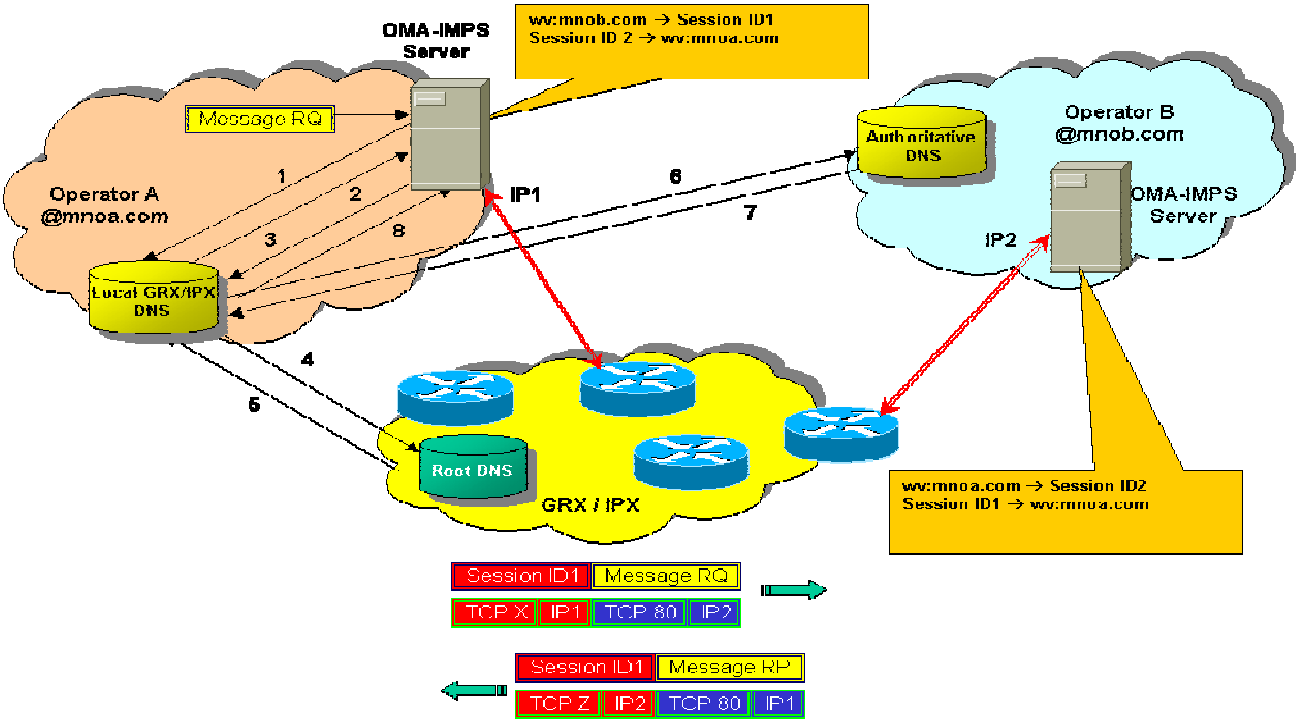

Figure 10: Next phases architecture.

1. The Resolver (e.g. an OMA-IMPS server trying to find out the IP address of its counterpart OMA-IMPS server in order to send a message) sends a query for the hostname (e.g. wv.domainame) for which it wants the IP address, to its local caching DNS server.

2. The local DNS rewrites the hostname using NAPTR records. The new format is compliant with the schema supported by GRX/IPX DNS framework (e.g. wv.domainame → domainame.mnc123.mcc456.3gppnetwork.org).

3. Using the new hostname format, the Resolver requests again to the local DNS server.

4. The local caching DNS server checks to see if it has the answer to the query in its cache. If it does it answers immediately (with message 8). Otherwise, it forwards the query on to the Root DNS server.

5. The Root DNS server returns a referral to the DNS server, which is authoritative for the queried domain name of the hostname (e.g. returns the authoritative server for "mnc123.mcc456.3gppnetwork.org").

6. The local caching DNS server caches the response for a specified amount of time (specified by the root DNS server) and then resends the query but to the authoritative DNS server as specified by the Root DNS server.

7. The Authoritative DNS server responds to the query with the address of the hostname (or responds with a hostname, if a reverse lookup is being performed) to the Local Caching server in the requesting network.

8. The Local Caching Server caches the response for a specified amount of time (specified by the authoritative server) and forwards it on to the Resolver.

With the learned IP address, originator OMA-IMPS server only has to use the well-known ports in order to create the right SSP message.

With this solution, scalability issues have been alleviated, but every OMA-IMPS server has to maintain individual records for each counterpart in order to store Session ID pairs established with them. Therefore, in some way, scalability concerns still remain.

Overcoming this scalability drawback leads to a more long term solution using OMA-IMPS hubs. The figure shows how the solution could look like:

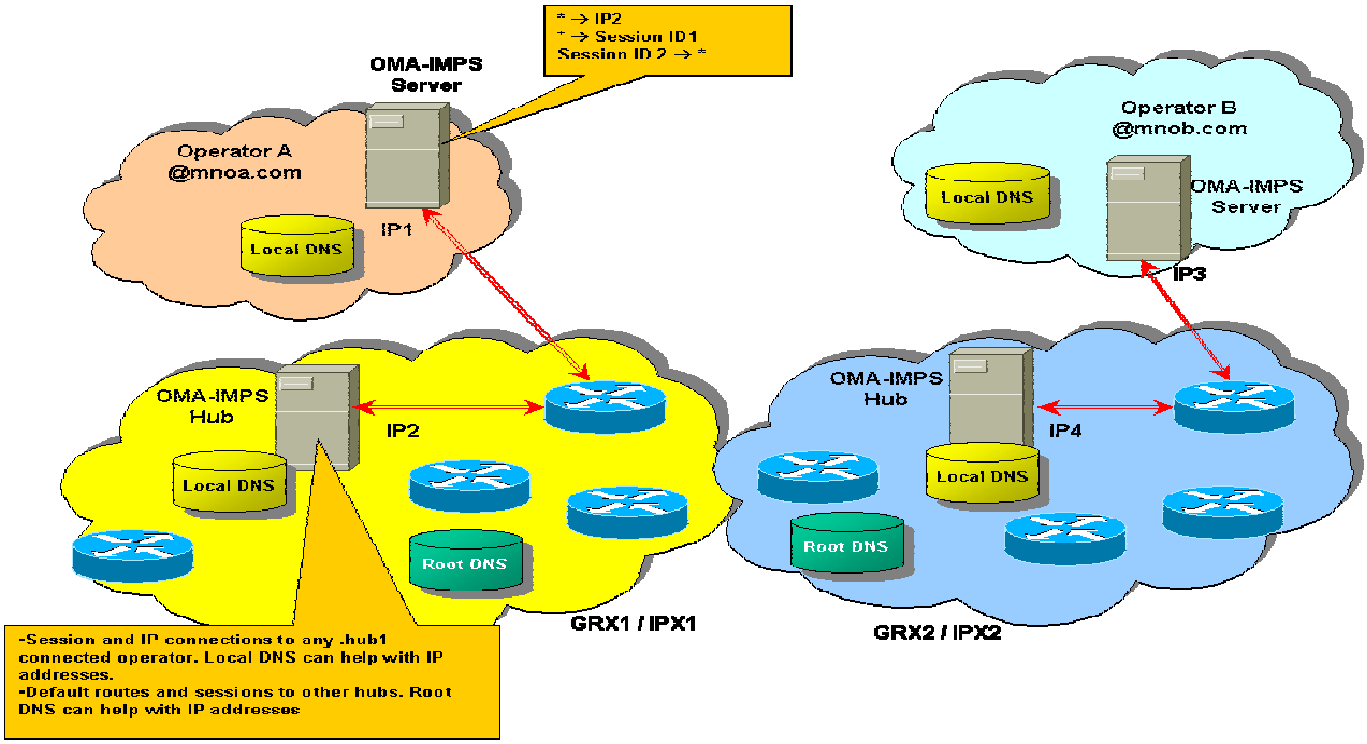

Figure 11: Hub architecture.

In this solution, every OMA-IMPS server is linked to an OMA-IMPS operator hub. Those hubs can be part of the GRX/IPX network or operated by a 3rd party provider, but connected to GRX/IPX network.

Every individual OMA-IMPS server will now establish the pair of TCP connection and sessions towards the hub. Notice that the routing table is now very simple in every MNO, it only has to point to the hub. It is the hub's responsibility to find the right server to contact with. The hub might have static routing tables for directly connected OMA-IMPS servers (those belonging to hub 1) or might use Local DNS services for solving addresses, using the solutions pointed out above (NAPTR rewriting, SRV RR, etc). When the required domain belongs to another hub (e.g.. hub 2), hub 1 must found out routes to those other hubs, which can be eiether statically recorded or dynamically discovered using Root DNS services.

In general, IMPS Hubs are high capacity IMPS servers. Therefore, they must implement all the features that have been defined for IMPS server in the specs. Some additional features, like IM protocol interworking (e.g. translation between IM protocols) could be enabled.

6.5 Other interworking services.

Although IM services are pure packet data services, there are some IM use cases that requires other interworking interactions. For instance, a user can use MSISDN as a valid identity for a user, if he/she doesn't have another identity available. This identity must be sent to the right IM Server in order to do ID checking and service authorisation. In addition a proper IM UserID should be provided backwards in order to start IM interactions. See [2] for details of the related use cases, but following figure 13 summarises the case of adding a user to a contact list using the MSISDN as a valid identity:

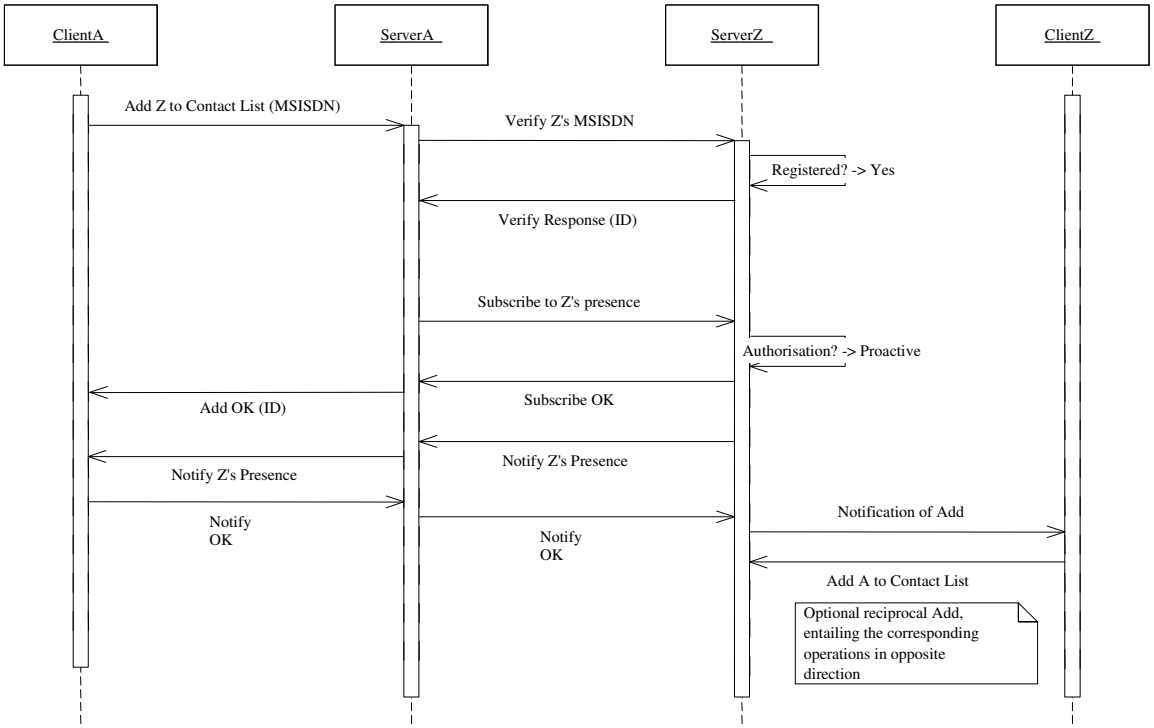

Figure 12: Adding a user to a contact list by MSISDN.

It is obvious from figure 13 that IMPS Server A must map non-routable MSIDSN address to receiver IMPS Server Z. This problem is identical to the one that is faced in IR.52 [5] Section 6 and similar solutions can be applied.

To solve this problem, ENUM is identified as the long-term solution for that mapping, but in the short term a solution of leveraging a MAP query for an IMSI address is a mechanism for creating a IMPS Server Z address form a dialed MSIDSN number. This MAP interface has been a requirement to IMPS vendors in all Personal IM GSMA documents.

It should be noted that it is the role of the originating operator and its IMPS Server to handle outgoing interworking related messages appropriately therefore the recipient operator should not receive messages it cannot be forwarded to an addressed recipient.

That means that the requirement for number portability support is concrete in uses cases like this, since a number of countries already use MNP for calls and SMS. This means that IM systems must be capable of doing this.

The following list describes different methods in order of simplicity, to implement address discovery in IMPS interworking:

- Static table mapping: A receiver's MSISDN prefix is turned directly into a receiving IMPS Server IP address (e.g. +35840 <=> 194.251.253.74). This *will not cope with MNP*. Note that this is a non-scalable solution, thus it should really be used only as a

quick'n'dirty interim address discovery solution, until more advanced solutions are implemented.

- IMSI based look-up: First get the IMSI based on receiver's MSISDN from HLR using a MAP query, then resolve MNC + MCC from IMSI, finally use DNS query to resolve the IMPS Server IP address (e.g.mnc111.mcc222.3gppnetwork.org. <=> 194.251.253.74). This option *supports MNP, but requires DNS support* (*can be established within a short period of time if the GRX/IPX DNS is used)*. Please refer to Figure 14

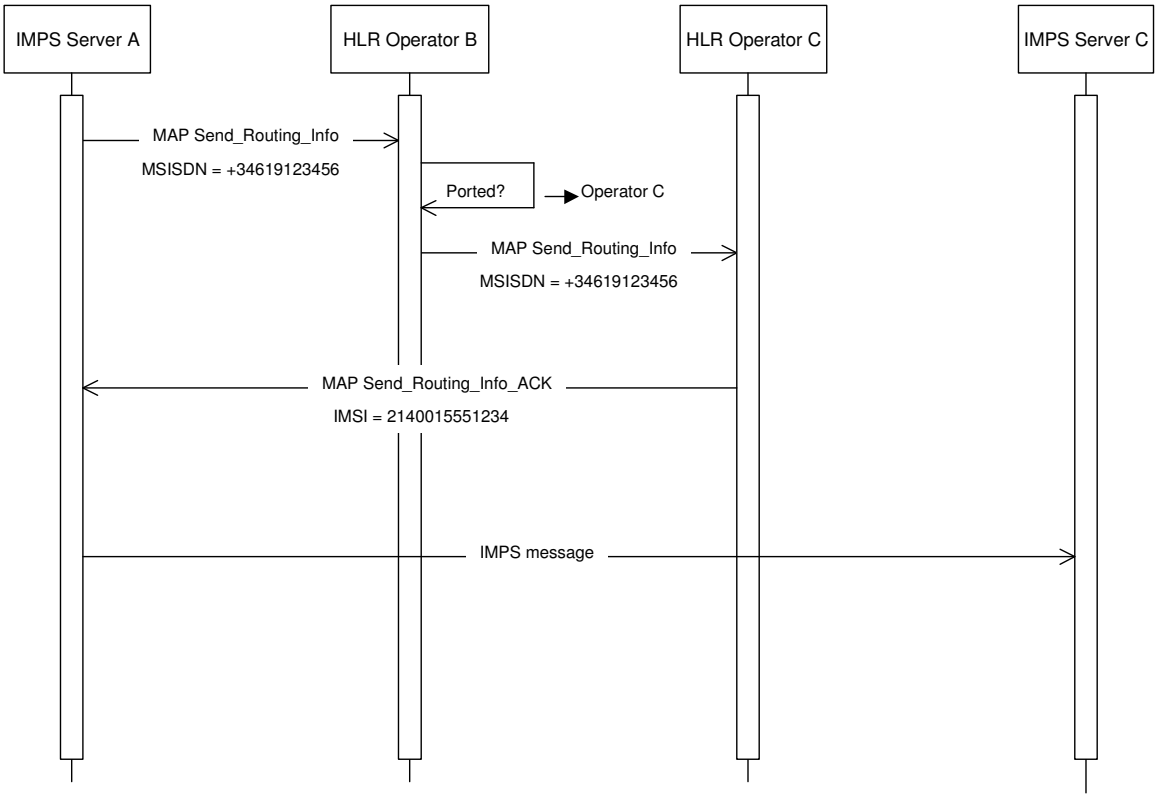

Figure 13: MAP based solution for MNP.

- ENUM based look-up: Please refer to [5]. If the GRX/IPX provider is used, private ENUM can be established within a short period of time. See IR.67 [6] for details about MNP and ENUM possible schemes.

7 PROTOCOL COMPLIANCE

From a functional perspective, this specification uses the IMPS – Server to Server Protocol for interworking, i.e. it defines what IMPS functional elements (messages and primitives) are required in order to implement the Use Cases outlined in [2]. The required IMPS version for those Use Cases is IMPS v1.2.

7.1 Standards Compliancy Matrix

Table 1 below details which standards with which the IM Interworking function should comply. The degree of compliancy with the standards shall be based on what is required for interoperability. The specific transactions which the IM Interworking Function must be compliant with are explicitly detailed in this document in the Annex. The Annex contains only a snapshot of the OMA standard protocol for information and it should be noted that the OMA specifications themselves should be used as the authoritative reference.

Table 1 Standard Compliancy

Doc	Standard	Functional Description	Reference alias
	OMA-WV-SSP_SCR-V1_1-20021001-A	OMA SSP - Server Protocol Static Conformance Requirement v 1.1	[SSP SCR] v 1.1
	OMA-IMPS-WV-SSP_SCR-v1_2-20030221-C	OMA WV 055 SSP - Server-Server Protocol Static Conformance Requirement v 1.2	[SSP SCR] v 1.2
	OMA-WV-SSP-V1_1-20021001-A	OMA WV 032 Server - Server Protocol Semantics v 1.1	[SSP]
	OMA-IMPS-WV-CSP-V1_2-20050125-A	OMA WV 042 Client-Server Protocol Session and Transaction v 1.2	[FeaFun]
	OMA-IMPS-WV-CSP-WBXML-V1_2-20050125-A	OMA WV 047 Client-Server Protocol Binary XML Definition and Examples v 1.2	[CSP WBXML]

The minimum requirement is for support of WV-SSP version 1.2.

8 CONCLUSION

Successful interworking is crucial for the success of IM using IMPS, thus it must be handled as soon as possible. The Preferred way is to follow common GSM Association guidelines right from the start rather than try to implement a number of different solutions and later see what solution becomes the de facto standard.

In an early deployment, GRX/IPX DNS services are not necessarily required there is an option of building a “private operator DNS network” that implements any of the following can solve the problem:

- Option 1: terminal A records

- Option 2: Referral NS records
- {Future Phases} DDS and NAPTR records

The third one for future phases is the most flexible and this can be merged with the required NAPTR RR necessary to accommodate domain names to the constraints (.3gppnetwork.org) in the GRX/IPX DNS

For a future proof deployment an operator should take account that IMPS is intended to interwork on a direct link basis. This means that every IMPS server builds two TCP connections and two logical connections (Sessions) with any counterpart and stores IP addresses and Session Ids of each of the other IMPS domains it has relations with. This leads to a problem of scalability that can be solved using GRX/IPX DNS services.

In IMPS, DNS services will be used by the IMPS server to find out the IP address of the point of contact, another IMPS server by association to a target IM domain name. In summary, what this DNS should be able to resolve is the following:

- Distinguishing outbound traffic to a user belonging to another IMPS mobile domain, from traffic addressed to IM Internet fixed users (MSN, AOL, etc.).
- Distinguishing IM traffic towards a target domain from other IP traffic, addressed to the same domain.
- Resolving to the right IP address for each type of traffic.

GRX/IPX ENUM-MNP services may be required in the future as many IM use cases deal with MSISDN as the valid identity of the user. In the interim deployment phase, tactical use of MAP solutions is recommended in order to solve those interactions.

9 REFERENCES

1. PRD IR.34 Inter-PLMN Backbone Guidelines
2. IPIAG Gen Doc 002_06r3. Personal IM Use Case (IM Phase 1)
3. OMA IMPS specification. OMA-ERELD-IMPS-V1_2-20050801-A.pdf
4. PRD IR.65 IMS Roaming & Interworking Guidelines/
5. PRD IR.52 MMS Interworking Guidelines
6. PRD IR.67 DNS Guidelines for Operators

ANNEX A IMPS-SSP SUPPORT

This section contains a snapshot of the OMA standards to which interworking protocols should comply. This information is provided for information and the OMA specifications themselves should be used as the authoritative references.

A.1 Version

The minimum requirement is for support of WV-SSP version 1.2

A.2 Transport layer

Transport layer is using HTTP/HTTPS protocol over TCP connection. Refer to OMA-IMPS standard document for details.

The following extra headers are required in the HTTP protocol:

A.2.1 x-wv-transactionid

The x-wv-transactionid header extension must be used to carry the transaction identifier in all HTTP / HTTPS POST requests:

```
header      = x-wv-transactionid ":"  header-value  CRLF
header-value  =  1*alphanum
alphanum = alpha | digit | "_"
```

A.2.2 x-wv-sessionid

The x-wv-sessionid extension must be used to carry the session identifier in all HTTP / HTTPS POST requests if session is established:

```
header      = x-wv-sessionid ":"  header-value  CRLF
header-value  =  1*alphanum
alphanum = alpha | digit | "_"
```

A.2.3 x-fw-alias

The x-fw-alias extension carries the alias of the requestor, if present.

```
header      = x-fw-alias ":"  header-value  CRLF
header-value  =  1*alphanum
alphanum = alpha | digit | "_"
```

A.3 Transaction management

The transaction management defines the necessary common information elements in the service requests and service responses at transaction level, regulates the behaviour in the transaction flows, and handles the exception and error conditions at the transaction level.

Table 2 Parameter Legend

Req	Description
M	Indicates that the parameter is mandatory
O	Indicates that the parameter is optional
C	Conditional. Indicates that the parameter appears only if certain conditions are true.

- Meta-Information

SSP service requests must contain the meta-information as defined in Table 3.

Table 3 Meta-Information Information Elements

Information Element	Req	Type	Description
Client-Originated	M	Boolean	Indicates whether the request is originated from the client ("True") or from the service element ("False"). The IM Interworking Function sets that value to False.
Session-ID	M	String	Identifies the session managed by the Provider Server.
Transaction-ID	M	String	Identifies the transaction originated from the transaction initiator (either requestor server, or provider server).
Service-ID	M	String	Identifies the initiator domain (and the service element if needed).
User-ID	C	String	Identifies the user represented by the requestor server domain. It is present if the request is originated from a client.
Client-ID	O	String	Identifies the Client-ID of the user. It optionally presents whether the request is originated from a client. The IM Interworking Function does not need to use that parameter.

The Session-ID is unique for each session at the Provider Server.
The Transaction-ID is unique for each transaction originated from the server which initiates the transaction.
An SSP service response in a two-way transaction must contain the same Session-ID and Transaction-ID as those in the service request.

- Status Primitive

The status primitive in the service response is defined in Table 4.

Table 4 Information Element in the Status Primitive

Information Element	Req	Type	Description
Session-ID	M	String	Identifies the session. It must be consistent with the Session-ID in Meta-Information within the request.
Transaction-ID	M	String	Identified the transaction. It must be

Information Element	Req	Type	Description
			consistent with the Transaction-ID in Meta-Information within the request.
Status code	M	String	Status code of the processing result.
Status description	O	String	Textual description of the status.

DTD:

```
<!-- Status is contained in every normal response and it may also be a separate message or can be sent in error cases of login phase messages -->
<!ELEMENT Status (StatusDescription?)>
<!ATTLIST Status
            code      CDATA  #REQUIRED>
<!ELEMENT StatusDescription (#PCDATA)>
```

Example:

```
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="123456@pgsm.hu">
 <Transaction mode="Response" transactionID="420042">
  <Status code="400">
   <StatusDescription>Bad request</StatusDescription>
  </Status>
 </Transaction>
 <Transaction mode="Response" transactionID="420043">
  <Status code="200"/>
 </Transaction>
</Session>
</WV-SSP-Message>
```

A.4 Session management

SSP session management includes session establishment, session termination and session maintenance.

Basic Steps for Session Establishment:

1. Server A originates a connection 1 to Server B based on its own registration record about Server B, containing {A-Service-ID, A-secret-token} tuple.
2. Server B looks for {A-Service-ID} in its own registration record. If it is not found, Server B terminates the connection.
3. Server B initiates connection 2 to the Server A containing {B-Service-ID, B-secret-token}.
4. Server A looks for {B-Service-ID} in its own registration record. If it is not found, Server A terminates the connection.

5. Server A sends the LoginRequest to Server B through connection 1, containing {A-Service-ID, A-password-digest}. The "A-password-digest" is generated with A-password and B-secret-token based on the common digest schema in the registration record.
6. Server B sends the LoginRequest to Server A through connection 2, containing {B-Service-ID, B-password-digest}. The "B-password-digest" is generated with B-password and A-secret-token based on the common digest schema in the registration record.
7. Server B verifies the A-password-digest. If verification fails, it terminates the connection.
8. Server B responds to Server A with the LoginResponse through connection 2, containing the status of the transaction and the new session information maintained by Server B.
9. Server A verifies the B-password-digest. If verification fails, it terminates the connection.
10. Server A responds to Server B with the LoginResponse through connection 1, containing the status of the transaction and the new session information maintained by Server A. The LoginResponse may contain an optional list of Redirect (Host) Names, also known as the Redirect List.
11. The secret-token is a random string generated by the connection originator at each server.
12. After step 10 succeeds, two domains are authenticated with each other. The session pair between Server A and Server B is established with trust over two connections, i.e. the connection pair. The connection pair (1 and 2) between A-Host-ID and B-Host-ID is known as "Master Connection Pair". Session establishment is achieved through a "Login" transaction.

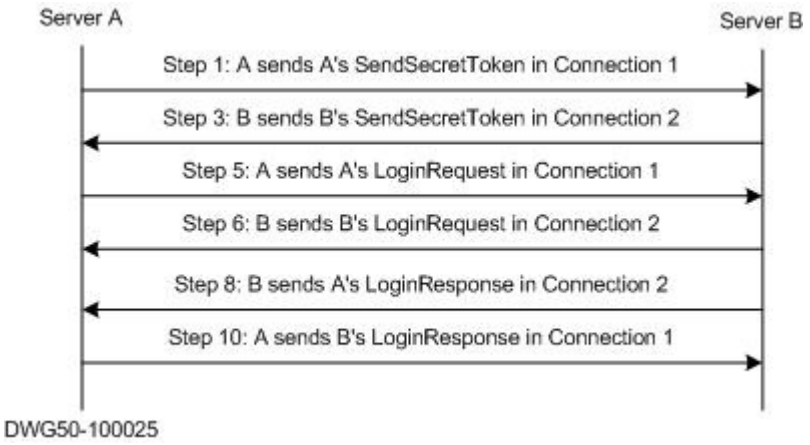

Figure 14 The Login Transaction

A.4.1 Primitives

A.4.1.1 The "SendSecretToken" Primitive

The "SendSecretToken" primitive is issued by the requestor server to send the secret token for the provider server as the first step of the CALLBACK connection establishment.

Table 5 Information Element in the SendSecrectToken primitive

Information Element	Req	Type	Description
Message-Type	M	SendSecretToken	Message identifier
Transaction-ID	M	String	Identifies the transaction originated from the initiating provider server
Service-ID	M	String	Identifies the requestor server
Protocol	M	"WV-SSP"	SSP protocol
Protocol-Version	M	"1.2"	SSP protocol version
SecretToken	M	String	Secret token originated by the requestor

DTD:

```
<!-- SendSecretToken is a special request-like message without session and MetaInfo -->
<!ELEMENT SendSecretToken (SecretToken)>
<!ATTLIST SendSecretToken
        serviceID               CDATA  #REQUIRED
        protocol                CDATA  #FIXED  "WV-SSP"
        protocolVersion         CDATA  #FIXED  "1.2">
<!ELEMENT SecretToken           (#PCDATA)>
<!ATTLIST SecretToken
        encoding                CDATA  "base64">
```

Example:

```
?>xml version="1.0" encoding="UTF-8<?"
```

```
>WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2<"
>SetupTransaction mode="Request" transactionID="8165<"
>SendSecretToken serviceID="wv:@mnoz.com" protocol="WV-SSP" protocolVersion="1.2<"
>SecretToken encoding="base64">Y2U2MGMxMTQ5Nzlh</SecretToken<
/>SendSecretToken<
/>SetupTransaction<
/>WV-SSP-Message<
```

A.4.1.2 The "LoginRequest"" Primitive

The LoginRequest primitive is issued from the requestor server to create a new session with the provider server. The LoginRequest primitive specifies initial status of the requestor server. The LoginRequest primitive MAY also contain the time-to-live attribute, which specifies the time interval for session/connection expiration. If time-to-live attribute is omitted, the requestor server requests an infinite session or connection until the service agreement expires.

Table 6 Information Element in the LoginRequest Primitive

Information Element	Req	Type	Description
Message-Type	M	LoginRequest	Message identifier
Session-ID	C	String	Identifies the session. It is present when creating additional redirect connection pairs within the existing session
Transaction-ID	M	String	Identifies the transaction. It must be consistent with the Transaction-ID in the SendSecretToken originated from the provider server
Service-ID	M	String	Identifies the requestor server
Password-Digest	M	String	The password digest generated with password and secret token, based on a common digest-schema (MD5 or SHA). IM Interworking Function may support both MD5 and SHA schemas. Password digest is either MD5 (Password A || SecretToken B) or MD5 (SecretToken B || Password A). Base64 encoding may be used for the Password-Digest.
Time-To-Live	O	Integer in Seconds	Interval for a valid session/connection before expiration. If omitted, the requestor server requests an infinite session/connection. This value is used in the KeepAlive transaction.

DTD

```
<!-- LoginRequest is a special response-like message without Session and MetaInfo -->
<!ELEMENT LoginRequest (PasswordDigest)>
```

```
<!ATTLIST LoginRequest
        serviceID               CDATA  #REQUIRED
        redirectHostId          CDATA  #IMPLIED>
        timeToLive              CDATA  #IMPLIED>
<!ELEMENT PasswordDigest        (#PCDATA)>
<!ATTLIST PasswordDigest
        encoding                CDATA  "base64">
```

Example:

```
?>xml version="1.0" encoding="UTF-8<?"
>WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2<"
>SetupTransaction mode="Response" transactionID="26388" timeToLive="30<"
>LoginRequest serviceID="wv:@mnoz.com<"
>PasswordDigest encoding="base64">iKmSfqOxaPd9yze4DxaGlw==</PasswordDigest<
/>LoginRequest<
/>SetupTransaction<
/>WV-SSP-Message<
```

A.4.1.3 The "LoginResponse"" Primitive

The LoginResponse primitive is issued from the provider server to accept session creation with the requestor server. In the response, the provider server may specify the time-to-live of the current session. This time-to-live differs from that of the LoginRequest from the requestor server.

Table 7 Information Element in the LoginResponsePrimitive

Information Element	Req	Type	Description
Message-Type	M	LoginResponse	Message identifier
Status-Info	M	Structure of Status-Primitive	The necessary status information in a service response
Time-To-Live	O	Integer in Seconds	Interval for a valid session/connection before expiration. This time may be any numeric value larger than zero
List-of-Hosts	O	Structure	“Redirect” list, which indicates the actual connection addresses in its own domain

DTD

```
<!-- LoginResponse -->
<!ELEMENT LoginResponse (Status, HostsList)>
<!ATTLIST LoginResponse
        sessionID               CDATA  #IMPLIED
        timeToLive              CDATA  #IMPLIED>
```

```
<!-- HostsList -->
<!ELEMENT HostsList (redirectHostId*)>
<!ELEMENT redirectHostId (#PCDATA)>
```

Example:

```
?>xml version="1.0" encoding="UTF-8<?"
>WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2<"
>SetupTransaction mode="Response" transactionID="26388<"
>LoginResponse sessionID="5430<"
>Status code="200<"
>StatusDescription>Success</StatusDescription<
/>Status<
>HostsList</
/>LoginResponse<
/>SetupTransaction<
/>WV-SSP-Message<
```

A.4.1.4 The "LogoutRequest" Primitive

The LogoutRequest primitive allows the requestor server to close the session with the provider server.

Table 8 Information Element in the LogoutRequest Primitive

Information Element	Req	Type	Description
Message-Type	M	LogoutRequest	Message identifier
Session-ID	M	String	Identifies the session
Transaction-ID	M	String	Identifies the transaction

DTD

```
<!-- LogoutRequest -->
<!ELEMENT LogoutRequest EMPTY>
```

Example:

```
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="123456@someone.za">
 <Transaction mode="Request" transactionID="5006">
  <LogoutRequest/>
 </Transaction>
</Session>
</WV-SSP-Message>
```

A.5.1 The "Disconnect" Primitive

The Disconnect primitive allows the provider server to indicate that it accepts the LogoutRequest from the requestor server and terminates the session.
If the provider server does not receive any session-maintenance update within the time-to-live interval from requestor server, the provider server terminates this session, by sending the Disconnect message to the requestor server.

Table 9 Information Element in the Disconnect Primitive

Information Element	Req	Type	Description
Message-Type	M	Disconnect	Message identifier
Session-ID	C	String	Identifies the session. Present if the provider server initiates the Disconnect
Transaction-ID	C	String	Identifies the transaction. Present if the provider server initiates the Disconnect
Status-Info	C	Structure of Status-Primitive	The status information. Present if the requestor server Logout

DTD:

```
<!-- Disconnect -->
<!ELEMENT Disconnect (Status?)>
```

Example:

```
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="123456@operator.hu">
 <Transaction mode="Response" transactionID="5006">
  <Disconnect>
   <Status code="200"/>
  </Disconnect>
 </Transaction>
</Session>
</WV-SSP-Message>
```

A.5.1.1 The "KeepAliveRequest" Primitive

The “KeepAliveRequest“primitive allows the requestor server to maintain the session and update the time-to-live interval with the provider server. The session maintenance is performed over all connections used by this session, thus covering the connection maintenance for each connection. The TTL may have different values for different connections.

Table 10 Information Element in the KeepAliveRequest Primitive

Information Element	Req	Type	Description
Message-Type	M	KeepAliveRequest	Message identifier
Session-ID	M	String	Identifies the session
Transaction-ID	M	String	Identifies the transaction
Time-to-live	O	Integer in Seconds	Indicates the time-to-live of the session over this connection

DTD:

```
<!-- KeepAliveRequest -->
<!ELEMENT KeepAliveRequest EMPTY >
<!ATTLIST KeepAliveRequest
```

```
        timeToLive                  CDATA  #IMPLIED>
```

Example:

```
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="123456@someone.za">
 <Transaction mode="Request" transactionID="5003">
  <KeepAliveRequest timeToLive="3600"/>
 </Transaction>
</Session>
</WV-SSP-Message>
```

A.5.1.2 The "KeepAliveResponse" Primitive

The KeepAliveResponse primitive allows the provider server to maintain the session and update the time-to-live interval with the requestor server. The session maintenance is performed over all connections used by this session, thus covering the connection maintenance for each connection. The TTL may have different value for different connection.

Table 11 Information Element in the KeepAliveResponse Primitive

Information Element	Req	Type	Description
Message-Type	M	KeepAliveResponse	Message identifier
Status-Info	M	Structure of Status-Primitive	The status information
Time-to-live	O	Integer in Seconds	Indicates the time-to-live of the session over this connection.

DTD:

```
<!-- KeepAliveResponse -->
<!ELEMENT KeepAliveResponse (Status)>
<!ATTLIST KeepAliveResponse
        timeToLive                  CDATA  #IMPLIED>
```

Example:

```
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="123456@operator.hu">
 <Transaction mode="Response" transactionID="5003">
  <KeepAliveResponse timeToLive="3600">
   <Status code="200"/>
  </KeepAliveResponse>
 </Transaction>
</Session>
</WV-SSP-Message>
```

A.5 IMPS-SSP Supported Transactions.

A.5.1 Contact Management and Presence Transactions and Primitives

A.5.1.1 Subscribe Transaction

Figure 15 The "Subscribe" Transaction

The subscription for obtaining the notification about changes of the presence information is accomplished through a “Subscribe” transaction.

The requestor server sends a SubscribeRequest request to the provider server to subscribe to notifications about changes in the presence information of specific publishing users. The provider server returns a Status message indicating that the provider server has accepted and processed the request.

If subscription succeeds, the requestor server immediately receives the current presence information through a “PresenceNotification” transaction. The requestor server also receives notification for any further presence information changes.

Table 12 Information Element in the KeepAliveResponse Primitive

Primitive	Direction
SubscribeRequest	Requestor Server → Provider Server
Status	Requestor Server ← Provider Server

A.5.1.2 The “SubscribeRequest” Primitive

The SubscribeRequest primitive is used to create subscriptions to obtain notifications about changes of PRESENCE INFORMATION and attributes of other PRINCIPALS. The scope of subscription is a single user.

Table 14 Information elements in SubscribeRequest Primitive

Information Element	Req	Type	Description
Message-Type	M	SubscribeRequest	Message identifier
Meta-Information	M	Structure of Meta-Information	The meta-information
User-ID-List	C	Structure	Identifies the IM users to be

			subscribed. Only one user is supported by the IM Interworking Function
Presence-Attribute-List	O	Structure	A list of presence attributes to which watchers are subscribed. An empty list or missing list indicates all presence attributes are desired
Auto-Subscribe	M	Boolean	A 'No' value is set by the IM Interworking Function

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="156">
<Transaction mode="Request"
transactionID="d14357905db4A1AFWQ9pD3jUe5TbLA1ASubscribe4fd40b3578f00
03404011441506590194090014502464 60f2a4a8f00">
<SubscribeRequest>
<MetaInfo clientOriginated="Yes">
<Requestor serviceID="wv:@mnoz.com">
<User userID="wv:i0524412331@mnoz.com"/>
</Requestor>
</MetaInfo>
<UserID userID="wv:user00100044@mnoz.com"/>
<AttributeList>
<PresenceSubList xmlns="http://www.openmobilealliance.org/DTD/WV-PA1.2" xmlns:Ext="http://www.fol
lowap.com/DTD/FW-PA2.0">
<OnlineStatus/>
<UserAvailability/>
<StatusText/>
<Ext:UI/>
<Alias/>
</PresenceSubList>
</AttributeList>
<AutoSubscribe>No</AutoSubscribe>
</SubscribeRequest>
</Transaction>
</Session>
</WV-SSP-Message>
```

A.5.1.3 Unsubscribe Transaction

Figure 16 The "Unsubscribe" Transaction

The cancellation of a current subscription is accomplished using an “Unsubscribe” transaction. The provider server returns a Status message indicating that the provider server has accepted and processed the request.

Table 15 Primitive Directions for Unsubscribe Transaction

Primitive	Direction
UnsubscribeRequest	Requestor Server → Provider Server
Status	Requestor Server ← Provider Server

A.5.1.4 The “UnsubscribeRequest” Primitive

The UnsubscribeRequest primitive is used to cancel the current subscription.

Table 16 Information elements in UnsubscribeRequest Primitive

Information Element	Req	Type	Description
Message-Type	M	UnsubscribeRequest	Message identifier
Meta-Information	M	Structure of Meta-Information	The meta-information
User-ID-List	C	Structure	Identifies the IM users to be unsubscribed. Only one user is supported
Contact-List-ID-List	O	Structure	Identifies the set of users

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="7474">
<Transaction mode="Request"
transactionID="6528b2fbaa43A1AFWQ9pD3jUe5TbLA1AUnsubscribead2eed2838e2007640115124256302120
14300021024e85c8ba0fb">
<UnsubscribeRequest>
<MetaInfo clientOriginated="Yes">
<Requestor serviceID="wv:@mnoz.com">
<User userID=" wv:i0524412331@mnoz.com "/>
</Requestor>
</MetaInfo>
<UserID userID=" wv:user00100044@mnoz.com "/>
```

```
</UnsubscribeRequest>
</Transaction>
```

A.5.1.5 Presence Notification Transaction

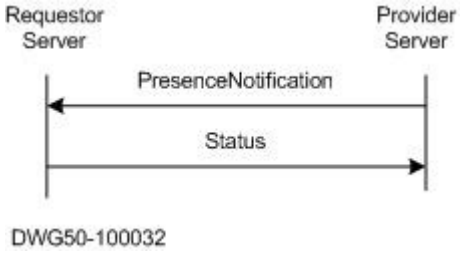

Figure 17 The "PresenceNotification" Transaction

The requestor server is informed of the change in presence information through a "PresenceNotification" transaction, originated by the provider server.

Table 17 Primitive Directions for PresenceNotification Transaction

Primitive	Direction
PresenceNotification	Requestor Server ← Provider Server
Status	Requestor Server → Provider Server

A.5.1.6 The "PresenceNotification" Primitive

The PresenceNotification primitive allows the provider server to send notifications about changes of presence information to the requestor server.

Table 18 Information elements in PresenceNotification Primitive

Information Element	Req	Type	Description
Message-Type	M	PresenceNotification	Message identifier
Meta-Information	M	Structure of Meta-Information	Meta-information
Subscribing-User-ID-List	M	Structure	Identifies the users who subscribed to the presence change. Only one user is supported
Presence-Value-List	M	Structure	List of User IDs and corresponding presence values

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="156">
<Transaction mode="Request" transactionID="45c9e061a1d7A1AFWQ9pD3jUe5TbLA1ANotif
yaf5f8089d413003404011441506590194090014502 46b4b08bce0a29">
<PresenceNotification xmlns="">
```

```
<MetaInfo clientOriginated="Yes">
<Requestor serviceID="wv:@mnoz.com">
<User userID="wv:i0524412331@mnoz.com"/>
</Requestor>
</MetaInfo>
<Subscribers>
<UserID userID="wv: mnoz@xyz.com"/>
</Subscribers>
<PresenceValue userID="wv:i0524412331@mnoz.com">
<PresenceSubList xmlns="http://www.openmobilealliance.org/DTD/WV-PA1.2" xmlns:Ex
t="http://www.mnoz.com/DTD/FW-PA2.0">
<OnlineStatus>
<Qualifier>T</Qualifier>
<PresenceValue>T</PresenceValue>
</OnlineStatus>
<UserAvailability>
<Qualifier>T</Qualifier>
<PresenceValue>NOT_AVAILABLE</PresenceValue>
</UserAvailability>
</PresenceSubList>
</PresenceValue>
</PresenceNotification>
</Transaction>
</Session>
</WV-SSP-Message>
```

A.5.1.7 Get Presence Transaction

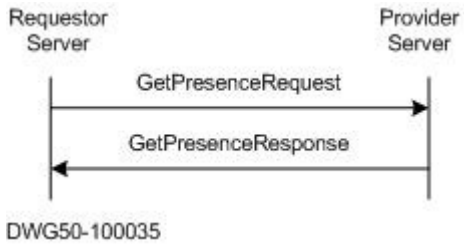

Figure 18 The "GetPresence" Transaction

The purpose of the GetPresence transaction is to allow the requestor server to retrieve presence information of other users.

The requestor server sends a GetPresenceRequest to the provider server for updated presence information of the publishing users. A GetPresenceResponse message from the provider server contains result code(s) and if the request is successful it relays the requested PRESENCE INFORMATION.

Table 19 Primitive Directions for GetPresence Transaction

Primitive	Direction
GetPresenceRequest	Requestor Server → Provider Server
GetPresenceResponse	Requestor Server ← Provider Server

A.5.1.8 The "GetPresenceRequest" Primitive

The GetPresenceRequest primitive allows the requestor server to retrieve the updated presence information. If the presence attribute list is missing from the request, the server sends all available presence information.

Table 20 Information elements in GetPresenceRequest Primitive

Information Element	Req	Type	Description
Message-Type	M	GetPresenceRequest	Message identifier
Meta-Information	M	Structure of Meta-Information	The meta-information
User-ID-List	C	Structure	Identifies the publishing users
Contact-List-ID-List	O	Structure	Identifies the set of publishing users
Presence-Attribute-List	O	Structure	A list of presence attributes to be retrieved. An empty or missing list indicates all presence attributes are desired

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="5426">
<Transaction mode="Request"
transactionID="1320ad1c4825A1AFWQ9pD3jUe5TbLA1AGetPresenceba28c47fbba000
6030115123443502120143000239b9ead5b003c8">
<GetPresenceRequest>
<MetaInfo clientOriginated="Yes">
<Requestor serviceID="wv:@mnoz.com">
<User userID="wv:uk7654321@mnoz.com"/>
</Requestor>
</MetaInfo>
<VerUserID userID="wv:it1234567@mnoz.com"/>
<AttributeList>
<PresenceSubList xmlns="http://www.openmobilealliance.org/DTD/WV-PA1.2"
xmlns:Ext="http://www.mnoz.com/DTD/FW-PA2.0">
<OnlineStatus/>
<ClientInfo/>
<UserAvailability/>
<StatusMood/>
```

```
<Alias/>
</PresenceSubList>
</AttributeList>
</GetPresenceRequest>
</Transaction>
</Session>
</WV-SSP-Message>
```

A.5.1.9 The "GetPresenceResponse" Primitive

The GetPresenceResponse primitive allows the provider server to send the updated presence information to the requestor server.

Table 21 Information elements in GetPresenceResponse Primitive

Information Element	Req	Type	Description
Message-Type	M	GetPresenceResponse	Message identifier
Status-Info	M	Structure of Status-Primitive	Status information
Presence-Value-List	O	Structure	List of User IDs and corresponding presence values

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="5426">
<Transaction mode="Response"
transactionID="1320ad1c4825A1AFWQ9pD3jUe5TbLA1AGetPresenceba28c47fbba00
06030115123443502120143000239b9ead5b003c8">
<GetPresenceResponse xmlns="">
<Status code="200"/>
<PresenceValue userID="wv:it1234567@mnoz.com">
<PresenceSubList  xmlns="http://www.openmobilealliance.org/DTD/WV-PA1.2"
xmlns:Ext =http://www.mnoz.com/DTD/FW-PA2.0>
<OnlineStatus>
<Qualifier>T</Qualifier>
<PresenceValue>F</PresenceValue>
</OnlineStatus>
<UserAvailability>
<Qualifier>T</Qualifier>
<PresenceValue>NOT_AVAILABLE</PresenceValue>
</UserAvailability>
</PresenceSubList>
</PresenceValue>
</GetPresenceResponse>
```

```
</Transaction>
</Session>
</WV-SSP-Message>
```

A.5.1.10 Verify ID Transaction

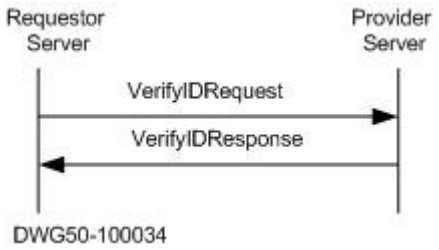

Figure 19 The "VerifyWVID" Transaction

The “VerifyWVIDUserid” transaction is used by the requestor server to verify that a list of WV IDs User-IDs in use are valid at the provider server, i.e. the Home Domain of the WV User-IDs. The transaction is performed before the WV IDUser-ID is stored in the requestor sever, to ensure that all locally-stored WV IDsUser-IDs are valid. The VerifyWVIDUserid response contains the result of the verification, a list of WV IDs with a subset of User-IDs in use.

Table 22 Primitive Directions for the VerifyUserid Transaction

Primitive	Direction
VerifyIDRequest	Requestor Server → Provider Server
VerifyIDResponse	Requestor Server ← Provider Server

A.5.1.11 The “VerifyIDRequest” Primitive

The VerifyIDRequest primitive allows the requestor server to verify that user ids are valid in the provider server.

Table 23 Information elements in VerfifyIDRequest Primitive

Information Element	Req	Type	Description
Message-Type	M	VeifyIDRequest	Message identifier
Meta-Information	M	Structure of Meta-Information	The meta-information
WV-ID-List	M	Structure	The list contains the WV-ID’s to be verified. Only one user in the list is supported.

Example:

```
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="156">
<Transaction mode="Request"
transactionID="6fa884b4209fA1AFWQ9pD3jUe5TbLA1AVerifya80887d740f3003404011441506590194090014
50246a14d529714a9">
<VerifyIDRequest>
```

```
<MetaInfo clientOriginated="No">
<Requestor serviceID="wv:@mnoz.com">
<User userID="wv:i0524412331@mnoz.com"/>
</Requestor>
</MetaInfo>
<WVIDList>
<VerifyUserID userID="wv:mnoz@xyz.com"/>
</WVIDList>
</VerifyIDRequest>
</Transaction>
</Session>
</WV-SSP-Message>
```

A.5.1.12 The "VerifyIDResponse" Primitive

The VerifyIDResponse primitive allows the provider server to return the result of the verification, and the list of valid WV IDs.

Table 24 Information elements in VerifyIDResponse Primitive

Information Element	Req	Type	Description
Message-Type	M	VerifyUseridRespons e	Message identifier
Status-Info	M	Structure of Status-Primitive	The status information
WV-ID-List	M	Structure	The list contains the valid WV Ids. Only one user is present in the list

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="156">
<Transaction mode="Response"
transactionID="6fa884b4209fA1AFWQ9pD3jUe5TbLA1AVerifya80887d740f3003404011441506590194090014
0246a14d529714a9">
<VerifyIDResponse>
<Status code="200">
</Status>
<WVIDList xmlns="">
<VerifyUserID userID="wv:mnoz@xyz.com"/>
</WVIDList>
</VerifyIDResponse>
</Transaction>
</Session>
</WV-SSP-Message>
```

A.5.1.13 Search by MSISDN Transaction

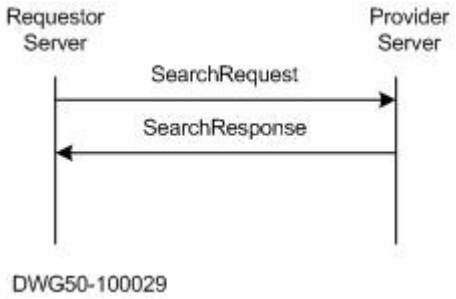

Figure 20 The "GeneralSearch" Transaction

The requestor server sends a SearchRequest message to the provider server, which includes the Search-Pair-List, the type of the search, the USER_MOBILE_NUMBER and the Search-Limit (maximum number of results at a time). The provider server responds with the SearchResponse message, which includes the Status of the search. If the search is successful, it includes the Search-ID, the Search-Findings (the number of items found that match the criteria), and the Search-Results (the username). The IM Interworking Function returns Search-Findings value of ”1” if search is successful, otherwise it returns a value of ”0”. Consecutive searches with the same Search-ID are not supported.

Table 25 Primitive Directions for GeneralSearch Transaction

Primitive	Direction
SearchRequest	Requestor Server → Provider Server
SearchResponse	Requestor Server ← Provider Server

A.5.1.14 The “SearchRequest” Primitive

See A.5.1.16 for details.

A.5.1.15 The “SearchResponse” Primitive

See A.5.1.17 for details.

A.5.1.16 Search and Subscribe by MSISDN Transaction

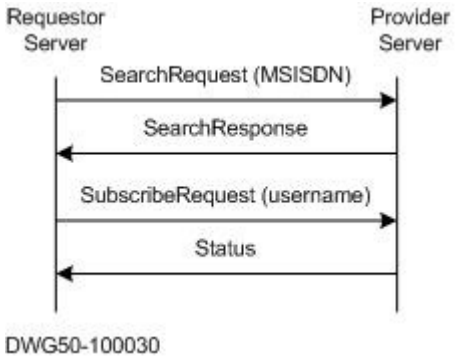

Figure 21 Search by MSISDN Transaction

The requestor server sends a SearchRequest message to the provider server.

The type of the search is USER_MOBILE_NUMBER.
The provider server responds with the SearchResponse message, which includes the Status of the search. Found username value is returned within this transaction.

Afterwards the requestor server sends SubscribeRequest message, with the replied username.

Table 26 Primitive Directions for GeneralSearch Transaction

Primitive	Direction
SearchRequest	Requestor Server → Provider Server
SearchResponse	Requestor Server ← Provider Server
SubscribeRequest	Requestor Server → Provider Server
Status	Requestor Server ← Provider Server

A.5.1.17 The "SearchRequest" Primitive

The SearchRequest primitive allows a user to search for users, based on MSISDN of the user.
The search is performed using one Search-Pair. A Search-Pair consists of a Search-Element and a Search-String. The Search-Element indicates which property of the user is searched for by the Search-String. The IM Interworking Function shall support the MSISDN property.

The result of a user search is always a user-ID, if user is found.

Search-Element for User Search is listed below.

Table 27 Search-Element for User Search

Search-Element	Description
USER_MOBILE_NUMBER	The Search-String is a mobile number [E.164].

Table 28 Information elements in SearchRequest Primitive

Information Element	Req	Type	Description
Message-Type	M	SearchRequest	Message identifier
Meta-Information	M	Structure of Meta-Information	The meta-information
Search-Pair-List	C	Structure	Search criteria in terms of properties. It is present only in the first search request
Search-Limit	C	Integer	Indicates the number of maximum search results that can be received at one time. It is Present only in the first search request
Search-ID	C	String	Uniquely identifies a search transaction. The Responding server

			assigns this ID when the first search is performed, thus it is not present in the first search request
Search-Index	O	Integer	Indicates that the results are sent starting from this particular index. It is present only when the search is continued

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="5426">
<Transaction mode="Request"
transactionID="f76ea84fca20A1AFWQ9pD3jUe5TbLA1ASearch5e8d45b1b1f5006030115123443502120143000
239511b0fae0039">
<SearchRequest searchType="U" searchLimit="10" searchID="11145">
<MetaInfo clientOriginated="Yes">
<Requestor serviceID="wv:@mnoz.com">
<User userID="uk00100010@mnoz.com"/>
</Requestor>
</MetaInfo>
<SearchTerm attr="USER_MOBILE_NUMBER " value="3971712771"/>
</SearchRequest>
</Transaction>
</Session>
</WV-SSP-Message>
```

A.5.1.18 The "SearchResponse" Primitive

Table 29 Information elements in SearchResponse Primitive

Information Element	Req	Type	Description
Message-Type	M	SearchResponse	Message identifier
Status-Info	M	Structure of Status-Primitive	The status information
Search-ID	C	String	Uniquely identifies a search transaction. The server assigns this ID when the first search is performed successfully
Search-Findings	M	Integer	Indicates the number of current findings
Completed	O	Boolean	Indicates if the client can expect new results. 'No' if server may provide new results (still searching), 'Yes' if new results cannot be provided
Search-Index	O	Integer	Indicates the index of the last result. This provides the user with information as to where to continue the next search

Search-Results	C	Structure	Search results

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="5426">
<Transaction mode="Response"
transactionID="9e421ab27c9fA1AFWQ9pD3jUe5TbLA1ASearchcb0a0b0613040060301151234435021201430
0239ddfa6fb198ed">
<SearchResponse searchType="U" searchFindings="1" searchID="11145">
<Status code="200"/>
<SearchResult>
<User userID="it00100009@mnoz.com"/>
</SearchResult>
</SearchResponse>
</Transaction>
</Session>
</WV-SSP-Message>
```

A.5.1.19 Search Response codes

- **Successful (200)**

When search is not successful, for instance when the user associated with the requested MSISDN is not found, IM Interworking Function replies with a “200 OK” response. In this case, the Search-Findings value is ”0”.

When the user is found, IM Interworking Function replies with a “200 OK” and a Search-Findings value is ”1”.

A.5.1.20 Instant Messaging Transactions and Primitives

A.5.1.21 Send Message Transaction

Figure 22 The "SendMessage" Transaction

The purpose of a “SendMessage” transaction is to allow the requestor server to send instant messages through the provider server.

The requestor server sends a SendMessageRequest message to the provider server. The provider server returns a SendMessageResponse response, containing the result and the message ID.

Table 30 Primitive Directions for SendMessage Transaction

Primitive	Direction
SendMessageRequest	Requestor Server → Provider Server
SendMessageResponse	Requestor Server ← Provider Server

A.5.1.22 The “SendMessageRequest” Primitive

The SendMessageRequest primitive allows the requesting server to send instant messages to the users through the requested server.

Table 31 Information elements in SendMessageRequest Primitive

Information Element	Req	Type	Description
Message-Type	M	SendMessageRequest	Message identifier
Meta-Information	M	Structure of Meta-Information	Meta-information
Delivery-Report-Request	M	“Yes” | “No”	Indicates if the user wants a delivery report. “No” is used by the IM Interworking Function.
Message-Info	M	Structure	Message information data, including {Message-ID or Message-URI, Content-type / MIME, encoding, size, sender and recipients, date and time, validity}. Message-ID is not present. The content is text/html base64 encoded.

Content	C	String or Binary data	The content of the instant message. The content is text/html base64 encoded.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="65">
<Transaction mode="Request"
transactionID="c45be14bf20eA1AFWQ9pD3jUe5TbLA1A18924A1AFWQ9pD3jUe5TbLA1A">
<SendMessageRequest deliveryReport="No">
<MetaInfo clientOriginated="Yes">
<Requestor serviceID="wv:@icgw.com">
<User userID="wv:MonitorUser@IMServer.mnoz.com"/>
</Requestor>
</MetaInfo>
<MessageInfo messageID="" contentType="text/html" contentSize="32" validity="Yes">
<Recipient>
<User userID="wv:toUser@load4.com"/>
</Recipient>
<Sender>
<User userID="wv:MonitorUser@IMServer.mnoz.com"/>
</Sender>
<DateTime>20060409T034412</DateTime>
</MessageInfo>
<ContentData contentType="text/html"
encoding="base64">TW9uaXRvcmluZyB0ZXN0IG1lc3NhZ2U=</ContentData>
</SendMessageRequest>
</Transaction>
</Session>
</WV-SSP-Message>
```

A.5.1.23 The "SendMessageResponse" Primitive

The SendMessageResponse primitive allows the requested server to inform the requesting server of the message sending result.

Table 32 Information elements in SendMessageResponse Primitive

Information Element	Req	Type	Description
Message-Type	M	SendMessageResponse	Message identifier
Status-Info	M	Structure of Status-Primitive	Status information
Message-ID	C	String	Requesting Server-generated message-id for this message

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="20311">
<Transaction mode="Response"
transactionID="76c4c54edb65A1AFWQ9pD3jUe5TbLA1ASendMessage5efaf0d970b6002302701151949777021201430
001982b9874b7777f">
<SendMessageResponse messageID="1234">
<Status code="200"/>
</SendMessageResponse>
</Transaction>
</Session>
</WV-SSP-Message>
```

A.5.1.24 General Status Code

All SSP transactions may return the following status codes:

- Successful (200).
- **Partially successful (201)**. This is used to indicate that the request was successfully completed, but some parts were not completed due to certain errors. The details of the error case(s) are indicated in the response. Example: Subscription to more than one user presence. Some of those subscriptions could have run well and others, badly.
- Internal Server Error (500)

A.5.1.25 General Errors.

The IMPS 1.2 standard Error codes shall be utilised.

Table 33 General Errors

Status Code	Status Description
400	Bad Request
402	Bad Parameter
404	Not Found
405	Service Not Supported
500	Internal Server Error
531	Unknown User
601	Forced Logout
604	Invalid Session / Not Logged In
606	Invalid Service-ID
608	Invalid Password
620	Invalid Server Session

700	Contact List Does Not Exist
750	Invalid or Unsupported Presence Attributes

The following table lists the error codes and condition that can result in an error being returned by any of the MNO involved:

A.6 Presence Framework.

A.6.1 Supported Presence Attributes

A.6.1.1 Mandatory attributes

- UserAvailability

Table 34 User Availability

Information element	UserAvailability
Data type	An enumerated String
Format	One of the following values: AVAILABLE – User is available for all forms of communication NOT_AVAILABLE – User is not available for instant (e.g. call and IM) communication. DISCREET – Communication with the publisher is left to the user's discretion
Description	Defines the availability attribute
Range	AVAILABLE | NOT_AVAILABLE | DISCREET

- OnlineStatus

Table 35 Online Status

Information element	OnlineStatus
Data type	Boolean
Format	Following values: T – At least one of the user's client applications is logged on to the WV server F – None of the user's client application are logged on to the WV server
Description	The login status of the client
Range	T | F

- StatusText

This is a short text string that gives a free form description of user status.

Table 36 Status Text

Information element	StatusText
Data type	String
Format	Free text format
Description	A personal status given as a free text
Range	

- Presence Mapping

PersonalIM defined Presence attributes map the IMPS status in the following manner:

Table 37 Presence Mapping

Mobile State	User Availability	OnlineStatus
Online	Available	True
Offline	Not Available	False
Busy/Away/Not Available	Not Available	True

A.6.1.2 Optional attributes

Any Presence attribute defined in IMPS 1.2 standard document can be used and branded as optional to PersonalIM framework. Mutual support of those attributes must be agreed in a bilateral basis.

A.6.1.3 Alias

The alias presence attribute is transferred in HTTP headers (see A.2). Subscriber sends his alias in HTTP headers during the SubscribeRequest transaction.
When first PresenceNotification is sent to a subscriber, the alias of the owner is sent with Alias PA.

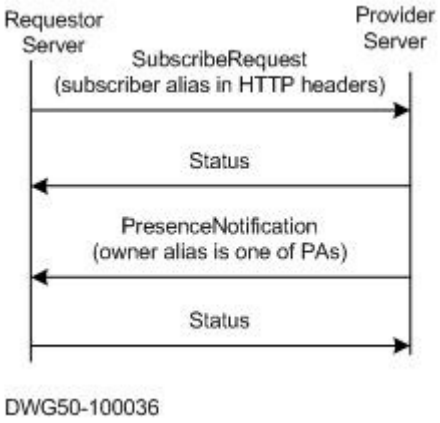

Figure 23 Subscribe Flow

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<WV-SSP-Message xmlns="http://www.openmobilealliance.org/DTD/WV-SSP1.2">
<Session sessionID="20379">
<Transaction mode="Request"
transactionID="56c0d3e07b90A1AFWQ9pD3jUe5TbLA1Az9hG4bK2AF08E4F.3ED360E">
<PresenceNotification xmlns="">
<MetaInfo clientOriginated="Yes">
<Requestor serviceID="wv:@mobile.mnoz.com">
<User userID="wv:james.sharp@mnoz.com"/>
```

```
</Requestor>
</MetaInfo>
<Subscribers>
<UserID userID="wv:447766432181@mobile.mnoz.com"/>
</Subscribers>
<PresenceValue userID="wv:james.sharp@mnoz.com">
<PresenceSubList xmlns="http://www.openmobilealliance.org/DTD/WV-PA1.2"
xmlns:Ext="http://www.mnoz.com/DTD/FW-PA2.0">
<Alias>
<Qualifier>T</Qualifier>
<PresenceValue>James Sharp</PresenceValue>
</Alias>
<OnlineStatus>
<Qualifier>T</Qualifier>
<PresenceValue>T</PresenceValue>
</OnlineStatus>
<UserAvailability>
<Qualifier>T</Qualifier>
<PresenceValue>AVAILABLE</PresenceValue>
</UserAvailability>
</PresenceSubList>
</PresenceValue>
</PresenceNotification>
</Transaction>
</Session>
</WV-SSP-Message>
```

ANNEX B IMPS TEST CASES

B.1 Scope of tests

This annex describes tests for interworking of instant messaging using the IMPS protocol between mobile network operators and between a mobile network and an internet-based client that belongs to an interworking partner. Clause 6 describes a number of addressing and routing solutions for interworking between mobile networks, or between a mobile network and an internet-based client. Clause 6 does not mandate a particular solution for addressing and routing and does not add to the IMPS client client-server protocol (CLP), therefore this annex references tests in existing OMA IMPS specifications. This annex is a guide to using existing IMPS client tests to verify the addressing and routing solution implemented between two networks.

Terminal testing is the responsibility of the relevant testing groups in 3GPP and the Global Certification Forum (GCF).

Interworking of the messaging service between particular messaging clients is not explicitly specified. Tests of services intended for the home network can be adapted for roaming, by using a UE that is GPRS roamed.

This annex does not include tests for mobile number portability (MNP) because an MNP solution has yet to be agreed. MNP is not excluded from the scope of this specification.

B.2 Related OMA Specifications

OMA has specified conformance testing and interworking testing for two versions of IMPS in the specifications listed below. The OMA testing concentrates on ensuring that an IMPS client and an IMPS server conform with the protocol specifications, and also that two different clients, i.e. clients from different vendors, are able to interwork with each other when connected to the same server.

OMA Instant Messaging and Presence Service V1.2.1	ETS for IMPS V1.2.1	15 Nov 2005	OMA-ETS-IMPS_CSP-V1_2_1-20051115-A
OMA Instant Messaging and Presence Service V1.3	ETS for IMPS CSP Interoperability V1.3	30 May 2006	OMA-ETS-IMPS_CSP_INT-V1_3-20060530-C
	ETS for IMPS CSP Conformance V1.3	14 Nov 2006	OMA-ETS-IMPS_CSP_CON-V1_3-20061114-C

OMA testing can be illustrated in terms of the IMPS architecture in IR.76 section 3.3 as shown below.

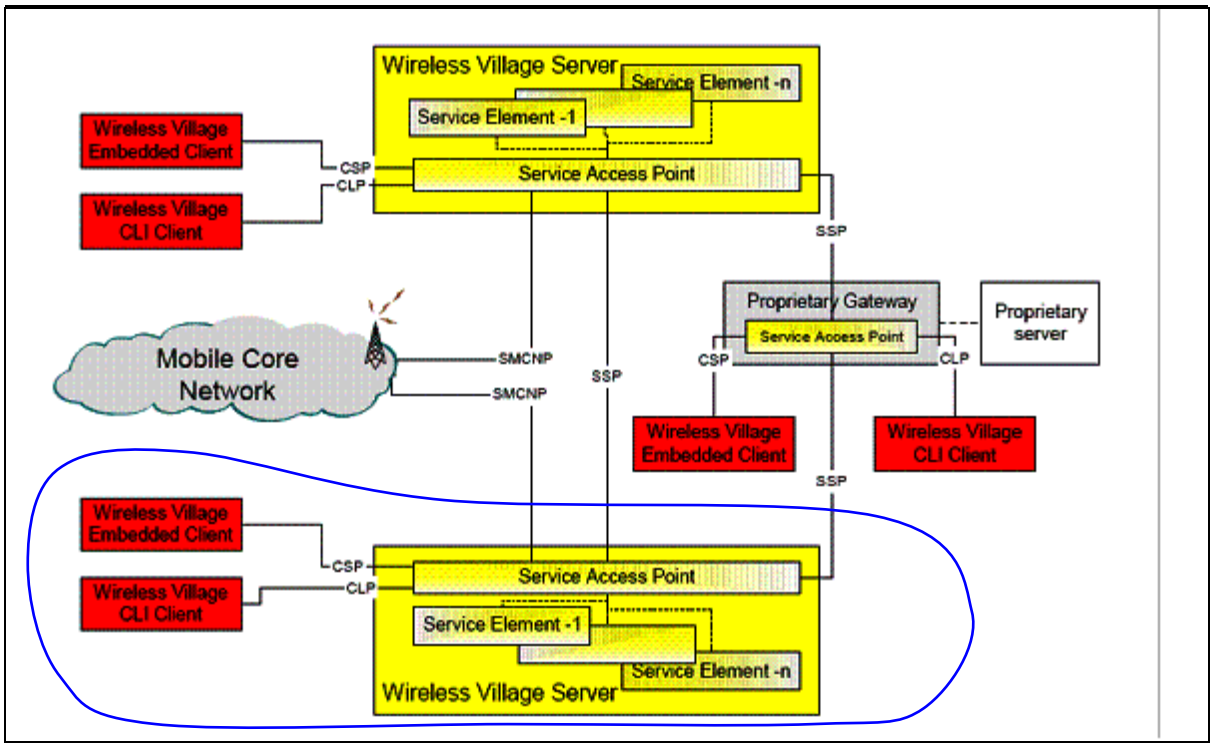

Figure 1 Scope of OMA test specifications

This specification expands the scope to cover interworking between two wireless village servers, or between a wireless village server and a proprietary server. The SMNCP and SSP protocols are not explicitly tested. The OMA IMPS tests for testing interworking of two different clients logged in to the same server are re-used in this specification to test interworking between two clients logged in to servers in different networks. The clients in this specification may be two instances of the same client.

B.3 Objectives of Tests

The overall objective of the tests is to confirm that instant messaging, which is already known to operate correctly within each separate PLMN, also operates correctly for interworking between PLMNs and between a PLMN and an internet-based interworking partner.

The specific objectives of this testing is to confirm the capability of instant messaging when setting up sessions between a UE in MNO Z and a UE in MNO Y, or between a UE in MNO Z and an internet-based instant messaging client. Consequently, the tests are restricted to top-level capability testing. There is no provocative or inopportune behaviour testing.

Not all possible use cases within instant messaging should be tested. The focus is on address resolution and routing between instant messaging servers. It is also tested that authorization of non-home network subscribers to view presence is also tested.

Note: The tests described here do not replace international roaming tests or interworking tests for GPRS.

B.3.1 Single Ended Testing

Testing should be done by one person. To achieve this, one or both terminals may be roaming, or one end can use loop back or a software automaton.

B.3.2 Service Testing or Signalling and User Data Testing

This test specification is for interworking of instant messaging, not for individual requests or responses or testing the user data path. Services tested are:

- Add a contact in reactive mode using an MSISDN as the address;
- Add a contact in reactive mode using an address in the user@domain format.

B.3.3 Usefulness of Test Results

Tests are designed such that if a test passes, then the service can be considered to be working. If the test fails, then something key to the service is not working.

B.3.4 Tests are Simple to Perform

Tests might be performed regularly with many different roaming partners. The tests require the minimum of equipment and are simple to perform.

B.3.5 Build Up Testing in Stages

Participating networks should thoroughly test their instant messaging service internally before performing roaming and interworking tests.

B.3.6 Relevance to Interworking

All tests exercise interworking, to another PLMN or to the Internet. Tests that are isolated to functions within a single network are not to be done.

B.3.7 Prerequisites

For interworking tests between to mobile network operators, two instant messaging capable mobile terminals and one internet-based interworking partner client are needed, one in each of MNO Z, MNO Y, and interworking partner A.

B.4 TEST CONFIGURATION

This specification describes a number of options for addressing and routing between networks for the purpose of instant messaging. The test configuration requires routing between two mobile network operators (MNO Z and MNO Y), or between a mobile network operator (MNO Z) and an internet-based server (IWP A).

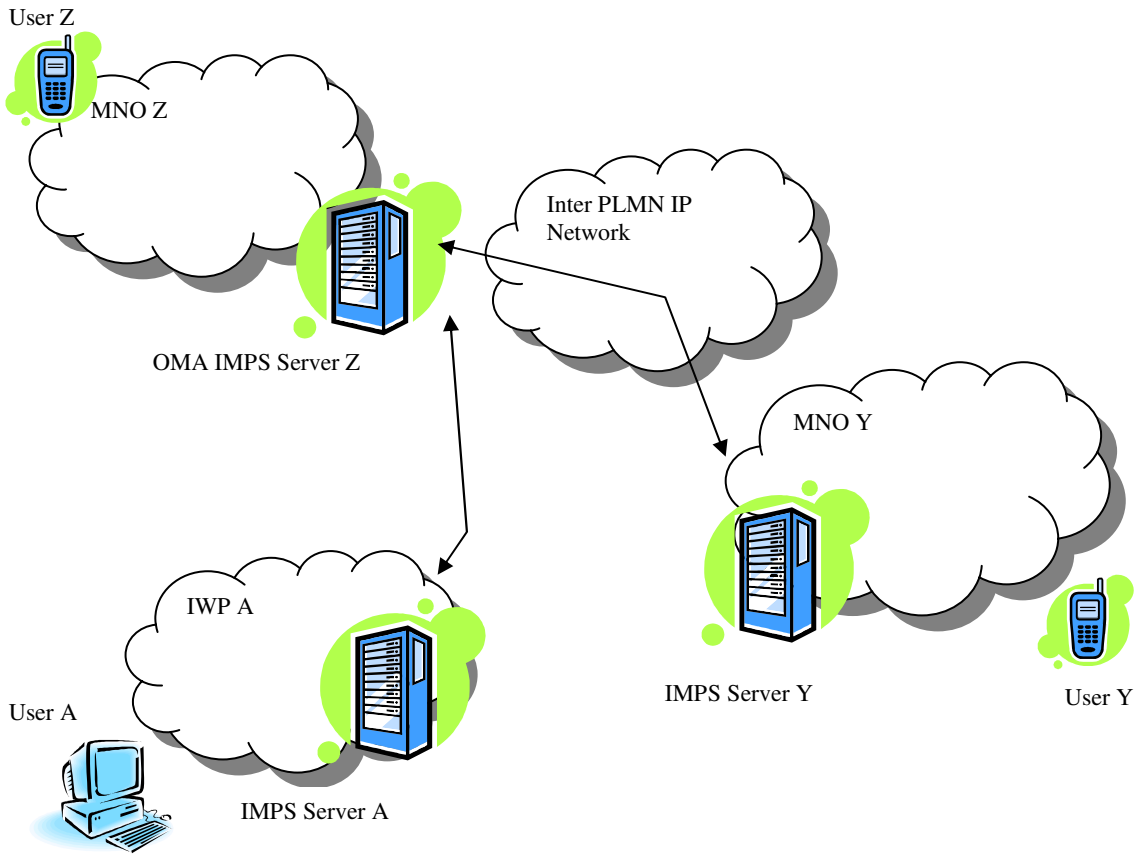

Figure 2 Test Configuration

Compared to the IMPS architecture shown in *Figure 1 Scope of OMA test specifications*, all parts of the architecture are tested, not in isolation but end-to-end, apart from explicitly testing the Server-Server Protocol (SSP).

B.5 ADDRESS RESOLUTION OPTIONS

This specification describes the following solutions for addressing and routing between instant messaging servers. See clause 6 for a full description of each option.

B.5.1 Direct Interconnection Scheme

Direct interconnection is described in clause 6.3. The routing intelligence is configured in the OMA-IMPS servers. Each OMA-IMPS server has two types of records, one related to the pair of Sessions built with the other domain and another one resolving the IP address to reach that domain.

B.5.2 IMPS and the Domain Name Service-DNS

DNS options are described in clause 6.4.

B.5.2.1 Option 1 – Terminal A Records

Similar to direct interconnection. DNS is implemented with address records (A records), which directly identify an IP address for each domain name. Unique domain names are needed to differentiate IM traffic from other traffic addressed to the same operator, as well as to uniquely identify domain names belonging to Internet IM operators like MSN and AOL.

B.5.2.2 Option 2 – Redirection NS records

Domains are mapped to the address of a DNS server that can then resolve the domain name. The DNS server is maintained by the owner of the domain.

```
domainame.           IN NS    dns1.domainame
dns1.domainame.           A    101.1.2.3
```

The IP addresses of DNS servers in other networks must be available to the Local DNS Static server, but the amount of manual updates required could be seen as reasonable.

Unique domain names must be agreed bilaterally to differentiate IM traffic from other traffic addressed to the same operator, as well as to uniquely identify domain names belonging to Internet IM operators like MSN and AOL.

B.5.2.3 Option 3 – DDDS and NAPTR records

The main advantage of this option is its total flexibility and scalability. A single domain name can be mapped to a list of services provided at that domain, which can be used as inputs to service queries using SRV records.

The main drawback is the use of additional SRV queries, or the need to provision domain names in the DNS that are unique to the IM service (i.e. `im.domainame`).

B.5.3 Other Interworking Services (resolving MSISDNs)

Other interworking services are described in clause 6.5. Some use cases require MSISDN as the identity of the destination, and this must be resolved to an IMPS server. This is possible using static configuration or by using MAP to translate to IMSI followed by DNS lookup. This functionality is not currently available, but could be provided quite quickly by the GRX/IPX.

B.6 MOBILE NETWORK TO MOBILE NETWORK INTERWORKING

OMA-ETS-IMPS_CSP-V1_2_1-20051115-A contains tests in the following sections that can be re-used for interworking tests.

In all cases, the preconditions of the test must be changed such that client A and client B have accounts on IMPS servers belonging to mobile network operators MNO Z and MNO Y in .

Relevant sections of the OMA test specification are:

Function	**Section in OMA-ETS-IMPS_CSP-V1_2_1-20051115-A**
Presence	6.1.2 IMPS-1.2.1-int-PRSE
Instant messaging	6.1.3 IMPS-1.2.1-int-IMSE
Group functions (joining, creating, leaving)	6.1.4 IMPS-1.2.1-int-GROUPS
Shared content	6.1.5 IMPS-1.2.1-int-COSE

B.7 MOBILE NETWORK TO INTERNET-BASED CLIENT INTERWORKING

OMA-ETS-IMPS_CSP-V1_2_1-20051115-A contains tests in the following sections that can be re-used for interworking tests.

In all cases, the preconditions of the test must be changed such that client A has an account with an IMPS server belonging to the mobile network operator and client B has an account with the internet-based instant messaging service.

Function	**Section in OMA-ETS-IMPS_CSP-V1_2_1-20051115-A**
Presence	6.1.2 IMPS-1.2.1-int-PRSE
Instant messaging	6.1.3 IMPS-1.2.1-int-IMSE
Group functions (joining, creating, leaving)	6.1.4 IMPS-1.2.1-int-GROUPS
Shared content	6.1.5 IMPS-1.2.1-int-COSE