



**Implementation Agreement of Parlay X API for
GMI 2006**

MSF-IA-PARLAY.004-FINAL

MultiService Forum Implementation Agreement

Contribution Number: msf2005.135.03

Document Filename: MSF-IA-PARLAY.004-FINAL

Working Group: Protocol and Control

Title: Implementation Agreement of Parlay X API for GMI 2006

Editors:

ETRI

Hyung-Hwan Kim
hhkim@etri.re.kr

ETRI

Young-il Choi
yichoi@etri.re.kr

ETRI

Byung-sun Lee
bslee@etri.re.kr

Working Group Chairperson: Chris Gallon, Fujitsu

Date: 19 May 2006

Abstract: This document is an Implementation Agreement (IA) of the interface between Parlay X gateway and application server for the value added service in GMI 2006.

The MultiService Forum (MSF) is responsible for developing Implementation Agreements, Product Specifications or Architectural Frameworks, which can be used by developers and network operators to ensure interoperability between components from different vendors. MSF Implementation Agreements, Product Specifications and Architectural Frameworks are formally ratified via a Straw Ballot and then a Principal Member Ballot. Draft MSF Implementation Agreements, Product Specifications and Architectural Frameworks may be published before formal ratification via Straw or Principal Member Ballot. In order for this to take place, the MSF Technical Committee must formally agree that a draft Implementation Agreement, Product Specifications or Architectural Framework should be progressed through the balloting process. A Draft MSF Implementation Agreement, Product Specification or Architectural Framework is given a document number in the same manner as an Implementation Agreement. Draft Implementation Agreements, Product Specifications or Architectural Frameworks may be revised before or during the full balloting process. The revised document is allocated a new major or minor number and is published. The original Draft Implementation Agreement, Product Specifications or Architectural Framework remains published until the Technical Committee votes to withdraw it. After being ratified by a Principal Member Ballot, the Draft Implementation Agreement, Product Specifications or Architectural Framework becomes final. Earlier Draft Implementation Agreements, Product Specifications or Architectural Frameworks remain published until the Technical Committee votes to withdraw them.

The goal of the MSF is to promote multi-vendor interoperability as part of a drive to accelerate the deployment of next generation networks. To this end the MSF looks to adopt pragmatic solutions in order to maximize the chances for early deployment in real world networks.

To date the MSF has defined a number of detailed Implementation Agreements and detailed Test Plans for the signaling protocols between network components and is developing additional Implementation Agreements and Test Plans addressing some of the other technical issues such as QoS and Security to assist vendors and operators in deploying interoperable solutions.

The MSF welcomes feedback and comment and would encourage interested parties to get involved in this work program. Information about the MSF and membership options can be found on the MSF website <http://www.msforum.org/>

DISCLAIMER

The information in this publication is believed to be accurate as of its publication date. Such information is subject to change without notice and the MultiService Forum is not responsible for any errors or omissions. The MultiService Forum does not assume any responsibility to update or correct any information in this publication. Notwithstanding anything to the contrary, neither the MultiService Forum nor the publisher make any representation or warranty, expressed or implied, concerning the completeness, accuracy, or applicability of any information contained in this publication. No liability of any kind whether based on theories of tort, contract, strict liability or otherwise, shall be assumed or incurred by the MultiService Forum, its member companies, or the publisher as a result of reliance or use by any party upon any information contained in this publication. All liability for any implied or express warranty of merchantability or fitness for a particular purpose is hereby disclaimed.

The receipt or any use of this document or its contents does not in any way create by implication or otherwise:

Any express or implied license or right to or under any MultiService Forum member company's patent, copyright, trademark or trade secret rights which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor

Any warranty or representation that any MultiService Forum member companies will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor

Any commitment by a MultiService Forum company to purchase or otherwise procure any product(s) and/or service(s) that embody any or all of the ideas, technologies, or concepts contained herein; nor

Any form of relationship between any MultiService Forum member companies and the recipient or user of this document.

Implementation or use of specific MultiService Forum Implementation Agreements, Architectural Frameworks or recommendations and MultiService Forum specifications will be voluntary, and no company shall agree or be obliged to implement them by virtue of participation in the MultiService Forum.

For addition information contact:

MultiService Forum
39355 California Street, Suite 307
Fremont, CA 94538
USA
Phone: +1 510 608-5922
Fax: +1 510 608-5917
info@msforum.org
<http://www.msforum.org>

1. Scope

This Implementation Agreement defines a profile for the interface between the Parlay X gateway and Parlay X application Server.

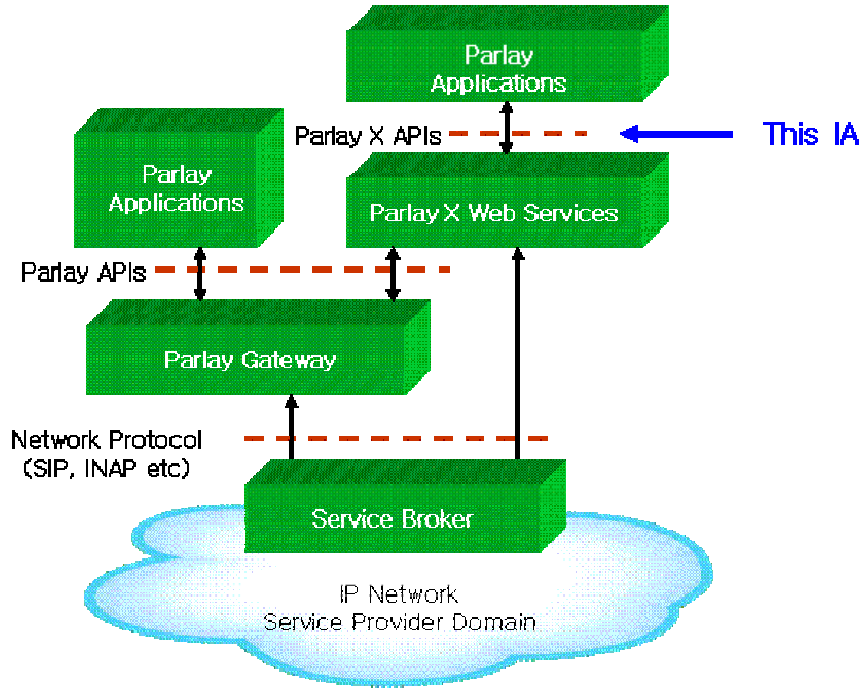


Figure 1.Document scope

Global MSF Interoperability 2006 (GMI2006) will cover a service layer with application server, media server, and service broker functionality and demonstrate how services can reside in multiple locations including Call Agents, SIP Application Servers, and Parlay X gateways and applications. To encourage more service providers and carriers to join MSF interoperability test, we need to address service layer components such as Parlay X gateways and Parlay X application servers, and interfaces between them.

This document describes interfaces between a Parlay X gateway and a Parlay X application server as shown Figure 1.

Note: The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, “OPTIONAL”, “CONDITIONAL” and “IF” in this document are to be interpreted as described in the Technical Committee Operation Procedures.

2. References

The interfaces are based on API standards from Parlay group.

	Document Identification	Document Title
1	ETSI ES 202 391-1 v1.1.1	Open Service Access(OSA); Parlay X Web Services; Part 1: Common
2	ETSI ES 202 391-2 v1.1.1	Open Service Access(OSA); Parlay X Web Services; Part 2: Third Party Call
3	ETSI ES 202 391-3 v1.1.1	Open Service Access(OSA); Parlay X Web Services; Part 3: Call Notification
4	ETSI ES 202 391-4 v1.1.1	Open Service Access(OSA); Parlay X Web Services; Part 4: Short Messaging
5	ETSI ES 202 391-9 v1.1.1	Open Service Access(OSA); Parlay X Web Services; Part 9: Terminal Location
6	ETSI ES 202 391-10 v1.1.1	Open Service Access(OSA); Parlay X Web Services; Part 10: Call Handling
7	ETSI ES 202 391-12 v1.1.1	Open Service Access(OSA); Parlay X Web Services; Part 12: Multimedia Conference
8	ETSI ES 202 391-14 v1.1.1	Open Service Access(OSA); Parlay X Web Services; Part 14: Presence

3. Definition

This section provides an overview of the primary public specifications dealing with the Parlay X gateway's main interfaces. Complete details on all these specifications are listed in the References section. This IA essentially defines a profile of each specification for use in GMI2006.

3.1. Parlay X APIs

3.1.1. General

GMI 2006 target services which are provided using Parlay X gateways and Parlay X Application servers are as the followings:

- Presence enabled Conference Call
- Rule based Call Handling
- Location based Directory Service
- One Number Service
- Click-to-Connect
- Click-to-Conference

3.1.2. Version

The Parlay X Web Services specification V2.0 has been defined by Parlay X Working Group of the Parlay Group since March 2005. It defines 13 APIs as a following table. Parlay X gateways SHOULD support APIs which are marked as “yes” in “Part of this IA” column for proposed services as value added services in GMI 2006.

Parlay X APIs	Part of this IA
Third Party Call	Yes
Call Notification	Yes
Short Messaging	Yes
Multimedia Messaging	Yes
Payment	No
Account Management	No
Terminal Status	No
Terminal Location	Yes
Call Handling	Yes
Audio Call	No
Multimedia Conference	Yes
Address List Management	No
Presence	Yes

3.1.3. Third Party Call

The overall scope of Third Party Call API is to provide functions for application developers to create a call in a simple way. Using this, application developers can make a call between two phones without specific Telco knowledge.

The following table describes the REQUIRED Third Party Call APIs for GMI2006 test services.

API		Parameters		
Name	M/O	Mandatory	Optional	Output
MakeCall	Mandatory	CallingParty, CalledParty	Charging	CallIdentifier
GetCallInformation	Mandatory	CallIdentifier		CallInformation
EndCall	Mandatory	CallIdentifier		None
CancelCall	Mandatory	CallIdentifier		None

3.1.4. Call Notification API

The overall scope of Call Notification API is to provide simple functions for application developers to determine how a call should be treated. Using this, application developers can perform simple handling of network-initiated calls without specific Telco knowledge.

The following table describes the REQUIRED Call Notification APIs for GMI2006 test services.

API		Parameters		
Name	M/O	Mandatory	Optional	Output
HandleBusy	Optional	CallingParty, CalledParty		Action
HandleNotReachable	Optional	CallingParty, CalledParty		Action
HandleNoAnswer	Optional	CallingParty, CalledParty		Action
HandleCalledNumber	Mandatory	CallingParty, CalledParty		Action
NotifyBusy	Optional	CallingParty, CalledParty		None
NotifyNotReachable	Optional	CallingParty, CalledParty		None
NotifyNoAnswer	Optional	CallingParty, CalledParty		None
NotifyCalledNumber	Optional	CallingParty, CalledParty		None

3.1.5. Short Messaging API

The overall scope of Short Messaging API is to provide primitives for application developers to handle SMS in a simple way. Using this, application developers can invoke SMS functions without specific Telco knowledge. The following table describes the REQUIRED Short Messaging APIs for GMI2006 test services.

API		Parameters		
Name	M/O	Mandatory	Optional	Output
SendSms	Mandatory	Addresses, SenderName, Message	Charging	RequestIdentifier
SendSmsLogo	Optional	Addresses, Image, SmsFormat	SenderName, Charging	RequestIdentifier
SendSmsRingTone	Optional	Addresses, Ringtone, SmsFormat	SenderName, Charging	RequestIdentifier
GetSmsDeliveryStatus	Optional	RequestIdentifier		DeliveryStatus
NotifySmsReception	Optional	RegistrationIdentifier, Message		None
GetReceivedSms	Optional	RegistrationIdentifier		ReceivedSms

3.1.6. Multimedia Messaging API

The overall scope of Multimedia Messaging API is to provide functions for application developers to receive and send Multimedia Messages programmatically in a simple way. Using this, application developers can get advantages without detailed Telco knowledge as follows:

- improved service portability
- lower complexity, by providing support for generic user terminal capabilities only.

The following table describes the REQUIRED Multimedia Messaging APIs for GMI2006 test services.

API		Parameters		
Name	M/O	Mandatory	Optional	Output
SendMessage*	Mandatory	Addresses	SenderAddress, Subject, Priority, Charging	RequestIdentifier
GetMessageDeliveryStatus	Optional	RequestIdentifier		DeliveryStatus
GetReceivedMessages	Optional	RegistrationIdentifier	Priority	Messages
GetMessageURIs	Optional	MessageRefIdentifier		Message
GetMessage	Optional	MessageRefIdentifier		None
NotifyMessageReception	Optional	RegistrationIdentifier, Message		None

* Note: The content is sent as a form of SOAP message with attachments. The SendSms of SMS API has a parameter for the content to be sent, but the SendMessage of MMS API has not any parameter for the content to be sent.

3.1.7. Terminal Location API

The overall scope of Terminal Location API is to provide functions for application developers to access to the location of a terminal through: Request for the location of a terminal, Request for the location of a group of terminals, Notification of a change in the location of a terminal, Notification of terminal location on a periodic basis. Location is expressed through a latitude, longitude, altitude and accuracy.

The following table describes the REQUIRED Terminal Location APIs for GMI2006 test services.

API		Parameters		
Name	M/O	Mandatory	Optional	Output
GetLocation	Mandatory	Address, RequestedAccuracy, AcceptableAccuracy		Result
GetTerminalDistance	Optional	Address, Latitude, Longitude		Result

GetLocationForGroup	Optional	Addresses, RequestedAccuracy, AcceptableAccuracy		Result
StartGeographicalNotification	Optional	Reference, Addresses, Latitude, Longitude, Radius, Criteria, CheckImmediate, Frequency, Duration, Count		None
StartPeriodicNotification	Optional	Reference, Addresses, RequestedAccuracy, Frequency, Duration		None
EndNotification	Optional	RegistrationIdentifier		None
LocationNotification	Optional	Correlator, Data, Criteria		None
LocationError	Optional	Correlator, Address, Reason		None
LocationEnd	Optional	Correlator		None

3.1.8. Call Handling API

The overall scope of Call Handling API is to provide functions for application developers to specify how calls are to be handled for a specific number in a simple way. Using this, application developers can specify call handling without detailed Telco knowledge as follows:

- Call accepting - only accepting calls from a list of numbers.
- Call blocking - blocking calls if they are on a blocking list.
- Conditional call forwarding - changing the destination of a call to another number for a specific calling number.
- Unconditional call forwarding - changing the destination of a call to another number.
- Play audio - initiate audio with the caller (e.g. an announcement or menu).

The following table describes the REQUIRED Call Handling APIs for GMI2006 test services.

API		Parameters		
Name	M/O	Mandatory	Optional	Output
SetRules	Mandatory	Address, Rules		None
SetRulesForGroup	Mandatory	Addresses, Rules		Result
GetRules	Mandatory	Addresses		Rules
ClearRules	Mandatory	Addresses		None

3.1.9. Multimedia Conference API

The overall scope of Multimedia Conference API is to provide functions for application developers to create a multimedia conference in a simple way. Using this, application developers can create a multimedia conference and manage the participants and media involved dynamically without detailed Telco knowledge.

The following table describes the REQUIRED Multimedia Conference APIs for GMI2006 test services.

API		Parameters		
Name	M/O	Mandatory	Optional	Output
createConference	Mandatory		ConferenceType, ConferenceDescription, Charging, MaximumDuration, MaximumNumberOfPart icipants, ConferenceOwner	ConferenceIdent ifier
getConferenceInfo	Optional	ConferenceIdentifier		ConferenceInfo
endConference	Mandatory	ConferenceIdentifier		None
inviteParticipant	Mandatory	ConferenceIdentifier, Participant		None
disconnectParticipant	Optional	ConferenceIdentifier, Participant		None
getParticipantInfo	Optional	ConferenceIdentifier, Participant		ParticipantInfo
getParticipants	Optional	ConferenceIdentifier		Participants

addMediaForParticipant	Optional	ConferenceIdentifier, Participant, Media, MediaDirection		None
deleteMediaForParticipant	Optional	ConferenceIdentifier, Participant, Media, MediaDirection		None

3.1.10. Presence API

The overall scope of Presence API is to provide functions for application developers to obtain and register presence information about one or more users in a simple way. Using this, application developers can obtain and register presence information about one or more users without detailed Telco knowledge.

The following table describes the REQUIRED Presence APIs for GMI2006 test services.

API		Parameters		
Name	M/O	Mandatory	Optional	Output
subscribePresence	Mandatory	Presentity, Attributes, Application, Reference		None
getUserPresence	Mandatory	Presentity, Attributes		Result
startPresenceNotification	Mandatory	Presentity, Attributes, Reference, Frequency, Duration, Count, CheckImmediate		Presentities
endPresenceNotification	Mandatory	Correlator		None
statusChanged	Mandatory	Correlator, Presentity, ChangedAttributes		None
statusEnd	Optional	Correlator		None
notifySubscription	Optional	Presentity, Decisions		None
subscriptionEnded	Mandatory	Presentity, Reason		None

publish	Mandatory	Presence		None
getOpenSubscriptions	Optional	None		OpenRequests
updateSubscriptionAuthorization	Optional	Watcher, Decisions		None
getMyWatchers	Optional	None		Result
getSubscribedAttributes	Optional	Watcher		Result
blockSubscription	Optional	Watcher		None

3.2. Web Service Interface

The web service specifications incorporated in this IA are UDDI, WSDL, SOAP, XML and XML Schema. UDDI is implementation specific and its use is optional in this IA. This IA refers to the capabilities of the currently published versions of each specification:

Specification	Version	M/O	Organization
UDDI	2.0	Optional	OASIS
WSDL	1.1	Mandatory	W3C
SOAP	1.1	Mandatory	W3C
XML	1.0	Mandatory	W3C
XML Schema	1.0	Mandatory	W3C
WSS: SOAP Message Security	1.0	Mandatory	OASIS
WSS: SOAP Message Security UsernameToken Profile	1.0	Mandatory	OASIS

3.2.1. Access control

The Parlay X gateway SHALL execute an access control for the purpose of verifying its user for the secure use of Parlay X Web services. The access control is the first step for achieving this purpose. Generally, identifier(i.e., claim which is a declaration made by an entity such as name, identity, key, group, privilege, capability, etc) is used for representing who the user of service is. The identifier is conveyed within a header of a SOAP message to Parlay X gateway. According to the Parlay X specification, “if a SOAP message contains an identifier and/or credentials representing the sender of the message then these SHALL be provided in a manner prescribed by WS-Security(Web Services Security: SOAP Message Security 1.0 by OASIS).”

WS-Security describes enhancements to SOAP messaging to provide protection through message integrity, confidentiality, and authentication. It also provides a mechanism for associating security tokens with messages and describes how to encode binary security tokens such as X.509 certificates and Kerberos tickets as well as how to include encrypted keys. It's for end-to-end security not point-to-point security such as HTTPS over SSL. According to the WS-Security specification, security tokens are classified into unsigned security tokens and signed security tokens. Hence, we need to clarify which mechanism we should use for the access control.

The ways of using security tokens are specified in the WS-Security specification looks like a figure 2 below. The former is more simpler than the latter in the view of implementation and enough for the access control functionality of Parlay X gateway. Even though the former seems to be an efficient way for the access control, it is less secure method than the latter because the former is plain text based. The unsigned security token based access control has the meaning of a stepping stone for applying security to Parlay X GW.

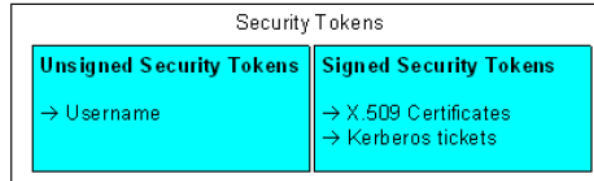


Figure 2. Classification of security tokens

The security token SHALL be used as follows.

Method	M/O
Unsigned Security Token - <wsse:UsernameToken> - <wsse:Username> - <wsse:Password Type="PasswordText">	Mandatory
Signed Security Token	Optional

*** Terminology**

- Security Token: a security token represents a collection of claims where a claim is a declaration that an entity makes (e.g. name, identity, key, group, privilege, capability, etc).
- Unsigned security tokens: a security token such as a username and password those are plain text.
- Signed security tokens: a security token that is asserted and cryptographically signed by a specific authority (e.g. an X.509 certificate or a Kerberos ticket)

The following is an example of SOAP message which unsigned security token is applied to.

```
(000) <?xml version="1.0" encoding="utf-8"?>
(001) <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
(002)   <soap:Header>
(003)     <wsse:Security soap:mustUnderstand="1">
(004)       <wsu:Timestamp wsu:Id="Timestamp-59d37628-10ea-4278-bb2b-843765cf987d">
(005)         <wsu:Created>2006-01-03T09:22:25Z</wsu:Created>
(006)         <wsu:Expires>2006-01-03T09:27:25Z</wsu:Expires>
(007)       </wsu:Timestamp>
```

```
(008)    <wsse:UsernameToken wsu:Id="SecurityToken-04f16ebd-8ccf-492a-9a74-f5d3787cbb28"  
          xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-  
          utility-1.0.xsd">  
(009)    <wsse:Username>AS-Name</wsse:Username>  
(010)    <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-  
          username-token-profile-1.0#PasswordText">AS-Password</wsse:Password>  
(011)    <wsse:Nonce>9IsE+uQCIOyV8mAOI9cbmw==</wsse:Nonce>  
(012)    <wsu:Created>2006-01-03T09:22:25Z</wsu:Created>  
(013)    </wsse:UsernameToken>  
(014)    </wsse:Security>  
(015)    </soap:Header>  
(016)    <soap:Body>  
(017)      Application Specific Messages  
(018)    </soap:Body>  
(019)    </soap:Envelope>
```

The <wsse:UsernameToken> element at lines (008) to (013) is a way of providing a username such as line (009). Within <wsse:UsernameToken> element, a <wsse:Password> element SHALL be specified according to Web Services Security UsernameToken Profile 1.0 specification.

All error handling during the access control, SHALL obey WS-Security 1.0 and WS-Security UsernameToken Profile 1.0 specification.

Parlay X application servers are users of services those are provided by Parlay X GW. Hence, Parlay X GW must manage Parlay X application server's profile which is composed of username and password. The user name between <wsse:Username> and </wsse:Username> tags like line (009) and password between <wsse:Password> and </wsse:Password> tags like line (010) are same with those in Parlay X Application server's profile.

3.2.2. Requester concerns in Presence API

Even though the Presence Web service needs the identity of a requester (i.e., the watcher and the presentity) to execute its own functionality, the Presence Web service specification (ETSI ES 202 391-14 v1.1.1) does not has parameters which represent the identity of requester. Since it is assumed that the watcher and the presentity have been previously authenticated so that the identity is known.

Currently, there is no explicitly defined ways how to transfer the identity of a requester. Therefore, application developers, who use the functions described in ETSI ES 202 391-14 v1.1.1, convey the identity of requester in the SOAP message header. However, it is not enough for achieving interoperability between the Parlay X gateway and Parlay X application server because how to deliever the identity of a requester is not identified.

This document suggests a method to insert the following XML tags in the table into the header of SOAP message to maximize the interoperability.

Parameter	Description in SOAP header
watcher	<watcher>xsd:anyURI of watcher</watcher>
presentity	<presentity>xsd:anyURI of presentity</ presentity>

Hence, application developers, who use the functions described in ETSI ES 202 391-14 v1.1.1, SHOULD use <watcher>, </watcher> tags for watcher and <presentity>, </presentity> tags for presentity in the header of SOAP message.

4. NAT and Firewall Traversal

The use of NAT and firewall is outside the scope of this IA.

5. QoS Aspects

None

6. Security aspects

None

7. Redundancy and Resilience

None

8. Management Information Model

None

End of document