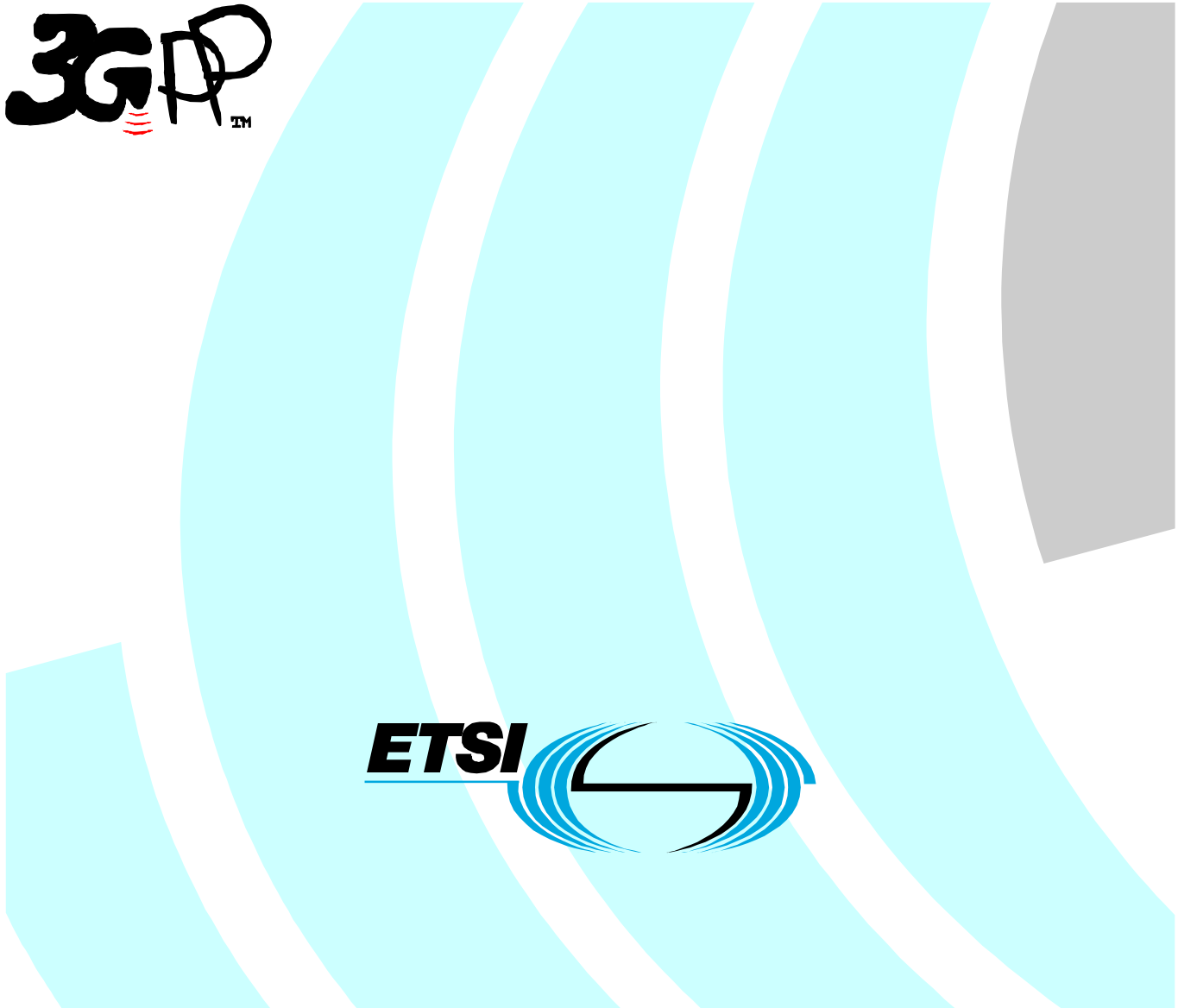
ETSI TS 129 199-14 V6.5.0 (2006-12)

Technical Specification

Universal Mobile Telecommunications System (UMTS);
Open Service Access (OSA);
Parlay X web services;
Part 14: Presence
(3GPP TS 29.199-14 version 6.5.0 Release 6)



Reference
RTS/TSGC-0529199-14v650

Keywords
UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp .

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

3GPP acknowledges the contribution of the Parlay X Web Services specifications from The Parlay Group. The Parlay Group is pleased to see 3GPP acknowledge and publish the present document, and the Parlay Group looks forward to working with the 3GPP community to improve future versions of the present document.

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

The present document is part 14 of a multi-part deliverable covering the 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Open Service Access (OSA); Parlay X Web Services, as identified below:

Part 1: "Common";
Part 2: "Third party call";
Part 3: "Call Notification";
Part 4: "Short Messaging";
Part 5: "Multimedia Messaging";
Part 6: "Payment";
Part 7: "Account management";
Part 8: "Terminal Status";
Part 9: "Terminal location";
Part 10: "Call handling";
Part 11: "Audio call";
Part 12: "Multimedia conference";
Part 13: "Address list management";
Part 14: "Presence".

1 Scope

The present document is Part 14 of the Stage 3 Parlay X Web Services specification for Open Service Access (OSA).

The OSA specifications define an architecture that enables application developers to make use of network functionality through an open standardized interface, i.e. the OSA APIs. The concepts and the functional architecture for the OSA are contained in 3GPP TS 23.198 [3]. The requirements for OSA are contained in 3GPP TS 22.127 [2].

The present document specifies the Presence Web Service aspects of the interface. All aspects of the Presence Web Service are defined here, these being:

- Name spaces.
- Sequence diagrams.
- Data definitions.
- Interface specification plus detailed method descriptions.
- Fault definitions.
- Service policies.
- WSDL Description of the interfaces.

The present document has been defined jointly between 3GPP TSG CT WG5, ETSI TISPAN and the Parlay Consortium.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 22.127: "Service Requirement for the Open Services Access (OSA); Stage 1".

[3] 3GPP TS 23.198: "Open Service Access (OSA); Stage 2".

[4] 3GPP TS 22.101: "Service aspects; Service principles".

[5] W3C Recommendation (2 May 2001): "XML Schema Part 2: Datatypes". http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/.

[6] 3GPP TS 29.199-1: "Open Service Access (OSA); Parlay X Web Services; Part 1: Common".

[7] Void.

[8] 3GPP TS 29.198-14: "Open Service Access (OSA) Application Programming Interface (API); Part 14: Presence and Availability Management (PAM)".

[9] RFC 3856: "A Presence Event Package for the Session Initiation Protocol (SIP)". http://www.ietf.org/rfc/rfc3856.txt

[10] Void.

[11] Void.

[12] 3GPP TS 23.141: "Presence service; Architecture and functional description; Stage 2".

[13] 3GPP TS 29.199-13: "Open Service Access (OSA); Parlay X Web Services; Part 13: Address list management".

[11] IETF RFC 3265: "Session Initiation Protocol (SIP)-Specific Event Notification".

[15] Void.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 29.199-1 [6] and the following apply:

applications: for Instant Messaging, Push to Talk, or call control and other purposes may become clients of the presence Web Service
We assume that these applications belong to a watcher and authenticate to the services in the name of the watcher.

identity: represents a user in the real world

NOTE: See OSA/Parlay PAM identities [8], section 4.4.1.

presence attributes: contain information about a presentity
An attribute has a name and a value and can be supplied by any device, application or network module that can be associated to the presentity's identity. A watcher can obtains attributes only after he has successfully subscribed to them. Examples for attributes are activity, location type, communication means, etc.

presence information: consists of a set of attributes that characterize the presentity such as current activity, environment, communication means and contact addresses
Only the system and the presentity have direct access to this information, which may be collected and aggregated from **several** devices associated to the presentity.

subscription: before a watcher can access presence data, he has to subscribe to it
One possibility the API provides is an end-to-end subscription concept, in which only identities that have accepted a subscription to their presence can be addressed. Subscriptions can be also automatically handled by server policies edited by the presentity or other authorized users. The service/protocol to manage those policies is out of the scope of the present document.

NOTE: This definition is not related to the term "subscription" in 3GPP TR 21.905 [1].

watcher and presentity: We use these names to denote the role of the client connected to the presence services
Like in OSA/Parlay PAM [8] the watcher and the presentity have to be associated to identities registered to the system, i.e. users, groups of users or organizations.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TS 29.199-1 [6] and the following apply:

ACL	Access Control List
DMS	Data Manipulation Server
GM	Group Management
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
ISC	IP multimedia subsystem Service Control interface
MMS	Multimedia Message Service
PAM	Presence and Availability Management
RLS	Resource List Server
SCF	Service Capability Feature

SIMPLE	SIP for Instant Messaging and Presence Leveraging Extensions
SIP	Session Initiation Protocol
SMS	Short Message Service
URI	Uniform Resource Identifier
WS	Web Service
WSDL	Web Services Definition Language
XCAP	XML Configuration Access Protocol
XML	eXtensible Markup Language
XMPP	eXtensible Messaging and Presence Protocol
XSD	XML Schema Definition

4 Detailed service description

The presence service allows for presence information to be obtained about one or more users and to register presence for the same. It is assumed that the typical client of these interfaces is either a supplier or a consumer of the presence information. An Instant Messaging application is a canonical example of such a client of this interface.

Figure 4.1 shows the architecture of the presence Web Service and the underlying services. The OSA/Parlay PAM SCF is the straightforward option and implements the presence server with extended identity, device capability and presence agent management. OSA/Parlay PAM allows aggregation of presence information from internet, mobile and enterprise users, etc. using a presence transport network of SIP or XMPP servers. The Presence Web Service can however communicate directly for example with IMS presence network elements (presence and resource list servers) using the ISC (SIP/SIMPLE) protocol interface.

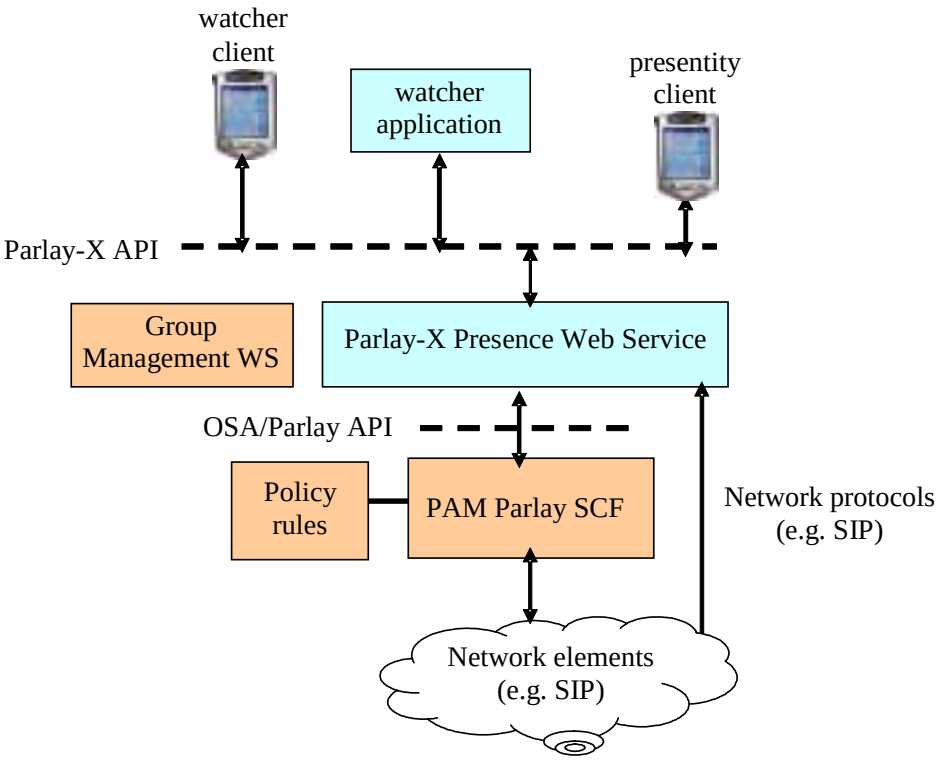


Figure 4.1: The PAM Web Service Environment

Relationship to Similar or Supplanted Specifications:

The most important relations are to:

- Parlay-X Terminal Status and Terminal Location: Both services deal with information that could be considered part of the user's presence information. Communication abilities can be derived from terminal status information, and the user's placetype can be derived from his location.
- OSA/Parlay PAM: The OSA/Parlay Presence and Availability specification can be considered the big brother of this specification. While ParlayX Presence stays behind OSA PAM in terms of flexibility and power - especially concerning attributes and management interfaces - it also extends PAM by introducing end-to-end authorization. This specification aims to be mappable to OSA PAM.
- SIP SIMPLE [9]: This specification aims to be mappable to the SIP/SIMPLE architecture.
- XMPP (Jabber): Many principles of this specification (see Bibliography) have been adopted, especially the end-to-end authorization.
- IETF Rich Presence (see Bibliography). The set of attributes the present document specifies is closely aligned with the IETF's Rich Presence ideas.
- Group Management [13]: Presence of groups is supported by this specification, however their creation and manipulation has to be done using the GM PX Web Service. In the 3GPP presence context, contact lists and group manipulation is done with the XCAP protocol (see Bibliography).

5 Namespaces

The PresenceConsumer interface uses the namespace:

http://www.csapi.org/wsdl/parlayx/presence/consumer/v2_3

The PresenceNotification interfaces use the namespace:

http://www.csapi.org/wsdl/parlayx/presence/notification/v2_3

The PresenceSupplier interfaces use the namespace:

http://www.csapi.org/wsdl/parlayx/presence/supplier/v2_3

The data types are defined in the namespace:

http://www.csapi.org/schema/parlayx/presence/v2_3

The 'xsd' namespace is used in the present document to refer to the XML Schema data types defined in XML Schema [5]. The use of the name 'xsd' is not semantically significant.

6 Sequence diagrams

6.1 Interface flow overview

The sequence diagram shows the interactions in case both watcher application and presentity are Web Service clients. Compared to the SIP interactions, the subscription notification is separated from the delivery of presence information itself. Based on the subscription result, the watcher can select the polling or notification mode for presence events. Changes in the authorization of presence attributes are propagated to the watchers via notifySubscription() message, the blocking of a subscription by the presentity is propagated via an endSubscriptionNotification message.

The sequence diagram does not show the internal communication within the presence server. It is assumed that the Presence Consumer and Supplier interfaces are implemented by the same instance. If implementers of the API find other solutions preferable, he has to take care of the internal communication himself.

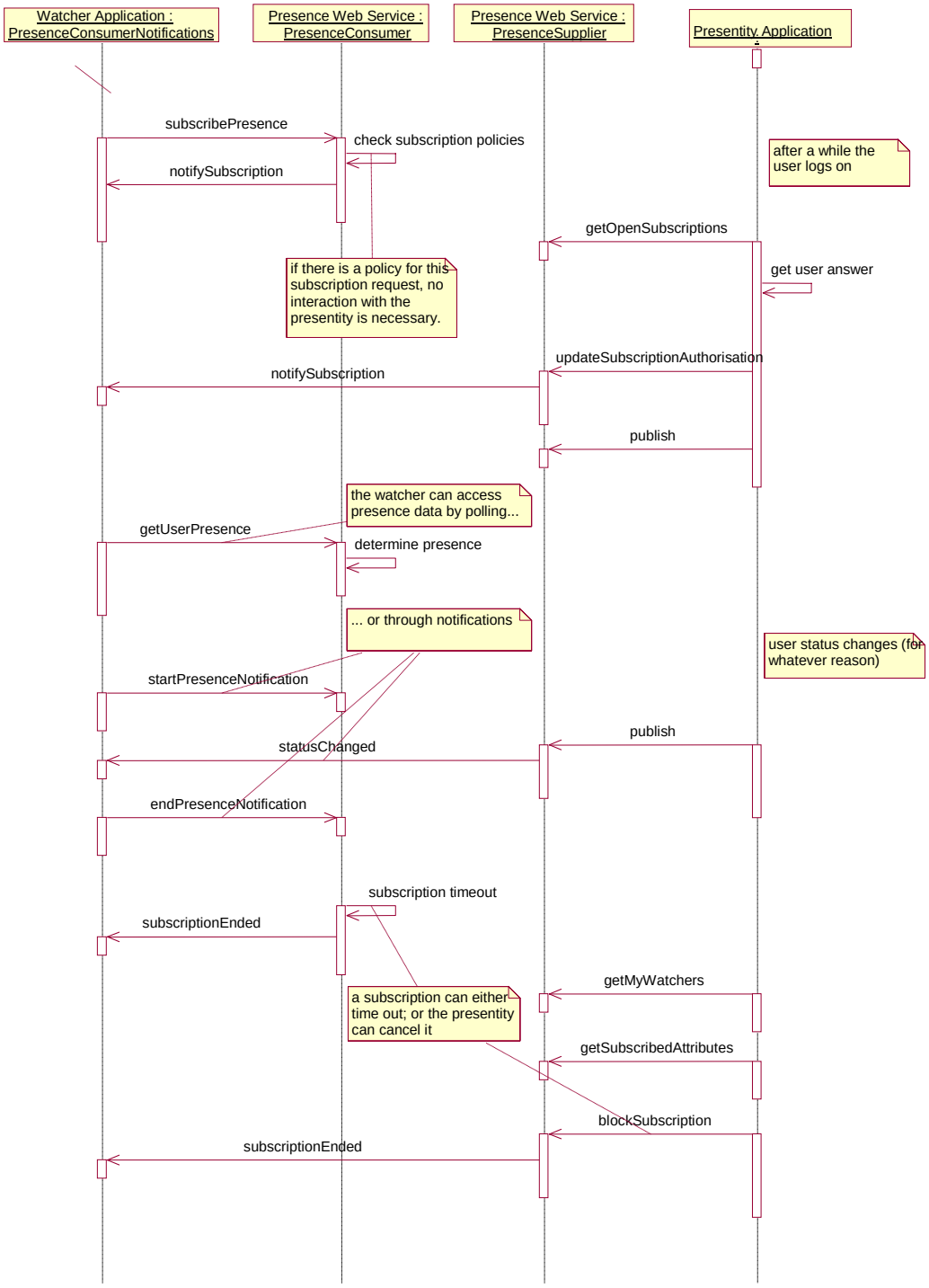


Figure : Message interaction overview

7 XML Schema data type definition

Presence attributes are inspired by the IETF's Rich Presence ideas (see Bibliography).

7.1 PresenceAttributeType enumeration

The different types of attributes. For each entry in this enumeration there is a separate value type.

Enumeration	Description
Activity	The presentity's activity (available, busy, lunch, etc.)
Place	At what kind of place the presentity is (home, office, etc.)
Privacy	The amount of privacy the user wants (public, quiet, etc.)
Sphere	The user's current environment (work, home)
Communication	The user's means of communication (phone, mail, etc.)
Other	A name - value pair for arbitrary presence information

7.2 ActivityValue enumeration

This enumeration shows the user's current activity. If the activity is unknown, the attribute value will be `ActivityNone`, meaning the attribute was not set. If the user is doing something not in this list, the value will be set to `ActivityOther`.

Enumeration	Description
ActivityNone	Not set.
Available	The user is available for communication.
Busy	The user is busy and is only available for urgent matters.
DoNotDisturb	The user is very busy and does not wish to be disturbed.
OnThePhone	The user is on the phone.
Steering	The user is driving a car / train / airplane, etc.
Meeting	The user is in a meeting.
Away	No idea what the user is doing, but he is away.
Meal	The user is eating.
PermanentAbsence	The user is away and will not return for an extended period.
Holiday	The user is on holidays.
Performance	The user is in a theatre / concert.
InTransit	The user is in the transit area of an (air)port.
Travel	The user is travelling.
Sleeping	The user is sleeping.
ActivityOther	The user is doing something not in this list.

7.3 PlaceValue enumeration

This enumeration shows the type of the user's current location. If the place type is unknown, the attribute value will be `PlaceNone`, meaning the attribute was not set. If the user in a place not in this list, the value will be set to `PlaceOther`.

Enumeration	Description
PlaceNone	Not set.
Home	The user is at home.
Office	The user is in an office.
PublicTransport	The user is on public transport.
Street	Walking on the street.
Outdoors	Generally outdoors.
PublicPlace	The user is in a public place.
Hotel	The user is in a hotel.
Theatre	The user is in a theatre or concert.
Restaurant	The user is in a restaurant / bar / etc.
School	The user is at school.
Industrial	The user is in an industrial building.
Quiet	The user is in a quiet area.
Noisy	The user is in a noisy area.
Aircraft	The user is on an aircraft.
Ship	The user is on a ship.
Bus	The user is in a bus.
Station	The user is in a bus- or railway station.
Mall	The user is in a mall.
Airport	The user is in an airport.
Train	The user is in a train.
PlaceOther	The user is in a kind of place not listed here.

7.4 PrivacyValue enumeration

This enumeration shows the amount of privacy a user currently has. If the privacy is unknown, the attribute value will be `PrivacyNone`, meaning the attribute was not set. If the privacy is not in this list, the value will be set to `PrivacyOther`.

Enumeration	Description
PrivacyNone	Not set.
PrivacyPublic	The user is surrounded by other people and cannot discuss openly.
PrivacyPrivate	The user is alone and able to talk openly.
PrivacyQuiet	The user is in a quiet environment and cannot talk at all.
PrivacyOther	None of the other values applies.

7.5 SphereValue enumeration

This enumeration shows the sphere within which the user acts. If the sphere is unknown, the attribute value will be `SphereNone`, meaning the attribute was not set. If the sphere is not in this list (neither work nor home), the value will be set to `SphereOther`.

Enumeration	Description
SphereNone	Not set.
SphereWork	The user is acting within his work sphere, i.e. as a member of his company
SphereHome	The user is acting within his home sphere, i.e. as a private person.
SphereOther	The user is acting neither within his work nor within his home sphere.

7.6 CommunicationMeansType enumeration

This enumeration lists communication means. If the communication attribute referrers to a means not in this list, it will point to `MeansOther`.

Enumeration	Description
Phone	The communication attribute refers to a phone (fixed line or mobile or SIP).
Chat	The communication attribute refers to a chat client.
SMS	The communication attribute refers to an SMS client.
Video	The communication attribute refers to a video phone (fixed line or mobile or SIP).
Web	The communication attribute refers to a web client.
EMail	The communication attribute refers to an e-mail client.
MMS	The communication attribute refers to an MMS client.
MeansOther	The communication attribute refers to any other client.

7.7 CommunicationMeans structure

This structure describes on way of reaching the presentity.

Element name	Element type	Optional	Description
Priority	xsd:float	No	The priority of this communication means. Between 0 and 1, the latter meaning the highest priority.
Contact	xsd:anyURI	No	The presentity's contact address for this communication means.
Type	CommunicationMeansType	No	The type of this communication means.

7.8 CommunicationValue structure

This structure describes the various ways of reaching a presentity.

Element name	Element type	Optional	Description
Means	CommunicationMeans [0..unbounded]	Yes	The different ways of reaching the presentity.

7.9 OtherValue structure

This structure can be used for storing arbitrary data about a presentity.

Element name	Element type	Optional	Description
Name	xsd:string	No	Description of the content.
Value	xsd:string	No	Attribute content.

7.10 PresenceAttribute structure

Presence data published by a presentity and retrieved by watchers.

Element name	Element type	Optional	Description
LastChange	xsd:dateTime	No	The time and date when the attribute was changed last.
Note	xsd:string	Yes	An explanatory note.
Type	PresenceAttributeType	No	Determines the type of the value field.
Value	One of the six value types; depends on field "type"	No	The actual value of the attribute.

This data structure is split into two types in the XSD file: A `PresenceAttribute` contains an `AttributeTypeAndValue`.

7.11 SubscriptionRequest structure

This structure is returned to the presentity by the PAM Web Service and contains the requesting watcher and the attributes he wants to subscribe.

Element name	Element type	Optional	Description
Watcher	xsd:anyURI	No	The watcher who wants to gain access to data.
Attributes	PresenceAttributeType [1..unbounded]	No	The attributes the watcher wants to see.
Application	xsd:string	No	The name of the application running on behalf of the watcher. Note that this field has solely informative purposes, access rights management is based on watcher id only.

7.12 PresencePermission structure

The answer from the service to the watcher in the notifySubscriptionRequest message.

Element name	Element type	Optional	Description
Attribute	PresenceAttributeType	No	The name of the attribute type the watcher wanted to subscribe
Decision	xsd:Boolean	No	Indicates whether the presentity accepted the subscription to the attribute type (true) or rejected it (false).

8 Web Service interface definition

This API is separated into three interfaces:

- PresenceConsumer interface: watcher methods for requesting and subscribing presence data.
- PresenceNotification interface: is the watcher notification interface for presence events.
- PresenceSupplier interface: presentity methods for supplying presence data and managing subscriptions.

8.1 Interface: PresenceConsumer

Client role: watcher.

This set of methods is used by the watcher to obtain presence data. After the subscription to presence data, the watcher can select between a polling mode or a notification mode in order to receive presence data.

8.1.1 Operation: subscribePresence

We assume that the watcher has been previously authenticated, so that his identity is known and can be associated with the subscription at the server.

The presentity is contacted and requested to authorize the watcher. As this process generally involves user interaction there cannot be an immediate response. The watcher is notified with `notifySubscription()`. If the presentity is a group, every member of the group will be contacted for authorization. The watcher will get one notification for each member.

Only after the subscription is completed (and the presentity has allowed access to attributes) may the watcher get information when he uses `getUserPresence()` or `startPresenceNotification()`.

The Reference part contains the Web Service reference that provides the information necessary for the Presence Supplier to notify the watcher with the results of the subscription request. Consistent with the definition provided in clause 12.4.1.7 of [6], the correlator element of this Web service reference should be an empty string because the presence attribute subscription logic in the watcher application is stateless.

NOTE: Pending the decision of the presentity concerning the subscription request, the watcher may have invoked the subscribePresence operation multiple times: i.e. for different presence attribute types. The response from the Presence Supplier thus may reflect the result of multiple, preceding subscription requests by the watcher.

At this interface level, the subscription has no expiration, although at can be ended from the presentity of the underlying layers (see subscriptionEnded method).

8.1.1.1 Input message: subscribePresenceRequest

Part name	Part type	Optional	Description
Presentity	xsd:anyURI	No	A presentity or a group of presentities whose attributes the watcher wants to monitor.
Attributes	PresenceAttributeType [0..unbounded]	Yes	The attribute types the watcher wants to access. (The same attribute types for all the group members). An empty array means subscription to all attribute types.
Application	xsd:string	No	Describes the application the watcher needs the data for.
Reference	common:SimpleReference	No	The notification interface.

8.1.1.2 Output message: subscribePresenceResponse

Part name	Part type	Optional	Description
None			

8.1.1.3 Referenced faults

ServiceException from 3GPP TS 29.199-1 [6]:

- SVC0001: Service error.
- SVC0002: Invalid input value.
- SVC0004: No valid addresses - if the presentity address does not exist.

PolicyException from 3GPP TS 29.199-1 [6]:.

- POL0006: Groups not allowed.
- POL0007: Nested groups not allowed.

8.1.2 Operation: getUserPresence

Returns the aggregated presence data of a presentity. Only the attributes which the watcher is entitled to see will be returned. This method does not support group identities.

Before getting these attributes, the watcher has to subscribe to them (see above). The presentity needs not be informed of the access, as he has already consented when the watcher called `subscribePresence()`.

8.1.2.1 Input message: getUserPresenceRequest

Part name	Part type	Optional	Description
Presentity	xsd:anyURI	No	The presentity whose data the watcher wants to see.
Attributes	PresenceAttributeType [0..unbounded]	Yes	The attribute types the watcher wants to see. An empty array means all attribute types.

8.1.2.2 Output message: getUserPresenceResponse

Part name	Part type	Optional	Description
Result	PresenceAttribute [0..unbounded]	Yes	The actual presence data.

8.1.2.3 Referenced faults

ServiceException from 3GPP TS 29.199-1 [6]:

- SVC0001: Service error.
- SVC0002: Invalid input value.
- SVC0004: No valid addresses - if the presentity address does not exist.

PolicyException from 3GPP TS 29.199-1 [6]. The presentity has the possibility to cancel or block a subscription by manipulating the policy rules. The exception informs the watcher about this status change.

- POL0002: Privacy error - if the watcher is not subscribed to the requested data.
- POL0006: Groups not allowed.

8.1.3 Operation: startPresenceNotification

The notification pattern with correlation is used in order to be able to correlate the notification events with the request. The Attributes message part specifies a subset of all possible attribute types that can be subscribed and can be used as a filter.

The watcher sets a notification trigger on certain user presence attribute changes. If the Attributes message part is empty, the watcher wants to be notified about changes to all subscribed attribute types.

In case the presentity is a group the watcher will receive notifications for every single member of the group. The watcher will only get notifications for those attributes and presentities he subscribed successfully prior to the call. The service will return a list of presentities where the notifications could not be set up.

The presentity needs not be informed of the access, as he has already consented when the watcher called `subscribePresence()`.

Note that the SimpleReference contains the correlator string used in subsequent messages to the notification interface.

8.1.3.1 Input message: startPresenceNotificationRequest

Part name	Part type	Optional	Description
Presentity	xsd:anyURI	No	The presentity (or group of presentities) whose attribute types the watcher wants to monitor.
Attributes	PresenceAttributeType [0..unbounded]	Yes	The attribute types the watcher wants to monitor. An empty array means monitoring of all attribute types.
Reference	common:SimpleReference	No	The notification interface
Frequency	common:TimeMetric	No	Maximum frequency of notifications (can also be considered minimum time between notifications). In case of a group subscription the service must make sure this frequency is not violated by notifications for various members of the group, especially in combination with `checkImmediate`.
Duration	common:TimeMetric	Yes	Length of time notifications occur for, do not specify to use default notification time defined by service policy.
Count	xsd:int	Yes	Maximum number of notifications. For no maximum, either do not specify this part or specify a value of zero.
CheckImmediate	xsd:boolean	No	Whether to check status immediately after establishing notification.

8.1.3.2 Output message: startPresenceNotificationResponse

Part name	Part type	Optional	Description
result	xsd:anyURI [0..unbounded]	Yes	The presentities for which the requested notifications could not be set up. Empty if notifications were set up for all the specified presentities.

8.1.3.3 Referenced faults

ServiceException from 3GPP TS 29.199-1 [6]:

- SVC0001: Service error.
- SVC0002: Invalid input value.
- SVC0004: No valid addresses - if the presentity URI does not exist.
- SVC0005: Duplicate correlator.

PolicyException from 3GPP TS 29.199-1 [6]. The presentity has the possibility to cancel or block a subscription by manipulating the policy rules. The exception informs the watcher about this status change.

- POL0001: Policy error.
- POL0004: Unlimited notifications not supported.
- POL0005: Too many notifications requested.
- POL0006: Groups not allowed.
- POL0007: Nested groups not allowed.

8.1.4 Operation: endPresenceNotification

Indicates that the watcher does not want further notifications for a specific notification request (identified by the correlator). Note that the subscription to presence data stays active; the caller of this method remains a watcher and can still use getUserPresence() or reactivate the notifications.

8.1.4.1 Input message: endPresenceNotificationsRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	The notification the watcher wants to cancel.

8.1.4.2 Output message: endPresenceNotificationResponse

Part name	Part type	Optional	Description
None			

8.1.4.3 Referenced faults

ServiceException from 3GPP TS 29.199-1 [6]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from 3GPP TS 29.199-1 [6]:

- POL0001: Policy error.

8.2 Interface: PresenceNotification

This client callback interface is used by the presence consumer interface to send notifications.

8.2.1 Operation: statusChanged

The asynchronous operation is called by the Web Service when an attribute for which notifications were requested changes.

8.2.1.1 Input message: statusChangedRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	Identifies the notification request
Presentity	xsd:anyURI	No	The presentity whose presence status has changed
ChangedAttributes	PresenceAttribute [1..unbounded]	No	The new presence data

8.2.1.2 Output message: statusChangedResponse

Part name	Part type	Optional	Description
None			

8.2.1.3 Referenced faults

None.

8.2.2 Operation: statusEnd

The notifications have ended for this correlator. This message will be delivered when the duration or count for notifications have been completed. This message will not be delivered in the case of an error ending the notifications or deliberate ending of the notifications (using endPresenceNotification operation).

8.2.2.1 Input message: statusEndRequest

Part name	Part type	Optional	Description
Correlator	xsd:string	No	Correlator provided in request to set up this notification

8.2.2.2 Output message: statusEndResponse

Part name	Part type	Optional	Description
None			

8.2.2.3 Referenced faults

None.

8.2.3 Operation: notifySubscription

This asynchronous method notifies the watcher that the server or the presentity handled the pending subscription.

NOTE: There is no correlator message part for reasons explained in clause 8.1.1; i.e. in the description of the subscribePresence operation.

8.2.3.1 Input message: notifySubscriptionRequest

Part name	Part type	Optional	Description
Presentity	xsd:anyURI	No	The presentity whose attribute types the watcher wants to monitor
Decisions	PresencePermission [0..unbounded]	Yes	Denotes the decision of the server/presentity to the subscription request for each attribute type. An empty array means subscription accepted for all requested attribute types.

8.2.3.2 Output message: notifySubscriptionResponse

Part name	Part type	Optional	Description
None			

8.2.4 Operation: subscriptionEnded

This asynchronous operation is called by the Web Service to notify the watcher (application) that the subscription has terminated. Typical reasons are a timeout of the underlying SIP soft state subscription (in accordance with [14] and [9]) or the decision of the presentity to block further presence information to that watcher. Since the subscription request has no expiration parameters, the service implementation may provide an inactivity timer that also triggers the subscriptionEnded message.

NOTE: There is no correlator message part for reasons explained in clause 8.1.1; i.e. in the description of the subscribePresence operation.

8.2.4.1 Input message: subscriptionEndedRequest

Part name	Part type	Optional	Description
Presentity	xsd:anyURI	No	The presentity to which the subscription has terminated
Reason	xsd:string	No	Timeout, Blocked

8.2.4.2 Output message: subscriptionEndedResponse

Part name	Part type	Optional	Description
None			

8.3 Interface: PresenceSupplier

These methods are used by the presentity to supply presence data and manage access to the data by its watchers. We assume that the presentity has been previously authenticated, so that his Identity is known.

8.3.1 Operation: publish

The presentity publishes data about herself. This data will then be filtered by the system and forwarded to the watchers who have ordered notifications.

8.3.1.1 Input message: publishRequest

Part name	Part type	Optional	Description
Presence	PresenceAttribute [0..unbounded]	Yes	The presence attributes the devices of the presentity supports

8.3.1.2 Output message: publishResponse

Part name	Part type	Optional	Description
None			

8.3.1.3 Referenced faults

ServiceException from 3GPP TS 29.199-1 [6]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from 3GPP TS 29.199-1 [6]:

- POL0001: Policy error.

8.3.2 Operation: getOpenSubscriptions

Called periodically by the presentity to see if any watchers wants to subscribe to presence data. The client will answer open requests with `updateSubscriptionAuthorization()`.

8.3.2.1 Input message: getOpenSubscriptionsRequest

Part name	Part type	Optional	Description
None			

8.3.2.2 Output message: getOpenSubscriptionsResponse

Part name	Part type	Optional	Description
result	SubscriptionRequest [0..unbounded]	Yes	Any open requests

8.3.2.3 Referenced faults

ServiceException from 3GPP TS 29.199-1 [6]:

- SVC0001: Service error.

PolicyException from 3GPP TS 29.199-1 [6]:

- POL0001: Policy error.

8.3.3 Operation: updateSubscriptionAuthorization

The presentity answers with this operation to watcher subscriptions for which no authorization policy exists. The answer consists of the attribute and the watcher involved and the permissions for each attribute. Subscription requests that are not answered are assumed pending.

The operation can be used by the presentity to change anytime the authorization for a certain watcher or group to monitor one or several attributes.

If the watcher did not try to subscribe the attribute - i.e. there is not pending subscription from this watcher to an attribute in the decisions array, a PresenceException will be raised and the entire authorization request ignored.

8.3.3.1 Input message: updateSubscriptionAuthorizationRequest

Part name	Part type	Optional	Description
Watcher	xsd:anyURI	No	watcher or group of watchers
Decisions	PresencePermission [1..unbounded]	No	The answers to open requests

8.3.3.2 Output message updateSubscriptionAuthorizationResponse

Part name	Part type	Optional	Description
None			

8.3.3.3 Referenced faults

ServiceException from 3GPP TS 29.199-1 [6]:

- SVC0001: Service error.
- SVC0002: Invalid input value.
- SVC0004: No valid addresses.
- SVC0220: NoSubscriptionRequest.

PolicyException from 3GPP TS 29.199-1 [6]:

- POL0001: Policy error.

8.3.4 Operation: getMyWatchers

Returns an array of watching identities that are subscribed to the presentity's attributes. They are not necessarily users of the notification system, the mere fact that they are allowed to see the presentity's attributes is enough to be on this list.

8.3.4.1 Input message: getMyWatchersRequest

Part name	Part type	Optional	Description
None			

8.3.4.2 Output message: getMyWatchersResponse

Part name	Part type	Optional	Description
Result	xsd:anyURI [0..unbounded]	Yes	The list of identities that currently have access to the presentity's attributes.

8.3.4.3 Referenced faults

ServiceException from 3GPP TS 29.199-1 [6]:

- SVC0001: Service error.

PolicyException from 3GPP TS 29.199-1 [6]:

- POL0001: Policy error.

8.3.5 Operation: getSubscribedAttributes

Returns an array of attributes that a specific watcher has subscribed.

8.3.5.1 Input message: getSubscribedAttributesRequest

Part name	Part type	Optional	Description
Watcher	xsd:anyURI	No	The watcher whose subscriptions the presentity wants to know

8.3.5.2 Output message: getSubscribedAttributesResponse

Part name	Part type	Optional	Description
Result	PresenceAttributeType [0..unbounded]	Yes	The attributes the watcher is subscribed to.

8.3.5.3 Referenced faults

ServiceException from 3GPP TS 29.199-1 [6]:

- SVC0001: Service error.
- SVC0004: No valid addresses.
- SVC0221: Not a watcher - if the URI in the field watcher is not a watcher of the presentity.

PolicyException from 3GPP TS 29.199-1 [6]:

- POL0001: Policy error.

8.3.6 Operation: blockSubscription

With this operation the presentity can block entirely the flow of presence information to a certain subscribed watcher by cancelling the subscription. The watcher will be notified with an subscriptionEnded() message.

8.3.6.1 Input message: blockSubscriptionRequest

Part name	Part type	Optional	Description
Watcher	xsd:anyURI	No	The watcher whose subscriptions the presentity wants to cancel

8.3.6.2 Output message: blockSubscriptionResponse

Part name	Part type	Optional	Description
None			

8.3.6.3 Referenced faults

ServiceException from 3GPP TS 29.199-1 [6]:

- SVC0001: Service error.
- SVC0002: Invalid input value.
- SVC0004: No valid addresses.
- SVC0221: Not a watcher - if the URI in the field watcher is not a watcher of the presentity.

PolicyException from 3GPP TS 29.199-1 [6]:

- POL0001: Policy error.

9 Fault definitions

9.1 ServiceException

From 3GPP TS 29.199-1 [6].

9.1.1 SVC0220: No subscription request

Name	Description
Message Id	SVC0220
Text	No subscription request from watcher %1 for attribute %2
Variables	%1 - watcher URI %2 - type of attribute, from clause 7.1

9.1.2 SVC0221: Not a watcher

Name	Description
Message Id	SVC0221
Text	%1 is not a watcher
Variables	%1 - watcher URI

10 Service policies

Name	Type	Description
MaximumNotificationFrequency	common:TimeMetric	Maximum rate of notification delivery (also can be considered minimum time between notifications)
MaximumNotificationDuration	common:TimeMetric	Maximum amount of time a notification may be set up for
DefaultNotificationDuration	common:TimeMetric	Default amount of time a notification will be set up for.
MaximumCount	xsd:int	Maximum number of notifications that may be requested
UnlimitedCountAllowed	xsd:boolean	Allowed to specify unlimited notification count (i.e. either by not specifying the optional Count message part in StartPresenceNotificationRequest or by specifying a value of zero)
GroupSupport	xsd:boolean	Groups may be included with addresses
NestedGroupSupport	xsd:boolean	Are nested groups supported in group definitions

Annex A (normative): WSDL of Presence API

The document/literal WSDL representation of this interface specification is compliant to 3GPP TS 29.199-1 [6] and is contained in text files (contained in archive 29199-14-650-doclit.zip) which accompanies the present document.

Annex B (informative): Bibliography

3GPP: "IETF Dependencies and Priorities". http://www.3gpp.org/tb/Other/IETF_archive_07.03.05/IETF.htm

draft-ietf-simple-event-filter-funct-05: "Functional Description of Event Notification Filtering". http://www.ietf.org/internet-drafts/draft-ietf-simple-event-filter-funct-05.txt. This version expires September 16, 2005. Also reference item #42 in 3GPP: "IETF Dependencies and Priorities".

draft-ietf-simple-rpid-10: "RPID: Rich Presence: Extensions to the Presence Information Data Format (PIDF)". http://www.ietf.org/internet-drafts/draft-ietf-simple-rpid-10.txt. This version expires June 23, 2006. Also reference item #54 in 3GPP: "IETF Dependencies and Priorities".

draft-ietf-simple-xcap-11: "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)". http://www.ietf.org/internet-drafts/draft-ietf-simple-xcap-11.txt. This version expires November 6, 2006. Also reference item #57 in 3GPP: "IETF Dependencies and Priorities".

Repository of information about the Extensible Messaging and Presence Protocol (XMPP), which was contributed by the Jabber Software Foundation (JSF) to the IETF. http://www.xmpp.org/

Annex C (informative): Change history

Change history								
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Cat**	**Old**	**New**
Sep 2004	CN_25	NP-040360	--	--	Draft v100 submitted to TSG CN#25 for Approval.	--	1.0.0	6.0.0
Dec 2004	CN_26	NP-040487	0001	--	Correct the Presence WSDL source code	F	6.0.0	6.1.0
Jun 2005	CT_28	CP-050162	0002	--	Correction of Presence.	F	6.1.0	6.2.0
Jun 2005	CT_28	CP-050162	0003	--	Update & Move Informative Document References to Bibliography	D	6.1.0	6.2.0
Jun 2005	CT_28	CP-050221	0004	--	Optionals for Part 14	F	6.1.0	6.2.0
Dec 2005	CT_30	CP-050579	0005	--	Clarify how to specify an unlimited notifications count	F	6.2.0	6.3.0
Dec 2005	CT_30	CP-050579	0006	--	Inconsistent part naming in PX response messages	F	6.2.0	6.3.0
Jun 2006	CT_32	CP-060203	0009	--	Apply Union data type element naming convention	F	6.3.0	6.4.0
Dec 2006	CT_34	CP-060594	0011	--	Corrections to text descriptions to remove ambiguity	F	6.4.0	6.5.0

History

Document history		
V6.1.0	December 2004	Publication
V6.2.0	June 2005	Publication
V6.3.0	December 2005	Publication
V6.4.0	June 2006	Publication
V6.5.0	December 2006	Publication