# Presence SIMPLE Requirements

Approved Version 1.1.1 – 25 Feb 2010

**Open Mobile Alliance**

OMA-RD-Presence_SIMPLE-V1_1_1-20100225-A

# Contents

# Figures

# 1. Scope

This document contains use-cases and requirements for a Presence Service, taking into considerations the demands of end-users, service providers, and system implementers.

# 2. References

## 2.1 Normative References


**[3GPP PS]** "Presence Service; (Release 6)", ch. 5-9. 3GPP TS 22.1412005.
URL: http://www.3gpp.org/ftp/Specs/archive/22_series/22.141/

**[3GPP PS STAGE2]** "Presence Service; Architecture and functional description (Release 6)", 3GPP TS23.141 2005,
URL: http://www.3gpp.org/ftp/Specs/archive/23_series/23.141/

**[3GPP2 PS]** "Presence for Wireless Systems; Stage 1 Requirements". 3GPP2 S.R0062 Version 1.0, October 2002.
URL: http://www.3gpp2.org/Public_html/specs/S.R0062-0_v1.0.pdf

**[Privacy]** "OMA Privacy Requirements for Mobile Services", Open Mobile Alliance™, OMA-RD_Privacy-V1_0_0, Version 1.0.0,
URL: http://www.openmobilealliance.org

**[RFC2119]** "Key words for use in RFCs to Indicate Requirement Levels". S. Bradner. March 1997.
URL: http://www.ietf.org/rfc/rfc2119.txt

**[RFC2778]** "A Model for Presence and Instant Messaging", RFC2778, M. Day et al., February 2000,
URL: http://www.ietf.org/rfc/rfc2778.txt

**[RFC3261]** "SIP: Session Initiation Protocol", Rosenberg et al, June 2002,
URL: http://www.ietf.org/rfc/rfc3261.txt

**[XDMREQ]** "XML Document Management Requirements", Open Mobile Alliance™, OMA-RD-XDM-V1_1, Version 1.1,
URL: http://www.openmobilealliance.org


## 2.2 Informative References


**[3GPP GM]** "IP Multimedia Subsystem (IMS) group management, (Release 6)", ch. 5-7. 3GPP TS 22.250 2002.
URL: http://www.3gpp.org/ftp/Specs/archive/22_series/22.250/

**[3GPP IM]** "IP Multimedia System (IMS) Messaging; (Release 6)", ch. 6-11.3GPP TR 22.3402005.
URL: http://www.3gpp.org/ftp/Specs/archive/22_series/22.340/

**[3GPP Security]** "Presence Service; Security", 3GPP TS 33.141, 2004.
URL: http://www.3gpp.org/ftp/Specs/archive/33_series/33.141/

**[3GPP2 IM]** "Wireless Immediate Messaging; Stage 1 Requirements". 3GPP2 S.R0061 Version 1.0. October 2002.
URL: http://www.3gpp2.org/Public_html/specs/S.R0061-0_v1.0.pdf

**[3GPP2 Security]** S.R0092-A, "IMS Security Framework", 3GPP2, July 2004,
URL: http://www.3gpp2.org/Public_html/specs/S.R0086-A_v1.0_040614.pdf

**[RFC2779]** "Instant Messaging/Presence Protocol Requirements", M. Day et al., February 2000,
URL: http://www.ietf.org/rfc/rfc2779.txt

# 3. Terminology and Conventions

## 3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

## 3.2 Definitions

| | |
|---|---|
| **Application-specific** | A qualifier that designates a presence element that is either specific to a communication means (such as PoC or IM), or pertains to an application (such as a networked game application). |
| **Application-specific availability** | Available indicates that it is possible to initiate a communication of this type; “Not Available” indicates that it is not possible to initiate a communication of this type. For example, if a user is provisioned with the PoC Service, within coverage, has an appropriate handset, etc., they would be available for PoC, whereas if any of those were not true, they would be ”Not Available”.<br><br>Note: this is mostly unrelated to whether the user is willing or not to accept this particular type of communications. “The Application-specific availability would be supplied by various network elements; and is commonly referred to as “network presence. |
| **Application-specific willingness** | Indicates whether the user is willing to accept communications of this type. If the application-specific availability is set to “Not Available” this presence element has no relevance. If this presence element is not present in a presence document, it may be deduced from the “Default Willingness” presence element [see Default Willingness definition]. This value of this presence element may be overridden by the “Overriding Subscriber Willingness” presence element [see Overriding Subscriber Willingness definition]. |
| **Authorization Categories** | An Authorization Category consists of a list of watchers that identifies a trust relation like Family, Friends or Colleagues. |
| **Default willingness** | Indicates the end-users default willingness to communicate in the absence of an application specific willingness presence element. For example, if a user is “Available” on a particular application, but has not published their willingness for that application, the default value would be used. This presence element, where applicable, may still be overridden by the “Overriding Subscriber Willingness” presence element [see Overriding Subscriber Willingness definition]. |
| **One-time Event Subscription and Notification** | One-time Event Subscription and Notification is the feature that enables subscribed-watchers to place a subscription that will generate a single notification and then terminate itself. |
| **Overriding Subscriber Willingness** | The Overriding Subscriber Willingness provides an indication, set by an end user, that takes precedence over both the Application-specific willingness and the Default willingness settings. For example, when a Overriding Subscriber Willingness indication is present, a positive setting indicates that the user is willing to accept communications for all available communications types, while a negative setting indicates that the user is not willing to accept any communications. |
| **Presence** | Ambiguous term. Not used within OMA specifications. |
| **Presence Enabled Phonebook** | A convenient way of referring to a client displaying presence information about one or more presentities. This is a generic name, not mandating nor implying any particular implementation or set of features. |
| **Presence Information** | Dynamic set of information pertaining to a Presentity that may include presence elements such as the status, reachability, willingness, and capabilities of that Presentity.<br><br>Note: This definition is compatible with the 3GPP/3GPP2 definitions, as well as the IETF definition, though the latter is quite generic. |
| **Presence Information Element** | A basic unit of Presence Information. |
| **Presence Server** | A logical entity that receives Presence Information from a multitude of Presence Sources pertaining to the Presentities it serves and makes this information available to Watchers according to the rules associated with those Presentities. |

| | |
|---|---|
| | Note: In IETF SIMPLE Presence a Presence Server is referred to as a Presence Agent. |
| **Presence Source** | A logical entity that provides Presence Information pertaining to exactly one or more Presentities to the Presence Server. 3GPP/3GPP2 Presence User Agents, Presence Network Agents, and Presence External Agents are examples of Presence Sources.<br>Note: In IETF SIMPLE Presence, Presence Sources are referred to as Presence User Agents. In IETF 2778, they are referred to as Presentities. |
| **Presentity** | A logical entity that has Presence Information (see definition below) associated with it. This Presence Information may be composed from a multitude of Presence Sources. A Presentity is most commonly a reference for a person, although it may represent a role such as “help desk” or a resource such as “ conference room #27”. The Presentity is identified by a SIP URI, and may additionally be identified by a tel URI or a pres URI .<br>Note: This definition maps better to the RFC2778 definition of a Principal, rather than that of RFC2778 Presentity. This definition is compatible with the 3GPP/3GPP2 definitions of presentity, as well as that of IETF SIMPLE Presence. |
| **Principal** | Not used within OMA Presence specifications.<br>Note: Defined in RFC2778. In OMA/3GPP/3GPP2 a Presentity resembles an IETF Principal. |

**The following definitions are referenced from [RFC2778].**

| | |
|---|---|
| **Communication address** | Consists of communication means and contact address. |
| **Communication means** | Indicates a method whereby communication can take place. Instant message service is one example of a communication means. |
| **Contact address** | A specific point of contact via some communication means. When using an instant message service, the contact address is an instant inbox address. |
| **Notification** | A message sent from the presence service to a subscribed-watcher when there is a change in the presence information of some presentity of interest, as recorded in one or more subscriptions. |
| **Subscription** | The information kept by the presence service about a subscribed-watcher’s request to be notified of changes in the presence information of one or more presentities. (Source: [RFC2778], [3GPP-TS_22.141])<br>NOTE: This definition represents an entity’s request to obtain Presence Information, and is not related to the term “subscription” in [3GPP-TS_22.250]. |

**The following definitions are referenced from [3GPP PS].**

| | |
|---|---|
| **Fetcher** | A form of watcher that has asked the presence service for the presence information of one or more presentities, but is not requesting a notification from the presence service of (future) changes in a presentity’s presence information. (Differs slightly from [RFC2778] definition). (Identical to [3GPP2 PS]). |
| **Poller** | A fetcher that requests presence information on a regular basis. (Identical to RFC2778) and [3GPP2 PS] definitions). |
| **Presence service** | The capability to support management of presence information between watchers and presentities, in order to enable applications and services to make use of presence information. (Differs from [RFC2778] definition, identical to [3GPP2 PS].) |
| **Subscribed-watcher** | A type of watcher, which requests notification from the presence service of changes in a presentity’s presence information, resulting in a watcher-subscription, as they occur in the future. (Source: [3GPP-TS_22.141])<br>NOTE: In [RFC2778], Subscribed-watchers are referred to as subscribers. |
| **Watcher** | Any uniquely identifiable entity that requests presence information about a presentity from the presence |

service. Special types of watcher are fetcher, poller, and subscribed-watcher. (Differs slightly from [RFC2778] and [3GPP2 PS] definitions).

**Watcher information** Information about watchers that have received or may receive presence information about a particular presentity within a particular recent span of time. (Differs slightly from [RFC2778], is identical to [3GPP2 PS] definition).

**The following definitions are referenced from [3GPP PS STAGE2].**

**Presence network agent** A network located element that collects and sends network related presence information on behalf of the presentity to a presence server. This is a type of *presence source*.

**Presence user agent** A terminal or network located element that collects and sends user related presence information to a presence server on behalf of a presentity. This is a type of *presence source*. (Differs from [RFC2778] definition).

**The following definitions are referenced from [3GPP2 PS].**

**Interoperable services** Two implementations are interoperable if they can interact without protocol interworking devices.

**Provisioning** An action taken by the service provider to make the presence service available to a user. Provisioning may be general, where the service may be made available to all users without prior arrangements being made with the service provider, or it may be pre-arranged, where the service is made available to an individual user only after the necessary arrangements (e.g., login name, password) have been made with the service provider.

## 3.3 Abbreviations

| | |
|---|---|
| **3GPP** | 3rd Generation Partnership Project |
| **3GPP2** | 3rd Generation Partnership Project 2 |
| **CIPID** | Contact Information in Presence Information Data Format |
| **DND** | Do Not Disturb |
| **FIFO** | First In, First Out |
| **GGSN** | Gateway GPRS Support Node |
| **GMT** | Greenwich Mean Time |
| **GPRS** | General Packet Radio Service |
| **IETF** | Internet Engineering Task Force |
| **IM** | Instant Messaging |
| **IMPS** | Instant Messaging and Presence Service (aka Wireless Village) |
| **IMS** | IP Multimedia Subsystem |
| **IP** | Internet Protocol |
| **ISDN** | Integrated Services Digital Network |
| **MMS** | Multimedia Messaging Service |
| **MSISDN** | Mobile Station International ISDN Number |
| **OMA** | Open Mobile Alliance |

| | |
|---|---|
| **PEP** | Presence Enabled Phonebook |
| **PoC** | Push to talk Over Cellular |
| **RFC** | Request For Comments |
| **RPID** | Rich Presence Information Data |
| **SGSN** | Serving GPRS Support Node |
| **SIMPLE** | SIP for Instant Messaging and Presence Leveraging Extensions |
| **SIP** | Session Initiation Protocol |
| **SMS** | Short Messaging Service |
| **UMTS** | Universal Mobile Telecommunications System |
| **URL** | Universal Resource Locator |
| **VIP** | Very Important Person |
| **VMS** | Voicemail Service |
| **VoIP** | Voice over IP |
| **WV** | Wireless Village (aka IMPS) |
| **XCAP** | XML Configuration Access Protocol |
| **XML** | Extensible Markup Language |

# 4. Introduction

## 4.1 OMA Presence Mandate

SIP/SIMPLE by IETF is accompanied with a set of Internet Drafts and RFCs (such as RPID, Rich Presence Information Data Format; CIPID, Contact Information in Presence Information Data Format; and XCAP, XML Configuration Access Protocol). 3GPP and 3GPP2 specify the practical implementations of IETF specifications in IMS (IP Multimedia Subsystem) and MMD (MultiMedia Domain) respectively. Both transport IP traffic and use SIP as signalling protocol, and are commonly known as SIP/IP Core. On top of all this, OMA SIMPLE presence service specifications, developed in REQ and PAG WGs, define a SIP/SIMPLE-based Presence Service.

The specifications mentioned above leave a number of degrees of freedom open; for example, the tuples as defined by IETF can be used any which way to convey Presence Information. Not even the Presence Information content is specified. The situation is akin to having a functioning phone line but no common language between the conversing parties. This makes it difficult to achieve interoperability between different vendor's products.

OMA's role is to create application level specifications for Presence Service. This includes presence information semantics and guidelines for presence applications, please see Figure 1. These specifications shall be agnostic to the underlying network technology, be it specified by 3GPP, 3GPP2, or by somebody else.

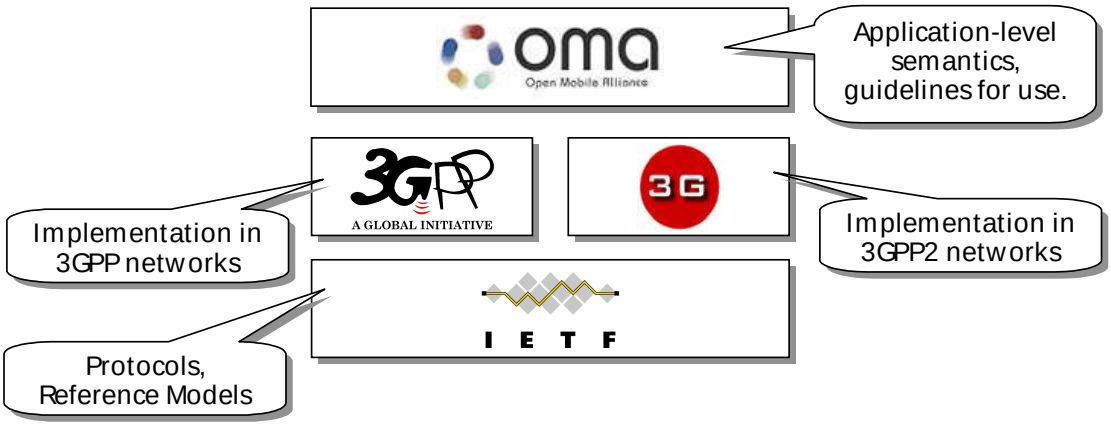



**Figure 1. Presence specification layers**

## 4.2 Service Overview

A Presence Service is a software system whose role is to collect and disseminate presence information, subject to a wide variety of controls. The requirements established into this document fall into two categories:

- Requirements pertaining to the mechanisms utilized in collecting and disseminating presence information, including the means to do so in a controlled way (e.g. publish, subscribe, notify, etc.)
- Requirements pertaining specific types of presence information content (e.g. willingness to communicate, device/application status, etc.)

Note that the Presence Service features are not limited to any particular type of presence information. However, in the context of this specification we will only be standardizing a limited set of presence elements.

## 4.3 Presence Information, Sources and Watchers

Wireless Village specifications define a set of attributes that convey various properties of a human user, such as UserAvailability and StatusText. An extension mechanism is also specified. SIP/SIMPLE was originally designed from

communication channel point of view, e.g., to express whether a certain communication channel (voice, SMS, etc) is available, and what the priority of that channel is. SIP/SIMPLE has been extended towards more Wireless Village –like personal attributes (e.g. RPID).

Developing the Presence Service concept requires thinking about presence-related issues in broader terms still because both WV and SIP/SIMPLE do only part of the job. We need to be able to convey information about the person using Presence Service, but also about the communication means he or she uses. Further, there may be presentities and watchers that do not correspond to any human being at all, but to a service. Finally, also network elements may produce and consume Presence Information.

Figure 2 illustrates the presence information sources and watchers that need to be taken into account in OMA specifications:

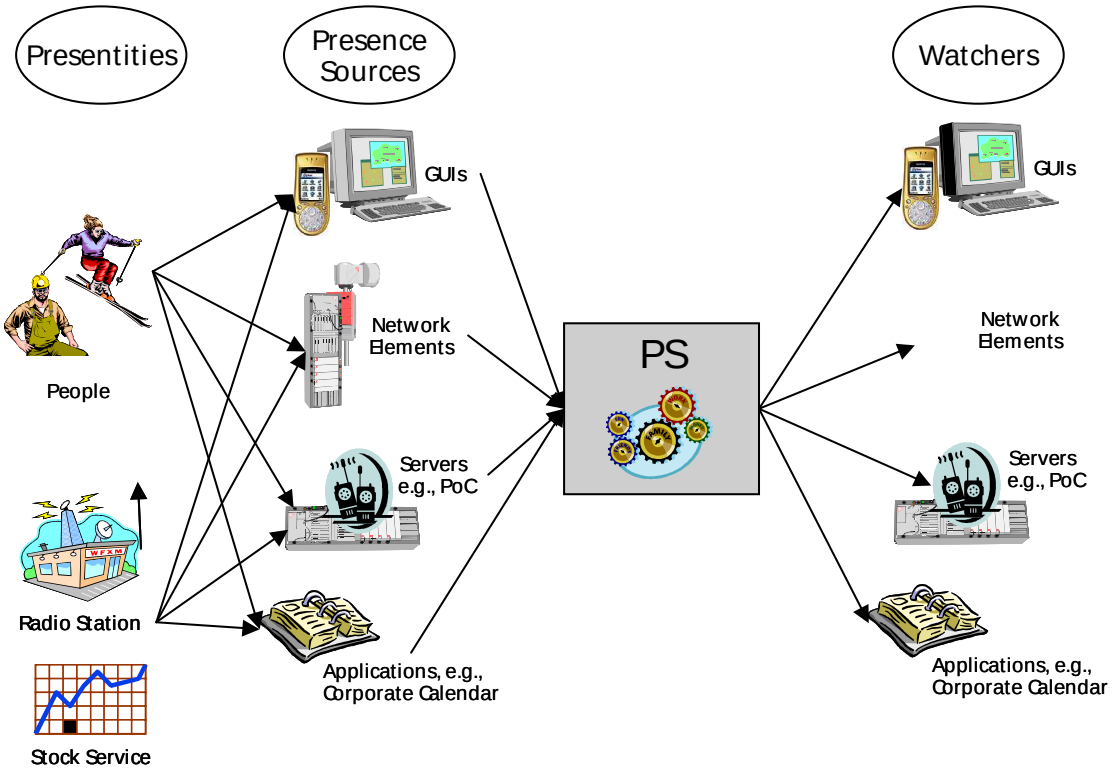



**Figure 2. Presence Service components.**

- *People*, human presence users, publish their personal presence information. To do this, a user may use an application in her mobile phone or a desktop application. Please note that the presentity's state may also include communication channels' states, such as VoiP, video, or PoC.

- *Non-human presentities* may also publish presence information, for example, a radio station may want to publish the song currently playing, and a call center might publish information about congestion situation such as waiting line size and expected waiting time.

- *Network elements*, for example, may produce presence information pertaining to a person, for example, whether a person is registered to the network or not. Networks elements may also consume Presence Information.

- Yet another group of presence information sources and watchers are *application servers* and *applications* in the network. For instance, a corporate calendar application could update an employee's availability information based on behaviour.

## 4.4 Presence Information Processing

Figure 2 is simplified in the sense that Presence Server is more than a mere relay station for the Presence Information. There are several stages of processing that need to take place before the Presence Information can be disseminated further, please see Figure 3.

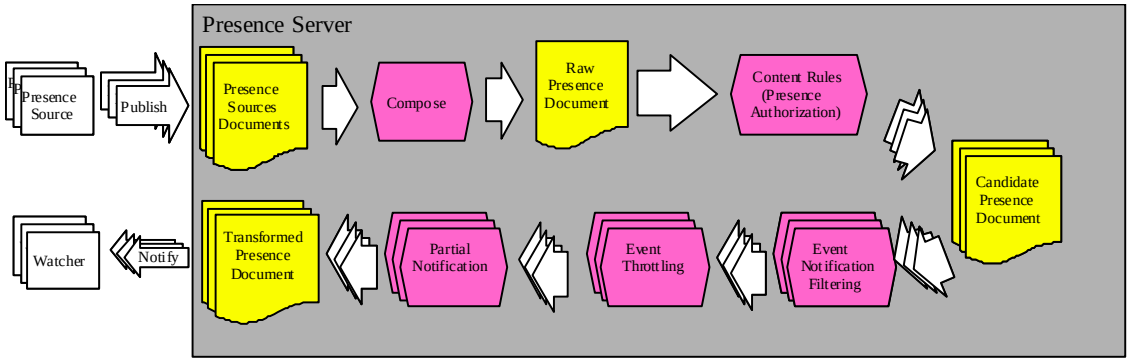



**Figure 3. Processing of presence information**

The Compose function takes the presence information originating from several Presence Sources but pertaining to one Presentity and, according to composition rules, creates a raw presence document. After that, authorization is performed for all watchers subscribed to the Presentity's Presence Information, and the presence document is changed accordingly. This is necessary because not all watchers are intended to see the same presence elements; this stage creates several parallel documents. Each of these documents is subjected to per-watcher or per-watcher-group transformation, which includes, e.g., filtering and partial notification.

The resulting notifications are passed outside Presence Service; after this, the number of notifications may be cut down by throttling, if such a feature is supported.

## 4.5 Communicating preferences to end-users using Presence Information

One of the purposes of publishing presence information about an end user to authorized watchers is to set appropriate expectations among communications partners, thereby increasing the chance of successful communications using the most suitable or preferred communications means.

To set the correct expectations to potential communicating partners, a user's presentity may provide different presence information to different watchers, for instance "willing/available" to some and "unwilling/not available" to others.

In addition, a watcher may receive a status of "unwilling/unavailable" that applies to one or more, or all communication means. As such, presentities can communicate which types of incoming sessions they want to accept.

Furthermore, the presence service provides certain mechanisms to presence sources. For example, presence state may be associated with a certain validity duration, in order to communicate its "freshness".

Thus, watchers should ensure they understand and use those mechanisms in order to interpret the presence information in an consistent and accurate manner.. Additionally, when a user agent renders presence information to be viewed by a end-user, it should do so in a way that accurately portrays the received presence information. For example, if a certain element of presence information expires, then the user agent should present this appropriately to the user.

One of the roles for the requirements in this document is to facilitate the definition of presence information and the publication and notification of that information in order to convey it to potential partners in communications, such that both parties can consistently interpret this this information and have the correct expectations for any subsequent communications.

## 4.6 Requirements Fulfilled

All the requirements as defined in the present RD are met in the present Enabler Release apart from those listed below.

| Requirement ID/Number | Phase Met |
|---|---|
| 6.1.3.2 #3, #4, #23, #26 | Phase not specified |
| 6.1.3.3 #4 | Phase not specified |
| 6.1.6 #2 | Phase not specified |
| 6.1.6 #3 | Phase not specified |
| 6.1.3#1 | Not met |
| 6.1.3.1 #2, #5, #8 | Not met |
| 6.1.3.5 | Not met |
| 6.1.4.1 #3 | This requirement is met through the use of publication expiration. However this expiration is applied to the publication as a whole and cannot be applied to individual elements |
| 6.1.4.2 #3 | Partially met |
| 6.1.9.1 | The details are outside the current scope of the specification |
| 6.1.9.2 | The details are outside the current scope of the specification |

# 5. Use Cases (Informative)

The following use cases are provided to further illustrate the functions and roles of the various system elements in the Presence framework and the inter-related functions performed by the Presence Server

## 5.1 Provisioning

### 5.1.1 Setting Up My Presence Service

#### 5.1.1.1 Short Description

This use case shows how mobile subscribers configure their Presence Service preferences (e.g. preferred means of communication, blocking of a user). The mobile subscriber is able to indicate and configure how he/she wants to be contacted (e.g. indicate to others that she wants to be contacted through Instant Messaging when he/she is in a meeting) and to which person(s) he/she does not want to provide his/her presence information.

#### 5.1.1.2 Actors

- Alice – Configures the presence service according to her preferences
- Bob – Wants to get Alice's presence information and contact her
- John – Wants to get Alice's presence information and contact her
- Presence Client (PC) – Resides in the mobile device of Alice, Bob and John
- Instant Messaging Client – Resides in the mobile device of Bob and Alice
- Presence Enabled Phonebook – Resides in mobile devices of Bob and John and is able to get presence information from the Presence Client (PC)
- Presence Server – Resides in the network

#### 5.1.1.3 Actor Specific Issues

Alice wants:

- To configure her presence service in order to indicate that she wants to be contacted only by Instant Messaging, when she is in a meeting
- To configure her presence service in order not to be contacted by Bob at any time, without him knowing that he is blocked (polite blocking)

Bob wants to:

- Get in touch with Alice when she becomes available

John wants to:

- Get in touch with Alice regardless of the means of communication (e.g. messaging, voice)

The Presence Service to work automatically behind the scenes, requiring a minimal amount of user interaction

#### 5.1.1.4 Actor Specific Benefits

Alice:

- Is able to be contacted by John according to her preferences (only by Instant Messaging when she is in a meeting)
- Is able to block John, without him knowing that he is blocked

Bob:

- Is able to get in touch with Alice

#### 5.1.1.5 Preconditions

- Alice, Bob and John are all provisioned to use the Presence Service
- All the actors are able to make changes in their presence information by themselves (i.e. they are in coverage area all the time)

#### 5.1.1.6 Postconditions

- Alice is successfully contacted with Instant Messaging when she is in a meeting
- Bob does not manage to contact Alice
- John manages to contact Alice with Instant Messaging while she is in a meeting

#### 5.1.1.7 Normal Flow

1) Alice invokes her Presence Service Settings menu in her mobile device
2) Alice chooses the “Preferences” option and in the “Profiles” option defines a new profile named “in a meeting”
3) Alice inside the “in a meeting” profile chooses the “Communication preferences” option and defines that she wants to be contacted only by “Instant Messaging” when she is using this profile
4) Alice chooses again the “Preferences” option and in the “Contact List” option chooses Bob’s entry
5) Alice defines that she does not want Bob to have access to her Presence information and “blocks” him
6) Bob invokes his PEP in his mobile device and chooses Alice’s entry trying to retrieve Alice’s presence information from the Presence Client (PC) in her mobile device
7) As Alice has “blocked” him and the presence information he receives is fake and he sees that Alice is “not available” although in reality she is
8) John tries to retrieve Alice’s presence information from his PEP in his mobile device
9) John sees that Alice is “in a meeting” and the only indicated way he can get in touch with her is through “Instant Messaging”
10) John sends her an instant message asking her what time she finishes her meeting
11) Alice replies back to him with an instant message informing him about the finishing time of meeting

#### 5.1.1.8 Alternative Flow

None

#### 5.1.1.9 Operational and Quality of Experience Requirements

- Bob does not realize that he has been “blocked” and thinks that Alice is not available
- Alice is able to modify her “Profile” Settings according to her preferences in an easy way
- Alice is not contacted with other means of communication apart from the ones indicated (in the example of being “in a meeting” only by Instant Messaging)

# 5.2 Basic Presence Usage

## 5.2.1 Sharing Presence Information A

This use case will demonstrate how a user can share their presence information with presence aware applications and users. This will include how to make their presence information available and how to manage the authorisation of how/what can use their presence information and what piece(s) of their presence information can be accessed.

### 5.2.1.1 Short Description

In this use case the user wants to share his presence information with applications that require or are enhanced with presence information. This use case focuses on a phonebook application which requests presence information and is authorized access to this information.

### 5.2.1.2 Actors

Bob – The owner of the presence information

Sue – User of a PEP application

### 5.2.1.3 Actor Specific Issues

Bob is subscribed to a presence service and can provide presence information about his availability and status to applications that are presence enabled. In this use case Bob gets a request from Sue's PEP application requesting that he share his presence information. Bob will need to authorise the phonebook application to receive his presence information, what presence information is to be shared, and for how long it can receive presence information without renewing its authorisation from Bob.

Sue is Bob's new business colleague and she has added his contact information to her phonebook. Her phonebook has the ability to receive and display presence information if authorised to do so. She will indicate in Bob's information to also display his presence information.

### 5.2.1.4 Actor Specific Benefits

Bob will provide better contact information to his co-workers and reduce missed calls.

Sue will not only have Bob's contact information, but will also be able to see if he is reachable and how.

### 5.2.1.5 Preconditions

Bob and Sue both have a subscription to a presence service.

Sue will need to have a PEP service.

Sue knows how to enable the presence-related features in her phonebook. This may require her to know the presence service provider information (presence server) and any other needed information.

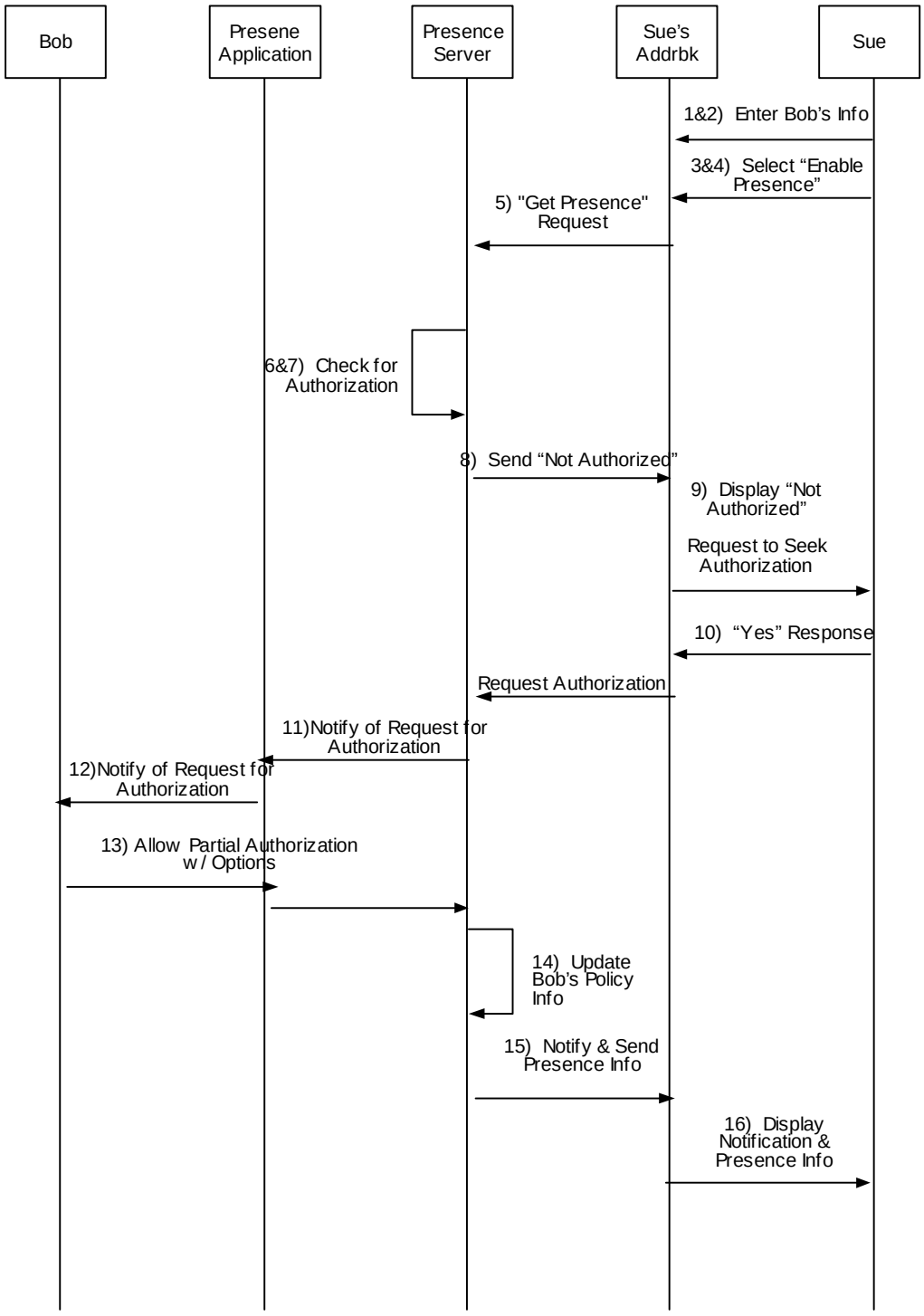
### 5.2.1.6 Postconditions

Sue will be able to see Bob's presence information when she looks in her phonebook for him. This will allow Sue to know the best method for contacting Bob.

### 5.2.1.7 Normal Flow

1) Sue and Bob have recently exchanged business contact information due to a project they will be working on together.
2) Sue enters Bob's information into her PEP.

3) Once Sue has entered all of Bob's information into her phonebook she selects the "enable presence" feature.
4) Sue then enters the necessary information to allow her phonebook application to contact the presence server providing Bob's presence service. Once all needed information is input Sue selects "complete".
5) The phonebook application will contact Bob's presence server asking for Bob's presence information.
6) Bob's presence server will check to see if Sue's phonebook is authorised to receive this information.
7) Sue's phonebook is not currently approved to receive Bob's presence information.
8) Since Sue's phonebook does not have authorisation for this information the presence server will send a response to Sue's phonebook application that it is not authorised for this information, along with an indication to Sue's phonebook asking if it would like to attempt to receive authorisation.
9) The phonebook provides and indication to Sue that it is not authorised for this information, and asks if she would like to attempt to receive authorisation.
10) Sue selects "yes" and the phonebook sends a request to the presence server to attempt authorisation.
11) When the presence server receives the request from Sue's phonebook it sends a notification to Bob that Sue's phonebook would like to receive his presence information.
12) The presence service provided to Bob by the presence server will provide Bob options to deny the request, fully accept the request, or partially accept the request (along with granular options to select what information is to be made available). "Deny" will cause the presence server to not provide presence information to Sue; "accept" will provide all of Bob's presence information to Sue; and "partially accept" will allow Bob to select what presence information is provided to Sue.
13) Bob selects "Partially Accept" and inputs his settings and conditions
14) The presence service receives Bob's input and updates its policy information on Bob regarding what can be presented to Sue's phonebook and for how long.
15) The presence server then sends a notification to Sue's phonebook approving its authorisation request, and sends the authorised presence information to Sue's phonebook.
16) The phonebook will notify Sue of the authorisation and will present the presence information about Bob in his entry in Sue's phonebook.



#### 5.2.1.8 Alternative Flow

An alternative flow would be that Bob was not available when the phonebook attempted authorisation. If that is the case then there would be a need for the presence server to store the request until Bob was available. This would cause a few extra steps to be inserted between steps 10 and 11 to basically store the request until Bob becomes available and to notify Sue's phonebook that it is attempting authorisation, but that it will be delayed until Bob is available.

#### 5.2.1.9 Operational and Quality of Experience Requirements

By granting his business related presence information to Sue they are able to better coordinate their work activities. Sue is able to know when Bob is available and how best to contact him. This will make their professional interaction more efficient and organized, hopefully reducing costs and improving overall performance.

### 5.2.2 Sharing Presence Information B

This use case will demonstrate how a user can share their presence information with presence aware applications. This will include how to make their presence available and how to manage the authorisation of who can access their presence and what piece(s) of their presence information can be accessed. Presence may be shared from the presence server to another application server (e.g. PoC Server). This use case uses a generic application as the application with which to share presence information.

#### 5.2.2.1 Short Description

In this use case the user wants to share his presence information with applications that require or are enhanced with presence information. There are several ways that presence information can be shared. In this use case the presence information owner initiates the sharing of presence information.

#### 5.2.2.2 Actors

Bob – The owner of the Presence information

The Team – Bob's friends and business colleagues. They will be provisioned to the generic application service and will benefit from Bob authorizing the application access to his presence information.

Presence Server – This is the presence server associated with Bob's Presence information

Application Server – The Server that will receive Bob's presence information

#### 5.2.2.3 Actor Specific Issues

Bob has presence information about his availability and status. In this use case Bob needs to authorise an application server to receive his presence information.

The Team would like to be able to tell when Bob is available and able to be contacted. They will enjoy the benefits of a presence enabled service.

#### 5.2.2.4 Actor Specific Benefits

The value to Bob is that the application will be able to post his availability to his colleagues and friends.

#### 5.2.2.5 Preconditions

Bob is provisioned to both a Presence service and an application service

Bob's application service is subscribed to Bob's presence information.

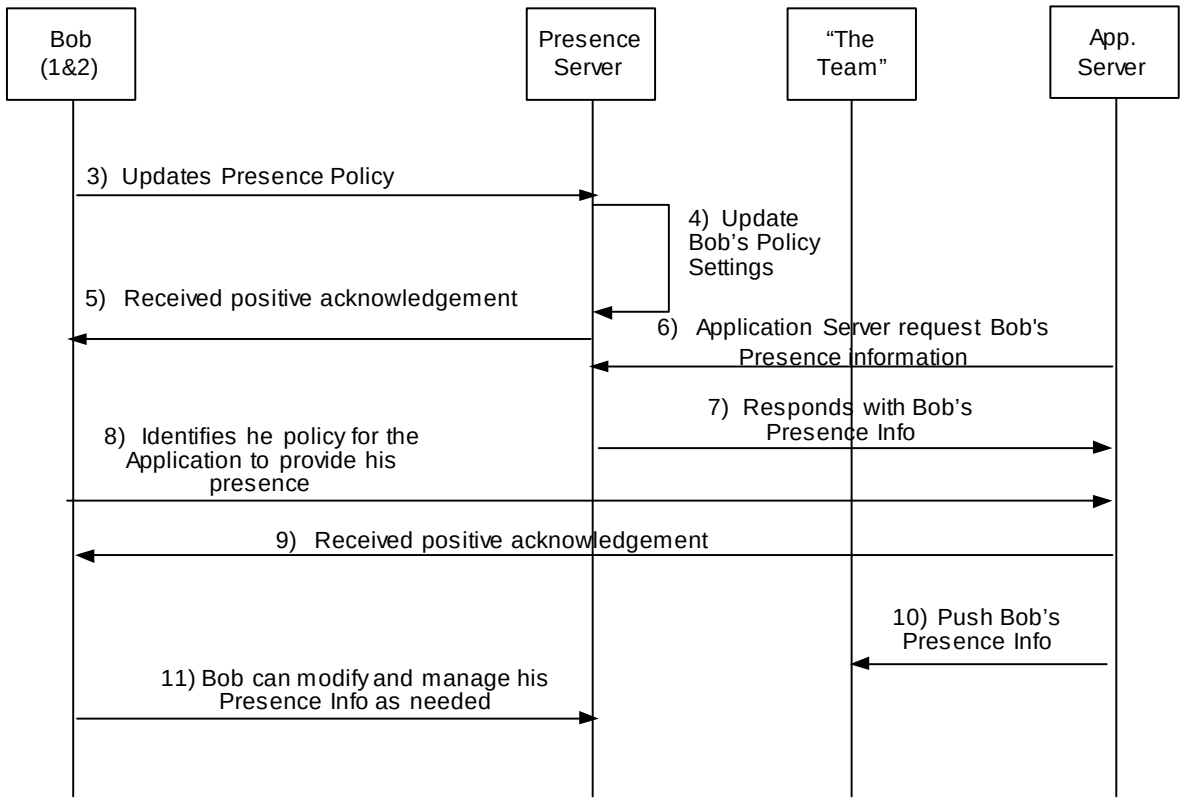
The presence service is handled by a presence server that is logically separate from the application server.

#### 5.2.2.6 Postconditions

Bob will have authorised the Team's application to access his presence service information, and. When members of his team use the application they will be able to tell if Bob is available.

#### 5.2.2.7 Normal Flow

1) Bob has a new device that allows him to share his presence information with other users, such as his Team (his old device did not support presence services).
2) Bob decides that allowing his friends and business colleagues to see his presence information would be valuable to him.
3) Bob updates his presence information and privacy parameters via his presence application.
4) The presence server updates its policy information for Bob.
5) Bob receives a positive acknowledgement from the presence server indicating that his update was successful.
6) The application service request Bob's presence information from his presence service
7) The presence service responds with Bob's presence information to the application service.
8) Bob accesses his application and identifies who is authorised to receive his presence information, and indicates that all of his presence information should be made available.
9) Bob receives a positive acknowledgement from the server indicating that his update was successful.
10) The application can now use Bob's presence information. The Team members that are currently connected to the Service will have their presence information on Bob updated accordingly. Team members that connect to the Service later will receive Bob's updated presence information
11) Bob is able to modify and change his presence information as he needs depending on if he does not want to be disturbed, is in a meeting, or is away from the office, etc.




#### 5.2.2.8 Operational and Quality of Experience Requirements

- By having the ability to manage and provide his Presence information, Bob is now able to be more effective at work, and to have a more enjoyable life away from work.

### 5.2.3 Finding Other Presence Users

This use case will demonstrate how a user can find other users that utilize presence enabled services.

#### 5.2.3.1 Short Description

In this use case we will see a user (Bob) trying to find another user of presence-enabled services. This use case includes how Bob is authorized for this service, and privacy issues that will need to be considered in this use case. A successful out come of this use case will have Bob successfully find the other user of presence-enabled services. The term "find" in this use case means that Bob is able to identify the other user; it does not mean that he will literally locate the person.

#### 5.2.3.2 Actors

Bob – The person that is attempting to find another user of presence-enabled services.

Jan – This is the user that Bob finds

Presence Server – This is the server that is providing the presence service to Bob and Jan

##### 5.2.3.2.1 Actor Specific Issues

Bob has just signed up for a new service [Presence] and he is trying to figure out how it works and can be used on his device.

Jan is a subscriber of presence-enabled services and has been using them for sometime, she is also a friend of Bob's and is helping Bob to understand his service.

The Presence Server is the network entity that manages both Bob and Jan's presence service.

#### 5.2.3.2.2 Actor Specific Benefits

As a new user, Bob will be able to enable different clients on his device with presence information and provide enriched capabilities that will save him time and money, and make him a more effective in his work and day-to-day life.

Since Jan is already a user of presence-enabled services in her life she is focused on showing Bob how it works and how he will benefit from presence information.

### 5.2.3.3 Preconditions

Bob has just started using the service and has a device with several clients (IM, PoC, and phonebook) that can use and benefit from presence information, but he is unsure how to use it. The device also has an application that will allow Bob to learn how to use presence-enabled services and to locate other users, depending on the other user's privacy configurations. He has called his friend [Jan] and asked if she can help him understand the service.

Jan has been a user of presence-enabled services for sometime and is already enjoying their benefits. She is also competent in how it works and can help Bob to understand his new capability. She is currently with Bob and is working with Bob to understand the service.

Bob and Jan are together with their presence enabled devices.

Bob and Jan both have their wireless access and presence services with the same provider.

Jan has given Bob a brief description on what presence information is, and how it can be used on his applications

Jan has already setup her privacy settings to allow Bob to see her "Present/Not Present"

The Presence Server has been provisioned to provide both Bob and Jan with presence capabilities.

The Presence Server in this use case resides in the Operator network. This is not to imply that it could not reside in a 3rd party network.

### 5.2.3.4 Postconditions

Bob will understand how presence-enabled services work, and how to use them to enrich his IM, PoC, and phonebook on his devices.

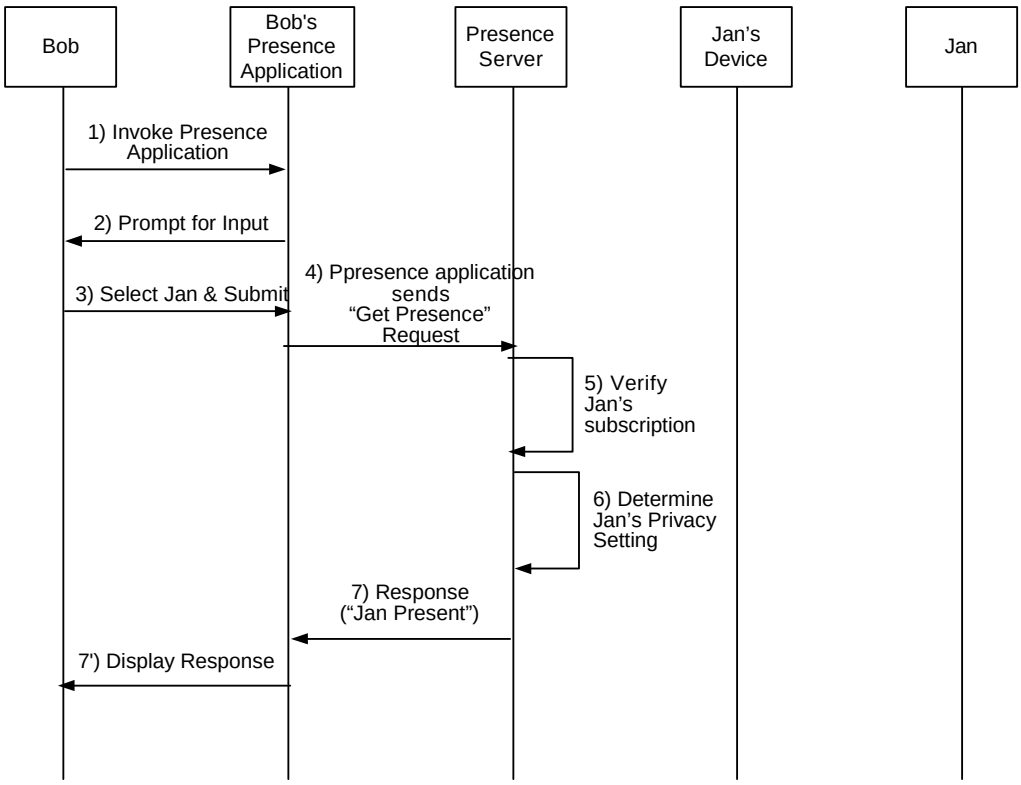
Jan will still be a user of presence-enabled services and will have successfully demonstrated to Bob how to use the service.

### 5.2.3.5 Normal Flow

1) Jan shows Bob how to access the presence-enabled application on his device. This application will help Bob to find other users of presence-enabled services.
2) The application will prompt Bob to input the information to try and see if the other user is provisioned to use presence enabled services. This application will allow Bob to either select a contact from his phonebook or to input the user manually.
3) Jan suggest that he select her from his phonebook.
4) Bob selects Jan from his phonebook, and submits the request. The client will then send a "get presence" request to the Presence Server.
5) The Presence Server will take the request and verify first that the user is a valid user of presence-enabled services [which Jan is], and then will check and see what privacy setting the user [Jan] has set up for this requestor.
6) The Presence Server determines the user [Jan] is configured to allow a "present/not present" response to be sent.

7) The Presence Server formats the "response" and sends it back to Bob's device. The device will display a notification to Bob stating "Jan is present."

8) Jan then shows Bob how to enable the presence features on his phonebook, PoC service, and IM service. Bob is amazed at how his user experience is changed [for the better] with this capability.




### 5.2.3.6 Alternative flow: Blocking Presence Information

9) Jan shows Bob how to access the presence application on his device. This application will help Bob to find other users of presence-enabled services.

10) The application will prompt Bob to input the information to try and see if the user is provisioned to use presence-enabled services. This application will allow Bob to either select a contact from his phonebook or to input the user manually.

11) Jan suggest that he select her from his phonebook (Jan has her privacy setting set to not give presence information).

12) Bob selects Jan from his phonebook, and submits the request. The client will then send a "get presence" request to the Presence Server.

13) The Presence Server will take the request and verify first that the user is a valid user of presence-enabled services [which Jan is], and then will check and see what privacy setting the user [Jan] has set up for this request.

14) The Presence Server determines the user [Jan] is configured to NOT send her presence information.

15) The Presence Server formats the "response" and sends it back to Bob's device. The device will display a notification to Bob stating "Presence Information Unavailable."

16) Jan then shows Bob how to manage his presence privacy settings.

#### 5.2.3.7 Alternative Flow: Privacy Setting has been setup to allow all Presence Information

17) Jan shows Bob how to access the presence application on his device. This application will help Bob to find other users of presence-enabled services.

18) The application will prompt Bob to input the information to try and see if the user is provisioned to use presence enabled services. This application will allow Bob to either select a contact from him phonebook or to input the user manually.

19) Jan suggest that he select her from his phonebook.

20) Bob selects Jan from his phonebook, and submits the request. The client will then send a "get presence" request to the Presence Server.

21) The Presence Server will take the request and verify first that the user is a valid user of presence enabled services [which Jan is], and then will check and see what privacy setting the user [Jan] has set up for this request.

22) The Presence Server determines the user [Jan] is configured to allow detailed presence information in the response to be sent.

23) The Presence Server formats the "response" and sends it back to Bob's device. The device will display a notification to Bob stating "Jan is present via PoC, IM, and phonebook on her mobile device."

24) Jan then shows Bob how to enable the presence features on his phonebook, PoC service, and IM service. Bob is amazed at how his user experience is changed [for the better] with this capability.

#### 5.2.3.8 Operational and Quality of Experience Requirements

The Presence Server has the ability to modify/manage the level of presence information provided to the presentity based on privacy rules set by both the presentity [Jan] and the watcher [Bob]. This use case does not define how the Presence Server gets the information. It could be managed on the Presence Server or could be via an external database that the Presence Server can access.

The Presence Server will need to have access to the information on what type of device the users have and their presence capabilities. This could be internally or externally available to the Presence Server.

### 5.2.4 Updating Presence Information

#### 5.2.4.1 Short Description

This use case shows how the Presence Service is updated when a mobile subscriber changes his/her presence status (e.g. he/she is becoming "online" from "offline"). Also the friends of this user that have subscribed to get his presence information are getting notified about the change in his presence status.

#### 5.2.4.2 Actors

- Alice – The user that changes her presence status
- Alice's Friends – Number of users that have subscribed to have access to Alice's presence information
- Presence Client (PC) – Resides on the mobile devices of all human actors
- Presence Enabled Phonebook – Resides in mobile devices of all human actors and is able to get presence information from the Presence Client (PC)
- Presence Server – Resides in the network

#### 5.2.4.3 Actor Specific Issues

Alice wants to:

- allow all her friends to have access to her presence status

Alice's Friends want to:

- get a notification when Alice becomes "online"

#### 5.2.4.4 Actor Specific Benefits

Alice's Friends:

- Get notifications when Alice is becoming "online" and they are able to contact her

Alice:

- Shows to her friends her presence status and she is able to be contacted when it is appropriate

#### 5.2.4.5 Preconditions

- Alice and Alice's friends are all provisioned to use the presence service
- Alice allows her friends to get all the presence information in the notifications when changing presence status
- Alice's presence service is configured to automatically update her presence status from « offline » to « online » when she is switching on her mobile phone
- Alice's friends have all subscribed in order to get notifications when Alice changes her presence status
- Alice's mobile phone is able to perform presence service related actions for all the duration of the use case (meaning that it has coverage all the time)

#### 5.2.4.6 Postconditions

- Alice's friends successfully manage to get a notification when Alice changes her presence status

#### 5.2.4.7 Normal Flow

1) Alice has her mobile switched off as she is in the cinema and her friends see in their PEP in their mobile phones that she is "offline
2) Alice switches on her mobile and as it is configured to automatically update her presence status when she switches it on, her status is updated in the Presence Server.
3) The Presence Server generates notifications to Alice's Friends Presence Clients (PCs) to inform them that Alice is now "online".
4) The presence status of Alice in Alice's Friends PEP in their mobile phones is now changed from "offline" to "online"

#### 5.2.4.8 Alternative Flow

None

#### 5.2.4.9 Operational and Quality of Experience Requirements

- The Presence Server shall be able to generate notifications for Alice's presence status on behalf of Alice when she is not able to update her presence status by herself (e.g. her mobile is out of coverage).
- Alice's Friends shall be able to define how often they will receive notifications about Alice's status changes

- The Presence Service shall be able to send notifications about Alice's presence information to her friends only for the presence elements that change or that they are interested in.

## 5.2.5 Presence-enabled Address-book

### 5.2.5.1 Short Description

This use case describes how two wireless subscribers can utilize the Presence Service to initiate different types of communication sessions. In the flows presented below, Alice is trying to reach Bob through her presence-enabled address-book. In each of the flows, Bob becomes available via different means (voice, MMS, and IM, respectively), and Alice initiates communications using each of those means.

### 5.2.5.2 Actors

- Alice
- Bob
- Presence Service

### 5.2.5.3 Actor Specific Issues

Alice wants to:

- Get in touch with Bob as soon as possible
- Use the most appropriate means (e.g. send an MMS message instead of calling Bob so as not to disturb him if he's in a meeting)

Bob wants:

- To be able to express a set of preferences, regarding how he wants to be contacted.
    - Telephone call vs. text messaging
    - Home vs. Office vs. Cell phone
- The Presence Service to work automatically behind the scenes, requiring a minimal amount of user interaction.

### 5.2.5.4 Actor Specific Benefits

Alice:

- Is able to contact Bob according to her preferences

Bob:

- Get better control of his incoming communications

### 5.2.5.5 Pre-conditions

1) Alice and Bob are both provisioned to use the Presence Service
2) Bob has his phone turned off
3) Bob is not logged in his Instant Messaging service
4) Bob has two profiles set-up on the Presence Server: "Available", and "In Meeting". The former shows Bob as reachable via all available communication means, whereas the latter shows him as available only via text communications.
5) The "Available" profile is the default profile that it automatically selected when a device is powered up.

#### 5.2.5.6 Postconditions

1) Alice successfully manages to communicate with Bob

#### 5.2.5.7 Normal Flow – Voice call

1) Alice invokes her presence-enabled address-book
2) Alice scrolls down to Bob, and sees he is not available
3) Alice selects an option to be alerted when Bob becomes available
4) Later, Bob turns on his cell-phone, and the "Available" profile is automatically selected
5) Bob's cell-phone publishes his availability, as well as some additional information such as the capability of his phone to send and receive MMS messages *to the Presence Service*
6) The Presence Service generates a notification which is routed to Alice's phone
7) Alice's phone displays an alert and/or makes an alert sound
8) Alice selects the alert which takes her to her address-book and automatically scrolls to Bob
9) Alice sees he is available for telephone calls, and places the call

#### 5.2.5.8 Alternative Flow 1 – MMS message

1) Alice invokes her presence-enabled address-book
2) Alice scrolls down to Bob, and sees he is not available
3) Alice selects an option to be alerted when Bob becomes available
4) *Later, Bob turns his cell-phone on, and he selects the "In Meeting" profile*
5) Bob's cell-phone publishes Bob's availability, as well as some additional information, such as the capability of the phone to send and receive MMS messages *to the Presence Service*
6) The Presence Service generates a notification which is routed to Alice's phone
7) Alice's phone displays an alert and/or makes an alert sound
8) Alice selects the alert which takes her to her address-book and automatically scrolls to Bob
9) *Alice sees he is available for text messaging, composes a text message and sends it to Bob*
10) *Bob receives the message as an MMS message*

#### 5.2.5.9 Alternative Flow 2 – IM Session

1) Alice invokes her presence-enabled address-book
2) Alice scrolls down to Bob, and sees he is not available
3) Alice selects an option to be alerted when Bob becomes available
4) *Later, Bob logs on to the Instant Messaging application on his desktop PC, and the "Available" profile is automatically selected*
5) *Bob's Instant Messaging application publishes Bob's availability, as well as some additional information, such as the capability of the Instant Messaging application to send and receive IM message to the Presence Service*
6) The Presence Service generates a notification which is routed to Alice's phone
7) Alice's phone displays an alert and/or makes an alert sound
8) Alice selects the alert which takes her to her address-book and automatically scrolls to Bob
9) *Alice sees he is available for text messaging, composes a text message and sends it to Bob*
10) *Bob receives the message via Instant Messaging*

#### 5.2.5.10 Operational and Quality of Experience Requirements

- The Presence Service should have response times such that the end-user does not experience any significant delays when using applications that require the retrieval of presence information
- The Presence Service should be able to generate notifications in a timely fashion, after relevant presence information has been published to it.

### 5.2.6 Validity Period

#### 5.2.6.1 Short Description

Certain presence elements such as a user's availability, their location, and address are subject to the elapse of time. For example, a person's availability may change over time. The validity period is used to describe this time duration. In this use case it's shown how the validity period can be used to eliminate the task of an end user having to modify their presence information.

#### 5.2.6.2 Actors

Sally – End user who subscribes to a presence service.

Tom – End user who subscribes to a presence service.

Presence Client – Application in Sally's and Tom's mobile device.

Presence Server – Server that communicates with the presence client in managing the presence service.

##### 5.2.6.2.1 Actor Specific Issues

No specific issues exist with any actors.

##### 5.2.6.2.2 Actor Specific Benefits

Sally

- Availability status is automatically updated based on the validity period.
- Eliminates the problem of Sally forgetting to reset or update her availability manually.
- Sally is able to override an element that was earlier set with a validity period.

#### 5.2.6.3 Pre-conditions

Sally's mobile device is enabled with a Presence service.

Sally's mobile device is powered on and in a good coverage area.

#### 5.2.6.4 Postconditions

Sally's presence availability status is accurate according to her settings.

#### 5.2.6.5 Normal Flow

1) Sally has an appointment for a scheduled time of 2 hours.
2) Just before her appointment, Sally modifies her presence status to unavailable, and adds the validity period qualifier of 2 hours. Sally indicates to the Presence Service that she wants to be available after the validity period has elapsed
3) The presence client in Sally's mobile device communicates with the presence server to update Sally's status, including the validity period.

4) Sally's updated presence information is available to all those who have subscribed to watch her status, including Tom.
5) Before calling Sally, Tom decides to retrieve an update of Sally's presence information and receives her unavailability status, along with the validity period for which that status applies.
6) Tom decides not to call Sally based on her unavailable presence status and its validity period.
7) Some time later, Tom again retrieves Sally's presence information. Sally's presence status is available and therefore Tom calls her.

#### 5.2.6.6 Alternative Flow

1) Sally has an appointment for a scheduled time of 2 hours.
2) Just before her appointment, Sally modifies her presence status to unavailable, and adds the validity period qualifier of 2 hours. Sally indicates to the Presence Service that she wants to be available after the validity period has elapsed
3) The presence client in Sally's mobile device communicates with the presence server to update Sally's status, including the validity period.
4) Sally's updated presence information is available to all those who have subscribed to watch her status, including Tom.
5) Before calling Sally, Tom decides to retrieve an update of Sally's presence information and receives her unavailability status, along with the validity period for which that status applies.
6) Sally's appointment is finished well before the 2 hour scheduled period, and therefore Sally modifies her presence status to available.
7) The presence client in Sally's mobile device communicates with the presence server to update Sally's status. Sally's updated presence information is available to all those who have subscribed to watch her status, including Tom.
8) Some time later, Tom again retrieves Sally's presence information. Sally's presence status is available and therefore Tom calls her.

#### 5.2.6.7 Operational and Quality of Experience Requirements

- Sally is eliminated the task of modifying her availability when using the validity period qualifier.
- Sally's subscribers are confidant that her presence information is accurate and timely.
- Presence elements having an associated validity period can be modified by the end user.

### 5.2.7 One-time Event Subscription and Notification

#### 5.2.7.1 Short Description

**One-time Event Subscription and 'unavailable->available' Notification**

One-time subscription and notification mechanism enables the caller to get notified of a callee's availability. As a result of this mechanism monitoring the busy/unreachable callee the caller can decide whether to call the callee again upon notification of the callee's availability status.

The caller is eligible for a one-time "unavailable->available" Event Subscription though may or may not be in the callee's contact list. The callee in this case may put a request to the Presence server to allow subscription to his/her presence only for people in his/her contact list and also request only people in his/her contact list can subscribe to the one-time event notification.

#### 5.2.7.2 Actors

- Bob – who wants to make a voice call from his mobile to Sue
- Sue – who does not want to be disturbed while on a PoC session

- Presence client (PC) – resides in Bob's and Sue's mobile devices
- Presence server – resides in the network
- Voice Call system and Voice Mail System – reside in the network

#### 5.2.7.3 Actor specific Issues

- Bob wants to contact Sue via a voice call
- Sue is currently unreachable for voice calls
- Bob wants to know about Sue's presence availability as soon as Sue becomes available
- Sue wants to make sure that Bob calls her only when she is available

Actor Specific benefits

- Bob wants to receive a notification from Sue's presence server about her availability therefore giving him the option to call her as soon as possible

#### 5.2.7.4 Pre-conditions

- Both, Sue and Bob have cell phones capable of Presence capabilities supporting new feature proposed in this use case
- Both of them are registered users in the Presence Service
- Bob MAY or MAY not be in Sue's Contact List (This is a policy issue which can be agreed by Sue, but the idea is to allow some flexibility for the first time users not in Sue's contact list such as some travel agents, mortgage advisers etc that would be of interest for Sue).
- Bob is not subscribed to Sue's Presence information.
- The Presence Server supports one-time event subscription and notification.

#### 5.2.7.5 Postconditions

Bob made a one-time event subscription to Sue's Presence Client and was notified of Sue's availability when Sue became available

#### 5.2.7.6 Normal flow

1) Sue is currently unreachable (out of coverage, for example) for voice calls
2) Bob makes a voice call to Sue from his mobile
3) Since Sue is unreachable, Bob's call is forwarded to VMS
4) A response from VMS system goes to Bob telling him that Sue's phone is "unavailable" and asks him if he wants to leave a message.
5) Bob decides not to leave a message because he wants to talk to her as soon as she becomes available.
6) Bob makes a one-time subscription request to Sue's Presence Server.
7) Sue's Presence Server checks Sue's policy for the one-time event subscription and then decides if the subscription from Bob is allowed or not.
8) If it's allowed, the event subscription is successful and Bob gets acknowledged then knows that he will be notified as soon as Sue becomes available
9) Sue becomes available again.
10) Sue's Presence server notifies Bob about her availability with a status change unavailable -> available only, therefore protecting her other private information

11) Bob makes a voice call to Sue knowing her status is available

#### 5.2.7.7 Alternative flow

#### 5.2.7.8 Operational and quality of Experience Requirements

- The advantage of this added feature over VMS is the caller is notified when the callee is available. In the case of VMS the callee may be reluctant to call back for different reasons, such as cost, not recognizing the caller etc.
- If for some reason (due to unstable coverage and weak radio reception etc) the one-time event subscription request is delayed from Bob to Sue and in this time Sue becomes available, Bob is notified immediately
- Sue's Presence service should be able to generate notifications within a defined time frame, so the end-user does not experience significant delays.
- By understanding the callee's presence status a caller can make a decision whether or not to call and therefore the caller avoids unnecessary and unsuccessful call attempts.

## 5.3 Presence Information

### 5.3.1 P2P, Presence Information

#### 5.3.1.1 Short Description

This use-case is intended to illustrate how Presence Information can be:

- Used to concisely express the state and preferences of end-users.
- Re-used across multiple applications, in a consistent manner.

#### 5.3.1.2 Actors

**Alice, Bob:** Users that wants to communicate with other users using a variety of communication means.

**Presence-enabled Phonebook (PEP):** One out of many potential presence-enabled applications. In this instance, it used to view the presence information of other contacts.

**Presence Service**: Collects and disseminates presence information, subject to a variety of controls.

#### 5.3.1.3 Actor Specific Issues

**Presence-enabled Phonebook:** This type of Presence-enabled application utilizes two types of Presence elements: "availability" and "willingness".

"Availability" for a particular communication means specifies whether it is possible, at that point in time, to initiate a session using that communication means. Several factors may be used to determine this, such as network status, device status, device capabilities, subscriber provisioning, etc.

"Willingness" for a particular communication means specifies whether the end-user, at that point in time, is willing to receive communications of that type. This presence element is meaningful only if the user is also "available" for that communications means.

This type of PEP is extensible, such that it can initiate communications for any communication means, by launching an appropriate application handler. Should it need to initiate communications for an application that is not supported on the device, it may prompt the user to download that application. There are several ways to do this that are beyond the scope of this use-case.

This type of PEP will display an icon, next to every contact. This icon will show as green if the user is available and willing to communicate with at least one communication means, and red in all other cases.

#### 5.3.1.4 Actor Specific Benefits

- Alice and Bob can initiate sessions with different types of communication means, without having to constantly upgrade their PEPs.

#### 5.3.1.5 Pre-conditions

- Alice and Bob are provisioned as users of the Presence Service, and any other communication services they want to initiate (e.g. SMS, OMA IMPS, etc.)

#### 5.3.1.6 Postconditions

- Alice and Bob were able to communicate successfully according to their presence preferences

#### 5.3.1.7 Normal Flow

1) Alice invokes her Presence-enabled Phonebook (PEP) on her mobile terminal.
2) Alice sees that Bob's icon is green, meaning he is available and willing to receive at least on type of communication.
3) Alice selects the entry representing Bob and is presented with the different types of communications Bob is currently publishing as available and willing
4) Alice selects one of those communication means (say, SMS) and initiates such a session with Bob

#### 5.3.1.8 Alternative Flow

1) Alice invokes her Presence-enabled Phonebook (PEP) on her mobile terminal.
2) Alice sees that Bob's icon is green, meaning he is available and willing to receive at least on type of communication.
3) Alice selects the entry representing Bob and is presented with the different types of communications Bob is currently publishing as available and willing.
4) Alice selects on of those communication means (say, OMA IMPS messaging) but does not have the appropriate application on her terminal, so she is prompted to download it. (Note: additional presence information would be required to implement this mechanism, and indeed other mechanisms are also possible. This aspect of the use-case is intended only as an example, and is not meant to be prescriptive.)
5) Alice accepts the download, and installs the application on her terminal.
6) An OMA IMPS session is initiated with Bob.

#### 5.3.1.9 Operational and Quality of Experience Requirements

- Presence elements are defined in an extensible way, such that applications, such as PEP, know exactly how to interpret their meaning and appropriately act on them.

### 5.3.2 P2P, User Setting Presence

#### 5.3.2.1 Short Description

This use-case highlights the many different mechanisms that can be used to update one's presence information.

#### 5.3.2.2 Actors

**Alice, Bob, Charlie:** End-users that own one or more presence-enabled applications.

**Presence-enabled Phonebook (PEP):** One out of many potential presence-enabled applications. In this instance, it used to view the presence information of other contacts.

**Presence-enabled Phonebook on Steroids (PEPoS):** One out of many potential presence-enabled applications. This is a very sophisticated version of PEP, intended for advanced users that have complex communications requirements.

**Presence-enabled PoC client:** One out of many potential presence-enabled applications. This is a normal PoC client that enables the end-user to see whether the user's contacts are available and willing to accept PoC communications. Similarly it can be used to select whether the user is willing to accept incoming PoC communications.

**Presence Service**: Collects and disseminates presence information, subject to a variety of controls.

#### 5.3.2.3 Actor Specific Issues

- Each actor uses different applications, and has different communication preferences.

#### 5.3.2.4 Actor Specific Benefits

- Alice, Bob & Charlie can communicate their presence information to each other, according to their preferences, regardless of the applications that they each use.
- Alice, Bob & Charlie get a consistent view of each other's presence information, regardless of the applications that they use.

#### 5.3.2.5 Pre-conditions

**Note: the profiles defined below are UI concepts that conveniently combine a set of end-user preferences. They are unrelated to application specific statuses such as the OMA PoC states (Reachable, Busy, DND, etc.), or OMA IMPS states (Available, Discreet, etc.) defined in the relevant specifications.**

- Alice is provisioned to use the Presence Service and the PEP.
- Alice has configured her "Discreet" profile to silence her ringer and indicate willingness to accept only text-based communications.
- Alice has configured her "Normal" profile to show her as willing to accept any type of communication that her device supports and set her ringer to "loud".
- Alice has configured her "DND" profile to show her as unwilling to accept any type of communication.
- Bob is provisioned to use the Presence Service and the PoC Service
- Bob uses a simplified PoC client that uses only "Available" and "Unavailable" presence states. The manual of Bob's phone states that if Bob wants to receive PoC calls he should set this feature to "Available" and if not to "Unavailable".
- Charlie is provisioned to the Presence Service and the PEPoS.
- Charlie has defined several profiles such as "Work", "Work – discreet", "Traveling", "Home – working", "Private time", and more.
- Each of Charlie's profiles contains setting with respect to published presence information, ringer settings, call forwarding features, and more. In addition, some of those profiles can be automatically engages, as triggered by events such as change of physical device location. For example his "Work" profile allows all types of communications (voice, PoC, SMS, etc.) whereas his "Private time" profile only allows only voice and text from a certain list of individuals.
- The PEPoS application also allows Charlie to associate each of his profiles with a text message that is shown to other PEPoS users, or users of other applications that support this text field.
- Alice, Bob and Charlie are subscribed to each other's presence information, each through their own preferred application.

#### 5.3.2.6 Postconditions

- Alice, Bob and Charlie receive each other's presence information in a consistent way.

#### 5.3.2.7 Normal Flow

1) Alice is going into a meeting.
2) Alice invokes her Presence-enabled Phonebook (PEP) on her mobile terminal.
3) Alice invokes the "Select Profile" feature and is presented with the following list: "Normal", "Discreet", and "DND".
4) Alice selects the "Discreet" feature.
5) Later, Alice is finished with her meeting and uses the above sequence to select the "Normal" profile.
6) Later, Alice wants to focus on finishing a report and uses the above sequence to select the "DND" profile.

#### 5.3.2.8 Alternative Flow 1

1) Bob is going into the library.
2) Bob invokes his PoC client and sets his status to "Unavailable".
3) Bob's terminal, directly or indirectly updates Bob's presence information to the Presence Server. Other actions may also need to be taken (e.g. inform the PoC server of this selection), but those are out of scope for this use-case.
4) Bob comes out of the library.
5) Bob invokes his PoC client and sets his status to "Available".

#### 5.3.2.9 Alternative Flow 2

1) Charlie is in a meeting and thus has his "Work – discreet" profile selected.
2) Charlie exits the meeting and selects his "Work" profiles.
3) Charlie receives incoming voice, PoC calls and SMS messages.
4) Charlie goes home and his "Personal Time" profile is automatically triggered due to his change of location.
5) Charlie only receives voice calls and text messages from his friends.

#### 5.3.2.10 Alternative Flow 3

1) The three flows above are happening simultaneously.
2) Alice, Bob & Charlie can see each other's presence state, as presented by their respective applications.

**Notes:**

a) Alice's PEP allows her to have a fairly comprehensive view her contacts' presence information, as it supports multiple types of communication. However, this type of PEP doesn't support the "text field" and as such she cannot see that information from, say, Charlie.

b) Bob is using a PoC client; therefore he can only see PoC-related presence information, and initiate PoC sessions only.

c) Charlie's PEP allows him to have a fairly comprehensive view his contacts' presence information, as it supports multiple types of communication. In addition, Charlie can see the "text field" from those contacts of his that publish it.

#### 5.3.2.11 Operational and Quality of Experience Requirements

- The Presence Service SHALL support a wide variety of mechanisms for the end-users to set their presence information.

Presence information SHALL be specified in an extensible manner, such that a wide variety of terminals with different applications and capabilities will be able to meaningfully and consistently interpret it.

### 5.3.3 Set Global Do-Not-Disturb (DND)

This use case will demonstrate how users of presence-enabled services can set their Do-Not-Disturb parameters. The □ehaviour□ of other services to the DND are out the scope of this use case.

#### 5.3.3.1 Short Description

In this use case, we will see a user (Bob) setting up his Do-No-Disturb (DND) status to block all incoming communications. A successful out come of this use case will have Bob successfully setup his DND status and will have the ability of the Presence server to communicate this status with the different Presence enabled services.

#### 5.3.3.2 Actors

Bob – The person that is attempting to setup the DND status to block all incoming communications.

Jan – This is the user that attempts to communicate with Bob via IM.

John – This is the user that attempts to communicate with Bob via PoC.

Presence Server – This is the server that is providing the presence capability to Bob, Jan, and John

#### 5.3.3.3 Actor Specific Issues

Bob is subscribed to the Presence Service and he would like to block all incoming communications during his meeting.

Jan and John would like to communicate with Bob using IM and PoC.

The Presence Server is the network entity that manages Bob, John, and Jan's presence service.

#### 5.3.3.4 Actor Specific Benefits

Bob will be able to enable different clients on his device with presence information and provide capabilities to set his presence status. Bob will use the Do-Not-Disturb (DND) presence status to block all the incoming communication to him during the meeting.

#### 5.3.3.5 Preconditions

Bob has a device with several profiles setup (e.g., General, Meeting, Silent, etc.) in his client. For the meeting profile one of these profile attributes sets the presence status to DND.

Bob also has several clients (IM, PoC, and Gaming) that can use and benefit from presence information.

Jan, John, and Bob have already setup their privacy settings to allow each other access to their presence status.

The Presence Server has been provisioned to provide Bob, John, and Jan with presence capabilities.

The Presence Server in this use case resides in the Operator network. This does not imply that it could not reside in a 3rd party network.

#### 5.3.3.6 Postconditions

Bob is able to set his DND status and the IM and PoC communications are blocked.

Jan and John are still users of presence-enabled services and are able to access Bob's presence status.

All communications from presence enabled services to Bob will be blocked

#### 5.3.3.7 Normal Flow

1) Bob will access his presence client and will setup his profile to "meeting", which sets Bob's presence status as DND.
2) Jan will attempt to send Bob an IM. When Jan starts her IM client, the client will show Bob's status as DND.
3) In the same time, John will attempt to use PoC to communicate with Bob.
4) When John starts his PoC client, he will be notified that Bob is in the DND mode and will not accept any PoC communications.
5) If John attempts to communicate with Bob, the PoC service will block John's attempt.

#### 5.3.3.8 **Alternative Flow:** Restricted Presence Information Views

Bob has blocked his presence information being delivered to Jan and John.

1) When Jan accesses her IM client, Jan will not be able to determine Bob's status.
2) When Jan attempts to send an IM to Bob, the message will not be delivered to Bob immediately and Jan may get a notification that her IM was not delivered.
3) When John accesses his PoC client, John will not be able to determine Bob's status.
4) John will not be able to communicate with Bob using PoC.

#### 5.3.3.9 **Alternative Flow:** Using Service- Specific DND

Bob set his PoC DND status and did not set his global DND

1) Jan will be able to successfully send and receive IM with Bob
2) John will not be able to communicate with Bob using PoC

#### 5.3.3.10 Operational and Quality of Experience Requirements

The user has the ability to manage and edit his presence status such as DND. This use case does not define how the Presence Server gets the information. It could be managed on the Presence Server or could be via an external database that the Presence Server can access.

The different presence enabled services such as IM, and PoC have the capability to access users' status. The behaviour of these services to the different presence statuses are specific to these services and are defined by these services.

Each Presence enabled service need to have a separate DND presence status that can be independently set by the presentity and can be notified to a watcher.

### 5.3.4 Reset Global DND

#### 5.3.4.1 Short Description

A user has set Global DND presence subscriber status and now intends to reset that Global DND subscriber status i.e., the user's Willingness Override value is reset. In this context, when Global DND presence subscriber status is reset, the result is that the application-level status information settings to take precedence.

#### 5.3.4.2 Actors

Bob – The presence service end user that is resetting his Global DND presence information.

Jan – This is the user that attempts to communicate with Bob using an IM application.

John – This is the user that attempts to communicate with Bob using a PoC application and also a Voice application.

Presence Server – This is the network server that is providing presence capability to Bob, Jan, and John.

Presence Client – This is part of the presence application that resides at the end user's terminal that is able to send and receive presence information to and from the Presence Server on behalf of the presence service end user.

#### 5.3.4.3 Actor Specific Issues

When an end user sets Global DND Presence Subscriber Status to block incoming requests for new communications, a mechanism is needed that allows the end user to revert back to the condition where the Global DND status is reset in a manner that allows application-level presence status to take precedence. This use case addresses the issue of resetting the Global DND subscriber status.

#### 5.3.4.4 Actor Specific Benefits

The end user is able to revert to a presence status that removes effect of the Global DND Presence Subscriber Status.

Users allowed to subscribe for presence information of other users have the capability to be aware of the Global DND Presence Subscriber Status of other users.

#### 5.3.4.5 Pre-conditions

Bob has SET his Global DND Presence Subscriber Status information as described in section *Use Case – Set Global Do-Not-Disturb (DND).*

Bob, Jan, and John's presence-enabled multimedia services include IM, PoC and Voice.

Bob has allowed both Jan and John to subscribe to receive his presence information, including Global Presence information.

Bob's IM subscriber presence status is set to 'Available' and 'Willing'.

Bob's PoC subscriber presence status is set to 'Available' and 'Unwilling'.

Bob's Voice subscriber presence status is set to 'Available' and 'Willing'.

Both Jan and John have subscribed to receive updates to Bob's presence information, including Global Subscriber Status Presence information.

#### 5.3.4.6 Postconditions

Bob's Global DND presence subscriber status is reset on, or removed from, the Presence Server.

Jan and John's Presence Clients have received Bob's updated application-specific and Global presence subscriber status.

#### 5.3.4.7 Normal Flow

1) Bob indicates to his Presence Client that Global DND presence subscriber status is now reset.
2) The Presence Client forwards the updated Global DND presence information to the Presence Server.
3) Jan and John's Presence Clients receive the updated Global DND presence information and make this information available to Jan and John.

4) Jan is able to see that Bob has reset his Global DND presence subscriber status and is now 'Available' to receive IM communications. Jan uses her IM application to communicate with Bob.
5) Bob receives Jan's IM communication and is able to respond.
6) John is able to see that Bob has removed his Global DND presence subscriber status, however John is also able to see that Bob remains 'Unavailable' to receive PoC communications. John remains unable to communicate with Bob using the PoC application.
7) John is able to see that Bob is 'Available' for Voice communications and sends Bob an invitation to talk using the Voice application.

## 5.3.5 Use Case – Global DND with Interactions

This Use Case is intended to demonstrate how applications are effected when Global DND presence subscriber status is set, i.e., the user's application-specific status is 'Available and Willing', and the Willingness Override value is set to 'Unwilling'. When the Global DND status is set, service enablers that subscribe to receive Global DND information will receive and act on the Global DND status according to the rules defined by that service enabler, to achieve the Global DND effect.

### 5.3.5.1 Short Description (Global DND with Interactions)

When Bob engages in a PoC session with John, Bob sets his Global DND status so that he can communicate undisturbed. Jan intends to communicate with Bob using an IM application, however seeing Bob's DND status, Jan does not initiate the IM session.

### 5.3.5.2 Actors

Bob – The presence service end user that is resetting his Global DND presence information.

Jan – This is the user that attempts to communicate with Bob using an IM application.

John – This is the user that attempts to communicate with Bob using a PoC application and also a Voice application.

Robert – This is a IM user that attempts to communicate with Bob, but does not receive Bob's presence information.

Presence Server – This is the network server that is providing presence capability to Bob, Jan, and John.

Presence Client – This is part of the presence application that resides at the end user's terminal that is able to send and receive presence information to and from the Presence Server on behalf of the end user.

### 5.3.5.3 Actor Specific Issues

End users of presence service enablers should experience a consistent □ehaviour when using application-level and global-level DND presence subscriber status.

### 5.3.5.4 Actor Specific Benefits

When a end user sets Global DND Presence Subscriber Status to block incoming requests for new communications, the Global DND status is treated in a consistent manner by each presence service enabler in accordance to that enabler's specific rules for DND.

### 5.3.5.5 Preconditions

Bob has SET his Global DND Presence Subscriber Status information as described in section *Use Case – Set Global Do-Not-Disturb (DND).*

Bob, Jan, Robert and John's presence-enabled multimedia services include IM, PoC and Voice.

Bob has allowed both Jan and John to subscribe to receive his presence information, including Global Presence information.

Both Jan and John have subscribed to receive updates to Bob's presence information, including Global Subscriber Status Presence information.

Bob's application-level presence subscriber status for IM is 'Available' and 'Willing'.

Bob's application-level presence subscriber status for Voice is 'Available' and 'Willing'.

Bob's application-level presence subscriber status for PoC is 'Busy'. Bob is currently engaged in a PoC session with John.

Robert is able to use his IM application to communicate with Bob but does not receive Bob's presence information.

#### 5.3.5.6 Postconditions

Bob's Global DND presence subscriber status is reset and published by the Presence Server.

Jan and John's Presence Clients have received Bob's updated application-specific and Global presence subscriber status.

#### 5.3.5.7 Normal Flow

1) Bob, while engaged in a PoC session with John, Bob sets his Global DND presence subscriber status as described in section Use Case – Set Global Do-Not-Disturb (DND).
2) Bob and John are able to continue their PoC session.
3) Jan is aware of Bob's Global DND presence subscriber status and does not initiate an IM session with Bob.

#### 5.3.5.8 Alternative Flow

In this flow a user, Robert, who does not subscribe to Bob's presence information wants to initiate an IM session with Bob. This flow demonstrates how Bob is not disturbed by Robert.

1) Bob, while engaged in a PoC session with John, Bob sets his Global DND presence subscriber status as described in section Use Case – Set Global Do-Not-Disturb (DND)
2) Bob and John are able to continue their PoC session.
3) Robert is not aware of Bob's presence status and invites Bob to join an IM session.
4) Robert receives an indication that Bob is not able to receive the IM invitation.
5) Bob is not disturbed by the IM invitation from Robert.

## 5.4 Network Presence

### 5.4.1 Update the presence status when the mobile is out of coverage

#### 5.4.1.1 Short Description

This use case describes how the presence status of a user is updated when his/her mobile device is out of coverage.

#### 5.4.1.2 Actors

- Alice – Moves to an area where her mobile does not have coverage
- Bob – Wants to see Alice's presence status in his PEP
- Presence Network Agent – Detects that Alice is out of coverage by getting information from various network elements (e.g. SGSN,GGSN).
- Bob's PEP
- Presence Server

#### 5.4.1.3 Actor Specific Issues

Bob:

- Wants to get Alice's Presence status in his PEP

#### 5.4.1.4 Actor Specific Benefits

Bob:

- Is able to get Alice's accurate Presence status even when she is out of coverage

#### 5.4.1.5 Pre-conditions

- Alice and Bob are provisioned to use the Presence Service
- Presence Network Agent is able to detect when Alice is out of coverage and is not able to update her Presence status by getting information from various network elements (e.g. SGSN,GGSN etc)
- When Alice is out of coverage, her status shall be shown as "Not available"
- Alice has authorized the Presence Network Agent to publish presence information on her behalf

#### 5.4.1.6 Postconditions

- Bob is successfully getting Alice's accurate Presence status, although she does not have coverage

#### 5.4.1.7 Normal Flow

1) As Alice is in an area that has coverage her entry in Bob's PEP is shown as "Available"
2) Alice moves to a no-coverage area. The Presence Network Agent by getting information from network elements detects that she is not connected to the network anymore and informs the Presence Server about that.
    1. The Presence Server notifies Bob about this change in Alice's Presence status and now her entry in Bob's Presence-enabled Address is shown as "Not Available".

#### 5.4.1.8 Alternative Flow

None

#### 5.4.1.9 Operational and Quality of Experience Requirements

- It shall be possible to determine how often Alice's Presence status gets updated as a result of notifications from the Presence Network Agent, as it is a trade off between Presence information accuracy and network efficiency

## 5.5 Application-Specific Use Cases

### 5.5.1 Event Buddy

#### 5.5.1.1 Short Description

This use case describes an individual who subscribes as a watcher to presentities that represent social events (e.g., football game, music concert, or a live internet web cast) and is notified of status changes of these social events.

Note: There are no requirements currently specified as a result of this use case. This may be done in a future release.

#### 5.5.1.2 Actors

John – end user with a mobile device enabled with presence service.

Content Server – Server having the responsibility of managing media content and communicating it to other servers.

Presence Client – Application on John's mobile device that manages the presence service.

Presence Server – Entity providing the necessary server requirements of a Presence service.

#### 5.5.1.3 Actor Specific Issues

None.

#### 5.5.1.4 Actor Specific Benefits

John can easily monitor events on subjects that he's interested in.

#### 5.5.1.5 Pre-conditions

- John is a subscriber to presentity A which represents the Chicago Auto Show.
- John is a subscriber to presentity B which represents the Acme Corporation's quarterly CEO live web broadcast.
- The Content Server is the Presentity of events A and B.

#### 5.5.1.6 Postconditions

No particular post conditions exist.

#### 5.5.1.7 Normal Flow

1) The Presence Client on John's mobile phone shows Events A and B along with their pertinent information.
2) Time elapses and presence information for Event B changes.
3) The Content Server communicates with the Presence Server providing the updated presence information for Event B.
4) Since John is a subscriber of Event B, the Presence Client reflects Event B's updated presence information.
5) John selects the relevant information for Event B that being a URL to the corporation's web site.
6) The browser on John's phone shows the live web broadcast0

#### 5.5.1.8 Alternative Flow

No alternate flow exists.

#### 5.5.1.9 Operational and Quality of Experience Requirements

- Subscribing to events shall be intuitive.
- Presence service may be preconfigured with presentities that represent specific social events such that watchers can subscribe to them.

## 5.6 Security And Privacy

Privacy protects user data against unwanted and unauthorized access. The presence privacy protects the data about the availability and presence information of a mobile user against unwanted and unauthorized data access.

All watchers that want to get data about the Presence Information status of another user need to be authorized for their access. This authorization is realized in three different modes:

- The user who owns the mobile terminal (presentity) is directly asked for permission if a watcher tries to fetch or subscribe his presence information. (Reactive Authorization mode) To accomplish users comfort the reactive mode should enable the transition to proactive mode by decision reuse.
- A system that acts on behalf of the user decides for permission if a watcher that is known tries to fetch or subscribe his presence information. (Proactive Authorization mode)
- Reactive Authorization and transition from reactive to proactive authorization mode could be substantially enhanced, if decision supporting features are available as basis for the presentity's evaluation.

Effective support during the decision process for reactive authorization mode will be very beneficial for the presentity. For the users comfort the number of decisions should be as small as possible decisions. Define once reusable for future access.

Since the proactive authorization is normally done without any notification to the presentity, he needs to be sure that his privacy is managed in a way he is able to control and to understand completely.

For authorization the presentity is thinking in relation of trust. Therefore authorization lists (Family, Friend, Colleague…) that connect contacts with the same relation should be used. Since the trust relations normally didn't drive the list generation of communication lists (e.g. used for Instant Messaging and Chat session) a reuse of such lists should normally be avoided.

Additionally it might be meaningful to support relations that are hosted outside the mobile domain. (E.g. Companies Directory Service to decide for proactive authorization or support with public key infrastructure)

## 5.6.1 Presence Privacy

### 5.6.1.1 Short Description

This use case describes how two wireless subscribers can configure the Presence Service such that their privacy preferences are taken into consideration when their presence information is disseminated.

### 5.6.1.2 Actors

- Alice
- Bob
- Presence Service
- Presence-enabled Address-book

### 5.6.1.3 Actor Specific Issues

Alice wants to:

- Share all her presence information to a few family members
- Share most of her presence information with her colleagues
- Share some presence information with a few friends
- Prevent anyone else from receiving her presence information

Bob wants to:

- Share all his presence information with his boss & wife
- Block a few people from receiving any presence information
- Share his cell-phone number & approximate location (e.g. state or region) with everyone else

### 5.6.1.4 Actor Specific Benefits

Alice & Bob:

- Get better control of their incoming communications
- Allow certain individuals to receive relevant information
- Prevent others from receiving private information

#### 5.6.1.5 Pre-conditions

- Alice and Bob are both provisioned to use the Presence Service
- Alice and Bob are both provisioned with three default groups (Friends, Family & Colleagues)
- Alice and Bob access the Presence Service through their presence-enabled address-books

#### 5.6.1.6 Postconditions

None.

#### 5.6.1.7 Normal Flow

1) Alice invokes her presence-enabled address-book.
2) Alice selects the “Edit Privacy Options” menu.
3) Alice goes through her address-book and associates some of her contacts with particular groups.
4) Alice specifies that her “Family” group should receive all her presence information.
5) Alice specifies that her “Colleagues” group should receive several parts of her presence information.
6) Alice specifies that her “Friends” group should receive very few parts of her presence information.
7) Alice specifies a default rule (for watchers not belonging to any group), where no presence information is shared.
8) Alice updates her presence information using some mechanism provided by the presence-enabled address-book.
9) The Presence Server applies Alice’s preferences and sends out appropriate notifications to each of her Subscribed-watchers.

#### 5.6.1.8 Alternative Flow 1

1) Bob invokes his presence-enabled address-book.
2) Bob selects the “Edit Privacy Options” menu.
3) Bob doesn’t like the default groups so he removes them all.
4) Bob creates a “VIP” group.
5) Bob creates a “Blocked” group.
6) Bob assigns his wife and boss to the “VIP” group.
7) Bob assigns a few of his unwanted contacts to the “Blocked” group.
8) Bob allows the “VIP” group to access all his presence information.
9) Bob does not allow the “Blocked” group to access any of his presence information.
10) Bob specifies a default rule (for watchers not belonging to any group), where only his cell-phone and obfuscated location are allowed.
11) Bob updates his presence information using some mechanism provided by the presence-enabled address-book.
12) The Presence Server applies Bob’s preferences and sends out appropriate notifications to each of his Subscribed-watchers.

#### 5.6.1.9 Operational and Quality of Experience Requirements

- Alice and Bob are able to specify different rules for different watchers.
- Rules determine how Alice and Bob's presence information is disseminated.

### 5.6.2 Using the Presence Service for Advertising Capabilities

#### 5.6.2.1 Short Description

This use case describes how presence information can be used in order to advertise specific capabilities. In this specific example the capabilities advertised are those of the access network and they are used to adapt the streaming content delivered to a user.

#### 5.6.2.2 Actors

- Alice – Is downloading a video stream in a multi-access network environment and has Presence Client (PC) and a Streaming Client (SC) installed in her mobile phone
- Presence-enabled Streaming Server (PSS) – Delivers the video stream to Alice's streaming client
- Streaming Client (SC) – Gets the video stream from the streaming server
- Presence Client (PC) – Resides in the mobile device of Alice
- Presence Server – Resides in the network
- Alice's multi-access network mobile phone

#### 5.6.2.3 Actor Specific Issues

Alice wants:

- To configure her presence service in order to indicate to the PSS the type of access network that she is attached to such that the video stream server can adapt the video in the appropriate format.

#### 5.6.2.4 Actor Specific Benefits

Alice:

- Is able to get the video stream in her mobile phone in the most appropriate format according to the access network that she is attached to

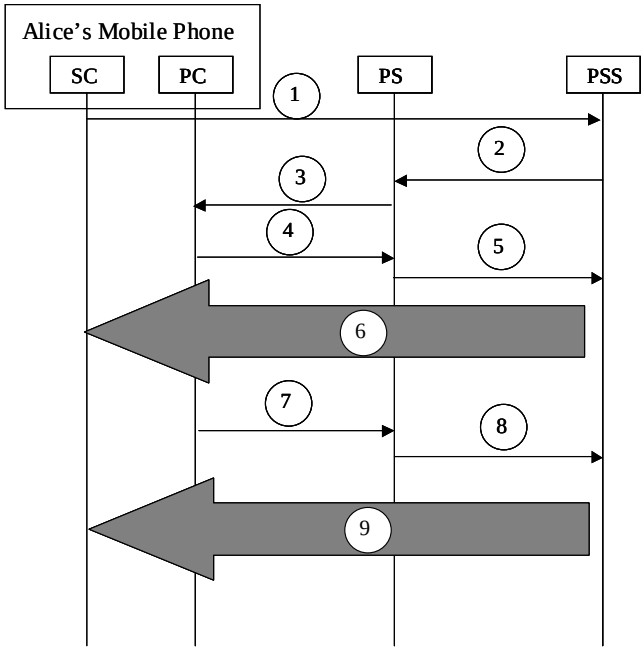
#### 5.6.2.5 Pre-conditions

- Alice and the PSS are all provisioned to use the Presence Service
- Alice is willing to get adaptable content (e.g. have reduced quality when the access network that she is attached to is not fast)
- The Streaming Client (SC) in Alice's mobile phone has the capability to play all the types of the content required (e.g. both high quality and low quality)
- PC is configured to automatically get information from Alice's mobile phone for the access network that it is attached to
- All the actors are able to make changes in their presence information by themselves (i.e. they are in coverage area all the time)
    - Access network information is included in the presence elements

### 5.6.2.6 Postconditions

- Alice is successfully getting the video stream in her mobile in the most appropriate format according to the access network that she is attached to

### 5.6.2.7 Normal Flow




1) Alice's mobile phone is attached to a fast access network (such as a UMTS or WLAN network). Alice wants to download a video stream in her mobile phone while she is on the move from a Presence-enabled Streaming Server (SS) and invokes her Streaming Client (SC) to do so. The Streaming Client requests the specific stream from the Presence-enabled Streaming Server (PSS).
2) The Presence-enabled Streaming Server (PSS) is subscribing in Alice's Presence Service Server to get Alice's presence information regarding the "Access Network" that she is attached to.
3) The Presence Server informs the Presence Client (PC) in Alice's mobile phone about the request and Alice sees in the screen of the her mobile phone "Do you want to use adaptive streaming?"
4) Alice accepts and the Presence Client (PC) authorizes the Presence Server to disseminate Alice's presence information regarding the access network that she is attached to.
5) The Presence Server informs the Presence-enabled Streaming Server (PSS) that Alice is attached at the moment in a fast access network.
6) The Presence-enabled Streaming Server (PSS) sends the video stream to Alice's Streaming Client (SC) in "high quality" format as she is attached in a high bandwidth access network.
7) As Alice is moving her mobile does a handover in a slow access network (such as GPRS). As the Presence Client (PC) is informed about that reports this change in her presence status to the Presence Server (PS).
8) The Presence Server informs the Presence-enabled Streaming Server (PSS) about the change in Alice's presence status including the presence element indicating that she is now attached in a slow access network (such as GPRS).
9) As this access network has lower bandwidth in comparison to the previous one, the Presence-enabled Streaming Server (PSS) adapts the video stream in the conditions by reducing the quality.

### 5.6.2.8 Alternative Flow

None

#### 5.6.2.9 Operational and Quality of Experience Requirements

- Alice should be able to selectively disseminate her device capability to relevant services

### 5.6.3 Reactive Authorization

#### 5.6.3.1 Short Description for Reactive Use Case:

A watcher named Maria is unknown to the presentity named Juliet. Therefore Maria is authorized reactively. (Juliet didn't have Maria in her phonebook) Additionally Juliet wants to reuse her decisions about watcher requests. She likes to decide spontaneously about new contacts and data access of people she will meet in the future. Therefore a watcher unknown to her like Maria shall be authorized reactively only the first time. Future requests should be handled proactively but depend on her decisions during first access request.

#### 5.6.3.2 Actors for Reactive Use Case:

The pure reactive Use Case has only two Actors but decision reuse also needs the Privacy Enforcement entity too:

- **Watcher:** Maria is interested in the presence information of Juliet (Asks for Data Access). Maria got Juliet's number from a common friend and wishes to contact her.
- **Presentity:** Juliet decides about the authorization for Data Access and possibly wants to assign Maria a member of her "known by friend" list. (Decide about the Data Access and Decision Reuse)
- **Privacy Enforcement Entity:** Needs preparation to act on behalf of the Juliet during future request from Maria. (Store Authorization/Identification data for future use)

#### 5.6.3.3 Actors Benefits:

- **Watcher:** The watcher can hope to get data access even if he is not known to the presentity since the presentity has different options to answer his requests. Due to his willingness to present additional information (Call the presentity or send him an SMS before the data request arrives) about himself the watcher can hope for a proactive authorization for future requests.
- **Presentity:** The presentity has the ability to decide once and to reuse his decision for future calls if he trusts the watcher and is able to find an appropriately element setting. If he is not sure about the watcher he can also allow the data request directly but not persistently.

#### 5.6.3.4 Pre-conditions for Reactive Use Case:

- **Watcher:** Maria wants to subscribe to presence information of Juliet but is unknown to her
- **Presentity:** Juliet is willing to decide about data access on request as scarce as possible and is willing to delegate authorization in a controlled manner to the privacy enforcement Entity.
- **Privacy Enforcement Entity**: Supports decision reuse if there is enough information delivered about the watcher to securely identify a second call of the same watcher.

#### 5.6.3.5 Postconditions for Reactive Use Case:

- **Watcher:**

  In the case: **Authorized and proactive enabled**

  Maria is informed by sending status and/or presence information information and her proactive authorization privilege may be shown to her.

  In the case: **Authorized once**

Maria is informed by sending status and/or presence information information and the duration her request is active

In the case: **Not Authorized**:

Maria is informed by sending status (failed or not allowed)

- **Presentity:**

  In the case: **Authorized and proactive enabled**

  Record the actual decision to support Juliet to allow her to remember the event when she decided the proactive authorization. One option may be to allow the presentity to trace his decisions taken in the past.

  In the case: **Authorized once**

  Record the actual decision to support Juliet during next request of the same watcher that was not allowed the proactive authorization. One option may be to propose the presentity an authorization category for a watcher if he got authorized several times in the past.

  In the case: **Not Authorized**

  Make a record for the decision with a time stamp to block watchers from fast retry. One option may be an automatic entry in the black list with information to Juliet after a fixed number of not authorized requests is reached.

- **Privacy Enforcement Entity**:

  In the case: **Authorized and proactive enabled**

  Supports decision reuse of Julies by storing them. If there is enough information delivered about the watcher to securely identify a second call of the same watcher he will be authorized proactively during future requests.

  In the case: **Authorized once**

  Record the decision to allow additional support of Julie during future requests of Maria.

  In the case: **Not Authorized**

  Record the decision to avoid fast retry of unwanted requests.

#### 5.6.3.6 Normal Flow for Reactive Use Case:

1) Maria sends a request to Juliet
2) The Juliet receives the request
3) The Juliet decides but would like to know more about the watcher (See also the use cases for Decision Support). She calls Maria and identified her as a friend of a common friend.
4) Maria gets the requested information and is informed about Julie's decision to assign her to the authorization category friends

#### 5.6.3.7 Alternative Flow for Reactive Use Case 1:

1) Maria sends a request to Juliet
2) Juliet receives the request
3) Juliet decides but would like to know more about the watcher (See also the use cases for Decision Support). Since she is in a hurry she is only willing to allow Maria to get the default requesters access rights once for a short lifetime. (Authorization Category – Default; Lifetime)
4) 4) Maria gets the requested information Julie allows her to see for the default lifetime.

#### 5.6.3.8 Alternative Flow for Reactive Use Case 2:

1) Maria sends a request to Juliet

2) Juliet receives the request
3) Juliet decides to disallow Maria to get any information
4) Maria is assigned to a black list.
5) Maria gets the information that Julie didn't want to give her the requested data.

#### 5.6.3.9 Operational and Quality of Experience Requirements for Reactive Use Case:

Juliet is able to decide once and to let the system reuse her decision. Juliet is flexible in her decision:

- Since she felt secure she allows proactive authorization for the requesting watcher
- Since she felt insecure about a request she can allow limited access for a small time period but can also disallow any information going to the requestor.

### 5.6.4 Proactive Authorization:

#### 5.6.4.1 Short Description for Proactive Use Cases:

- A watcher named Romeo is known to the presentity (named Juliet) is authorized due to his membership to an authorization category of Juliet called "friends"

#### 5.6.4.2 Actors for Proactive Use Cases:

The simple proactive Use Case needs:

- **Watcher:** Romeo is interested in the Juliet's presence information (Asks for Data Access)
- **Privacy Enforcement Entity:** Acts on behalf of the Juliet. Normally she is not informed about the data access during proactive authorization.

#### 5.6.4.3 Actors Benefits:

- **Watcher:** The proactive authorization disables the need to disturb the presentity for presence information access. Presence enhanced callers can chose if their call is acceptable (depending on the item of the call) for the called party (e.g. respectfully call).
- **Presentity:** Is not disturbed by request for data access for presence information he would allow anyway. Allowing automated presence information distribution to selected communication partners enables more convenient communication.

#### 5.6.4.4 Pre-conditions for Proactive Use Cases:

- **Watcher**: Romeo is member of Juliet's authorization category called friends and is allowed to get access to her presence information.
- **Presentity**: Juliet delegates the authorization to the Privacy Enforcement Entity and has assigned the watcher (e.g. Romeo) as member of a watcher list with an authorization category (e.g. friends) in the past
- **Privacy Enforcing Entity**: Acts on behalf of Juliet and record some information

#### 5.6.4.5 Postconditions for Proactive Use Cases:

- **Watcher:** Romeo receives the requested data as result of the proactive authorization
- **Presentity:** Juliet is able to access recorded information that contain information who has been watching her presence information
- **Privacy Enforcing Entity:** Save recorded information to support Juliet with information about watcher activities

#### 5.6.4.6 Normal Flow for Proactive Use Cases:

1) Romeo wants to subscribe or fetch Juliet's presence information
2) The Privacy Enforcement Entity acting for Juliet verifies if Romeo is known and enabled for proactive authorization and which authorization category he is assigned to
3) Due to a positive result for authorization Romeo is subscribed for the presence information as requested
4) Romeo receives the presence information that reflects the result of his request

#### 5.6.4.7 Operational and Quality of Experience Requirements for Proactive Use Cases:

The authorization for presence information access of known communication partners is automated and is under the presentity's control.

### 5.6.5 Proactive Authorization – Common Group, Strictly Secure

#### 5.6.5.1 Short Description for Proactive Use Cases:

This use case describes a scenario where membership in common groups is strictly secure, i.e. the group owner has specified a closed membership model that he or she controls.

A watcher (new colleague named Bob) unknown to the presentity (named Juliet) is authorized proactively because both are members [metadata membership] of the same potentially externally hosted group (Company, Division, Department…).

- The presentity's authorization engine needs support by the group owner to identify if the watcher asking for presence information is really a member of the group as he disclosed to the presentity. Therefore the privacy enforcement entity sends a request to the group owner's entity (E.g. Companies Directory Service) asking for verification of the identity delivered by the requesting watcher. Only positive return values from the verification authority enable the watcher automatically. (First contact of watcher may be shown optional to the presentity.)

#### 5.6.5.2 Actors for Proactive Use Cases:

For the "common group" use case two modes for authorization arise alternatively:

- **Watcher**: Bob (a new colleague of Juliet) is interested in Juliet's presence information since he is asked by their boss to harmonize a presentation he should finish since Juliet is on a business trip but he didn't know where she is. The proactive authorization feature is able to inform Bob about her time zone details to support him making respectful calls (E.g. don't disturb her during the night hours).
- **Privacy Enforcement Entity**: Acts on behalf of Juliet which is normally not informed about the data access requests
- **Directory Service** (Company, Operator) activated by the Privacy Enforcement Entity of the presentity identifies that Bob as a group member of department (May be with Certificate)

#### 5.6.5.3 Actors Benefits:

- **Watcher** can use its membership as authorization to get information that supports his mobile communication.
- **Presentity** doesn't need to handle his own lists if he allows team members to get parts of his presence information. (Company, Sport Club)
- **Organizations:** Members are able to communicate more efficient. They do not spend time to manage other team members. The centralized directory service exists anyway within organizations and could be reused rather simple to support the organization members during mobile communication.

#### 5.6.5.4 Pre-conditions for Proactive Use Cases:

- **Watcher**: Bob is known to the company's directory service. Bob and Juliet share the membership to a list called like their department hosted by their company. This enables him to send Juliet as unknown watcher a request for her presence information.
- **Presentity**: Trust the Privacy Enforcement Entity using groups to verify watcher identity
- **Privacy Enforcing Entity**: Proactive authorization of external groups is enabled and Juliet allows the verification of the watchers identity as a key to automatically access her presence information (e.g. verify watchers metadata signature…)

#### 5.6.5.5 Postconditions for Proactive Use Cases:

- **Watcher** : Bob receives the requested data or an error message depending on the result of the proactive authorization
- **Presentity**: Juliet receives a notice that someone unknown before is accessing her presence information and is able to watch the membership assigned authorization lists s. (If notification of new proactive watchers is enabled)
- **Privacy Enforcing Entity**: Store the subscription of the requesting watcher to the presence information of Juliet. Keys received from the Directory Service during watcher identification might be stored to reduce transaction costs for future requests (proxy for the keys).
- **Directory Service** (Company, Service provider, …) Record information about approved identity and supporting information for identity verification. (E.g. public keys)

#### 5.6.5.6 Normal Flow for Proactive Use Cases:

1) Bob also sending metadata with signature about his relation to a common group (e.g. Company employee) wants to subscribe or fetch Juliet's presence information
2) The Privacy Enforcement Entity tries to verify the group relation
3) Due to the group type "Hosted by my Company" high security is necessary
4) The Identity delivered by the watcher is send to the company's certification authority
5) The Company's Certification Authority sends the public key of the watcher to the Privacy Enforcement Entity
6) The Privacy Enforcement Entity is now able to verify the metadata signature
7) In the positive case Bob unknown before but verified now is allowed to see the requested presence information controlled by the mapping between authorization profile and the group membership.
8) Juliet may be informed about the watcher that got her data

#### 5.6.5.7 Operational and Quality of Experience Requirements for Proactive Use Cases:

Supporting managed and externally hosted groups for proactive authorization simplifies the group communication for all group members. Groups are very effective in reaching a common treatment for a class of subscribers that need information to enable services that deliver more comfort to its users (e.g. respectfully call by using presence information).

This has to be done carefully since the service provider has to guarantee fraud protection in his own and the user's interest. Privacy needs to be enforced since membership declaration to a group could be faked.

### 5.6.6 Proactive Authorization – Common Group

#### 5.6.6.1 Short Description for Proactive Use Cases:

Due to different common group categories different levels of trust should be supported. This use case describes a model where a relationship of trust exists, as described below:

A watcher (new team member of a Volleyball team named Francesca) unknown to the presentity (named Julie) is authorized proactively because both are members of a group moderated by their trainer. Since Julie trusts her trainer that he is moderating the group very responsible she allows all group members to request her presence information with the authorization profile used for the group.

- **Trust by relation:** The moderator and owner of the group is someone the presentity trusts. He is the only one that is able to change membership. (E.g. Trainer of the Volleyball team) Any request for presence information the presentity allows if the watcher is a member of the moderated group (E.g. Volleyball team). The privacy enforcement entity should verify for the presentity if the watcher is known as group member. (New group member requests may be shown optional to the presentity.)

#### 5.6.6.2 Actors for Proactive Use Cases:

For the "common group" use case two modes for authorization arise alternatively:

- **Watcher:** Francis (a member of Juliet's Volleyball team) is interested in Juliet's presence information to identify if Juliet or some one else of the team is able to guide her to the match.
- **Privacy Enforcement Entity:** Acts on behalf of Juliet which is normally not informed about the data access requests
- **Trust by relation**: The presentity (Juliet) allows the Privacy Enforcement Entity to approve group members (e.g. Volleyball team) to access her presence information. May be restricted to a time frame booked as activity with that group (e.g. match of the Volleyball team Juliet is member of). (May be without Certificate)

#### 5.6.6.3 Actors Benefits:

- **Watcher** can use its membership as authorization to get information that supports his mobile communication.
- **Presentity** doesn't need to handle his own lists if he allows team members to get parts of his presence information. (Company, Sport Club)
- **Organizations**. Members are able to communicate more efficient. They do not spend time to manage other team members. The centralized directory service exists anyway within organizations and could be reused rather simple to support the organization members during mobile communication.

#### 5.6.6.4 Pre-conditions for Proactive Use Cases:

- **Watcher:** Bob is known to the company's directory service. Bob and Juliet share the membership to a list called like their department hosted by their company. This enables him to send Juliet as unknown watcher a request for her presence information.
- **Presentity:** Trust other members moderating the group member list
- **Privacy Enforcing Entity:** Proactive authorization of external groups is enabled and Juliet allows the automatic access of the group members

#### 5.6.6.5 Postconditions for Proactive Use Cases:

- **Watcher**: Bob receives the requested data or an error message depending on the result of the proactive authorization
- **Presentity:** Juliet receives a notice that someone unknown before is accessing her presence information and is able to watch the membership assigned authorization lists elements. (If notification of new proactive watchers is enabled)
- **Privacy Enforcing Entity**. Store the subscription of the requesting watcher to the presence information of Juliet. Keys received from the Directory Service during watcher identification might be stored to reduce transaction costs for future requests (proxy for the keys)
- **Directory Service (Company, Service provider, …)** Record information about approved identity and supporting information for identity verification. (E.g. public keys)

#### 5.6.6.6 Normal Flow for Proactive Use Cases:

This flow demonstrates the behaviour in a group where the group moderator is a person the presentity knows and trusts.

1) Francis is sending metadata about relations to a common group (e.g. Volleyball team) wants to subscribe or fetch Juliet's presence information
2) The Privacy Enforcement Entity verifies the group relation delivered by Francis. (Group Type "Hosted by a moderator of trust")
3) Francis is allowed to see the presence information of Juliet if the verification results are positive
4) The presentity may be informed about the authorization of a new group member

#### 5.6.6.7 Operational and Quality of Experience Requirements for Proactive Use Cases:

Supporting managed and externally hosted groups for proactive authorization simplifies the group communication for all group members. Groups are very effective in reaching a common treatment for a class of subscribers that need information to enable services that deliver more comfort to its users (e.g. respectfully call by using presence information).

This has to be done carefully since the service provider has to guarantee fraud protection in his own and the user's interest. Privacy needs to be enforced since membership declaration to a group could be faked.

### 5.6.7 Authorization Category:

#### 5.6.7.1 Short Description for Authorization Category:

All decision supporting use cases try to reduce the effort necessary for the presentity to decide about data requests by watchers. They are mainly relevant for the reactive mode since they support the presentity to prepare the proactive uses case.

Authorization Categories (**ACs**) support the presentity to categorize incoming requests of watchers by identifying and selecting a relation of trust. This selection automatically assigns all presence elements at once since they are predefined (normally by the service provider) for every single **AC**s. If the presentity needs other **AC**s he should be able to change them or to create new **AC**s.

Reactive authorization decisions about access to presence elements should be as much simplified as possible. Therefore **AC**s should be used by the presentity to formulate groups of people he has the same relation with. (E.g. Family, Colleagues, Friends, interesting Man…) The assigned presence elements for every AC guarantees the presentity the control of the elements shown to requesting watchers.

#### 5.6.7.2 Actors for Decision Support Use Cases:

- **Watcher**: Harry wants to fetch or subscribe to PIdata of Juliet
- **Presentity**: Juliet is willing to decide on request about her presence information (During reactive use case)
- **Privacy Enforcing Entity**:
  - Stores and manages supporting features (e.g. Authorization categories)
  - Enable persistency for decisions

#### 5.6.7.3 Actors Benefits:

- **Watcher**: Due to the simplified transaction for the presentity the chances to get what the watcher wants improve since it is less complicated to judge about data requests.
- **Presentity**: The assignment of a watcher to an Authorization Category is a very intuitive and easy understandable way to decide about privacy settings. The number of decision is reduced significantly without losing control

- **Privacy Enforcing Entity**: Due to a common mindset of the presentity and the watcher about a common AC it might be meaningful to support sharing them as an option.

#### 5.6.7.4 Pre-conditions for Decision Supporting Use Cases:

- **Watcher**: Harry wants to fetch or subscribe to the presence information of Juliet
- **Presentity**: Juliet wants to use ACs to decide about publishing her data. She owns some ACs (e.g. Operator defaults and/or self generated)
- **Privacy Enforcing Entity**: ACs are supported (e.g. Storage; Management; Predefined)

#### 5.6.7.5 Postconditions for Decision Supporting Use Cases:

- **Watcher:** Acknowledge for the request getting PIdata and/or status
- **Presentity:** Harry is assigned to a dedicated **AC**.
- **Privacy Enforcing Entity: AC** assignment is stored (Storage; Management)

#### 5.6.7.6 Normal Flow for Decision Supporting Use Cases:

1) Harry starts a request for Juliet's presence information
2) Juliet receives the request as reactive authorization request
3) Juliet decides to assign Harry for the **AC "**interesting Man"
4) Harry gets an acknowledge that reflects Juliet's decision

#### 5.6.7.7 Alternative Flow for Decision Supporting Use Cases:

1) Harry starts a request for Juliet's presence information
2) Juliet receives the request as reactive authorization request
3) Juliet couldn't remember Harry very well therefore she decides to assign Harry for the **AC "**interesting man" but only for a small time period not for proactive requests.
4) Harry gets an acknowledge that reflects Juliet's decision

#### 5.6.7.8 Operational and Quality of Experience Requirements for Decision Supporting Use Cases:

Juliet needs to understand and to easy control her presence information shown to different watchers. Additionally the number of decision about presence elements should be as low as possible. Therefore:

- Authorization Categories significantly reduce the number of decisions necessary to control data access for watchers

## 5.7 Open Issues

None.

# 6. Requirements (Normative)

## 6.1 High-Level Functional Requirements

### 6.1.1 General

1) The Presence Service SHALL be specified in such a manner that no specific execution environment, operating system, or programming language is unfairly favoured for the implementation of a conforming Presence Service.
2) Required interfaces to the Presence Service SHALL not be specified in terms that unfairly favour any execution environment, operating system, or programming language.
3) The Presence Service SHALL be independent of the technology of the access network. For example, it would be inappropriate to specify a Presence Service that works only for GPRS networks. This requirement SHALL NOT preclude making information specific to an access network available to the Presence Service.
4) The Presence Service SHALL interact with external presence services using industry-standard protocols and data formats to the extent enabled by those industry-standard protocols and data formats.
5) The Presence Service SHALL continue to be supported when the user roams to another network that supports the presence service.
6) The Presence Service SHALL comply with the privacy requirement described in [Privacy].
7) The presence service SHOULD allow efficient use of transport resources.

### 6.1.2 User Experience

1) It SHALL be possible for presentities to utilize the presence service in order to communicate to others certain information and preferences (presence information), such as their willingness and availability to communicate using particular communication means.
2) Presentities MAY communicate this information to the Presence Service by creating and activating Presence Profiles such as “Working”, “Meeting”, “Out to lunch”, “Discreet”, “Busy”, “Do Not Disturb”, etc.
3) Presentities SHALL be able to synchronize Presence Profiles with the Presence Service.
4) The Presence Service SHALL not limit such profiles to pre-specified content.
5) The Presence Service SHALL allow for such profiles to be suitably customized to meet the needs of a variety of applications and end-users.
6) While the number and content of those profiles will vary, the presence elements that will be communicated as a result SHALL be defined such that their semantics are very precise in order to ensure that they are consistently interpreted across applications. [For more information see Section 6.1.4 Presence Information]

### 6.1.3 Features

The Presence Service SHALL be configurable as detailed below.

1) It SHALL be possible for a service provider to assign appropriate permissions to certain individuals (e.g. system administrators), such that they can override settings of individual users.
2) Presence service SHALL support presence information for multiple, concurrent presence enabled services for each presence user.
3) Presence service SHALL support concurrent, multiple terminal devices for each presence user.

#### 6.1.3.1 Publish

1) The Presence Service and Presence Sources SHALL support the publication of presence information.

2) The Presence Service SHOULD support the publication of presence information on behalf of other presentities (refer to Delegation Section).

3) The Presence Service SHALL support the aggregation and storage of multiple presence elements for each user.

4) The Presence Service SHALL support the publication of one or more presence elements at a time.

5) The Presence Service SHALL support the retrieval of presence information from presence sources (network entities, users agents, etc.) either ad-hoc or on a periodic basis.

6) The Presence Service SHALL support the reception of presence information from authorized presence sources (network entities, users, etc.).

7) More than one Presence Source MAY publish the same presence elements on behalf of a presentity (refer to Delegation Section).

8) More than one Presence Source MAY publish the same presence elements on behalf of another presentity.

#### 6.1.3.2 Subscribe

1) Watchers SHALL be able to request the presence information of presentities (including lists that represent multiple presentities).

2) Watchers SHALL be able to request that notifications are sent on a subscription basis when there is new or modified presence information.

3) Watchers SHALL be able to request the presence information of presentities (including lists that represent multiple presentities) on subscription basis, where notifications are sent periodically, i.e., at regular intervals.

4) Watchers SHOULD be able to specify one or more conditions upon which presence notifications are generated and sent to them. These conditions SHOULD include at least: specific changes in presence status of a presentity or list of presentities; and, time constraint conditions, such as buffering or throttling mechanisms.

5) Watchers SHALL be able to specify that a particular subscription generates full or partial (i.e. incremental) notifications.

6) Presence subscriptions MAY have an expiration time (a.k.a. duration). When the duration of a subscription elapses, the subscription is terminated.

7) The Presence Service SHALL notify a Subscribed-watcher when their subscription expires, unless the Subscribed-watcher requested not to receive such notifications.

8) The Presence Service SHALL provide the means for a Subscribed-watcher to renew a subscription before it expires.

9) The Presence Service SHALL provide a mechanism for the subscribing watcher to request a particular subscription duration, which MAY be overridden by the subscribed-to presentity's preferences or configuration parameters of the service provider..

10) The Presence Service SHALL provide a mechanism to cancel a Subscribed-watcher's subscription.

11) The Presence Service SHALL provide a mechanism that can be used to notify a Subscribed-watcher of the cancellation of their subscription, subject to the preferences of the presentity (see next requirement).

12) The Presence Service SHALL provide a mechanism to allow a presentity to suppress a notification to a watcher regarding a cancelled subscription.

13) A presence service user SHALL have the ability to decide whether to accept or deny incoming presence subscription requests as those arrive. This is named reactive authorization.

14) A presence service user SHALL have the ability to define rules that will determine if future incoming subscription requests are accepted or denied. This is named proactive authorization.

15) Presence Service user SHALL have the ability to decide during the reactive authorization procedure to enable the watcher for proactive authorization for future requests.

16) Presence Service user SHALL determine which potential watchers or groups of watchers (e.g. friends, family) shall be proactively authorized to receive his/her Presence Information.

17) Presence Service user SHALL determine which potential watchers or groups of watchers (e.g. work mates) shall be reactively authorized in order to receive his/her Presence Information.

18) The Presence Service User MAY be provided with information related to the watchers that request his/her Presence information (e.g. name, MSISDN, etc).

19) A Subscribed-Watcher SHALL be notified as to whether the requested subscription was accepted or denied.

20) A presentity MAY deny an incoming subscription, while indicating it accepted it (polite blocking).

21) The Presence Service SHALL provide the means to enable a presentity to be notified about any unauthorized Watcher requests (whether ad-hoc, subscription-based, or otherwise) for th e Presentity's presence information.

22) The Presence Service SHALL provide the means to enable a presentity to be notified about any Subscriptions for the Presentity's presence information that have just expired.

23) A presentity SHALL be able to authorize a watcher to retrieve its presence information, via one or more of the mechanisms described here, on behalf of another watcher.

24) It SHALL be possible for a watcher to request that they receive a particular subset of a presentity's presence information, subject to the presentity's preferences.

25) It SHALL be possible for a watcher or a presentity to perform subscription-related operations in bulk, i.e. where the target is more than one presentity or watcher respectively.

26) It SHALL be possible for a subscribing watcher to specify a maximum desired notification frequency.

27) Presence service SHALL support One-time Event subscription and Notification

28) Presence Service User MAY be able to make One-time Event subscription to Presence Service

#### 6.1.3.3 Notify

1) The Presence Service SHALL be able to generate asynchronous notifications in response to subscribed-to events.

2) The Presence Service SHALL support a mechanism such the order of transmitted notifications can be maintained.

3) The Presence Service MAY cancel a subscription, if the notifications pertaining to that subscription are undeliverable.

4) It MAY be possible for the Presence Service to buffer or otherwise store notifications, so that the Subscribed-watcher, in lieu of asynchronous notifications, can retrieve them.

5) It MAY be possible to retrieve buffered notifications pertaining to more than one presentity in bulk.

#### 6.1.3.4 Preferences

1) Presentities SHALL be able to control how their presence information is disseminated.

2) Presentities SHALL be able to define policies such that the Presence Service disseminates different information to individual watchers or groups of watchers.

3) The defined policies SHALL cover the possibility of anonymous or unauthenticated watchers.

4) It SHALL be possible to define default policies that apply to watchers that do not fall in any of the specified groups.

5) It SHALL be possible to apply a policy to a particular watcher, to a particular request, or to a particular request type.

6) For each said watcher or group of watchers, presentities SHALL be able to define policies such that the Presence Service will reveal all their presence information, a subset of their presence information, or any other information (whether that is true or not), fully or partially based on their presence information.

7) The Presence Service SHALL provide mechanisms which may be used to limit the number of times a watcher can retrieve the Presence Information of a presentity.

8) Presentities and/or administrators SHALL be able to define default policies on a per-presentity, per-watcher, or per watcher group basis.

#### 6.1.3.5 Delegation

1) The Presence Service SHALL allow the selective authorization of presentities to perform publication related features on behalf of other presentities.
2) The Presence Service SHALL allow the selective authorization of presentities or watchers to perform subscription related features on behalf of other presentities or watchers.
3) The Presence Service SHOULD allow the selective authorization of presentities to configure preferences on behalf of other presentities.
4) The Presence Service SHOULD allow the selective delegation of features to the Presence Service, such that those features can be applied by the service when the presentities or watchers are out of contact.

### 6.1.4 Presence Information

#### 6.1.4.1 Presence Information Content

1) Presence information relating to a particular presentity SHALL be segmented in zero or more presence elements.
2) The Presence Service SHALL provide a common mechanism that can be used to associate priorities with particular presence information elements. The semantics of this prioritization will depend on the elements being prioritized. The definitions of those presence information elements will include the semantics of the prioritization.
3) Presence Service SHALL provide a means where presence elements may be associated with a time at which the presence element should no longer be considered valid

#### 6.1.4.2 Presence Information Format

1) The Presence Service SHALL support a format that is able to represent a rich set of presence information.
2) Presence Information SHALL be represented using a standard format, for the purpose of exchanging presence information.
3) A standard format and information semantics (including values where applicable) SHALL be defined for the following common information:
    a) Default Willingness (e.g. willing, not willing, etc.)
    b) Application-specific Willingness (e.g. willing for PoC, not willing for IM etc);
    c) Overriding Willingness (e.g. willing, not willing);
    d) Application-specific Availability (e.g. registered with the PoC service);
    e) Network Availability (e.g. the phone is attached to the network, out-of-coverage, etc.);
    f) Communication address (e.g. email address, phone number, etc.);
    g) Presentity supplied activity and location
        i) Activity (e.g. in a meeting, at the movies, on the phone etc.);
        ii) Textual location (e.g. at home, at work, at the supermarket, etc.);
    h) Location (e.g. device-derived location, network-derived location, etc.);
    i) Client device capabilities
        i) Application capabilities(e.g. voice, text,, multimedia, etc.);
        ii) Bearer capabilities (e.g. UMTS, GPRS etc);
    j) Time-zone (e.g. GMT etc);

k) Personal information

i) Mood (e.g. textual: happy, angry, sad, etc. or picture: smiley face, frowning face, etc.)

ii) Hobbies (football, fishing, computing, dancing, etc.).

iii) Preferred language (e.g. English, Spanish etc);

iv) Icon (e.g. a status icon of the presentity's choice)

4) The Presence Information format SHALL comply with standard IETF formats, where relevant.

5) The Presence Information format SHALL be registered with IANA as a MIME-type.

6) The Presence Information format SHOULD use a standard mark-up language.

7) In order to transfer Presence Information over a wireless link (e.g. low bandwidth, high latency, and high error rate link) it may be necessary to define an additional format. In this case, appropriate mappings to the standard format SHALL be defined.

8) The Presence Information format SHALL be able to represent the Presence Information as a set of zero or more presence elements.

9) The Presence Information format SHALL provide the means to uniquely identify a presence element.

10) The Presence Information format SHALL provide the means to associate a presence element with an expiration date.

11) It SHOULD be possible to extend the presence format, without affecting previously defined aspects.

12) The Presence Information format SHALL support multiple character sets.

13) The Presence Information format SHALL include a way to identify the presentity to which it pertains.

14) The Presence Information format SHOULD include a way to include Presence Information indirectly (e.g. by providing a link to a different location)

#### 6.1.4.3 Enabler specific issues

1) The Presence Service SHALL specify the presence elements in such a way that they can be used consistently and without ambiguity across multiple enablers.

2) Enablers that use the presence service SHOULD re-use the presence elements defined above where appropriate, instead of redefining them in an application-specific manner.

3) The Presence Service SHALL allow other enablers to define new presence elements that are application specific.

4) Enablers that use the presence service and need to define new presence elements SHOULD define a standard format and information semantics (including values where applicable) for those presence elements.

### 6.1.5 Group Management for the Presence Service

1) The Presence Service SHALL allow presentities and watchers to utilize group lists, e.g. contact lists as defined in [XDMREQ].

### 6.1.6 Network Interfaces

1) The Presence Service SHALL support a SIP-based network interface [RFC-3261].

2) The supported network interfaces SHALL make it possible for a logical entity, such as presentity or watcher, to simultaneously access the Presence Service from multiple physical locations.

3) The supported network interfaces SHALL be suitable for a variety of other enablers or applications to access the Presence Service.

4) The supported network interfaces SHALL be designed to support extensions, while maintaining backwards compatibility.

## 6.1.7 Security

1) Presence Service SHALL include mechanisms to securely authenticate entities that require access to the Presence Service.
2) Presentities and watchers SHALL support mechanisms to securely authenticate themselves to the Presence Service.
3) The supported network interfaces SHALL include mechanisms to support non-authenticated watchers that require access to the Presence Service.
4) The supported network interfaces SHALL include suitable mechanisms to prevent denial-of-service attacks.
5) The supported network interfaces SHALL include suitable mechanisms to prevent replay attacks.
6) The supported network interfaces SHALL include suitable mechanisms to prevent maintain the privacy of exchanged information.
7) The supported network interfaces SHALL include suitable mechanisms to prevent third parties from interfering with the provided services.
8) The supported network interfaces SHALL include suitable mechanisms to verify the authenticity of the source of the published presence information.
9) The supported network interfaces SHALL include suitable mechanisms to verify the integrity of exchanged messages.

## 6.1.8 Presence Sources and Watchers

The Presence source and watchers SHALL

1) Support the authentication with the Presence service.

The Presence Source and watchers SHOULD

2) Support default presence settings (e.g. default profile, default groups) that are automatically selected when a device is powered up for the first time (5.2.5).

## 6.1.9 Collecting accounting information

The Presence service SHALL collect accounting information for all presence transactions.

The Presence Service SHALL support the following:

- Both online and offline charging.
- Pre-paid and post-paid charging.
- Different tariff rules depending on service providers' policies.
- Flat fee: per time period independent of usage.
- Correlation between presence service charging data and transport or bearer level charging data (e.g. charging at GPRS).
- Correlation between presence service charging data and session level charging data (e.g. charging at IMS).
- Correlation between presence service charging data and other presence service enabled service's charging data (e.g. charging of PoC).

### 6.1.9.1 Charging of Presentity

The charging of presentity can be made on at least the following events:

- Presence Information Publication

The tariff rule can be based on at least the following criteria:

- The size of the presence information notified to watchers.
- The number of watchers subscribed.

#### 6.1.9.2 Charging of Watcher

The charging of watcher can be made on at least the following events.

- Presence Information Subscriptions
- Presence Information Notifications
- Searching for Presentities

The tariff rule can be based on at least the following criteria.

- The size of the presence information retrieved from the Presence Server
- The number of presentities subscribed to.

### 6.1.10 Operational & Quality of Experience

1) Presence Service notifications SHALL be sent out as close as possible to the generating event, subject to throttling requirements.

### 6.1.11 Interoperability between Presence Service Providers & Service Entities

1) Presence users SHALL be able to seamlessly utilise Presence features involving other Presence users regardless of their Presence service provider. For example, a list of presentities to subscribe to may include a presentity that is subscribed to another service provider.

# Appendix A. Change History (Informative)

## A.1 Approved Version History

| Reference | Date | Description |
|---|---|---|
| OMA-RD-Presence_SIMPLE-V1_0 | 25 Jul 2006 | TP approved:<br>OMA-TP-2006-0223R04-INP_Presence_SIMPLE_V1_0_for_final_approval |
| Approved Version<br>OMA-RD-Presence_SIMPLE-V1_1 | 27 Jun 2008 | Status changed to Approved by TP<br>TP ref# OMA-TP-2008-0250-<br>INP_Presence_SIMPLE_V1_1_ERP_for_Final_Approval |
| Draft Version<br>OMA-RD-Presence_SIMPLE-V1_0 | 28 Jan 2009 | Incorporated CRs:<br>OMA-PAG-2008-0851<br>OMA-PAG-2008-0852<br>OMA-PAG-2008-0864R01 |
| Draft Version<br>OMA-RD-Presence_SIMPLE-V1_0 | 12 Feb 2010 | Modification of 28 Jan 2009 history box row due to re-classification of CR: OMA-PAG-2008-0864 became OMA-PAG-2008-0864R01 |
| Approved Version<br>OMA-RD-Presence_SIMPLE-V1_1_1 | 25 Feb 2010 | Status changed to Approved by TP<br>TP ref# OMA-TP-2010-0110-<br>INP_Presence_SIMPLE_V1_1_1_ERP_for_notification |