Presence Service

Russ Clark
September 29, 2008
What is Presence?

• A service that indicates the ability and willingness of a user to communicate

• Supports multiple devices and interfaces

• Perhaps the most important SIP application service for the IMS architecture

• Nearly all interesting IMS applications should leverage the presence service

• Instant Messaging (IM) is closely linked with Presence
Presence Architecture

Presentity - Presence Entity

PUA - Presence User Agent
Gathers presence info and publishes it to the service.

PA - Presence Agent
The main presence servers

PUBLISH

SUBSCRIBE

NOTIFY

Watchers - End users or other Services
Presence Standards

- RFC 2778 - A Model for Presence and Instant Messaging - lots of terminology definitions
- RFC 3856 - Presence Event Package for SIP
- RFC 3863 - Presence Information Data Format (PIDF)
- OMA: Open Mobile Alliance
  - SIMPLE: SIP Instant Messaging and Presence Leveraging Extensions from the Presence and Availability working group
- 3GPP Presence Service Framework
Presence User Agents

Each PUA may present different types of presence information

- Online Status - open, closed
- Availability - available, busy, current activity
- Available for - text chat, voice, video, etc
- Mood - tolerance for interrupts

The PA (service) must merge all of these inputs to create a unified view of the Presentity.

Other information is available based on the IMS registration status.
Privacy

Users must be able to control

• what information the PUA can access
• what information is published to the PA
• which watchers can see the data
• which subsets of the data are available to each watcher

For usability, some form of group feature is necessary.
Presence Data Format

PIDF: Presence Information Data Format

- RFC 3863
- XML document format
- minimal specification
- designed to be extensible
- many current efforts to extend this
  - additional semantics, details
  - temporal information - when this data applies
  - geographic location
## Presence Data Format

### Service-related information

<table>
<thead>
<tr>
<th><code>&lt;tuple&gt;</code></th>
<th>Service-related information</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;status&gt;</code></td>
<td>An indication whether the presentity is able to receive an incoming communication request</td>
</tr>
<tr>
<td></td>
<td>for the described service (if specified)</td>
</tr>
<tr>
<td><code>&lt;registration-state&gt;</code></td>
<td>An indication whether the presentity has an active registration with the described service</td>
</tr>
<tr>
<td></td>
<td>(if specified)</td>
</tr>
<tr>
<td><code>&lt;barring-state&gt;</code></td>
<td>An indication whether the presentity has activated communication barring for the described</td>
</tr>
<tr>
<td></td>
<td>service (if specified)</td>
</tr>
<tr>
<td><code>&lt;willingness&gt;</code></td>
<td>An indication whether the presentity wants to receive incoming communication requests for</td>
</tr>
<tr>
<td></td>
<td>the described service (if specified)</td>
</tr>
<tr>
<td><code>&lt;status-icon&gt;</code></td>
<td>A small icon, e.g. to represent the service in a GUI</td>
</tr>
<tr>
<td><code>&lt;session-participation&gt;</code></td>
<td>An indication of the presentity's involvement in at least one session of the described</td>
</tr>
<tr>
<td></td>
<td>service (if specified)</td>
</tr>
<tr>
<td><code>&lt;service-description&gt;</code></td>
<td>An identification of the service by means of a service ID and a version number, which is</td>
</tr>
<tr>
<td></td>
<td>optionally enhanced with a short textual description.</td>
</tr>
<tr>
<td><code>&lt;deviceID&gt;</code></td>
<td>An indication of the device on which the described service is running.</td>
</tr>
<tr>
<td><code>&lt;class&gt;</code></td>
<td>The <code>&lt;tuple&gt;</code> information's class</td>
</tr>
<tr>
<td><code>&lt;contact&gt;</code></td>
<td>The URI to be used to invoke the service</td>
</tr>
<tr>
<td><code>&lt;note&gt;</code></td>
<td>Free text</td>
</tr>
<tr>
<td><code>&lt;timestamp&gt;</code></td>
<td>The latest update/refresh of the provided service-related information</td>
</tr>
</tbody>
</table>

### Device-related information

<table>
<thead>
<tr>
<th><code>&lt;device&gt;</code></th>
<th>Device-related information</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;network-availability&gt;</code></td>
<td>An indication of the network(s) to which the device is connected</td>
</tr>
<tr>
<td><code>&lt;network&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;geopriv&gt;</code></td>
<td>The current (geographical) location of the device (a postal address or geographical</td>
</tr>
<tr>
<td></td>
<td>coordinates)</td>
</tr>
<tr>
<td><code>&lt;class&gt;</code></td>
<td>The <code>&lt;device&gt;</code> information's class</td>
</tr>
<tr>
<td><code>&lt;deviceID&gt;</code></td>
<td>The device's unique identifier</td>
</tr>
<tr>
<td><code>&lt;note&gt;</code></td>
<td>Free text</td>
</tr>
<tr>
<td><code>&lt;timestamp&gt;</code></td>
<td>The latest update/refresh of the provided device-related information</td>
</tr>
</tbody>
</table>
Presence Mood?

The 'mood' presence attribute comprises:
- Informative text in one message (optional)
- Either the unknown value, or at least one of the following values:

<table>
<thead>
<tr>
<th>Afraid</th>
<th>Amazed</th>
<th>Angry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annoyed</td>
<td>Anxious</td>
<td>Ashamed</td>
</tr>
<tr>
<td>Bored</td>
<td>Brave</td>
<td>Calm</td>
</tr>
<tr>
<td>Cold</td>
<td>Confused</td>
<td>Contented</td>
</tr>
<tr>
<td>Cranky</td>
<td>Curious</td>
<td>Depressed</td>
</tr>
<tr>
<td>Disappointed</td>
<td>Disgusted</td>
<td>Distracted</td>
</tr>
<tr>
<td>Embarrassed</td>
<td>Excited</td>
<td>Flirtatious</td>
</tr>
<tr>
<td>Frustrated</td>
<td>Grumpy</td>
<td>Guilty</td>
</tr>
<tr>
<td>Happy</td>
<td>Hot</td>
<td>Humbled</td>
</tr>
<tr>
<td>Humiliated</td>
<td>Hungry</td>
<td>Hurt</td>
</tr>
<tr>
<td>Impressed</td>
<td>In-awe</td>
<td>In_love</td>
</tr>
<tr>
<td>Indignant</td>
<td>Interested</td>
<td>Invincible</td>
</tr>
<tr>
<td>Jealous</td>
<td>Lonely</td>
<td>Mean</td>
</tr>
<tr>
<td>Moody</td>
<td>Nervous</td>
<td>Neutral</td>
</tr>
<tr>
<td>Offended</td>
<td>Playful</td>
<td>Proud</td>
</tr>
<tr>
<td>Relieved</td>
<td>Remorseful</td>
<td>Restless</td>
</tr>
<tr>
<td>Sad</td>
<td>Sarcastic</td>
<td>Serious</td>
</tr>
<tr>
<td>Shocked</td>
<td>Shy</td>
<td>Sick</td>
</tr>
<tr>
<td>Sleepy</td>
<td>Stressed</td>
<td>Surprised</td>
</tr>
<tr>
<td>Thirsty</td>
<td>Worried</td>
<td>Other &lt;+text&gt;</td>
</tr>
</tbody>
</table>
A presence document describing the following:
- PoC-Session Specific Availability: Available/Registered/ISB not activated
- PoC-Session Specific Willingness: Willing
- Activity: Meal & some comment
- Mood: Happy & Cheerful
- Geographical Location: Coord <X> and <Y>.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
          xmlns:op="urn:oma:prs:pidf:oma-pres"
          xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
          xmlns:gml="urn:ogc:specification:gml:schema-xsd:
                     feature:v3.0"
          entity="sip:someone@example.com">
  <tuple id="a1232">
    <status>
      <basic>open</basic>
    </status>
    <op:willingness>
      <op:basic>open</op:basic>
    </op:willingness>
    <op:registration-state>active</op:registration-state>
    <op:barring-state>terminated</op:barring-state>
    <op:service-description>
      <op:id>org.openmobilealliance;
      PoC-session</op:id>
      <op:version>1.0</op:version>
    </op:service-description>
    <contact>sip:someone@example.com</contact>
    <timestamp>2005-02-23T12:14:56Z</timestamp>
  </tuple>
  <pdm:person id="a1233">
    <rpid:activities>
      <rpid:note>Very tasteful!</rpid:note>
      <rpid:meal/>
    </rpid:activities>
    <rpid:mood>
      <rpid:other>cheerful</rpid:other>
      <rpid:happy/>
    </rpid:mood>
    <gp:geopriv>
      <gp:location-info>
        <gml:Point gid="point1" srsName="epsg:4326">
          <gml:coordinates>
            <gml:X>30 16 28S</gml:X>
            <gml:Y>45 15 33W</gml:Y>
          </gml:coordinates>
        </gml:Point>
      </gp:location-info>
      <gp:usage-rules/>
    </gp:geopriv>
    <pdm:timestamp>2005-02-23T12:14:56Z</pdm:timestamp>
  </pdm:person>
</presence>
```
A presence document describing the following:
- PoC-Alert Specific Availability: Not Available/Registered/ISB activated
- PoC-Alert Specific Willingness: Not Willing
- Mood: happy
- Location: Restaurant
- Geographical Location: 77 Downing Street, London, United Kingdom
- Icon: http://example.com/~someone/myicon.gif
- Device Identifier: urn:uuid:4866e19-5fbf-43fc-a2fd-d23002787599
- Network-Availability: IMS-registered.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:op="urn:oma:xml:prs:pidf:oma-pres"
  <tuple id="a1232">
    <status>
      <basic>closed</basic>
    </status>
    <op:willingness>
      <basic>closed</basic>
    </op:willingness>
    <op:registration-state>active</op:registration-state>
    <op:barring-state>active</op:barring-state>
    <op:service-description>
      <op:service-id>org.openmobilealliance:PoC-Alert</op:service-id>
      <op:version>1.0</op:version>
      <op:description>This is the OMA PoC-Alert service</op:description>
    </op:service-description>
    <pdm:deviceID>urn:uuid:4866e19-5fbf-43fc-a2fd-d23002787599</pdm:deviceID>
    <contact>sip:someone@example.com</contact>
    <timestamp>2005-02-22T20:07:07Z</timestamp>
  </tuple>
</presence>
```
We need to manage lists of contacts with whom we share presence information.

- A list of watchers, identified by their public identifier (e.g. sip:Russ.Clark@gatech.edu)
- Need to be able to add, modify, delete entries
- The list is maintained in a separate service
- RLS: Resource List Server
- Instead of subscribing individually to presence information for all of your contacts, you subscribe to the list.
- The list server subscribes to the individuals
Figure 17.2: SIP-based presence architecture in the IMS

(From The 3G IP Multimedia Subsystem (IMS) Second Edition by Gonzalo Camarillo and Miguel A. García-Martín © 2006 John Wiley & Sons, Ltd)
17.3 WATCHER SUBSCRIPTION

Figure 17.2: SIP-based presence architecture in the IMS

(From The 3G IP Multimedia Subsystem (IMS) Second Edition by Gonzalo Camarillo and Miguel A. García-Martín © 2006 John Wiley & Sons, Ltd)
The flow is illustrated in Figure 17.3. The watcher application residing in the IMS terminal sends a SUBSCRIBE request (1) addressed to her list (e.g., sip:alice-list@home1.net). The SUBSCRIBE request contains an Event header field set to eventlist to indicate that the subscription is addressed to a list rather than a single presentity. The request (2) is received at the S-CSCF, which evaluates the initial filter criteria. One of those criteria indicates that the request (3) ought to be forwarded to an Application Server that happens to be an RLS. The RLS, after verifying the identity of the subscriber and authorizing the subscription, sends a 200 (OK) response (4). The RLS also sends a NOTIFY request (7), although it does not contain any presence information at this stage. The RLS subscribes one by one to all the presentities listed in the resource list and, when enough information has been received, generates another NOTIFY request (13) that includes a presence document with the aggregated presence information received from the presentities’ PUAs.

Figure 17.4 shows the RLS subscribing to one of the presentities contained in the resource list. The RLS sends a SUBSCRIBE request (1) addressed to a presentity in the list. The request contains an Event header field set to presence. The request is forwarded via the S-CSCF in the RLS home network (2) to the I-CSCF in the presentity’s network. The I-CSCF queries the HSS using the Diameter protocol, (3), (4) in Figure 17.4, to locate the S-CSCF allocated to the presentity and forwards the SUBSCRIBE request (5) to the allocated S-CSCF. The S-CSCF evaluates the initial filter criteria where there is a criterion indicating that the request ought to be forwarded to the presentity’s PA (6). After sending the 200 (OK) response (7) the PA sends a NOTIFY request (11) that contains the presentity’s presence information.

In the example in Figure 17.3 neither the presentity’s S-CSCF nor the presentity’s I-CSCF record the route. Therefore, the PA sends the NOTIFY request (11) directly to the next
The flow is illustrated in Figure 17.3. The watcher application residing in the IMS terminal sends a SUBSCRIBE request (1) addressed to her list (e.g., sip:alice-list@home1.net). The SUBSCRIBE request contains an Event header field set to eventlist to indicate that the subscription is addressed to a list rather than a single presentity. The request (2) is received at the S-CSCF, which evaluates the initial filter criteria. One of those criteria indicates that the request (3) ought to be forwarded to an Application Server that happens to be an RLS. The RLS, after verifying the identity of the subscriber and authorizing the subscription, sends a 200 (OK) response (4). The RLS also sends a NOTIFY request (7), although it does not contain any presence information at this stage. The RLS subscribes one by one to all the presentities listed in the resource list and, when enough information has been received, generates another NOTIFY request (13) that includes a presence document with the aggregated presence information received from the presentities’ PUAs.

Figure 17.4 shows the RLS subscribing to one of the presentities contained in the resource list. The RLS sends a SUBSCRIBE request (1) addressed to a presentity in the list. The request contains an Event header field set to presence. The request is forwarded via the S-CSCF in the RLS home network (2) to the I-CSCF in the presentity’s network. The I-CSCF queries the HSS using the Diameter protocol, (3), (4) in Figure 17.4, to locate the S-CSCF allocated to the presentity and forwards the SUBSCRIBE request (5) to the allocated S-CSCF. The S-CSCF evaluates the initial filter criteria where there is a criterion indicating that the request ought to be forwarded to the presentity’s PA (6). After sending the 200 (OK) response (7) the PA sends a NOTIFY request (11) that contains the presentity’s presence information.

In the example in Figure 17.3 neither the presentity’s S-CSCF nor the presentity’s I-CSCF record the route. Therefore, the PA sends the NOTIFY request (11) directly to the next

(Figure 17.2: SIP-based presence architecture in the IMS

(From The 3G IP Multimedia Subsystem (IMS) Second Edition  Gonzalo Camarillo and Miguel A. García-Martín © 2006 John Wiley & Sons, Ltd)
We discussed in Section 4.15.1 a few optimizations that are applicable to the event notification framework in SIP. These are generic optimizations that apply to any event package. In Section 16.15 we discussed some other optimizations that are applicable to the presence event package. These optimizations were standardized in the IETF in co-operation with engineers from a wireless background. The result is a standard that is not only applicable to any Internet host, but also to wireless devices. Consequently, the IMS networks and terminals implement all the extensions that provide presence optimization.

As mentioned before, 3GPP defined the Ut interface that runs between an IMS terminal and an Application Server. When we refer to the presence service, the Application Server is either a PUA or an RLS.

The Ut interface is not used for live traffic. Instead, it provides the user with the ability to configure resource lists (such as presence lists), authorize watchers in the PUA, and do any other type of data manipulation.

The Ut interface is implemented with HTTP 1.1 (Hypertext Transfer Protocol, specified in RFC 2616 [101]). On top of it there is the XML Configuration Access Protocol (XCAP), specified in the Internet-Draft "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)" [210], and on top of XCAP there is an XCAP usage that depends on the particular aim.
17.4 Subscription to Watcher Information

When the presence application is started in the IMS terminal, PUAs subscribe to their own watcher information state, so they can be updated on who is watching their presence information and with the state of the subscriptions to their presence states. We described the watcher information subscription in Section 16.12.

The high-level flow is similar to the one we presented in Figure 17.3. The Event header field in the SUBSCRIBE request (1) is set to the value presence.winfo to indicate to the PA the state the presentity is interested in.

Figure 17.3: Watcher subscription to her own list
When the IMS presence application starts, it publishes the current presentity's presence information. Figure 17.5 shows the flow, and we can see there is not much difference from the mechanism we explained in Section 16.10. The IMS terminal sends a PUBLISH request (1) that includes a `Event` header set to `presence`. The S-CSCF receives the request (2) and evaluates the initial filter criteria for the presentity. One of the initial filter criteria indicates that PUBLISH requests containing a `Event` header set to `presence` ought to be forwarded to the PA where the presentity's presence information is stored. So, the S-CSCF forwards the PUBLISH request (3) to that Application Server. The PA authorizes the publication and sends a 200 (OK) response (4).

Apart from the IMS terminal, other PUAs can publish presence information related to the presentity. For instance, an S-CSCF can send a third-party REGISTER request to the presentity’s PA to indicate that the presentity has registered with the IMS. Any other entity that contains presence information for that presentity can publish it to the PA or to a Presence Network Agent, that in turn publishes it to the presentity’s PA.

Figure 17.4: The RLS subscribes to a presentity

Scaling the service

The PA and RLS are important for scaling the service to multiple PAs and multiple watchers

• I send my presence updates once to the PA
• If multiple individuals are subscribed, they can each get info from the PA
• If multiple lists are subscribed, the RLS only needs to fetch the info once from the PA
Presence Data Volume

Large volumes of data with frequent updates

Need to manage the impact on the network and on the devices.

1) Event Throttling
   - Watcher sets minimum update interval

2) SIP Signaling Compression
   - RFC 3320, 3321 SIGCOMP
   - A dictionary based compression algorithm
   - Between UE and Proxy - focused on the wireless link
3) Partial Notification

- Much of the information in a presence update is redundant
- A partial update indicates only the information that has changed
- In SUBSCRIBE, watcher includes the following:
  
  \[ \text{Accept: application/pidf-partial+xml} \]
- First NOTIFY message includes the full presence state
- Subsequent NOTIFY messages have only changes
Universal Presence

• Today, we have many different presence services.
• We really need to combine this in one federated presence service.
• I shouldn’t have to have all of these clients running.
• This is a place that IMS could perhaps play a role.