Presence and Location Service: Operation and Client Requirements

version 0.2

Jamey Hicks <jamey.hicks@hp.com>

Operation

This document describes the operation of the Presence and Location Server (PALS) used by the Internet2 Presence Integrated Communication Working Group. The intent of this document is to highlight the requirements on rich-presence enabled clients (user agents) working with the PALS.

This document is a work in progress. Please send feedback on the document to the PIC working group at the following address:

```
<wg-pic@internet2.edu>
```

Domain Hijacking

In most situations, the domain name of a user's address of record will match the domain name of the SIP registrar and PALS. If we followed this practice in the demo, we would have to assign a new address of record to each user. The PIC working group felt that would be too much overhead to running the demo and participating in the demo.

Instead, the UA's are configured to user the participants own email address as the address of record, but registered to the `pals.internet2.edu` SIP Registrar and PALS. This server is also used by the UA as the outgoing proxy.

Registration follows the standard protocol, but the registrar/proxy keeps both user name and domain name for each entry to distinguish users with the same user names but different domain names. When a non-registration message arrives at the server, the full address of record is looked up in the contact table and the message is forked to each contact URI registered for that address.

Registration and Publishing Presence

Figure 1 shows the sequence of messages for user Bob's UA getting online, registering, and publishing Bob's presence information.

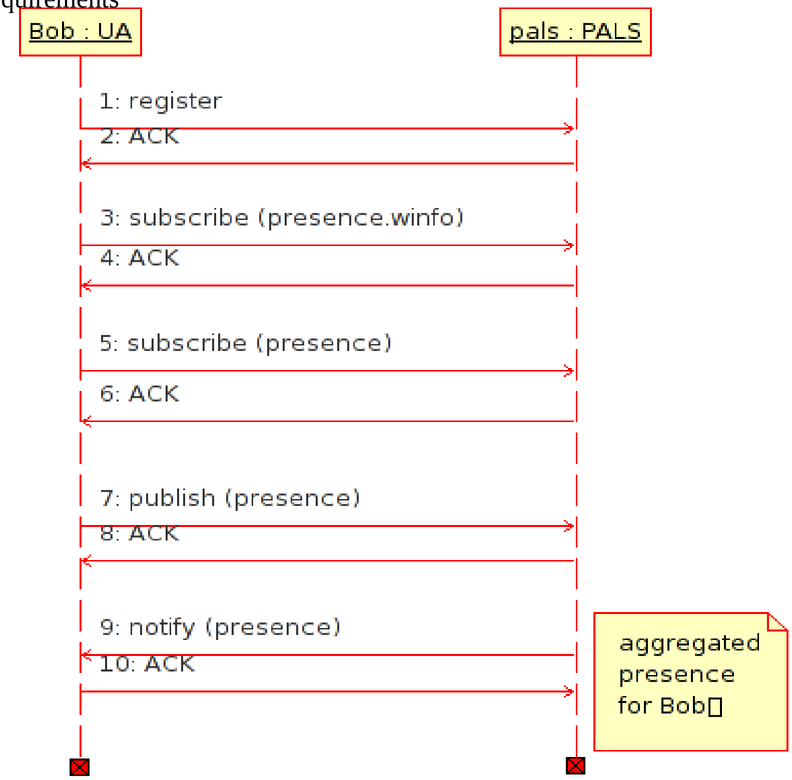

Figure 1: UA Registration and Publication of Presence

In the pals registration picture, a typical user (Bob) turns on his user agent. The UA registers with the proxy server, which is also a presence and location server (PALS). In message 3, Bob's UA subscribes to presence.winfo event package on bob@example.com so that he will be notified when anyone subscribes to his presence info. That is covered in the second diagram.

Because the PALS also publishes presence info, Bob's UA subscribes in message 5 to the presence event package [PEP] on bob@example.com in order to receive aggregated presence info from all of Bob's user agents and from location information gathered by the PALS.

The PALS can determine basic online/offline via the registration message, but more detailed information must be published [PUBLISH] by Bob's UA in message 7. Messages 5 and 7 could have been issued in either order.

In message 9, the PALS sends a notify because Bob's presence information was changed by message 7. There could also have been a notify after the message 5 subscribe, depending on the timing of the messages.

Discussion

One thing to note about the operation of SUBSCRIBE messages in this architecture is that they are handled by the PALS server and are not forwarded to the UA's. One rationale is we needed a service to aggregate presence. Another rationale is that the PALS may be able to respond with some information even if a user is currently offline.

Managing Buddy Lists via XCAP

The UA's manage buddy (contact) lists via XCAP according to [XCAP-LIST]. The rationale for doing this in the demos is to get users up-to-speed as quickly as possible. Each user's contact list will be pre-populated with each of the other participants demo contact info. In other usage scenarios, XCAP-LIST enables users to synchronize their contact lists across multiple UA’s.

The URL for the presence list for user `bob@example.com` is:

- `http://pals.internet2.edu/presence-lists/users/bob@example.com/presence.xml`

UA’s SHOULD fetch the initial contact list for the demo via XCAP. The format of the response will be application/cpim-pidf+xml. The response will be a full response.

If additional contacts are added to the contact list, the UA SHOULD update the contact list on the PALS via an XCAP PUT containing a full copy of the presence list document.

Subscribing to a User’s Presence Information

Figure 2 shows the sequence of messages for a previously unauthorized user Alice subscribing to Bob’s presence information.

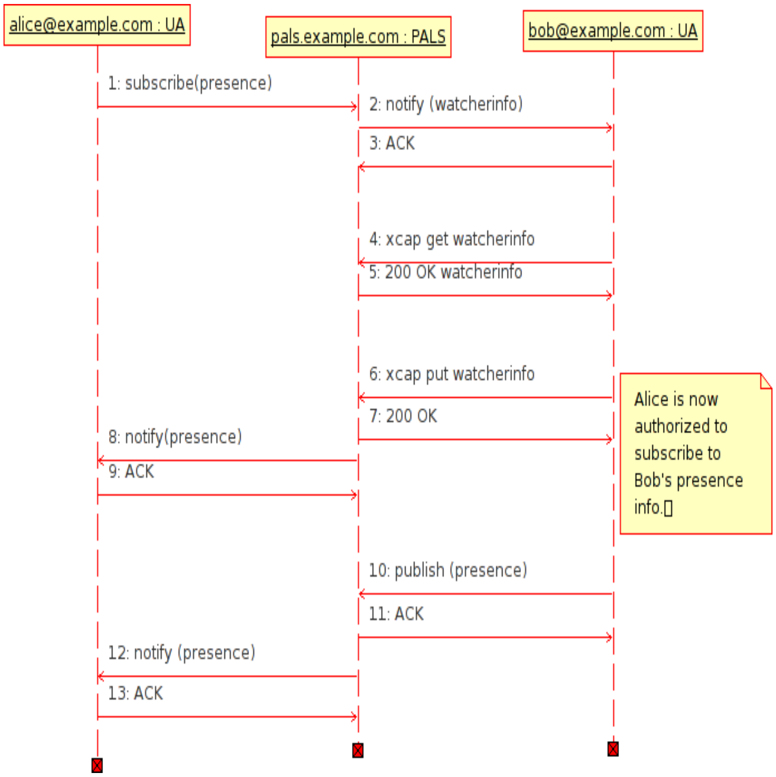

Figure 2: Previously Unauthorized User Subscribing

To begin with, Alice’s UA sends message 1 to the PALS server pals.example.com to request to subscribe to Bob’s presence information. The event package is "presence" and the accepts format is "application/cpim-pidf+xml", which is defined in [CPIM-PIDF].

Because alice@example.com is not currently authorized to subscribe to Bob’s presence information, PALS sends a presence.winfo notification [WINFO] in format application/watcherinfo+xml [WATCHERINFO] to Bob’s UA. Upon receipt of the notification, Bob’s UA would query the Bob whether to authorize Alice to watch him.

Bob authorizes Alice, and his UA uses an XCAP [XCAP] GET request (message 4) to fetch the watcherinfo list from the PALS. This usage of XCAP to manage watcher authentication conforms to [XCAP-AUTH].

For the demo, the URL that is fetched is: `http://pals.internet2.edu/watcherinfo/users/bob@example.com/watc`

Bob's UA updates the watcher list, changing the status of the entry for alice@example.com from "pending" to "active", and then puts the updated list back on the server via XCAP PUT in message 6, acknowledged by message 7.

Updating the watcherinfo on PALS triggers it to send a presence notification (message 8) to Alice, containing the current state of Bob's presence. This message is in format application/cpim-pidf+xml.

Messages 9 and 10 contain PUBLISH and ACK of new presence information for Bob. This update triggers PALS to send a new presence notification to Alice's UA (message 11 and ack 12).

Presence Information Format

The overall presence information format is CPIM-PIDF [CPIM-PIDF]. We also use the extensions defined in [RPID]. In addition, PALS adds location information (location-info elements). The contents of the location-info element will conform to the format described in [GEOPRIV-PIDF-LO].

Here is an example presence document:

Example 1. Example cpim-pidf+xml presence document

```
<?xml version="1.0" encoding="UTF-8"?>
<impp:presence xmlns:impp="urn:ietf:params:xml:ns:pidf"
    entity="pres:someone@example.com">
  <impp:tuple id="sg89ae">
    <impp:status>
      <impp:basic>open</impp:basic>
      <gp:geopriv>
        <gp:location-info>
          <gml:location>
            <cl:civilAddress>
              <cl:country>US</cl:country>
              <cl:A1>New York</cl:A1>
              <cl:A3>New York</cl:A3>
              <cl:A6>Broadway</cl:A6>
              <cl:HNO>123</cl:HNO>
              <cl:LOC>Suite 75</cl:LOC>
              <cl:PC>10027-0401</cl:PC>
            </cl:civilAddress>
           </gml:location>
        </gp:location-info>
      </gp:geopriv>
    </impp:status>
    <impp:contact priority="0.8">tel:+09012345678</impp:contact>
  </impp:tuple>
</impp:presence>
```

Example 2. RPID presence document with privacy tag

For example, if room 123 has a session of the conference running, so the PALS would add a privacy tag to the presence documents of users in room 123.

```
<status>
  <basic>open</basic>
  <ep:privacy>quiet</ep:privacy>
</status>
```

Schema for HP Labs Location Format

Example 3. Example cpim-pidf+xml presence document with HP Labs-format location info

```
<?xml version="1.0" encoding="UTF-8"?>
<impp:presence xmlns:impp="urn:ietf:params:xml:ns:pidf" xmlns:crl="urn:crl:location"
    entity="pres:someone@example.com">
  <impp:tuple id="sg89ae">
    <impp:status>
      <impp:basic>open</impp:basic>
      <gp:geopriv>
        <gp:location-info>
          <gml:location>
            <cl:civilAddress>
              <loc>
                <crl:site>East West Center</cl:A1>
                <crl:floor>2</cl:A1>
                <crl:room>Kamehameha</cl:A1>
              </loc>
            </cl:civilAddress>
           </gml:location>
        </gp:location-info>
      </gp:geopriv>
    </impp:status>
    <impp:contact priority="0.8">tel:+09012345678</impp:contact>
  </impp:tuple>
</impp:presence>
```

Describe the additional schema elements here.

<crl:site> Multi-site organizations often have their own internal designators for a site in addition to its civil address. Occurs within a >CivilLocation< element.

<crl:floor> Indicates the floor of the building where the UA is located. Occurs within a >CivilLocation< element.

<crl:room> Indicates the room of the building where the UA is located. Occurs within a >CivilLocation< element.

<crl:x> Indicates the X coordinate of the position of the UA relative to the origin location of the building in which the UA is located. Occurs within a >CivilLocation< element.

<crl:y> Indicates the Y coordinate of the position of the UA relative to the origin location of the building in which the UA is located. Occurs within a >CivilLocation< element.

<crl:radius> Indicates the radius of the position estimate of the UA relative to the origin location of the building in which the UA is located. Occurs within a >CivilLocation< element.

<crl:weather> Indicates the network weather in the vicinity of the UA. Occurs within a >status< element.

Instant Messaging

UA’s that support Instant Messaging should support the `message/CPIM` message format. [CPIM-MSGFMT]

Referencing and Fetching Maps

The presence document for a user MAY include the URL of a map as an extension to [GEOPRIV-PIDF-LO]. This is not currently supported by the PIC-WG PALS.

Example 4.

```
<gp:geopriv>
<gp:location-info>
<gp:map url="http://www.examples.com/floor4/map.html"/>
<cl:civilAddress>
<cl:country>US</cl:country>
<cl:A1>New York</cl:A1>
<cl:A3>New York</cl:A3>
<cl:A6>Broadway</cl:A6>
<cl:HNO>123</cl:HNO>
<cl:LOC>Suite 75</cl:LOC>
<cl:PC>10027-0401</cl:PC>
</cl:civilAddress>
</gp:location-info>
```

When a UA (such as sipc) gets the presence notification, it can retrieve the map and display the presentity’s location on the map.

UA Capabilities

Presence and Location Service: Operation and Client Requirements

In the demo setting and in real life, users have a mix of UA's supporting a subset of audio, video, and text communication. A UA may support more modes than the user will use, because of personal preferences, activities, or context.

When opening communication to another user, it would be helpful to know what modes are supported by that user. There are two methods proposed for doing this, [CALLEE-CAPS] and [PRESCAPS].

In [CALLEE-CAPS], the UA adds fields to the contact header in a REGISTER message encoding various pre-defined and extensible callee capabilities. Callers learn the capabilities of callees by sending an OPTIONS message to the user's UA. In the PIC-WG architecture, the registration process could add this capability information to the CPIM-PIDF document that is sent in NOTIFY messages to watchers.

In [PRESCAPS], callee capabilities are advertised by publishing them as part of the CPIM-PIDF presence document for the user. This method seems to fit more naturally into the SUBSCRIBE/PUBLISH/NOTIFY event architecture.

Bibliography

[GEOPRIV-PIDF-LO] *A Presence-based GEOPRIV Location Object Format*. draft-ietf-geopriv-pidf-lo-01.txt.

[CPIM] *Common Profile for Instant Messaging (CPIM)*. draft-ietf-impp-im-04.txt.

[PEP] *A Presence Event Package for the Session Initiation Protocol (SIP)*. draft-ietf-simple-presence-10.txt.

[CPIM-PIDF] *Presence Information Data Format (PIDF)*. draft-ietf-impp-cpim-pidf-08.txt.

[CPIM-MSGFMT] *Common Presence and Instant Messaging: Message Format*. draft-ietf-impp-cpim-msgfmt-08.txt.

[RPID] *RPID - Rich Presence Information Data Format*. draft-ietf-simple-rpid-01.

[CIPID] *CIPID: Contact Information in Presence Information Data Format*. draft-ietf-simple-cipid-00.

[WATCHERINFO] *An Extensible Markup Language (XML) Based Format for Watcher Information*. draft-ietf-simple-winfo-format-04.txt.

[WINFO] *A Watcher Information Event Template-Package for the Session Initiation Protocol (SIP)*. draft-ietf-simple-winfo-package-05.txt.

[XCAP] *The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)*. draft-ietf-simple-xcap-02.

[XCAP-AUTH] *Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usages for Setting Presence Authorization* . draft-ietf-simple-xcap-auth-usage-01.txt.

[XCAP-LIST] *An Extensible Markup Language (XML) Format for Representing Resource Lists*. draft-ietf-simple-xcap-list-usage-02.

[PUBLISH] *An Event State Publication Extension to the Session Initiation Protocol (SIP)*. draft-ietf-sip-publish-03.txt.

[SIP] *SIP: Session Initiation Protocol*. RFC 3261.

[SIP-EVENT] *Session Initiation Protocol (SIP)-Specific Event Notification*. RFC 3265.

[SIP-INFO] *The SIP INFO Method*. RFC 2976.

Presence and Location Service: Operation and Client Requirements

[SIP-IM] *Session Initiation Protocol (SIP) Extension for Instant Messaging*. RFC 3428.

[SIP-REFER] *The Session Initiation Protocol (SIP) Refer Method*. RFC 3515.

Glossary

watcher A user subscribing to a type of information about another user. A watcher is notified with updated values of that information when it changes.

event package A type of information to which a watcher can subscribe.