Internet Engineering Task Force
INTERNET-DRAFT
draft-schulzrinne-simple-rpids-02.ps
H. Schulzrinne (ed.)
Columbia U.
February 21, 2003
Expires: August 2003

RPIDS – Rich Presence Information Data Format for Presence Based on the Session Initiation Protocol (SIP)

Status of this Memo

Copyright Notice

Abstract

The Rich Presence Information Data Format for the Session Initiation Protocol (SIP) (RPIDS) adds elements to the Presence Information Data Format (PIDF) that provide additional information about the presentity, its contacts and their capabilities. This information can be translated into call routing behavior or be delivered to watchers. The information is designed so that much of it can be derived automatically, e.g., from calendar files or user activity. The capabilities information is compatible with the caller preferences extensions to SIP, but does not depend on these.

Contents

1 Introduction

The PIDF definition [1] describes a basic presence infornation data format for exchanging presence information in CPIM-compliant systems. It consists of a <presence> root element, zero or more <tuple> elements carrying presence information, zero or more <note> elements and zero or more extension elements from other name spaces. Each tuple defines a basic status of either "open" or "closed".

This document provides additional status information for presentities and defines a *Rich Presence Information Data Format for Presence Based on the Session Initiation Protocol (SIP)* (RPIDS) to convey this information.

This extension has three main goals:

1. Provide rich presence indication that is at least as powerful as common commercial presence systems. Such feature-parity simplifies transition to CPIM-compliant systems, both in terms of user acceptance and protocol conversion.

2. Allow easy translation between SIP user agent capabilities expressed as caller preferences information [2] and presence information.

3. Maintain compatibility with PIDF, so that PIDF-only watchers and gateways can continue to function properly.

We make no assumptions how the information in the RPIDS is generated. Experience has shown that users are not always diligent about updating their presence status. Thus, we want to make it as easy as possible to derive RPIDS information from other information sources, such as calendars, the status of communication devices such as telephones, typing activity and physical presence detectors as commonly found in energy-management systems.

The information in a presence document can be generated by a single entity or can be composed from information published by multiple entities.

Many of the elements correspond to data commonly found in personal calendars. Thus, we attempted to align some of the extensions with the usage found in calendar formats such as iCal [12] and xCal [13], as noted below.

Note that PIDF documents and this extension can be used in two different contexts, namely by the presentity to publish its presence status and by the presence server to notify some set of watchers. The presence server MAY compose, translate or filter the published presence state before delivering customized presence information to the watcher. For example, it may merge presence information from multiple PUAs, remove whole elements, translate values in elements or remove information from elements. Mechanisms that filter calls and other communications to the presentity can subscribe to this presence information just like a regular watcher and in turn generate automated rules, such as scripts [14], that govern the actual communications behavior of the presentity.

The flow diagram below illustrates this process.

```
presentity
   |
   --> publish
          |
          --> PA (filter)
                    --> notification 1 to A, B, C
                    --> notification 2 to D, E
                    --> notification 3 to F
                    --> notification 4 to script gen.
```

2 RPIDS Features

Below, we summarize and motivate the major additional features that RPIDS adds to PIDF.

The PIDF definition does not clearly describe what a <tuple> represents. We add an <class> element (Section 7.5) that labels each tupel as being a presentity, a group of presentities or a device.

While the PIDF definition describes which means of communications are available for a presentity, it does not describe the activity that the presentity is currently engaged in. The <category> (Section 7.4) element adds this information.

To help the watcher gauge the appropriateness of different types of communications, we indicate the type of place the user is currently in, via the <placetype> element (Section 7.9).

PIDF defines a <timestamp> element indicating the date and time of the status change of a tuple. RPIDS adds a validity period for status values, <from> and <until>, as a hint how long the current status is likely to be valid (Section 7.6 and Section 7.13).

The <activity> (Section 7.2) and <idlesince> (Section 7.7) provide information on when the device has last been used.

Presence information can provide hints as to how interruptible the presentity is, thus aiding in finding a time and manner of communications that is mutually convenient for both watcher and presentity. The "priority" callee capability described in [2]. This appears to be more expressive than the simple "do-not-disturb" indication found in some IM and presence systems.

An important sub-case is that a presentity is interruptible only under unusual circumstances, after mediation by some, typically human, authority such as a secretary or supervisor. We allow the presentity to convey that certain contact addresses actually belong to a different person, presumably one that can either interrupt the presentity or otherwise assist. The <relationship> (Section 7.11) element allows to indicate that a particular tuple refers to a different principal or presentity.

The PIDF document format [1] defines a <contact> element which may appear once inside every <tuple> element. The content of the <contact> element encodes the CONTACT ADDRESS and CONTACT MEANS as defined in [3]. The <contact> element is defined to be an URI. This URI can be of any URI scheme. Some URI schemes uniquely identify the application the tuple intends to describe (e.g., "im" URIs). However, this is not be the case for all schemes. For example, a SIP URI can represent different kinds of applications, including voice, video, or messaging. If it is not known by other means, it can be hard for applications processing the presence document containing only SIP URI contact addresses to know what particular application the tuple intends to describe. Also, watchers receiving presence information would benefit for getting more descriptive information about what particular communication means or applications are supported by the presentity. Section 7.3 shows how the callee capabilities mechanisms from [2] can fill this gap.

PIDF only defines tuples for one presentity. In many cases, it is useful to allow presentities to refer to groups of other presentities. For example, a presentity `all@example.com` might consist of

`marketing@example.com`,
`engineering@example.com`, `finance@example.com`.

`engineering@example.com` might in turn have presentities

`alice@example.com`,
`bob@example.org` (an intern), `carol@example.com`

We add multiple layer to PIDF by defining an extension (Section 7.14) that can in turn contain multiple PIDF presence elements, thus allowing recursion.

We establish the convention that a tuple that has no contact address indicates face-to-face communications. PIDF already notes that "there might be tuples not related to any communications means".

We generally assume that the presence element describes a single human being or a group of humans. However, this is not required. A presentity can also be a "bot" or "avatar", for example.

Note that this document does not defined a new content type. Rather, it inherits the content type from [1], namely `application/cpim-pidf+xml`.

3 Scope

This extension does not replace media negotiation mechanisms defined for SIP (e.g. SDP [4]), therefore media negotiation (e.g., choice of voice and video codecs) MUST be performed according to [5]. This extension is only aimed to give the watchers hints about the presentity's preferences, willingness and capabilities to communicate before watchers initiate SIP-based communication with the presentity.

4 Terminology and Conventions

This memo makes use of the vocabulary defined in the IMPP Model document [3]. Terms such as CLOSED, INSTANT MESSAGE, OPEN, PRESENCE SERVICE, PRESENTITY, WATCHER, and WATCHER USER AGENT in the memo are used in the same meaning as defined therein. The key words MUST, MUSTNOT, REQUIRED, SHOULD, SHOULDNOT, RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in BCP XX, RFC 2119 [6].

5 The Meaning of "open" and "closed"

PIDF describes the basic status values of "open" or "closed" only as "have meanings of general availability for other communications means". We define "closed" in our context as meaning that communication to the contact address will in all likelihood not succeed, is undesired or will not reach the intended party. (For example, a presentity may include a hotel phone number as a contact. After check-out, the phone number will still ring, but reach the chambermaid or the next guest. Thus, it would be declared "closed".) For "pres" contacts, "closed" means that no presence status information is available.

> The interpretation of "closed" was chosen since there is no other status value to indicate that a communications address is not reachable. Omitting the <contact> element does not work since it would confuse watchers that have not previously seen an "open" status for the same contact address.

6 Groups of Presentities

In many practical applications, a watcher wants to subscribe to groups of presentities rather than individuals. For example, the group membership may change over time and it may thus be difficult to subscribe to all members. If the group is large, the effort of subscription and their renewals may add significant burden to the watcher.

There are several different approaches to group subscriptions:

Group only: The watcher subscribes to a group and only cares about the status of the group as a whole. There is no protocol difference to subscribing to an individual and thus no need for extensions.

Subscription only: The watcher subscribes to a group, but receives individual notifications. This does not require an extensions to PIDF. Details are in [15].

Subscription with redirection: The watcher subscribes to a group. The presence document identifies the group members and allows the watcher to subscribe to each member individually. In PIDF, this is expressed by a "pres" URI in the <contact> element. Each such presentity can in turn be a group, recursively. TBD: How does the watcher find out if group membership has changed? We don't want to list all members in each PIDF notification. This basically becomes draft-roach-sip-list-template.

Subscription with full status: A single notification contains tuples from all presentities that have changed status since the last notification. We allow "recursive" presence definitions, where a <presence> element contains other <presence> elements, encapsulated as <members> (Section 7.14).

7 RPIDS Elements

7.1 Introduction

Below, we describe the RPIDS elements in detail. <activity>, <capabilities>, <category>, <class>, <from>, <idlesince> <label>, <placetype>, <privacy>, <relationship>, <timed-status>, <until> extend <status>. <members> extends the <presence> element.

In general, it is highly unlikely that a presentity will publish or announce all of these elements at the same time. Rather, these elements were chosen to give the presentity maximum flexibility in deriving this information from existing sources, such as calendaring tools, device activity sensors or location trackers, as well as to manually configure this information.

The namespace URI for elements defined by this specification is a URN [7], using the namespace identifier 'ietf' defined by [8] and extended by [9]:

```
urn:ietf:params:xml:ns:sip-rpids
```

7.2 Activity Element

The <activity> element describes whether the owner of the device has recently been actively using the device or not. It can take the values "active" and "inactive". For example, for a PC, the value "inactive" may be inferred from the lack of keyboard and mouse activity. For a telephone, an ongoing call translates into "active".

> The idle indication has been available in many "finger" implementations for several decades.
>
> The <activity> indication provides a qualitative indication that reveals less information to watchers than the <idlesince> element

.

7.3 Device Capabilities

The *Caller Preferences and Callee Capabilities* document [2] defines a set of extensions which allow callers and callees to express preferences about how requests should be handled in SIP servers. The same information also provides value to presence watchers so that they can make more informed decisions on how the presentity should be contacted. This information is generally used by a human watcher to choose the appropriate means of contact and therefore, matching rules that mediate between caller preferences and callee capabilities are not needed here.

Below, we present the caller preferences and callee capabilities extension namespace, its elements, their values, and semantics. This section also describes how this extension can be further extended. This extension is intended to be used with the `application/cpim-pidf+xml` content type and that particular usage is described here. This extension may also be used with other content types if appropriate.

7.3.1 Capabilities Extension Overview

This extension adds most of the features presented in Caller Preferences and Callee Capabilities document [2] into the PIDF presence document format. Features presented in [2] are added here without any change in their semantics, simply by mapping them into XML elements. The priority feature is not needed here because PIDF can already express this notion.

When this extension namespace is congregated with base PIDF document, combined document MUST follow the formatting rules specified in [1], Section 4.1.

7.3.2 Handling of Caller Preferences and Callee Capabilities Matching Rules

The Caller Preferences and Callee Capabilities document defines a set of matching rules which can be used to determine if capabilities represented by two different URLs are equal. This extension does not provide means to compare two presence documents to determine whether capabilities represented by those two documents match.

7.3.3 <capabilities> Element

The root element of this extension namespace is <capabilities>. The root element MUST be always present. This element MAY contain one or more <feature> elements as specified below.

The <capabilities> element does not have any attributes and it MAY contain other namespace declarations for other extensions used in the presence XML document.

7.3.4 <feature> Element

The <feature> element can represent any feature tag as defined in Caller Preferences and Callee Capabilities [2] or any media feature tag registered through [10]. This element MAY appear under <capabilities> element one or more times. ¡feature¿ element has one mandatory attribute called name. This attribute represents the name of the feature tag. Each media feature tag (e.g., media, feature) SHOULD appear only once inside the root element. If a media feature tag has multiple values, these values SHOULD be nested inside a single <feature> element. The <feature> cannot be extended using other namespace declarations.

7.3.5 <value> Element

The <value> element represents the value of media feature tag. Each value element SHOULD contain only a single tag value. If an <feature> contains multiple values, then all these values should be presented in different <value> elements. The <value> element has an optional attribute called “negated”. This attribute is of boolean type. It can be used to indicate whether the feature value is supported or not. The value ’true’ indicates that the value is not supported and the value ’false’ indicates that the value is supported. The default value is ’false’. The <value> element cannot be extended using other namespace declarations.

7.3.6 Handling of the Caller Preferences and Callee Capabilities Priority Feature Tag

The PIDF document format allows priority attribute to be used within <contact> element. The Caller Preferences and Callee Capabilities priority feature tag can be mapped into this attribute and so there is no need to represent the priority feature tag in this extension.

7.4 Category Element

The <category> indication describes what the presentity is currently doing. This can be quite helpful to the watcher in judging how appropriate a communication attempt is and which means of communications is most likely to succeed and not annoy the presentity. The activity indications correspond roughly to the category field in calendar entries, such as Section 4.8.1.2 of RFC 2445 [9].

> Use of an enumerated, but extensible, set of activity categories simplifies automated generation and processing of presence information. The categories can be readily selected from a drop-down list by the user or translated from the corresponding category field in calendars. Recipients of this information can render at least a subset as icons, automatically translate them into different languages or convert them to sound "jingles" and speech, or use them to generate call processing rules.

A category indication consists of one or more values drawn from the list below, any other token string or IANA-registered values (Section 11). Communities of interest such as a profession or an organization may define additional activity labels for their internal use.

On-the-phone: The presentity is talking on the telephone. This category is included since it can often be derived automatically.

Away: The presentity is physically away from the device location. This category was included since it can often be derived automatically from security systems, energy management systems or entry badge systems.

Appointment: The presentity has a calendar appointment.

Holiday: This is a scheduled national or local holiday. This information can typically be derived automatically from calendars.

Meal: The presentity is scheduled for a meal. This category can often be generated automatically from a calendar.

Meeting: This category can often be generated automatically from a calendar.

Steering: The presentity is controlling a vehicle, ship or plane.

In-transit: The presentity is riding in a vehicle, such as a car, but not steering.

Travel: The presentity is on a business or personal trip, but not necessarily in-transit. This category can often be generated automatically from a calendar.

Vacation: This category can often be generated automatically from a calendar.

Busy: User is busy, without further details. This category would typically be indicated manually.

Permant-absence: Presentity will not return for the foreseeable future, e.g., because it is no longer working for the company.

7.5 Contact-Type Element

The <contact-type> element describes the type of the tuple. A tuple can represent a communication facility ("device"), a single presentity ("individual") or a group of presentities ("group"). Additional classes can be registered with IANA.

> URI schema are insufficient to distinguish the different types of tuples. For example, a SIP URI can designate a single device, a presentity, or a group of presentities.

7.6 From Element

The <from> element indicates how long the current status has been valid, expressed as an absolute time.

7.7 Idlesince Element

The <idlesince> records the time and date the communication device was last used. This provides an indication as to how likely a user is to answer the device. Depending on the device, this element can be used together with <activity>, either "active" or "inactive". For example, a keyboard activity detector may still declare a PC that has not seen keyboard activity in two minutes as "active". For session-based devices such as telephones and video conferencing systems, <idlesince> would only be used with an activity value of "inactive".

7.8 Label Element

The <label> attribute is used by the presentity to label tuples. The value is chosen arbitrarily and MUST NOT be modified by a composing server or PA. There is no requirement that all tuples within a presence document differ in their label or have a label at all. Typically, the label remains the same across subscriptions and across watchers.

> The <label> makes it easier for policies to operate on presence documents. The 'id' <tuple> attribute is not guaranteed to remain constant across subscriptions. The PIDF specification does not prevent a PA from modifying the 'id' attribute. An element, rather than an attribute, was chosen since it appears less likely to cause interoperability problems with plain PIDF parsers.

7.9 Type of Place Element

The <placetype> element describes the type of place the presentity is currently at. This offers the watcher an indication what kind of communication is likely to be appropriate. We define an initial set of values below:

home: The presentity is in a private or residential setting, not necessarily the personal residence of the presentity, e.g., including hotel or a friend's home.

office: The presentity is in a business setting, such as an office.

public: The presentity is in a public area such as a shopping mall, street, park, public building, train station, airport or in public conveyance such as a bus, train, plane or ship.

This list can be augmented by free-text values or additional IANA-registered values (Section 11).

7.10 Privacy Element

public: Others may be able to see or hear the communications.

private: Inappropriate individuals are not likely to see or hear the communications.

quiet: The presentity is in a place such as a library, restaurant, place-of-worship, or theater that discourages noise, conversation and other distractions.

This indication is not limited to voice communications. For example, a presentity might label her privacy as "quiet" when giving a talk, since it would be inappropriate if an instant message popped up on the laptop screen that is being projected for the audience.

7.11 Relationship Element

The <relationship> element designates the type of relationship an alternate contact has with the presentity. This element is provided only if the tuple refers to somebody other than the presentity. Relationship values include "family", "associate" (e.g., for a colleague), "assistant", "supervisor". Other free-text values and additional IANA-registered values (Section 11) can be used as well.

The <contact> element can contain either a communication URI such as "im", "sip"/"sips", "h323", "tel" or "mailto", or a presence URI, such as "pres" or "sip".

7.12 Timed Status Element

The <timed-status> element describes status information that is either no longer valid or covers some future timeperiod.

> Timed status cannot be expressed with <tuples> elements where the period between <status> since PIDF parsers would not be able to distinguish current from future or past information. It is occasionally useful to represent past information since it may be the only known presence information; it may give watchers an indication of the current status. For example, indicating that the presentity was at a meeting that ended an hour ago indicates that the presentity is likely in transit at the current time.

7.13 Until Element

The <until> element indicates how long the current basic status (open or closed) is likely to be valid, expressed as an absolute time.

This indication allows the watcher to make better decisions. For example, if a presentity indicates that it is likely to be unreachable for an extended period of time, the watcher may decide to request assistance from somebody else, rather than waiting for the presentity to return.

Often, the duration of the status information is not known precisely. Thus, it is helpful to indicate the precision, here expressed in seconds. For example, an absence of "a few hours" can easily be expressed as a time some hours into the future, with a precision of 7200 seconds.

> An absolute time was chosen to simplify integration with calendaring applications. This combination appears to be sematically cleaner than enumerating various measurement units such as "months", "weeks", "days" or "hours".

Both the <from> and <until> information might be derived from calendar information, reflecting the start and end time of an activity. (Examples include the Date Time Start and Date Time End properties of RFC 2445. For simplicity, RPIDS only supports single events, without repetition.)

Any statements such as anticipated validity are not historical facts and are forward-looking statements that involve risks and uncertainties; actual results may differ from the forward-looking statements.

7.14 Members

The <members> element contains zero or more <presence> elements, each describing a member of the group. It is not necessary to provide the <basic> status for each member.

> Since the extension namespace for <presence> is restricted to ##other, we cannot include the PIDF <presence> directly.

8 Examples

8.1 Single Presentity with Capabilities

```
<?xml version="1.0" encoding="UTF-8"?>
   <presence xmlns="urn:ietf:params:xml:ns:cpim-pidf"
        xmlns:cap="urn:ietf:params:xml:ns:sip-prescaps"
        xmlns:ep="urn:ietf:params:xml:ns:sip-rpids"
        entity="pres:someone@example.com">

     <note>I'm in a boring meeting</note>

     <tuple id="7c8dqui">
       <status>
         <basic>open</basic>
         <contact>sip:secretary@example.com</contact>
         <cap:feature name="Media">
           <cap:value>voice</cap:value>
           <cap:value negated="true">message</cap:value>
         </cap:feature>
         </cap:capabilities>
       </status>
       <ep:relationship>assistant</ep:relationship>
       <note>My secretary</note>
     </tuple>

     <tuple id="18x765">
       <status>
         <basic>open</basic>
         <ep:category>meeting</ep:category>
         <ep:placetype>office</ep:placetype>
         <ep:privacy>quiet</ep:privacy>
         <ep:activity>inactive</ep:activity>
         <ep:idlesince>2003-01-27T10:43:00Z</ep:idlesince>
         <ep:until>2003-01-27T17:30:00Z</ep:unitl>
```

```
        <cap:capabilities>
          <cap:feature name="Media">
            <cap:value>voice</cap:value>
            <cap:value>message</cap:value>
          </cap:feature>
        </cap:capabilities>
      </status>
      <contact priority="0.8">sip:someone@example.com</contact>
      <timestamp>2001-10-27T16:49:29Z</timestamp>

      <ep:timed-status>
        <basic>closed</basic>
        <ep:from>2003-01-27T17:30:00Z</ep:from>
        <ep:until>2003-01-27T19:30:00Z</ep:until>
      </ep:timed-status>
    </tuple>

    <tuple id="35bs9r">
      <status>
        <basic>open</basic>
      </status>
      <contact priority="0.8">im:someone@mobilecarrier.net</contact>
      <timestamp>2001-10-27T16:49:29Z</timestamp>
    </tuple>

    <tuple id="8eg92n">
      <status>
        <basic>open</basic>
      </status>
      <contact priority="1.0">mailto:someone@example.com</contact>
    </tuple>
  </presence>
```


8.2 Multiple Presentities


```
<?xml version="1.0" encoding="UTF-8"?>
  <presence xmlns="urn:ietf:params:xml:ns:cpim-pidf"
       xmlns:im="urn:ietf:params:xml:ns:cpim-pidf:im"
       xmlns:ep="urn:ietf:params:xml:ns:sip-rpids"
       entity="pres:engineering@example.com">

  <tuple id="478">
    <status>
      <basic>open</basic>
    </status>
```

```
</tuple>

<members>
  <presence ... entity="pres:alice@example.com">
    <tuple id="1">
      <status>
        <basic>open</basic>
      </status>
      <contact>sip:alice@example.com</contact>
    </tuple>
  </presence>

  <presence ... entity="pres:bob@example.com">
    <tuple id="2">
      <status>
        <basic>closed</basic>
      </status>
      <contact>sip:bob@example.com</contact>
    </tuple>
  </presence>

  <presence ... entity="pres:widget-engineering@example.com">
    <tuple id="3">
      <status>
        <contact-category>group</contact-category>
      </status>
    </tuple>
  </presence>

</members>

</presence>
```

9 XML Schema Definitions

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:ietf:params:xml:ns:sip-prescaps"
   xmlns:tns="urn:ietf:params:xml:ns:sip-prescaps"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   elementFormDefault="qualified"
   attributeFormDefault="unqualified">

   <!-- This import brings in the XML language attribute xml:lang-->
   <xs:import namespace="http://www.w3.org/XML/1998/namespace"
   schemaLocation="http://www.w3.org/2001/xml.xsd"/>
```

```
      <xs:element name="capabilities">

      <xs:complexType name="capabilities">
         <xs:sequence>
             <xs:element name="feature" type="tns:Media" minOccurs="0"
                      maxOccurs="unbounded"/>
             <xs:any namespace="##other" processContents="lax" minOccurs="0"
                      maxOccurs="unbounded"/>
         </sequence>
      </complexType>

      <xs:complexType name="feature">
         <xs:sequence>
            <xs:element name="value" minOccurs="0"
                      maxOccurs="unbounded">
               <xs:complexType>
                  <xs:simpleContent>
     <xs:extension base="xs:string">
             <xs:attribute name="negated" type="xs:boolean"
             use="optional" default="false"/>
     </xs:extension>
     </xs:simpleContent>
         </xs:complexType>
            <xs:attribute name="name" type="xs:string" use="required"/>
         </xs:sequence>
      </xs:complexType>
   </xs:schema>

<?xml version="1.0" encoding="UTF-8"?>
   <xs:schema targetNamespace="urn:ietf:params:xml:ns:sip-rpids"
      xmlns:tns="urn:ietf:params:xml:ns:sip-rpids"
      xmlns:pidf="urn:ietf:params:xml:ns:cpim-pidf"
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
      elementFormDefault="qualified"
      attributeFormDefault="unqualified">

  <!-- This import brings in the XML language attribute xml:lang-->
      <xs:import namespace="http://www.w3.org/XML/1998/namespace"
      schemaLocation="http://www.w3.org/2001/xml.xsd"/>

  <xs:element name="contact-type" type="tns:contact-type"/>
  <xs:element name="placetype" type="xs:token"/>
  <xs:element name="privacy" type="tns:privacy"/>
  <xs:element name="category" type="xs:token"/>
```

```
<xs:element name="relationship" type="xs:token"/>
<xs:element name="from" type="tns:fromuntil">
<xs:element name="until" type="tns:fromuntil">
<xs:element name="idlesince" type="xs:dateTime">

<xs:element name="timed-status" type="tns:timed-status">

<xs:simpleType name="contact-type">
  <xs:restriction base="xs:string">
     <xs:enumeration value="individual"/>
     <xs:enumeration value="device"/>
     <xs:enumeration value="group"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="privacy">
  <xs:restriction base="xs:string">
     <xs:enumeration value="private"/>
     <xs:enumeration value="public"/>
     <xs:enumeration value="quiet"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="fromuntil">
  <xs:simpleContent>
    <xs:extension base="xs:dateTime">
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

<xs:complexType name="timed-status">
  <xs:sequence>
    <xs:element name="basic" type="pidf:basic" minOccurs="0"/>
    <xs:element name="from" type="tns:fromuntil">
    <xs:element name="until" type="tns:fromuntil">
    <xs:element name="note" type="pidf:note">
    <xs:any namespace="##other" processContents="lax" minOccurs="0"
       maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="members">
  <xs:sequence>
    <xs:any namespace="pidf" processContents="lax" minOccurs="0"
       maxOccurs="unbounded"/>
```

```
    </xs:sequence>
  </xs:complexType>
```

10 Security Considerations

The security considerations in [1] apply, as well as [11]. Compared to PIDF, this presence document format reveals additional information that can be highly sensitive. Beyond traditional security measures to protect confidentiality and integrity, systems should offer a means to selectively reveal information to particular watchers and to inspect the information that is being published, particularly if it is generated automatically from other sources, such as calendars or sensors.

11 IANA Considerations

This document calls for IANA to:

- register two new XML namespace URNs per [9];
- establish registry for categories (Section 7.4), place types (Section 7.9), and relationships (Section 7.11).

Note that this document does not need a new content type. It inherits the content type from [1], namely `application/cpim-pidf+xml`.

11.1 URN Sub-Namespace Registration for 'urn:ietf:params:xml:ns:sip-rpids'

URI: urn:ietf:params:xml:ns:sip-rpids

Description: This is the XML namespace for XML elements defined by RFCXXXX to describe a rich presence information extension for the CPIM-PIDF presence document format in the

```
application/cpim-pidf+xml
```

content type.

Registrant Contact: IETF, SIMPLE working group, <simple@ietf.org>,
Henning Schulzrinne, <hgs@cs.columbia.edu>

XML:
```
BEGIN
 <?xml version="1.0"?>
 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
 "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
 <html xmlns="http://www.w3.org/1999/xhtml
 <head>
      <meta http-equiv="content-type"
      content="text/html;charset=iso-8859-1"/>
      <title>RPIDS -- Rich Presence Information Data Format
for Presence Based on the Session
Initiation Protocol (SIP)</title>
```

```
</head>
<body>
    <h1>Namespace for SIMPLE rich presence extension</h1>
    <h2>application/cpim-pidf+xml</h2>
    <p>See <a href="[[[URL of published RFC]]]">RFCXXXX</a>.</p>
 </body>
 </html>
END
```

11.2 URN Sub-Namespace Registration for 'urn:ietf:params:xml:ns:sip-prescaps'

URI: urn:ietf:params:xml:ns:sip-prescaps

Description: This is the XML namespace for XML elements defined by RFCXXXX to describe a capabilities presence information extension for the CPIM-PIDF presence document format in the

```
application/cpim-pidf+xml
```

content type.

Registrant Contact: IETF, SIMPLE working group, <simple@ietf.org>,
Henning Schulzrinne, <hgs@cs.columbia.edu>

XML:

```
 BEGIN
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
"http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml
<head>
     <meta http-equiv="content-type"
     content="text/html;charset=iso-8859-1"/>
     <title>SIMPLE PIDF presence capabilities extension</title>
</head>
<body>
    <h1>Namespace for SIMPLE presence capabilities extension</h1>
    <h2>application/cpim-pidf+xml</h2>
    <p>See <a href="[[[URL of published RFC]]]">RFCXXXX</a>.</p>
 </body>
 </html>
END
```

11.3 Place Type, Device Type, Categories, Relationships

This document creates new IANA registries for categories, device types, place types and relationships. All are XML tokens. Registered tokens must be documented at the time of registration, as most descriptions are expected to be brief.

The SIMPLE working group, or, if no longer available, the SIP working group should be consulted prior to registration.

12 Acknowledgements

The document reflects the discussion on the SIMPLE mailing list, with contributions from many individuals. Markus Isomaki, Hisham Khartabil, Jon Peterson and Brian Rosen provided detailed comments and suggestions.

13 References

Normative References

[1] H. Sugano, S. Fujimoto, *et al.*, "Common presence and instant messaging (cpim)presence information data format," internet draft, Internet Engineering Task Force, Jan. 2003. Work in progress.

[2] H. Schulzrinne and J. Rosenberg, "Session initiation protocol (SIP) caller preferences and callee capabilities," internet draft, Internet Engineering Task Force, Nov. 2002. Work in progress.

[3] M. Day, J. Rosenberg, and H. Sugano, "A model for presence and instant messaging," RFC 2778, Internet Engineering Task Force, Feb. 2000.

[4] M. Handley and V. Jacobson, "SDP: session description protocol," RFC 2327, Internet Engineering Task Force, Apr. 1998.

[5] J. Rosenberg and H. Schulzrinne, "An offer/answer model with session description protocol (SDP)," RFC 3264, Internet Engineering Task Force, June 2002.

[6] S. Bradner, "Key words for use in rfcs to indicate requirement levels," RFC 2119, Internet Engineering Task Force, Mar. 1997.

[7] R. Moats, "URN syntax," RFC 2141, Internet Engineering Task Force, May 1997.

[8] R. Moats, "A URN namespace for IETF documents," RFC 2648, Internet Engineering Task Force, Aug. 1999.

[9] M. Mealling, "The IETF XML registry," internet draft, Internet Engineering Task Force, July 2002. Work in progress.

[10] K. Holtman, A. Mutz, and T. Hardie, "Media feature tag registration procedure," RFC 2506, Internet Engineering Task Force, Mar. 1999.

[11] J. Rosenberg, "A presence event package for the session initiation protocol (SIP)," internet draft, Internet Engineering Task Force, Jan. 2003. Work in progress.

Informative References

[12] F. Dawson and D. Stenerson, "Internet calendaring and scheduling core object specification (icalendar)," RFC 2445, Internet Engineering Task Force, Nov. 1998.

[13] F. D. Jr., S. M. Reddy, D. Royer, and E. Plamondon, "icalendar DTD document (xcal)," internet draft, Internet Engineering Task Force, July 2002. Work in progress.

[14] J. Lennox and H. Schulzrinne, “CPL: a language for user control of Internet telephony services,” internet draft, Internet Engineering Task Force, Nov. 2001. Work in progress.

[15] A. B. Roach, J. Rosenberg, and B. Campbell, “A session initiation protocol (SIP) event notification extension for collections,” internet draft, Internet Engineering Task Force, Feb. 2003. Work in progress.

14 Authors' and Editor's Addresses

The addresses of authors and editors are listed below in alphabetical order.

Vijay Gurbani
Lucent
2000 Naperville Rd., Room 6G-440
Naperville, IL 60566-7033
USA
Email: vkg@lucent.com

Krisztian Kiss
Nokia Research Center
P.O BOX 100
33721 Tampere
Finland
EMail: krisztian.kiss@nokia.com

Paul Kyzivat
Cisco Systems
Mail Stop LWL3/12/2
900 Chelmsford St.
Lowell, MA 01851
USA
Email: pkzivat@cisco.com

Mikko Lonnfors
Nokia Research Center
P.O. Box 407
FIN-00045 NOKIA GROUP
Finland
Email: mikko.lonnfors@nokia.com

Jonathan Rosenberg
dynamicsoft
72 Eagle Rock Avenue
First Floor
East Hanover, NJ 07936
USA
Email: jdrosen@dynamicsoft.com

Henning Schulzrinne
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 10027
USA
Email: schulzrinne@cs.columbia.edu