3GPP TSG-CN Meeting #26 **NP-040503**
8th – 10th December 2004. Athens, Greece.

Source: **TSG CN WG1**

Title: **CRs to Rel-6 WI "PRESNC" for TS 24.141**

Agenda item: **9.2**

Document for: **APPROVAL**

This document contains **12 CRs on Rel-6 Work Item "PRESNC"**, that have been agreed by TSG CN WG1 CN#36 meeting and forwarded to TSG CN Plenary meeting #26 for approval.

TDoc #	Tdoc Title	Spec	CR #	Rev	CAT	C_Version	WI	Rel
N1-041970	Clarifications to Ut	24.141	19	1	F	6.1.0	PRESNC	Rel-6
N1-041696	Alignment between PUA and watcher for draft-ietf-geopriv-pidf-lo-01	24.141	20		F	6.1.0	PRESNC	Rel-6
N1-041772	Introduction of XCAP client and XCAP server	24.141	21		F	6.1.0	PRESNC	Rel-6
N1-041971	Correction XCAP change flow	24.141	22	1	F	6.1.0	PRESNC	Rel-6
N1-041774	Delete Authentication Proxy Requirements	24.141	23		F	6.1.0	PRESNC	Rel-6
N1-041972	Aligning Presence data model with IETF	24.141	24	1	F	6.1.0	PRESNC	Rel-6
N1-042006	IETF reference update (SIP specific parts)	24.141	25	1	F	6.1.0	PRESNC	Rel-6
N1-041993	IETF reference update (XCAP)	24.141	26	1	F	6.1.0	PRESNC	Rel-6
N1-041992	Updates to Partial publication	24.141	27	1	F	6.1.0	PRESNC	Rel-6
N1-041973	Correction to Watcher Information message flow	24.141	28	1	F	6.1.0	PRESNC	Rel-6
N1-041974	Preventing loop in RLS subscriptions	24.141	30	1	C	6.1.0	PRESNC	Rel-6
N1-041880	Filter criteria update	24.141	32		F	6.1.0	PRESNC	Rel-6

3GPP TSG-CN1 Meeting #36 **Tdoc N1-041696**
Seoul, Korea, 15-19 November 2004

CR-Form-v7.1

CHANGE REQUEST

⌘ **24.141** **CR** **020** ⌘ **rev** **-** ⌘ Current version: **6.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	Alignment between PUA and watcher for draft-ietf-geopriv-pidf-lo-01
Source:	⌘	Ericsson
Work item code:	⌘	PRESNC
Date:	⌘	20/10/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	The draft-ietf-geopriv-pidf-lo-01 is not included for the watcher.
Summary of change:	⌘	The text for the watcher is aligned with the PUA with respect to draft-ietf-geopriv-pidf-lo-01. The text in clause 4 (Securing PIDF) of the internet draft was intentionally left out when draft-ietf-geopriv-pidf-lo was introduced for the PUA. Cleanup for the text – 'watcher application' consistently aligned to 'watcher'
Consequences if not approved:	⌘	Incomplete specification.

Clauses affected:	⌘	5.3.2.2, 5.3.2.3, 5.4.2.4, 5.3.3.2, A.3.2.1, A.3.3.1, A.3.3.2, A.3.6.1

		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

5.3.2 Watcher

5.3.2.1 General

A watcher is an entity that is subscribed or requests presence information about a presentity from the PS.

In addition to the procedures specified in subclause 5.3.2, the watcher shall support the procedures specified in 3GPP TS 24.229 [9] appropriate to the functional entity in which the watcher is implemented.

5.3.2.2 Subscription for presence information state changes and notification acceptance

When the watcher ~~application~~ intends to subscribe for presence information state changes of a presentity, it shall generate a SUBSCRIBE request in accordance with RFC 3265 [19] and draft-ietf-simple-presence-10 [27].

The watcher ~~application~~ shall implement the "application/pidf+xml" content type as described in draft-ietf-impp-cpim-pidf-08 [21] together with the PIDF extensions defined in draft-ietf-simple-rpid-03 [26].

The watcher ~~application~~ may implement the PIDF extensions defined in draft-ietf-simple-cipid-01 [32].

The watcher may implement location information according to the format defined in draft-ietf-geopriv-pidf-lo-01 [37].

The watcher ~~application~~ shall implement draft-ietf-simple- prescaps-ext-00 [25]if it wants to make use of SIP user agent capabilities extensions included in the presence document. The extension may be used by the watcher ~~application~~ for interpreting the type of the service described by the presence tuple. The watcher ~~application~~ may include filters in the body of the SUBSCRIBE request in accordance with draft-ietf-simple-filter-format-00 [30] and draft-ietf-simple-event-filter-funct-00 [31].

The watcher ~~application~~ may indicate its support for partial notification using the Accept header field in accordance with draft-ietf-simple-partial-notify-01 [24].

The watcher ~~application~~ shall interpret the received presence information according to the following:

a) a tuple including a <contact-type> element as defined in draft-ietf-simple-rpid-03 [26] with the value "presentity" means general information about the presentity;

b) a tuple including a <relationship> element and <contact-type> element with the value "presentity" as defined in draft-ietf-simple-rpid-03 [26] means information about an alternate contact to the presentity;

c) a tuple including a <contact-type> element as defined in draft-ietf-simple-rpid-03 [26] with the value "service" means communication mean specific information. The communication mean described by the tuple is deduced from the URI scheme of the contact address information present in the <contact> element as defined in draft-ietf-impp-cpim-pidf-08 [21]. If the URI scheme of the contact address information provides ambiguous information about the communication means, the watcher ~~application~~ shall further examine other elements of the tuple to decide the communication mean. Such elements can be the <methods> element, any of the different media type specific elements as defined in draft-ietf-simple-prescaps-ext-00 [25], or the <relationship> element as defined in draft-ietf-simple-rpid-03 [26].

Additional extensions can be used to express application specific attributes, but their usage is outside the scope of this version of the specification.

5.3.2.3 Subscription for presence information state changes of presentity collections

When the watcher ~~application~~ intends to subscribe for presence information state changes of a presentity collection, it shall generate a SUBSCRIBE request in accordance with draft-ietf-simple-event-list-04 [22], additionally to the procedures described in subclause 5.3.2.2.

5.3.2.4 Subscription for the watcher information event template package

Upon activation of the presence service, the watcher ~~application~~ may subscribe recursively for the watcher information state changes in accordance with draft-ietf-simple-winfo-package-05 [28] and draft-ietf-simple-winfo-format-04 [29].

The watcher ~~application~~ may include filters in the body of the SUBSCRIBE request in accordance with draft-ietf-simple-filter-format-00 [30] and draft-ietf-simple-event-filter-funct-00 [31].

5.3.2.5 Subscription for xcap-change

In order to get notifications of changes to XML documents manipulated via the Ut reference point the watcher may generate a SUBSCRIBE request in accordance with draft-ietf-simple-xcap-package-02 [39] and draft-ietf-sipping-config-framework-04 [43].

5.3.3 Presence Server (PS)

5.3.3.1 General

A PS is an entity that accepts, stores, and distributes presence information.

In addition to the procedures specified in subclause 5.3.3, the PS shall support the procedures specified in 3GPP TS 24.229 [9] appropriate to the functional entity in which the PS is implemented.

5.3.3.2 Subscription acceptance to presence information and notification of state changes

When the PS receives a SUBSCRIBE request for the presence information event package, the PS shall first attempt to verify the identity of the source of the SUBSCRIBE request as described in 3GPP TS 24.229 [9] subclause 5.7.1.4, then perform authorization according to 3GPP TS 24.229 [9] subclause 5.7.1.5. In case of successful subscription, the PS shall generate a response to the SUBSCRIBE request and notifications in accordance with RFC 3265 [19] and draft-ietf-simple-presence-10 [27].

Additionally, in the special case of a watcher subscription if the subscription authorization policy results in the action to confirm the watcher subscription from the PUA and the PUA has a valid watcher information subscription, see draft-ietf-simple-winfo-package-05 [28], then, the PS shall inform the PUA about the watcher subscription attempt.

If the watcher ~~application~~ has indicated the need for partial notification using the Accept header field, then the PS shall generate partial notifications in accordance with draft-ietf-simple-partial-notify-01 [24] and draft-ietf-simple-partial-pidf-format-00 [38].

If the body of the SUBSCRIBE request from the watcher contains filters, the PS shall apply the requested filtering function on notifications in accordance with draft-ietf-simple-filter-format-00 [30] and draft-ietf-simple-event-filter-funct-00 [31].

If the watcher ~~application~~ has indicated support for the "multipart/related" content type using the Accept header field, then the PS may generate notifications using "multipart/related" content type which aggregates "application/pidf+xml" formatted presence information with other MIME objects in accordance with RFC 2387 [14]. In this case, the PS shall modify the value of the presence attribute in the PIDF document to refer to the MIME object included in the corresponding MIME multipart body. If the watcher ~~application~~ has not indicated support for the "multipart/related" or a MIME object cannot be accessed by the PS, the PS should exclude the presence attribute from the notification.

*** Next Change ***

A.3.2.1 Successful subscription

< First part of subclause omitted >

17. **NOTIFY request (P-CSCF to UE) - see example in table A.3.2.1-17**

The P-CSCF forwards the NOTIFY request to the watcher ~~application~~ in the UE.

< Remaining text of subclause omitted >

*** Next Change ***

A.3.3.1 Watcher subscribing to his own resource list, UE in visited network - Successful subscription

< First part of subclause omitted >

11. **NOTIFY request (P-CSCF to UE) – see example in table A.3.3.1-11**

The P-CSCF forwards the NOTIFY request to the watcher ~~application~~ in the UE.

< Part of subclause omitted >

18. **NOTIFY request (P-CSCF to UE) - see example in table A.3.3.1-18**

The P-CSCF forwards the NOTIFY request to the watcher ~~application~~ in the UE.

< Remaining text of subclause omitted >

*** Next Change ***

A.3.3.2 Watcher subscribing to a resource list, UE in visited network - successful subscription

< First part of subclause omitted >

14. **NOTIFY request (P-CSCF to UE) - see example in table A.3.3.2-14**

The P-CSCF forwards the NOTIFY request to the watcher ~~application~~ in the UE.

< Part of subclause omitted >

21. **NOTIFY request (P-CSCF to UE) - see example in table A.3.3.2-21**

The P-CSCF forwards the NOTIFY request to the watcher ~~application~~ in the UE.

< Remaining part of subclause omitted>

*** Next Change ***

A.3.6.1 Watcher subscribing to XCAP change in his resource list, UE in visited network - Successful subscription

< Part of subclause omitted>

11. **NOTIFY request (P-CSCF to UE) - see example in table A.3.6.1-11**

The P-CSCF forwards the NOTIFY request to the watcher ~~application~~ in the UE.

< Part of subclause omitted >

18. **NOTIFY request (P-CSCF to UE) – see example in table A.3.6.1-18**

The P-CSCF forwards the NOTIFY request to the watcher ~~application~~ in the UE.

< Remaining part of subclause omitted>

3GPP TSG-CN1 Meeting #36
Seoul, Korea, 15-19 November 2004

Tdoc N1-041772

CR-Form-v7.1

CHANGE REQUEST

⌘ **24.141** **CR** **21** ⌘ **rev** **-** ⌘ Current version: **6.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	Introduction of XCAP client and XCAP server
Source:	⌘	Siemens
Work item code:	⌘	PRESNC
Date:	⌘	16/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	24.141 defines the roles Data Manipulator DM and Data Manipulation Server. However, in the XCAP drafts the terms “XCAP client” and “XCAP server” are defined and used for that purpose. As already discussed in the last meeting, the terms DM and DMS shall be replaced by XCAP client and XCAP server
Summary of change:	⌘	See reason for change
Consequences if not approved:	⌘	24.141 defines roles that are already defined in IETF, ambiguity

Clauses affected:	⌘	3.2, 6.2.1, 6.2.2, 6.3.1, 6.3.2, A.8.2, A.8.3, A.8.4

Other specs affected: ⌘	Y	N	
		X	Other core specifications
		X	Test specifications
		X	O&M Specifications

⌘

Other comments: ⌘

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

*** 1st change ***

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AS	Application Server
AUID	Application Usage ID
CN	Core Network
CPIM	Common Profile for Instant Messaging
CSCF	Call Session Control Function
~~DM~~	~~Data Manipulator~~
~~DMS~~	~~Data Manipulation Server~~
EPA	Event Publication Agent
ESC	Event State Compositor
HSS	Home Subscriber Server
HTTP	HyperText Transfer Protocol
I-CSCF	Interrogating - CSCF
IM	IP Multimedia
IOI	Inter Operator Identifier
IP	Internet Protocol
MIME	Multipurpose Internet Mail Extensions
P-CSCF	Proxy - CSCF
PIDF	Presence Information Data Format
PNA	Presence Network Agent
PS	Presence Server
PSI	Public Service Identity
PUA	Presence User Agent
RLMI	Resource List Meta-Information
RLS	Resource List Server
RPID	Rich Presence Information Data
S-CSCF	Serving - CSCF
SIP	Session Initiation Protocol
TLS	Transport Layer Security
UE	User Equipment
URI	Universal Resource Identifier
XCAP	XML Configuration Access Protocol
XML	Extensible Markup Language

*** next change ***

6 Protocol for data manipulation at the Ut reference point

6.1 Introduction

Hypertext Transfer Protocol (HTTP) and XML Configuration Access Protocol (XCAP) are used to store, alter and delete data related to the presence service. The general information that can be manipulated is user groups, subscription authorization policy, resource lists, hard state presence publication, MIME objects referenced from the hard state presence information, etc. Soft state presence information manipulated with a PUBLISH request is not manipulated by the mechanism provided over the Ut reference point.

6.2 Functional entities

6.2.1 User Equipment (UE)

The UE implements the ~~Data Manipulator (DM)~~XCAP client role as described in subclause 6.3.1.

The UE shall implement HTTP digest AKA (see RFC 3310 [20]) and it shall initiate a bootstrapping procedure with the bootstrapping server function located in the home network, as described in 3GPP TS 24.109 [7].

The UE shall acquire the subscriber's certificate from PKI portal by using a bootstrapping procedure, as described in 3GPP TS 24.109 [7].

The UE shall implement HTTP digest authentication (see RFC 2617 [15A]).

The UE shall implement Transport Layer Security (TLS) (see RFC 2246 [13]). The UE shall be able to authenticate the network application function based on the received certificate during TLS handshaking phase.

6.2.2 Application Server (AS)

If an AS implements the role of a PS (see subclause 5.3.3) or of a RLS (see subclause 5.3.4), then the AS shall also implement the role of a ~~Data Manipulation Server (DMS)~~XCAP server (see subclause 6.3.2).

If there is no authentication proxy in the network, then the AS shall:

1) implement the role of a network application function, as described in 3GPP TS 24.109 [7];

2) implement TLS (see RFC 2246 [13]);

implement HTTP digest authentication (see RFC 2617 [15A]); and

4) support certificate authentication.

Editor's note: It needs to be clarified what physical entities can contain the Authentication Proxy and its relationship with the IMS architecture.

6.2.3 Authentication proxy

The authentication proxy shall implement the role of a network application function, as described in 3GPP TS 24.109 [7] and it shall support HTTP Digest Authentication (see RFC 2617 [15A] and certificate authentication.

The Authentication Proxy shall authenticate the UE and integrity protect the messages sent towards the UE.

Editor's note: It is FFS how the Authentication Proxy passes the user's identity to the Application Server (AS).

6.3 Roles

6.3.1 ~~Data Manipulator (DM)~~XCAP client

6.3.1.1 Introduction

The ~~DM~~ XCAP client is a logical function ~~that implements the requirements of a XCAP client~~ as defined in draft-ietf-simple-xcap-03 [33]. The ~~DM~~ XCAP client provides the means to manipulate the general data such us user groups, subscription authorization policy, resource lists, hard state presence publication, MIME objects referenced from the hard state presence information, etc.

NOTE: In order to be able to manipulate data stored on the ~~DMS~~XCAP server, the ~~DM~~ XCAP client has the root directory on the ~~DMS~~XCAP server pre-configured or use some means to discover it. Discovery mechanisms are outside the scope of the present document.

6.3.1.2 Manipulating a presencelist

When the DM intends to manipulate a presencelist, it shall generate an HTTP PUT, GET or DELETE request in accordance with RFC 2616 [15], draft-ietf-simple-xcap-03 [33] and draft-ietf-simple-xcap-list-usage-02 [36].

6.3.1.3 Manipulating the subscription authorization policy

When the ~~DM~~ XCAP client intends to manipulate the subscription authorization policy, it shall generate an HTTP PUT, HTTP GET or HTTP DELETE request in accordance with RFC 2616 [15], draft-ietf-simple-xcap-03 [33] and draft-ietf-simple-xcap-presence-rules-00 [35].

The ~~DM~~ XCAP client may use an HTTP GET in accordance with RFC 2616 [15], draft-ietf-simple-xcap-03 [33] and draft-rosenberg-simple-common-policy-caps-01 [42] for fetching of the authorization policy capabilities which the ~~DMS~~XCAP server supports.

When the ~~DM~~ XCAP client intends to authorize a different value of the same presence attribute to different watchers or watcher groups, the ~~DM~~ XCAP client shall authorize a single tuple including one of the different values of the same presence attribute to every watcher or watcher groups by using a specific "inclusion set" as specified in draft-ietf-simple-xcap-presence-rules-00 [35].

6.3.1.4 Publishing hard state presence information

The ~~DM~~ XCAP client shall implement draft-isomaki-simple-xcap-pidf-manipulation-usage-00 [34] in order to be able to manipulate hard state presence information. Hard state presence information uses the same format as soft state information, namely "application/pidf+xml" content type as described in draft-ietf-impp-cpim-pidf-08 [21] together with any of its extensions.

When the hard state presence information contains one ore more MIME objects to be aggregated with the "application/pidf+xml" content type and any of its extensions, the ~~DM~~ XCAP client shall:

a) construct as many HTTP URIs as many objects to be stored and formulate every HTTP URI according a predefined directory structure;

NOTE: In order to be able to manipulate data stored on the ~~DMS~~XCAP server, the ~~DM~~ XCAP client has the root directory on the ~~DMS~~XCAP server pre-configured or use some means to discover it. Discovery mechanisms are outside the scope of the present document.

b) store the objects on the data manipulation server behind the HTTP URI(s) created in the previous step using standard HTTP procedures as defined in RFC 2616 [15];

c) include every HTTP URI as a value of the corresponding XML element in the published "application/pidf+xml" presence document referencing the stored object(s) in the previous step; and

d) publish the hard state presence information according to draft-isomaki-simple-xcap-pidf-manipulation-usage-00 [34].

6.3.2 ~~Data Manipulation Server (DMS)~~XCAP server

6.3.2.1 Introduction

The ~~Data Manipulation Server (DMS~~XCAP server~~)~~XCAP server is a logical function that ~~implements the requirements of a XCAP server~~ as defined in draft-ietf-simple-xcap-03 [33]. The ~~DMS~~ XCAP server can store data such us user groups, subscription authorization policy, resource lists, hard state presence information, MIME objects referenced from the hard state presence information, etc.

6.3.2.2 Resource list manipulation acceptance

When the data manipulation server receives an HTTP PUT, HTTP GET or HTTP DELETE request for manipulating or fetching a resource list, the ~~DMS~~ XCAP server shall first authenticate the request in accordance with 3GPP TS 24.109 [7] and then perform authorization. Afterwards the ~~DMS~~ XCAP server shall perform the requested action and generate a response in accordance with RFC 2616 [15], draft-ietf-simple-xcap-03 [33] and draft-ietf-simple-xcap-list-usage-02 [36].

6.3.2.3 Subscription authorization policy manipulation acceptance

When the ~~DMS~~XCAP server receives an HTTP PUT, HTTP GET or HTTP DELETE request for manipulating or fetching of the subscription authorization policy, the data manipulation server shall first authenticate the request in accordance with 3GPP TS 24.109 [7] and then perform authorization. Afterwards the ~~DMS~~ XCAP server shall perform the requested action and generate a response in accordance with RFC 2616 [15], draft-ietf-simple-xcap-03 [33] and draft-ietf-simple-xcap-presence-rules-00 [35].

When the ~~DMS~~XCAP server receives an HTTP GET request for fetching of the authorization policy capabilities information, the ~~DMS~~XCAP server shall generate a response in accordance with RFC 2616 [15], draft-ietf-simple-xcap-02 [33] and draft-rosenberg-simple-pres-policy-caps-00 [42].

6.3.2.4 Publication acceptance of hard state presence information

When the ~~DMS~~XCAP server receives an HTTP PUT, HTTP GET or HTTP DELETE request for publishing, fetching or deleting of hard state presence information, the ~~DMS~~XCAP server shall first authenticate the request in accordance with 3GPP TS 24.109 [7] and then perform authorization. Afterwards the ~~DMS~~XCAP server shall:

a) if the HTTP URI points to a predefined directory reserved for storing MIME objects and the request is an HTTP PUT request, replace any existing content referenced by the Request-URI with the content of the request;

b) if the Request-URI points to an uncreated directory, create the directory, store the content there and associate the content with the Request-URI. For all requests, i.e. HTTP PUT, HTTP GET and HTTP DELETE requests, generate an appropriate response in accordance with RFC 2616 [15]; or

c) if the HTTP URI points to an XCAP directory and the Application Usage ID (AUID) part of the HTTP URI is set to "pidf-manipulation", process the request and generate an appropriate response in accordance with draft-ietf-simple-xcap-03 [33], draft-isomaki-simple-xcap-pidf-manipulation-usage-00 [34] and RFC 2616 [15].

*** Next change ***

A.8 Example signalling flows of HTTP based presence service operation

A.8.1 Introduction

This subclause shows signalling flows relating to the manipulation of presence service data over the Ut reference point using XCAP.

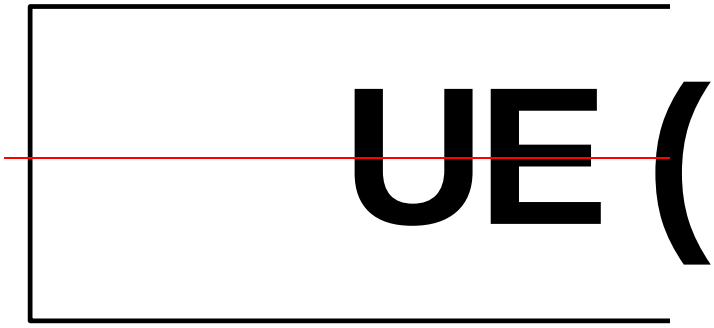
Each example signalling flow shows several sequences of manipulation of data for the presence service.

NOTE: Error conditions are not considered in the examples e.g. if authorization checks fail in the XCAP server, XML Schema compliancy checks fail or the file specified by the URI does not exist then an appropriate 4xx response is sent to the client.

Editor's note: Clarifications how XCAP is using HTTP is needed.

A.8.2 Signalling flows demonstrating how ~~DMs~~XCAP server manipulate resource lists

Editor's note: The possible proxies (e.g., handling authentication matters) between the data manipulator client and HTTP server are bypassed. Also the authentication related headers are missing.



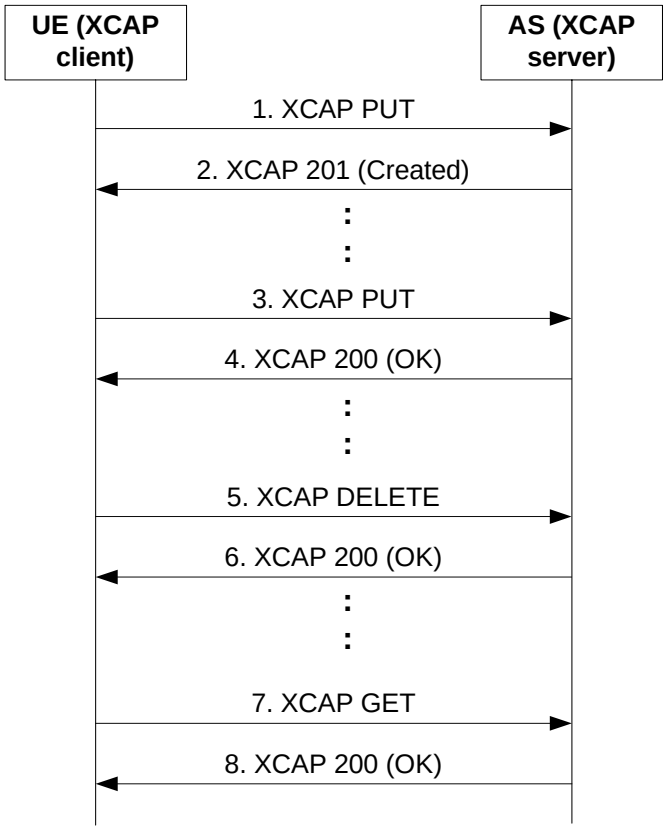


Figure A.8.2-1: ~~DM~~ XCAP client manipulating a resource list on ~~DMS~~XCAP server

Figure A.8.2-1 shows a how a ~~DM~~ XCAP client may manipulate a resource list on a ~~DMS~~XCAP server. The details of the signalling flows are as follows:

1. **XCAP PUT request (DM to ~~DMS~~XCAP server - see example in table A.8.2-1**

   The ~~DM~~ XCAP client generates an XCAP PUT request to create a new resource list on the ~~DMS~~XCAP server. The resource list has one entry.

Table A.8.2-1: XCAP PUT request (~~DM~~ XCAP client to ~~DMS~~XCAP server)

```
PUT http://xcap.home1.net/services/resource-lists/users/user1/pf.xml HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 10:13:17 GMT
Content-Type: application/resource-lists+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
   <resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists">
     <list name="Presence_fellows" uri="sip:user1_list1@home1.net" subscribeable="true">
       <entry name="user2" uri="sip:user2_public1@home2.net">
         <display-name>User2</display-name>
       </entry>
     </list>
   </resource-lists>
```

2. **XCAP 201 (Created) response (~~DMS~~XCAP server to ~~DM~~XCAP client) – see example in table A.8.2-2**

   After the ~~DMS~~XCAP server has performed the necessary authorization checks on the originator to ensure the ~~DM~~ XCAP client is allowed to create a file, the ~~DMS~~XCAP server sends an XCAP 201 (Created) response to the ~~DM~~XCAP client.

Table A.8.2-2: XCAP 201 (Created) response (~~DMS~~XCAP server to ~~DM~~XCAP client)

```
HTTP/1.1 201 CREATED
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 10:50:35 GMT
Content-Type: text/html
Content-Length: 0
```

3. **XCAP PUT request (~~DM~~ XCAP client to ~~DMS~~XCAP server) – see example in table A.8.2-3**

   The ~~DM~~ XCAPclient adds a new entry to the previously created resource list by generating a new XCAP PUT request.

Table A.8.2-3: XCAP PUT request (~~DM~~ XCAP client to ~~DMS~~XCAP server)

```
PUT http://xcap.home1.net/services/resource-lists/users/user1/pf.xml?resource-
lists/list[@name="Presence_fellows"]/entry HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 10:13:17 GMT
Content-Type: application/xml-fragment-body
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
         <entry name="user3" uri="sip:user3_public1@home3.net">
           <display-name>User3</display-name>
         </entry>
```

4. **XCAP 200 (OK) response (~~DMS~~XCAP server to ~~DM~~XCAP client) - see example in table A.8.2-4**

   After the ~~DMS~~XCAP server XCAP server has performed the necessary authorization checks, XML document validations and XML schema compliancy checks the ~~DMS~~XCAP server sends an XCAP 201 (Created) response to the ~~DM~~XCAP client.

Table A.8.2-4: XCAP 200 (OK) response (~~DMS~~XCAP server to ~~DM~~XCAP client)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 10:50:45 GMT
Content-Type: text/html
Content-Length: 0
```

5. **XCAP DELETE request (~~DM~~ XCAP client to ~~DMS~~XCAP server) - see example in table A.8.2-5**

The ~~DM~~ XCAP client decides to delete the entry "user2" from the resource list. The ~~DM~~ XCAP client generates an XCAP DELETE request.

Table A.8.2-5: XCAP DELETE request (~~DM~~ XCAP client to ~~DMS~~XCAP server)

```
DELETE http://xcap.home1.net/services/resource-lists/users/user1/pf.xml?resource-
lists/list[@name="Presence_fellows"]/entry[@name=user2"] HTTP/1.1
Host: oper.example.com:9999
User-Agent: IMS subscriber
Date: Thu, 08 Jan 2004 10:14:17 GMT
Referer: http://oper.home1.net:1234/service
```

6. **XCAP 200 (OK) response (~~DMS~~XCAP server to ~~DM~~XCAP client) – see example in table A.8.2-6**

After the ~~DMS~~XCAP server has performed the necessary authorization checks on the originator to ensure the DM is allowed to delete an entry from the resource list, the ~~DMS~~XCAP server sends an XCAP 200 (OK) response.

Table A.8.2-6: XCAP 200 (OK) response (~~DMS~~XCAP server to ~~DM~~XCAP client)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 11:00:47 GMT
Content-Type: image/jpeg
Content-Length: 0
```

7. **XCAP GET request (~~DM~~XCAP client to ~~DMS~~XCAP server) – see example in table A.8.2-7**

The ~~DM~~XCAP client wishes to check the result of the previous transaction by generating an XCAP GET request.

Table A.8.2-7: XCAP GET request (DM to ~~DM~~XCAP client~~S~~XCAP server)

```
GET http://xcap.home1.net/services/resource-lists/users/user1/pf.xml HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 11:13:17 GMT
Content-Length: 0
```

8. **XCAP 200 (OK) response (~~DMS~~XCAP server to ~~DM~~XCAP client) - see example in table A.8.2-8**

After the ~~DMS~~XCAP server has performed the necessary authorization checks on the originator to ensure the ~~DM~~XCAP client is allowed to fetch the resource list, the ~~DMS~~XCAP server sends an XCAP 200 (OK) response to the ~~DM~~XCAP client including the resource list in the body of the response.

Table A.8.2-8: XCAP 200 (OK) response (~~DM~~XCAP client~~S~~XCAP server to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Etag: "askdajdsaj"
Date: Thu, 08 Jan 2004 11:50:35 GMT
Content-Type:application/resource-lists+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
```

```
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists">
  <list name="Presence_fellows" uri="sip:user1_list1@home1.net" subscribeable="true">
    <entry name="user3" uri="sip:user3_public1@home3.net">
      <display-name>User3</display-name>
    </entry>
  </list>
</resource-lists>
```

A.8.3 Signalling flows demonstrating how ~~DM~~XCAP client~~s~~ manipulate presence authorization policy

Editor's note: The possible proxies (e.g., handling authentication matters) between the data manipulator client and HTTP server are bypassed. Also the authentication related headers are missing.

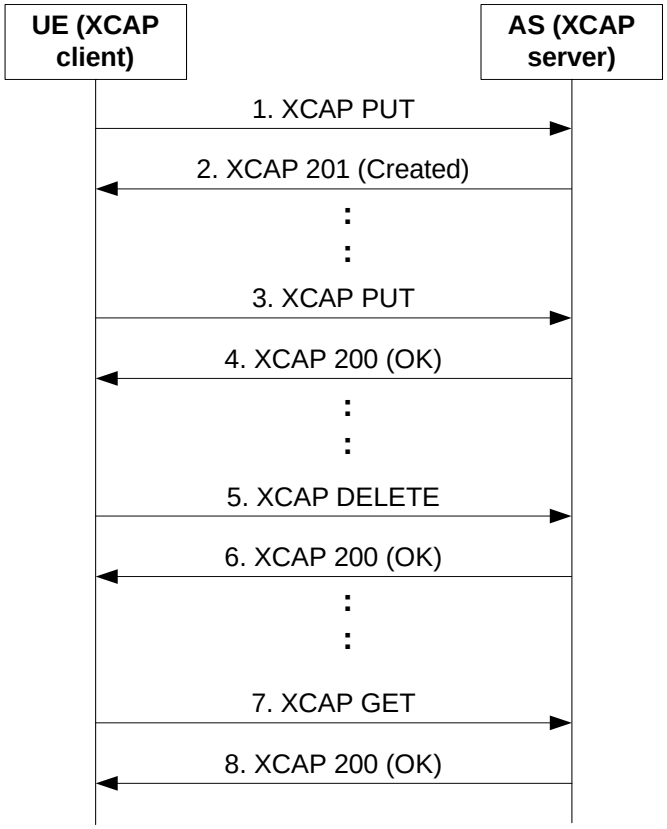


Figure A.8.3-1: ~~DM~~XCAP client manipulating presence authorization policy on ~~DMS~~XCAP server

Figure A.8.3-1 shows a ~~DM~~XCAP client manipulating presence authorization policy on a ~~DMS~~XCAP server. The details of the signalling flows are as follows:

1. **XCAP PUT request (DM to ~~DM~~XCAP client~~S~~XCAP server) – see example in table A.8.3-1**

   The DM generates an XCAP PUT request to create a presence authorization policy on the ~~DM~~XCAP client~~S~~XCAP server. The presence authorization policy has one permission statement allowing for sip:user2_public1@home2.net to see all information from the basic PIDF namespace along with the "video" element from the prescaps namespace.

Table A.8.3-1: XCAP PUT request (DM to ~~DM~~XCAP client~~S~~XCAP server)

```
PUT http://xcap.home1.net/services/permission-statements/users/user1/ps.xml HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 10:13:17 GMT
Content-Type:application/permission-statements+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
   <permission-statements xmlns="urn:ietf:params:xml:ns:permission-statements"
     xmlns:pidf="urn:ietf:params:xml:ns:pidf"
     xmlns:prescaps="urn:ietf:params:xml:ns:simple-prescaps-ext">
     <statement id="dsafa43232">
       <applies-to>
         <uri>sip:user2_public1@home2.net</uri>
       </applies-to>

       <permissions>
         <accept/>
         <show-namespace>urn:ietf:params:xml:ns:pidf</show-namespace>
         <show-element>prescaps:video</show-element>
       </permissions>

     </statement>
   </permission-statements>
```

2. **XCAP 201 (Created) response (~~DMS~~XCAP server to ~~DM~~XCAP client) - see example in table A.8.3-2**

After the ~~DMS~~XCAP server has performed the necessary authorization checks on the originator to ensure the ~~DM~~XCAP client is allowed to create a file, the ~~DMS~~XCAP server sends an XCAP 201 (Created) response to the DM.

Table A.8.3-2: XCAP 201 (Created) response (~~DMS~~XCAP server to ~~DM~~XCAP client)

```
HTTP/1.1 201 CREATED
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 10:50:35 GMT
Content-Type: text/html
Content-Length: 0
```

3. **XCAP PUT request (~~DM~~XCAP client to ~~DMS~~XCAP server) – see example in table A.8.3-3**

The ~~DM~~XCAP client adds a new permission-statement to the previously created presence authorization policy by generating a new XCAP request. The new permission statement allows the user named sip:user3_public1@home3.net to see the tuple with class element specifying "sip".

Table A.8.3-3: XCAP PUT request (DM to ~~DM~~XCAP client~~S~~XCAP server)

```
PUT http://xcap.home1.net/services/permission-statements/users/user1/ps.xml/permission-
      statements HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 10:13:27 GMT
Content-Type: application/xml-fragment-body
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
    <statement id="dsffdsfrrr32423">
       <applies-to>
         <uri>sip:user3_public1@home3.net</uri>
       </applies-to>

       <permissions>
         <accept/>
         <show-tuple>sip</show-tuple>
       </permissions>

     </statement>
```

4. **XCAP 200 (OK) response (~~DMS~~XCAP server to ~~DM~~XCAP client) - see example in table A.8.3-4**

After the ~~DMS~~XCAP server has performed the necessary authorization checks, XML document validations and XML schema compliancy checks the ~~DMS~~XCAP server sends an XCAP 201 (Created) response to the ~~DM~~XCAP client.

Table A.8.3-4: XCAP 200 (OK) response (~~DMS~~XCAP server to ~~DM~~XCAP client)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 10:50:45 GMT
Content-Type: text/html
Content-Length: 0
```

5. **XCAP DELETE request (DM to ~~DMS~~XCAP server) - see example in table A.8.3-5**

The ~~DM~~ XCAP clientdecides to delete the permission-statement for sip:user2_public1@home2.net from the authorization policy. The DM generates an XCAP DELETE request.

Table A.8.3-5: XCAP DELETE request (~~DM~~ XCAP clientto ~~DMS~~XCAP server)

```
DELETE http://xcap.home1.net/services/presence-lists/users/user1/ps.xml/permission-
      statements/statement[@id="dsafa43232"]/permissions/show-namespace HTTP/1.1
Host: oper.example.com:9999
```

```
User-Agent: IMS subscriber
Date: Thu, 08 Jan 2004 10:14:17 GMT
Referer: http://oper.home1.net:1234/service
```

6. **XCAP 200 (OK) response (~~DMS~~XCAP server to DM) – see example in table A.8.3-6**

After the ~~DMS~~XCAP server has performed the necessary authorization checks on the originator to ensure the DM is allowed to delete an entry from the resource list, the ~~DMS~~XCAP server sends an XCAP 200 (OK) response.

Table A.8.3-6: XCAP 200 (OK) response (~~DMS~~XCAP server to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 11:00:47 GMT
Content-Length: 0
```

7. **XCAP GET request (~~DM~~ XCAP clientto ~~DMS~~XCAP server) – see example in table A.8.3-7**

The DM wishes to check the result of the previous transaction by generating an XCAP GET request.

Table A.8.3-7: XCAP GET request (~~DM~~ XCAP clientto ~~DMS~~XCAP server)

```
GET http://xcap.home1.net/services/permission-statements/users/user1/ps.xml HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 11:13:17 GMT
Content-Length: 0
```

8. **XCAP 200 (OK) response (~~DMS~~XCAP server to DM) – see example in table A.8.3-8**

After the ~~DMS~~XCAP server has performed the necessary authorization checks on the originator to ensure the DM is allowed to fetch the resource list, the ~~DMS~~XCAP server sends an XCAP 200 (OK) response to the ~~DM~~ XCAP clientincluding the resource list in the body of the response.

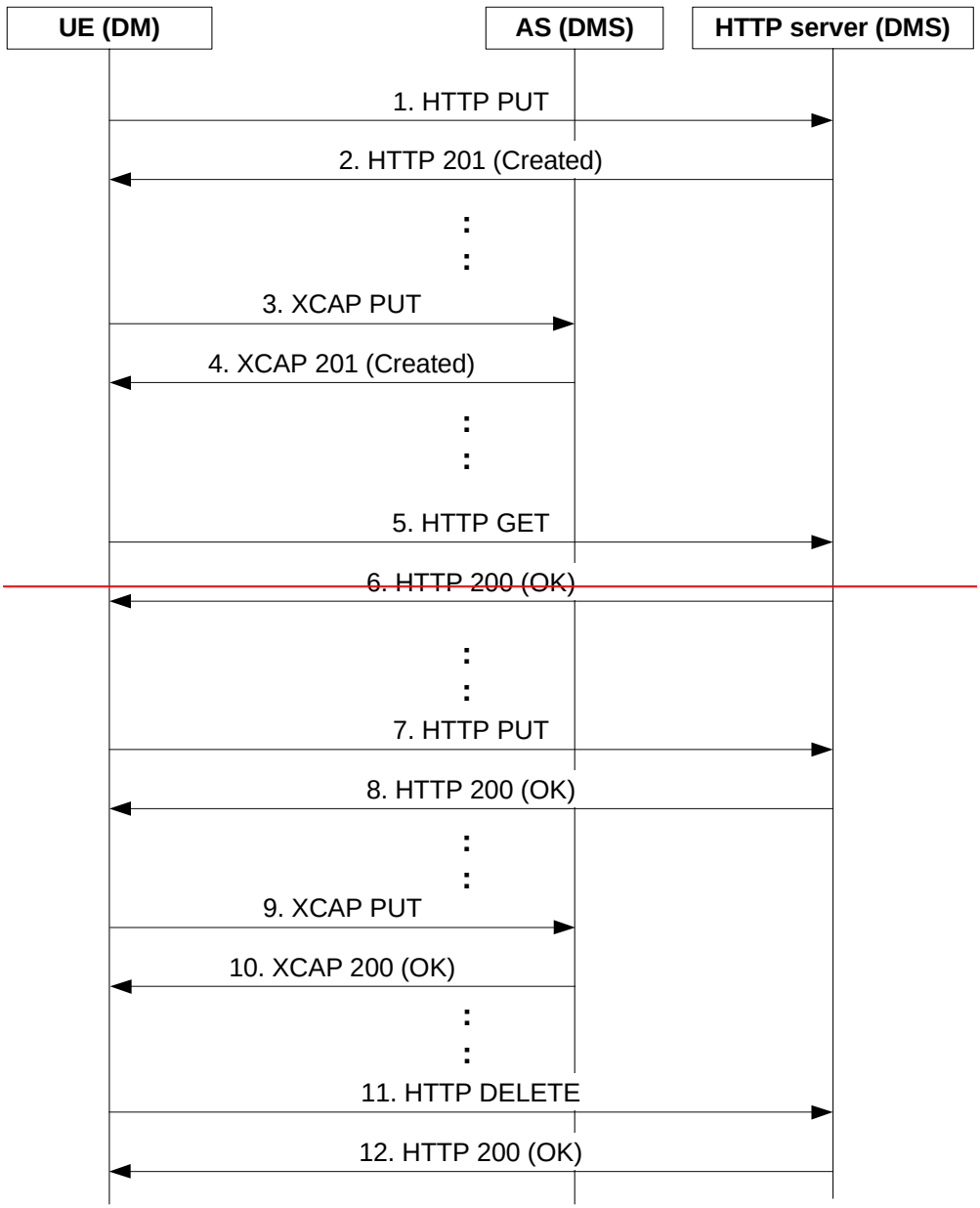
Table A.8.3-8: XCAP 200 (OK) response (~~DMS~~XCAP server to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Etag: "eiuuekksks"
Date: Thu, 08 Jan 2004 11:50:35 GMT
Content-Type:application/permission-statements+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
   <permission-statements xmlns="urn:ietf:params:xml:ns:permission-statements"
     xmlns:pidf="urn:ietf:params:xml:ns:pidf"
     xmlns:prescaps=" urn:ietf:params:xml:ns:simple-prescaps-ext">
     <statement id="dsffdsfrrr32423">
       <applies-to>
         <uri>sip:user3_public1@home3.net</uri>
       </applies-to>
       <permissions>
         <accept/>
         <show-tuple>sip</show-tuple>
       </permissions>
     </statement>
   </permission-statements>
```

A.8.4 Storing external content (successful operation)

Editor's note: The possible proxies (e.g., handling authentication matters) between the data manipulator client and HTTP server are bypassed. Also the authentication related headers are missing.



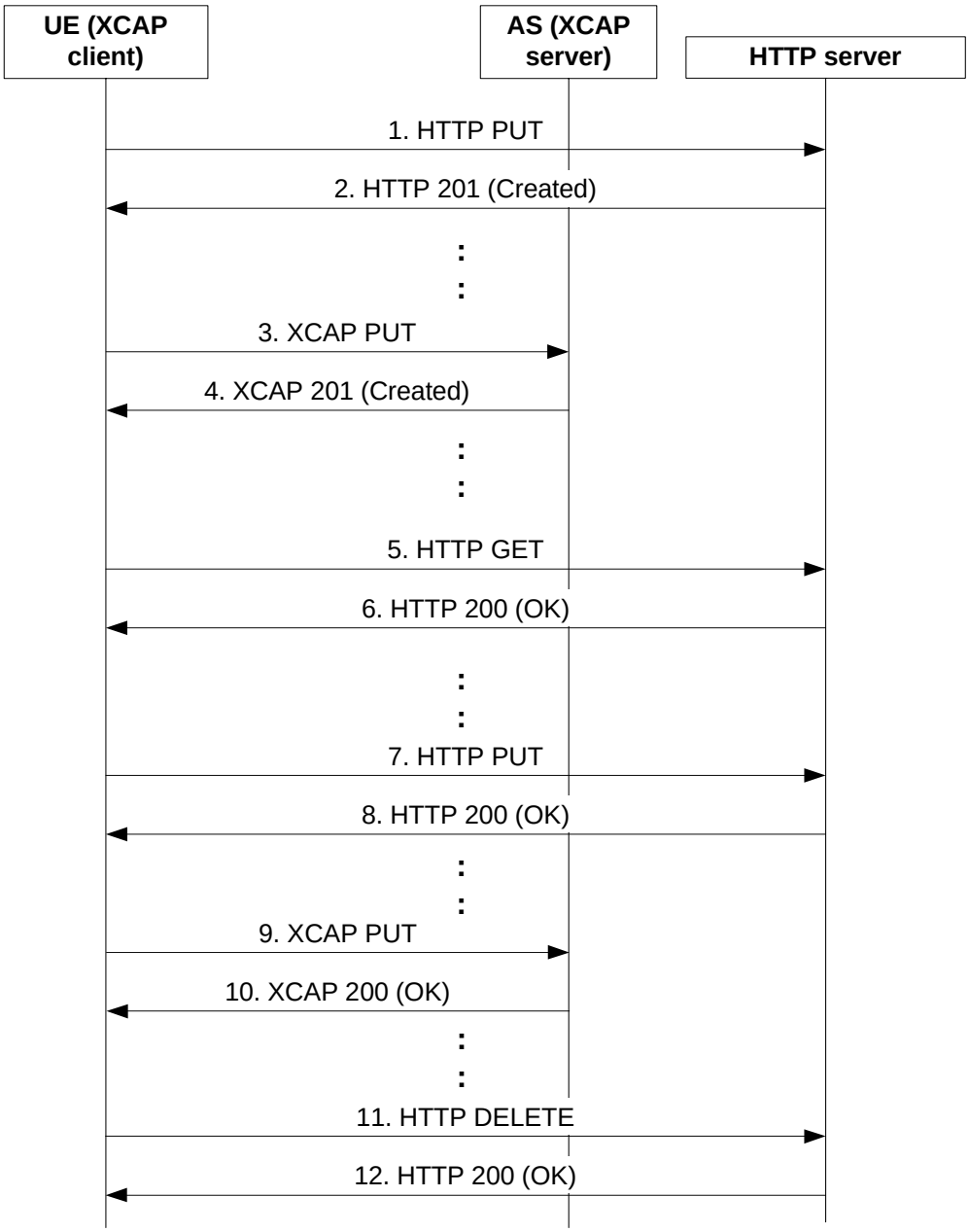


Figure A.8.4.-1: ~~DM~~ XCAP client manipulating hard-state presence document on ~~DMS~~XCAP server

Figure A.8.4-1 shows a DM manipulating hard-state presence document on a ~~DMS~~XCAP server when the presence document has an aggregated storing MIME object with the "application/pidf+xml" content type and any of its extensions. The details of the signalling flows are as follows:

1. **HTTP PUT request (~~DM~~ XCAP client(client) to ~~DMS~~XCAP server) – see example in table A.8.2-1**

   In order to store the content, the DM generates an HTTP PUT request containing the MIME object in the body of the request. The request-URI points to the directory where the content is stored and shows the name of the file to be created.

Table A.8.4-1: HTTP PUT request (~~DM~~ XCAP clientto ~~DMS~~XCAP server)

```
PUT http://operator.example.com/services/users/bill/pictureX HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 10:13:17 GMT
Content-Type: image/jpeg
Content-Length: (…)

{pictureX.jpg}
```

2. **HTTP 201 (Created) response (~~DMS~~XCAP server to DM) – see example in table A.8.4-2**

   After the ~~DMS~~XCAP server has performed the necessary authorization checks on the originator to ensure the ~~DM~~ XCAP client is allowed to create a file the HTTP server sends an HTTP 201 (Created) response to the client.

Table A.8.4-2: HTTP 201 (Created) response (~~DMS~~XCAP server to DM)

```
HTTP/1.1 201 CREATED
Server: Apache/1.3.22 (Unix) mod_perl/1.27Content-Type: text/html
Date: Thu, 08 Jan 2004 10:50:35 GMT
Content-Length: 1234
```

3. **XCAP PUT request (~~DM~~ XCAP clientto ~~DMS~~XCAP server) - see example in table A.8.2-3**

   The ~~DM~~ XCAP clientgenerates an XCAP PUT request in order to store XML encoded presence document which includes a URI reference to the MIME object stored on the ~~DMS~~XCAP server. The AUID part of the request URI is 'pidf-manipulation' as defined in draft-isomaki-simple-xcap-pidf-manipulation-usage-00 [34].

Table A.8.4-3: XCAP PUT request (~~DM~~ XCAP clientto ~~DMS~~XCAP server)

```
PUT http://xcap.example.com/services/pidf-manipulation/users/bill/pidf.xml HTTP/1.1
User-Agent: IMS subscriber
Referer: http://xcap.home1.net:1234/service
Date: Thu, 08 Jan 2004 10:13:27 GMT
Content-Type: application/pidf+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
   <presence xmlns="urn:ietf:params:xml:ns:cpim-pidf"
          xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
          xmlns:ext="urn:ietf:params:xml:ns:ext-cont"
          entity="sip:bill@example.com">

     <tuple id="123sd">
       <status>
         <basic>open</basic>
       </status>
       <et:type>service</et:type>
         <contact>sip:bill@example.com</contact>
     </tuple>

     <tuple id="432sd">
       <status>
         <basic>open</basic>
       </status>
       <et:type>presentity</et:type>
       <ext:photo>
```

```
          http://operator.example.com/services/users/bill/pictureX.jpg
        </ext:photo>
        <note xml:lang="en">At home</note>
      </tuple>
    </presence>
```

4. **XCAP 201 (CREATED) response (~~DMS~~XCAP server to DM) - see example in table A.8.4-4**

After the ~~DMS~~XCAP server has performed the necessary authorization checks, XML document validations and XML schema compliancy checks the ~~DMS~~XCAP server sends an XCAP 201 (Created) response to the DM.

Table A.8.4-4: XCAP 201 (Created) response (~~DMS~~XCAP server to DM)

```
HTTP/1.1 201 CREATED
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 10:50:45 GMT
Content-Length: 0
```

5. **HTTP GET request (~~DM~~ XCAP clientto ~~DMS~~XCAP server) – see example in table A.8.4-5**

The ~~DM~~ XCAP clientwishes to fetch the MIME object from the ~~DMS~~XCAP server. The client generates an HTTP GET request. The request URI points to the directory where the object is stored and indicates the name of the file to be fetched.

Table A.8.4-5: HTTP GET request (~~DM~~ XCAP clientto ~~DMS~~XCAP server)

```
GET http://operator.example.com/services/users/bill/pictureX HTTP/1.1
Host: oper.example.com:9999
User-Agent: IMS subscriber
Date: Thu, 08 Jan 2004 10:43:17 GMT
Accept: image/jpeg
Referer: http://oper.home1.net:1234/service
```

6. **HTTP 200 (OK) response (~~DMS~~XCAP server to DM) – see example in table A.8.4-6**

After the ~~DMS~~XCAP server has performed the necessary authorization checks on the originator to ensure the ~~DM~~ XCAP clientis allowed to fetch the file the ~~DMS~~XCAP server sends an HTTP 200 (OK) response having the object in the body to the DM.

Table A.8.4-6: HTTP 200 (OK) response (~~DMS~~XCAP server to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 11:00:47 GMT
Content-Type: image/jpeg
Content-Length: (…)

{pictureX}
```

7. **HTTP PUT request (~~DM~~ XCAP clientto ~~DMS~~XCAP server) – see example in table A.8.4-7**

The ~~DM~~ XCAP clientwishes to modify the earlier stored MIME object by replacing the picture X with a new picture X with new content. To modify the object the ~~DM~~ XCAP clientgenerates an HTTP PUT request using the same request URI as has been used for the modified (old) object. The new object is conveyed in the body of the request.

Table A.8.4-7: HTTP PUT request (~~DM~~ XCAP clientto ~~DMS~~XCAP server)

```
PUT http://operator.example.com/services/users/bill/pictureX HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 11:13:17 GMT
Content-Type: image/jpeg
Content-Length: (…)

{pictureX.jpg}
```

8. **HTTP 200 (OK) response (~~DMS~~XCAP server to DM) – see example in table A.8.4-8**

After the ~~DMS~~XCAP server has performed the necessary authorization checks on the originator to ensure the ~~DM~~ XCAP clientis allowed to replace the existing MIME object with the new one the ~~DMS~~XCAP server sends an HTTP 200 (OK) response to the DM.

Table A.8.4-8: HTTP 200 (OK) response (~~DMS~~XCAP server to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 11:50:35 GMT
Content-Length: 0
```

9. **XCAP PUT request (~~DM~~ XCAP clientto ~~DMS~~XCAP server) – see example in table A.8.4-9**

The ~~DM~~ XCAP clientwishes to remove the MIME object from his presence information. The ~~DM~~ XCAP clientgenerates an XCAP PUT request to modify the XML encoded presence document to remove the reference to the MIME object from the presence document. The request URI contains a node selector to the requested tuple according to draft-ietf-simple-xcap-02 [33]. Because the signalling flow does not contain the XCAP GET request the use of the If-Match header is omitted in this example.

Table A.8.4-9: XCAP PUT request (~~DM~~ XCAP clientto ~~DMS~~XCAP server)

```
PUT http://xcap.example.com/services/pidf-
  manipulation/users/bill/pidf.xml?presence/tuple[@id='432sd'] HTTP/1.1
Date: Thu, 08 Jan 2004 11:13:37 GMT
Content-Type: text/plain
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
   <tuple id="432sd">
     <status>
       <basic>open</basic>
     </status>
     <et:type>presentity</et:type>
     <note xml:lang="en">At home</note>
   </tuple>
```

10. **XCAP 200 (OK) response (~~DMS~~XCAP server to DM) - see example in table A.8.4-10**

After the ~~DMS~~XCAP server has performed the necessary authorization checks, XML document validations and XML Schema compliancy checks the ~~DMS~~XCAP server sends an XCAP 200 (OK) response to the DM.

Table A.8.4-10: XCAP 200 (OK) response (~~DMS~~XCAP server to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 11:50:59 GMT
Content-Length: 0
```

11. **HTTP DELETE request (~~DM~~ XCAP clientto ~~DMS~~XCAP server) – see example in table A.8.4-11**

The ~~DM~~ XCAP clientremoves the MIME object from the ~~DMS~~XCAP server by generating an HTTP DELETE request.

Table A.8.4-11: HTTP DELETE request (~~DM~~ XCAP clientto ~~DMS~~XCAP server)

```
DELETE http://operator.example.com/services/users/bill/pictureX HTTP/1.1
Host: oper.example.com:9999
User-Agent: IMS subscriber
Date: Thu, 08 Jan 2004 11:52:00 GMT
Referer: http://oper.home1.net:1234/service
```

12. **HTTP 200 (OK) response (~~DMS~~XCAP server to DM) – see example in table A.8.4-12**

After the ~~DMS~~XCAP server has performed the necessary authorization checks on the originator to ensure that the ~~DM~~ XCAP clientis allowed to delete the object, the ~~DMS~~XCAP server sends an HTTP 200 (OK) response to the DM.

Table A.8.4-12: HTTP 200 (OK) response (~~DMS~~XCAP server to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 11:52:35 GMT
Content-Length: 0
```

3GPP TSG-CN1 Meeting #36 **Tdoc N1-041774**
Seoul, Korea, 15-19 November 2004

CR-Form-v7.1

CHANGE REQUEST

⌘ **24.141** **CR** **23** ⌘ **rev** **-** ⌘ Current version: **6.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network ☒

Title:	⌘	Delete Authenticaiton Proxy requirements
Source:	⌘	Siemens
Work item code:	⌘	PRESNC
Date:	⌘	17/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	The Presence TS describes generic authenitcation related requirements for the functional entity “authentication proxy”. However, the stage-3 description for the Authentication proxy shall only be defined in 24.109. The Presence TS shall only describe presence service specific requirements.
Summary of change:	⌘	Subclause 6.2.3 for Authentication Proxy is deleted.
Consequences if not approved:	⌘	Double definition of requrirements for the authentication proxy. Authentication proxy can not be impelemented in a consistent manner.

Clauses affected:	⌘	3.1, 6.2.3

	Y	N		
Other specs affected: ⌘		X	Other core specifications	⌘
		X	Test specifications	
		X	O&M Specifications	

Other comments: ⌘

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

*** 1st change ***

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply:

subscription authorization policy: a policy that determines which watchers are allowed to subscribe to a presentity's presence information
The subscription authorization policy also determines to which presence tuples of the presentity's presence information the watcher has access.

For the purposes of the present document, the following terms and definitions given in 3GPP TS 23.141 [4] apply:

Presence list server
Presence Network Agent (PNA)
Presence Server (PS)
Presence User Agent (PUA)

For the purposes of the present document, the following terms and definitions from RFC 2778 [16] apply:

Presence tuple
Presentity

For the purposes of the present document, the following terms and definitions from draft-ietf-sip-publish-03 [23] apply:

Event Publication Agent (EPA)
Event State Compositor (ESC)

For the purposes of the present document, the following terms and definitions from draft draft-ietf-simple-xcap-03 [33] apply:

XCAP client
XCAP server

For the purposes of the present document, the following terms and definitions from draft-ietf-simple-event-list-04 [22] apply:

Resource List Server (RLS)

For the purposes of the present document, the following terms and definitions given in RFC 1594 [12].

Fully-Qualified Domain Name (FQDN)

For the purposes of the present document, the following terms and definitions given in RFC 3261 [17] apply (unless otherwise specified see clause 6).

Final response
Header
Header field
Method
Request
Response
(SIP) transaction
Status-code (see RFC 3261 [17], subclause 7.2)
Tag (see RFC 3261 [17], subclause 19.3)

For the purposes of the present document, the following terms and definitions given in 3GPP TS 23.002 [3], subclauses 4.1.1.1 and 4a.7 apply:

Call Session Control Function (CSCF)
Home Subscriber Server (HSS)

For the purposes of the present document, the following terms and definitions given in 3GPP TS 23.218 [5], subclause 3.1 apply:

Filter criteria
Initial filter criteria
Subsequent request

For the purposes of the present document, the following terms and definitions given in 3GPP TS 23.228 [6], subclauses 4.3.3.1 and 4.6 apply:

Interrogating-CSCF (I-CSCF)
Proxy-CSCF (P-CSCF)
Serving-CSCF (S-CSCF)

For the purposes of the present document, the following terms and definitions given in 3GPP TR 21.905 [1] apply:

User Equipment (UE)

For the purposes of the present document, the following terms and definitions from 3GPP TS 33.~~141~~ 222 [11] apply:

Authentication Proxy

*** next change ***

6.2.3 Authentication proxy

~~The authentication proxy shall implement the role of a network application function, as described in 3GPP TS 24.109 [7] and it shall support HTTP Digest Authentication (see RFC 2617 [15A] and certificate authentication.~~

~~The Authentication Proxy shall authenticate the UE and integrity protect the messages sent towards the UE.~~

Editor's note: ~~It is FFS how the Authentication Proxy passes the user's identity to the Application Server (AS).~~This sub-clause will contain service specific requirements for an authentication proxy. It is for further study whether presence specific requirements exist or whether all requirements are covered in 3GPP TS 24.109 [7].

3GPP TSG-CN1 Meeting #36
Seoul, Korea, 15-19 November 2004

Tdoc N1-041880

CR-Form-v7.1

CHANGE REQUEST

⌘ **24.141** **CR** **032** ⌘ **rev** **-** ⌘ Current version: **6.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network ☒

Title:	⌘	Filter criteria update
Source:	⌘	Lucent Technologies
Work item code:	⌘	PRESNC
Date:	⌘	05/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	At the previous meeting CN1#35, and at the subsequent plenary CN#25, N1-041413 (CR011 to 24.141) was approved. During the discussion of this prior to CN1#35, some improved wording was identified on the mailing list which unfortunately was not incorporated into the change. This CR seeks to make that change.
Summary of change:	⌘	Text appearing three times in document: "The S-CSCF#2 has preconfigured information not to record route for this request." is replaced by: "The S-CSCF#2 has preconfigured information not to create a Record-Route entry for this request.
Consequences if not approved:	⌘	Imprecise specification.

Clauses affected:	⌘	A.3.2.1, A.3.4.1, A.3.5

		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

PROPOSED CHANGE

A.3.2 Watcher and presentity in different networks, UE in home network

A.3.2.1 Successful subscription

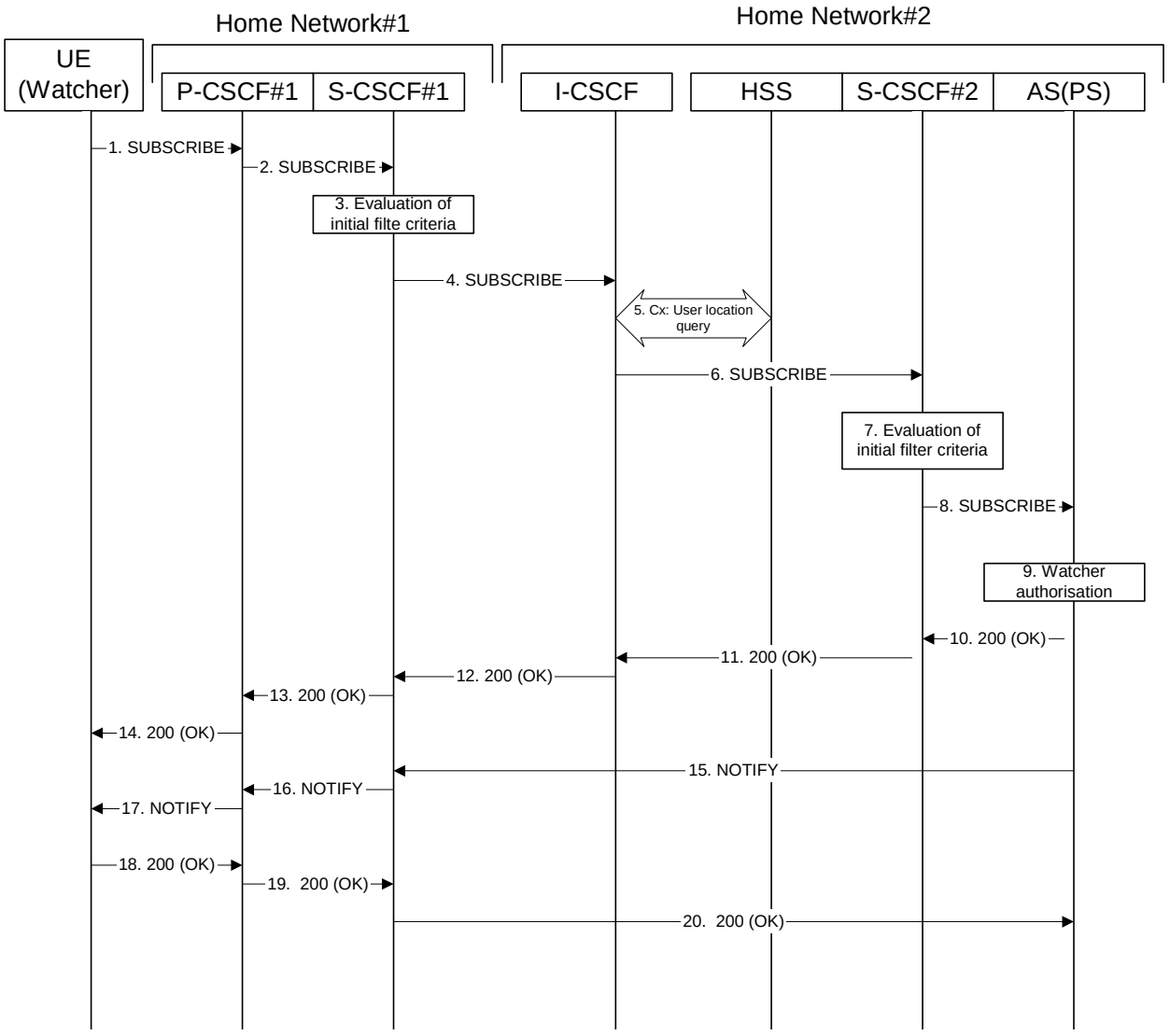


Figure A.3.2.1-1: Watcher subscribing for presence information

Figure A.3.2.1-1 shows a watcher subscribing to presence event notification about a presentity. The presentity is in a different IM CN subsystem. The details of the signalling flows are as follows:

1. **SUBSCRIBE request (UE (watcher) to P-CSCF) - see example in table A.3.2.1-1**

   A watcher agent in a UE wishes to watch a presentity, or certain presence tuples of the presentity. To initiate a subscription, the UE generates a SUBSCRIBE request containing the "presence" event that it wishes to be notified of, together with an indication of the length of time this periodic subscription should last and the support for partial notification.

Table A.3.2.1-1: SUBSCRIBE request (UE (watcher) to P-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
Max-Forwards: 70
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
Route: <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>, <sip:orig@scscf1.home1.net;lr>
P-Preferred-Identity: <sip:user1_public1@home1.net>
Privacy: none
From: <sip:user1_public1@home1.net>;tag=31415
To: <sip:user2_public1@home2.net>
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 61 SUBSCRIBE
Require: sec-agree
Proxy-Require: sec-agree
Security-Verify: ipsec-3gpp; q=0.1; alg=hmac-sha-1-96; spi-c=98765432; spi-s=87654321; port-
      c=8642; port-s=7531
Event: presence
Expires: 7200
Accept: application/pidf+xml;q=0.3, application/pidf-partial+xml;q=1
Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>
Content-Length: 0
```

Request-URI: Public user identity whose events the subscriber subscribes to.

Event: This field is populated with the value "presence" to specify the use of the presence package.

Accept: This field is populated with the value 'application/pidf+xml' and 'application/pidf-partial+xml', latter one with higher preference.

To: Same as the Request-URI.

2. **SUBSCRIBE request (P-CSCF to S-CSCF) - see example in table A.3.2.1-2**

The P-CSCF looks up the serving network information for the public user identity that was stored during the registration procedure. The SUBSCRIBE request is forwarded to S-CSCF. A Route header is inserted into SUBSCRIBE request. The information for the Route header is taken from the service route determined during registration.

Table A.3.2.1-2: SUBSCRIBE request (P-CSCF to S-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Access-Network-Info:
Max-Forwards: 69
P-Asserted-Identity: <sip:user1_public1@home1.net>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"
Privacy:
Route: <sip:orig@scscf1.home1.net;lr>
Record-Route: <sip:pcscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

3. **Evaluation of initial filter criteria**

S-CSCF#1 validates the service profile of this subscriber and evaluates the initial filter criteria. For this example, assume no Application Server involvement.

4. **SUBSCRIBE request (S-CSCF to I-CSCF) - see example in table A.3.2.1-4**

S-CSCF#1 performs an analysis of the destination address, and determines the network operator to whom the destination subscriber belongs. Since the originating operator does not desire to keep their internal configuration hidden, S-CSCF#1 forwards the SUBSCRIBE request directly to the I-CSCF in the destination network.

Table A.3.2.1-4: SUBSCRIBE (S-CSCF to I-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;;branch=z9hG4bKnashds7
Max-Forwards: 68
P-Asserted-Identity: <sip:user1_public1@home1.net>, <tel:+1-212-555-1111>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"; orig-ioi=home1.net
Privacy:
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

5. **Cx: User Location Query procedure**

The I-CSCF sends a query to the HSS to find out the S-CSCF of the called user. The HSS responds with the address of the current S-CSCF for the terminating subscriber.

For detailed message flows see 3GPP TS 29.228 [10].

Table A.3.2.1-5a provides the parameters in the SIP SUBSCRIBE request (flow 4), which are sent to the HSS.

Table A.3.2.1-5a: Cx: User registration status query procedure (I-CSCF to HSS)

Message source and destination	Cx: Information element name	Information source in SIP SUBSCRIBE	Description
I-CSCF to HSS	User Public Identity	Request-URI	This information element indicates the public user identity

Table A.3.2.1-5b provides the parameters sent from the HSS that need to be mapped to the SIP SUBSCRIBE request (flow 6) and sent to the S-CSCF.

Table A.3.2.1-5b: Cx: User registration status query procedure (HSS to I-CSCF)

Message source and destination	Cx: Information element name	Mapping to SIP header in SIP SUBSCRIBE	Description
HSS to I-CSCF	S-CSCF name	Route header field	This information indicates the serving CSCF's name of that user

6. **SUBSCRIBE request (I-CSCF to S-CSCF) - see example in table A.3.2.1-6**

The I-CSCF forwards the SUBSCRIBE request to the S-CSCF (S-CSCF#2) that will handle the termination.

Table A.3.2.1-6: SUBSCRIBE request (I-CSCF to S-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
Max-Forwards: 67
P-Asserted-Identity:
P-Charging-Vector:
Privacy:
Route: <sip:scscf2.home2.net;lr>
Record-Route:
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

NOTE: The I-CSCF does not add itself to the Record-Route header, as it has no need to remain in the signalling path for the subsequent requests.

7. **Evaluation of initial filter criteria**

S-CSCF#2 validates the service profile of this subscriber and evaluates the initial filter criteria. For sip:user2_public1@home2.net S-CSCF#2 has termination initial filter criteria with service points of interest of Method = SUBSCRIBE and Event = "presence" that informs the S-CSCF to route the SUBSCRIBE request to the Application Server ps.home2.net. The S-CSCF#2 has preconfigured information not to ~~record route~~ create a Record-Route entry for this request.

8. **SUBSCRIBE request (S-CSCF to PS) – see example in table A.3.2.1-8**

The S-CSCF forwards the SUBSCRIBE request to the PS.

Table A.3.2.1-8: SUBSCRIBE request (S-CSCF to PS)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP scscf2.home2.net;branch=z9hG4bK764z87.1, SIP/2.0/UDP
      icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
Max-Forwards: 66
P-Asserted-Identity:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Privacy:
Route: <sip:ps.home2.net;lr>, <sip:scscf2.home2.net;lr>
Record-Route:
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the originating

Inter Operator Identifier (IOI) parameter of this header and removes the terminating IOI parameter.

P-Charging-Function-Addresses: The S-CSCF populates the P-Charging-Function-Addresses header field to be passed to the PS.

9. **Authorization of watcher**

The PS performs the necessary authorization checks on the originator to ensure it is allowed to watch the presentity. In this example all privacy conditions are met, so the PS sends a 200 (OK) response to the S-CSCF.

In the case where the privacy/authorization checks failed, then a necessary 2xx or 4xx response would be sent to the S-CSCF. The selection of the correct response code depends on the presentity's subscription authorization policy document.

10. **200 (OK) response (PS to S-CSCF) - see example in table A.3.2.1-10**

The PS sends the response to S-CSCF#2.

Table A.3.2.1-10: 200 (OK) response (PS to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf2.home2.net;branch=z9hG4bK764z87.1, SIP/2.0/UDP
icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"; orig-ioi=home1.net;
      term-ioi=home2.net
P-Charging-Functio-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Record-Route:
From:
To: <sip:user2_public1@home2.net>;tag=151170
Call-ID:
CSeq:
Expires:
Contact: <sip:ps.home2.net>
Content-Length:
```

P-Charging-Vector: The PS stores the originating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The PS stores the P-Charging-Function-Addresses header field and passes this header to the S-CSCF.

11. **200 (OK) response (S-CSCF to I-CSCF) - see example in table A.3.2.1-11**

S-CSCF#2 forwards the response to I-CSCF#2.

Table A.3.2.1-11: 200 (OK) response (S-CSCF to I-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Charging-Vector:
P-Charging-Function-Addresses:
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the I-CSCF.

12. **200 (OK) response (I-CSCF to S-CSCF) - see example in table A.3.2.1-12**

I-CSCF#2 forwards the response to S-CSCF#1.

Table A.3.2.1-12: 200 (OK) response (I-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Charging-Vector:
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

13. **200 (OK) response (S-CSCF to P-CSCF) - see example in table A.3.2.1-13**

S-CSCF#1 forwards the response to P-CSCF#1.

Table A.3.2.1-13: 200 (OK) response (S-CSCF to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

14. **200 (OK) response (P-CSCF to UE) - see example in table A.3.2.1-14**

P-CSCF#1 forwards the response to the watcher agent in the UE.

Table A.3.2.1-14: 200 (OK) response (P-CSCF to UE)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
Record-Route: <sip:orig@scscf1.home1.net;lr>, <sip:pcscf1.home1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

15. **NOTIFY request (PS to S-CSCF) - see example in table A.3.2.1-15**

As soon as the PS sends a 200 (OK) response to accept the subscription, it sends a NOTIFY request with the current state of the presentity's presence tuples that the watcher has subscribed and been authorized to. The

NOTIFY request is sent to S-CSCF#1. Based on the Accept header field of the SUBSCRIBE request, the PS decides to use the 'application/pidf-partial+xml' content type in the NOTIFY request.

Table A.3.2.1-15: NOTIFY request (PS to S-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 70
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=123551024"; orig-ioi=home2.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.home1.net;lr>
From: <sip:user2_public1@home2.net>;tag=151170
To: <sip:user1_public1@home1.net>;tag=31415
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 42 NOTIFY
Subscription-State: active ;expires=7200
Event: presence
Contact: <sip:ps.home2.net>
Content-Type: application/pidf-partial+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
   <pidf-part:presence xmlns="urn:ietf:params:xml:ns: pidf"
              xmlns:pidf-part="urn:ietf:params:xml:ns:pidf-partial"
              xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
              xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
              xmlns:pcp="urn:ietf:params:xml:ns:simple-prescaps-ext"
              xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
         entity="pres:user2_public1@home2.net" version="0" state="full">

     <tuple id="a8098a.672364762364">
       <status>
         <basic>open</basic>
         <es:activity>meeting</es:activity>
         <es:placetype until="2003-08-27T17:30:00Z">office</es:placetype>
         <es:privacy>private</es:privacy>
         <es:idle since="2003-08-27T10:43:00Z"/>
         <pcp:prescaps>
           <pcp:video negated="false"></pcp:video>
           <pcp:mobility>mobile</pcp:mobility>
           <pcp:audio negated="true"></pcp:audio>
         </pcp:prescaps>
       </status>
       <et:class>sip</et:class>
       <et:contact-type>service</et:contact-type>
       <contact priority="0.8">sip:user2_public1@home2.net</contact>
       <note xml:lang="en">Don't Disturb Please!</note>
       <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
       <timestamp>2003-08-27T11:49:29Z</timestamp>
     </tuple>

     <tuple id="sfddsj74.78">
       <status>
         <basic>open</basic>
       </status>
       <et:class>presentity</et:class>
       <et:contact-type>presentity</et:contact-type>
       <ci:homepage>http://example.com/~user2</ci:homepage>
       <ci:icon>http://example.com/~user2/icon.gif</ci:icon>
       <ci:card>http://example.com/~user2/card.vcd</ci:card>
       <note xml:lang="en">I'm in a boring meeting</note>
       <note xml:lang="en">I'll be in Tokyo next week</note>
       <timestamp>2003-08-27T11:49:29Z</timestamp>
     </tuple>

     <tuple id="jklhgf9788934774.78">
       <status>
         <basic>open</basic>
       </status>
       <et:class>assistant</et:class>
       <et:contact-type>presentity</et:contact-type>
       <et:relationship>assistant</et:relationship>
       <contact priority="1.0">tel:+1-212-555-2222</contact>
       <note xml:lang="en">She's my secretary</note>
       <timestamp>2003-08-27T11:49:29Z</timestamp>
     </tuple>

   </pidf-part:presence>
```

P-Charging-Vector: The PS populates the icid parameter with a globally unique identifier and adds the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The PS populates the P-Charging-Function-Addresses header field to be passed to the S-CSCF.

Content-Type: Set to the preferred value of the Accept header received in the SUBSCRIBE request.

The message body in the NOTIFY request that carries the presence information of the presentity is formed as indicated in draft-ietf-impp-cpim-pidf-08 [21], draft-ietf-simple-rpid-03 [26], draft-ietf-simple-prescaps-ext-00 [25], draft-ietf-simple-cipid-01 [32] draft-ietf-simple-partial-notify-01 [24] and draft-ietf-simple-partial-pidf-format-00 [38].

16. **NOTIFY request (S-CSCF to P-CSCF) - see example in table A.3.2.1-16**

The S-CSCF#1 forwards the NOTIFY request to P-CSCF#1.

Table A.3.2.1-16: NOTIFY request (S-CSCF to P-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 69
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=123551024"
P-Charging-Function-Addresses:
Privacy:
Record-Route: <sip:scscf1.home1.net;lr>
Route: sip:<pcscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter of this header and removes the parameter from this header.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the P-CSCF.

17. **NOTIFY request (P-CSCF to UE) - see example in table A.3.2.1-17**

The P-CSCF forwards the NOTIFY request to the watcher application in the UE.

Table A.3.2.1-17: NOTIFY request (P-CSCF to UE)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 68
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.home1.net:7531;lr;comp=sigcomp>
Privacy:
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

18. **200 (OK) response (UE to P-CSCF) - see example in table A.3.2.1-18**

The UE generates a 200 (OK) response to the NOTIFY request.

Table A.3.2.1-18: 200 (OK) response (UE to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      scscf2.home2.net;branch=z9hG4bK764z87.1, SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

19. **200 (OK) response (P-CSCF to S-CSCF) – see example in table A.3.2.1-19**

The P-CSCF forwards the 200 (OK) response to S-CSCF#1.

Table A.3.2.1-19: 200 (OK) response (P-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      ps.home2.net;branch=z9hG4bK348923.1
P-Access-Network-Info:
From:
To:
Call-ID:
CSeq:
Content-Length:
```

20. **200 (OK) response (S-CSCF to P-S) - see example in table A.3.2.1-20**

S-CSCF#2 forwards the 200 (OK) response to the PS.

Table A.3.2.1-20: 200 (OK) response (S-CSCF to PS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
From:
To:
Call-ID:
CSeq:
Content-Length:
```

PROPOSED CHANGE

A.3.4 RLS subscribing to presentities in different network

A.3.4.1 Successful subscription

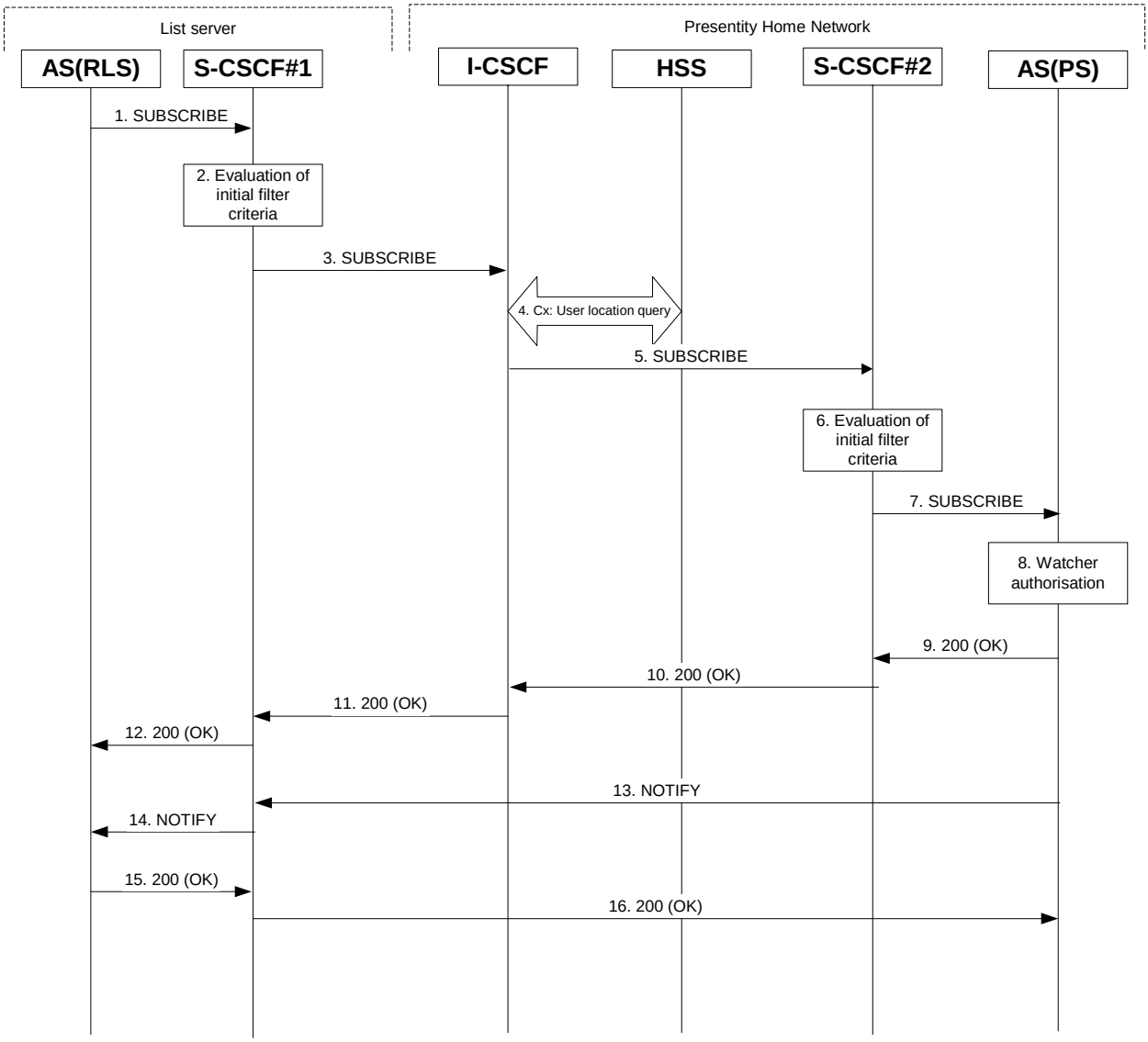


Figure A.3.4.1-1 RLS subscribing to presentities in different network

Figure A.3.4.1-1 shows the RLS subscribing to presence event notification about a presentity. The presentity is in a different IM CN subsystem. The details of the signalling flows are as follows:

1. **SUBSCRIBE request (RLS to S-CSCF) – see example in table A.3.4.1-1**

   The RLS resolves the watcher's resource address (the address is received according to subclause A.3.3) and subscribes to presence event notification at all the presentities that are represented by the resource list SIP URI. The home network of these presentities can be different or in the same network, as the RLS. In this example only a single subscription is shown where the home network of the presentity is another network. Subscriptions to other presentities follow a similar procedure. To initiate a subscription, the RLS generates a SUBSCRIBE request containing the "presence" event that it wishes to be notified of, together with an indication of the length of time this periodic subscription should last. The RLS sends the SUBSCRIBE request to the S-CSCF of "sip:user1_public1@home1.net" (S-CSCF#1). The address of S-CSCF#1 is either

remembered from previous transactions (when "sip:user1_public1@home1.net" has subscribed for the resource list) or queried by the RLS using the Sh interface.

Table A.3.4.1-1 SUBSCRIBE request (RLS to S-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bKehuefdam
Max-Forwards: 70
Route: <sip:scscf1.home1.net;lr>
P-Asserted-Identity: <sip:user1_public1@home1.net>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
From: <sip:user1_public1@home1.net>;tag=31415
To: <sip:user2_public1@home2.net>
Call-ID: q987a9a87g087abgf7qyg7ag
CSeq: 123 SUBSCRIBE
Event: presence
Expires: 7200
Accept: application/pidf+xml
Contact: <sip:rls.home1.net>
Content-Length: 0
```

Request-URI: Public user identity whose events the RLS subscribes to.

P-Charging-Vector: The RLS populates the icid parameter with a new globally unique value and populates the originating Inter Operator Identifier (IOI) parameter with the identifier of its own network of RLS.

P-Charging-Function-Addresses: The RLS populates the P-Charging-Function-Addresses header field to be passed to the S-CSCF.

To: Same as the Request-URI.

Event: This field is populated with the value "presence" to specify the use of the presence package.

Accept: This field is populated with the value "application/pidf+xml".

2. **Evaluation of initial filter criteria**

S-CSCF#1 validates the service profile of this subscriber and evaluates the initial filter criteria. For this example, assume no application server involvement.

3. **SUBSCRIBE request (S-CSCF to I-CSCF) – see example in table A.3.4.1-3**

S-CSCF#1 performs an analysis of the destination address, and determines the network operator to whom the destination subscriber belongs. S-CSCF#1 forwards the request to the I-CSCF.

Table A.3.4.1-3 SUBSCRIBE request (S-CSCF to I-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bKehuehjgt, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bKehuefdam
Max-Forwards: 69
Record-Route: <sip:orig@scscf1.home1.net;lr>
P-Asserted-Identity:
P-Charging-Vector:
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter received.

4. **Cx: User Location Query procedure**

The I-CSCF sends a query to the HSS to find out the S-CSCF of the presentity. The HSS responds with the address of the current S-CSCF for the presentity.

For detailed message flows see 3GPP TS 29.228 [10].

Table A.3.4.1-4a provides the parameters in the SIP SUBSCRIBE request (flow 3), which are sent to the HSS.

Table A.3.4.1-4a: Cx: User registration status query procedure (I-CSCF to HSS)

Message source and destination	Cx: Information element name	Information source in SIP SUBSCRIBE	Description
I-CSCF to HSS	User Public Identity	Request-URI	This information element indicates the public user identity

Table A.3.4.1-4b provides the parameters sent from the HSS that need to be mapped to SIP SUBSCRIBE request (flow 5) and sent to the S-CSCF.

Table A.3.4.1-4b: Cx: User registration status query procedure (HSS to I-CSCF)

Message source and destination	Cx: Information element name	Mapping to SIP header in SIP SUBSCRIBE	Description
HSS to I-CSCF	S-CSCF name	Route header field	This information indicates the serving CSCF's name of that user

5. **SUBSCRIBE request (I-CSCF to S-CSCF) - see example in table A.3.4.1-5**

The I-CSCF forwards the SUBSCRIBE request to the S-CSCF#2 that will handle the termination.

Table A.3.4.1-5: SUBSCRIBE request (I-CSCF to S-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP icscf2_s.home2.net;branch=z9hG4bKj5hgrt2o, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bKehuehjgt, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bKehuefdam
Max-Forwards: 68
P-Asserted-Identity:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"; orig-ioi=home1.net;
Route: <sip:scscf2.home2.net;lr>
Record-Route:
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

6. **Evaluation of initial filter criteria**

S-CSCF#2 validates the service profile of this subscriber and evaluates the initial filter criteria. For sip:user2_public1@home2.net the S-CSCF has Termination initial Filter Criteria with Service Points of Interest of Method = SUBSCRIBE AND Event = "presence"that informs the S-CSCF to route the SUBSCRIBE request to the Application Server ps.home2.net. The S-CSCF#2 has preconfigured information not to ~~record route~~ create a Record-Route entry for this request.

7. **SUBSCRIBE request (S-CSCF to PS) - see example in table A.3.4.1-7**

The S-CSCF#2 forwards the SUBSCRIBE request to the PS.

Table A.3.4.1-7 SUBSCRIBE request (S-CSCF to PS)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP scscf2.home2.net;branch=z9hG4bK344a65.1, SIP/2.0/UDP
      icscf2_s.home2.net;branch=z9hG4bKj5hgrt2o, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bKehuehjgt, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bKehuefdam
Max-Forwards: 67
P-Asserted-Identity:
P-Charging-Vector:
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:ps.home2.net;lr>, <sip:scscf2.home2.net;lr>
Record-Route: <sip:orig@scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF populates the P-Charging-Function-Addresses header field to be passed to the PS.

8. **Authorization of watcher**

The PS performs the necessary authorization checks on the originator to ensure it is allowed to watch the presentity. In this example all privacy conditions are met, so the PS sends a 200 (OK) response to the S-CSCF.

In the case where the privacy/authorization checks failed, then a necessary 2xx or 4xx response would be sent to the S-CSCF. The selection of the correct response code depends on the presentity's subscription authorization policy document.

9. **200 (OK) response (PS to S-CSCF) - see example in table A.3.4.1-9**

The PS sends the response to S-CSCF#2.

Table A.3.4.1-9: 200 (OK) response (PS to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf2.home2.net;branch=z9hG4bK344a65.1, SIP/2.0/UDP
      icscf2_s.home2.net;branch=z9hG4bKj5hgrt2o, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bKehuehjgt, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bKehuefdam
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"; orig-
      ioi=home2.net:term-ioi=home2.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Record-Route:
From:
To: <sip:user2_public1@home2.net>;tag=151170
Call-ID:
CSeq:
Expires:
Contact: <sip:ps.home2.net;lr>
Content-Length: 0
```

P-Charging-Vector: The PS stores the originating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The PS stores the P-Charging-Function-Addresses header field and passes this header to the S-CSCF.

10. **200 (OK) response (S-CSCF to I-CSCF) - see example in table A.3.4.1-10**

S-CSCF#2 forwards the response to the I-CSCF.

Table A.3.4.1-10: 200 (OK) response (S-CSCF to I-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP icscf2_s.home2.net;branch=z9hG4bKj5hgrt2o, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bKehuehjgt, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bKehuefdam
P-Charging-Vector:
P-Charging-Function-Addresses:
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the terminating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the I-CSCF.

11. **200 (OK) response (I-CSCF to S-CSCF) - see example in table A.3.4.1-11**

The I-CSCF forwards the response to S-CSCF#1.

Table A.3.4.1-11: 200 (OK) response (I-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bKehuehjgt, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bKehuefdam
P-Charging-Vector:
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

12. **200 (OK) response (S-CSCF to RLS) - see example in table A.3.4.1-12**

S-CSCF#1 forwards the response to the RLS.

Table A.3.4.1-12: 200 (OK) response (S-CSCF to RLS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bKehuefdam
P-Charging-Vector:
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the terminating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the RLS.

13. **NOTIFY request (PS to S-CSCF) - see example in table A.3.4.1-13**

As soon as the PS sends a 200 (OK) response to accept the subscription, it sends a NOTIFY request with the current state of the presentity's presence tuples that the watcher has subscribed and been authorized to. The NOTIFY request is sent to S-CSCF#1. Further notification sent by the PS may either contain the complete set of presence information, or only those presence tuples that have changed since the last notification.

Table A.3.4.1-13: NOTIFY request (PS to S-CSCF)

```
NOTIFY sip:rls.home1.net SIP/2.0
Via: SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 70
Route: <sip:scscf1.home1.net;lr>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"; orig-ioi=home2.net
From: <sip:user1_public1@home2.net>;tag=151170
To: <sip:rls.home1.net>;tag=31415
Call-ID: q987a9a87g087abgf7qyg7ag
CSeq: 42 NOTIFY
Subscription-State:active;expires=7200
Event: presence
Contact: <sip:ps.home2.net>
Content-Type: application/pidf+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
   <presence xmlns="urn:ietf:params:xml:ns:pidf"
             xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
             xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
             xmlns:pcp="urn:ietf:params:xml:ns:simple-prescaps-ext"
             xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
         entity="pres:user2_public1@home2.net ">

     <tuple id="a8098a.672364762364">
       <status>
         <basic>open</basic>
         <es:activity>meeting</es:activity>
         <es:placetype until="2003-08-27T17:30:00Z">office</es:placetype>
         <es:privacy>private</es:privacy>
         <es:idle since="2003-08-27T10:43:00Z"/>
         <pcp:prescaps>
           <pcp:video negated="false"></pcp:video>
           <pcp:mobility>mobile</pcp:mobility>
           <pcp:audio negated="true"></pcp:audio>
         </pcp:prescaps>
       </status>
       <et:class>sip</et:class>
       <et:contact-type>service</et:contact-type>
       <contact priority="0.8">im:user2_public1@home2.net</contact>
       <note xml:lang="en">Don't Disturb Please!</note>
       <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
       <timestamp>2003-08-27T11:49:29Z</timestamp>
     </tuple>

     <tuple id="sfddsj74.78">
       <status>
         <basic>open</basic>
       </status>
       <et:class>presentity</et:class>
       <et:contact-type>presentity</et:contact-type>
       <ci:homepage>http://example.com/~user2</ci:homepage>
       <ci:icon>http://example.com/~user2/icon.gif</ci:icon>
       <ci:card>http://example.com/~user2/card.vcd</ci:card>
       <note xml:lang="en">I'm in a boring meeting</note>
       <note xml:lang="en">I'll be in Tokyo next week</note>
       <timestamp>2004-10-10T12:00:30Z</timestamp>
     </tuple>

     <tuple id="jklhgf9788934774.78">
       <status>
         <basic>open</basic>
       </status>
       <et:class>assistant</et:class>
       <et:contact-type>presentity</et:contact-type>
       <et:relationship>assistant</et:relationship>
       <contact priority="1.0">tel:+1-212-555-2222</contact>
       <note xml:lang="en">She's my secretary</note>
```

```
        <timestamp>2003-08-27T11:49:29Z</timestamp>
      </tuple>

   </presence>
```

P-Charging-Vector: The PS populates the icid parameter with a globally unique value and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

Content-Type: Set to the value of the Accept header received in the SUBSCRIBE request or "application/pidf+xml".

The message body in the NOTIFY request that carries the subscriber's registration state is formed as indicated in draft-ietf-impp-cpim-pidf-08 [21], draft-ietf-simple-rpid-03 [26], draft-ietf-simple-cipid-01 [32] and draft-ietf-simple-prescaps-ext-00 [25].

14. **NOTIFY request (S-CSCF to RLS) - see example in table A.3.4.1-14**

The S-CSCF#1 forwards the NOTIFY request to the RLS.

Table A.3.4.1-14: NOTIFY request (S-CSCF to RLS)

```
NOTIFY sip:rls.home1.net SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bKehuehjgt, SIP/2.0/UDP
      ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 69
P-Charging-Vector:
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Record-Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF populates the P-Charging-Function-Addresses header field to be passed to the RLS.

15. **200 (OK) response (RLS to S-CSCF) – see example in table A.3.4.1-15**

The RLS generates a 200 (OK) response to the NOTIFY request.

Table A.3.4.1-15: 200 (OK) response (RLS to S-CSCF)

```
SIP/2.0 200 OK
Via:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"; orig-
      ioi=home1.net:term-ioi=home1.net
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

P-Charging-Vector: The RLS stores the originating Inter Operator Identifier (IOI) parameter and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

16. **200 (OK) response (S-CSCF to PS) – see example in table A.3.4.1-16**

The S-CSCF#1 forwards the 200 (OK) response to the PS.

Table A.3.4.1-16: 200 (OK) response (S-CSCF to PS)

```
SIP/2.0 200 OK
P-Charging-Vector:
Via: SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

P-Charging-Vector: The S-CSCF stores the terminating Inter Operator Identifier (IOI) parameter and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

PROPOSED CHANGE

A.3.5 Network based watcher subscribing on behalf of IMS watcher to IMS presentities

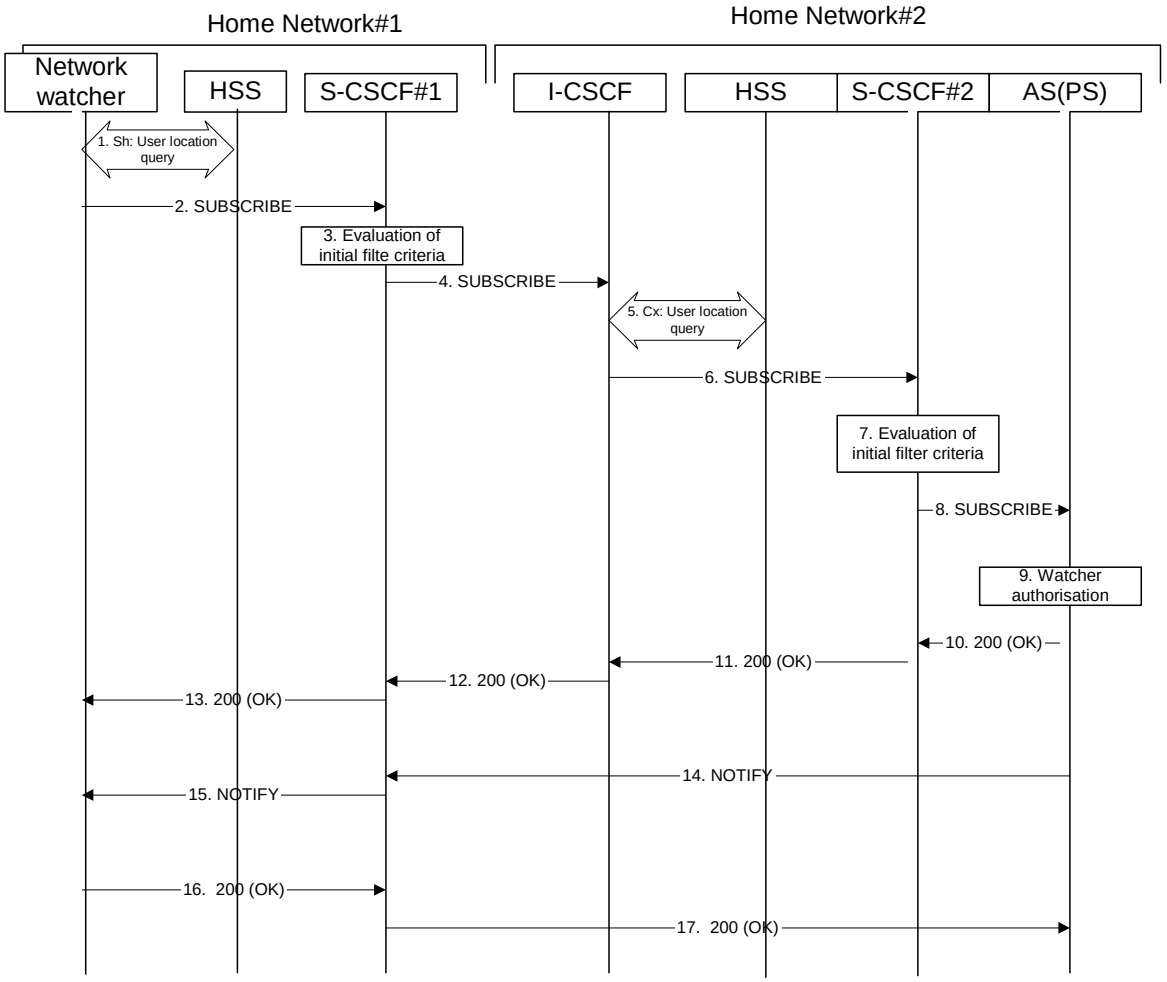


Figure A.3.5-1: Network based watcher subscribing on behalf of IMS watcher for presence information of IMS presentities

Figure A.3.5-1 shows a trusted network based watcher subscribing on behalf of an IMS watcher to presence event notification about an IMS based presentity. The presentity is in a different IM CN subsystem than the network based watcher and the signalling flow assumes that the IMS watcher on whose behalf the network based watcher subscribes is registered to the IMS network. The details of the signalling flows are as follows:

1. **Sh: User Location Query procedure**

   The network based watcher sends a query to the HSS to find out the S-CSCF of the user on whose behalf the subscription is initiated. The HSS responds with the address of the current S-CSCF for the originating subscriber.

2. **SUBSCRIBE request (Network based watcher to S-CSCF) - see example in table A.3.5-2**

The SUBSCRIBE request is constructed and forwarded to S-CSCF. The S-CSCF is inserted into the Route header of the SUBSCRIBE request.

Table A.3.5-2: SUBSCRIBE request (network watcher to S-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP watcher.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
Max-Forwards: 69
P-Asserted-Identity: <sip:user1_public1@home1.net>
Privacy: none
Route: <sip:scscf1.home1.net;lr;orig>
From: <sip:user1_public1@home1.net>;tag=31415
To: <sip:user2_public1@home2.net>
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 61 SUBSCRIBE
Event:PRESENCE
Expires: 7200
Accept: application/pidf+xml;q=0.3, application/pidf-partial+xml;q=1
Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>
Content-Length: 0
```

Request-URI: Public user identity of the user to whose events the subscriber subscribes to.

P-Asserted-Identity: The network based watcher inserts the public user identity of the watcher on whose behalf the subscription is made into the P-Asserted-Identity header field..

Route: The Route header is populated with the address of the S-CSCF obtained from the response to the user location query performed by the network based watcher on the Sh interface.

Event: This field is populated with the value "presence" to specify the use of the presence package.

Contact: The contact information of the network based watcher.

3. **Evaluation of initial filter criteria**

S-CSCF#1 validates the service profile of the subscriber identified in the P-Asserted-Identity header field and evaluates the initial filter criteria. For this example, assume no Application Server involvement.

4. **SUBSCRIBE request (S-CSCF to I-CSCF) - see example in table A.3.5-4**

S-CSCF#1 performs an analysis of the destination address, and determines the network operator to whom the destination subscriber belongs. Since the originating operator does not desire to keep their internal configuration hidden, S-CSCF#1 forwards the SUBSCRIBE request directly to the I-CSCF in the destination network.

Table A.3.5-4: SUBSCRIBE (S-CSCF to I-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      network.home1.net;branch=z9hG4bK240f34.1,
Max-Forwards: 68
P-Asserted-Identity: <sip:user1_public1@home1.net>
Privacy:
Record-Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

5. **Cx: User Location Query procedure**

The I-CSCF sends a query to the HSS to find out the S-CSCF of the called user. The HSS responds with the address of the current S-CSCF for the terminating subscriber.

For detailed message flows see 3GPP TS 29.228 [10].

Table A.3.5-5a provides the parameters in the SIP SUBSCRIBE request (flow 4), which are sent to the HSS.

Table A.3.5-5a: Cx: User registration status query procedure (I-CSCF to HSS)

Message source and destination	Cx: Information element name	Information source in SIP SUBSCRIBE	Description
I-CSCF to HSS	User Public Identity	Request-URI	This information element indicates the public user identity

Table A.3.5-5b provides the parameters sent from the HSS that need to be mapped to the SIP SUBSCRIBE request (flow 6) and sent to the S-CSCF.

Table A.3.5-5b: Cx: User registration status query procedure (HSS to I-CSCF)

Message source and destination	Cx: Information element name	Mapping to SIP header in SIP SUBSCRIBE	Description
HSS to I-CSCF	S-CSCF name	Route header field	This information indicates the serving CSCF's name of that user

6. **SUBSCRIBE request (I-CSCF to S-CSCF) - see example in table A.3.5-6**

The I-CSCF forwards the SUBSCRIBE request to the S-CSCF (S-CSCF#2) that will handle the termination.

Table A.3.5-6: SUBSCRIBE request (I-CSCF to S-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      network.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 67
P-Asserted-Identity:
Privacy:
Route: <sip:scscf2.home2.net;lr>
Record-Route:
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

NOTE: The I-CSCF does not add itself to the Record-Route header, as it has no need to remain in the signalling path for the subsequent requests.

7. **Evaluation of initial filter criteria**

S-CSCF#2 validates the service profile of this subscriber and evaluates the initial filter criteria. For sip:user2_public1@home2.net S-CSCF#2 has termination initial filter criteria with service points of interest of Method = SUBSCRIBE and Event = "presence" that informs the S-CSCF to route the SUBSCRIBE request to the Application Server ps.home2.net. The S-CSCF#2 has preconfigured information not to ~~record route~~ create a Record-Route entry for this request.

8. **SUBSCRIBE request (S-CSCF to PS) - see example in table A.3.5-8**

The S-CSCF forwards the SUBSCRIBE request to the PS.

Table A.3.5-8: SUBSCRIBE request (S-CSCF to PS)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP scscf2.home2.net;branch=z9hG4bK764z87.1, SIP/2.0/UDP
      icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      network.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 66
P-Asserted-Identity:
Privacy:
Route: <sip:ps.home2.net;lr>, <sip:scscf2.home2.net;lr>
Record-Route: <sip: scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

9. **Authorization of watcher**

The PS performs the necessary authorization checks on the watcher whose behalf the subscription is being made to ensure it is allowed to watch the presentity. In this example all privacy conditions are met, so the PS sends a 200 (OK) response to the S-CSCF.

In the case where the privacy/authorization checks failed, then a necessary 2xx or 4xx response would be sent to the S-CSCF. The selection of the correct response code depends on the presentity's authorization policy document.

10. **200 (OK) response (PS to S-CSCF) - see example in table A.3.5-10**

The PS sends the response to S-CSCF#2.

Table A.3.5-10: 200 (OK) response (PS to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf2.home2.net;branch=z9hG4bK764z87.1, SIP/2.0/UDP
icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      network.home1.net;branch=z9hG4bK240f34.1
Record-Route:
From:
To: <sip:user2_public1@home2.net>;tag=151170
Call-ID:
CSeq:
Expires:
Contact: <sip:ps.home2.net>
Content-Length:
```

11. **200 (OK) response (S-CSCF to I-CSCF) - see example in table A.3.5-11**

S-CSCF#2 forwards the response to I-CSCF#2.

Table A.3.5-11: 200 (OK) response (S-CSCF to I-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      network.home1.net;branch=z9hG4bK240f34.1
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

12. **200 (OK) response (I-CSCF to S-CSCF) - see example in table A.3.5-12**

I-CSCF#2 forwards the response to S-CSCF#1.

Table A.3.5-12: 200 (OK) response (I-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      network.home1.net;branch=z9hG4bK240f34.1
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

13. **200 (OK) response (S-CSCF to network watcher) - see example in table A.3.5-13**

S-CSCF#1 forwards the response to request originator.

Table A.3.5-13: 200 (OK) response (S-CSCF to network watcher)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP network.home1.net;branch=z9hG4bK240f34.1
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

14. **NOTIFY request (PS to S-CSCF) - see example in table A.3.5-14**

As soon as the PS sends a 200 (OK) response to accept the subscription, it sends a NOTIFY request with the current state of the presentity's presence information that the watcher has subscribed and been authorized to. The NOTIFY request is sent to S-CSCF#1. Based on the Accept header field of the SUBSCRIBE request, the PS decides to use the 'application/pidf-partial+xml' content type in the NOTIFY request.

Table A.3.5-14: NOTIFY request (PS to S-CSCF)

```
NOTIFY sip: network.home1.net;branch=z9hG4bK240f34.1 SIP/2.0
Via: SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 70
Route: <sip:scscf1.home1.net;lr>
From: <sip:user2_public1@home2.net>;tag=151170
To: <sip:user1_public1@home1.net>;tag=31415
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 42 NOTIFY
Subscription-State: active; expires=7200
Event: presence
Contact: <sip:ps.home2.net>
Content-Type: application/pidf-partial+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
   <pidf-part:presence xmlns="urn:ietf:params:xml:ns: pidf-partial"
              xmlns:pidf-part="urn:ietf:params:xml:ns:pidf-partial"
              xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
              xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
              xmlns:pcp="urn:ietf:params:xml:ns:simple-prescaps-ext"
              xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
                   entity="pres:user2_public1@home2.net" version="1" state="full">

     <tuple id="a8098a.672364762364">
       <status>
         <basic>open</basic>
         <es:activity>meeting</es:activity>
         <es:placetype until="2003-08-27T17:30:00Z">office</es:placetype>
         <es:privacy>private</es:privacy>
         <es:idle since="2003-08-27T10:43:00Z"/>
         <pcp:prescaps>
           <pcp:video negated="false"></pcp:video>
           <pcp:mobility>mobile</pcp:mobility>
           <pcp:audio negated="true"></pcp:audio>
         </pcp:prescaps>
       </status>
       <et:class>sip</et:class>
       <et:contact-type>service</et:contact-type>
       <contact priority="0.8">sip:user2_public1@home2.net</contact>
       <note xml:lang="en">Don't Disturb Please!</note>
       <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
       <timestamp>2003-08-27T11:49:29Z</timestamp>
     </tuple>

     <tuple id="sfddsj74.78">
       <status>
         <basic>open</basic>
       </status>
       <et:class>presentity</et:class>
       <et:contact-type>presentity</et:contact-type>
       <ci:homepage>http://example.com/~user2</ci:homepage>
       <ci:icon>http://example.com/~user2/icon.gif</ci:icon>
       <ci:card>http://example.com/~user2/card.vcd</ci:card>
       <note xml:lang="en">I'm in a boring meeting</note>
       <note xml:lang="en">I'll be in Tokyo next week</note>
       <timestamp>2004-10-10T12:00:30Z</timestamp>
     </tuple>

     <tuple id="jklhgf9788934774.78">
       <status>
         <basic>open</basic>
       </status>
       <et:class>assistant</et:class>
       <et:contact-type>presentity</et:contact-type>
       <et:relationship>assistant</et:relationship>
       <contact priority="1.0">tel:+1-212-555-2222</contact>
       <note xml:lang="en">She's my secretary</note>
```

```
        <timestamp>2003-08-27T11:49:29Z</timestamp>
      </tuple>

   </pidf-part:presence>
```

From: The tag of this field matches that of the To field in the received 200 (OK) response for the SUBSCRIBE request.

Content-Type: Set to the preferred value of the Accept header received in the SUBSCRIBE request.

The message body in the NOTIFY request that carries the presence information of the presentity is formed as indicated in draft-ietf-impp-cpim-pidf-08 [21], draft-ietf-simple-rpid-03 [26], draft-ietf-simple-cipid-01 [32], draft-ietf-simple-prescaps-ext-00 [25] and draft-ietf-simple-partial-notify-01 [24] and draft-ietf-simple-partial-pidf-format-00 [38].

15. **NOTIFY request (S-CSCF to network watcher) - see example in table A.3.5-15**

The S-CSCF#1 forwards the NOTIFY request to the network watcher

Table A.3.5-15: NOTIFY request (S-CSCF to network watcher)

```
NOTIFY sip: network.home1.net;branch=z9hG4bK240f34.1SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 69
Privacy:
Record-Route: <sip:scscf1.home1.net;lr>

From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

16. **200 (OK) response (network watcher to S-CSCF) – see example in table A.3.5-16**

The network watcher forwards the 200 (OK) response to S-CSCF#1.

Table A.3.5-16: 200 (OK) response (network watcher to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      ps.home2.net;branch=z9hG4bK348923.1
P-Access-Network-Info:
From:
To:
Call-ID:
CSeq:
Content-Length:
```

17. **200 (OK) response (S-CSCF to PS) – see example in table A.3.5-17**

S-CSCF#2 forwards the 200 (OK) response to the PS.

Table A.3.5-17: 200 (OK) response (S-CSCF to PS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
From:
To:
Call-ID:
CSeq:
Content-Length:
```

3GPP TSG-CN1 Meeting #36 **Tdoc N1-041970**
Seoul, Korea, 15-19 November 2004

CR-Form-v7.1

CHANGE REQUEST

⌘ **24.141** **CR** **019** ⌘ **rev** **1** ⌘ Current version: **6.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	Clarifications to Ut
Source:	⌘	Ericsson
Work item code:	⌘	PRESNC
Date:	⌘	20/10/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	A number of errors for the normative text as well as the example flows has been discovered
Summary of change:	⌘	Various corrections to the normative text and flows for Ut is corrected.
Consequences if not approved:	⌘	Incorrect specification

Clauses affected:	⌘	6.1, 6.3.1.2, A.8.1, A.8.2, A.8.3, A.8.4			
Other specs affected:	⌘	Y	N		⌘
			X	Other core specifications	
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘				

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be

downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

6 Protocol for data manipulation at the Ut reference point

6.1 Introduction

~~Hypertext Transfer Protocol (HTTP) and~~ XML Configuration Access Protocol (XCAP) ~~are~~is used to store, alter and delete data related to the presence service. XCAP is designed according to the Hypertext Transfer Protocol (HTTP) framework, and uses the HTTP methods PUT, GET and DELETE for communication over the Ut reference point. The general information that can be manipulated is user groups, subscription authorization policy, resource lists, hard state presence publication, MIME objects referenced from the hard state presence information, etc. Soft state presence information manipulated with a PUBLISH request is not manipulated by the mechanism provided over the Ut reference point.

6.2 Functional entities

6.2.1 User Equipment (UE)

The UE implements the Data Manipulator (DM) role as described in subclause 6.3.1.

The UE shall implement HTTP digest AKA (see RFC 3310 [20]) and it shall initiate a bootstrapping procedure with the bootstrapping server function located in the home network, as described in 3GPP TS 24.109 [7].

The UE shall acquire the subscriber's certificate from PKI portal by using a bootstrapping procedure, as described in 3GPP TS 24.109 [7].

The UE shall implement HTTP digest authentication (see RFC 2617 [15A]).

The UE shall implement Transport Layer Security (TLS) (see RFC 2246 [13]). The UE shall be able to authenticate the network application function based on the received certificate during TLS handshaking phase.

6.2.2 Application Server (AS)

If an AS implements the role of a PS (see subclause 5.3.3) or of a RLS (see subclause 5.3.4), then the AS shall also implement the role of a Data Manipulation Server (DMS) (see subclause 6.3.2).

If there is no authentication proxy in the network, then the AS shall:

1) implement the role of a network application function, as described in 3GPP TS 24.109 [7];

2) implement TLS (see RFC 2246 [13]);

implement HTTP digest authentication (see RFC 2617 [15A]); and

4) support certificate authentication.

Editor's note: It needs to be clarified what physical entities can contain the Authentication Proxy and its relationship with the IMS architecture.

6.2.3 Authentication proxy

The authentication proxy shall implement the role of a network application function, as described in 3GPP TS 24.109 [7] and it shall support HTTP Digest Authentication (see RFC 2617 [15A] and certificate authentication.

The Authentication Proxy shall authenticate the UE and integrity protect the messages sent towards the UE.

Editor's note: It is FFS how the Authentication Proxy passes the user's identity to the Application Server (AS).

6.3 Roles

6.3.1 Data Manipulator (DM)

6.3.1.1 Introduction

The DM is a logical function that implements the requirements of a XCAP client as defined in draft-ietf-simple-xcap-03 [33]. The DM provides the means to manipulate the general data such us user groups, subscription authorization policy, resource lists, hard state presence publication, MIME objects referenced from the hard state presence information, etc.

NOTE: In order to be able to manipulate data stored on the DMS, the DM has the root directory on the DMS pre-configured or use some means to discover it. Discovery mechanisms are outside the scope of the present document.

6.3.1.2 Manipulating a resource ~~presence~~list

When the DM intends to manipulate a resource ~~presence~~list, it shall generate an HTTP PUT, HTTP GET or HTTP DELETE request in accordance with RFC 2616 [15], draft-ietf-simple-xcap-03 [33] and draft-ietf-simple-xcap-list-usage-02 [36].

6.3.1.3 Manipulating the subscription authorization policy

When the DM intends to manipulate the subscription authorization policy, it shall generate an HTTP PUT, HTTP GET or HTTP DELETE request in accordance with RFC 2616 [15], draft-ietf-simple-xcap-03 [33] and draft-ietf-simple-xcap-presence-rules-00 [35].

The DM may use an HTTP GET in accordance with RFC 2616 [15], draft-ietf-simple-xcap-03 [33] and draft-rosenberg-simple-common-policy-caps-01 [42] for fetching of the authorization policy capabilities which the DMS supports.

When the DM intends to authorize a different value of the same presence attribute to different watchers or watcher groups, the DM shall authorize a single tuple including one of the different values of the same presence attribute to every watcher or watcher groups by using a specific "inclusion set" as specified in draft-ietf-simple-xcap-presence-rules-00 [35].

6.3.1.4 Publishing hard state presence information

The DM shall implement draft-isomaki-simple-xcap-pidf-manipulation-usage-00 [34] in order to be able to manipulate hard state presence information. Hard state presence information uses the same format as soft state information, namely "application/pidf+xml" content type as described in draft-ietf-impp-cpim-pidf-08 [21] together with any of its extensions.

When the hard state presence information contains one ore more MIME objects to be aggregated with the "application/pidf+xml" content type and any of its extensions, the DM shall:

a) construct as many HTTP URIs as many objects to be stored and formulate every HTTP URI according a predefined directory structure;

NOTE: In order to be able to manipulate data stored on the DMS, the DM has the root directory on the DMS pre-configured or use some means to discover it. Discovery mechanisms are outside the scope of the present document.

b) store the objects on the data manipulation server behind the HTTP URI(s) created in the previous step using standard HTTP procedures as defined in RFC 2616 [15];

c) include every HTTP URI as a value of the corresponding XML element in the published "application/pidf+xml" presence document referencing the stored object(s) in the previous step; and

d) publish the hard state presence information according to draft-isomaki-simple-xcap-pidf-manipulation-usage-00 [34].

6.3.2 Data Manipulation Server (DMS)

6.3.2.1 Introduction

The Data Manipulation Server (DMS) is a logical function that implements the requirements of a XCAP server as defined in draft-ietf-simple-xcap-03 [33]. The DMS can store data such us user groups, subscription authorization policy, resource lists, hard state presence information, MIME objects referenced from the hard state presence information, etc.

6.3.2.2 Resource list manipulation acceptance

When the data manipulation server receives an HTTP PUT, HTTP GET or HTTP DELETE request for manipulating or fetching a resource list, the DMS shall first authenticate the request in accordance with 3GPP TS 24.109 [7] and then perform authorization. Afterwards the DMS shall perform the requested action and generate a response in accordance with RFC 2616 [15], draft-ietf-simple-xcap-03 [33] and draft-ietf-simple-xcap-list-usage-02 [36].

6.3.2.3 Subscription authorization policy manipulation acceptance

When the DMS receives an HTTP PUT, HTTP GET or HTTP DELETE request for manipulating or fetching of the subscription authorization policy, the data manipulation server shall first authenticate the request in accordance with 3GPP TS 24.109 [7] and then perform authorization. Afterwards the DMS shall perform the requested action and generate a response in accordance with RFC 2616 [15], draft-ietf-simple-xcap-03 [33] and draft-ietf-simple-xcap-presence-rules-00 [35].

When the DMS receives an HTTP GET request for fetching of the authorization policy capabilities information, the DMS shall generate a response in accordance with RFC 2616 [15], draft-ietf-simple-xcap-02 [33] and draft-rosenberg-simple-pres-policy-caps-00 [42].

6.3.2.4 Publication acceptance of hard state presence information

When the DMS receives an HTTP PUT, HTTP GET or HTTP DELETE request for publishing, fetching or deleting of hard state presence information, the DMS shall first authenticate the request in accordance with 3GPP TS 24.109 [7] and then perform authorization. Afterwards the DMS shall:

a) if the HTTP URI points to a predefined directory reserved for storing MIME objects and the request is an HTTP PUT request, replace any existing content referenced by the Request-URI with the content of the request;

b) if the Request-URI points to an uncreated directory, create the directory, store the content there and associate the content with the Request-URI. For all requests, i.e. HTTP PUT, HTTP GET and HTTP DELETE requests, generate an appropriate response in accordance with RFC 2616 [15]; or

c) if the HTTP URI points to an XCAP directory and the Application Usage ID (AUID) part of the HTTP URI is set to "pidf-manipulation", process the request and generate an appropriate response in accordance with draft-ietf-simple-xcap-03 [33], draft-isomaki-simple-xcap-pidf-manipulation-usage-00 [34] and RFC 2616 [15].

***** Next change *****

A.8 Example signalling flows of HTTP based presence service operation

A.8.1 Introduction

This subclause shows signalling flows relating to the manipulation of presence service data over the Ut reference point using XCAP. The flows only shows the signalling between the XCAP server and the XCAP client, thus possible proxies located in between the entities are not shown in the example signalling flows.

Each example signalling flow shows several sequences of manipulation of data for the presence service.

NOTE: Error conditions are not considered in the examples e.g. if authorization checks fail in the XCAP server, XML Schema compliancy checks fail or the file specified by the URI does not exist then an appropriate 4xx response is sent to the client.

~~Editor's note: Clarifications how XCAP is using HTTP is needed.~~

Clarifications how XCAP is using HTTP are described in in draft-ietf-simple-xcap-03 [33].

NOTE: The authentication proxy resides between UE (XCAP client) and AS (XCAP server), and examples of signalling flows for the authentication proxy are provided in 3GPP TS 24.109 [7].

A.8.2 Signalling flows demonstrating how DMs manipulate resource lists

~~Editor's note: The possible proxies (e.g., handling authentication matters) between the data manipulator client and HTTP server are bypassed. Also the authentication related headers are missing.~~

Figure A.8.2-1: DM manipulating a resource list on DMS

Figure A.8.2-1 shows a how a DM may manipulate a resource list on a DMS. The details of the signalling flows are as follows:

1. **XCAP PUT request (DM to DMS - see example in table A.8.2-1**

   The DM generates an XCAP PUT request to create a new resource list on the DMS. The resource list has one entry.

Table A.8.2-1: XCAP PUT request (DM to DMS)

```
PUT http://xcap.home1.net/services/resource-lists/users/user1/pf.xml HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 10:13:17 GMT
Content-Type: application/resource-lists+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
   <resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists">
     <list name="Presence_fellows" uri="sip:user1_list1@home1.net" subscribeable="true">
       <entry name="user2" uri="sip:user2_public1@home2.net">
         <display-name>User2</display-name>
       </entry>
     </list>
   </resource-lists>
```

2. **XCAP 201 (Created) response (DMS to DM) – see example in table A.8.2-2**

After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to create a file, the DMS sends an XCAP 201 (Created) response to the DM.

Table A.8.2-2: XCAP 201 (Created) response (DMS to DM)

```
HTTP/1.1 201 CREATED
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 10:50:35 GMT
Content-Type: text/html
Content-Length: 0
```

3. **XCAP PUT request (DM to DMS) – see example in table A.8.2-3**

The DM adds a new entry to the previously created resource list by generating a new XCAP PUT request.

Table A.8.2-3: XCAP PUT request (DM to DMS)

```
PUT http://xcap.home1.net/services/resource-lists/users/user1/pf.xml?resource-
lists/list[@name="Presence_fellows"]/entry HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 10:13:17 GMT
Content-Type: application/xml-fragment-body
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
         <entry name="user3" uri="sip:user3_public1@home3.net">
           <display-name>User3</display-name>
         </entry>
```

4. **XCAP 200 (OK) response (DMS to DM) - see example in table A.8.2-4**

After the DMS has performed the necessary authorization checks, XML document validations and XML schema compliancy checks the DMS sends an XCAP 201 (Created) response to the DM.

Table A.8.2-4: XCAP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 10:50:45 GMT
Content-Type: text/html
Content-Length: 0
```

5. **XCAP DELETE request (DM to DMS) - see example in table A.8.2-5**

The DM decides to delete the entry "user2" from the resource list. The DM generates an XCAP DELETE request.

Table A.8.2-5: XCAP DELETE request (DM to DMS)

```
DELETE http://xcap.home1.net/services/resource-lists/users/user1/pf.xml?resource-
lists/list[@name="Presence_fellows"]/entry[@name=user2"] HTTP/1.1
Host: oper.example.com:9999
User-Agent: IMS subscriber
Date: Thu, 08 Jan 2004 10:14:17 GMT
Referer: http://oper.home1.net:1234/service
```

6. **XCAP 200 (OK) response (DMS to DM) – see example in table A.8.2-6**

After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to delete an entry from the resource list, the DMS sends an XCAP 200 (OK) response.

Table A.8.2-6: XCAP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 11:00:47 GMT
Content-Type: image/jpeg
Content-Length: 0
```

7. **XCAP GET request (DM to DMS) – see example in table A.8.2-7**

The DM wishes to check the result of the previous transaction by generating an XCAP GET request.

Table A.8.2-7: XCAP GET request (DM to DMS)

```
GET http://xcap.home1.net/services/resource-lists/users/user1/pf.xml HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 11:13:17 GMT
Content-Length: 0
```

8. **XCAP 200 (OK) response (DMS to DM) - see example in table A.8.2-8**

After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to fetch the resource list, the DMS sends an XCAP 200 (OK) response to the DM including the resource list in the body of the response.

Table A.8.2-8: XCAP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Etag: "askdajdsaj"
Date: Thu, 08 Jan 2004 11:50:35 GMT
Content-Type:application/resource-lists+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
   <resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists">
     <list name="Presence_fellows" uri="sip:user1_list1@home1.net" subscribeable="true">
       <entry name="user3" uri="sip:user3_public1@home3.net">
         <display-name>User3</display-name>
       </entry>
     </list>
   </resource-lists>
```

A.8.3 Signalling flows demonstrating how DMs manipulate presence authorization policy

~~Editor's note: The possible proxies (e.g., handling authentication matters) between the data manipulator client and HTTP server are bypassed. Also the authentication related headers are missing.~~

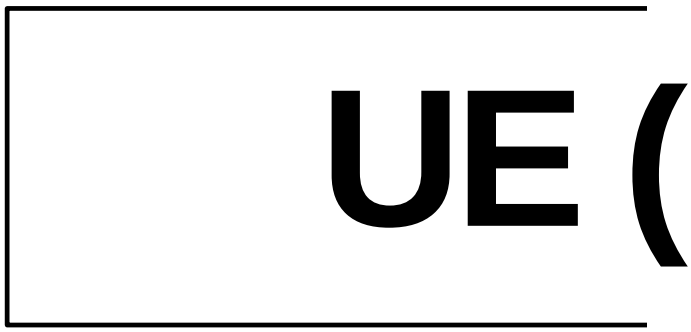


Figure A.8.3-1: DM manipulating presence authorization policy on DMS

Figure A.8.3-1 shows a DM manipulating presence authorization policy on a DMS. The details of the signalling flows are as follows:

1. **XCAP PUT request (DM to DMS) – see example in table A.8.3-1**

   The DM generates an XCAP PUT request to create a presence authorization policy on the DMS. The presence authorization policy has one permission statement allowing for sip:user2_public1@home2.net to see all information from the basic PIDF namespace along with the "video" element from the prescaps namespace.

Table A.8.3-1: XCAP PUT request (DM to DMS)

```
PUT http://xcap.home1.net/services/permission-statements/users/user1/ps.xml HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 10:13:17 GMT
Content-Type:application/permission-statements+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
   <permission-statements xmlns="urn:ietf:params:xml:ns:permission-statements"
     xmlns:pidf="urn:ietf:params:xml:ns:pidf"
     xmlns:prescaps="urn:ietf:params:xml:ns:simple-prescaps-ext">
     <statement id="dsafa43232">
       <applies-to>
         <uri>sip:user2_public1@home2.net</uri>
       </applies-to>
```

```
       <permissions>
         <accept/>
         <show-namespace>urn:ietf:params:xml:ns:pidf</show-namespace>
         <show-element>prescaps:video</show-element>
       </permissions>

     </statement>
   </permission-statements>
```

2. **XCAP 201 (Created) response (DMS to DM) - see example in table A.8.3-2**

After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to create a file, the DMS sends an XCAP 201 (Created) response to the DM.

Table A.8.3-2: XCAP 201 (Created) response (DMS to DM)

```
HTTP/1.1 201 CREATED
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 10:50:35 GMT
Content-Type: text/html
Content-Length: 0
```

3. **XCAP PUT request (DM to DMS) – see example in table A.8.3-3**

The DM adds a new permission-statement to the previously created presence authorization policy by generating a new XCAP request. The new permission statement allows the user named sip:user3_public1@home3.net to see the tuple with class element specifying "sip".

Table A.8.3-3: XCAP PUT request (DM to DMS)

```
PUT http://xcap.home1.net/services/permission-statements/users/user1/ps.xml/permission-
      statements HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 10:13:27 GMT
Content-Type: application/xml-fragment-body
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
    <statement id="dsffdsfrrr32423">
       <applies-to>
         <uri>sip:user3_public1@home3.net</uri>
       </applies-to>

       <permissions>
         <accept/>
         <show-tuple>sip</show-tuple>
       </permissions>

     </statement>
```

4. **XCAP 200 (OK) response (DMS to DM) - see example in table A.8.3-4**

After the DMS has performed the necessary authorization checks, XML document validations and XML schema compliancy checks the DMS sends an XCAP 201 (Created) response to the DM.

Table A.8.3-4: XCAP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 10:50:45 GMT
Content-Type: text/html
Content-Length: 0
```

5. **XCAP DELETE request (DM to DMS) - see example in table A.8.3-5**

The DM decides to delete the permission-statement for sip:user2_public1@home2.net from the authorization policy. The DM generates an XCAP DELETE request.

Table A.8.3-5: XCAP DELETE request (DM to DMS)

```
DELETE http://xcap.home1.net/services/presence-lists/users/user1/ps.xml/permission-
      statements/statement[@id="dsafa43232"]/permissions/show-namespace HTTP/1.1
Host: oper.example.com:9999
User-Agent: IMS subscriber
Date: Thu, 08 Jan 2004 10:14:17 GMT
Referer: http://oper.home1.net:1234/service
```

6. **XCAP 200 (OK) response (DMS to DM) – see example in table A.8.3-6**

After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to delete an entry from the resource list, the DMS sends an XCAP 200 (OK) response.

Table A.8.3-6: XCAP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 11:00:47 GMT
Content-Length: 0
```

7. **XCAP GET request (DM to DMS) – see example in table A.8.3-7**

The DM wishes to check the result of the previous transaction by generating an XCAP GET request.

Table A.8.3-7: XCAP GET request (DM to DMS)

```
GET http://xcap.home1.net/services/permission-statements/users/user1/ps.xml HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 11:13:17 GMT
Content-Length: 0
```

8. **XCAP 200 (OK) response (DMS to DM) – see example in table A.8.3-8**

After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to fetch the resource list, the DMS sends an XCAP 200 (OK) response to the DM including the resource list in the body of the response.

Table A.8.3-8: XCAP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Etag: "eiuuekksks"
Date: Thu, 08 Jan 2004 11:50:35 GMT
Content-Type:application/permission-statements+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
   <permission-statements xmlns="urn:ietf:params:xml:ns:permission-statements"
     xmlns:pidf="urn:ietf:params:xml:ns:pidf"
     xmlns:prescaps=" urn:ietf:params:xml:ns:simple-prescaps-ext">
     <statement id="dsffdsfrrr32423">
       <applies-to>
         <uri>sip:user3_public1@home3.net</uri>
       </applies-to>
       <permissions>
         <accept/>
         <show-tuple>sip</show-tuple>
       </permissions>
     </statement>
   </permission-statements>
```

A.8.4 Storing external content (successful operation)

~~Editor's note: The possible proxies (e.g., handling authentication matters) between the data manipulator client and HTTP server are bypassed. Also the authentication related headers are missing.~~

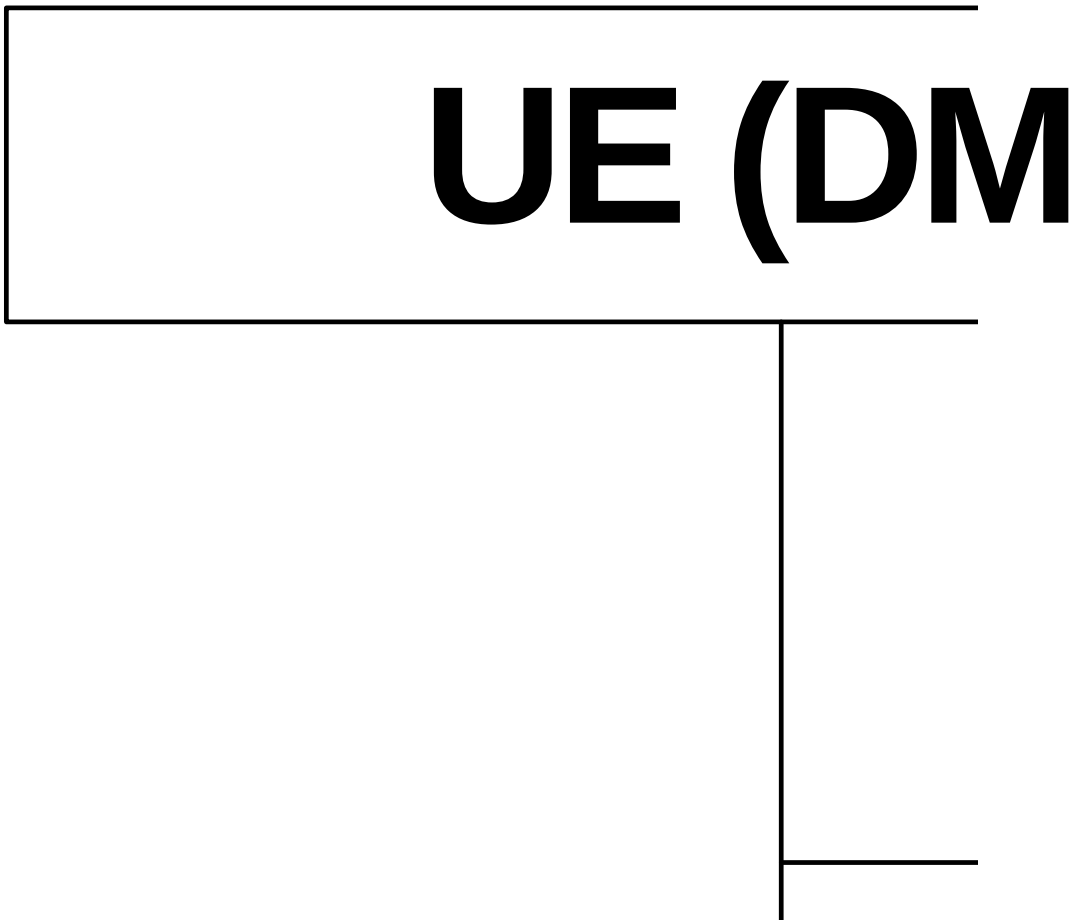


Figure A.8.4.-1: DM manipulating hard-state presence document on DMS

Figure A.8.4-1 shows a DM manipulating hard-state presence document on a DMS when the presence document has an aggregated storing MIME object with the "application/pidf+xml" content type and any of its extensions. The details of the signalling flows are as follows:

1. **HTTP PUT request (DM (client) to DMS) – see example in table A.8.2-1**

   In order to store the content, the DM generates an HTTP PUT request containing the MIME object in the body of the request. The request-URI points to the directory where the content is stored and shows the name of the file to be created.

Table A.8.4-1: HTTP PUT request (DM to DMS)

```
PUT http://operator.example.com/services/users/bill/pictureX HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 10:13:17 GMT
Content-Type: image/jpeg
Content-Length: (…)

{pictureX.jpg}
```

2. **HTTP 201 (Created) response (DMS to DM) – see example in table A.8.4-2**

   After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to create a file the HTTP server sends an HTTP 201 (Created) response to the client.

Table A.8.4-2: HTTP 201 (Created) response (DMS to DM)

```
HTTP/1.1 201 CREATED
Server: Apache/1.3.22 (Unix) mod_perl/1.27Content-Type: text/html
Date: Thu, 08 Jan 2004 10:50:35 GMT
Content-Length: 1234
```

3. **XCAP PUT request (DM to DMS) - see example in table A.8.2-3**

   The DM generates an XCAP PUT request in order to store XML encoded presence document which includes a URI reference to the MIME object stored on the DMS. The AUID part of the request URI is 'pidf-manipulation' as defined in draft-isomaki-simple-xcap-pidf-manipulation-usage-00 [34].

Table A.8.4-3: XCAP PUT request (DM to DMS)

```
PUT http://xcap.example.com/services/pidf-manipulation/users/bill/pidf.xml HTTP/1.1
User-Agent: IMS subscriber
Referer: http://xcap.home1.net:1234/service
Date: Thu, 08 Jan 2004 10:13:27 GMT
Content-Type: application/pidf+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
   <presence xmlns="urn:ietf:params:xml:ns:cpim-pidf"
          xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
          xmlns:ext="urn:ietf:params:xml:ns:ext-cont"
          entity="sip:bill@example.com">

     <tuple id="123sd">
       <status>
         <basic>open</basic>
       </status>
       <et:type>service</et:type>
         <contact>sip:bill@example.com</contact>
     </tuple>

     <tuple id="432sd">
       <status>
         <basic>open</basic>
       </status>
       <et:type>presentity</et:type>
       <ext:photo>
         http://operator.example.com/services/users/bill/pictureX.jpg
       </ext:photo>
```

```
        <note xml:lang="en">At home</note>
      </tuple>
    </presence>
```

4. **XCAP 201 (CREATED) response (DMS to DM) - see example in table A.8.4-4**

After the DMS has performed the necessary authorization checks, XML document validations and XML schema compliancy checks the DMS sends an XCAP 201 (Created) response to the DM.

Table A.8.4-4: XCAP 201 (Created) response (DMS to DM)

```
HTTP/1.1 201 CREATED
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 10:50:45 GMT
Content-Length: 0
```

5. **HTTP GET request (DM to DMS) – see example in table A.8.4-5**

The DM wishes to fetch the MIME object from the DMS. The client generates an HTTP GET request. The request URI points to the directory where the object is stored and indicates the name of the file to be fetched.

Table A.8.4-5: HTTP GET request (DM to DMS)

```
GET http://operator.example.com/services/users/bill/pictureX HTTP/1.1
Host: oper.example.com:9999
User-Agent: IMS subscriber
Date: Thu, 08 Jan 2004 10:43:17 GMT
Accept: image/jpeg
Referer: http://oper.home1.net:1234/service
```

6. **HTTP 200 (OK) response (DMS to DM) – see example in table A.8.4-6**

After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to fetch the file the DMS sends an HTTP 200 (OK) response having the object in the body to the DM.

Table A.8.4-6: HTTP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 11:00:47 GMT
Content-Type: image/jpeg
Content-Length: (…)

{pictureX}
```

7. **HTTP PUT request (DM to DMS) – see example in table A.8.4-7**

The DM wishes to modify the earlier stored MIME object by replacing the picture X with a new picture X with new content. To modify the object the DM generates an HTTP PUT request using the same request URI as has been used for the modified (old) object. The new object is conveyed in the body of the request.

Table A.8.4-7: HTTP PUT request (DM to DMS)

```
PUT http://operator.example.com/services/users/bill/pictureX HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 11:13:17 GMT
Content-Type: image/jpeg
Content-Length: (…)

{pictureX.jpg}
```

8. **HTTP 200 (OK) response (DMS to DM) – see example in table A.8.4-8**

After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to replace the existing MIME object with the new one the DMS sends an HTTP 200 (OK) response to the DM.

Table A.8.4-8: HTTP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 11:50:35 GMT
Content-Length: 0
```

9. **XCAP PUT request (DM to DMS) – see example in table A.8.4-9**

The DM wishes to remove the MIME object from his presence information. The DM generates an XCAP PUT request to modify the XML encoded presence document to remove the reference to the MIME object from the presence document. The request URI contains a node selector to the requested tuple according to draft-ietf-simple-xcap-02 [33]. Because the signalling flow does not contain the XCAP GET request the use of the If-Match header is omitted in this example.

Table A.8.4-9: XCAP PUT request (DM to DMS)

```
PUT http://xcap.example.com/services/pidf-
  manipulation/users/bill/pidf.xml?presence/tuple[@id='432sd'] HTTP/1.1
Date: Thu, 08 Jan 2004 11:13:37 GMT
Content-Type: text/plain
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
   <tuple id="432sd">
     <status>
       <basic>open</basic>
     </status>
     <et:type>presentity</et:type>
     <note xml:lang="en">At home</note>
   </tuple>
```

10. **XCAP 200 (OK) response (DMS to DM) - see example in table A.8.4-10**

After the DMS has performed the necessary authorization checks, XML document validations and XML Schema compliancy checks the DMS sends an XCAP 200 (OK) response to the DM.

Table A.8.4-10: XCAP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 11:50:59 GMT
Content-Length: 0
```

11. **HTTP DELETE request (DM to DMS) – see example in table A.8.4-11**

The DM removes the MIME object from the DMS by generating an HTTP DELETE request.

Table A.8.4-11: HTTP DELETE request (DM to DMS)

```
DELETE http://operator.example.com/services/users/bill/pictureX HTTP/1.1
Host: oper.example.com:9999
User-Agent: IMS subscriber
Date: Thu, 08 Jan 2004 11:52:00 GMT
Referer: http://oper.home1.net:1234/service
```

12. **HTTP 200 (OK) response (DMS to DM) – see example in table A.8.4-12**

After the DMS has performed the necessary authorization checks on the originator to ensure that the DM is allowed to delete the object, the DMS sends an HTTP 200 (OK) response to the DM.

Table A.8.4-12: HTTP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 11:52:35 GMT
Content-Length: 0
```

3GPP TSG-CN1 Meeting #36
Seoul, Korea, 15-19 November 2004

Tdoc N1-041971

CR-Form-v7.1

CHANGE REQUEST

⌘ **24.141** **CR** **22** ⌘ **rev** **1** ⌘ Current version: **6.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	Correction of XCAP change flow
Source:	⌘	Siemens
Work item code:	⌘	PRESNC
Date:	⌘	16/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
Ph2 *(GSM Phase 2)*
R96 *(Release 1996)*
R97 *(Release 1997)*
R98 *(Release 1998)*
R99 *(Release 1999)*
Rel-4 *(Release 4)*
Rel-5 *(Release 5)*
Rel-6 *(Release 6)*
Rel-7 *(Release 7)*

Reason for change:	⌘	The concept of the XCAP change procedure has changed with draft-ietf-simple-xcap-package-02. The draft just describes the XML change document while the package that is used for the subscription is defined in draft-ietf-sipping-config-framework-03
Summary of change:	⌘	Use of the Event header value “sip-profile”. Introduce the XCAP diff element according to draft-ietf-simple-xcap-package-02 in NOTIFY requests
Consequences if not approved:	⌘	Interoperability problems

Clauses affected:	⌘	A.3.6

Other specs affected: ⌘	Y	N	
		X	Other core specifications
		X	Test specifications
		X	O&M Specifications

⌘

Other comments: ⌘

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

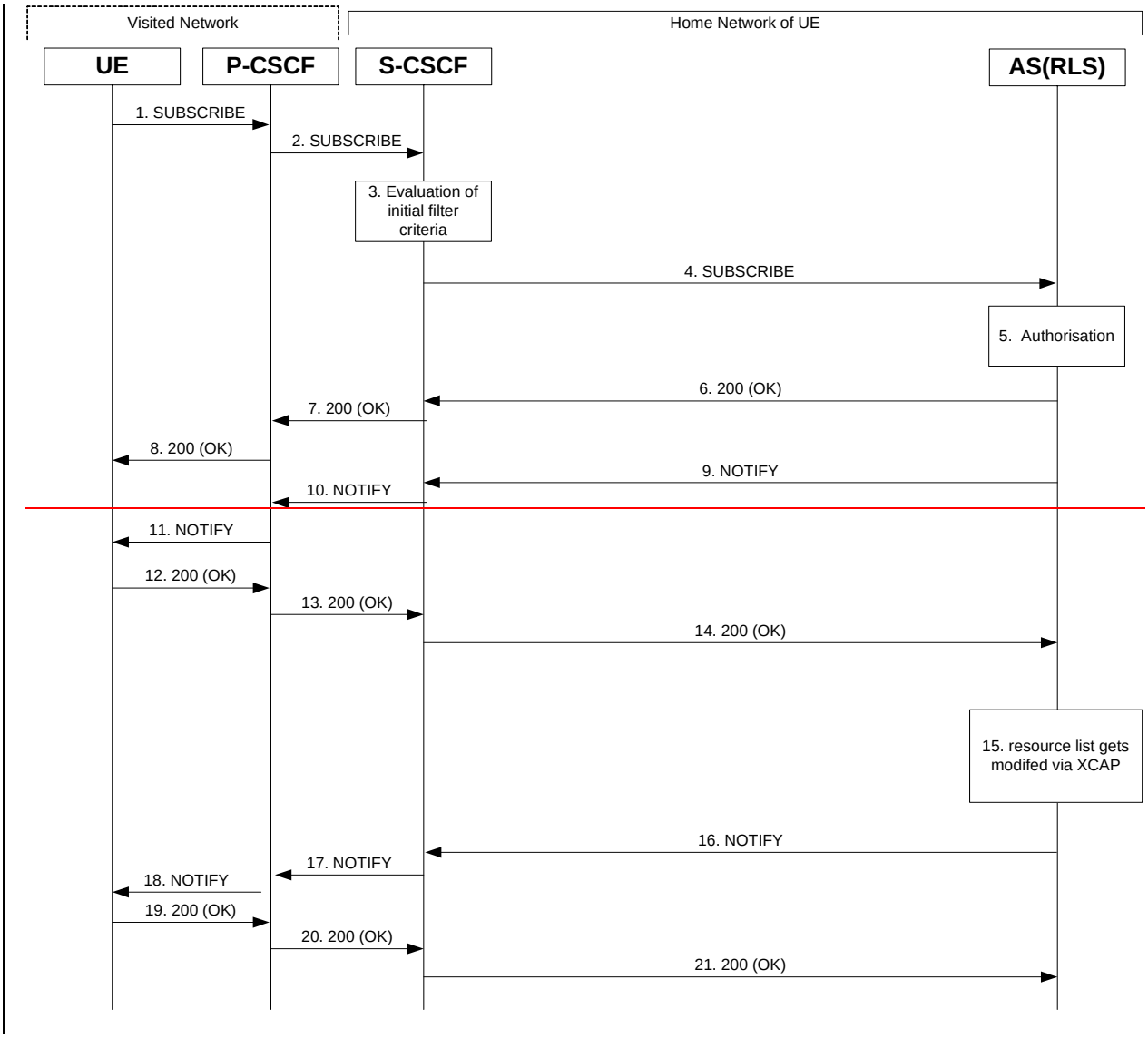
1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

A.3.6 Watcher subscribing to XCAP change, UE in visited network

A.3.6.1 Watcher subscribing to XCAP change in his resource list, UE in visited network - Successful subscription



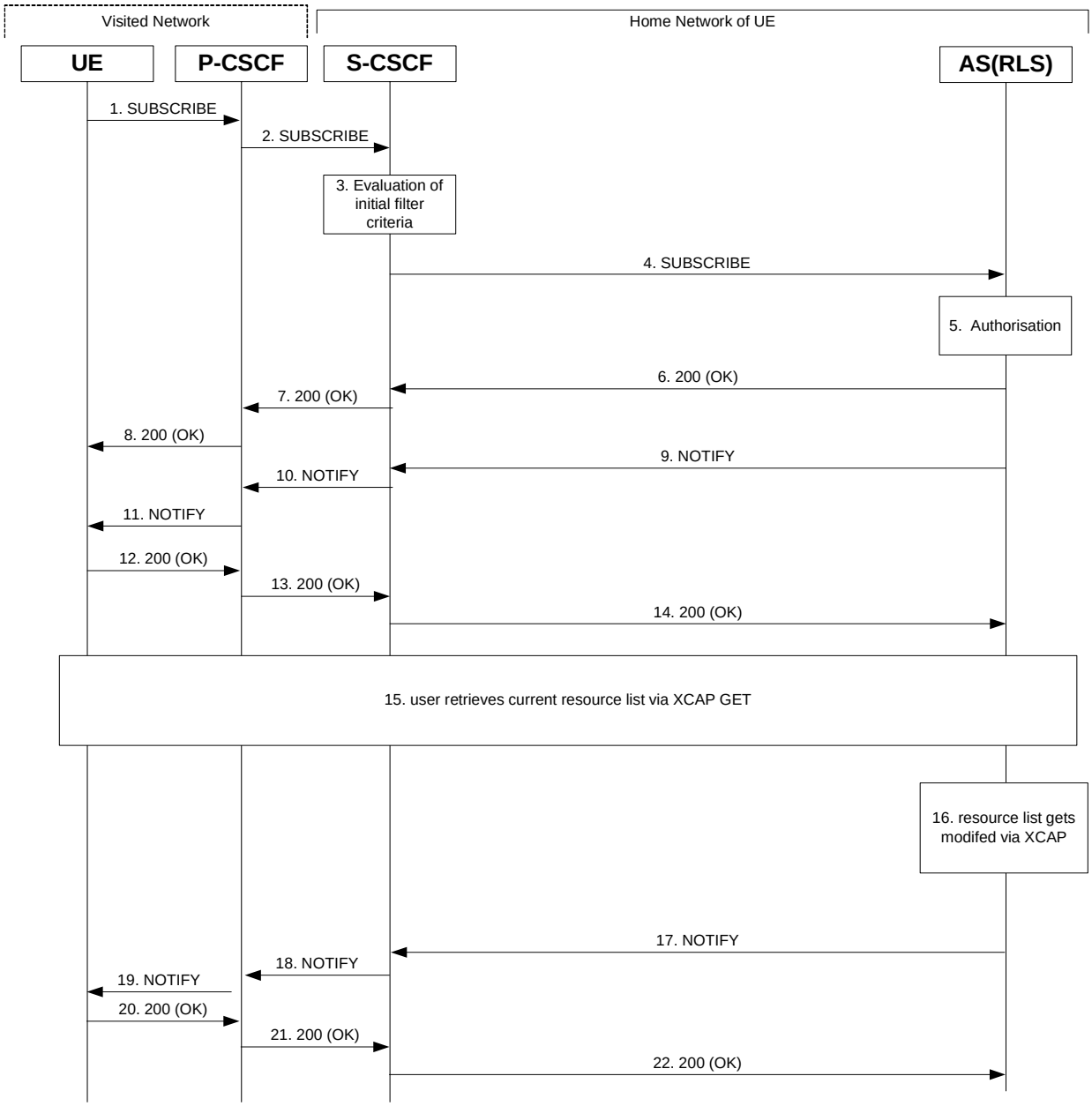


Figure A.3.6.1-1: Watcher subscribing to XCAP change in his resource list

Figure A.3.6.1-1 shows a watcher subscribing to XCAP change event notification. The details of the flows as follows:

1. **SUBSCRIBE request (UE to P-CSCF) – see example in table A.3.6.1-1**

   A watcher agent in a UE wishes to get notification when his resource list gets modified via XCAP. In order to initiate a subscription to XCAP changes in RLS, the UE generates a SUBSCRIBE request indicating support for "xcap-change", together with an indication of the length of time this periodic subscription should last.

Table A.3.6.1-1: SUBSCRIBE request (UE to P-CSCF)

```
SUBSCRIBE sip:user1_public1@home1.net SIP/2.0
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
Max-Forwards: 70
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
Route: <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>, <sip:orig@scscf1.home1.net;lr>
P-Preferred-Identity: <sip:user1_public1@home1.net>
Privacy: none
From: <sip:user1_public1@home1.net>;tag=31415
To: <sip:user1_public1@home1.net>
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 123 SUBSCRIBE
Require: sec-agree
Proxy-Require: sec-agree
Security-Verify: ipsec-3gpp; q=0.1; alg=hmac-sha-1-96; spi-c=98765432; spi-s=87654321; port-
      c=8642; port-s=7531
Event: sip-profilexcap-change
Expires: 7200
Accept: application/xcap-changediff+xml
Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>
Content-Length: 0
```

Request-URI: The users own SIP URI to get notifications of changes on all lists owned by the user.

Event: This field is populated with the value "sip-profilexcap-change " to specify the use of the sip-profilexcap-change package to get notified of changes to XCAP documents.

Accept: This field is populated with the value application/xcap-changediff+xml ' indicating that the UE supports the xcap-change diff MIME type.

To: Same as the Request-URI.

2. **SUBSCRIBE request (P-CSCF to S-CSCF) - see example in table A.3.6.1-2**

The P-CSCF looks up the serving network information for the public user identity that was stored during the registration procedure. The SUBSCRIBE request is forwarded to S-CSCF#1. A Route header is inserted into SUBSCRIBE request. The information for the Route header is taken from the service route determined during registration.

Table A.3.6.1-2: SUBSCRIBE request (P-CSCF to S-CSCF)

```
SUBSCRIBE sip:user1_public1@home1.net SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
P-Access-Network-Info:
Route: <sip:orig@scscf1.home1.net;lr>
Max-Forwards: 69
P-Asserted-Identity: <sip:user1_public1@home1.net>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"
Privacy:
Record-Route: <sip:pcscf1.visited1.net;lr>
Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Supported:
Expires:
Accept:
Contact:
Content-Length:
```

3. **Evaluation of initial filter criteria**

The S-CSCF validates the service profile of this subscriber and evaluates the initial filter criteria. For sip:user1_public1@home1.net the S-CSCF has originating initial Filter Criteria with Service Point Trigger of Method = SUBSCRIBE AND Event = "xcap-change" that informs the S-CSCF to route the SUBSCRIBE request to the Application Server sip:rls.home1.net.

4. **SUBSCRIBE request (S-CSCF to RLS) - see example in table A.3.6.1-4**

The S-CSCF forwards the SUBSCRIBE request to the RLS.

Table A.3.6.1-4 SUBSCRIBE request (S-CSCF to RLS)

```
SUBSCRIBE sip:user1_public1@home1.net SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK344a65.1, SIP/2.0/UDP
      pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
Max-Forwards: 68
P-Access-Network-Info:
P-Asserted-Identity: <sip:user1_public1@home1.net>, <tel:+1-212-555-1111>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Privacy:
Record-Route: <sip:orig@scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
Route: <sip:rls.home1.net;lr>, <sip:orig@scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Supported:
Expires:
Accept:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF populates the P-Charging-Function-Addresses header field to be passed to the RLS.

5. **Authorization**

The RLS performs the necessary authorization checks on the originator to ensure that he/she is authorized to subscribe to xcap-change. In this example this condition has been met, so the RLS sends a 200 (OK) response to the S-CSCF.

6. **200 (OK) response (RLS to S-CSCF) - see example in table A.3.6.1-6**

The RLS sends the response to the S-CSCF.

Table A.3.6.1-6: 200 (OK) response (RLS to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK344a65.1, SIP/2.0/UDP
      pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"; orig-ioi=home1.net;
      term-ioi=home1.net
Record-Route:
From:
To: <sip:user1_public1@home1.net>;tag=151170
Call-ID:
CSeq:
Expires:
Contact:
Content-Length: 0
```

7. **200 (OK) response (S-CSCF to P-CSCF) - see example in table A.3.6.1-7**

The S-CSCF forwards the response to the P-CSCF.

Table A.3.6.1-7: 200 (OK) response (S-CSCF to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

8. **200 (OK) response (P-CSCF to UE) - see example in table A.3.6.1-8**

The P-CSCF forwards the response to the watcher agent in the UE.

Table A.3.6.1-8: 200 (OK) response (P-CSCF to UE)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
Record-Route: <sip:orig@scscf1.home1.net;lr>, <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

9. **NOTIFY request (RLS to S-CSCF) – see example in table A.3.6.1-9**

The RLS generates a NOTIFY request including the xcap-change document as a result of the SUBSCRIBE request. As this is the initial NOTIFY it contains only the URI and the new-etag element.

Table A.3.6.1-9 NOTIFY request (RLS to S-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 70
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
From: <sip:user1_@home1.net>;tag=151170
To: <sip:user1_public1@home1.net>;tag=31415
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 89 NOTIFY
Subscription-State: active;expires=7200
Event: xcap-change
Contact: <sip:rls.home1.net>
Content-Type: application/xcap-change+xml;charset="UTF-8"
Content-Length:

<?xml version="1.0" encoding="UTF-8"?>
   <xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff"documents
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xcap-root="
      http://xcap.home1.net/services/">
      <document>
         uridoc-selector="http://xcap.home1.net/services/"resource-lists/users/user1/pf.xml"
         new-etag="asdnasd9asd8asd7"
         previous-etag="asdnasd9asd8asd7"
      </document>
   </xcap-diffdocuments>
```

The content of the document element contains a new-etag and a previous etag element with identical value and no list of instructions. This way it is indicated that this is the reference XML diff document. This documents has only the information about the etags and the document URI's covered by that subscription.

10. **NOTIFY request (S-CSCF to P-CSCF) - see example in table A.3.6.1-10**

The S-CSCF forwards the NOTIFY request to the P-CSCF.

Table A.3.6.1-10: NOTIFY request (S-CSCF to P-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 69
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"
P-Charging-Function-Addresses:
Route: <sip:pcscf1.visited1.net;lr>
Record-Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Length:

(…)
```

11. **NOTIFY request (P-CSCF to UE) - see example in table A.3.6.1-11**

The P-CSCF forwards the NOTIFY request to the watcher application in the UE.

Table A.3.6.1-11: NOTIFY request (P-CSCF to UE)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 68
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Length:

(…)
```

12. **200 (OK) response (UE to P-CSCF) - see example in table A.3.6.1-12**

The UE acknowledges the NOTIFY request with a 200 (OK) response to the P-CSCF.

Table A.3.6.1-12: 200 (OK) response (UE to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

13. **200 (OK) response (P-CSCF to S-CSCF) – see example in table A.3.6.1-13**

The P-CSCF forwards the 200 (OK) response to the S-CSCF.

Table A.3.6.1-13: 200 (OK) response (P-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"
From:
To:
Call-ID:
CSeq:
Content-Length:
```

14. **200 (OK) response (S-CSCF to RLS) - see example in table A.3.6.1-14**

The S-CSCF#2 forwards the response to the RLS in the home network of the UE.

Table A.3.6.1-14: 200 (OK) response (S-CSCF to RLS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"; orig-ioi=home1.net;
      term-ioi=home1.net
From:
To:
Call-ID:
CSeq:
Content-Length:
```

15. ~~**Resource List gets modified via XCAP**~~**User retrieves current resource list via XCAP get**

~~The resource list of user1 gets modified via XCAP procedures~~as user1 does not have a local copy of the resource list identified by the etag he retrieves the corresponding list via XCAP get.

16. **Resource List gets modified via XCAP**

The resource list of user1 gets modified via XCAP procedures.

~~16~~17. **NOTIFY request (RLS to S-CSCF) - see example in table A.3.6.1-~~16~~17**

In this example it is assumed that the RLS has received a XCAP request to delete user2_public@home1.net from the resource list of user1.

Table A.3.6.1-~~16~~ 17 NOTIFY request (RLS to S-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 70
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
From: <sip:user1_public1@home1.net>;tag=151170
To: <sip:user1_public1.home1.net>;tag=31415
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 90 NOTIFY
Subscription-State: active;expires=5000
Event: xcap-changesip-profile
Contact: <sip:rls.home1.net>
Content-Type: application/xcap-changediff+xml;charset="UTF-8"
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
   <xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff"documents
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xcap-root="
      http://xcap.home1.net/services/">
     <document
        doc-selectoruri=" http://xcap.home1.net/services/resource-lists/users/user1/pf.xml–"
        new-etag="asdnasd8asd7asd6"
        previous-etag="asdnasd9asd8asd7"
        hash="<hash value>">
       <remove-elementchange>
          method="DELETE"
          uri="http://xcap.home1.net/services/resource-lists/users/user1/pf.xml?resource-
      lists/entryresource[@name="user2_public@home1.net"]/uri"</remove-element>
     </document>
   </xcap-diffdocuments>
```

Content-Type: Set to application/xcap-~~change~~diff+xml.

The message body in the NOTIFY request contains information of the new-etag of the changed document, the change method and the element that was changed in accordance with draft-ietf-simple-xcap-package-~~01~~ 02 [39].

~~17~~18. **NOTIFY request (S-CSCF to P-CSCF) - see example in table A.3.6.1-~~17~~18**

The S-CSCF forwards the NOTIFY request to the P-CSCF.

Table A.3.6.1-~~17~~18: NOTIFY request (S-CSCF to P-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 69
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"
P-Charging-Function-Addresses:
Route: <sip:pcscf1.visited1.net;lr>
Record-Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content Type:
Content-Length:

(…)
```

~~18~~19. **NOTIFY request (P-CSCF to UE) – see example in table A.3.6.1-~~18~~19**

The P-CSCF forwards the NOTIFY request to the watcher application in the UE.

Table A.3.6.1-~~18~~19: NOTIFY request (P-CSCF to UE)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 68
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Require:
Content-Type:
Content-Length:

(…)
```

~~19~~20. **200 (OK) response (UE to P-CSCF) – see example in table A.3.6.1-~~19~~20**

The UE acknowledges the NOTIFY request with a 200 (OK) response to the P-CSCF.

Table A.3.6.1-~~19~~20: 200 (OK) response (UE to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

~~20~~21. **200 (OK) response (P-CSCF to S-CSCF) – see example in table A.3.6.1-~~20~~21**

The P-CSCF forwards the 200 (OK) response to the S-CSCF.

Table A.3.6.1-~~20~~21: 200 (OK) response (P-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"
From:
To:
Call-ID:
CSeq:
Content-Length:
```

~~21~~22. **200 (OK) response (S-CSCF to RLS) – see example in table A.3.6.1-~~21~~22**

The S-CSCF#2 forwards the response to the RLS in the home network of the UE.

Table A.3.6.1-~~21~~22: 200 (OK) response (S-CSCF to RLS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"; orig-ioi=home1.net;
      term-ioi=home1.net
From:
To:
Call-ID:
CSeq:
Content-Length:
```

3GPP TSG-CN1 Meeting #36
Seoul, Korea, 15-19 November 2004

Tdoc N1-041972

CR-Form-v7.1

CHANGE REQUEST

⌘ **24.141** **CR** **24** ⌘ **rev** **1** ⌘ Current version: **6.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	Aligning Presence data model with IETF
Source:	⌘	Nokia
Work item code:	⌘	PRESNC
Date:	⌘	16/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	Clarfies and enhances interoperability between publishers and presence information subscribers. Alignment with IETF solution.
Summary of change:	⌘	This CR proposes adding a new reference to IETF SIMPLE provided Presence Data Model draft. The draft contains basis how presence information should be structured, provides further semantics to presence attributes, provides two XML Schemas for presenting device and person specific information, and gives quidelines how the composer policy might be implemented in the first phase etc. The CR also aligns the normative texts of TS according to the Data Model.
Consequences if not approved:	⌘	Presence data model is not in-line with IETF provided quidelines. Also, the current specification is not specific enough, and thus may cause interoperability problems between publishers and watchers in understanding what presence information means.

Clauses affected:	⌘	2; 5.3.1; 5.3.2; 6.3.1.3

Other specs affected: ⌘	Y	N		
		X	Other core specifications	⌘
		X	Test specifications	
		X	O&M Specifications	

Other comments: ⌘

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

*** 1st change ***

2 References

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

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

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

[2] 3GPP TS 22.141: "Presence Service; Stage 1".

[3] 3GPP TS 23.002: "Network architecture".

[4] 3GPP TS 23.141: "Presence service; Architecture and functional description; Stage 2".

[5] 3GPP TS 23.218: "IP Multimedia (IM) session handling; IM call model; Stage 2".

[6] 3GPP TS 23.228: "IP Multimedia Subsystem (IMS); Stage 2".

[7] 3GPP TS 24.109: "Bootstrapping interface (Ub) and Network application function interface (Ua); Protocol details".

[8] 3GPP TS 24.228: "Signalling flows for the IP multimedia call control based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3".

[9] 3GPP TS 24.229: "Internet Protocol (IP) multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3".

[10] 3GPP TS 29.228: "IP Multimedia (IM) Subsystem Cx and Dx Interfaces; Signalling flows and message contents".

[11] 3GPP TS 33.141: "Presence service; Security".

[12] IETF RFC 2664 (1999): "FYI on Questions and Answers - Answers to Commonly asked New Internet User Questions".

[13] IETF RFC 2246 (1999): "The TLS Protocol Version 1.0".

[14] IETF RFC 2387 (August 1998): "The MIME Multipart/Related Content-type".

[15] IETF RFC 2616 (June 1999): "Hypertext Transfer Protocol -- HTTP/1.1".

[15A] IETF RFC 2617 (June 1999): " HTTP Authentication: Basic and Digest Access Authentication".

[16] IETF RFC 2778 (2000): "A Model for Presence and Instant Messaging".

[17] IETF RFC 3261 (June 2002): "SIP: Session Initiation Protocol".

[18] IETF RFC 3263 (June 2002): "Session Initiation Protocol (SIP): Locating SIP Servers".

[19] IETF RFC 3265 (March 2002): "Session Initiation Protocol (SIP)-Specific Event Notification".

[20] IETF RFC 3310 (2002): "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)".

[21] draft-ietf-impp-cpim-pidf-08 (May 2003): "Presence Information Data Format (PIDF)".

[22] draft-ietf-simple-event-list-04 (June 2003): "A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[23] draft-ietf-sip-publish-03 (February 2004): "An Event State Publication Extension to the Session Initiation Protocol (SIP)".

[24] draft-ietf-simple-partial-notify-01 (January 2004): "Partial Notification of Presence Information".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[25] draft-ietf-simple-prescaps-ext-00 (February 2004): "Device capability PIDF status extension".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[26] draft-ietf-simple-rpid-03 (March 2004): "RPID: Rich Presence: Extensions to the Presence Information Data Format (PIDF)".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[27] draft-ietf-simple-presence-10 (January 2003): "A Presence Event Package for the Session Initiation Protocol (SIP)".

[28] draft-ietf-simple-winfo-package-05 (January 2003): "A Session Initiation Protocol (SIP) Event Template-Package for Watcher Information".

[29] draft-ietf-simple-winfo-format-04 (January 2003): "An Extensible Markup Language (XML) Based Format for Watcher Information".

[30] draft-ietf-simple-filter-format-00 (February 2004): "An Extensible Markup Language (XML) Based Format for Event Notification Filtering".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[31] draft-ietf-simple-event-filter-funct-00 (February 2003): "Functional Description of Event Notification Filtering".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[32] draft-ietf-simple-cipid-01 (March 2004): "CIPID: Contact Information in Presence Information Data Format".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[33] draft-ietf-simple-xcap-03 (July 2004): "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[34] draft-isomaki-simple-xcap-pidf-manipulation-usage-00 (February 2004): "An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usage for Manipulating Presence Document Contents".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[35] draft-ietf-simple-xcap-presence-rules-00 (May 2004): "Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usages for Setting Presence Authorization".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[36] draft-ietf-simple-xcap-list-usage-02 (February 2004): "An Extensible Markup Language (XML) Format for Representing Resource Lists".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[37] draft-ietf-geopriv-pidf-lo-01 (February 2004): "A Presence-based GEOPRIV Location Object Format".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[38] draft-ietf-simple-partial-pidf-format-00 (January 2004): "Presence Information Data Format (PIDF) Extension for Partial Presence".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[39] draft-ietf-simple-xcap-package-02 (July 2004): "A Session Initiation Protocol (SIP) Event Package for Modification Events for the Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Managed Documents".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[40] draft-ietf-sip-content-indirect-mech-03 (June 2003): "A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[41] draft-rosenberg-simple-common-policy-caps-01 (July 2004): "An Extensible Markup Language (XML) Representation for Expressing Policy Capabilities".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[42] draft-rosenberg-simple-pres-policy-caps-01 (July 2004): "An Extensible Markup Language (XML) Representation for Expressing Presence Policy Capabilities".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[43] draft-ietf-sipping-config-framework-04 (July 2004): "A Framework for Session Initiation Protocol User Agent Profile Delivery".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[X] draft-ietf-simple-presence-data-model-01 (October 2004): "A Data Model for Presence".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

*** next change ***

5.3 Roles

5.3.1 Presence User Agent (PUA)

5.3.1.1 General

A PUA is an entity that provides presence information to a PS.

In addition to the procedures specified in subclause 5.3.1, the PUA shall support the procedures specified in 3GPP TS 24.229 [9] appropriate to the functional entity in which the PUA is implemented.

5.3.1.2 Publication of presence information

When the PUA intends to publish its own view of the presentity's presence information it shall generate a PUBLISH request by acting as an Event Publication Agent (EPA) in accordance with draft-ietf-sip-publish-03 [23].

The PUA shall implement the "application/pidf+xml" content type as described in draft-ietf-impp-cpim-pidf-08 [21] together with the Presence Information Data Format (PIDF) extensions defined in draft-ietf-simple-rpid-03 [26].

The PUA may implement the PIDF extensions defined in draft-ietf-simple-cipid-01 [32].

The PUA may implement location information according to the format defined in draft-ietf-geopriv-pidf-lo-01 [37].

NOTE 1: The categorization of presence attributes to generic information attributes and communication address specific attributes is done using the ~~<contact type>~~<person> and <tuple> elements as defined draft-ietf-simple-presence-data-model-01 [X]. ~~The value of the <contact type> element categorizes the content of the tuple in such a way that the value "presentity" denotes general information about the presentity and the value "service" denotes the communication mean specific information. The presence document consists of one or more tuples describing the presentity related information and one or more tuples describing information about different communication means.~~

The PUA shall implement draft-ietf-simple-prescaps-ext-00 [25] if it wants to make use of SIP user agent capabilities in the presence document. The extension may be used for describing the type of the service described by the presence tuple.

The PUA may indicate its support for partial publishing by including the "application/pidf-partial+xml" content type as described in draft-ietf-simple-partial-pidf-format-00 [38]. The first partial PUBLISH request shall contain the full publication, where the state attribute is set to value "full" and the value of the 'version' attribute is initialized.

On successful response from the PS, the PUA in subsequent PUBLISH requests shall generate partial publications, where the state attribute is set to value 'partial' and the value of the 'version' attribute is incremented by one. The partial publication should contain only the new and changed tuples, information about the removed tuples and presence information outside the tuple elements.

If the PUA receives a 415 (Unsupported Media Type) response to the PUBLISH request with "application/pidf+xml" in the Accept header field, the PUA shall send a PUBLISH request including the "application/pidf+xml".content type.

Editor's Note: The above procedures on partial publishing will be replaced by references to the IETF draft-lonnfors-simple-publish-partial-00 once the draft has been discussed in IETF. If IETF defines another solution for partial publishing or indicate the reuse of existing procedures as a solution, then the above procedures on partial publishing will be revised.

The PUA shall update the presence information, either 600 s before the publication expiration time if the publication period indicated from the PS in the response to the PUBLISH request was for greater than 1 200 s, or when half of the time has expired if the publication period was for 1 200 s or less, unless the UE has determined that an update to the presence information is not required.

When the PUA intends to show different value of the same presence attribute to different watchers, the PUA shall publish a tuple or person element for every value it intends to show, all including a different value of the same presence attribute. The PUA shall label different information with different value of the <class> element in every published tuple or person element as defined in draft-ietf-simple-rpid-03 [26]. The PUA shall also authorize different tuples to different watchers or watcher groups by manipulating the subscription authorization policy as defined in subclause 6.3.1.2.

If a local configuration information limiting the rate at which PUA is allowed to generate PUBLISH requests is available, then PUA shall take that information into account. Such local configuration information could be e.g. the shortest time period between consecutive PUBLISH requests.

5.3.1.3 Mapping of presence attributes

The eXtensible Markup Language (XML) Schema Definition of the "application/pidf+xml" format or the "application/pidf-partial+xml" format cover the definition of the 3GPP subscriber's presence attributes and the PUA shall perform the following mapping:

- the communication address (containing communication means, status and contact address) attribute and the priority attribute are represented by a <tuple> element including a basic <status> element and ~~one or more~~a <contact> element~~s~~ containing a priority attribute as defined in draft-ietf-impp-cpim-pidf-08 [21].

  The PUA represents the subscriber's status by including a <person>~~<contact type>~~ element defined in draft-ietf-simple-presence-data-model-01 [X]~~draft-ietf-simple-rpid-03 [26] with the value "presentity" and~~ a basic <status> element defined in draft-ietf-impp-cpim-pidf-08 [21]. In order to express more granularity in values, <activities~~y~~> and <place-type~~rivacy~~> elements both defined in draft-ietf-simple-rpid-03 [26] can be used inside the <status> elements. Further PIDF extensions as defined in draft-ietf-simple-cipid-01 [32] can also be used.

  ~~In case of including multiple presentity related tuples in the presence document, all the presentity related tuples except one contains information about an alternate contact related to the presentity; the type of the alternate contact shall be indicated using the <relationship> element defined in draft-ietf-simple-rpid-03 [26];~~

NOTE 1: draft-ietf-simple-presence-data-model-01 [X]~~draft-ietf-simple-rpid-03 [26]~~ defines also a <device> element which can be used to present device specific information.~~other values of the <contact type> element. Those values can be used to create additional categories.~~

- the text attribute is represented by the <note> element as defined in draft-ietf-impp-cpim-pidf-08 [21]; and
- the location attribute is represented by the elements defined in draft-ietf-geopriv-pidf-lo-01 [37] and the <place-type> element defined in draft-ietf-simple-rpid-03 [26].

NOTE 2: Only information elements either relevant for the application or recommented by the presence data model draft-ietf-simple-presence-data-model-01 [X] are~~is~~ included in the PUBLISH request. Attributes not relevant or available (e.g. the text attribute or the location attribute) are omitted.

Additional extensions can be used to express application specific attributes, but their usage is outside the scope of this version of the specification.

*** next change ***

5.3.2 Watcher

5.3.2.1 General

A watcher is an entity that is subscribed or requests presence information about a presentity from the PS.

In addition to the procedures specified in subclause 5.3.2, the watcher shall support the procedures specified in 3GPP TS 24.229 [9] appropriate to the functional entity in which the watcher is implemented.

5.3.2.2 Subscription for presence information state changes and notification acceptance

When the watcher application intends to subscribe for presence information state changes of a presentity, it shall generate a SUBSCRIBE request in accordance with RFC 3265 [19] and draft-ietf-simple-presence-10 [27].

The watcher application shall implement the "application/pidf+xml" content type as described in draft-ietf-impp-cpim-pidf-08 [21] together with the PIDF extensions defined in draft-ietf-simple-rpid-03 [26].

The watcher application may implement the PIDF extensions defined in draft-ietf-simple-cipid-01 [32].

The watcher application shall implement draft-ietf-simple- prescaps-ext-00 [25]if it wants to make use of SIP user agent capabilities extensions included in the presence document. The extension may be used by the watcher application for interpreting the type of the service described by the presence tuple. The watcher application may include filters in the

body of the SUBSCRIBE request in accordance with draft-ietf-simple-filter-format-00 [30] and draft-ietf-simple-event-filter-funct-00 [31].

The watcher application may indicate its support for partial notification using the Accept header field in accordance with draft-ietf-simple-partial-notify-01 [24].

The watcher application shall interpret the received presence information according to draft-ietf-simple-presence-data-model-01 and the following:

a) a <person>~~tuple including a <contact-type>~~ element as defined in draft-ietf-simple-presence-data-model-01 [X]~~draft-ietf-simple-rpid-03 [26] with the value "presentity" means general~~ information about the presentity;

b) a tuple including a <relationship> element ~~and <contact-type> element with the value "presentity" as~~ defined in draft-ietf-simple-rpid-03 [26] means information about an alternate contact to the presentity;

c) ~~c)~~ a tuple ~~including a <contact-type> element as defined in draft-ietf-simple-rpid-03 [26] with the value "service"~~ means communication mean specific information. The communication mean described by the tuple is deduced from the URI scheme of the contact address information present in the <contact> element as defined in draft-ietf-impp-cpim-pidf-08 [21]. If the URI scheme of the contact address information provides ambiguous information about the communication means, the watcher application shall further examine other elements of the tuple to decide the communication mean. Such elements can be the <methods> element, any of the different media type specific elements as defined in draft-ietf-simple-prescaps-ext-00 [25]~~, or the <relationship> element as defined in draft-ietf-simple-rpid-03 [26]~~.

d) a <device> element as defined in draft-ietf-simple-presence-data-model-01 [X] means information about a device.

Additional extensions can be used to express application specific attributes, but their usage is outside the scope of this version of the specification.

*** next change ***

6.3.1.3 Manipulating the subscription authorization policy

When the DM intends to manipulate the subscription authorization policy, it shall generate an HTTP PUT, HTTP GET or HTTP DELETE request in accordance with RFC 2616 [15], draft-ietf-simple-xcap-03 [33] and draft-ietf-simple-xcap-presence-rules-00 [35].

The DM may use an HTTP GET in accordance with RFC 2616 [15], draft-ietf-simple-xcap-03 [33] and draft-rosenberg-simple-common-policy-caps-01 [42] for fetching of the authorization policy capabilities which the DMS supports.

When the DM intends to authorize a different value of the same presence attribute to different watchers or watcher groups, the DM shall authorize a single tuple or person element including one of the different values of the same presence attribute to every watcher or watcher groups by using a specific "inclusion set" as specified in draft-ietf-simple-xcap-presence-rules-00 [35].

3GPP TSG-CN1 Meeting #36
Seoul, Korea, 15-19 November 2004

Tdoc N1-041973

CR-Form-v7.1

CHANGE REQUEST

⌘ **24.141** **CR** **28** ⌘ **rev** **1** ⌘ Current version: **6.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	Correction to Watcher Information message flow
Source:	⌘	Nokia
Work item code:	⌘	PRESNC
Date:	⌘	16/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	Correction to Watcher Information message flows.
Summary of change:	⌘	This CR clarifies the default functionality of the watcher information regarding the message flows. The default functionality according to IETF document is that the notifications triggered from a SUBSCRIBE contain full state, but notifications triggered from a change in watcher state only contain information on the watcher whose state has changed. The CR also clarifies the initial FC setting in the case when a watcher subscribes watcher information of its own subscriptions.
Consequences if not approved:	⌘	The watcher information message flow is not so well in-line with the corresponding IETF functionality.

Clauses affected:	⌘	A.6

Other specs affected:	⌘	Y	N		⌘
			X	Other core specifications	
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

*** CHANGES ***

A.6 PUA subscribing to his own watcher list and receiving notification of new watcher subscriptions

A.6.1 Introduction

This subclause covers the signalling flows that show how a PUA can subscribe to his own watcher list.

A.6.2 PUA subscribing to watcher list and receiving a notification of an already pending watcher subscription followed by a notification of a subscription from a new watcher not already in the watcher list

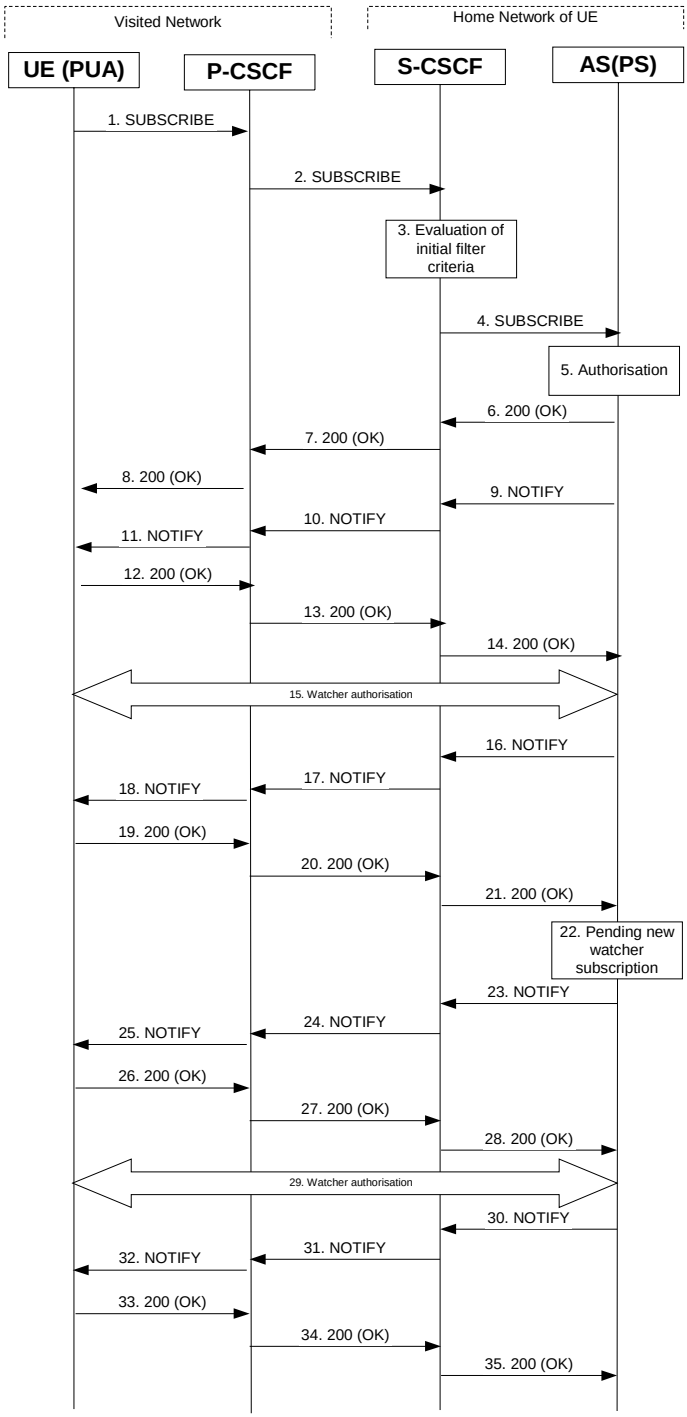


Figure A.6.2-1: PUA subscribing to watcher list and receiving a notification of an already pending watcher subscription followed by a notification of a subscription from a new watcher not already in the watcher list

Figure A.6.2-1 shows a PUA subscribing to watcher list and receiving a notification of an already pending watcher subscription followed by a notification of a subscription from a new watcher not already in the watcher list. In this example the default watcherinfo subscription filtering policy is applied meaning that a partial state of a watcher-info

document is transported in the notifys ~~for the second watcher subscription~~. The details of the signalling flows as follows:

1. **SUBSCRIBE request (UE to P-CSCF) – see example in table A.6.2-1**

   The presentity wishes to watch his own watcher information, therefore he subscribes for the watcher information template-package of presence. The UE generates a SUBSCRIBE request containing the presence.winfo event, together with an indication of the length of time this periodic subscription should last.

Table A.6.2-1: SUBSCRIBE request (UE to P-CSCF)

```
SUBSCRIBE sip:user1_public1@home1.net SIP/2.0
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
Max-Forwards: 70
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
Route: <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>, <sip:orig@scscf1.home1.net;lr>
P-Preferred-Identity: <sip:user1_public1@home1.net>
Privacy: none
From: <sip:user1_public1@home1.net>;tag=31415
To: <sip:user1_public1@home1.net>
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 123 SUBSCRIBE
Require: sec-agree
Proxy-Require: sec-agree
Security-Verify: ipsec-3gpp; q=0.1; alg=hmac-sha-1-96; spi-c=98765432; spi=87654321; port-
      c=8642; port-s=7531
Event: presence.winfo
Expires: 7200
Accept: application/watcherinfo+xml
Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>
Content-Length: 0
```

Request URI: Public user identity whose events the subscriber subscribes to. In this case the Public User Identity of the presentity in SIP URI format.

Event: This field is populated with the value "presence.winfo" to specify the use of the watcher information template-package of presence.

Accept: This field is populated with the value 'application/watcherinfo+xml' indicating that the UE supports this body type for notification.

To: Same as the Request-URI.

2. **SUBSCRIBE request (P-CSCF to S-CSCF) – see example in table A.6.2-2**

The P-CSCF looks up the serving network information for the public user identity that was stored during the registration procedure. The SUBSCRIBE request is forwarded to the S-CSCF. A Route header is inserted into SUBSCRIBE request.

Table A.6.2-2: SUBSCRIBE request (P-CSCF to S-CSCF)

```
SUBSCRIBE sip:user1_public1@home1.net SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=z9hG4bK120f34.1 ,SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
P-Access-Network-Info:
Max-Forwards: 69
P-Asserted-Identity: <sip:user1_public1@home1.net>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"
Privacy:
Route: <sip:orig@scscf1.home1.net;lr>
Record-Route: <sip:pcscf1.visited1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

3. **Evaluation of initial filter criteria**

The S-CSCF validates the service profile of this subscriber and evaluates the initial filter criteria. For sip:user1_public1@home1.net the S-CSCF has originating initial Filter Criteria with Service Point Trigger of Method = SUBSCRIBE AND Event = "presence.winfo" that informs the S-CSCF to route the SUBSCRIBE request to the Application Server sip:ps.home1.net.

4. **SUBSCRIBE request (S-CSCF to PS) – see example in table A.6.2-4**

The S-CSCF forwards the SUBSCRIBE request to the PS.

Table A.6.2-4: SUBSCRIBE request (S-CSCF to PS)

```
SUBSCRIBE sip:user1_public1@home1.net SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK344a65.1, SIP/2.0/UDP
      pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
P-Access-Network-Info:
Max-Forwards: 68
P-Asserted-Identity: <sip:user1_public1@home1.net>, <tel:+1-212-555-1111>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Privacy:
Route: <sip:ps.home1.net;lr>, <sip:scscf1.home1.net;lr>
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF inserts the originating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the PS.

5. **Authorization**

The PS performs the necessary authorization checks on the originator. In this example, the originator is the owner of the watcher information, so he/she is authorized to see the full watcher information.

In other examples (when the originator is not the owner of the watcher information) subscribers are only allowed to monitor the state of their own subscription, which means that they will receive notifications only containing the state of their own subscription. This requires that a terminating initial Filter Criteria with Service Point Trigger of Method = SUBSCRIBE AND Event = "presence.winfo" AND To = "sip:user1_public1@home1.net" has been defined for the user sip:user1_public1@home1.net.

6. **200 (OK) response (PS to S-CSCF) - see example in table A.6.2-6**

The PS sends the response to the S-CSCF.

Table A.6.2-6: 200 (OK) response (PS to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK344a65.1, SIP/2.0/UDP
      pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"; orig-
      ioi=home1.net:term-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Record-Route:
From:
To: <sip:user1_public1@home1.net>;tag=151170
Call-ID:
CSeq:
Expires:
Contact: <sip:ps.home1.net>
Content-Length: 0
```

P-Charging-Vector: The PS stores the originating Inter Operator Identifier (IOI) parameter and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The PS stores the P-Charging-Function-Addresses header field and passes this header to the S-CSCF.

7. **200 (OK) response (S-CSCF to P-CSCF) - see example in table A.6.2-7**

The S-CSCF forwards the response to the P-CSCF.

Table A.6.2-7: 200 (OK) response (S-CSCF to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the terminating Inter Operator Identifier (IOI) parameter.

8. **200 (OK) response (P-CSCF to UE) - see example in table A.6.2-8**

The P-CSCF forwards the response to the PUA in the UE.

Table A.6.2-8: 200 (OK) response (P-CSCF to UE)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
Record-Route: <sip:orig@scscf1.home1.net;lr>, <sip:pcscf1.home1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

9. **NOTIFY request (PS to S-CSCF) - see example in table A.6.2-9**

After the PS generated a 200 (OK) response to the SUBSCRIBE request from the UE, it generates a NOTIFY request containing the current state of the watcher information. The watcher information contains one pending subscription.

Table A.6.2-9 NOTIFY request (PS to S-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP ps.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 70
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=123551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
From: <sip:user1_public1@home1.net>;tag=151170
To: <sip:user1_public1@home1.net>;tag=31415
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 89 NOTIFY
Subscription-State: active;expires=7200
Event: presence.winfo
Contact: <sip:ps.home1.net>
Content-Type: application/watcherinfo+xml
Content-Length: (...)

<?xml version="1.0"?>
   <watcherinfo xmlns="urn:ietf:params:xml:ns:watcherinfo"
                version="0" state="full">
     <watcher-list resource="sip:user1_public1@home1.net" package="presence">
       <watcher id="77ajsyy76" event="subscribe"
                status="pending">sip:user2_public1@home2.net</watcher>
     </watcher-list>
   </watcherinfo>
```

P-Charging-Vector: The PS populates the icid parameter with a globally unique value and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The PS populates the P-Charging-Function-Addresses header field to be passed to the S-CSCF.

10. **NOTIFY request (S-CSCF to P-CSCF) – see example in table A.6.2-10**

The S-CSCF forwards the NOTIFY request to the P-CSCF.

Table A.6.2-10: NOTIFY request (S-CSCF to P-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      ps.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 69
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=123551024"
P-Charging-Function-Addresses:
Route: <sip:pcscf1.visited1.net;lr>
Record-Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter received.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the P-CSCF.

11. **NOTIFY request (P-CSCF to UE) - see example in table A.6.2-11**

The P-CSCF forwards the NOTIFY request to the PUA in the UE.

Table A.6.2-11: NOTIFY request (P-CSCF to UE)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.home1.net:7531;lr;comp=sigcomp>
Max-Forwards: 68
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

12. **200 (OK) response (UE to P-CSCF) – see example in table A.6.2-12**

The PUA on the UE determines that this is a full state watcher-info document and replaces any current watcher-info with the new document. The UE acknowledges the NOTIFY request with a 200 (OK) response to the P-CSCF.

Table A.6.2-12: 200 (OK) response (UE to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf2.home2.net;branch=z9hG4bK764z87.1, SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

13. **200 (OK) response (P-CSCF to S-CSCF) – see example in table A.6.2-13**

The P-CSCF forwards the 200 (OK) response to the S-CSCF.

Table A.6.2-13: 200 (OK) response (P-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      ps.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=123551024"
From:
To:
Call-ID:
CSeq:
Content-Length:
```

14. **200 (OK) response (S-CSCF to PS) – see example in table A.6.2-14**

The P-CSCF forwards the response to the PS in the home network of the UE.

Table A.6.2-14: 200 (OK) response (S-CSCF to PS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP ps.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=123551024"; orig-
      ioi=home1.net:term-ioi=home1.net
From:
To:
Call-ID:
CSeq:
Content-Length:
```

P-Charging-Vector: The S-CSCF insertes the terminating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

15. **Authorization of watcher**

The presentity determines to allow the watcher to access the presence information. The PUA modifies the subscription authorization policy by authorizing presence information for sip:user2_public1@home1.net.

16. **NOTIFY request (PS to S-CSCF) – see example in table A.6.2-16**

The authorization event means changes in the watcher information, which triggers a new NOTIFY request. The watcher information included in the NOTIFY request contains only information on the watcher whose state has changed, which in this example is the accepted subscription of sip:user2_public1@home1.net.

Table A.6.2-16: NOTIFY request (PS to S-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP ps.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 70
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
From: <sip:user1_public1@home1.net>;tag=151170
To: <sip:user1_public1.home1.net>;tag=31415
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 90 NOTIFY
Subscription-State: active;expires=4900
Event: presence.winfo
Contact: <sip:ps.home1.net>
Content-Type: application/watcherinfo+xml
Content-Length: (...)

<?xml version="1.0"?>
   <watcherinfo xmlns="urn:ietf:params:xml:ns:watcherinfo"
                version="0" state="partialfull">
     <watcher-list resource="sip:user1_public1@home1.net" package="presence">
       <watcher id="77ajsyy76" event="subscribe"
                status="active">sip:user2_public1@home2.net</watcher>
     </watcher-list>
   </watcherinfo>
```

P-Charging-Vector: The PS populates the icid parameter with a globally unique value and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The PS populates the P-Charging-Function-Addresses header field to be passed to the S-CSCF.

17. **NOTIFY request (S-CSCF to P-CSCF) – see example in table A.6.2-17**

The S-CSCF forwards the NOTIFY request to the P-CSCF.

Table A.6.2-17: NOTIFY request (S-CSCF to P-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      ps.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 69
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:pcscf1.visited1.net;lr>
Record-Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

P-Charging-Vector: The S-CSCF passes this header received.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the P-CSCF.

18. **NOTIFY request (P-CSCF to UE) - see example in table A.6.2-18**

The P-CSCF forwards the NOTIFY request to the PUA in the UE.

Table A.6.2-18: NOTIFY request (P-CSCF to UE)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.home1.net:7531;lr;comp=sigcomp>
Max-Forwards: 68
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

19. **200 (OK) response (UE to P-CSCF) - see example in table A.6.2-19**

The PUA determines that this is a full state watcher-info document and replaces any current watcher-info with the new document. The UE acknowledges the NOTIFY request with a 200 (OK) response to the P-CSCF.

Table A.6.2-19: 200 (OK) response (UE to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf2.home2.net;branch=z9hG4bK764z87.1, SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

20. **200 (OK) response (P-CSCF to S-CSCF) – see example in table A.6.2-20**

The P-CSCF forwards the 200 (OK) response to the S-CSCF.

Table A.6.2-20: 200 (OK) response (P-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      ps.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"
From:
To:
Call-ID:
CSeq:
Content-Length:
```

21. **200 (OK) response (S-CSCF to PS) – see example in table A.6.2-21**

The P-CSCF forwards the response to the PS in the home network of the UE.

Table A.6.2-21: 200 (OK) response (S-CSCF to PS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP ps.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"; orig-ioi=home1.net
      term-ioi=visited1.net
From:
To:
Call-ID:
CSeq:
Content-Length:
```

P-Charging-Vector: The PS inserts the originating Inter Operator Identifier (IOI) parameter received and.populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

22. **Pending new watcher subscription**

The PS receives a SUBSCRIBE request from a new watcher and performs the necessary authorization checks on the originator and determines that this is a new watcher that is not yet in the watcher list.

23. **NOTIFY request (PS to S-CSCF) - see example in table A.6.2-23**

The PS generates a NOTIFY request containing ~~the current state of the~~ watcher information of the new watcher pending subscription. Thus, ~~T~~the watcher information contains the partial state. ~~one pending subscription. Partial state is used for the watcher-info document transported in this subsequent notification.~~

Table A.6.2-23 NOTIFY request (PS to S-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP ps.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 70
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
From: <sip:user1_public1@home1.net>;tag=151170
To: <sip:user1_public1.home1.net>;tag=31415
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 90 NOTIFY
Subscription-State: active;expires=5000
Event: presence.winfo
Content-Type: application/watcherinfo+xml
Contact: <sip:ps.home1.net;lr>
Content-Length: (...)

<?xml version="1.0"?>
   <watcherinfo xmlns="urn:ietf:params:xml:ns:watcherinfo"
                version="0" state="partial">
     <watcher-list resource="sip:user1_public1@home1.net" package="presence">
       <watcher id="34bytzx54" event="subscribe"
                status="pending">sip:user3_public1@home3.net</watcher>
     </watcher-list>
   </watcherinfo>
```

P-Charging-Vector: The PS populates the icid parameter with a globally unique value and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The PS populates the P-Charging-Function-Addresses header field to be passed to the S-CSCF.

24. **NOTIFY request (S-CSCF to P-CSCF) - see example in table A.6.2-24**

The S-CSCF forwards the NOTIFY request to the P-CSCF.

Table A.6.2-24: NOTIFY request (S-CSCF to P-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      ps.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 69
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"
P-Charging-Function-Addresses:
Route: <sip:pcscf1.visited1.net;lr>
Record-Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Content-Type:
Contact:
Content-Length:

(…)
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter received.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the P-CSCF.

25. **NOTIFY request (P-CSCF to UE) - see example in table A.6.2-25**

The P-CSCF forwards the NOTIFY request to the PUA in the UE.

Table A.6.2-25: NOTIFY request (P-CSCF to UE)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 68
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.home1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Content-Type:
Contact:
Content-Length:

(…)
```

26. **200 (OK) response (UE to P-CSCF) - see example in table A.6.2-26**

The PUA determines that this is a partial state notification of watcher-info and adds the new pending subscription to its existing watcher-info document. The UE acknowledges the NOTIFY request with a 200 (OK) response to the P-CSCF.

Table A.6.2-26: 200 (OK) response (UE to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf2.home2.net;branch=z9hG4bK764z87.1, SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

27. **200 (OK) response (P-CSCF to S-CSCF) - see example in table A.6.2-27**

The P-CSCF forwards the 200 (OK) response to the S-CSCF.

Table A.6.2-27: 200 (OK) response (P-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      ps.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"
From:
To:
Call-ID:
CSeq:
Content-Length:
```

28. **200 (OK) response (S-CSCF to PS) - see example in table A.6.2-28**

The P-CSCF forwards the response to the PS in the home network of the UE.

Table A.6.2-28: 200 (OK) response (S-CSCF to PS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP ps.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"; orig-ioi=home1.net;
      term-ioi=visited1.net
From:
To:
Call-ID:
CSeq:
Content-Length:
```

P-Charging-Vector: The S-CSCF insertes the originating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

29. **Authorization of watcher**

The presentity determines to allow the watcher to access the presence information. The PUA modifies the authorization policy by authorizing presence information for sip:user3_public1@home3.net.

30. **NOTIFY request (PS to S-CSCF) - see example in table A.6.2-30**

The authorization event means changes in the watcher information, which triggers a new NOTIFY request. The watcher information included in the NOTIFY request contains the accepted subscription of sip:user3_public1@home3.net.

Table A.6.2-30 NOTIFY request (PS to S-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP ps.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 70
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
From: <sip:user1_public1@home1.net>;tag=151170
To: <sip:user1_public1.home1.net>;tag=31415
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 90 NOTIFY
Subscription-State: active;expires=4900
Event: presence.winfo
Content-Type: application/watcherinfo+xml
Contact: <sip:ps.home1.net;lr>
Content-Length: (...)

<?xml version="1.0"?>
   <watcherinfo xmlns="urn:ietf:params:xml:ns:watcherinfo"
                version="0" state="partial">
     <watcher-list resource="sip:user1_public1@home1.net" package="presence">
       <watcher id="34bytzx54" event="subscribe"
                status="active">sip:user3_public1@home3.net</watcher>
     </watcher-list>
   </watcherinfo>
```

P-Charging-Vector: The PS populates the icid parameter with a globally unique value and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The PS populates the P-Charging-Function-Addresses header field to be passed to the S-CSCF.

31. **NOTIFY request (S-CSCF to P-CSCF) – see example in table A.6.2-31**

The S-CSCF forwards the NOTIFY request to the P-CSCF.

Table A.6.2-31: NOTIFY request (S-CSCF to P-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      ps.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 69
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"
P-Charging-Function-Addresses:
Route: <sip:pcscf1.visited1.net;lr>
Record-Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Content-Type:
Contact:
Content-Length:

(…)
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter received.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the P-CSCF.

32. **NOTIFY request (P-CSCF to UE) - see example in table A.6.2-32**

The P-CSCF forwards the NOTIFY request to the PUA in the UE.

Table A.6.2-32: NOTIFY request (P-CSCF to UE)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 68
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.home1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Content-Type:
Contact:
Content-Length:

(…)
```

33. **200 (OK) response (UE to P-CSCF) - see example in table A.6.2-33**

The PUA determines that this is a partial state notification of watcher-info and updates the active subscription to its existing watcher-info document. The UE acknowledges the NOTIFY request with a 200 (OK) response to the P-CSCF.

Table A.6.2-33: 200 (OK) response (UE to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf2.home2.net;branch=z9hG4bK764z87.1, SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

34. **200 (OK) response (P-CSCF to S-CSCF) - see example in table A.6.2-34**

The P-CSCF forwards the 200 (OK) response to the S-CSCF.

Table A.6.2-34: 200 (OK) response (P-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      ps.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
From:
To:
Call-ID:
CSeq:
Content-Length:
```

35. **200 (OK) response (S-CSCF to PS) - see example in table A.6.2-35**

The P-CSCF forwards the response to the PS in the home network of the UE.

Table A.6.2-35: 200 (OK) response (S-CSCF to PS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP ps.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
From:
To:
Call-ID:
CSeq:
Content-Length:
```

3GPP TSG-CN1 Meeting #36 **Tdoc N1-041974**
Seoul, Korea, 15-19 November 2004

CR-Form-v7.1

CHANGE REQUEST

⌘ **24.141** **CR** **30** ⌘ **rev** **1** ⌘ Current version: **6.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network ☒

Title:	⌘	Preventing loop in RLS subscriptions
Source:	⌘	Nokia
Work item code:	⌘	PRESNC
Date:	⌘	16/11/2004
Category:	⌘	C
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	By the ability to nest resource lists it is possible to create lists which ultimately contain themselves.
Summary of change:	⌘	Back-end subsciptions for lists are forbidden
Consequences if not approved:	⌘	Loops in RLS subscriptions are possible.

Clauses affected:	⌘	5.3.4.3

	Y	N			
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be

downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

=======FIRST CHANGE=======

5.3.4.3 Subscription to presence information

When the RLS receives a SUBSCRIBE request for the presence information event package of a presentity collection and installs the corresponding subscription, the RLS shall resolve the list URI to individual URIs and generate SUBSCRIBE requests for each of the individual URIs as per the procedures in RFC 3265 [19], draft-ietf-simple-presence-10 [27] and draft-ietf-simple-event-list-04 [22] if the state information for the resource represented by the individual URI is otherwise not available.

For internal virtual subscriptions the detection of loops potentially caused by lists of lists is possible in RLS. However for back-end subscriptions (see draft-ietf-simple-event-list-05 [22]) the detection of such situations is not possible in RLS. To prevent loops in subscriptions to non-local resources the RLS shall not insert "eventlist" in the "Supported" header of back-end subscriptions.

~~Editor's note: There is a need for a mechanism that can protect an IMS network from list loops potentially caused by lists of lists. Unless referenced IETF specifications provide support for implementation of this kind of protection, a mechanism or restrictions on the usage of list of lists must be identified and described here.~~

=======END OF CHANGES=======

3GPP TSG-CN1 Meeting #36
Seoul, Korea, 15-19 November 2004

Tdoc N1-041992

CR-Form-v7.1

CHANGE REQUEST

⌘ **24.141** **CR** **27** ⌘ **rev** **1** ⌘ Current version: **6.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	Updates to Partial publication
Source:	⌘	Nokia
Work item code:	⌘	PRESNC
Date:	⌘	16/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	A WG version of the partial publication I-D has been published. Removal of the Editor's note. IETF provided solution has changed and is no more in-line with the current texts.
Summary of change:	⌘	This CR proposes the removal of the Editor's Note concerning the partial publications and adding a reference to a IETF SIMPLE WG version of the partial publication draft. By adding the reference, some of the functionality specified by TS can be replaced by it.
Consequences if not approved:	⌘	IETF provided solution differs from 3GPP solution.

Clauses affected:	⌘	2, 5.3.1, 5.3.3

Other specs affected: ⌘	Y	N		⌘
		X	Other core specifications	
		X	Test specifications	
		X	O&M Specifications	

Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

**** 1st CHANGE ****

2 References

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

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

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

[2] 3GPP TS 22.141: "Presence Service; Stage 1".

[3] 3GPP TS 23.002: "Network architecture".

[4] 3GPP TS 23.141: "Presence service; Architecture and functional description; Stage 2".

[5] 3GPP TS 23.218: "IP Multimedia (IM) session handling; IM call model; Stage 2".

[6] 3GPP TS 23.228: "IP Multimedia Subsystem (IMS); Stage 2".

[7] 3GPP TS 24.109: "Bootstrapping interface (Ub) and Network application function interface (Ua); Protocol details".

[8] 3GPP TS 24.228: "Signalling flows for the IP multimedia call control based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3".

[9] 3GPP TS 24.229: "Internet Protocol (IP) multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3".

[10] 3GPP TS 29.228: "IP Multimedia (IM) Subsystem Cx and Dx Interfaces; Signalling flows and message contents".

[11] 3GPP TS 33.141: "Presence service; Security".

[12] IETF RFC 2664 (1999): "FYI on Questions and Answers - Answers to Commonly asked New Internet User Questions".

[13] IETF RFC 2246 (1999): "The TLS Protocol Version 1.0".

[14] IETF RFC 2387 (August 1998): "The MIME Multipart/Related Content-type".

[15] IETF RFC 2616 (June 1999): "Hypertext Transfer Protocol -- HTTP/1.1".

[15A] IETF RFC 2617 (June 1999): " HTTP Authentication: Basic and Digest Access Authentication".

[16] IETF RFC 2778 (2000): "A Model for Presence and Instant Messaging".

[17] IETF RFC 3261 (June 2002): "SIP: Session Initiation Protocol".

[18] IETF RFC 3263 (June 2002): "Session Initiation Protocol (SIP): Locating SIP Servers".

[19] IETF RFC 3265 (March 2002): "Session Initiation Protocol (SIP)-Specific Event Notification".

[20] IETF RFC 3310 (2002): "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)".

[21] draft-ietf-impp-cpim-pidf-08 (May 2003): "Presence Information Data Format (PIDF)".

[22] draft-ietf-simple-event-list-04 (June 2003): "A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists".

[23] draft-ietf-sip-publish-03 (February 2004): "An Event State Publication Extension to the Session Initiation Protocol (SIP)".

[24] draft-ietf-simple-partial-notify-01 (January 2004): "Partial Notification of Presence Information".

[25] draft-ietf-simple-prescaps-ext-00 (February 2004): "Device capability PIDF status extension".

[26] draft-ietf-simple-rpid-03 (March 2004): "RPID: Rich Presence: Extensions to the Presence Information Data Format (PIDF)".

[27] draft-ietf-simple-presence-10 (January 2003): "A Presence Event Package for the Session Initiation Protocol (SIP)".

[28] draft-ietf-simple-winfo-package-05 (January 2003): "A Session Initiation Protocol (SIP) Event Template-Package for Watcher Information".

[29] draft-ietf-simple-winfo-format-04 (January 2003): "An Extensible Markup Language (XML) Based Format for Watcher Information".

[30] draft-ietf-simple-filter-format-00 (February 2004): "An Extensible Markup Language (XML) Based Format for Event Notification Filtering".

[31] draft-ietf-simple-event-filter-funct-00 (February 2003): "Functional Description of Event Notification Filtering".

[32] draft-ietf-simple-cipid-01 (March 2004): "CIPID: Contact Information in Presence Information Data Format".

[33] draft-ietf-simple-xcap-03 (July 2004): "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)".

[34] draft-isomaki-simple-xcap-pidf-manipulation-usage-00 (February 2004): "An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usage for Manipulating Presence Document Contents".

[35] draft-ietf-simple-xcap-presence-rules-00 (May 2004): "Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usages for Setting Presence Authorization".

[36] draft-ietf-simple-xcap-list-usage-02 (February 2004): "An Extensible Markup Language (XML) Format for Representing Resource Lists".

[37] draft-ietf-geopriv-pidf-lo-01 (February 2004): "A Presence-based GEOPRIV Location Object Format".

[38] draft-ietf-simple-partial-pidf-format-02~~0~~ (October~~January~~ 2004): "Presence Information Data Format (PIDF) Extension for Partial Presence".

[39] draft-ietf-simple-xcap-package-02 (July 2004): "A Session Initiation Protocol (SIP) Event Package for Modification Events for the Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Managed Documents".

[40] draft-ietf-sip-content-indirect-mech-03 (June 2003): "A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages".

[41] draft-rosenberg-simple-common-policy-caps-01 (July 2004): "An Extensible Markup Language (XML) Representation for Expressing Policy Capabilities".

[42] draft-rosenberg-simple-pres-policy-caps-01 (July 2004): "An Extensible Markup Language (XML) Representation for Expressing Presence Policy Capabilities".

[43] draft-ietf-sipping-config-framework-04 (July 2004): "A Framework for Session Initiation Protocol User Agent Profile Delivery".

[44] draft-ietf-simple-partial-publish-01 (October 2004): "Partial Publication of Presence Information".

Editor's note: The above document cannot be formally referenced until it is published as an RFC.

****** NEXT CHANGE ******

5.3.1 Presence User Agent (PUA)

5.3.1.1 General

A PUA is an entity that provides presence information to a PS.

In addition to the procedures specified in subclause 5.3.1, the PUA shall support the procedures specified in 3GPP TS 24.229 [9] appropriate to the functional entity in which the PUA is implemented.

5.3.1.2 Publication of presence information

When the PUA intends to publish its own view of the presentity's presence information it shall generate a PUBLISH request by acting as an Event Publication Agent (EPA) in accordance with draft-ietf-sip-publish-03 [23].

The PUA shall implement the "application/pidf+xml" content type as described in draft-ietf-impp-cpim-pidf-08 [21] together with the Presence Information Data Format (PIDF) extensions defined in draft-ietf-simple-rpid-03 [26].

The PUA may implement the PIDF extensions defined in draft-ietf-simple-cipid-01 [32].

The PUA may implement location information according to the format defined in draft-ietf-geopriv-pidf-lo-01 [37].

NOTE 1: The categorization of presence attributes to generic information attributes and communication address specific attributes is done using the <contact-type> element. The value of the <contact-type> element categorizes the content of the tuple in such a way that the value "presentity" denotes general information about the presentity and the value "service" denotes the communication mean specific information. The presence document consists of one or more tuples describing the presentity related information and one or more tuples describing information about different communication means.

The PUA shall implement draft-ietf-simple-prescaps-ext-00 [25] if it wants to make use of SIP user agent capabilities in the presence document. The extension may be used for describing the type of the service described by the presence tuple.

The PUA may implement draft-ietf-simple-partial-publish-01 [44] if it wants to use the~~indicate its support for~~ partial publication mechanism.~~shing by including the "application/pidf-partial+xml" content type as described in draft-ietf-simple-partial-pidf-format-00 [38].~~ The first partial PUBLISH request shall contain the full publication using the

"application/pidf+xml".content type. The PUA uses the "application/pidf-diff+xml" content type as described in draft-ietf-simple-partial-pidf-format-02 [38] for delivering information about changes compared to previous publications of the PUA., where the state attribute is set to value "full" and the value of the 'version' attribute is initialized.

On successful response from the PS, the PUA in subsequent PUBLISH requests shall generate partial publications, where the state attribute is set to value 'partial' and the value of the 'version' attribute is incremented by one. The partial publication should contain only the new and changed tuples, information about the removed tuples and presence information outside the tuple elements.

If the PUA receives a 415 (Unsupported Media Type) response to the PUBLISH request with "application/pidf+xml" in the Accept header field, the PUA shall send a PUBLISH request including the "application/pidf+xml".content type.

Editor's Note: The above procedures on partial publishing will be replaced by references to the IETF draft-lonnfors-simple-publish-partial-00 once the draft has been discussed in IETF. If IETF defines another solution for partial publishing or indicate the reuse of existing procedures as a solution, then the above procedures on partial publishing will be revised.

The PUA shall update the presence information, either 600 s before the publication expiration time if the publication period indicated from the PS in the response to the PUBLISH request was for greater than 1 200 s, or when half of the time has expired if the publication period was for 1 200 s or less, unless the UE has determined that an update to the presence information is not required.

When the PUA intends to show different value of the same presence attribute to different watchers, the PUA shall publish a tuple for every value it intends to show, all including a different value of the same presence attribute. The PUA shall label different information with different value of the <class> element in every published tuple as defined in draft-ietf-simple-rpid-03 [26]. The PUA shall also authorize different tuples to different watchers or watcher groups by manipulating the subscription authorization policy as defined in subclause 6.3.1.2.

If a local configuration information limiting the rate at which PUA is allowed to generate PUBLISH requests is available, then PUA shall take that information into account. Such local configuration information could be e.g. the shortest time period between consecutive PUBLISH requests.

5.3.1.3 Mapping of presence attributes

The eXtensible Markup Language (XML) Schema Definition of the "application/pidf+xml" format or the "application/pidf-partial+xml" format covers the definition of the 3GPP subscriber's presence attributes and the PUA shall perform the following mapping:

- the communication address (containing communication means, status and contact address) attribute and the priority attribute are represented by a <tuple> element including a basic <status> element and one or more <contact> elements containing a priority attribute as defined in draft-ietf-impp-cpim-pidf-08 [21].

  The PUA represents the subscriber's status by including a <contact-type> element defined in draft-ietf-simple-rpid-03 [26] with the value "presentity" and a basic <status> element defined in draft-ietf-impp-cpim-pidf-08 [21]. In order to express more granularity in values, <activity> and <privacy> elements both defined in draft-ietf-simple-rpid-03 [26] can be used inside the <status> elements. Further PIDF extensions as defined in draft-ietf-simple-cipid-01 [32] can also be used.

  In case of including multiple presentity related tuples in the presence document, all the presentity related tuples except one contains information about an alternate contact related to the presentity; the type of the alternate contact shall be indicated using the <relationship> element defined in draft-ietf-simple-rpid-03 [26];

NOTE 1: draft-ietf-simple-rpid-03 [26] defines also other values of the <contact-type> element. Those values can be used to create additional categories.

- the text attribute is represented by the <note> element as defined in draft-ietf-impp-cpim-pidf-08 [21]; and

- the location attribute is represented by the elements defined in draft-ietf-geopriv-pidf-lo-01 [37] and the <placetype> element defined in draft-ietf-simple-rpid-03 [26].

NOTE 2: Only information elements relevant for the application is included in the PUBLISH request. Attributes not relevant or available (e.g. the text attribute or the location attribute) are omitted.

Additional extensions can be used to express application specific attributes, but their usage is outside the scope of this version of the specification.

**** NEXT CHANGE ****

5.3.3 Presence Server (PS)

5.3.3.1 General

A PS is an entity that accepts, stores, and distributes presence information.

In addition to the procedures specified in subclause 5.3.3, the PS shall support the procedures specified in 3GPP TS 24.229 [9] appropriate to the functional entity in which the PS is implemented.

5.3.3.2 Subscription acceptance to presence information and notification of state changes

When the PS receives a SUBSCRIBE request for the presence information event package, the PS shall first attempt to verify the identity of the source of the SUBSCRIBE request as described in 3GPP TS 24.229 [9] subclause 5.7.1.4, then perform authorization according to 3GPP TS 24.229 [9] subclause 5.7.1.5. In case of successful subscription, the PS shall generate a response to the SUBSCRIBE request and notifications in accordance with RFC 3265 [19] and draft-ietf-simple-presence-10 [27].

Additionally, in the special case of a watcher subscription if the subscription authorization policy results in the action to confirm the watcher subscription from the PUA and the PUA has a valid watcher information subscription, see draft-ietf-simple-winfo-package-05 [28], then, the PS shall inform the PUA about the watcher subscription attempt.

If the watcher application has indicated the need for partial notification using the Accept header field, then the PS shall generate partial notifications in accordance with draft-ietf-simple-partial-notify-01 [24] and draft-ietf-simple-partial-pidf-format-00 [38].

If the body of the SUBSCRIBE request from the watcher contains filters, the PS shall apply the requested filtering function on notifications in accordance with draft-ietf-simple-filter-format-00 [30] and draft-ietf-simple-event-filter-funct-00 [31].

If the watcher application has indicated support for the "multipart/related" content type using the Accept header field, then the PS may generate notifications using "multipart/related" content type which aggregates "application/pidf+xml" formatted presence information with other MIME objects in accordance with RFC 2387 [14]. In this case, the PS shall modify the value of the presence attribute in the PIDF document to refer to the MIME object included in the corresponding MIME multipart body. If the watcher application has not indicated support for the "multipart/related" or a MIME object cannot be accessed by the PS, the PS should exclude the presence attribute from the notification.

5.3.3.3 Publication acceptance of presence information

The PS shall act as an Event State Compositor (ESC).

When the PS receives a PUBLISH request, the PS shall first verify the identity of the source of the PUBLISH request as described in 3GPP TS 24.229 [9] subclause 5.7.1.4, then perform authorization according to 3GPP TS 24.229 [9] subclause 5.7.1.5. In case of successful authentication and authorization, the PS shall process the PUBLISH request in accordance with draft-ietf-sip-publish-03 [23].

If the PUBLISH request contained ~~indicated support for partial publishing using~~ the "application/pidf~~-partial~~diff+xml" content-type as described in draft-ietf-simple-partial-pidf-format-00 [38]~~ and if the PS supports partial publishing~~, the PS shall process the PUBLISH request in accordance with draft-ietf-sip-publish-03 [23] and draft-ietf-simple-partial-publish-01 [44]. ~~as follows:~~

~~- if the state attribute's value is "full" the PS shall store the received presence and version information;~~

~~- if the state attribute's value is "partial" the PS shall ensure that the version information is correct compared to the stored one; combine the received presence information with the stored one by replacing presence information outside the tuples, replacing changed tuples, adding new tuples and removing tuples which id attributes' values have been listed in the <removed> element defined in draft-ietf-simple-partial-pidf-format-00 [38]; and store the received version information.~~

~~If the PS does not support partial publishing, then the PS shall send a 415 (Unsupported Media Type) response with "application/pidf+xml" in the Accept header field.~~

~~Editor's Note: The above procedures on partial publishing will be replaced by references to the IETF draft lonnfors-simple-publish-partial-00 once the draft has been discussed in IETF. If IETF defines another solution for partial publishing or indicate the reuse of existing procedures as a solution, then the above procedures on partial publishing will be revised.~~

If the PUBLISH request contained the "multipart/related" content type and the PS supports the content type, the PS shall process the content as follows:

- if a MIME multipart contains a MIME object of a content type supported by the PS, either store the MIME object in case of initial publication or replace an existing content in case of modify operation; and

- if a multipart includes the "message/external-body" content type and the content indirection is supported by the PS, ensure that it has access to the MIME object indicated by the URI and that the MIME object exists; and associate the value of the presence attribute that refers to the MIME object with the MIME object and additional information about it.

If the PS does not support the content type used for publishing MIME objects then the PS shall send a 415 (Unsupported Media Type) response and indicate the supported content types in the Accept header.

NOTE: If the PS receives a HTTP request for storing a MIME object on the PS meaning that the HTTP URI points to a predefined directory reserved for storing MIME objects and the request is an HTTP PUT request, the PS replaces any existing content referenced by the Request-URI with the content of the request. If the Request-URI points to an uncreated directory, the PS creates the directory, stores the content there and associates the content with the Request-URI. For all requests, i.e. HTTP PUT, HTTP GET and HTTP DELETE requests, the PS generates an appropriate response in accordance with RFC 2616 [15].

3GPP TSG-CN1 Meeting #36
Seoul, Korea, 15-19 November 2004

Tdoc N1-041993

CR-Form-v7.1

CHANGE REQUEST

⌘ **24.141** **CR** **26** ⌘ **rev** **1** ⌘ Current version: **6.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	IETF reference update (XCAP)
Source:	⌘	Nokia
Work item code:	⌘	PRESNC
Date:	⌘	17/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	New versions and RFCs published from internet drafts.
Summary of change:	⌘	This CR provides a IETF reference update for XCAP and data manipulation specific parts of the TS. New versions of I-Ds have been published and there are changes. The CR also corrects and aligns the normative texts and message flows accordingly.
Consequences if not approved:	⌘	Current references refer to old versions of IETF documents, and thus also at least partly to "old" functionality.

Clauses affected:	⌘	1; 2; 3, 4, 6, A.8			
Other specs affected:	⌘	**Y**	**N**		⌘
			X	Other core specifications	
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘				

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

1 Scope

The present document provides the protocol details for the presence service within the IP Multimedia (IM) Core Network (CN) subsystem based on the Session Initiation Protocol (SIP) and SIP Events as defined in 3GPP TS 24.229 [9].

Where possible the present document specifies the requirements for this protocol by reference to specifications produced by the IETF within the scope of SIP and SIP Events, either directly, or as modified by 3GPP TS 24.229 [9].

Requirements for manipulation of presence data are defined by use of a protocol at the Ut reference point based on XML Configuration Access Protocol (XCAP) (draft-ietf~~rosenberg~~-simple-xcap-04~~3~~ [33]**Error! Bookmark not defined.**).

The present document is applicable to Application Servers (ASs) and User Equipment (UE) providing presence functionality.

2 References

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

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

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

[2] 3GPP TS 22.141: "Presence Service; Stage 1".

[3] 3GPP TS 23.002: "Network architecture".

[4] 3GPP TS 23.141: "Presence service; Architecture and functional description; Stage 2".

[5] 3GPP TS 23.218: "IP Multimedia (IM) session handling; IM call model; Stage 2".

[6] 3GPP TS 23.228: "IP Multimedia Subsystem (IMS); Stage 2".

[7] 3GPP TS 24.109: "Bootstrapping interface (Ub) and Network application function interface (Ua); Protocol details".

[8] 3GPP TS 24.228: "Signalling flows for the IP multimedia call control based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3".

[9] 3GPP TS 24.229: "Internet Protocol (IP) multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3".

[10] 3GPP TS 29.228: "IP Multimedia (IM) Subsystem Cx and Dx Interfaces; Signalling flows and message contents".

[11] 3GPP TS 33.141: "Presence service; Security".

[12] IETF RFC 2664 (1999): "FYI on Questions and Answers - Answers to Commonly asked New Internet User Questions".

[13] IETF RFC 2246 (1999): "The TLS Protocol Version 1.0".

[14] IETF RFC 2387 (August 1998): "The MIME Multipart/Related Content-type".

[15] IETF RFC 2616 (June 1999): "Hypertext Transfer Protocol -- HTTP/1.1".

[15A] IETF RFC 2617 (June 1999): " HTTP Authentication: Basic and Digest Access Authentication".

[16] IETF RFC 2778 (2000): "A Model for Presence and Instant Messaging".

[17] IETF RFC 3261 (June 2002): "SIP: Session Initiation Protocol".

[18] IETF RFC 3263 (June 2002): "Session Initiation Protocol (SIP): Locating SIP Servers".

[19] IETF RFC 3265 (March 2002): "Session Initiation Protocol (SIP)-Specific Event Notification".

[20] IETF RFC 3310 (2002): "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)".

[21] draft-ietf-impp-cpim-pidf-08 (May 2003): "Presence Information Data Format (PIDF)".

[22] draft-ietf-simple-event-list-0 (June 2003): "A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists".

[23] draft-ietf-sip-publish-03 (February 2004): "An Event State Publication Extension to the Session Initiation Protocol (SIP)".

[24] draft-ietf-simple-partial-notify-01 (January 2004): "Partial Notification of Presence Information".

[25] draft-ietf-simple-prescaps-ext-00 (February 2004): "Device capability PIDF status extension".

[26] draft-ietf-simple-rpid-03 (March 2004): "RPID: Rich Presence~~:~~ Extensions to the Presence Information Data Format (PIDF)".

[27] draft-ietf-simple-presence-10 (January 2003): "A Presence Event Package for the Session Initiation Protocol (SIP)".

[28] draft-ietf-simple-winfo-package-05 (January 2003): "A Session Initiation Protocol (SIP) Event Template-Package for Watcher Information".

[29] draft-ietf-simple-winfo-format-04 (January 2003): "An Extensible Markup Language (XML) Based Format for Watcher Information".

[30] draft-ietf-simple-filter-format-00 (February 2004): "An Extensible Markup Language (XML) Based Format for Event Notification Filtering".

[31] draft-ietf-simple-event-filter-funct-00 (February 2003): "Functional Description of Event Notification Filtering".

[32] draft-ietf-simple-cipid-01 (March 2004): "CIPID: Contact Information in Presence Information Data Format".

[33] draft-ietf-simple-xcap-04~~3~~ (October~~July~~ 2004): "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)".

[34] draft-ietf~~isomaki~~-simple-xcap-pidf-manipulation-usage-02~~0~~ (October~~February~~ 2004): "An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usage for Manipulating Presence Document Contents".

[35] draft-ietf-simple-~~xcap-~~presence-rules-01~~0~~ (October~~May~~ 2004): "Presence Authorization Rules ~~Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usages for Setting Presence Authorization~~".

[36] draft-ietf-simple-xcap-list-usage-04~~2~~ (October~~February~~ 2004): "An Extensible Markup Language (XML) Formats for Representing Resource Lists".

[37] draft-ietf-geopriv-pidf-lo-01 (February 2004): "A Presence-based GEOPRIV Location Object Format".

[38] draft-ietf-simple-partial-pidf-format-00 (January 2004): "Presence Information Data Format (PIDF) Extension for Partial Presence".

[39] draft-ietf-simple-xcap-package-02 (July 2004): "A Session Initiation Protocol (SIP) Event Package for Modification Events for the Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Managed Documents".

[40] draft-ietf-sip-content-indirect-mech-03 (June 2003): "A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages".

[41] draft-rosenberg-simple-common-policy-caps-01 (July 2004): "An Extensible Markup Language (XML) Representation for Expressing Policy Capabilities".

[42] draft-rosenberg-simple-pres-policy-caps-01 (July 2004): "An Extensible Markup Language (XML) Representation for Expressing Presence Policy Capabilities".

[43] draft-ietf-sipping-config-framework-04 (July 2004): "A Framework for Session Initiation Protocol User Agent Profile Delivery".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply:

subscription authorization policy: a policy that determines which watchers are allowed to subscribe to a presentity's presence information
The subscription authorization policy also determines to which presence tuples of the presentity's presence information the watcher has access.

For the purposes of the present document, the following terms and definitions given in 3GPP TS 23.141 [4] apply:

Presence list server
Presence Network Agent (PNA)
Presence Server (PS)
Presence User Agent (PUA)

For the purposes of the present document, the following terms and definitions from RFC 2778 [16] apply:

Presence tuple
Presentity

For the purposes of the present document, the following terms and definitions from draft-ietf-sip-publish-03 [23] apply:

Event Publication Agent (EPA)
Event State Compositor (ESC)

For the purposes of the present document, the following terms and definitions from draft draft-ietf-simple-xcap-04~~3~~ [33] apply:

XCAP client
XCAP server

For the purposes of the present document, the following terms and definitions from draft-ietf-simple-event-list-04 [22] apply:

Resource List Server (RLS)

For the purposes of the present document, the following terms and definitions given in RFC 1594 [12].

Fully-Qualified Domain Name (FQDN)

For the purposes of the present document, the following terms and definitions given in RFC 3261 [17] apply (unless otherwise specified see clause 6).

Final response
Header
Header field
Method
Request
Response
(SIP) transaction
Status-code (see RFC 3261 [17], subclause 7.2)
Tag (see RFC 3261 [17], subclause 19.3)

For the purposes of the present document, the following terms and definitions given in 3GPP TS 23.002 [3], subclauses 4.1.1.1 and 4a.7 apply:

Call Session Control Function (CSCF)
Home Subscriber Server (HSS)

For the purposes of the present document, the following terms and definitions given in 3GPP TS 23.218 [5], subclause 3.1 apply:

Filter criteria
Initial filter criteria
Subsequent request

For the purposes of the present document, the following terms and definitions given in 3GPP TS 23.228 [6], subclauses 4.3.3.1 and 4.6 apply:

Interrogating-CSCF (I-CSCF)
Proxy-CSCF (P-CSCF)
Serving-CSCF (S-CSCF)

For the purposes of the present document, the following terms and definitions given in 3GPP TR 21.905 [1] apply:

User Equipment (UE)

For the purposes of the present document, the following terms and definitions from 3GPP TS 33.141 [11] apply:

Authentication Proxy

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AS	Application Server
AUID	Application Usage ID
CN	Core Network
CPIM	Common Profile for Instant Messaging
CSCF	Call Session Control Function
DM	Data Manipulator
DMS	Data Manipulation Server
EPA	Event Publication Agent
ESC	Event State Compositor
HSS	Home Subscriber Server
HTTP	HyperText Transfer Protocol
I-CSCF	Interrogating - CSCF
IM	IP Multimedia
IOI	Inter Operator Identifier
IP	Internet Protocol
MIME	Multipurpose Internet Mail Extensions
P-CSCF	Proxy - CSCF
PIDF	Presence Information Data Format
PNA	Presence Network Agent
PS	Presence Server
PSI	Public Service Identity

PUA	Presence User Agent
RLMI	Resource List Meta-Information
RLS	Resource List Server
RPID	Rich Presence Information Data
S-CSCF	Serving - CSCF
SIP	Session Initiation Protocol
TLS	Transport Layer Security
UE	User Equipment
URI	Universal Resource Identifier
XCAP	XML Configuration Access Protocol
XML	Extensible Markup Language

4 Presence service overview

The presence service provides the ability for the home network to manage presence information of a user's device, service or service media even whilst roaming. A user's presence information may be obtained through input from the user, information supplied by network entities or information supplied by elements external to the home network. Consumers of presence information, watchers, may be internal or external to the home network. The architecture for the 3GPP presence service is specified in 3GPP TS 23.141 [4].

SIP and XCAP provide means to manipulate the presence status of a user. For details on the differences between those means refer to draft-ietf-sip-publish-03 [23] and draft-~~isomaki~~ietf-simple-xcap-pidf-manipulation-usage-02~~0~~ [34]. For details on the relationship of DMS to other roles see subclause 6.2.2.

Editor's note: It may be appropriate to include text in this clause pointing to the stage 1 on group management.

*** NEXT CHANGE ***

6.3 Roles

6.3.1 Data Manipulator (DM)

6.3.1.1 Introduction

The DM is a logical function that implements the requirements of a XCAP client as defined in draft-ietf-simple-xcap-04~~3~~ [33]. The DM provides the means to manipulate the general data such us user groups, subscription authorization policy, resource lists, hard state presence publication, MIME objects referenced from the hard state presence information, etc.

NOTE: In order to be able to manipulate data stored on the DMS, the DM has the root directory on the DMS pre-configured or use some means to discover it. Discovery mechanisms are outside the scope of the present document.

6.3.1.2 Manipulating a presencelist

When the DM intends to manipulate a presencelist, it shall generate an HTTP PUT, GET or DELETE request in accordance with RFC 2616 [15], draft-ietf-simple-xcap-04~~3~~ [33] and draft-ietf-simple-xcap-list-usage-04~~2~~ [36].

6.3.1.3 Manipulating the subscription authorization policy

When the DM intends to manipulate the subscription authorization policy, it shall generate an HTTP PUT, HTTP GET or HTTP DELETE request in accordance with RFC 2616 [15], draft-ietf-simple-xcap-04~~3~~ [33] and draft-ietf-simple-~~xcap-~~presence-rules-01~~0~~ [35].

The DM may use an HTTP GET in accordance with RFC 2616 [15], draft-ietf-simple-xcap-04~~3~~ [33] and draft-rosenberg-simple-common-policy-caps-01 [42] for fetching of the authorization policy capabilities which the DMS supports.

When the DM intends to authorize a different value of the same presence attribute to different watchers or watcher groups, the DM shall authorize a single tuple including one of the different values of the same presence attribute to every watcher or watcher groups by using a specific "inclusion set" as specified in draft-ietf-simple-~~xcap-~~presence-rules-01~~0~~ [35].

6.3.1.4 Publishing hard state presence information

The DM shall implement draft-ietf~~isomaki-~~simple-xcap-pidf-manipulation-usage-02~~0~~ [34] in order to be able to manipulate hard state presence information. Hard state presence information uses the same format as soft state information, namely "application/pidf+xml" content type as described in ~~draft-ietf-impp-cpim-pidf-08~~RFC 3863 [21] together with any of its extensions.

When the hard state presence information contains one ore more MIME objects to be aggregated with the "application/pidf+xml" content type and any of its extensions, the DM shall:

a) construct as many HTTP URIs as many objects to be stored and formulate every HTTP URI according a predefined directory structure;

NOTE: In order to be able to manipulate data stored on the DMS, the DM has the root directory on the DMS pre-configured or use some means to discover it. Discovery mechanisms are outside the scope of the present document.

b) store the objects on the data manipulation server behind the HTTP URI(s) created in the previous step using standard HTTP procedures as defined in RFC 2616 [15];

c) include every HTTP URI as a value of the corresponding XML element in the published "application/pidf+xml" presence document referencing the stored object(s) in the previous step; and

d) publish the hard state presence information according to draft-ietf~~isomaki-~~simple-xcap-pidf-manipulation-usage-02~~0~~ [34].

6.3.2 Data Manipulation Server (DMS)

6.3.2.1 Introduction

The Data Manipulation Server (DMS) is a logical function that implements the requirements of a XCAP server as defined in draft-ietf-simple-xcap-04~~3~~ [33]. The DMS can store data such us user groups, subscription authorization policy, resource lists, hard state presence information, MIME objects referenced from the hard state presence information, etc.

6.3.2.2 Resource list manipulation acceptance

When the data manipulation server receives an HTTP PUT, HTTP GET or HTTP DELETE request for manipulating or fetching a resource list, the DMS shall first authenticate the request in accordance with 3GPP TS 24.109 [7] and then perform authorization. Afterwards the DMS shall perform the requested action and generate a response in accordance with RFC 2616 [15], draft-ietf-simple-xcap-04~~3~~ [33] and draft-ietf-simple-xcap-list-usage-04~~2~~ [36].

6.3.2.3 Subscription authorization policy manipulation acceptance

When the DMS receives an HTTP PUT, HTTP GET or HTTP DELETE request for manipulating or fetching of the subscription authorization policy, the data manipulation server shall first authenticate the request in accordance with 3GPP TS 24.109 [7] and then perform authorization. Afterwards the DMS shall perform the requested action and

generate a response in accordance with RFC 2616 [15], draft-ietf-simple-xcap-04~~3~~ [33] and draft-ietf-simple-~~xcap~~-presence-rules-01~~0~~ [35].

When the DMS receives an HTTP GET request for fetching of the authorization policy capabilities information, the DMS shall generate a response in accordance with RFC 2616 [15], draft-ietf-simple-xcap-0~~2~~4 [33] and draft-rosenberg-simple-pres-policy-caps-~~00~~ 01 [42].

6.3.2.4 Publication acceptance of hard state presence information

When the DMS receives an HTTP PUT, HTTP GET or HTTP DELETE request for publishing, fetching or deleting of hard state presence information, the DMS shall first authenticate the request in accordance with 3GPP TS 24.109 [7] and then perform authorization. Afterwards the DMS shall:

a) if the HTTP URI points to a predefined directory reserved for storing MIME objects and the request is an HTTP PUT request, replace any existing content referenced by the Request-URI with the content of the request;

b) if the Request-URI points to an uncreated directory, create the directory, store the content there and associate the content with the Request-URI. For all requests, i.e. HTTP PUT, HTTP GET and HTTP DELETE requests, generate an appropriate response in accordance with RFC 2616 [15]; or

c) if the HTTP URI points to an XCAP directory and the Application Usage ID (AUID) part of the HTTP URI is set to "pidf-manipulation", process the request and generate an appropriate response in accordance with draft-ietf-simple-xcap-04~~3~~ [33], draft-ietf-~~isomaki~~ simple-xcap-pidf-manipulation-usage-02~~0~~ [34] and RFC 2616 [15].

*** NEXT CHANGES ****

A.8 Example signalling flows of HTTP based presence service operation

A.8.1 Introduction

This subclause shows signalling flows relating to the manipulation of presence service data over the Ut reference point using XCAP.

Each example signalling flow shows several sequences of manipulation of data for the presence service.

NOTE: Error conditions are not considered in the examples e.g. if authorization checks fail in the XCAP server, XML Schema compliancy checks fail or the file specified by the URI does not exist then an appropriate 4xx response is sent to the client.

Editor's note: Clarifications how XCAP is using HTTP is needed.

A.8.2 Signalling flows demonstrating how DMs manipulate resource lists

Editor's note: The possible proxies (e.g., handling authentication matters) between the data manipulator client and HTTP server are bypassed. Also the authentication related headers are missing.

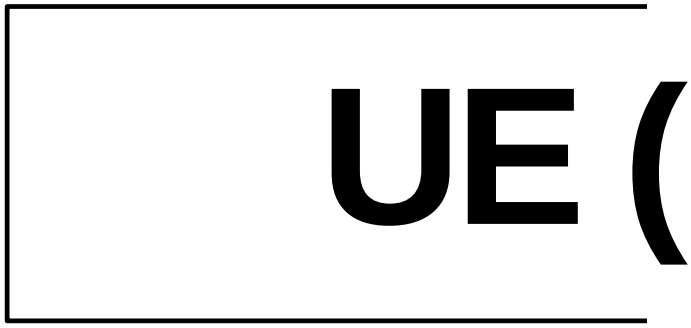


Figure A.8.2-1: DM manipulating a resource list on DMS

Figure A.8.2-1 shows a how a DM may manipulate a resource list on a DMS. The details of the signalling flows are as follows:

1. **XCAP PUT request (DM to DMS - see example in table A.8.2-1**

   The DM generates an XCAP PUT request to create a new resource list on the DMS. The resource list has one entry.

Table A.8.2-1: XCAP PUT request (DM to DMS)

```
PUT http://xcap.home1.net/services/resource-lists/users/user1/pf.xml HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 10:13:17 GMT
Content-Type: application/resource-lists+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
   <resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="urn:ietf:params:xml:ns:resource-lists">
     <list name="Presence_fellows" uri="sip:user1_list1@home1.net" subscribeable="true">
       <entry name="user2" uri="sip:user2_public1@home2.net">
         <display-name>User2</display-name>
       </entry>
     </list>
   </resource-lists>
```

2. **XCAP 201 (Created) response (DMS to DM) – see example in table A.8.2-2**

After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to create a file, the DMS sends an XCAP 201 (Created) response to the DM.

Table A.8.2-2: XCAP 201 (Created) response (DMS to DM)

```
HTTP/1.1 201 CreatedREATED
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Etag: "aaa"
Date: Thu, 08 Jan 2004 10:50:35 GMT
Content-Type: text/html
Content-Length: 0
```

3. **XCAP PUT request (DM to DMS) – see example in table A.8.2-3**

The DM adds a new entry to the previously created resource list by generating a new XCAP PUT request.

Table A.8.2-3: XCAP PUT request (DM to DMS)

```
PUT http://xcap.home1.net/services/resource-lists/users/user1/pf.xml/~~/?resource-
lists/list%5b[@name=%22"Presence_fellows%22"%5d]/entry HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 10:13:17 GMT
Content-Type: application/xcap-elxml-fragment-body
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
         <entry name="user3" uri="sip:user3_public1@home3.net">
           <display-name>User3</display-name>
         </entry>
```

4. **XCAP 200 (OK) response (DMS to DM) - see example in table A.8.2-4**

After the DMS has performed the necessary authorization checks, XML document validations and XML schema compliancy checks the DMS sends an XCAP 2001 (OKCreated) response to the DM.

Table A.8.2-4: XCAP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Etag: "aab"
Date: Thu, 08 Jan 2004 10:50:45 GMT
Content-Type: text/html
Content-Length: 0
```

5. **XCAP DELETE request (DM to DMS) - see example in table A.8.2-5**

The DM decides to delete the entry "user2" from the resource list. The DM generates an XCAP DELETE request.

Table A.8.2-5: XCAP DELETE request (DM to DMS)

```
DELETE http://xcap.home1.net/services/resource-lists/users/user1/pf.xml/~~/?resource-
lists/list%5b[@name=%22"Presence_fellows%22"%5d]/entry%5b[@name=%22user2%22"%5d] HTTP/1.1
Host: oper.example.com:9999
User-Agent: IMS subscriber
Date: Thu, 08 Jan 2004 10:14:17 GMT
Referer: http://oper.home1.net:1234/service
```

6. **XCAP 200 (OK) response (DMS to DM) – see example in table A.8.2-6**

After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to delete an entry from the resource list, the DMS sends an XCAP 200 (OK) response.

Table A.8.2-6: XCAP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 11:00:47 GMT
Content-Type: image/jpeg
Content-Length: 0
```

7. **XCAP GET request (DM to DMS) – see example in table A.8.2-7**

The DM wishes to check the result of the previous transaction by generating an XCAP GET request.

Table A.8.2-7: XCAP GET request (DM to DMS)

```
GET http://xcap.home1.net/services/resource-lists/users/user1/pf.xml HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 11:13:17 GMT
Content-Length: 0
```

8. **XCAP 200 (OK) response (DMS to DM) - see example in table A.8.2-8**

After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to fetch the resource list, the DMS sends an XCAP 200 (OK) response to the DM including the resource list in the body of the response.

Table A.8.2-8: XCAP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Etag: "askdajdsaj"
Date: Thu, 08 Jan 2004 11:50:35 GMT
Content-Type:application/resource-lists+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
   <resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:schemaLocation="urn:ietf:params:xml:ns:resource-lists">
     <list name="Presence_fellows" uri="sip:user1_list1@home1.net" subscribeable="true">
       <entry name="user3" uri="sip:user3_public1@home3.net">
         <display-name>User3</display-name>
       </entry>
     </list>
   </resource-lists>
```

A.8.3 Signalling flows demonstrating how DMs manipulate presence authorization policy

Editor's note: The possible proxies (e.g., handling authentication matters) between the data manipulator client and HTTP server are bypassed. Also the authentication related headers are missing.

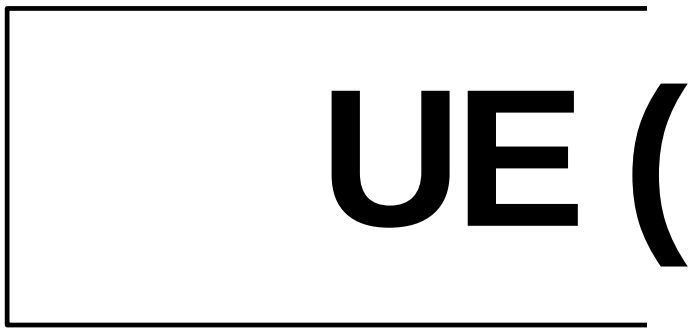


Figure A.8.3-1: DM manipulating presence authorization policy on DMS

Figure A.8.3-1 shows a DM manipulating presence authorization policy on a DMS. The details of the signalling flows are as follows:

1. **XCAP PUT request (DM to DMS) – see example in table A.8.3-1**

   The DM generates an XCAP PUT request to create a presence authorization policy on the DMS. The presence authorization policy has one ~~permission statement~~rule allowing for sip:user2_public1@home2.net to see all service information ~~from the basic PIDF namespace~~ along with the service related prescaps~~"video"~~ elements~~ from the prescaps namespace~~ defined in draft-ietf-simple-prescaps-ext-02 [25].

Table A.8.3-1: XCAP PUT request (DM to DMS)

```
PUT http://xcap.home1.net/services/pres-rules~~permission-statements~~/users/user1/ps.xml HTTP/1.1
User-Agent: IMS subscriber
~~Referer: http://oper.home1.net:1234/service~~
Date: Thu, 08 Jan 2004 10:13:17 GMT
Content-Type:application/auth-policy~~permission-statements~~+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
   <ruleset~~permission-statements~~ xmlns="urn:ietf:params:xml:ns:common-policy~~permission-statements~~"
      ~~xmlns:pidf="urn:ietf:params:xml:ns:pidf"~~
      xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps"
      xmlns:pr~~escaps~~="urn:ietf:params:xml:ns:pres-rules"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                xsi:schemaLocation="urn:ietf:params:xml:ns:pres-rules">~~simple-prescaps-ext">~~
      <rule~~statement~~ id="dsafa43232">
```

```
        <conditions>
          <applies-toidentity>
            <iduri>sip:user2_public1@home2.net</iduri>
          </identityapplies-to>
        <conditions>
        <actionspermissions>
           <pr:sub-handling>allow</pr:sub-handling>
          <accept/>
          <show-namespace>urn:ietf:params:xml:ns:pidf</show-namespace>
          <show-element>prescaps:video</show-element>
        </actionspermissions>
        <transformations>
           <pr:provide-services>
              <pr:all-services/>
              <pr:provide-person>true</pr:provide-person>
              <pr:provide-unknown-attribute name="sc:servcaps">true</pr:provide-unknown-
attribute>
           </pr:provide-services>
        </transformations>

     </rulestatement>
   </rulesetpermission-statements>
```

2. **XCAP 201 (Created) response (DMS to DM) - see example in table A.8.3-2**

After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to create a file, the DMS sends an XCAP 201 (Created) response to the DM.

Table A.8.3-2: XCAP 201 (Created) response (DMS to DM)

```
HTTP/1.1 201 CreatedREATED
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Etag: "bbb"
Date: Thu, 08 Jan 2004 10:50:35 GMT
Content-Type: text/html
Content-Length: 0
```

3. **XCAP PUT request (DM to DMS) – see example in table A.8.3-3**

The DM adds a new rulepermission statement to the previously created presence authorization policy by generating a new XCAP request. The new permission statementrule blocks allows the user named sip:user3_public1@home3.net to see presence informationthe tuple with class element specifying "sip".

Table A.8.3-3: XCAP PUT request (DM to DMS)

```
PUT http://xcap.home1.net/services/pres-rulespermission-
      statements/users/user1/ps.xml/~~/permission-statementsruleset/rule%5b2%5d HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 10:13:27 GMT
Content-Type: application/xcap-elxml-fragment-body
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
     <rule id="fdsjkf">
        <conditions>
          <identity>
            <id>user3_public1@home2.net</id>
          </identity>
        <conditions>
        <actions>
           <pr:sub-handling>block</pr:sub-handling>
        </actions>
```

```
    <statement id="dsffdsfrrr32423">
      <applies-to>
        <uri>sip:user3_public1@home3.net</uri>
      </applies-to>

      <permissions>
        <accept/>
        <show-tuple>sip</show-tuple>
      </permissions>

    </statement>
```

4. **XCAP 200 (OK) response (DMS to DM) - see example in table A.8.3-4**

After the DMS has performed the necessary authorization checks, XML document validations and XML schema compliancy checks the DMS sends an XCAP 2001 (OKCreated) response to the DM.

Table A.8.3-4: XCAP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Etag: "bbb2"
Date: Thu, 08 Jan 2004 10:50:45 GMT
Content-Type: text/html
Content-Length: 0
```

5. **XCAP DELETE request (DM to DMS) - see example in table A.8.3-5**

The DM decides to delete the rulepermission statement for sip:user2_public1@home2.net from the authorization policy. The DM generates an XCAP DELETE request.

Table A.8.3-5: XCAP DELETE request (DM to DMS)

```
DELETE http://xcap.home1.net/services/pres-rulespresence-
      lists/users/user1/ps.xml/~~/permission-
      statementsruleset/rulestatement[@id="dsafa43232"]/permissions/show-namespace HTTP/1.1
Host: oper.example.com:9999
User-Agent: IMS subscriber
Date: Thu, 08 Jan 2004 10:14:17 GMT
Referer: http://oper.home1.net:1234/service
```

6. **XCAP 200 (OK) response (DMS to DM) – see example in table A.8.3-6**

After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to delete an entry from the resource list, the DMS sends an XCAP 200 (OK) response.

Table A.8.3-6: XCAP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 11:00:47 GMT
Content-Length: 0
```

7. **XCAP GET request (DM to DMS) – see example in table A.8.3-7**

The DM wishes to check the result of the previous transaction by generating an XCAP GET request.

Table A.8.3-7: XCAP GET request (DM to DMS)

```
GET http://xcap.home1.net/services/pres-rulespermission-statements/users/user1/ps.xml HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 11:13:17 GMT
Content-Length: 0
```

8. **XCAP 200 (OK) response (DMS to DM) – see example in table A.8.3-8**

After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to fetch the resource list, the DMS sends an XCAP 200 (OK) response to the DM including the authorization rules~~resource list~~ in the body of the response.

Table A.8.3-8: XCAP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Etag: "eiuuekksks"
Date: Thu, 08 Jan 2004 11:50:35 GMT
Content-Type:application/auth-policypermission-statements+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
   <ruleset xmlns="urn:ietf:params:xml:ns:common-policy"
     xmlns:sc="urn:ietf:params:xml:ns:pidf:servcaps"
     xmlns:pr="urn:ietf:params:xml:ns:pres-rules"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="urn:ietf:params:xml:ns:pres-rules">
     <rule id="fdsjkf">
       <conditions>
         <identity>
           <id>user3_public1@home2.net</id>
         </identity>
       <conditions>
       <actions>
          <pr:sub-handling>block</pr:sub-handling>
       </actions>
   </ruleset>    <permission-statements xmlns="urn:ietf:params:xml:ns:permission-statements"
     xmlns:pidf="urn:ietf:params:xml:ns:pidf"
     xmlns:prescaps=" urn:ietf:params:xml:ns:simple-prescaps-ext">
     <statement id="dsffdsfrrr32423">
       <applies-to>
         <uri>sip:user3_public1@home3.net</uri>
       </applies-to>
       <permissions>
         <accept/>
         <show-tuple>sip</show-tuple>
       </permissions>
     </statement>
   </permission-statements>
```

A.8.4 Storing external content (successful operation)

Editor's note: The possible proxies (e.g., handling authentication matters) between the data manipulator client and HTTP server are bypassed. Also the authentication related headers are missing.

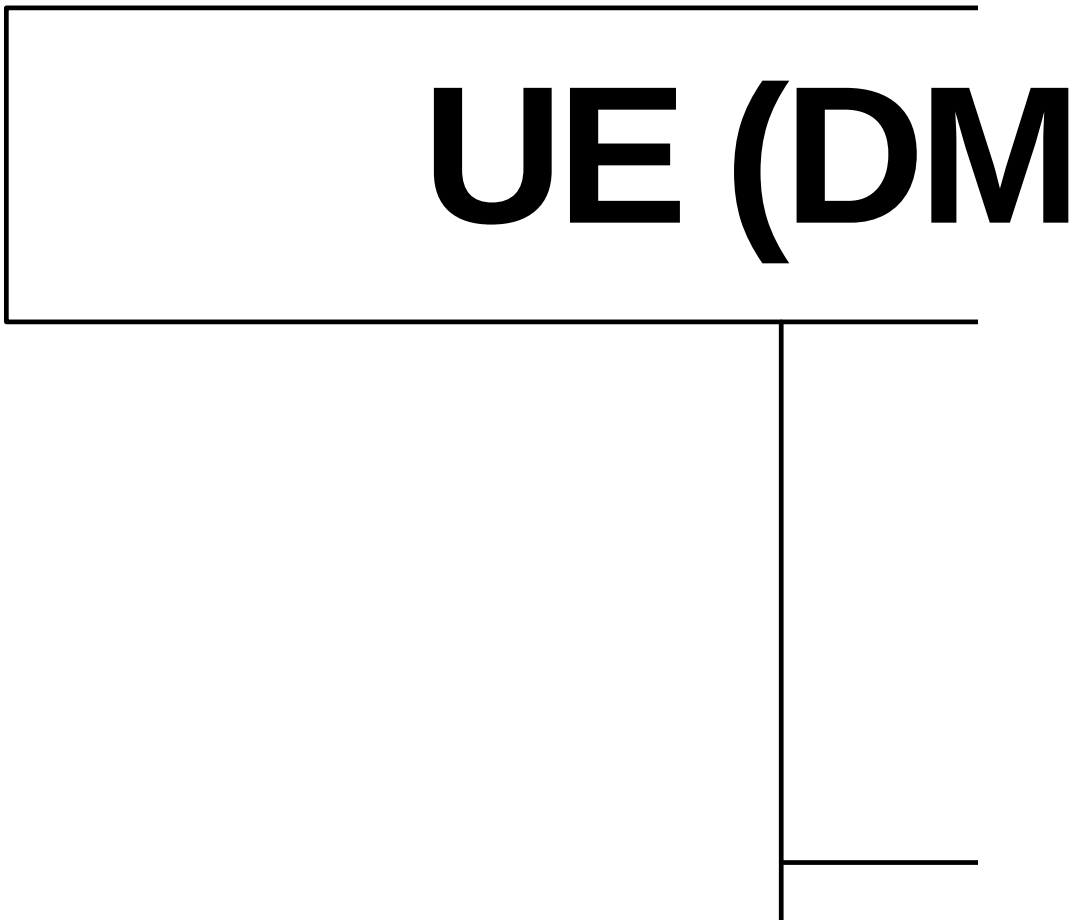


Figure A.8.4.-1: DM manipulating hard-state presence document on DMS

Figure A.8.4-1 shows a DM manipulating hard-state presence document on a DMS when the presence document has an aggregated storing MIME object with the "application/pidf+xml" content type and any of its extensions. The details of the signalling flows are as follows:

1. **HTTP PUT request (DM (client) to DMS) – see example in table A.8.2-1**

   In order to store the content, the DM generates an HTTP PUT request containing the MIME object in the body of the request. The request-URI points to the directory where the content is stored and shows the name of the file to be created.

Table A.8.4-1: HTTP PUT request (DM to DMS)

```
PUT http://operator.example.com/services/users/bill/pictureX HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 10:13:17 GMT
Content-Type: image/jpeg
Content-Length: (…)

{pictureX.jpg}
```

2. **HTTP 201 (Created) response (DMS to DM) – see example in table A.8.4-2**

   After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to create a file the HTTP server sends an HTTP 201 (Created) response to the client.

Table A.8.4-2: HTTP 201 (Created) response (DMS to DM)

```
HTTP/1.1 201 CreatedREATED
Server: Apache/1.3.22 (Unix) mod_perl/1.27Content-Type: text/html
Date: Thu, 08 Jan 2004 10:50:35 GMT
Content-Length: 01234
```

3. **XCAP PUT request (DM to DMS) - see example in table A.8.2-3**

   The DM generates an XCAP PUT request in order to store XML encoded presence document which includes a URI reference to the MIME object stored on the DMS. The AUID part of the request URI is 'pidf-manipulation' as defined in draft-ietfisomaki-simple-xcap-pidf-manipulation-usage-020 [34].

Table A.8.4-3: XCAP PUT request (DM to DMS)

```
PUT http://xcap.example.com/services/pidf-manipulation/users/bill/pidf.xml HTTP/1.1
User-Agent: IMS subscriber
Referer: http://xcap.home1.net:1234/service
Date: Thu, 08 Jan 2004 10:13:27 GMT
Content-Type: application/pidf+xml
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
   <presence xmlns="urn:ietf:params:xml:ns:cpim-pidf"
          xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
          xmlns:ext="urn:ietf:params:xml:ns:ext-cont"
          xmlns:p="urn:ietf.params:xml:ns:pidf:person"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="urn:ietf:params:xml:ns:resource-lists"
          entity="sip:bill@example.com">

     <tuple id="123sd">
       <status>
         <basic>open</basic>
       </status>
       <et:type>service</et:type>
         <contact>sip:bill@example.com</contact>
     </tuple>

     <p:persontuple id="432sd">
       <p:status>
         <et:activitiesbasic><et:vacation/>open</et:activitiesbasic>
       </p:status>
       <et:type>presentity</et:type>
```

```
        <ext:photo>
          http://operator.example.com/services/users/bill/pictureX.jpg
        </ext:photo>
        <note xml:lang="en">At home</note>
      </p:persontuple>
    </presence>
```

4. **XCAP 201 (CreatedREATED) response (DMS to DM) - see example in table A.8.4-4**

After the DMS has performed the necessary authorization checks, XML document validations and XML schema compliancy checks the DMS sends an XCAP 201 (Created) response to the DM.

Table A.8.4-4: XCAP 201 (Created) response (DMS to DM)

```
HTTP/1.1 201 CreatedREATED
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Etag: "ccc1"
Date: Thu, 08 Jan 2004 10:50:45 GMT
Content-Length: 0
```

5. **HTTP GET request (DM to DMS) – see example in table A.8.4-5**

The DM wishes to fetch the MIME object from the DMS. The client generates an HTTP GET request. The request URI points to the directory where the object is stored and indicates the name of the file to be fetched.

Table A.8.4-5: HTTP GET request (DM to DMS)

```
GET http://operator.example.com/services/users/bill/pictureX HTTP/1.1
Host: oper.example.com:9999
User-Agent: IMS subscriber
Date: Thu, 08 Jan 2004 10:43:17 GMT
Accept: image/jpeg
Referer: http://oper.home1.net:1234/service
Content-Length: 0
```

6. **HTTP 200 (OK) response (DMS to DM) – see example in table A.8.4-6**

After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to fetch the file the DMS sends an HTTP 200 (OK) response having the object in the body to the DM.

Table A.8.4-6: HTTP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 11:00:47 GMT
Content-Type: image/jpeg
Content-Length: (…)

{pictureX}
```

7. **HTTP PUT request (DM to DMS) – see example in table A.8.4-7**

The DM wishes to modify the earlier stored MIME object by replacing the picture X with a new picture X with new content. To modify the object the DM generates an HTTP PUT request using the same request URI as has been used for the modified (old) object. The new object is conveyed in the body of the request.

Table A.8.4-7: HTTP PUT request (DM to DMS)

```
PUT http://operator.example.com/services/users/bill/pictureX HTTP/1.1
User-Agent: IMS subscriber
Referer: http://oper.home1.net:1234/service
Date: Thu, 08 Jan 2004 11:13:17 GMT
Content-Type: image/jpeg
Content-Length: (…)

{pictureX.jpg}
```

8. **HTTP 200 (OK) response (DMS to DM) – see example in table A.8.4-8**

After the DMS has performed the necessary authorization checks on the originator to ensure the DM is allowed to replace the existing MIME object with the new one the DMS sends an HTTP 200 (OK) response to the DM.

Table A.8.4-8: HTTP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 11:50:35 GMT
Content-Length: 0
```

9. **XCAP PUT request (DM to DMS) – see example in table A.8.4-9**

The DM wishes to remove the MIME object from his presence information. The DM generates an XCAP PUT request to modify the XML encoded presence document to remove the reference to the MIME object from the presence document. The request URI contains a node selector to the requested tuple according to draft-ietf-simple-xcap-042 [33]. Because the signalling flow does not contain the XCAP GET request the use of the If-Match header is omitted in this example.

Table A.8.4-9: XCAP PUT request (DM to DMS)

```
PUT http://xcap.example.com/services/pidf-
  manipulation/users/bill/pidf.xml/~~/?presence/persontuple[@id='432sd'] HTTP/1.1
Date: Thu, 08 Jan 2004 11:13:37 GMT
If-Match: "ccc1"
Content-Type: application/xcap-eltext/plain
Content-Length: (…)

<?xml version="1.0" encoding="UTF-8"?>
   <persontuple id="432sd">
     <p:status>
         <et:activities><et:vacation/></et:activities>
       <basic>open</basic>
     </p:status>
     <et:type>presentity</et:type>
     <note xml:lang="en">At home</note>
   </persontuple>
```

10. **XCAP 200 (OK) response (DMS to DM) - see example in table A.8.4-10**

After the DMS has performed the necessary authorization checks, XML document validations and XML Schema compliancy checks the DMS sends an XCAP 200 (OK) response to the DM.

Table A.8.4-10: XCAP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Etag: "ccc2"
Date: Thu, 08 Jan 2004 11:50:59 GMT
Content-Length: 0
```

11. **HTTP DELETE request (DM to DMS) – see example in table A.8.4-11**

The DM removes the MIME object from the DMS by generating an HTTP DELETE request.

Table A.8.4-11: HTTP DELETE request (DM to DMS)

```
DELETE http://operator.example.com/services/users/bill/pictureX HTTP/1.1
Host: oper.example.com:9999
User-Agent: IMS subscriber
Date: Thu, 08 Jan 2004 11:52:00 GMT
Referer: http://oper.home1.net:1234/service
```

12. **HTTP 200 (OK) response (DMS to DM) – see example in table A.8.4-12**

After the DMS has performed the necessary authorization checks on the originator to ensure that the DM is allowed to delete the object, the DMS sends an HTTP 200 (OK) response to the DM.

Table A.8.4-12: HTTP 200 (OK) response (DMS to DM)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 11:52:35 GMT
Content-Length: 0
```

3GPP TSG-CN1 Meeting #36 **Tdoc N1-042006**
Seoul, Korea, 15-19 November 2004

CR-Form-v7.1

CHANGE REQUEST

⌘ **24.141** **CR** **25** ⌘ **rev** **1** ⌘ Current version: **6.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	IETF reference update (SIP specific parts)
Source:	⌘	Nokia
Work item code:	⌘	PRESNC
Date:	⌘	16/11/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	New versions and RFCs published from internet drafts.
Summary of change:	⌘	This CR provides a IETF reference update for SIP specific parts of the TS. Several new versions of I-Ds have been published and there are changes in the solutions. The CR also corrects and aligns the normative texts and message flows accordingly.
Consequences if not approved:	⌘	Current references refer to old versions of IETF documents, and thus also at least partly to "old" functionality.

Clauses affected:	⌘	2; 3; 4; 5.3; A3; A4; A5

Other specs affected:	⌘	Y	N		⌘
			X	Other core specifications	
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	Reference to draft-ietf-simple-presence-data-model is added in CR 24.

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

*** 1st CHANGE ****

2 References

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

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

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

[2] 3GPP TS 22.141: "Presence Service; Stage 1".

[3] 3GPP TS 23.002: "Network architecture".

[4] 3GPP TS 23.141: "Presence service; Architecture and functional description; Stage 2".

[5] 3GPP TS 23.218: "IP Multimedia (IM) session handling; IM call model; Stage 2".

[6] 3GPP TS 23.228: "IP Multimedia Subsystem (IMS); Stage 2".

[7] 3GPP TS 24.109: "Bootstrapping interface (Ub) and Network application function interface (Ua); Protocol details".

[8] 3GPP TS 24.228: "Signalling flows for the IP multimedia call control based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3".

[9] 3GPP TS 24.229: "Internet Protocol (IP) multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3".

[10] 3GPP TS 29.228: "IP Multimedia (IM) Subsystem Cx and Dx Interfaces; Signalling flows and message contents".

[11] 3GPP TS 33.141: "Presence service; Security".

[12] IETF RFC 2664 (1999): "FYI on Questions and Answers - Answers to Commonly asked New Internet User Questions".

[13] IETF RFC 2246 (1999): "The TLS Protocol Version 1.0".

[14] IETF RFC 2387 (August 1998): "The MIME Multipart/Related Content-type".

[15] IETF RFC 2616 (June 1999): "Hypertext Transfer Protocol -- HTTP/1.1".

[15A] IETF RFC 2617 (June 1999): " HTTP Authentication: Basic and Digest Access Authentication".

[16] IETF RFC 2778 (2000): "A Model for Presence and Instant Messaging".

[17] IETF RFC 3261 (June 2002): "SIP: Session Initiation Protocol".

[18] IETF RFC 3263 (June 2002): "Session Initiation Protocol (SIP): Locating SIP Servers".

[19] IETF RFC 3265 (March 2002): "Session Initiation Protocol (SIP)-Specific Event Notification".

[20] IETF RFC 3310 (2002): "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)".

[21] IETF RFC 3863draft-ietf-impp-cpim-pidf-08 (August 2004May 2003): "Presence Information Data Format (PIDF)".

[22] draft-ietf-simple-event-list-054 (October 2004June 2003): "A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists".

[23] IETF RFC 3903draft-ietf-sip-publish-03 (FebruaryOctober 2004): "An Event State Publication Extension to the Session Initiation Protocol (SIP) for Event State Publication ".

[24] draft-ietf-simple-partial-notify-031 (OctoberJanuary 2004): "Partial Notification of Presence Information".

[25] draft-ietf-simple-prescaps-ext-020 (OctoberFebruary 2004): "User Agent Capability Extension to Presence Information Data Format (PIDF) Device capability PIDF status extension".

[26] draft-ietf-simple-rpid-043 (OctoberMarch 2004): "RPID: Rich Presence: Extensions to the Presence Information Data Format (PIDF)".

[27] IETF RFC 3856 (August 2004)draft-ietf-simple-presence-10 (January 2003): "A Presence Event Package for the Session Initiation Protocol (SIP)".

[28] IETF RFC 3857 (August 2004): "A Watcher Information Event Template-Package for the Session Initiation Protocol (SIP)" draft-ietf-simple-winfo-package-05 (January 2003): "A Session Initiation Protocol (SIP) Event Template-Package for Watcher Information".

[29] IETF RFC 3858 (August 2004): draft-ietf-simple-winfo-format-04 (January 2003): "An Extensible Markup Language (XML) Based Format for Watcher Information".

[30] draft-ietf-simple-filter-format-030 (OctoberFebruary 2004): "An Extensible Markup Language (XML) Based Format for Event Notification Filtering".

[31] draft-ietf-simple-event-filter-funct-030 (October 2004February 2003): "Functional Description of Event Notification Filtering".

[32] draft-ietf-simple-cipid-031 (JulyMarch 2004): "CIPID: Contact Information in Presence Information Data Format".

[33] draft-ietf-simple-xcap-03 (July 2004): "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)".

[34] draft-isomaki-simple-xcap-pidf-manipulation-usage-00 (February 2004): "An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usage for Manipulating Presence Document Contents".

[35] draft-ietf-simple-xcap-presence-rules-010 (OctoberMay 2004): "Presence Authorization Rules Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usages for Setting Presence Authorization".

[36] draft-ietf-simple-xcap-list-usage-02 (February 2004): "An Extensible Markup Language (XML) Format for Representing Resource Lists".

[37] draft-ietf-geopriv-pidf-lo-031 (SeptemberFebruary 2004): "A Presence-based GEOPRIV Location Object Format".

[38] draft-ietf-simple-partial-pidf-format-020 (JanuaryOctober 2004): "Presence Information Data Format (PIDF) Extension for Partial Presence".

[39] draft-ietf-simple-xcap-package-02 (July 2004): "An Extensible Markup Language (XML) Document Format for Indicating Changes in XML Configuration Access Protocol (**XCAP**) ResourcesA Session Initiation Protocol (SIP) Event Package for Modification Events for the Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Managed Documents".

[40] draft-ietf-sip-content-indirect-mech-053 (October 2004June 2003): "A Mechanism for Content Indirection in Session Initiation Protocol (SIP) Messages".

[41] draft-rosenberg-simple-common-policy-caps-01 (July 2004): "An Extensible Markup Language (XML) Representation for Expressing Policy Capabilities".

[42] draft-rosenberg-simple-pres-policy-caps-01 (July 2004): "An Extensible Markup Language (XML) Representation for Expressing Presence Policy Capabilities".

[43] draft-ietf-sipping-config-framework-05~~4~~ (October~~July~~ 2004): "A Framework for Session Initiation Protocol User Agent Profile Delivery".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply:

subscription authorization policy: a policy that determines which watchers are allowed to subscribe to a presentity's presence information
The subscription authorization policy also determines to which ~~presence tuples of the~~ presentity's presence information the watcher has access.

For the purposes of the present document, the following terms and definitions given in 3GPP TS 23.141 [4] apply:

Presence list server
Presence Network Agent (PNA)
Presence Server (PS)
Presence User Agent (PUA)

For the purposes of the present document, the following terms and definitions from RFC 2778 [16] apply:

Presence tuple
Presentity

For the purposes of the present document, the following terms and definitions from ~~draft-ietf-sip-publish-03~~ RFC 3903 [23] apply:

Event Publication Agent (EPA)
Event State Compositor (ESC)

For the purposes of the present document, the following terms and definitions from draft draft-ietf-simple-xcap-03 [33] apply:

XCAP client
XCAP server

For the purposes of the present document, the following terms and definitions from draft-ietf-simple-event-list-05~~4~~ [22] apply:

Resource List Server (RLS)

For the purposes of the present document, the following terms and definitions given in RFC 1594 [12].

Fully-Qualified Domain Name (FQDN)

For the purposes of the present document, the following terms and definitions given in RFC 3261 [17] apply (unless otherwise specified see clause 6).

Final response
Header
Header field
Method

Request
Response
(SIP) transaction
Status-code (see RFC 3261 [17], subclause 7.2)
Tag (see RFC 3261 [17], subclause 19.3)

For the purposes of the present document, the following terms and definitions given in 3GPP TS 23.002 [3], subclauses 4.1.1.1 and 4a.7 apply:

Call Session Control Function (CSCF)
Home Subscriber Server (HSS)

For the purposes of the present document, the following terms and definitions given in 3GPP TS 23.218 [5], subclause 3.1 apply:

Filter criteria
Initial filter criteria
Subsequent request

For the purposes of the present document, the following terms and definitions given in 3GPP TS 23.228 [6], subclauses 4.3.3.1 and 4.6 apply:

Interrogating-CSCF (I-CSCF)
Proxy-CSCF (P-CSCF)
Serving-CSCF (S-CSCF)

For the purposes of the present document, the following terms and definitions given in 3GPP TR 21.905 [1] apply:

User Equipment (UE)

For the purposes of the present document, the following terms and definitions from 3GPP TS 33.141 [11] apply:

Authentication Proxy

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AS	Application Server
AUID	Application Usage ID
CN	Core Network
CPIM	Common Profile for Instant Messaging
CSCF	Call Session Control Function
DM	Data Manipulator
DMS	Data Manipulation Server
EPA	Event Publication Agent
ESC	Event State Compositor
HSS	Home Subscriber Server
HTTP	HyperText Transfer Protocol
I-CSCF	Interrogating - CSCF
IM	IP Multimedia
IOI	Inter Operator Identifier
IP	Internet Protocol
MIME	Multipurpose Internet Mail Extensions
P-CSCF	Proxy - CSCF
PIDF	Presence Information Data Format
PNA	Presence Network Agent
PS	Presence Server
PSI	Public Service Identity
PUA	Presence User Agent
RLMI	Resource List Meta-Information
RLS	Resource List Server
RPID	Rich Presence Information Data

S-CSCF	Serving - CSCF
SIP	Session Initiation Protocol
TLS	Transport Layer Security
UE	User Equipment
URI	Universal Resource Identifier
XCAP	XML Configuration Access Protocol
XML	Extensible Markup Language

4 Presence service overview

The presence service provides the ability for the home network to manage presence information of a user's device, service or service media even whilst roaming. A user's presence information may be obtained through input from the user, information supplied by network entities or information supplied by elements external to the home network. Consumers of presence information, watchers, may be internal or external to the home network. The architecture for the 3GPP presence service is specified in 3GPP TS 23.141 [4].

SIP and XCAP provide means to manipulate the presence status of a user. For details on the differences between those means refer to ~~draft-ietf-sip-publish-03~~RFC 3903 [23] and draft-isomaki-simple-xcap-pidf-manipulation-usage-00 [34]. For details on the relationship of DMS to other roles see subclause 6.2.2.

Editor's note: It may be appropriate to include text in this clause pointing to the stage 1 on group management.

5 SIP related procedures

5.1 Introduction

5.2 Functional entities

5.2.1 User Equipment (UE)

A UE shall implement the role of a PUA (see subclause 5.3.1), a watcher (see subclause 5.3.2) or both.

5.2.2 Application Server (AS)

An AS may implement either of the roles of a PUA (see subclause 5.3.1), watcher (see subclause 5.3.2), PS (see subclause 5.3.3), RLS (see subclause 5.3.4), or PNA (see subclause 5.3.5).

For this version of the present document, the interface between the PNA and the PS is not defined.

5.3 Roles

5.3.1 Presence User Agent (PUA)

5.3.1.1 General

A PUA is an entity that provides presence information to a PS.

In addition to the procedures specified in subclause 5.3.1, the PUA shall support the procedures specified in 3GPP TS 24.229 [9] appropriate to the functional entity in which the PUA is implemented.

5.3.1.2 Publication of presence information

When the PUA intends to publish its own view of the presentity's presence information it shall generate a PUBLISH request by acting as an Event Publication Agent (EPA) in accordance with ~~draft-ietf-sip-publish-03~~RFC 3903 [23].

The PUA shall implement the "application/pidf+xml" content type as described in ~~draft-ietf-impp-cpim-pidf-08~~RFC 3863 [21] together with the Presence Information Data Format (PIDF) extensions defined in draft-ietf-simple-rpid-04~~3~~ [26].

The PUA may implement the PIDF extensions defined in draft-ietf-simple-cipid-03~~1~~ [32].

The PUA may implement location information according to the format defined in draft-ietf-geopriv-pidf-lo-03~~1~~ [37].

NOTE 1: The categorization of presence attributes to generic information attributes and communication address specific attributes is done using the <contact-type> element. The value of the <contact-type> element categorizes the content of the tuple in such a way that the value "presentity" denotes general information about the presentity and the value "service" denotes the communication mean specific information. The presence document consists of one or more tuples describing the presentity related information and one or more tuples describing information about different communication means.

The PUA shall implement draft-ietf-simple-prescaps-ext-02~~0~~ [25] if it wants to make use of SIP user agent capabilities in the presence document. The extension may be used for describing the type of the service described by the presence tuple.

The PUA may indicate its support for partial publishing by including the "application/pidf-partial+xml" content type as described in draft-ietf-simple-partial-pidf-format-02~~0~~ [38]. The first partial PUBLISH request shall contain the full publication, where the state attribute is set to value "full" and the value of the 'version' attribute is initialized.

On successful response from the PS, the PUA in subsequent PUBLISH requests shall generate partial publications, where the state attribute is set to value 'partial' and the value of the 'version' attribute is incremented by one. The partial publication should contain only the new and changed tuples, information about the removed tuples and presence information outside the tuple elements.

If the PUA receives a 415 (Unsupported Media Type) response to the PUBLISH request with "application/pidf+xml" in the Accept header field, the PUA shall send a PUBLISH request including the "application/pidf+xml".content type.

Editor's Note: The above procedures on partial publishing will be replaced by references to the IETF draft-lonnfors-simple-publish-partial-00 once the draft has been discussed in IETF. If IETF defines another solution for partial publishing or indicate the reuse of existing procedures as a solution, then the above procedures on partial publishing will be revised.

The PUA shall update the presence information, either 600 s before the publication expiration time if the publication period indicated from the PS in the response to the PUBLISH request was for greater than 1 200 s, or when half of the time has expired if the publication period was for 1 200 s or less, unless the UE has determined that an update to the presence information is not required.

When the PUA intends to show different value of the same presence attribute to different watchers, the PUA shall publish a tuple for every value it intends to show, all including a different value of the same presence attribute. The PUA shall label different information with different value of the <class> element in every published tuple as defined in draft-ietf-simple-rpid-04~~3~~ [26]. The PUA shall also authorize different tuples to different watchers or watcher groups by manipulating the subscription authorization policy as defined in subclause 6.3.1.2.

If a local configuration information limiting the rate at which PUA is allowed to generate PUBLISH requests is available, then PUA shall take that information into account. Such local configuration information could be e.g. the shortest time period between consecutive PUBLISH requests.

5.3.1.3 Mapping of presence attributes

The eXtensible Markup Language (XML) Schema Definition of the "application/pidf+xml" format or the "application/pidf-partial+xml" format cover the definition of the 3GPP subscriber's presence attributes and the PUA shall perform the following mapping:

- the communication address (containing communication means, status and contact address) attribute and the priority attribute are represented by a <tuple> element including a basic <status> element and one or more <contact> elements containing a priority attribute as defined in ~~draft-ietf-impp-cpim-pidf-08~~RFC 3863 [21].

  The PUA represents the subscriber's status by including a <contact-type> element defined in draft-ietf-simple-rpid-04~~3~~ [26] with the value "presentity" and a basic <status> element defined in ~~draft-ietf-impp-cpim-pidf-08~~RFC 3863 [21]. In order to express more granularity in values, <activity> and <privacy> elements both defined in draft-ietf-simple-rpid-04~~3~~ [26] can be used inside the <status> elements. Further PIDF extensions as defined in draft-ietf-simple-cipid-03~~1~~ [32] can also be used.

  In case of including multiple presentity related tuples in the presence document, all the presentity related tuples except one contains information about an alternate contact related to the presentity; the type of the alternate contact shall be indicated using the <relationship> element defined in draft-ietf-simple-rpid-04~~3~~ [26];

NOTE 1: draft-ietf-simple-rpid-04~~3~~ [26] defines also other values of the <contact-type> element. Those values can be used to create additional categories.

- the text attribute is represented by the <note> element as defined in ~~draft-ietf-impp-cpim-pidf-08~~RFC 3863 [21]; and

- the location attribute is represented by the elements defined in draft-ietf-geopriv-pidf-lo-03~~1~~ [37] and the <placetype> element defined in draft-ietf-simple-rpid-04~~3~~ [26].

NOTE 2: Only information elements relevant for the application is included in the PUBLISH request. Attributes not relevant or available (e.g. the text attribute or the location attribute) are omitted.

Additional extensions can be used to express application specific attributes, but their usage is outside the scope of this version of the specification.

5.3.1.4 Storing presence attributes by multipart/related or content indirection

The PUA shall implement the "multipart/related" content type as described in RFC 2387 [14] if it wants to aggregate other Multipurpose Internet Mail Extensions (MIME) objects with the "application/pidf+xml" content type.

When a presence attribute has a value of a MIME object, the PUA shall either:

a) publish the presence document and the MIME object utilizing the "multipart/related" content-type in the PUBLISH request; or

b) make use of content indirection.

When the PUA decides to use the content indirection mechanism for publishing an initial or modified value of a presence attribute the PUA shall follow the following procedure:

a) either store the MIME object behind an HTTP URI on the PS or ensure that the MIME object and a HTTP URL pointing to that MIME object already exists on the PS;

b) use the "multipart/related" content type as described in RFC 2387 [14] with the content indirection mechanism as specified in draft-ietf-sip-content-indirect-mech-05~~3~~ [40] for the publication of presence information format as follows:

   - set a CID URI referencing to other MIME multipart body which contains the content indirection information as the value of the XML element whose value is delivered as an indirect content;

   - include the presence document of the format "application/pidf+xml" or "application/pidf-partial+xml" in the root of the body of the "multipart/related" content;

   - specify the part having information about the MIME object by using the "message/external-body" content type, defining the HTTP URI, versioning information and other information about the MIME object as described in draft-ietf-sip-content-indirect-mech-05~~3~~ [40].

NOTE 1: The versioning information is used for determining whether or not the MIME object indirectly referenced by a URI has changed or not;

When storing a MIME object on the PS the PUA shall:

a) construct as many HTTP URIs as many objects to be stored; and

b) formulate every HTTP URI according to a predefined directory structure.

NOTE 2: The PUA has the root directory for storing the MIME objects on the PS preconfigured.

NOTE 3: The PUA needs to store the MIME objects on the PS behind the HTTP URI(s) created previously using standard HTTP procedures as defined in RFC 2616 [15].

5.3.1.5 Subscription for the watcher information event template package

Upon activation of the presence service, the PUA application shall subscribe for the watcher information state changes in accordance with ~~draft-ietf-simple-winfo-package-05~~RFC 3857 [28] and ~~draft-ietf-simple-winfo-format-04~~RFC 3858 [29].

The PUA application may include filters in the body of the SUBSCRIBE request in accordance with draft-ietf-simple-filter-format-03~~0~~ [30] and draft-ietf-simple-event-filter-funct-03~~0~~ [31].

5.3.1.6 Subscription for xcap-change

In order to get notifications of changes to XML documents manipulated via the Ut reference point the PUA may generate a SUBSCRIBE request in accordance with draft-ietf-simple-xcap-package-02 [39] and draft-ietf-sipping-config-framework-05~~4~~ [43].

5.3.2 Watcher

5.3.2.1 General

A watcher is an entity that is subscribed or requests presence information about a presentity from the PS.

In addition to the procedures specified in subclause 5.3.2, the watcher shall support the procedures specified in 3GPP TS 24.229 [9] appropriate to the functional entity in which the watcher is implemented.

5.3.2.2 Subscription for presence information state changes and notification acceptance

When the watcher application intends to subscribe for presence information state changes of a presentity, it shall generate a SUBSCRIBE request in accordance with RFC 3265 [19] and ~~draft-ietf-simple-presence-10~~RFC 3856 [27].

The watcher application shall implement the "application/pidf+xml" content type as described in ~~draft-ietf-impp-cpim-pidf-08~~RFC 3863 [21] together with the PIDF extensions defined in draft-ietf-simple-rpid-04~~3~~ [26].

The watcher application may implement the PIDF extensions defined in draft-ietf-simple-cipid-03~~1~~ [32].

The watcher application shall implement draft-ietf-simple- prescaps-ext-02~~0~~ [25]if it wants to make use of SIP user agent capabilities extensions included in the presence document. The extension may be used by the watcher application for interpreting the type of the service described by the presence tuple. The watcher application may include filters in the body of the SUBSCRIBE request in accordance with draft-ietf-simple-filter-format-03~~0~~ [30] and draft-ietf-simple-event-filter-funct-03~~0~~ [31].

The watcher application may indicate its support for partial notification using the Accept header field in accordance with draft-ietf-simple-partial-notify-03~~1~~ [24].

The watcher application shall interpret the received presence information according to the following:

a) a tuple including a <contact-type> element as defined in draft-ietf-simple-rpid-04~~3~~ [26] with the value "presentity" means general information about the presentity;

b) a tuple including a <relationship> element and <contact-type> element with the value "presentity" as defined in draft-ietf-simple-rpid-04~~3~~ [26] means information about an alternate contact to the presentity;

c) a tuple including a <contact-type> element as defined in draft-ietf-simple-rpid-04~~3~~ [26] with the value "service" means communication mean specific information. The communication mean described by the tuple is deduced from the URI scheme of the contact address information present in the <contact> element as defined in ~~draft-ietf-impp-cpim-pidf-08~~RFC 3863 [21]. If the URI scheme of the contact address information provides ambiguous information about the communication means, the watcher application shall further examine other elements of the tuple to decide the communication mean. Such elements can be the <methods> element, any of the different media type specific elements as defined in draft-ietf-simple-prescaps-ext-02~~0~~ [25], or the <relationship> element as defined in draft-ietf-simple-rpid-04~~3~~ [26].

Additional extensions can be used to express application specific attributes, but their usage is outside the scope of this version of the specification.

5.3.2.3 Subscription for presence information state changes of presentity collections

When the watcher application intends to subscribe for presence information state changes of a presentity collection, it shall generate a SUBSCRIBE request in accordance with draft-ietf-simple-event-list-05~~4~~ [22], additionally to the procedures described in subclause 5.3.2.2.

5.3.2.4 Subscription for the watcher information event template package

Upon activation of the presence service, the watcher application may subscribe recursively for the watcher information state changes in accordance with ~~draft-ietf-simple-winfo-package-05~~RFC 3857 [28] and ~~draft-ietf-simple-winfo-format-04~~RFC 3858 [29].

The watcher application may include filters in the body of the SUBSCRIBE request in accordance with draft-ietf-simple-filter-format-03~~0~~ [30] and draft-ietf-simple-event-filter-funct-03~~0~~ [31].

5.3.2.5 Subscription for xcap-change

In order to get notifications of changes to XML documents manipulated via the Ut reference point the watcher may generate a SUBSCRIBE request in accordance with draft-ietf-simple-xcap-package-02 [39] and draft-ietf-sipping-config-framework-05~~4~~ [43].

5.3.3 Presence Server (PS)

5.3.3.1 General

A PS is an entity that accepts, stores, and distributes presence information.

In addition to the procedures specified in subclause 5.3.3, the PS shall support the procedures specified in 3GPP TS 24.229 [9] appropriate to the functional entity in which the PS is implemented.

5.3.3.2 Subscription acceptance to presence information and notification of state changes

When the PS receives a SUBSCRIBE request for the presence information event package, the PS shall first attempt to verify the identity of the source of the SUBSCRIBE request as described in 3GPP TS 24.229 [9] subclause 5.7.1.4, then perform authorization according to 3GPP TS 24.229 [9] subclause 5.7.1.5. In case of successful subscription, the PS shall generate a response to the SUBSCRIBE request and notifications in accordance with RFC 3265 [19] and ~~draft-ietf-simple-presence-10~~RFC 3856 [27].

Additionally, in the special case of a watcher subscription if the subscription authorization policy results in the action to confirm the watcher subscription from the PUA and the PUA has a valid watcher information subscription, see ~~draft-ietf-simple-winfo-package-05~~RFC 3857 [28], then, the PS shall inform the PUA about the watcher subscription attempt.

If the watcher application has indicated the need for partial notification using the Accept header field, then the PS shall generate partial notifications in accordance with draft-ietf-simple-partial-notify-03~~1~~ [24] and draft-ietf-simple-partial-pidf-format-02~~0~~ [38].

If the body of the SUBSCRIBE request from the watcher contains filters, the PS shall apply the requested filtering function on notifications in accordance with draft-ietf-simple-filter-format-03~~0~~ [30] and draft-ietf-simple-event-filter-funct-03~~0~~ [31].

If the watcher application has indicated support for the "multipart/related" content type using the Accept header field, then the PS may generate notifications using "multipart/related" content type which aggregates "application/pidf+xml" formatted presence information with other MIME objects in accordance with RFC 2387 [14]. In this case, the PS shall modify the value of the presence attribute in the PIDF document to refer to the MIME object included in the corresponding MIME multipart body. If the watcher application has not indicated support for the "multipart/related" or a MIME object cannot be accessed by the PS, the PS should exclude the presence attribute from the notification.

5.3.3.3 Publication acceptance of presence information

The PS shall act as an Event State Compositor (ESC).

When the PS receives a PUBLISH request, the PS shall first verify the identity of the source of the PUBLISH request as described in 3GPP TS 24.229 [9] subclause 5.7.1.4, then perform authorization according to 3GPP TS 24.229 [9] subclause 5.7.1.5. In case of successful authentication and authorization, the PS shall process the PUBLISH request in accordance with ~~draft-ietf-sip-publish-03~~RFC 3903 [23].

If the PUBLISH request indicated support for partial publishing using the "application/pidf-partial+xml" content-type described in draft-ietf-simple-partial-pidf-format-02~~0~~ [38] and if the PS supports partial publishing, the PS shall process the PUBLISH request in accordance with ~~draft-ietf-sip-publish-03~~RFC 3903 [23] and as follows:

- if the state attribute's value is "full" the PS shall store the received presence and version information;
- if the state attribute's value is "partial" the PS shall ensure that the version information is correct compared to the stored one; combine the received presence information with the stored one by replacing presence information outside the tuples, replacing changed tuples, adding new tuples and removing tuples which id attributes' values have been listed in the <removed> element defined in draft-ietf-simple-partial-pidf-format-02~~0~~ [38]; and store the received version information.

If the PS does not support partial publishing, then the PS shall send a 415 (Unsupported Media Type) response with "application/pidf+xml" in the Accept header field.

Editor's Note: The above procedures on partial publishing will be replaced by references to the IETF draft-lonnfors-simple-publish-partial-00 once the draft has been discussed in IETF. If IETF defines another solution for partial publishing or indicate the reuse of existing procedures as a solution, then the above procedures on partial publishing will be revised.

If the PUBLISH request contained the "multipart/related" content type and the PS supports the content type, the PS shall process the content as follows:

- if a MIME multipart contains a MIME object of a content type supported by the PS, either store the MIME object in case of initial publication or replace an existing content in case of modify operation; and
- if a multipart includes the "message/external-body" content type and the content indirection is supported by the PS, ensure that it has access to the MIME object indicated by the URI and that the MIME object exists; and associate the value of the presence attribute that refers to the MIME object with the MIME object and additional information about it.

If the PS does not support the content type used for publishing MIME objects then the PS shall send a 415 (Unsupported Media Type) response and indicate the supported content types in the Accept header.

NOTE: If the PS receives a HTTP request for storing a MIME object on the PS meaning that the HTTP URI points to a predefined directory reserved for storing MIME objects and the request is an HTTP PUT request, the PS replaces any existing content referenced by the Request-URI with the content of the request. If the Request-URI points to an uncreated directory, the PS creates the directory, stores the content there and associates the content with the Request-URI. For all requests, i.e. HTTP PUT, HTTP GET and HTTP DELETE requests, the PS generates an appropriate response in accordance with RFC 2616 [15].

5.3.3.4 Subscription acceptance to watcher information and notification of state changes

When the PS receives a SUBCRIBE request for the watcher information event template package, the PS shall first verify the identity of the source of the SUBSCRIBE request as described in 3GPP TS 24.229 [9] subclause 5.7.1.4, then perform authorization according to 3GPP TS 24.229 [9] subclause 5.7.1.5. In case of successful subscription, the PS shall generate a response to the SUBSCRIBE request and notifications in accordance with RFC 3265 [19], ~~draft-ietf-simple-winfo-package-05~~RFC 3857 [28] and ~~draft-ietf-simple-winfo-format-04~~RFC 3858 [29].

If the body of the SUBSCRIBE request from the PUA contains filters, the PS shall apply the requested filtering function on notifications in accordance with draft-ietf-simple-filter-format-03~~0~~ [30] and draft-ietf-simple-event-filter-funct-03~~0~~ [31].

5.3.3.5 Subscription acceptance to xcap-change and notification of state changes

When the PS receives a SUBSCRIBE request having the Event header value 'sip-profile', the PS shall first verify the identity of the source of the SUBSCRIBE request as described in 3GPP TS 24.229 [9] subclause 5.7.1.4, then it shall perform authorization as described in 3GPP TS 24.229 [9] subclause 5.7.1.5. Afterwards, the PS shall generate a response to the SUBSCRIBE request and notifications in accordance with draft-ietf-simple-xcap-package-02 [39] and draft-ietf-sipping-config-framework-05~~4~~ [43].

5.3.4 Resource List Server (RLS)

5.3.4.1 General

The Resource List Server (RLS) is an implementation of the presence list server. The RLS is an entity that accepts subscriptions to resource lists and sends notifications to update subscribers of the state of the resources in a resource list.

In addition to the procedures specified in subclause 5.3.4, the RLS shall support the procedures specified in 3GPP TS 24.229 [9] appropriate for an AS in which the RLS is implemented.

5.3.4.2 Subscription acceptance to resource lists and notification of state changes

When the RLS receives a SUBSCRIBE request for the presence information event package of a presentity collection, the RLS shall first verify the identity of the source of the SUBSCRIBE request as described in 3GPP TS 24.229 [9] subclause 5.7.1.4, then perform authorization according to 3GPP TS 24.229 [9] subclause 5.7.1.5. In case of successful subscription, the RLS shall generate a response to the SUBSCRIBE request and notifications in accordance with draft-ietf-simple-event-list-05~~4~~ [22] by adding a Require header field with value "eventlist" to the request.

If the body of the SUBSCRIBE request from the watcher contains filters, the RLS shall apply the requested filtering function on notifications in accordance with draft-ietf-simple-filter-format-03~~0~~ [30] and draft-ietf-simple-event-filter-funct-03~~0~~ [31].

5.3.4.3 Subscription to presence information

When the RLS receives a SUBSCRIBE request for the presence information event package of a presentity collection and installs the corresponding subscription, the RLS shall resolve the list URI to individual URIs and generate SUBSCRIBE requests for each of the individual URIs as per the procedures in RFC 3265 [19], ~~draft-ietf-simple-presence-10~~RFC 3856 [27] and draft-ietf-simple-event-list-05~~4~~ [22] if the state information for the resource represented by the individual URI is otherwise not available.

Editor's note: There is a need for a mechanism that can protect an IMS network from list loops potentially caused by lists of lists. Unless referenced IETF specifications provide support for implementation of this kind of protection, a mechanism or restrictions on the usage of list of lists must be identified and described here.

5.3.4.4 Subscription acceptance to xcap-change and notification of state changes

When the RLS receives a SUBSCRIBE request having the Event header value 'sip-profile', the RLS shall first verify the identity of the source of the SUBSCRIBE request as described in 3GPP TS 24.229 [9] subclause 5.7.1.4, then it shall

perform authorization as described in 3GPP TS 24.229 [9] subclause 5.7.1.5. Afterwards, the RLS shall generate a response to the SUBSCRIBE request and notifications in accordance with draft-ietf-simple-xcap-package-02 [39] and draft-ietf-sipping-config-framework-054 [43].

*** next change ***

A.3 Signalling flows demonstrating how watchers subscribe to presence event notification

A.3.1 Introduction

The subclause covers the signalling flows that show how watchers can request presence information about a presentity.

For the routeing of the Public Service Identity (PSI) towards the AS, there are two scenarios:

Subclause A.3.3.2 shows the case where the I-CSCF forwards the SUBSCRIBE request directly to the RLS when the RLS is located within the same network. There is another scenario where the I-CSCF forwards the SUBSCRIBE request towards the RLS, being involved with the S-CSCF located in the same network, but this scenario is not described in the present document.

A.3.2 Watcher and presentity in different networks, UE in home network

A.3.2.1 Successful subscription

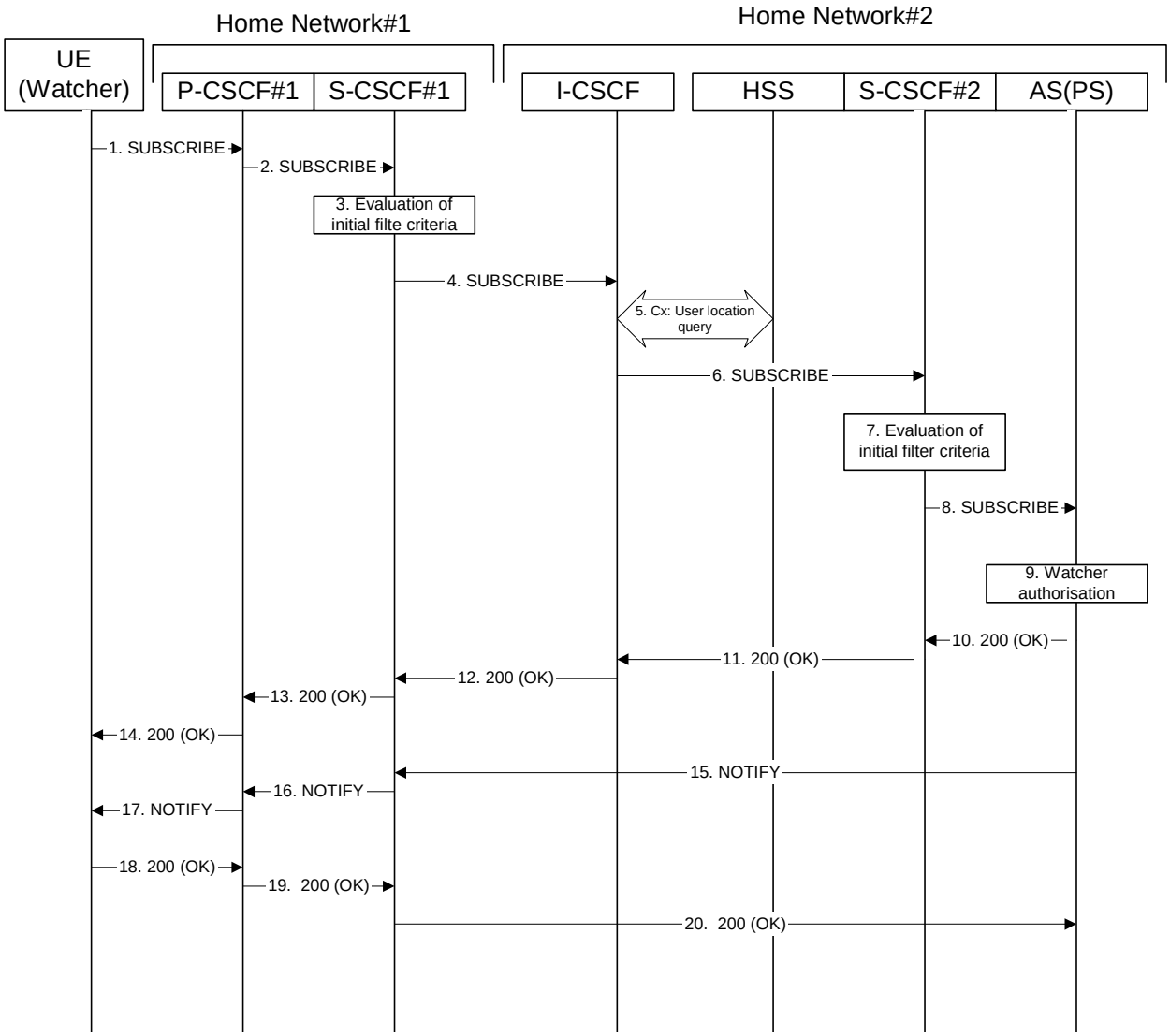


Figure A.3.2.1-1: Watcher subscribing for presence information

Figure A.3.2.1-1 shows a watcher subscribing to presence event notification about a presentity. The presentity is in a different IM CN subsystem. The details of the signalling flows are as follows:

1. **SUBSCRIBE request (UE (watcher) to P-CSCF) - see example in table A.3.2.1-1**

   A watcher agent in a UE wishes to watch a presentity, or certain presence information~~tuples~~ of the presentity. To initiate a subscription, the UE generates a SUBSCRIBE request containing the "presence" event that it wishes to be notified of, together with an indication of the length of time this periodic subscription should last and the support for partial notification.

Table A.3.2.1-1: SUBSCRIBE request (UE (watcher) to P-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
Max-Forwards: 70
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
Route: <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>, <sip:orig@scscf1.home1.net;lr>
P-Preferred-Identity: <sip:user1_public1@home1.net>
Privacy: none
From: <sip:user1_public1@home1.net>;tag=31415
To: <sip:user2_public1@home2.net>
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 61 SUBSCRIBE
Require: sec-agree
Proxy-Require: sec-agree
Security-Verify: ipsec-3gpp; q=0.1; alg=hmac-sha-1-96; spi-c=98765432; spi-s=87654321; port-
      c=8642; port-s=7531
Event: presence
Expires: 7200
Accept: application/pidf+xml;q=0.3, application/pidf-partial+xml;q=1
Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>
Content-Length: 0
```

Request-URI: Public user identity whose events the subscriber subscribes to.

Event: This field is populated with the value "presence" to specify the use of the presence package.

Accept: This field is populated with the value 'application/pidf+xml' and 'application/pidf-partial+xml', latter one with higher preference.

To: Same as the Request-URI.

2. **SUBSCRIBE request (P-CSCF to S-CSCF) - see example in table A.3.2.1-2**

The P-CSCF looks up the serving network information for the public user identity that was stored during the registration procedure. The SUBSCRIBE request is forwarded to S-CSCF. A Route header is inserted into SUBSCRIBE request. The information for the Route header is taken from the service route determined during registration.

Table A.3.2.1-2: SUBSCRIBE request (P-CSCF to S-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Access-Network-Info:
Max-Forwards: 69
P-Asserted-Identity: <sip:user1_public1@home1.net>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"
Privacy:
Route: <sip:orig@scscf1.home1.net;lr>
Record-Route: <sip:pcscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

3. **Evaluation of initial filter criteria**

S-CSCF#1 validates the service profile of this subscriber and evaluates the initial filter criteria. For this example, assume no Application Server involvement.

4. **SUBSCRIBE request (S-CSCF to I-CSCF) - see example in table A.3.2.1-4**

S-CSCF#1 performs an analysis of the destination address, and determines the network operator to whom the destination subscriber belongs. Since the originating operator does not desire to keep their internal configuration hidden, S-CSCF#1 forwards the SUBSCRIBE request directly to the I-CSCF in the destination network.

Table A.3.2.1-4: SUBSCRIBE (S-CSCF to I-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;;branch=z9hG4bKnashds7
Max-Forwards: 68
P-Asserted-Identity: <sip:user1_public1@home1.net>, <tel:+1-212-555-1111>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"; orig-ioi=home1.net
Privacy:
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

5. **Cx: User Location Query procedure**

The I-CSCF sends a query to the HSS to find out the S-CSCF of the called user. The HSS responds with the address of the current S-CSCF for the terminating subscriber.

For detailed message flows see 3GPP TS 29.228 [10].

Table A.3.2.1-5a provides the parameters in the SIP SUBSCRIBE request (flow 4), which are sent to the HSS.

Table A.3.2.1-5a: Cx: User registration status query procedure (I-CSCF to HSS)

Message source and destination	Cx: Information element name	Information source in SIP SUBSCRIBE	Description
I-CSCF to HSS	User Public Identity	Request-URI	This information element indicates the public user identity

Table A.3.2.1-5b provides the parameters sent from the HSS that need to be mapped to the SIP SUBSCRIBE request (flow 6) and sent to the S-CSCF.

Table A.3.2.1-5b: Cx: User registration status query procedure (HSS to I-CSCF)

Message source and destination	Cx: Information element name	Mapping to SIP header in SIP SUBSCRIBE	Description
HSS to I-CSCF	S-CSCF name	Route header field	This information indicates the serving CSCF's name of that user

6. **SUBSCRIBE request (I-CSCF to S-CSCF) - see example in table A.3.2.1-6**

The I-CSCF forwards the SUBSCRIBE request to the S-CSCF (S-CSCF#2) that will handle the termination.

Table A.3.2.1-6: SUBSCRIBE request (I-CSCF to S-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
Max-Forwards: 67
P-Asserted-Identity:
P-Charging-Vector:
Privacy:
Route: <sip:scscf2.home2.net;lr>
Record-Route:
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

NOTE: The I-CSCF does not add itself to the Record-Route header, as it has no need to remain in the signalling path for the subsequent requests.

7. **Evaluation of initial filter criteria**

S-CSCF#2 validates the service profile of this subscriber and evaluates the initial filter criteria. For sip:user2_public1@home2.net S-CSCF#2 has termination initial filter criteria with service points of interest of Method = SUBSCRIBE and Event = "presence" that informs the S-CSCF to route the SUBSCRIBE request to the Application Server ps.home2.net. The S-CSCF#2 has preconfigured information not to record route this request.

8. **SUBSCRIBE request (S-CSCF to PS) – see example in table A.3.2.1-8**

The S-CSCF forwards the SUBSCRIBE request to the PS.

Table A.3.2.1-8: SUBSCRIBE request (S-CSCF to PS)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP scscf2.home2.net;branch=z9hG4bK764z87.1, SIP/2.0/UDP
      icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
Max-Forwards: 66
P-Asserted-Identity:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Privacy:
Route: <sip:ps.home2.net;lr>, <sip:scscf2.home2.net;lr>
Record-Route:
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the originating

Inter Operator Identifier (IOI) parameter of this header and removes the terminating IOI parameter.

P-Charging-Function-Addresses: The S-CSCF populates the P-Charging-Function-Addresses header field to be passed to the PS.

9. **Authorization of watcher**

The PS performs the necessary authorization checks on the originator to ensure it is allowed to watch the presentity. In this example all privacy conditions are met, so the PS sends a 200 (OK) response to the S-CSCF.

In the case where the privacy/authorization checks failed, then a necessary 2xx or 4xx response would be sent to the S-CSCF. The selection of the correct response code depends on the presentity's subscription authorization policy document.

10. **200 (OK) response (PS to S-CSCF) - see example in table A.3.2.1-10**

The PS sends the response to S-CSCF#2.

Table A.3.2.1-10: 200 (OK) response (PS to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf2.home2.net;branch=z9hG4bK764z87.1, SIP/2.0/UDP
icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"; orig-ioi=home1.net;
      term-ioi=home2.net
P-Charging-Functio-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Record-Route:
From:
To: <sip:user2_public1@home2.net>;tag=151170
Call-ID:
CSeq:
Expires:
Contact: <sip:ps.home2.net>
Content-Length:
```

P-Charging-Vector: The PS stores the originating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The PS stores the P-Charging-Function-Addresses header field and passes this header to the S-CSCF.

11. **200 (OK) response (S-CSCF to I-CSCF) - see example in table A.3.2.1-11**

S-CSCF#2 forwards the response to I-CSCF#2.

Table A.3.2.1-11: 200 (OK) response (S-CSCF to I-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Charging-Vector:
P-Charging-Function-Addresses:
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the I-CSCF.

12. **200 (OK) response (I-CSCF to S-CSCF) - see example in table A.3.2.1-12**

I-CSCF#2 forwards the response to S-CSCF#1.

Table A.3.2.1-12: 200 (OK) response (I-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Charging-Vector:
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

13. **200 (OK) response (S-CSCF to P-CSCF) - see example in table A.3.2.1-13**

S-CSCF#1 forwards the response to P-CSCF#1.

Table A.3.2.1-13: 200 (OK) response (S-CSCF to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

14. **200 (OK) response (P-CSCF to UE) - see example in table A.3.2.1-14**

P-CSCF#1 forwards the response to the watcher agent in the UE.

Table A.3.2.1-14: 200 (OK) response (P-CSCF to UE)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
Record-Route: <sip:orig@scscf1.home1.net;lr>, <sip:pcscf1.home1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

15. **NOTIFY request (PS to S-CSCF) - see example in table A.3.2.1-15**

As soon as the PS sends a 200 (OK) response to accept the subscription, it sends a NOTIFY request with the current state of the presentity's presence information~~tuples~~ that the watcher has subscribed and been authorized to. The NOTIFY request is sent to S-CSCF#1. Based on the Accept header field of the

SUBSCRIBE request, the PS decides to use partial notifications to provide changes of presence information. Because the first notification always contains the full state the 'application/pidf ~~partial~~+xml' content type is used in the first NOTIFY request.

Table A.3.2.1-15: NOTIFY request (PS to S-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 70
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=123551024"; orig-ioi=home2.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.home1.net;lr>
From: <sip:user2_public1@home2.net>;tag=151170
To: <sip:user1_public1@home1.net>;tag=31415
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 42 NOTIFY
Subscription-State: active ;expires=7200
Event: presence
Contact: <sip:ps.home2.net>
Content-Type: application/pidf-partial+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
   <pidf-part:presence xmlns="urn:ietf:params:xml:ns:pidf"
              xmlns:pidf-part="urn:ietf:params:xml:ns:pidf-partial"
              xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
              xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
              xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid-person"
              xmlns:dmp="urn:ietf:params:xml:ns:pidf:person"
              xmlns:pcp="urn:ietf:params:xml:ns:pidf:servcapssimple-prescaps-ext"
              xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
        entity="pres:user2_public1@home2.net" version="0" state="full">

     <tuple id="a8098a.672364762364">
       <status>
         <basic>open</basic>
         <es:activity>meeting</es:activity>
         <es:placetype until="2003-08-27T17:30:00Z">office</es:placetype>
         <es:privacy><text/>private</es:privacy>
         <es:idle since="2003-08-27T10:43:00Z"/>
         <pcp:prescaps>
           <pcp:video negated="false"></pcp:video>
           <pcp:mobility>mobile</pcp:mobility>
           <pcp:audio negated="true"></pcp:audio>
         </pcp:prescaps>
         <es:status-icon>http://example.com/~user2/icon.gif</es:status-icon>
       </status>
       <et:class>sip</et:class>
       <et:contact-type>service</et:contact-type>
       <pcp:video>false</pcp:video>
       <pcp:audio>true</pcp:audio>
       <contact priority="0.8">sip:user2_public1@home2.net</contact>
       <note xml:lang="en">Don't Disturb Please!</note>
       <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
       <timestamp>2003-08-27T11:49:29Z</timestamp>
     </tuple>

     <tuple id="sfddsj74.78">
       <status>
         <basic>open</basic>
       </status>
       <et:class>presentity</et:class>
       <et:contact-type>presentity</et:contact-type>
       <ci:homepage>http://example.com/~user2</ci:homepage>
       <ci:icon>http://example.com/~user2/icon.gif</ci:icon>
       <ci:card>http://example.com/~user2/card.vcd</ci:card>
       <note xml:lang="en">I'm in a boring meeting</note>
       <note xml:lang="en">I'll be in Tokyo next week</note>
       <timestamp>2003-08-27T11:49:29Z</timestamp>
     </tuple>

     <tuple id="jklhgf9788934774.78">
       <status>
         <basic>open</basic>
       </status>
       <et:class>assistant</et:class>
       <et:contact-type>presentity</et:contact-type>
       <et:relationship>assistant</et:relationship>
       <contact priority="1.0">tel:+1-212-555-2222</contact>
       <note xml:lang="en">She's my secretary</note>
       <timestamp>2003-08-27T11:49:29Z</timestamp>
```

```
      </tuple>

      <dmp:person>
        <ep:class>presentity</ep:class>
        <ci:homepage>http://example.com/~user2</ci:homepage>
        <ci:card>http://example.com/~user2/card.vcd</ci:card>
        <dmp:status>
          <ep:activities><ep:meeting/></ep:activities>
          <ep:place-type until="2003-08-27T17:30:00Z">office</ep:place-type>
        </dmp:status>
      </dmp:person>

   </pidf-part:presence>
```

P-Charging-Vector: The PS populates the icid parameter with a globally unique identifier and adds the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The PS populates the P-Charging-Function-Addresses header field to be passed to the S-CSCF.

Content-Type: Set to the preferred value of the Accept header received in the SUBSCRIBE request.

The message body in the NOTIFY request that carries the presence information of the presentity is formed as indicated in draft-ietf-impp-cpim-pidf-08RFC 3863 [21], draft-ietf-simple-rpid-043 [26], draft-ietf-simple-prescaps-ext-020 [25], draft-ietf-simple-cipid-031 [32], draft-ietf-simple-partial-notify-031 [24] and draft-ietf-simple-presence-data-model-01 [x]draft-ietf-simple-partial-pidf-format-00 [38].

16. **NOTIFY request (S-CSCF to P-CSCF) - see example in table A.3.2.1-16**

The S-CSCF#1 forwards the NOTIFY request to P-CSCF#1.

Table A.3.2.1-16: NOTIFY request (S-CSCF to P-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 69
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=123551024"
P-Charging-Function-Addresses:
Privacy:
Record-Route: <sip:scscf1.home1.net;lr>
Route: sip:<pcscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter of this header and removes the parameter from this header.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the P-CSCF.

17. **NOTIFY request (P-CSCF to UE) - see example in table A.3.2.1-17**

The P-CSCF forwards the NOTIFY request to the watcher application in the UE.

Table A.3.2.1-17: NOTIFY request (P-CSCF to UE)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 68
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.home1.net:7531;lr;comp=sigcomp>
Privacy:
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

18. **200 (OK) response (UE to P-CSCF) - see example in table A.3.2.1-18**

The UE generates a 200 (OK) response to the NOTIFY request.

Table A.3.2.1-18: 200 (OK) response (UE to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      scscf2.home2.net;branch=z9hG4bK764z87.1, SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

19. **200 (OK) response (P-CSCF to S-CSCF) – see example in table A.3.2.1-19**

The P-CSCF forwards the 200 (OK) response to S-CSCF#1.

Table A.3.2.1-19: 200 (OK) response (P-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      ps.home2.net;branch=z9hG4bK348923.1
P-Access-Network-Info:
From:
To:
Call-ID:
CSeq:
Content-Length:
```

20. **200 (OK) response (S-CSCF to P-S) - see example in table A.3.2.1-20**

S-CSCF#2 forwards the 200 (OK) response to the PS.

Table A.3.2.1-20: 200 (OK) response (S-CSCF to PS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
From:
To:
Call-ID:
CSeq:
Content-Length:
```

A.3.3 Watcher subscribing to resource list, UE in visited network

A.3.3.1 Watcher subscribing to his own resource list, UE in visited network - Successful subscription

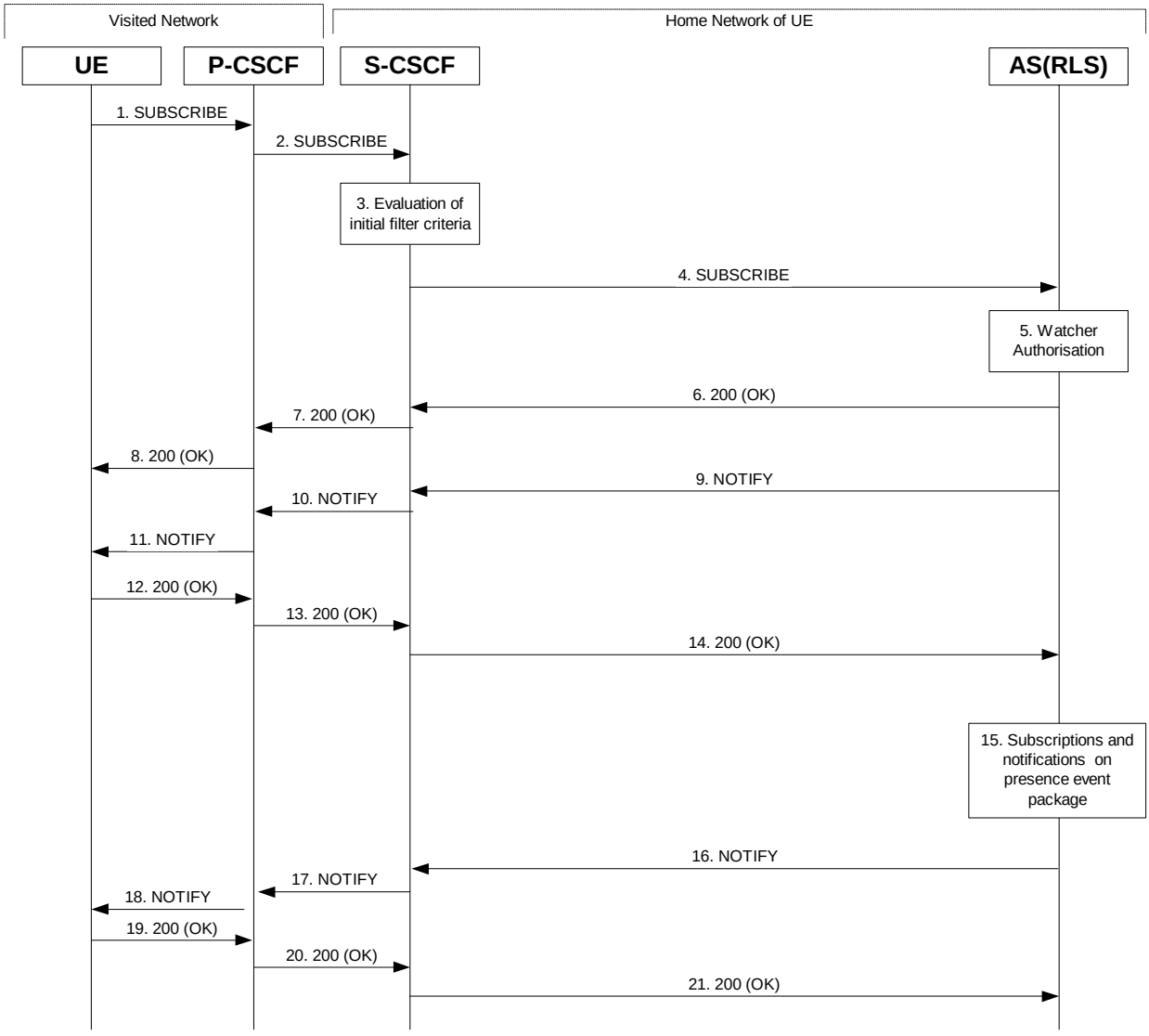


Figure A.3.3.1-1: Watcher subscribing to resource list

Figure A.3.3.1-1 shows a watcher subscribing to resource list event notification. The details of the signalling flows are as follows:

1. **SUBSCRIBE request (UE to P-CSCF) – see example in table A.3.3.1-1**

   A watcher agent in a UE wishes to watch a number of presentities, or certain presence information~~tuples~~ of these presentities. The list of presentities are identified by a SIP URI. In order to initiate a subscription to the RLS, the UE generates a SUBSCRIBE request indicating support for "eventlist", together with an indication of the length of time this periodic subscription should last.

Table A.3.3.1-1: SUBSCRIBE request (UE to P-CSCF)

```
SUBSCRIBE sip:user1_list1@home1.net SIP/2.0
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
Max-Forwards: 70
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
Route: <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>, <sip:orig@scscf1.home1.net;lr>
P-Preferred-Identity: <sip:user1_public1@home1.net>
Privacy: none
From: <sip:user1_public1@home1.net>;tag=31415
To: <sip:user1_list1@home1.net>
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 123 SUBSCRIBE
Require: sec-agree
Proxy-Require: sec-agree
Security-Verify: ipsec-3gpp; q=0.1; alg=hmac-sha-1-96; spi-c=98765432; spi-s=87654321; port-
      c=8642; port-s=7531
Event: presence
Supported: eventlist
Expires: 7200
Accept: application/pidf+xml, application/rlmi+xml, multipart/related
Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>
Content-Length: 0
```

Request-URI: SIP URI of the resource list representing the collection of public user identities whose events the subscriber subscribes to.

Event: This field is populated with the value "presence" to specify the use of the presence package.

Accept: This field is populated with the value "application/pidf+xml", "application/rlmi+xml" and "multipart/related" indicating that the UE supports both body types for the eventlist extension additionally to PIDF.

Supported: This field is populated with the value 'eventlist' to specify the support for the eventlist extension.

To: Same as the Request-URI.

2. **SUBSCRIBE request (P-CSCF to S-CSCF) – see example in table A.3.3.1-2**

   The P-CSCF looks up the serving network information for the public user identity that was stored during the registration procedure. The SUBSCRIBE request is forwarded to S-CSCF#1. A Route header is inserted into SUBSCRIBE request. The information for the Route header is taken from the service route determined during registration.

Table A.3.3.1-2: SUBSCRIBE request (P-CSCF to S-CSCF)

```
SUBSCRIBE sip:user1_list1@home1.net SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
P-Access-Network-Info:
Route: <sip:orig@scscf1.home1.net;lr>
Max-Forwards: 69
P-Asserted-Identity: <sip:user1_public1@home1.net>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"
Privacy:
Record-Route: <sip:pcscf1.visited1.net;lr>
Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Supported:
Expires:
Accept:
Contact:
Content-Length:
```

3. **Evaluation of initial filter criteria**

   The S-CSCF validates the service profile of this subscriber and evaluates the initial filter criteria. Assuming that sip:user1_list1@home1.net is a statically created PSI, sip:user1_list1@home1.net is included in the service profile as part of an originating initial Filter Criteria with Service Trigger Point of Method = SUBSCRIBE AND Supported = 'eventlist' AND Request-URI = sip:user1_list1@home1.net that informs the S-CSCF to route the SUBSCRIBE request to the application server sip:rls.home1.net.

   If there is no initial filter criteria for this PSI (sip:user1_list1@home1.net), the assumption is that the PSI is a sub domain-based PSI. The procedure defined in RFC 3263 [18] with DNS NAPTR and SRV queries may then be used to get the IP address of the application server home1.net.

Editor's note: The handling of alternative PSI routing examples should be described and expanded in subclause A.3.1 rather than in this location.

4. **SUBSCRIBE request (S-CSCF to RLS) – see example in table A.3.3.1-4**

   The S-CSCF forwards the SUBSCRIBE request to the RLS.

Table A.3.3.1-4: SUBSCRIBE request (S-CSCF to RLS)

```
SUBSCRIBE sip:user1_list1@home1.net SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK344a65.1, SIP/2.0/UDP
      pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
Max-Forwards: 68
P-Access-Network-Info:
P-Asserted-Identity: <sip:user1_public1@home1.net>, <tel:+1-212-555-1111>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Privacy:
Record-Route: <sip:orig@scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
Route: <sip:rls.home1.net;lr>, <sip:orig@scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Supported:
Expires:
Accept:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF populates the P-Charging-Function-Addresses header field to be passed to the RLS.

5. **Authorization of watcher**

The RLS performs the necessary authorization checks on the originator to ensure that he/she is authorized to use the resource list. In this example this condition has been met, so the PS sends a 200 (OK) response to the S-CSCF. If the previous condition failed, then a 403 (Forbidden) response would be sent to the S-CSCF.

6. **200 (OK) response (RLS to S-CSCF) - see example in table A.3.3.1-6**

The RLS sends the response to the S-CSCF.

Table A.3.3.1-6: 200 (OK) response (RLS to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK344a65.1, SIP/2.0/UDP
      pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"; orig-ioi=home1.net;
      term-ioi=home1.net
Record-Route:
From:
To: <sip:user1_list1@home1.net>;tag=151170
Call-ID:
CSeq:
Require: eventlist
Expires:
Contact:
Content-Length: 0
```

P-Charging-Vector: The RLS stores the terminating Inter Operator Identifier (IOI) parameter and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

7. **200 (OK) response (S-CSCF to P-CSCF) - see example in table A.3.3.1-7**

The S-CSCF forwards the response to the P-CSCF.

Table A.3.3.1-7: 200 (OK) response (S-CSCF to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"
Record-Route:
From:
To:
Call-ID:
CSeq:
Require:
Expires:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the terminating Inter Operator Identifier (IOI) parameter received.

8. **200 (OK) response (P-CSCF to UE) - see example in table A.3.3.1-8**

The P-CSCF forwards the response to the watcher agent in the UE.

Table A.3.3.1-8: 200 (OK) response (P-CSCF to UE)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
Record-Route: <sip:orig@scscf1.home1.net;lr>, <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Require:
Expires:
Contact:
Content-Length:
```

9. **NOTIFY request (RLS to S-CSCF) - see example in table A.3.3.1-9**

The RLS generates a NOTIFY request including the RLMI document as a result of the SUBSCRIBE request.

Table A.3.3.1-9 NOTIFY request (RLS to S-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 70
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
From: <sip:user1_list1@home1.net>;tag=151170
To: <sip:user1_public1@home1.net>;tag=31415
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 89 NOTIFY
Subscription-State: active;expires=7200
Require: eventlist
Event: presence
Contact: <sip:rls.home1.net>
Content-Type: application/rlmi+xml;charset="UTF-8"
Content-Length:

<?xml version="1.0" encoding="UTF-8"?>
  <list xmlns="urn:ietf:params:xml:ns:rmli"
       uri="sip:user1_list1@home1.net" version="1" fullState="true">
     <resource uri="pres:user2_public1@home2.net" name="Kovacs Janos">
       <instance id="hqzsuxtfyq" state="active" cid="ZvSvkz@rls.home1.net"/>
     </resource>
     <resource uri="pres:user3_public1@home3.net" name="Szabo Bela">
       <instance id="aakdsjklsa" state="active" cid="HJjbssk@rls.home1.net"/>
     </resource>
  </list>
```

P-Charging-Vector: The RLS inserts this header and populates the icid parameters with a globally unique value and adds the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The RLS populates the P-Charging-Function-Addresses header field to be passed to the S-CSCF.

10. **NOTIFY request (S-CSCF to P-CSCF) - see example in table A.3.3.1-10**

The S-CSCF forwards the NOTIFY request to the P-CSCF.

Table A.3.3.1-10: NOTIFY request (S-CSCF to P-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 69
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"
P-Charging-Function-Addresses:
Route: <sip:pcscf1.visited1.net;lr>
Record-Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Require:
Event:
Contact:
Content-Length:

(…)
```

P-Charging-Vector: The S-CSCF stores originating Inter Operator Identifier (IOI) parameter received.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the P-CSCF.

11. **NOTIFY request (P-CSCF to UE) – see example in table A.3.3.1-11**

The P-CSCF forwards the NOTIFY request to the watcher application in the UE.

Table A.3.3.1-11: NOTIFY request (P-CSCF to UE)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 68
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Require:
Event:
Contact:
Content-Length:

(…)
```

12. **200 (OK) response (UE to P-CSCF) – see example in table A.3.3.1-12**

The UE acknowledges the NOTIFY request with a 200 (OK) response to the P-CSCF.

Table A.3.3.1-12: 200 (OK) response (UE to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

13. **200 (OK) response (P-CSCF to S-CSCF) – see example in table A.3.3.1-13**

The P-CSCF forwards the 200 (OK) response to the S-CSCF.

Table A.3.3.1-13: 200 (OK) response (P-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"
From:
To:
Call-ID:
CSeq:
Content-Length:
```

14. **200 (OK) response (S-CSCF to RLS) - see example in table A.3.3.1-14**

The S-CSCF#2 forwards the response to the RLS in the home network of the UE.

Table A.3.3.1-14: 200 (OK) response (S-CSCF to RLS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"; orig-ioi=home1.net;
      term-ioi=home1.net
From:
To:
Call-ID:
CSeq:
Content-Length:
```

P-Charging-Vector: The S-CSCF inserts the originating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

15. **Subscriptions and notifications on presence event package**

After the RLS generated a NOTIFY request to inform the UE about the subscription state, the RLS generates the necessary SUBSCRIBE requests to the presentities present in the resource list as described in subclause A.3.4.1. As soon as it receives NOTIFY request(s) about a state change in one or more presentities, it generates a NOTIFY request.

16. **NOTIFY request (RLS to S-CSCF) – see example in table A.3.3.1-16**

The RLS copies the body of the incoming NOTIFY request(s) into the body of the outgoing NOTIFY request using MIME type multipart/related. Further notification sent by the RLS may contain either the full or the partial set of presence information (only the presence information that has changed since the last notification) as described in draft-ietf-simple-event-list-05~~4~~ [22].

In this example it is assumed that the RLS has received two NOTIFY requests from presentities sip:user2_public1@home2.net and sip:user3_public1@home3.net before generating the NOTIFY request in table A.3.3.1-16 to the UE.

Table A.3.3.1-16 NOTIFY request (RLS to S-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 70
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
From: <sip:user1_list1@home1.net>;tag=151170
To: <sip:user1_public1.home1.net>;tag=31415
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 90 NOTIFY
Subscription-State: active;expires=5000
Require: eventlist
Event: presence
Contact: <sip:rls.home1.net>
Content-Type: multipart/related;type="application/rlmi+xml";
      start="<nXYxAE@rls.home1.net>";boundary="50UBfW7LSCVLtggUPe5z"
Content-Length: (...)

--50UBfW7LSCVLtggUPe5z
Content-Transfer-Encoding: binary
Content-ID: <nXYxAE@rls.home1.net>
Content-Type: application/rlmi+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
  <list xmlns="urn:ietf:params:xml:ns:rmli"
        uri="sip:user1_list1@home1.net" version="1" fullState="true">
    <resource uri="pres:user2_public1@home2.net" name="Kovacs Janos">
      <instance id="hqzsuxtfyq" state="active" cid="ZvSvkz@rls.home1.net"/>
    </resource>
    <resource uri="pres:user3_public1@home3.net" name="Szabo Bela">
      <instance id="aakdsjklsa" state="active" cid="HJjbssk@rls.home1.net"/>
    </resource>
  </list>

--50UBfW7LSCVLtggUPe5z
Content-Transfer-Encoding: binary
Content-ID: <ZvSvkz@rls.home1.net>
Content-Type: application/pidf+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
   <presence xmlns="urn:ietf:params:xml:ns:pidf"
             xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
             xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
             xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid-person"
             xmlns:dmp="urn:ietf:params:xml:ns:pidf:person"
             xmlns:pcp="urn:ietf:params:xml:ns:pidf:servcaps~~simple-prescaps-ext~~"
             xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
             entity="pres:user2_public1@home2.net">

     <tuple id="a8098a.672364762364">
       <status>
         <basic>open</basic>
         <es:privacy><text/></es:privacy>
         <es:status-icon>http://example.com/~user2/icon.gif</es:status-icon>
       </status>
       <et:class>sip</et:class>
       <pcp:video>false</pcp:video>
       <pcp:audio>true</pcp:audio>
       <contact priority="0.8">sip:user2_public1@home2.net</contact>
       <note xml:lang="en">Don't Disturb Please!</note>
       <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
```

```
        <timestamp>2003-08-27T11:49:29Z</timestamp>
      </tuple>
      <tuple id="a8098a.672364762364">
        <status>
          <basic>open</basic>
          <es:activity>meeting</es:activity>
          <es:placetype until="2003-08-27T17:30:00Z">office</es:placetype>
          <es:privacy>private</es:privacy>
          <es:idle since="2003-08-27T10:43:00Z"/>
          <pcp:prescaps>
            <pcp:video negated="false"></pcp:video>
            <pcp:mobility>mobile</pcp:mobility>
            <pcp:audio negated="true"></pcp:audio>
          </pcp:prescaps>
        </status>
        <et:class>sip</et:class>
        <et:contact-type>service</et:contact-type>
        <contact priority="0.8">sip:user2_public1@home2.net</contact>
        <note xml:lang="en">Don't Disturb Please!</note>
        <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
        <timestamp>2003-08-27T11:49:29Z</timestamp>
      </tuple>

      <tuple id="sfddsj74.78">
        <status>
          <basic>open</basic>
        </status>
        <et:class>presentity</et:class>
        <et:contact-type>presentity</et:contact-type>
        <ci:homepage>http://example.com/~user2</ci:homepage>
        <ci:icon>http://example.com/~user2/icon.gif</ci:icon>
        <ci:card>http://example.com/~user2/card.vcd</ci:card>
        <note xml:lang="en">I'm in a boring meeting</note>
        <note xml:lang="en">I'll be in Tokyo next week</note>
        <timestamp>2003-08-27T11:49:29Z</timestamp>
      </tuple>

      <tuple id="jklhgf9788934774.78">
        <status>
          <basic>open</basic>
        </status>
        <et:class>assistant</et:class>
        <et:contact-type>presentity</et:contact-type>
        <et:relationship>assistant</et:relationship>
        <contact priority="1.0">tel:+1-212-555-2222</contact>
        <note xml:lang="en">She's my secretary</note>
        <timestamp>2003-08-27T11:49:29Z</timestamp>
      </tuple>

      <dmp:person>
        <ep:class>presentity</ep:class>
        <ci:homepage>http://example.com/~user2</ci:homepage>
        <ci:card>http://example.com/~user2/card.vcd</ci:card>
        <dmp:status>
          <ep:activities><ep:meeting/></ep:activities>
          <ep:place-type until="2003-08-27T17:30:00Z">office</ep:place-type>
        </dmp:status>
      </dmp:person>

   </presence>

--50UBfW7LSCVLtggUPe5z
Content-Transfer-Encoding: binary
Content-ID: <ZvSvkz@pres.example.com>
Content-Type: application/pidf+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
   <presence xmlns="urn:ietf:params:xml:ns:pidf"
             xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
             xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
             xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid-person"
             xmlns:dmp="urn:ietf:params:xml:ns:pidf:person"
             xmlns:pcp="urn:ietf:params:xml:ns:pidf:servcaps-simple-prescaps-ext"
             entity="pres:user3_public1@home3.net">

      <tuple id="h7833hjkk.dsajfjdsaf">
        <status>
          <basic>closed</basic>
```

```
        <es:activity>vacation</es:activity>
        <es:placetype until="2003-09-10T17:30:00Z">ship</es:placetype>
        <es:privacy><es:text/>private</es:privacy>
        <es:idle since="2003-06-27T10:43:00Z"/>
        <pcp:prescaps>
          <pcp:video negated="false">false</pcp:video>
          <pcp:mobility>mobile</pcp:mobility>
          <pcp:audio negated="true">true</pcp:audio>
        </pcp:prescaps>
      </status>
      <et:class>sip</et:class>
      <et:contact-type>service</et:contact-type>
      <contact priority="0.8">sip:user3_public1@home3.net</contact>
      <note xml:lang="en">Don't Disturb Please!</note>
      <note xml:lang="hu">Senki se merjen zavarni!</note>
      <timestamp>2003-08-27T11:48:59Z</timestamp>
    </tuple>

    <tuple id="sfddsj74.78">
      <status>
        <basic>open</basic>
      </status>
      <et:class>presentity</et:class>
      <et:contact-type>presentity</et:contact-type>
      <ci:homepage>http://example.com/~user3</ci:homepage>
      <ci:icon>http://example.com/~user3/icon.gif</ci:icon>
      <ci:card>http://example.com/~user3/card.vcd</ci:card>
      <note xml:lang="en">I'm on vacation</note>
      <timestamp>2004-10-10T12:00:30Z</timestamp>
    </tuple>

    <tuple id="sajdhdsahjh75vvcb774.78">
      <status>
        <basic>open</basic>
      </status>
      <et:class>supervisor</et:class>
      <et:type>presentity</et:type>
      <et:relationship>supervisor</et:relationship>
      <contact priority="1.0">tel:+1-858-204-9141</contact>
      <note xml:lang="en">He's my supervisor</note>
      <timestamp>2003-08-27T11:48:59Z</timestamp>
    </tuple>

    <dmp:person>
      <ci:homepage>http://example.com/~user3</ci:homepage>
      <ci:card>http://example.com/~user3/card.vcd</ci:card>
      <et:class>presentity</et:class>
      <dmp:status>
        <ep:activities><ep:vacation/></ep:activities>
        <ep:place-type until="2003-09-10T17:30:00Z">ship</ep:place-type>
      </dmp:status>
    </dmp:person>

  </presence>

--50UBfW7LSCVLtggUPe5z--
```

P-Charging-Vector: The RLS inserts this header and populates the icid parameters with a globally unique value and adds the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The RLS populates the P-Charging-Function-Addresses header field to be passed to the S-CSCF.

Content-Type: Set to the value of the Accept: header received in the SUBSCRIBE request.

The message body in the NOTIFY request that carries the presence information of the presentity is formed as indicated in draft-ietf-simple-event-list-054 [22], draft-ietf-simple-presence-data-model-01 [x], draft-ietf-simple-rpid-043 [26], draft-ietf-simple-cipid-031 [32] and draft-ietf-simple-prescaps-ext-020 [25].

17. **NOTIFY request (S-CSCF to P-CSCF) - see example in table A.3.3.1-17**

The S-CSCF forwards the NOTIFY request to the P-CSCF.

Table A.3.3.1-17: NOTIFY request (S-CSCF to P-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 69
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"
P-Charging-Function-Addresses:
Route: <sip:pcscf1.visited1.net;lr>
Record-Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Require:
Event:
Contact:
Content Type:
Content-Length:

(…)
```

P-Charging-Vector: The RLS stores the originating Inter Operator Identifier (IOI) parameter received.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the P-CSCF.

18. **NOTIFY request (P-CSCF to UE) - see example in table A.3.3.1-18**

The P-CSCF forwards the NOTIFY request to the watcher application in the UE.

Table A.3.3.1-18: NOTIFY request (P-CSCF to UE)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 68
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Require:
Event:
Content-Type:
Content-Length:

(…)
```

19. **200 (OK) response (UE to P-CSCF) – see example in table A.3.3.1-19**

The UE acknowledges the NOTIFY request with a 200 (OK) response to the P-CSCF.

Table A.3.3.1-19: 200 (OK) response (UE to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

20. **200 (OK) response (P-CSCF to S-CSCF) - see example in table A.3.3.1-20**

The P-CSCF forwards the 200 (OK) response to the S-CSCF.

Table A.3.3.1-20: 200 (OK) response (P-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"
From:
To:
Call-ID:
CSeq:
Content-Length:
```

21. **200 (OK) response (S-CSCF to RLS) - see example in table A.3.3.1-21**

The S-CSCF#2 forwards the response to the RLS in the home network of the UE.

Table A.3.3.1-21: 200 (OK) response (S-CSCF to RLS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"; orig-ioi=home1.net;
      term-ioi=home1.net
From:
To:
Call-ID:
CSeq:
Content-Length:
```

P-Charging-Vector: The S-CSCF inserts the originating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

A.3.3.2 Watcher subscribing to a resource list, UE in visited network - successful subscription

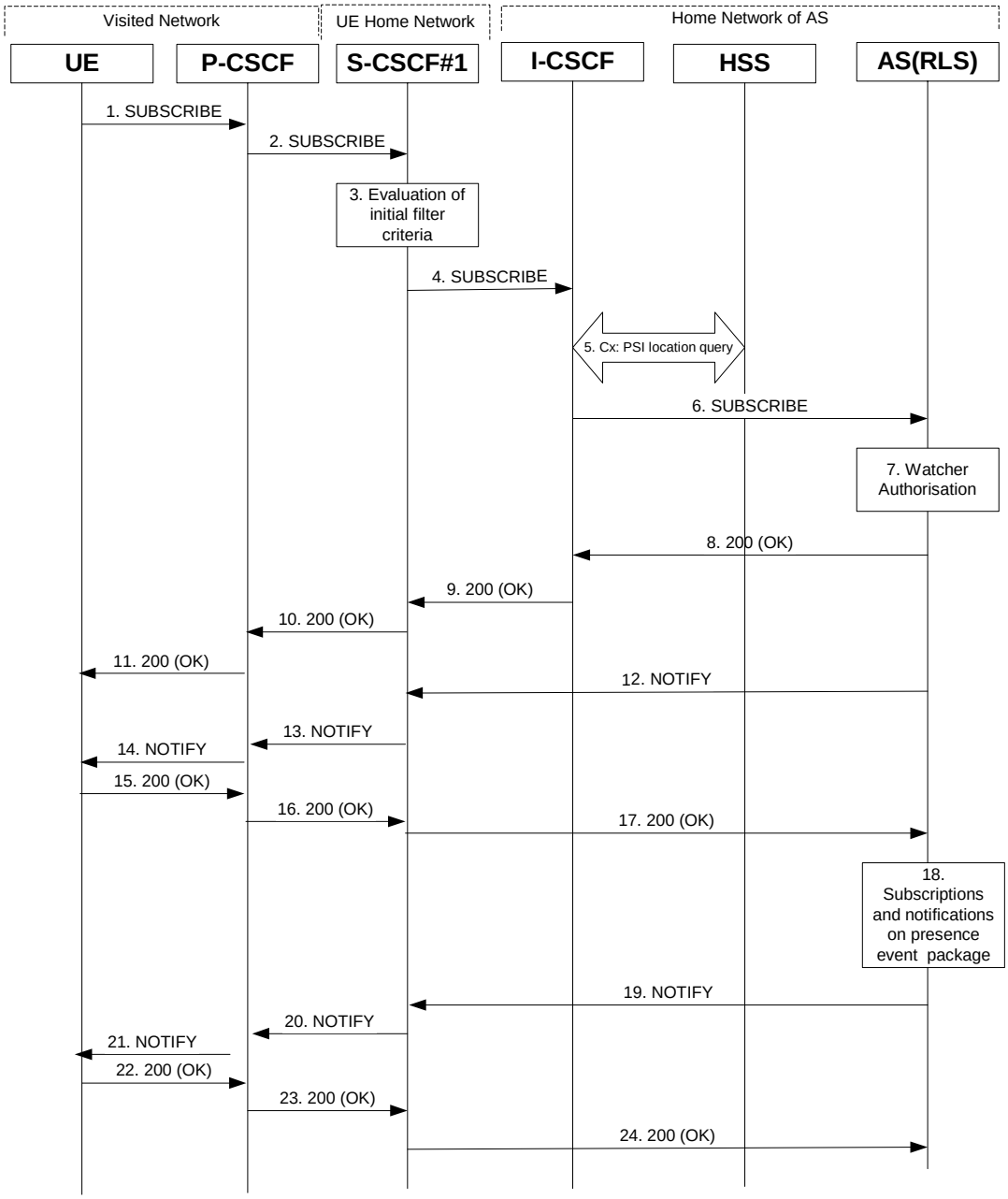


Figure A.3.3.2-1 Watcher subscribing to resource list

Figure A.3.3.2-1 shows a watcher subscribing to resource list event notification. The details of the signalling flows are as follows:

1. **SUBSCRIBE request (UE to P-CSCF) - see example in table A.3.3.2-1**

   A watcher agent in a UE wishes to watch a number of presentities, or certain presence information~~tuples~~ of these presentities. The list of presentities are identified by a SIP URI. In order to initiate a subscription to the RLS, the UE generates a SUBSCRIBE request indicating support for 'eventlist', together with an indication of the length of time this periodic subscription should last.

Table A.3.3.2-1: SUBSCRIBE request (UE to P-CSCF)

```
SUBSCRIBE sip:user2_list1@home2.net SIP/2.0
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
Max-Forwards: 70
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
Route: <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>, <sip:orig@scscf1.home1.net;lr>
P-Preferred-Identity: <sip:user1_public1@home1.net>
Privacy: none
From: <sip:user1_public1@home1.net>;tag=31415
To: <sip:user2_list1@home1.net>
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 123 SUBSCRIBE
Require: sec-agree
Proxy-Require: sec-agree
Security-Verify: ipsec-3gpp; q=0.1; alg=hmac-sha-1-96; spi-c=98765432; spi-s=87654321; port-
      c=8642; port-s=7531
Event: presence
Supported: eventlist
Expires: 7200
Accept: application/pidf+xml, application/rlmi+xml, multipart/related
Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>
Content-Length: 0
```

Request-URI: SIP URI of the resource list representing the collection of public user identities whose events the subscriber subscribes to.

Event: This field is populated with the value "presence" to specify the use of the presence package.

Accept: This field is populated with the value "application/pidf+xml", "application/rlmi+xml" and "multipart/related" indicating that the UE supports the eventlist extension additionally to PIDF.

Supported: This field is populated with the value 'eventlist' to specify the support for the eventlist extension.

To: Same as the Request-URI.

2. **SUBSCRIBE request (P-CSCF to S-CSCF) - see example in table A.3.3.2-2**

The P-CSCF looks up the serving network information for the public user identity that was stored during the registration procedure. The SUBSCRIBE request is forwarded to S-CSCF#1. A Route header is inserted into SUBSCRIBE request. The information for the Route header is taken from the service route determined during registration.

Table A.3.3.2-2: SUBSCRIBE request (P-CSCF to S-CSCF)

```
SUBSCRIBE sip:user2_list1@home2.net SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
P-Access-Network-Info:
Route: <sip:orig@scscf1.home1.net;lr>
Max-Forwards: 69
P-Asserted-Identity: <sip:user1_public1@home1.net>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"
Privacy:
Record-Route: <sip:pcscf1.visited1.net;lr>
Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Supported:
Expires:
Accept:
Contact:
Content-Length:
```

3. **Evaluation of initial filter criteria**

S-CSCF#1 validates the service profile of this subscriber and evaluates the initial filter criteria. For this example, assume no application server involvement.

4. **SUBSCRIBE request (S-CSCF to I-CSCF) - see example in table A.3.3.2-4**

S-CSCF#1 performs an analysis of the destination address. As the destination address points to a resource that is in a different network as the S-CSCF, the S-CSCF sends the request to the I-CSCF of home2.net.

Table A.3.3.2-4: SUBSCRIBE request (S-CSCF to I-CSCF)

```
SUBSCRIBE sip:user2_list1@home2.net SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK344a65.1, SIP/2.0/UDP
      pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
Max-Forwards: 68
P-Asserted-Identity: <sip:user1_public1@home1.net>, <tel:+1-212-555-1111>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Privacy:
Record-Route: <orig@sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Supported:
Expires:
Accept:
Contact:
Content-Length:
```

5. **PSI location query**

   The I-CSCF sends a query to the HSS to find the RLS where sip:user2_list1@home2.net is hosted. The HSS responds with the address of the RLS.

Editor's Note: More detailed information is needed here, similar to the Cx interface information given in 3GPP TS 24.228 [8].

6. **SUBSCRIBE request (I-CSCF to RLS) - see example in table A.3.3.2-6**

   The I-CSCF forwards the SUBSCRIBE request to the RLS.

Table A.3.3.2-6: SUBSCRIBE request (I-CSCF to S-CSCF)

```
SUBSCRIBE sip:user2_list1@home2.net SIP/2.0
Via: SIP/2.0/UDP icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK344a65.1, SIP/2.0/UDP
      pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
Max-Forwards: 67
P-Asserted-Identity:
P-Charging-Vector:
Privacy:
Record-Route:
Route: <sip:rls.home2.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Supported:
Expires:
Accept:
Contact:
Content-Length:
```

7. **Authorization of watcher**

   The RLS performs the necessary authorization checks on the originator to ensure that he/she is authorized to use the resource list. In this example this condition has been met, so the PS sends a 200 (OK) response to the S-CSCF. If the previous condition failed, then a 403 (Forbidden) response would be sent to the S-CSCF.

8. **200 (OK) response (RLS to I-CSCF) - see example in table A.3.3.2-8**

   The RLS sends the response to the S-CSCF.

Table A.3.3.2-8: 200 (OK) response (RLS to I-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK344a65.1, SIP/2.0/UDP
      pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"; orig-ioi=home1.net;
      term-ioi=home2.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Record-Route:
From:
To: <sip:user2_list1@home2.net>;tag=151170
Call-ID:
CSeq:
Require: eventlist
Expires:
Contact: <sip:rls.home2.net>
Content-Length: 0
```

P-Charging-Vector: The RLS stores the originating Inter Operator Identifier (IOI) parameter and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The RLS stores the P-Charging-Function-Addresses header field and passes this header to the I-CSCF.

9. **200 (OK) response (I-CSCF to S-CSCF) - see example in table A.3.3.2-9**

The I-CSCF forwards the response to the S-CSCF.

Table A.3.3.2-9: 200 (OK) response (I-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK344a65.1, SIP/2.0/UDP
      pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
P-Charging-Vector:
Record-Route:
From:
To:
Call-ID:
CSeq:
Require:
Expires:
Contact:
Content-Length: 0
```

P-Charging-Vector: The RLS stores the header and passes this header to the S-CSCF.

10. **200 (OK) response (S-CSCF to P-CSCF) - see example in table A.3.3.2-10**

The S-CSCF forwards the response to the P-CSCF.

Table A.3.3.2-10: 200 (OK) response (S-CSCF to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"
Record-Route:
From:
To:
Call-ID:
CSeq:
Require:
Expires:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the terminating Inter Operator Identifier (IOI) parameter.

11. **200 (OK) response (P-CSCF to UE) - see example in table A.3.3.2-11**

The P-CSCF forwards the response to the watcher agent in the UE.

Table A.3.3.2-11: 200 (OK) response (P-CSCF to UE)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
Record-Route: <sip:orig@scscf1.home1.net;lr>, <sip:pcscf1.home1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Require:
Expires:
Contact:
Content-Length:
```

12. **NOTIFY request (RLS to S-CSCF) – see example in table A.3.3.2-12**

The RLS generates a NOTIFY request including the RLMI document as a result of the SUBSCRIBE request.

Table A.3.3.2-12: NOTIFY request (RLS to S-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP rls.home2.net;branch=z9hG4bK240f34.1
Max-Forwards: 70
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=123551024"; orig-ioi=home1.net
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
From: <sip:user2_list1@home1.net>;tag=151170
To: <sip:user1_public1.home1.net>;tag=31415
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 89 NOTIFY
Subscription-State: active;expires=5000
Require: eventlist
Event: presence
Contact: <sip:rls.home1.net>
Content-Type: application/rlmi+xml;charset="UTF-8"
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
  <list xmlns="urn:ietf:params:xml:ns:rmli"
        uri="sip:user1_list1@home1.net" version="1" fullState="true">
     <resource uri="pres:user2_public1@home2.net" name="Kovacs Janos">
       <instance id="hqzsuxtfyq" state="active" cid="ZvSvkz@rls.home2.net"/>
     </resource>
     <resource uri="pres:user3_public1@home2.net" name="Szabo Bela">
       <instance id="aakdsjklsa" state="active" cid="HJjbssk@rls.home2.net"/>
     </resource>
  </list>
```

P-Charging-Vector: The RLS populates the icid parameter with a globally unique value and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

13. **NOTIFY request (S-CSCF to P-CSCF) - see example in table A.3.3.2-13**

The S-CSCF#1 forwards the NOTIFY request to the P-CSCF.

Table A.3.3.2-13: NOTIFY request (S-CSCF to P-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home2.net;branch=z9hG4bK240f34.1
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=123551024"
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Max-Forwards: 69
Record-Route: <sip:scscf1.home1.net;lr>
Route: <sip:pcscf1.visited1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Require:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter.

P-Charging-Function-Addresses: The RLS populates the P-Charging-Function-Addresses header field to be passed to the I-CSCF.

14. **NOTIFY request (P-CSCF to UE) - see example in table A.3.3.2-14**

The P-CSCF forwards the NOTIFY request to the watcher application in the UE.

Table A.3.3.2-14: NOTIFY request (P-CSCF to UE)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home2.net;branch=z9hG4bK240f34.1
Max-Forwards: 68
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.home1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Require:
Event: Contact:
Content-Type:
Content-Length:

(…)
```

15. **200 (OK) response (UE to P-CSCF) - see example in table A.3.3.2-15**

The UE acknowledges the NOTIFY request with a 200 (OK) to the P-CSCF.

Table A.3.3.2-15: 200 (OK) response (UE to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home2.net;branch=z9hG4bK240f34.1
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

16. **200 (OK) response (P-CSCF to S-CSCF) – see example in table A.3.3.2-16**

The P-CSCF forwards the 200 (OK) response to the S-CSCF#1.

Table A.3.3.2-16: 200 (OK) response (P-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home2.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=123551024"
From:
To:
Call-ID:
CSeq:
Content-Length:
```

17. **200 (OK) response (S-CSCF to RLS) - see example in table A.3.3.2-17**

The S-CSCF#1 forwards the response to the RLS in the home network of the UE.

Table A.3.3.2-17: 200 (OK) response (S-CSCF to RLS)

```
SIP/2.0 200 OK
Via: SP/2.0/UDP rls.home2.net;branch=z9hG4bK240f34.1
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=123551024"; orig-ioi=home1.net:
      term-ioi=home1.net
From:
To:
Call-ID:
CSeq:
Content-Length:
```

P-Charging-Vector: The S-CSCF insertes the originating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

18. **Subscriptions and notifications on presence event package**

After the RLS generated a 200 (OK) response to the SUBSCRIBE request from the UE, it generates the necessary SUBSCRIBE requests to the presentities present in the resource list as described in subclause A.3.4.1. As soon as it receives NOTIFY request(s) about a state change in one or more presentities, it generates a NOTIFY request.

19. **NOTIFY request (RLS to S-CSCF) – see example in table A.3.3.2-19**

The RLS copies the body of the incoming NOTIFY request(s) into the body of the outgoing NOTIFY request using MIME type multipart/related. Further notification sent by the RLS contain may contain either the full or the partial set of presence information (only the presence information that has changed since the last notification) as described in draft-ietf-simple-event-list-054 [22].

In this example it is assumed that the RLS receives two NOTIFY requests from presentities sip:user2_public1@home2.net and sip:user3_public1@home3.net before generating the NOTIFY request in subclause A.3.3.2-23 to the UE.

Table A.3.3.2-19: NOTIFY request (RLS to S-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP rls.home2.net;branch=z9hG4bK240f34.1
Max-Forwards: 70
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"; orig-ioi=home1.net
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
From: <sip:user2_list1@home1.net>;tag=151170
To: <sip:user1_public1.home1.net>;tag=31415
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 89 NOTIFY
Subscription-State: active;expires=5000
Require: eventlist
Event: presence
Contact: <sip:rls.home2.net>
Content-Type: multipart/related;type="application/rlmi+xml";
       start="<nXYxAE@rls.home2.net>";boundary="50UBfW7LSCVLtggUPe5z"
Content-Length: (...)

--50UBfW7LSCVLtggUPe5z
Content-Transfer-Encoding: binary
Content-ID: <nXYxAE@rls.home2.net>
Content-Type: application/rlmi+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
  <list xmlns="urn:ietf:params:xml:ns:rmli"
        uri="sip:user1_list1@home1.net" version="1" fullState="true">
     <resource uri="pres:user2_public1@home2.net" name="Kovacs Janos">
       <instance id="hqzsuxtfyq" state="active" cid="ZvSvkz@rls.home2.net"/>
     </resource>
     <resource uri="pres:user3_public1@home3.net" name="Szabo Bela">
       <instance id="aakdsjklsa" state="active" cid="HJjbssk@rls.home2.net"/>
     </resource>
   </list>

--50UBfW7LSCVLtggUPe5z
Content-Transfer-Encoding: binary
Content-ID: <ZvSvkz@rls.home2.net>
Content-Type: application/pidf+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
   <presence xmlns="urn:ietf:params:xml:ns:pidf"
             xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
             xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
             xmlns:pcp="urn:ietf:params:xml:ns:simple-prescaps-ext"
             xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
             entity="pres:user2_public1@home2.net">

     <tuple id="a8098a.672364762364">
       <status>
         <basic>open</basic>
         <es:activity>meeting</es:activity>
         <es:placetype until="2003-08-27T17:30:00Z">office</es:placetype>
         <es:privacy>private</es:privacy>
         <es:idle since="2003-08-27T10:43:00Z"/>
         <pcp:prescaps>
           <pcp:video negated="false"></pcp:video>
           <pcp:mobility>mobile</pcp:mobility>
           <pcp:audio negated="true"></pcp:audio>
         </pcp:prescaps>
       </status>
       <et:class>sip</et:class>
       <et:contact-type>service</et:contact-type>
       <contact priority="0.8">sip:user2_public1@home2.net</contact>
```

```
      <note xml:lang="en">Don't Disturb Please!</note>
      <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
      <timestamp>2003-08-27T11:49:29Z</timestamp>
    </tuple>

    <tuple id="sfddsj74.78">
      <status>
        <basic>open</basic>
      </status>
      <et:class>presentity</et:class>
      <et:contact-type>presentity</et:contact-type>
      <ci:homepage>http://example.com/~user2</ci:homepage>
      <ci:icon>http://example.com/~user2/icon.gif</ci:icon>
      <ci:card>http://example.com/~user2/card.vcd</ci:card>
      <note xml:lang="en">I'm in a boring meeting</note>
      <note xml:lang="en">I'll be in Tokyo next week</note>
      <timestamp>2004-10-10T12:00:30Z</timestamp>
    </tuple>

    <tuple id="jklhgf9788934774.78">
      <status>
        <basic>open</basic>
      </status>
      <et:class>assistant</et:class>
      <et:contact-type>presentity</et:contact-type>
      <et:relationship>assistant</et:relationship>
      <contact priority="1.0">tel:+1-212-555-2222</contact>
      <note xml:lang="en">She's my secretary</note>
      <timestamp>2003-08-27T11:49:29Z</timestamp>
    </tuple>

  </presence>

--50UBfW7LSCVLtggUPe5z
Content-Transfer-Encoding: binary
Content-ID: <ZvSvkz@pres.example.com>
Content-Type: application/pidf+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
  <presence xmlns="urn:ietf:params:xml:ns:pidf"
            xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
            xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
            xmlns:pcp="urn:ietf:params:xml:ns:simple-prescaps-ext"
            xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
            entity="pres:user3_public1@home3.net">

    <tuple id="h7833hjkk.dsajfjdsaf">
      <status>
        <basic>closed</basic>
        <es:activity>vacation</es:activity>
        <es:placetype until="2003-09-10T17:30:00Z">ship</es:placetype>
        <es:privacy>private</es:privacy>
        <es:idle since="2003-06-27T10:43:00Z"/>
        <pcp:prescaps>
          <pcp:video negated="false"></pcp:video>
          <pcp:mobility>mobile</pcp:mobility>
          <pcp:audio negated="true"></pcp:audio>
        </pcp:prescaps>
      </status>
      <et:class>sip</et:class>
      <et:content-type>service</et:content-type>
      <contact priority="0.8">sip:user3_public1@home3.net</contact>
      <note xml:lang="en">Don't Disturb Please!</note>
      <note xml:lang="hu">Senki se merjen zavarni!</note>
      <timestamp>2003-08-27T11:48:59Z</timestamp>
    </tuple>

    <tuple id="sfddsj74.78">
      <status>
        <basic>open</basic>
      </status>
      <et:class>presentity</et:class>
      <et:contact-type>presentity</et:contact-type>
      <ci:homepage>http://example.com/~user3</ci:homepage>
      <ci:icon>http://example.com/~user3/icon.gif</ci:icon>
      <ci:card>http://example.com/~user3/card.vcd</ci:card>
      <note xml:lang="en">I'm on vacation</note>
      <timestamp>2004-10-10T12:00:30Z</timestamp>
```

```
    </tuple>

    <tuple id="sajdhdsahjh75vvcb774.78">
      <status>
        <basic>open</basic>
      </status>
      <et:class>supervisor</et:class>
      <et:contact-type>presentity</et:contact-type>
      <et:relationship>supervisor</et:relationship>
      <contact priority="1.0">tel:+1-858-204-9141</contact>
      <note xml:lang="en">He's my supervisor</note>
      <timestamp>2003-08-27T11:48:59Z</timestamp>
    </tuple>

  </presence>
<?xml version="1.0" encoding="UTF-8"?>
  <presence xmlns="urn:ietf:params:xml:ns:pidf"
            xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
            xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
            xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid-person"
            xmlns:dmp="urn:ietf:params:xml:ns:pidf:person"
     xmlns:pcp="urn:ietf:params:xml:ns:pidf:servcaps"
            xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
            entity="pres:user2_public1@home2.net">

    <tuple id="a8098a.672364762364">
      <status>
        <basic>open</basic>
        <es:privacy><text/></es:privacy>
        <es:status-icon>http://example.com/~user2/icon.gif</es:status-icon>
      </status>
      <et:class>sip</et:class>
      <pcp:video>false</pcp:video>
      <pcp:audio>true</pcp:audio>
      <contact priority="0.8">sip:user2_public1@home2.net</contact>
      <note xml:lang="en">Don't Disturb Please!</note>
      <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
      <timestamp>2003-08-27T11:49:29Z</timestamp>
    </tuple>


    <tuple id="jklhgf9788934774.78">
      <status>
        <basic>open</basic>
      </status>
      <et:class>assistant</et:class>
      <et:relationship>assistant</et:relationship>
      <contact priority="1.0">tel:+1-212-555-2222</contact>
      <note xml:lang="en">She's my secretary</note>
      <timestamp>2003-08-27T11:49:29Z</timestamp>
    </tuple>

    <dmp:person>
      <ep:class>presentity</ep:class>
      <ci:homepage>http://example.com/~user2</ci:homepage>
      <ci:card>http://example.com/~user2/card.vcd</ci:card>
      <dmp:status>
        <ep:activities><ep:meeting/></ep:activities>
        <ep:place-type until="2003-08-27T17:30:00Z">office</ep:place-type>
      </dmp:status>
    </dmp:person>

  </presence>

--50UBfW7LSCVLtggUPe5z
Content-Transfer-Encoding: binary
Content-ID: <ZvSvkz@pres.example.com>
Content-Type: application/pidf+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
  <presence xmlns="urn:ietf:params:xml:ns:pidf"
            xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
            xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
            xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid-person"
            xmlns:dmp="urn:ietf:params:xml:ns:pidf:person"
            xmlns:pcp="urn:ietf:params:xml:ns:pidf:servcaps"
            entity="pres:user3_public1@home3.net">
```

```
    <tuple id="h7833hjkk.dsajfjdsaf">
      <status>
        <basic>closed</basic>
        <es:privacy><es:text/></es:privacy>
        <pcp:prescaps>
          <pcp:video>false</pcp:video>
          <pcp:audio>true</pcp:audio>
        </pcp:prescaps>
      </status>
      <et:class>sip</et:class>
      <contact priority="0.8">sip:user3_public1@home3.net</contact>
      <note xml:lang="en">Don't Disturb Please!</note>
      <note xml:lang="hu">Senki se merjen zavarni!</note>
      <timestamp>2003-08-27T11:48:59Z</timestamp>
    </tuple>


    <tuple id="sajdhdsahjh75vvcb774.78">
      <status>
        <basic>open</basic>
      </status>
      <et:class>supervisor</et:class>
      <et:relationship>supervisor</et:relationship>
      <contact priority="1.0">tel:+1-858-204-9141</contact>
      <note xml:lang="en">He's my supervisor</note>
      <timestamp>2003-08-27T11:48:59Z</timestamp>
    </tuple>

    <dmp:person>
      <ci:homepage>http://example.com/~user3</ci:homepage>
      <ci:card>http://example.com/~user3/card.vcd</ci:card>
      <et:class>presentity</et:class>
      <dmp:status>
        <ep:activities><ep:vacation/></ep:activities>
        <ep:place-type until="2003-09-10T17:30:00Z">ship</ep:place-type>
      </dmp:status>
    </dmp:person>

  </presence>

--50UBfW7LSCVLtggUPe5z--
```

P-Charging-Vector: The RLS populates the icid parameter with a globally unique value and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

Content-Type: Set to the value of the Accept: header received in the SUBSCRIBE request.

The message body in the NOTIFY request that carries the presence information of the presentity is formed as indicated in draft-ietf-simple-event-list-054 [22], draft-ietf-simple-presence-data-model-01 [x], draft-ietf-simple-rpid-043 [26], draft-ietf-simple-cipid-031 [32] and draft-ietf-simple-prescaps-ext-020 [25].

20. **NOTIFY request (S-CSCF to P-CSCF) - see example in table A.3.3.2-20**

The S-CSCF#1 forwards the NOTIFY request to the P-CSCF.

Table A.3.3.2-20: NOTIFY request (S-CSCF to P-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home2.net;branch=z9hG4bK240f34.1
Max-Forwards: 69
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:pcscf1.visited1.net;lr>
Record-Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Require:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

P-Charging-Vector: The S-CSCF stores the terminating Inter Operator Identifier (IOI) parameter received.

P-Charging-Function-Addresses: The S-CSCF populates the P-Charging-Function-Addresses header field to be passed to the P-CSCF.

21. **NOTIFY request (P-CSCF to UE) - see example in table A.3.3.2-21**

The P-CSCF forwards the NOTIFY request to the watcher application in the UE.

Table A.3.3.2-21: NOTIFY request (P-CSCF to UE)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1SIP/2.0/UDP rls.home2.net;branch=z9hG4bK240f34.1
Max-Forwards: 68
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.home1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Require:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

22. **200 (OK) response (UE to P-CSCF) - see example in table A.3.3.2-22**

The UE acknowledges the NOTIFY request with a 200 (OK) to the P-CSCF.

Table A.3.3.2-22: 200 (OK) response (UE to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1SIP/2.0/UDP rls.home2.net;branch=z9hG4bK240f34.1
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

23. **200 (OK) response (P-CSCF to S-CSCF) – see example in table A.3.3.2-23**

The P-CSCF forwards the 200 (OK) response to the S-CSCF#1.

Table A.3.3.2-23: 200 (OK) response (P-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home2.net;branch=z9hG4bK240f34.1
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024", orig-ioi=hom1.net,
      term-ioi=visited1.net
P-Access-Network-Info:
From:
To:
Call-ID:
CSeq:
Content-Length:
```

24. **200 (OK) response (S-CSCF to RLS) – see example in table A.3.3.2-24**

The S-CSCF#2 forwards the response to the RLS in the home network of the UE.

Table A.3.3.2-24: 200 (OK) response (S-CSCF to RLS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP rls.home2.net;branch=z9hG4bK240f34.1
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024", orig-ioi=hom1.net;
      term-ioi=home1.net
From:
To:
Call-ID:
CSeq:
Content-Length:
```

P-Charging-Vector: The S-CSCF inserts the originating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

A.3.4 RLS subscribing to presentities in different network

A.3.4.1 Successful subscription

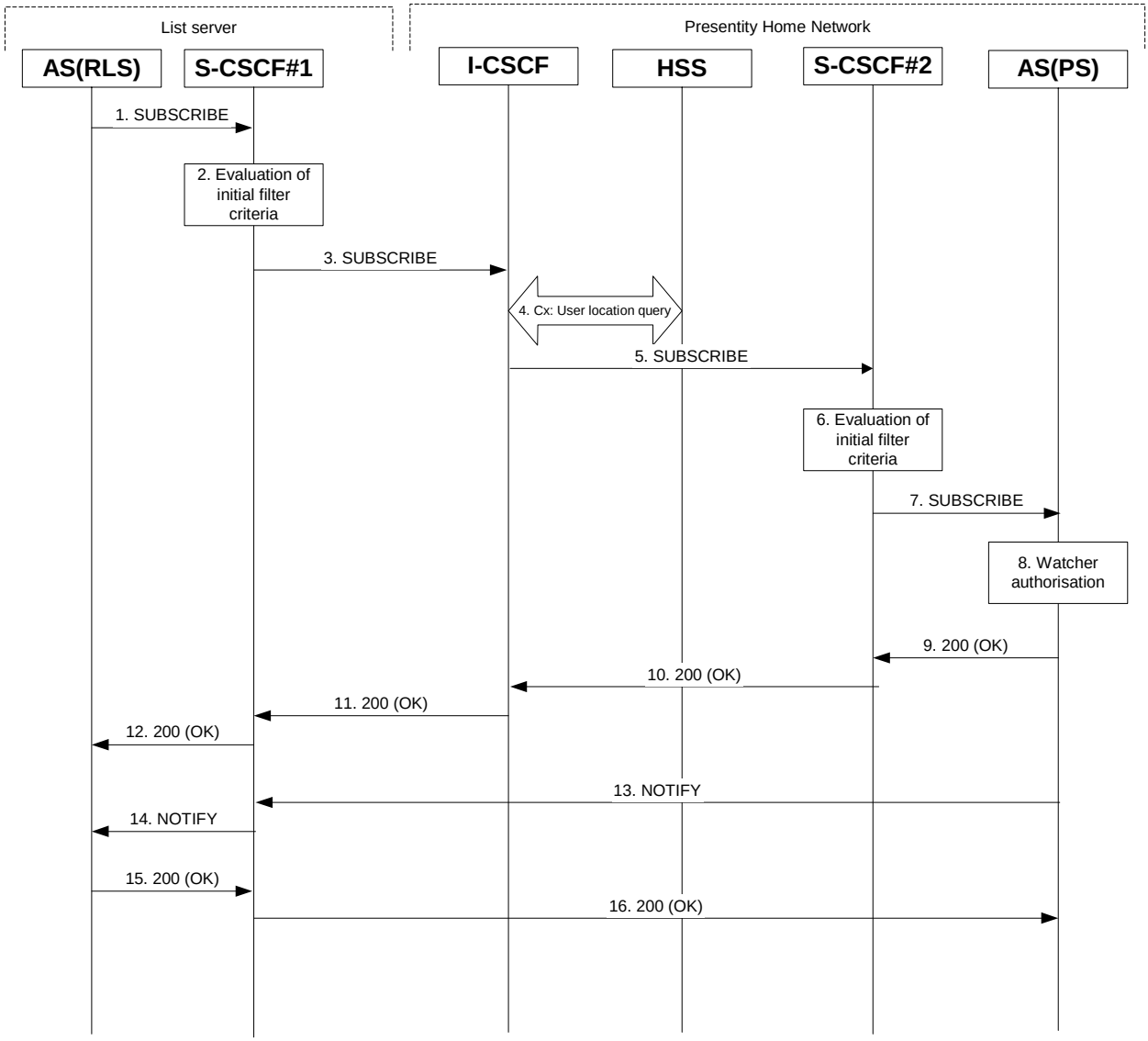


Figure A.3.4.1-1 RLS subscribing to presentities in different network

Figure A.3.4.1-1 shows the RLS subscribing to presence event notification about a presentity. The presentity is in a different IM CN subsystem. The details of the signalling flows are as follows:

1. **SUBSCRIBE request (RLS to S-CSCF) – see example in table A.3.4.1-1**

   The RLS resolves the watcher's resource address (the address is received according to subclause A.3.3) and subscribes to presence event notification at all the presentities that are represented by the resource list SIP URI. The home network of these presentities can be different or in the same network, as the RLS. In this example only a single subscription is shown where the home network of the presentity is another network. Subscriptions to other presentities follow a similar procedure. To initiate a subscription, the RLS generates a SUBSCRIBE request containing the "presence" event that it wishes to be notified of, together with an indication of the length of time this periodic subscription should last. The RLS sends the SUBSCRIBE request to the S-CSCF of "sip:user1_public1@home1.net" (S-CSCF#1). The address of S-CSCF#1 is either remembered from previous transactions (when "sip:user1_public1@home1.net" has subscribed for the resource list) or queried by the RLS using the Sh interface.

Table A.3.4.1-1 SUBSCRIBE request (RLS to S-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bKehuefdam
Max-Forwards: 70
Route: <sip:scscf1.home1.net;lr>
P-Asserted-Identity: <sip:user1_public1@home1.net>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
From: <sip:user1_public1@home1.net>;tag=31415
To: <sip:user2_public1@home2.net>
Call-ID: q987a9a87g087abgf7qyg7ag
CSeq: 123 SUBSCRIBE
Event: presence
Expires: 7200
Accept: application/pidf+xml
Contact: <sip:rls.home1.net>
Content-Length: 0
```

Request-URI: Public user identity whose events the RLS subscribes to.

P-Charging-Vector: The RLS populates the icid parameter with a new globally unique value and populates the originating Inter Operator Identifier (IOI) parameter with the identifier of its own network of RLS.

P-Charging-Function-Addresses: The RLS populates the P-Charging-Function-Addresses header field to be passed to the S-CSCF.

To: Same as the Request-URI.

Event: This field is populated with the value "presence" to specify the use of the presence package.

Accept: This field is populated with the value "application/pidf+xml".

2. **Evaluation of initial filter criteria**

   S-CSCF#1 validates the service profile of this subscriber and evaluates the initial filter criteria. For this example, assume no application server involvement.

3. **SUBSCRIBE request (S-CSCF to I-CSCF) – see example in table A.3.4.1-3**

   S-CSCF#1 performs an analysis of the destination address, and determines the network operator to whom the destination subscriber belongs. S-CSCF#1 forwards the request to the I-CSCF.

Table A.3.4.1-3 SUBSCRIBE request (S-CSCF to I-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bKehuehjgt, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bKehuefdam
Max-Forwards: 69
Record-Route: <sip:orig@scscf1.home1.net;lr>
P-Asserted-Identity:
P-Charging-Vector:
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter received.

4. **Cx: User Location Query procedure**

The I-CSCF sends a query to the HSS to find out the S-CSCF of the presentity. The HSS responds with the address of the current S-CSCF for the presentity.

For detailed message flows see 3GPP TS 29.228 [10].

Table A.3.4.1-4a provides the parameters in the SIP SUBSCRIBE request (flow 3), which are sent to the HSS.

Table A.3.4.1-4a: Cx: User registration status query procedure (I-CSCF to HSS)

Message source and destination	Cx: Information element name	Information source in SIP SUBSCRIBE	Description
I-CSCF to HSS	User Public Identity	Request-URI	This information element indicates the public user identity

Table A.3.4.1-4b provides the parameters sent from the HSS that need to be mapped to SIP SUBSCRIBE request (flow 5) and sent to the S-CSCF.

Table A.3.4.1-4b: Cx: User registration status query procedure (HSS to I-CSCF)

Message source and destination	Cx: Information element name	Mapping to SIP header in SIP SUBSCRIBE	Description
HSS to I-CSCF	S-CSCF name	Route header field	This information indicates the serving CSCF's name of that user

5. **SUBSCRIBE request (I-CSCF to S-CSCF) - see example in table A.3.4.1-5**

The I-CSCF forwards the SUBSCRIBE request to the S-CSCF#2 that will handle the termination.

Table A.3.4.1-5: SUBSCRIBE request (I-CSCF to S-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP icscf2_s.home2.net;branch=z9hG4bKj5hgrt2o, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bKehuehjgt, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bKehuefdam
Max-Forwards: 68
P-Asserted-Identity:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"; orig-ioi=home1.net;
Route: <sip:scscf2.home2.net;lr>
Record-Route:
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

6. **Evaluation of initial filter criteria**

S-CSCF#2 validates the service profile of this subscriber and evaluates the initial filter criteria. For sip:user2_public1@home2.net the S-CSCF has Termination initial Filter Criteria with Service Points of Interest of Method = SUBSCRIBE AND Event = "presence"that informs the S-CSCF to route the SUBSCRIBE request to the Application Server ps.home2.net. The S-CSCF#2 has preconfigured information not to record route for this request.

7. **SUBSCRIBE request (S-CSCF to PS) - see example in table A.3.4.1-7**

The S-CSCF#2 forwards the SUBSCRIBE request to the PS.

Table A.3.4.1-7 SUBSCRIBE request (S-CSCF to PS)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP scscf2.home2.net;branch=z9hG4bK344a65.1, SIP/2.0/UDP
      icscf2_s.home2.net;branch=z9hG4bKj5hgrt2o, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bKehuehjgt, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bKehuefdam
Max-Forwards: 67
P-Asserted-Identity:
P-Charging-Vector:
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:ps.home2.net;lr>, <sip:scscf2.home2.net;lr>
Record-Route: <sip:orig@scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF populates the P-Charging-Function-Addresses header field to be passed to the PS.

8. **Authorization of watcher**

The PS performs the necessary authorization checks on the originator to ensure it is allowed to watch the presentity. In this example all privacy conditions are met, so the PS sends a 200 (OK) response to the S-CSCF.

In the case where the privacy/authorization checks failed, then a necessary 2xx or 4xx response would be sent to the S-CSCF. The selection of the correct response code depends on the presentity's subscription authorization policy document.

9. **200 (OK) response (PS to S-CSCF) - see example in table A.3.4.1-9**

The PS sends the response to S-CSCF#2.

Table A.3.4.1-9: 200 (OK) response (PS to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf2.home2.net;branch=z9hG4bK344a65.1, SIP/2.0/UDP
      icscf2_s.home2.net;branch=z9hG4bKj5hgrt2o, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bKehuehjgt, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bKehuefdam
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"; orig-
      ioi=home2.net:term-ioi=home2.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Record-Route:
From:
To: <sip:user2_public1@home2.net>;tag=151170
Call-ID:
CSeq:
Expires:
Contact: <sip:ps.home2.net;lr>
Content-Length: 0
```

P-Charging-Vector: The PS stores the originating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The PS stores the P-Charging-Function-Addresses header field and passes this header to the S-CSCF.

10. **200 (OK) response (S-CSCF to I-CSCF) - see example in table A.3.4.1-10**

S-CSCF#2 forwards the response to the I-CSCF.

Table A.3.4.1-10: 200 (OK) response (S-CSCF to I-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP icscf2_s.home2.net;branch=z9hG4bKj5hgrt2o, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bKehuehjgt, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bKehuefdam
P-Charging-Vector:
P-Charging-Function-Addresses:
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the terminating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the I-CSCF.

11. **200 (OK) response (I-CSCF to S-CSCF) - see example in table A.3.4.1-11**

The I-CSCF forwards the response to S-CSCF#1.

Table A.3.4.1-11: 200 (OK) response (I-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bKehuehjgt, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bKehuefdam
P-Charging-Vector:
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

12. **200 (OK) response (S-CSCF to RLS) - see example in table A.3.4.1-12**

S-CSCF#1 forwards the response to the RLS.

Table A.3.4.1-12: 200 (OK) response (S-CSCF to RLS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bKehuefdam
P-Charging-Vector:
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the terminating Inter Operator Identifier (IOI) parameter received and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the RLS.

13. **NOTIFY request (PS to S-CSCF) - see example in table A.3.4.1-13**

As soon as the PS sends a 200 (OK) response to accept the subscription, it sends a NOTIFY request with the current state of the presentity's presence information~~tuples~~ that the watcher has subscribed and been authorized to. The NOTIFY request is sent to S-CSCF#1. ~~Further notification sent by the PS may either contain the complete set of presence information, or only those presence tuples that have changed since the last notification.~~

Table A.3.4.1-13: NOTIFY request (PS to S-CSCF)

```
NOTIFY sip:rls.home1.net SIP/2.0
Via: SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 70
Route: <sip:scscf1.home1.net;lr>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"; orig-ioi=home2.net
From: <sip:user1_public1@home2.net>;tag=151170
To: <sip:rls.home1.net>;tag=31415
Call-ID: q987a9a87g087abgf7qyg7ag
CSeq: 42 NOTIFY
Subscription-State:active;expires=7200
Event: presence
Contact: <sip:ps.home2.net>
Content-Type: application/pidf+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
   <presence xmlns="urn:ietf:params:xml:ns:pidf"
             xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
             xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
             xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid-person"
             xmlns:dmp="urn:ietf:params:xml:ns:pidf:person"
             xmlns:pcp="urn:ietf:params:xml:ns:pidf:servcaps"
             xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
        entity="pres:user2_public1@home2.net">

     <tuple id="a8098a.672364762364">
       <status>
         <basic>open</basic>
         <es:privacy><text/></es:privacy>
         <es:status-icon>http://example.com/~user2/icon.gif</es:status-icon>
       </status>
       <et:class>sip</et:class>
       <pcp:video>false</pcp:video>
       <pcp:audio>true</pcp:audio>
       <contact priority="0.8">sip:user2_public1@home2.net</contact>
       <note xml:lang="en">Don't Disturb Please!</note>
       <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
       <timestamp>2003-08-27T11:49:29Z</timestamp>
     </tuple>


     <tuple id="jklhgf9788934774.78">
       <status>
         <basic>open</basic>
       </status>
       <et:class>assistant</et:class>
       <et:relationship>assistant</et:relationship>
       <contact priority="1.0">tel:+1-212-555-2222</contact>
       <note xml:lang="en">She's my secretary</note>
       <timestamp>2003-08-27T11:49:29Z</timestamp>
     </tuple>

     <dmp:person>
       <ep:class>presentity</ep:class>
       <ci:homepage>http://example.com/~user2</ci:homepage>
       <ci:card>http://example.com/~user2/card.vcd</ci:card>
       <dmp:status>
         <ep:activities><ep:meeting/></ep:activities>
         <ep:place-type until="2003-08-27T17:30:00Z">office</ep:place-type>
       </dmp:status>
     </dmp:person>

   </presence>
<?xml version="1.0" encoding="UTF-8"?>
   <presence xmlns="urn:ietf:params:xml:ns:pidf"
             xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
```

```
                xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
                xmlns:pcp="urn:ietf:params:xml:ns:simple-prescaps-ext"
                xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
          entity="pres:user2_public1@home2.net ">

    <tuple id="a8098a.672364762364">
        <status>
          <basic>open</basic>
          <es:activity>meeting</es:activity>
          <es:placetype until="2003-08-27T17:30:00Z">office</es:placetype>
          <es:privacy>private</es:privacy>
          <es:idle since="2003-08-27T10:43:00Z"/>
          <pcp:prescaps>
            <pcp:video negated="false"></pcp:video>
            <pcp:mobility>mobile</pcp:mobility>
            <pcp:audio negated="true"></pcp:audio>
          </pcp:prescaps>
        </status>
        <et:class>sip</et:class>
        <et:contact-type>service</et:contact-type>
        <contact priority="0.8">im:user2_public1@home2.net</contact>
        <note xml:lang="en">Don't Disturb Please!</note>
        <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
        <timestamp>2003-08-27T11:49:29Z</timestamp>
    </tuple>

    <tuple id="sfddsj74.78">
        <status>
          <basic>open</basic>
        </status>
        <et:class>presentity</et:class>
        <et:contact-type>presentity</et:contact-type>
        <ci:homepage>http://example.com/~user2</ci:homepage>
        <ci:icon>http://example.com/~user2/icon.gif</ci:icon>
        <ci:card>http://example.com/~user2/card.vcd</ci:card>
        <note xml:lang="en">I'm in a boring meeting</note>
        <note xml:lang="en">I'll be in Tokyo next week</note>
        <timestamp>2004-10-10T12:00:30Z</timestamp>
    </tuple>

    <tuple id="jklhgf9788934774.78">
        <status>
          <basic>open</basic>
        </status>
        <et:class>assistant</et:class>
        <et:contact-type>presentity</et:contact-type>
        <et:relationship>assistant</et:relationship>
        <contact priority="1.0">tel:+1-212-555-2222</contact>
        <note xml:lang="en">She's my secretary</note>
        <timestamp>2003-08-27T11:49:29Z</timestamp>
    </tuple>

  </presence>
```

P-Charging-Vector: The PS populates the icid parameter with a globally unique value and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

Content-Type: Set to the value of the Accept header received in the SUBSCRIBE request or "application/pidf+xml".

The message body in the NOTIFY request that carries the presentitysubscriber's presenceregistration state is formed as indicated in draft-ietf-impp-cpim-pidf-08RFC 3863 [21], draft-ietf-simple-presence-data-model-01 [x], draft-ietf-simple-rpid-043 [26], draft-ietf-simple-cipid-031 [32] and draft-ietf-simple-prescaps-ext-020 [25].

14. **NOTIFY request (S-CSCF to RLS) - see example in table A.3.4.1-14**

The S-CSCF#1 forwards the NOTIFY request to the RLS.

Table A.3.4.1-14: NOTIFY request (S-CSCF to RLS)

```
NOTIFY sip:rls.home1.net SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bKehuehjgt, SIP/2.0/UDP
      ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 69
P-Charging-Vector:
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Record-Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF populates the P-Charging-Function-Addresses header field to be passed to the RLS.

15. **200 (OK) response (RLS to S-CSCF) – see example in table A.3.4.1-15**

The RLS generates a 200 (OK) response to the NOTIFY request.

Table A.3.4.1-15: 200 (OK) response (RLS to S-CSCF)

```
SIP/2.0 200 OK
Via:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"; orig-
      ioi=home1.net:term-ioi=home1.net
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

P-Charging-Vector: The RLS stores the originating Inter Operator Identifier (IOI) parameter and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

16. **200 (OK) response (S-CSCF to PS) – see example in table A.3.4.1-16**

The S-CSCF#1 forwards the 200 (OK) response to the PS.

Table A.3.4.1-16: 200 (OK) response (S-CSCF to PS)

```
SIP/2.0 200 OK
P-Charging-Vector:
Via: SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

P-Charging-Vector: The S-CSCF stores the terminating Inter Operator Identifier (IOI) parameter and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

A.3.5 Network based watcher subscribing on behalf of IMS watcher to IMS presentities

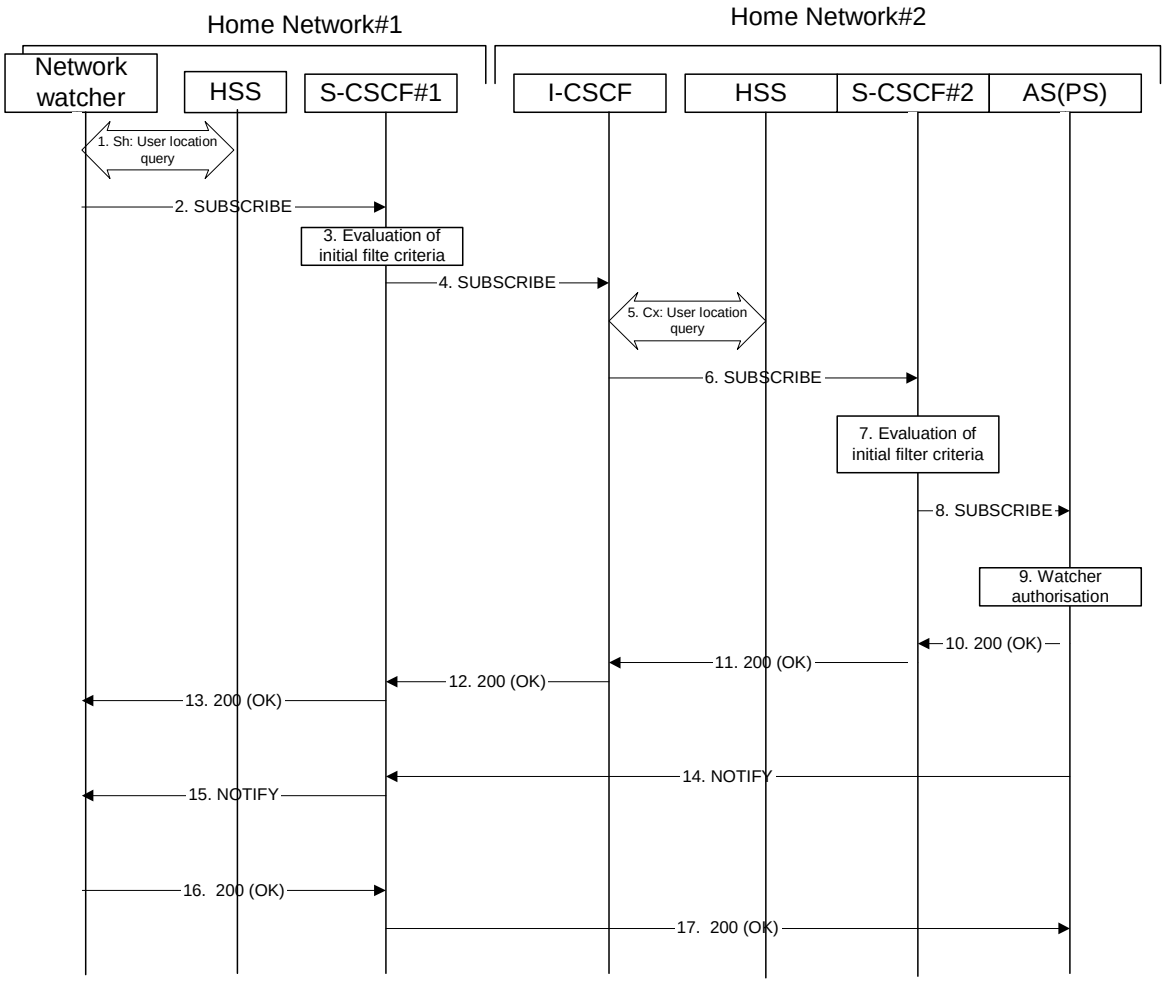


Figure A.3.5-1: Network based watcher subscribing on behalf of IMS watcher for presence information of IMS presentities

Figure A.3.5-1 shows a trusted network based watcher subscribing on behalf of an IMS watcher to presence event notification about an IMS based presentity. The presentity is in a different IM CN subsystem than the network based watcher and the signalling flow assumes that the IMS watcher on whose behalf the network based watcher subscribes is registered to the IMS network. The details of the signalling flows are as follows:

1. **Sh: User Location Query procedure**

   The network based watcher sends a query to the HSS to find out the S-CSCF of the user on whose behalf the subscription is initiated. The HSS responds with the address of the current S-CSCF for the originating subscriber.

2. **SUBSCRIBE request (Network based watcher to S-CSCF) - see example in table A.3.5-2**

The SUBSCRIBE request is constructed and forwarded to S-CSCF. The S-CSCF is inserted into the Route header of the SUBSCRIBE request.

Table A.3.5-2: SUBSCRIBE request (network watcher to S-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP watcher.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
Max-Forwards: 69
P-Asserted-Identity: <sip:user1_public1@home1.net>
Privacy: none
Route: <sip:scscf1.home1.net;lr;orig>
From: <sip:user1_public1@home1.net>;tag=31415
To: <sip:user2_public1@home2.net>
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 61 SUBSCRIBE
Event:PRESENCE
Expires: 7200
Accept: application/pidf+xml;q=0.3, application/pidf-partial+xml;q=1
Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>
Content-Length: 0
```

Request-URI: Public user identity of the user to whose events the subscriber subscribes to.

P-Asserted-Identity: The network based watcher inserts the public user identity of the watcher on whose behalf the subscription is made into the P-Asserted-Identity header field..

Route: The Route header is populated with the address of the S-CSCF obtained from the response to the user location query performed by the network based watcher on the Sh interface.

Event: This field is populated with the value "presence" to specify the use of the presence package.

Contact: The contact information of the network based watcher.

3. **Evaluation of initial filter criteria**

S-CSCF#1 validates the service profile of the subscriber identified in the P-Asserted-Identity header field and evaluates the initial filter criteria. For this example, assume no Application Server involvement.

4. **SUBSCRIBE request (S-CSCF to I-CSCF) - see example in table A.3.5-4**

S-CSCF#1 performs an analysis of the destination address, and determines the network operator to whom the destination subscriber belongs. Since the originating operator does not desire to keep their internal configuration hidden, S-CSCF#1 forwards the SUBSCRIBE request directly to the I-CSCF in the destination network.

Table A.3.5-4: SUBSCRIBE (S-CSCF to I-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      network.home1.net;branch=z9hG4bK240f34.1,
Max-Forwards: 68
P-Asserted-Identity: <sip:user1_public1@home1.net>
Privacy:
Record-Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

5. **Cx: User Location Query procedure**

The I-CSCF sends a query to the HSS to find out the S-CSCF of the called user. The HSS responds with the address of the current S-CSCF for the terminating subscriber.

For detailed message flows see 3GPP TS 29.228 [10].

Table A.3.5-5a provides the parameters in the SIP SUBSCRIBE request (flow 4), which are sent to the HSS.

Table A.3.5-5a: Cx: User registration status query procedure (I-CSCF to HSS)

Message source and destination	Cx: Information element name	Information source in SIP SUBSCRIBE	Description
I-CSCF to HSS	User Public Identity	Request-URI	This information element indicates the public user identity

Table A.3.5-5b provides the parameters sent from the HSS that need to be mapped to the SIP SUBSCRIBE request (flow 6) and sent to the S-CSCF.

Table A.3.5-5b: Cx: User registration status query procedure (HSS to I-CSCF)

Message source and destination	Cx: Information element name	Mapping to SIP header in SIP SUBSCRIBE	Description
HSS to I-CSCF	S-CSCF name	Route header field	This information indicates the serving CSCF's name of that user

6. **SUBSCRIBE request (I-CSCF to S-CSCF) - see example in table A.3.5-6**

The I-CSCF forwards the SUBSCRIBE request to the S-CSCF (S-CSCF#2) that will handle the termination.

Table A.3.5-6: SUBSCRIBE request (I-CSCF to S-CSCF)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      network.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 67
P-Asserted-Identity:
Privacy:
Route: <sip:scscf2.home2.net;lr>
Record-Route:
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

NOTE: The I-CSCF does not add itself to the Record-Route header, as it has no need to remain in the signalling path for the subsequent requests.

7. **Evaluation of initial filter criteria**

S-CSCF#2 validates the service profile of this subscriber and evaluates the initial filter criteria. For sip:user2_public1@home2.net S-CSCF#2 has termination initial filter criteria with service points of interest of Method = SUBSCRIBE and Event = "presence" that informs the S-CSCF to route the SUBSCRIBE request to the Application Server ps.home2.net. The S-CSCF#2 has preconfigured information not to record route this request.

8. **SUBSCRIBE request (S-CSCF to PS) - see example in table A.3.5-8**

The S-CSCF forwards the SUBSCRIBE request to the PS.

Table A.3.5-8: SUBSCRIBE request (S-CSCF to PS)

```
SUBSCRIBE sip:user2_public1@home2.net SIP/2.0
Via: SIP/2.0/UDP scscf2.home2.net;branch=z9hG4bK764z87.1, SIP/2.0/UDP
      icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      network.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 66
P-Asserted-Identity:
Privacy:
Route: <sip:ps.home2.net;lr>, <sip:scscf2.home2.net;lr>
Record-Route: <sip: scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Accept:
Contact:
Content-Length:
```

9. **Authorization of watcher**

The PS performs the necessary authorization checks on the watcher whose behalf the subscription is being made to ensure it is allowed to watch the presentity. In this example all privacy conditions are met, so the PS sends a 200 (OK) response to the S-CSCF.

In the case where the privacy/authorization checks failed, then a necessary 2xx or 4xx response would be sent to the S-CSCF. The selection of the correct response code depends on the presentity's authorization policy document.

10. **200 (OK) response (PS to S-CSCF) - see example in table A.3.5-10**

The PS sends the response to S-CSCF#2.

Table A.3.5-10: 200 (OK) response (PS to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf2.home2.net;branch=z9hG4bK764z87.1, SIP/2.0/UDP
icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      network.home1.net;branch=z9hG4bK240f34.1
Record-Route:
From:
To: <sip:user2_public1@home2.net>;tag=151170
Call-ID:
CSeq:
Expires:
Contact: <sip:ps.home2.net>
Content-Length:
```

11. **200 (OK) response (S-CSCF to I-CSCF) - see example in table A.3.5-11**

S-CSCF#2 forwards the response to I-CSCF#2.

Table A.3.5-11: 200 (OK) response (S-CSCF to I-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP icscf2_s.home2.net;branch=z9hG4bK871y12.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      network.home1.net;branch=z9hG4bK240f34.1
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

12. **200 (OK) response (I-CSCF to S-CSCF) - see example in table A.3.5-12**

I-CSCF#2 forwards the response to S-CSCF#1.

Table A.3.5-12: 200 (OK) response (I-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      network.home1.net;branch=z9hG4bK240f34.1
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

13. **200 (OK) response (S-CSCF to network watcher) - see example in table A.3.5-13**

S-CSCF#1 forwards the response to request originator.

Table A.3.5-13: 200 (OK) response (S-CSCF to network watcher)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP network.home1.net;branch=z9hG4bK240f34.1
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

14. **NOTIFY request (PS to S-CSCF) - see example in table A.3.5-14**

As soon as the PS sends a 200 (OK) response to accept the subscription, it sends a NOTIFY request with the current state of the presentity's presence information that the watcher has subscribed and been authorized to. The NOTIFY request is sent to S-CSCF#1. Based on the Accept header field of the SUBSCRIBE request, the PS decides to use partial notifications to provide further changes of presence information. Because the first notification always contains the full state the 'application/pidf ~~partial~~+xml' content type is used in the first NOTIFY request.

Table A.3.5-14: NOTIFY request (PS to S-CSCF)

```
NOTIFY sip: network.home1.net;branch=z9hG4bK240f34.1 SIP/2.0
Via: SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 70
Route: <sip:scscf1.home1.net;lr>
From: <sip:user2_public1@home2.net>;tag=151170
To: <sip:user1_public1@home1.net>;tag=31415
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 42 NOTIFY
Subscription-State: active; expires=7200
Event: presence
Contact: <sip:ps.home2.net>
Content-Type: application/pidf~~-partial~~+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
   <presence xmlns="urn:ietf:params:xml:ns:pidf"
             xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
             xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
             xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid-person"
             xmlns:dmp="urn:ietf:params:xml:ns:pidf:person"
             xmlns:pcp="urn:ietf:params:xml:ns:pidf:servcaps"
             xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
         entity="pres:user2_public1@home2.net">

     <tuple id="a8098a.672364762364">
       <status>
         <basic>open</basic>
         <es:privacy><text/></es:privacy>

         <es:status-icon>http://example.com/~user2/icon.gif</es:status-icon>
       </status>
       <et:class>sip</et:class>
       <pcp:video>false</pcp:video>
       <pcp:audio>true</pcp:audio>
       <contact priority="0.8">sip:user2_public1@home2.net</contact>
       <note xml:lang="en">Don't Disturb Please!</note>
       <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
       <timestamp>2003-08-27T11:49:29Z</timestamp>
     </tuple>


     <tuple id="jklhgf9788934774.78">
       <status>
         <basic>open</basic>
       </status>
       <et:class>assistant</et:class>
       <et:relationship>assistant</et:relationship>
       <contact priority="1.0">tel:+1-212-555-2222</contact>
       <note xml:lang="en">She's my secretary</note>
       <timestamp>2003-08-27T11:49:29Z</timestamp>
     </tuple>

     <dmp:person>
       <ep:class>presentity</ep:class>
       <ci:homepage>http://example.com/~user2</ci:homepage>
       <ci:card>http://example.com/~user2/card.vcd</ci:card>
       <dmp:status>
         <ep:activities><ep:meeting/></ep:activities>
         <ep:place-type until="2003-08-27T17:30:00Z">office</ep:place-type>
       </dmp:status>
     </dmp:person>

   </presence>
~~<?xml version="1.0" encoding="UTF-8"?>~~
~~   <pidf-part:presence xmlns="urn:ietf:params:xml:ns: pidf-partial"~~
```

```
            xmlns:pidf-part="urn:ietf:params:xml:ns:pidf-partial"
            xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
            xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
            xmlns:pcp="urn:ietf:params:xml:ns:simple-prescaps-ext"
            xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
                  entity="pres:user2_public1@home2.net" version="1" state="full">

    <tuple id="a8098a.672364762364">
      <status>
        <basic>open</basic>
        <es:activity>meeting</es:activity>
        <es:placetype until="2003-08-27T17:30:00Z">office</es:placetype>
        <es:privacy>private</es:privacy>
        <es:idle since="2003-08-27T10:43:00Z"/>
        <pcp:prescaps>
          <pcp:video negated="false"></pcp:video>
          <pcp:mobility>mobile</pcp:mobility>
          <pcp:audio negated="true"></pcp:audio>
        </pcp:prescaps>
      </status>
      <et:class>sip</et:class>
      <et:contact-type>service</et:contact-type>
      <contact priority="0.8">sip:user2_public1@home2.net</contact>
      <note xml:lang="en">Don't Disturb Please!</note>
      <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
      <timestamp>2003-08-27T11:49:29Z</timestamp>
    </tuple>

    <tuple id="sfddsj74.78">
      <status>
        <basic>open</basic>
      </status>
      <et:class>presentity</et:class>
      <et:contact-type>presentity</et:contact-type>
      <ci:homepage>http://example.com/~user2</ci:homepage>
      <ci:icon>http://example.com/~user2/icon.gif</ci:icon>
      <ci:card>http://example.com/~user2/card.vcd</ci:card>
      <note xml:lang="en">I'm in a boring meeting</note>
      <note xml:lang="en">I'll be in Tokyo next week</note>
      <timestamp>2004-10-10T12:00:30Z</timestamp>
    </tuple>

    <tuple id="jklhgf9788934774.78">
      <status>
        <basic>open</basic>
      </status>
      <et:class>assistant</et:class>
      <et:contact-type>presentity</et:contact-type>
      <et:relationship>assistant</et:relationship>
      <contact priority="1.0">tel:+1-212-555-2222</contact>
      <note xml:lang="en">She's my secretary</note>
      <timestamp>2003-08-27T11:49:29Z</timestamp>
    </tuple>

  </pidf-part:presence>
```

From: The tag of this field matches that of the To field in the received 200 (OK) response for the SUBSCRIBE request.

Content-Type: Set to the preferred value of the Accept header received in the SUBSCRIBE request.

The message body in the NOTIFY request that carries the presence information of the presentity is formed as indicated in ~~draft-ietf-impp-cpim-pidf-08~~RFC 3863 [21], draft-ietf-simple-rpid-04~~3~~ [26], draft-ietf-simple-cipid-03~~1~~ [32], draft-ietf-simple-prescaps-ext-02~~0~~ [25], draft-ietf-simple-presence-data-model-01 [x] and draft-ietf-simple-partial-notify-03~~1~~ [24]. ~~and draft-ietf-simple-partial-pidf-format-00 [38].~~

15. **NOTIFY request (S-CSCF to network watcher) - see example in table A.3.5-15**

The S-CSCF#1 forwards the NOTIFY request to the network watcher

Table A.3.5-15: NOTIFY request (S-CSCF to network watcher)

```
NOTIFY sip: network.home1.net;branch=z9hG4bK240f34.1SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 69
Privacy:
Record-Route: <sip:scscf1.home1.net;lr>

From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

16. **200 (OK) response (network watcher to S-CSCF) – see example in table A.3.5-16**

The network watcher forwards the 200 (OK) response to S-CSCF#1.

Table A.3.5-16: 200 (OK) response (network watcher to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      ps.home2.net;branch=z9hG4bK348923.1
P-Access-Network-Info:
From:
To:
Call-ID:
CSeq:
Content-Length:
```

17. **200 (OK) response (S-CSCF to PS) – see example in table A.3.5-17**

S-CSCF#2 forwards the 200 (OK) response to the PS.

Table A.3.5-17: 200 (OK) response (S-CSCF to PS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
From:
To:
Call-ID:
CSeq:
Content-Length:
```

A.3.6 Watcher subscribing to XCAP change, UE in visited network

A.3.6.1 Watcher subscribing to XCAP change in his resource list, UE in visited network - Successful subscription

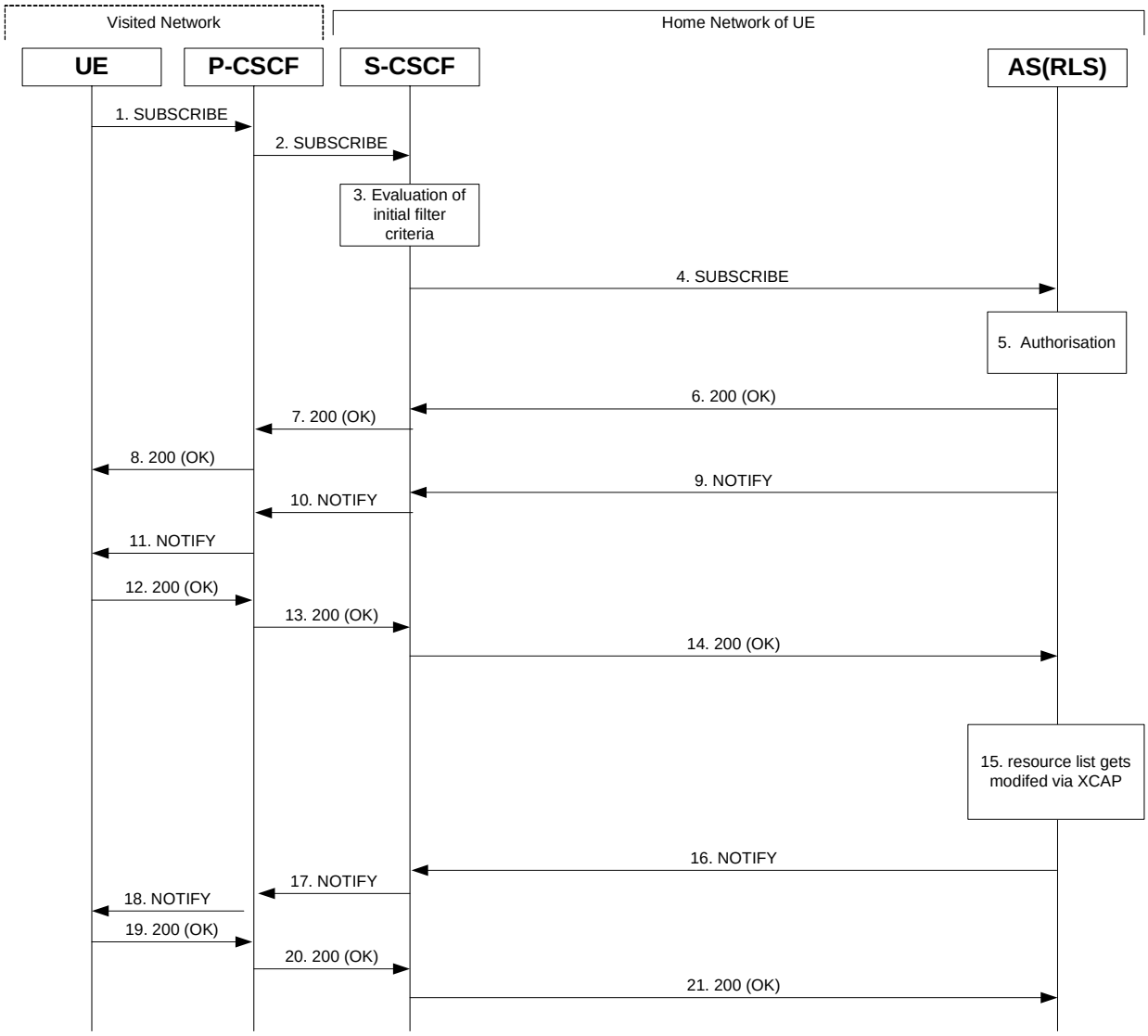


Figure A.3.6.1-1: Watcher subscribing to XCAP change in his resource list

Figure A.3.6.1-1 shows a watcher subscribing to XCAP change event notification. The details of the flows as follows:

1. **SUBSCRIBE request (UE to P-CSCF) – see example in table A.3.6.1-1**

   A watcher agent in a UE wishes to get notification when his resource list gets modified via XCAP. In order to initiate a subscription to XCAP changes in RLS, the UE generates a SUBSCRIBE request indicating support for "xcap-change", together with an indication of the length of time this periodic subscription should last.

Table A.3.6.1-1: SUBSCRIBE request (UE to P-CSCF)

```
SUBSCRIBE sip:user1_public1@home1.net SIP/2.0
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
Max-Forwards: 70
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
Route: <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>, <sip:orig@scscf1.home1.net;lr>
P-Preferred-Identity: <sip:user1_public1@home1.net>
Privacy: none
From: <sip:user1_public1@home1.net>;tag=31415
To: <sip:user1_public1@home1.net>
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 123 SUBSCRIBE
Require: sec-agree
Proxy-Require: sec-agree
Security-Verify: ipsec-3gpp; q=0.1; alg=hmac-sha-1-96; spi-c=98765432; spi-s=87654321; port-
      c=8642; port-s=7531
Event: xcap-change;profile-type=application;app-id=resource-lists;document="users/user1"
Expires: 7200
Accept: application/xcap-diff+xmlapplication/xcap-change+xml
Contact: <sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp>
Content-Length: 0
```

Request-URI: The users own SIP URI to get notifications of changes on all lists owned by the user.

Event: This field is populated with the value "xcap-change-" to specify the use of the xcap change package.

Accept: This field is populated with the value application/xcap-diff+xml application/xcap-change+xml ' indicating that the UE supports the xcap-change MIME type.

To: Same as the Request-URI.

2. **SUBSCRIBE request (P-CSCF to S-CSCF) - see example in table A.3.6.1-2**

The P-CSCF looks up the serving network information for the public user identity that was stored during the registration procedure. The SUBSCRIBE request is forwarded to S-CSCF#1. A Route header is inserted into SUBSCRIBE request. The information for the Route header is taken from the service route determined during registration.

Table A.3.6.1-2: SUBSCRIBE request (P-CSCF to S-CSCF)

```
SUBSCRIBE sip:user1_public1@home1.net SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
P-Access-Network-Info:
Route: <sip:orig@scscf1.home1.net;lr>
Max-Forwards: 69
P-Asserted-Identity: <sip:user1_public1@home1.net>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"
Privacy:
Record-Route: <sip:pcscf1.visited1.net;lr>
Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Supported:
Expires:
Accept:
Contact:
Content-Length:
```

3. **Evaluation of initial filter criteria**

The S-CSCF validates the service profile of this subscriber and evaluates the initial filter criteria. For sip:user1_public1@home1.net the S-CSCF has originating initial Filter Criteria with Service Point Trigger of Method = SUBSCRIBE AND Event = "xcap-change" that informs the S-CSCF to route the SUBSCRIBE request to the Application Server sip:rls.home1.net.

4. **SUBSCRIBE request (S-CSCF to RLS) - see example in table A.3.6.1-4**

The S-CSCF forwards the SUBSCRIBE request to the RLS.

Table A.3.6.1-4 SUBSCRIBE request (S-CSCF to RLS)

```
SUBSCRIBE sip:user1_public1@home1.net SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK344a65.1, SIP/2.0/UDP
      pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
Max-Forwards: 68
P-Access-Network-Info:
P-Asserted-Identity: <sip:user1_public1@home1.net>, <tel:+1-212-555-1111>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Privacy:
Record-Route: <sip:orig@scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
Route: <sip:rls.home1.net;lr>, <sip:orig@scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Supported:
Expires:
Accept:
Contact:
Content-Length:
```

P-Charging-Vector: The S-CSCF populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF populates the P-Charging-Function-Addresses header field to be passed to the RLS.

5. **Authorization**

The RLS performs the necessary authorization checks on the originator to ensure that he/she is authorized to subscribe to xcap-change. In this example this condition has been met, so the RLS sends a 200 (OK) response to the S-CSCF.

6. **200 (OK) response (RLS to S-CSCF) - see example in table A.3.6.1-6**

The RLS sends the response to the S-CSCF.

Table A.3.6.1-6: 200 (OK) response (RLS to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK344a65.1, SIP/2.0/UDP
      pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"; orig-ioi=home1.net;
      term-ioi=home1.net
Record-Route:
From:
To: <sip:user1_public1@home1.net>;tag=151170
Call-ID:
CSeq:
Expires:
Contact:
Content-Length: 0
```

7. **200 (OK) response (S-CSCF to P-CSCF) - see example in table A.3.6.1-7**

The S-CSCF forwards the response to the P-CSCF.

Table A.3.6.1-7: 200 (OK) response (S-CSCF to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=z9hG4bK120f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=223551024"
Record-Route:
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

8. **200 (OK) response (P-CSCF to UE) - see example in table A.3.6.1-8**

The P-CSCF forwards the response to the watcher agent in the UE.

Table A.3.6.1-8: 200 (OK) response (P-CSCF to UE)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKehuefdam
Record-Route: <sip:orig@scscf1.home1.net;lr>, <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Expires:
Contact:
Content-Length:
```

9. **NOTIFY request (RLS to S-CSCF) – see example in table A.3.6.1-9**

The RLS generates a NOTIFY request including the xcap-change document as a result of the SUBSCRIBE request. As this is the initial NOTIFY it contains only ~~the URI and~~ the new-etag, previous-etag and document-selector elements.

Table A.3.6.1-9 NOTIFY request (RLS to S-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 70
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
From: <sip:user1_@home1.net>;tag=151170
To: <sip:user1_public1@home1.net>;tag=31415
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 89 NOTIFY
Subscription-State: active;expires=7200
Event: xcap-change
Contact: <sip:rls.home1.net>
Content-Type: application/xcap-diff+xmlapplication/xcap-change+xml;charset="UTF-8"
Content-Length:

<?xml version="1.0" encoding="UTF-8"?>

   <documents xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <document
         uri=" http://xcap.home1.net/services/resource-lists/users/user1/pf.xml"
         new-etag="asdnasd9asd8asd7"
      </document>
   </documents>

<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff"
      xcap-root="http://xcap.home1.net/services">

    <document doc-selector="resource-lists/users/user1/friends"
              new-etag="7hahsd" previous-etag="7hahsd"/>
    </document>
    <document doc-selector="resource-lists/users/user1/coworkers"
              new-etag="ffds66a" previous-etag="ffds66a">
    </document>

</xcap-diff>
```

10. **NOTIFY request (S-CSCF to P-CSCF) - see example in table A.3.6.1-10**

The S-CSCF forwards the NOTIFY request to the P-CSCF.

Table A.3.6.1-10: NOTIFY request (S-CSCF to P-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 69
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"
P-Charging-Function-Addresses:
Route: <sip:pcscf1.visited1.net;lr>
Record-Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Length:

(…)
```

11. **NOTIFY request (P-CSCF to UE) - see example in table A.3.6.1-11**

The P-CSCF forwards the NOTIFY request to the watcher application in the UE.

Table A.3.6.1-11: NOTIFY request (P-CSCF to UE)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 68
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Length:

(…)
```

12. **200 (OK) response (UE to P-CSCF) - see example in table A.3.6.1-12**

The UE acknowledges the NOTIFY request with a 200 (OK) response to the P-CSCF.

Table A.3.6.1-12: 200 (OK) response (UE to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

13. **200 (OK) response (P-CSCF to S-CSCF) – see example in table A.3.6.1-13**

The P-CSCF forwards the 200 (OK) response to the S-CSCF.

Table A.3.6.1-13: 200 (OK) response (P-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"
From:
To:
Call-ID:
CSeq:
Content-Length:
```

14. **200 (OK) response (S-CSCF to RLS) - see example in table A.3.6.1-14**

The S-CSCF#2 forwards the response to the RLS in the home network of the UE.

Table A.3.6.1-14: 200 (OK) response (S-CSCF to RLS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=323551024"; orig-ioi=home1.net;
      term-ioi=home1.net
From:
To:
Call-ID:
CSeq:
Content-Length:
```

15. **Resource List gets modified via XCAP**

    The resource list of user1 gets modified via XCAP procedures.

16. **NOTIFY request (RLS to S-CSCF) - see example in table A.3.6.1-16**

    In this example it is assumed that the RLS has received a XCAP request to delete user2_public@home1.net from the resource list of user1.

Table A.3.6.1-16 NOTIFY request (RLS to S-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 70
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
From: <sip:user1_public1@home1.net>;tag=151170
To: <sip:user1_public1.home1.net>;tag=31415
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 90 NOTIFY
Subscription-State: active;expires=5000
Event: xcap-change
Contact: <sip:rls.home1.net>
Content-Type: application/xcap-diff+xmlapplication/xcap-change+xml;charset="UTF-8"
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
   <documents xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <document
        uri="http://xcap.home1.net/services/resource-lists/users/user1/pf.xml "
        new-etag="asdnasd8asd7asd6"
        previous-etag="asdnasd9asd8asd7"
        hash="<hash value>">
       <change
          method="DELETE"
          uri="http://xcap.home1.net/services/resource-lists/users/user1/pf.xml?resource-
      lists/entry[@name="user2_public@home1.net"]/uri"/>
     </document>
   </documents>
<xcap-diff xmlns="urn:ietf:params:xml:ns:xcap-diff"
      xcap-root="http://xcap.home1.net/services">

    <document doc-selector="resource-lists/users/user1/coworkers"
              new-etag="aaaab" previous-etag="ffds66a">
      <add-element parent="resource-lists/list[@name=&quot;coworkers&quot;]" position="1">
          <![CDATA[<entry uri="sip:new-worker@example.com"/>]]>
      </add-element>
    </document>
</xcap-diff>
```

Content-Type: Set to application/xcap-diff~~change~~+xml.

The message body in the NOTIFY request contains information of the new-etag of the changed document, the change method and the element that was changed in accordance with draft-ietf-simple-xcap-package-02~~1~~ [39].

17. **NOTIFY request (S-CSCF to P-CSCF) - see example in table A.3.6.1-17**

The S-CSCF forwards the NOTIFY request to the P-CSCF.

Table A.3.6.1-17: NOTIFY request (S-CSCF to P-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 69
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"
P-Charging-Function-Addresses:
Route: <sip:pcscf1.visited1.net;lr>
Record-Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content Type:
Content-Length:

(…)
```

18. **NOTIFY request (P-CSCF to UE) – see example in table A.3.6.1-18**

The P-CSCF forwards the NOTIFY request to the watcher application in the UE.

Table A.3.6.1-18: NOTIFY request (P-CSCF to UE)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 68
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Require:
Content-Type:
Content-Length:

(…)
```

19. **200 (OK) response (UE to P-CSCF) – see example in table A.3.6.1-19**

The UE acknowledges the NOTIFY request with a 200 (OK) response to the P-CSCF.

Table A.3.6.1-19: 200 (OK) response (UE to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

20. **200 (OK) response (P-CSCF to S-CSCF) – see example in table A.3.6.1-20**

The P-CSCF forwards the 200 (OK) response to the S-CSCF.

Table A.3.6.1-20: 200 (OK) response (P-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"
From:
To:
Call-ID:
CSeq:
Content-Length:
```

21. **200 (OK) response (S-CSCF to RLS) – see example in table A.3.6.1-21**

The S-CSCF#2 forwards the response to the RLS in the home network of the UE.

Table A.3.6.1-21: 200 (OK) response (S-CSCF to RLS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=423551024"; orig-ioi=home1.net;
      term-ioi=home1.net
From:
To:
Call-ID:
CSeq:
Content-Length:
```

A.4 Signalling flows demonstrating how presentities update presence information

A.4.1 Introduction

This subclause covers the signalling flows that show how presentities update presence information in the PS.

A.4.2 Initial publication or modification of presence information by UE

A.4.2.1 Successful publication

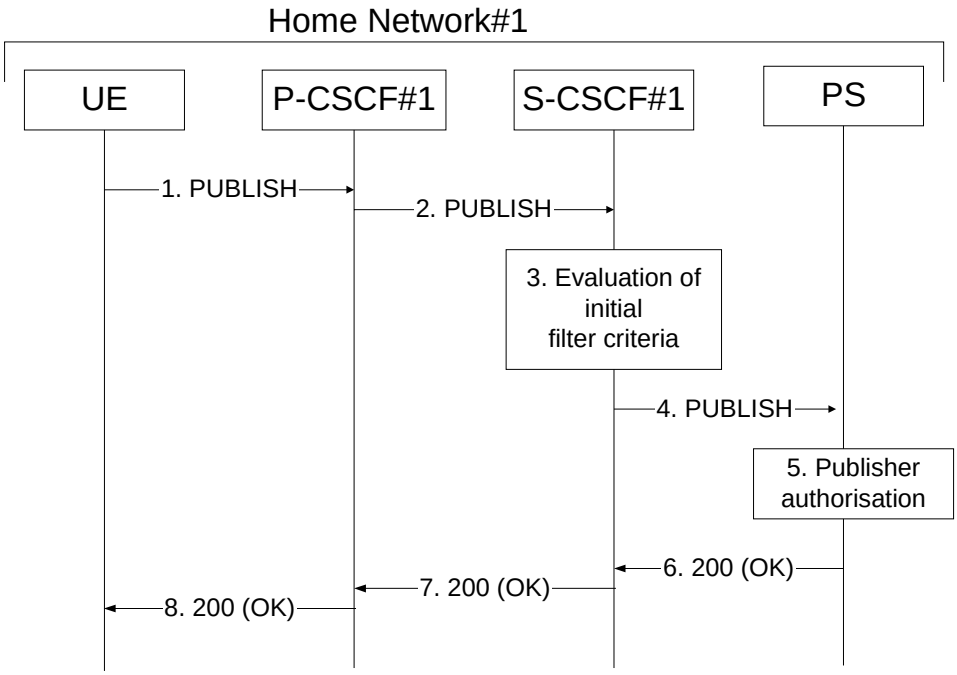


Figure A.4.2.1-1: UE publishing presence information

The UE may publish the partial presence information or the full presence information about a presentity to the PS.

In this example, it is assumed that the UE publishes the full presence information.

Figure A.4.2.1-1 shows a UE publishing or modifying already existing presence information about a presentity. The details of the signalling flows as follows:

1. **PUBLISH request (UE to P-CSCF) - see example in table A.4.2.1-1**

   A PUA in a UE wishes to publish presence information. To initiate the publication, the UE generates a PUBLISH request according to ~~draft-ietf-sip-publish-03~~RFC 3903 [23] containing the presence information that it wishes to publish.

   The message body in the PUBLISH request that carries the PUA~~publisher~~'s presence ~~update~~ state is formed as indicated in draft-ietf-simple-presence-data-model-01 [x], ~~draft-ietf-impp-cpim-pidf-08~~RFC 3863 [21], draft-ietf-simple-rpid-04~~3~~ [26], draft-ietf-simple-cipid-03~~1~~ [32], and draft-ietf-simple-prescaps-ext-02~~0~~ [25].

Table A.4.2.1-1: PUBLISH request (UE to P-CSCF)

```
PUBLISH sip:user1_public1@home1.net SIP/2.0
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
Max-Forwards: 70
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
Route: <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>, <sip:orig@scscf1.home1.net;lr>
P-Preferred-Identity: <sip:user1_public1@home1.net>
Privacy: none
From: <sip:user1_public1@home1.net>;tag=31415
To: <sip:user1_public1@home1.net>
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 61 PUBLISH
Require: sec-agree
Proxy-Require: sec-agree
Security-Verify: ipsec-3gpp; q=0.1; alg=hmac-sha-1-96; spi-c=98765432; spi-s=87654321; port-
      c=8642; port-s=7531
Event: presence
Expires: 7200
Content-Type: application/pidf+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
   <presence xmlns="urn:ietf:params:xml:ns:pidf"
             xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
             xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
             xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid-person"
             xmlns:dmp="urn:ietf:params:xml:ns:pidf:person"
             xmlns:pcp="urn:ietf:params:xml:ns:pidf:servcaps"
             xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
         entity="pres:user2_public1@home2.net">

     <tuple id="a8098a.672364762364">
       <status>
         <basic>open</basic>
         <es:privacy><text/></es:privacy>

         <es:status-icon>http://example.com/~user2/icon.gif</es:status-icon>
       </status>
       <et:class>sip</et:class>
       <pcp:video>false</pcp:video>
       <pcp:audio>true</pcp:audio>
       <contact priority="0.8">sip:user2_public1@home2.net</contact>
       <note xml:lang="en">Don't Disturb Please!</note>
       <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
       <timestamp>2003-08-27T11:49:29Z</timestamp>
     </tuple>


     <tuple id="jklhgf9788934774.78">
       <status>
         <basic>open</basic>
       </status>
       <et:class>assistant</et:class>
       <et:relationship>assistant</et:relationship>
       <contact priority="1.0">tel:+1-212-555-2222</contact>
       <note xml:lang="en">She's my secretary</note>
       <timestamp>2003-08-27T11:49:29Z</timestamp>
     </tuple>

     <dmp:person>
       <ep:class>presentity</ep:class>
```

```
        <ci:homepage>http://example.com/~user2</ci:homepage>
        <ci:card>http://example.com/~user2/card.vcd</ci:card>
        <dmp:status>
          <ep:activities><ep:meeting/></ep:activities>
          <ep:place-type until="2003-08-27T17:30:00Z">office</ep:place-type>
        </dmp:status>
      </dmp:person>

    </presence>
<?xml version="1.0" encoding="UTF-8"?>
    <presence xmlns="urn:ietf:params:xml:ns:pidf"
              xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
              xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
              xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
              xmlns:pcp="urn:ietf:params:xml:ns:simple-prescaps-ext"
              entity="pres:user2_public1@home2.net">

      <tuple id="a8098a.672364762364">
        <status>
          <basic>open</basic>
          <es:activity>meeting</es:activity>
          <es:placetype until="2003-08-27T17:30:00Z">office</es:placetype>
          <es:privacy>private</es:privacy>
          <es:idle since="2003-08-27T10:43:00Z"/>
          <pcp:prescaps>
            <pcp:video negated="false"></pcp:video>
            <pcp:mobility>mobile</pcp:mobility>
            <pcp:audio negated="true"></pcp:audio>
          </pcp:prescaps>
        </status>
        <et:class>sip</et:class>
        <et:contact-type>service</et:contact-type>
        <contact priority="0.8">sip:user2_public1@home2.net</contact>
        <note xml:lang="en">Don't Disturb Please!</note>
        <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
        <timestamp>2003-08-27T11:49:29Z</timestamp>
      </tuple>

      <tuple id="sfddsj74.78">
        <status>
          <basic>open</basic>
        </status>
        <et:class>presentity</et:class>
        <et:contact-type>presentity</et:contact-type>
        <ci:homepage>http://example.com/~user2</ci:homepage>
        <ci:icon>http://example.com/~user2/icon.gif</ci:icon>
        <ci:card>http://example.com/~user2/card.vcd</ci:card>
        <note xml:lang="en">I'm in a boring meeting</note>
        <note xml:lang="en">I'll be in Tokyo next week</note>
        <timestamp>2004-10-10T12:00:30Z</timestamp>
      </tuple>

      <tuple id="jklhgf9788934774.78">
        <status>
          <basic>open</basic>
        </status>
        <et:class>assistant</et:class>
        <et:contact-type>presentity</et:contact-type>
        <et:relationship>assistant</et:relationship>
        <contact priority="1.0">tel:+1-212-555-2222</contact>
        <note xml:lang="en">She's my secretary</note>
        <timestamp>2003-08-27T11:49:29Z</timestamp>
      </tuple>

    </presence>
```

Request-URI: Public user identity whose presence information the PUA intends to publish.

Event: This field is populated with the value "presence" to specify the use of the presence package.

To: Same as the Request-URI.

Content-Type: Set to the value 'application/pidf+xml'.

2. **PUBLISH request (P-CSCF to S-CSCF) - see example in table A.4.2.1-2**

P-CSCF looks up the serving network information for the public user identity that was stored during the registration procedure. The PUBLISH request is forwarded to the S-CSCF. A Route header is inserted into PUBLISH request. The information for the Route header is taken from the service route determined during registration.

Table A.4.2.1-2: PUBLISH request (P-CSCF to S-CSCF)

```
PUBLISH sip:user1_public1@home1.net SIP/2.0
Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Access-Network-Info:
Max-Forwards: 69
P-Asserted-Identity: <sip:user1_public1@home1.net>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"
Privacy:
Route: <sip:orig@scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Content-Type:
Content-Length:

(…)
```

3. **Evaluation of initial filter criteria**

S-CSCF validates the service profile of the publisher and evaluates the initial filter criteria. For user1_public1@home1.net S-CSCF#1 has originating initial Filter Criteria with Service Points of Interest of Method = PUBLISH AND Event = "presence" AND To = "sip:user1_public1@home1.net" that informs the S-CSCF to route the PUBLISH request to the PS ps.home1.net.

4. **PUBLISH (S-CSCF to PS) - see example in table A.4.2.1-4**

The S-CSCF#1 forwards the PUBLISH request to the PS.

Table A.4.2.1-4: PUBLISH request (S-CSCF to PS)

```
PUBLISH sip:user1_public1@home1.net SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Access-Network-Info:
Max-Forwards: 68
P-Asserted-Identity:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Privacy:
Route: <sip:ps.home1.net;lr>, <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
Expires:
Content-Type:
Content-Length:

(…)
```

P-Charging-Vector: The S-CSCF populates the icid parameter with a globally unique value and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF populates the P-Charging-Function-Addresses header field to be passed to the PS.

5. **Authorization of publisher**

The PS performs the necessary authorization checks on the originator to ensure it is allowed to publish the presentity's presence information. In this example all privacy conditions are met, so the PS sends a 200 (OK) response to the S-CSCF.

6. **200 (OK) response (PS to S-CSCF) - see example in table A.4.2.1-6**

The PS sends the response to S-CSCF.

Table A.4.2.1-6: 200 (OK) response (PS to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"; orig-
      ioi=home1.net:term-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
From:
To: <sip:user1_public1@home1.net>;tag=151170
Call-ID:
CSeq:
Expires: 7200
SIP-ETag: 123xy
Content-Length: 0
```

P-Charging-Vector: The PS stores the originating Inter Operator Identifier (IOI) parameter and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The PS stores the P-Charging-Function-Addresses header field and passes this header to the S-CSCF.

SIP-ETag: This field is populated with a locally unique entity-tag to associate further publications ~~refreshments~~ of this event state ~~segment~~

7. **200 (OK) response (S-CSCF to P-CSCF) - see example in table A.4.2.1-7**

S-CSCF forwards the response to P-CSCF.

Table A.4.2.1-7: 200 (OK) response (S-CSCF to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"
P-Charging-Function-Addresses:
From:
To:
Call-ID:
CSeq:
Expires:
SIP-ETag:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the terminating Inter Operator Identifier (IOI) parameter and removes the orig-ioi and the term-ioi parameters.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the P-CSCF.

8. **200 (OK) response (P-CSCF to UE) - see example in table A.4.2.1-6**

P-CSCF forwards the response to the PUA in the UE.

Table A.4.2.1-8: 200 (OK) response (P-CSCF to UE)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
From:
To:
Call-ID:
CSeq:
Expires:
SIP-ETag:
Content-Length:
```

A.4.3 Refreshing of presence information by UE

A.4.3.1 Successful refresh

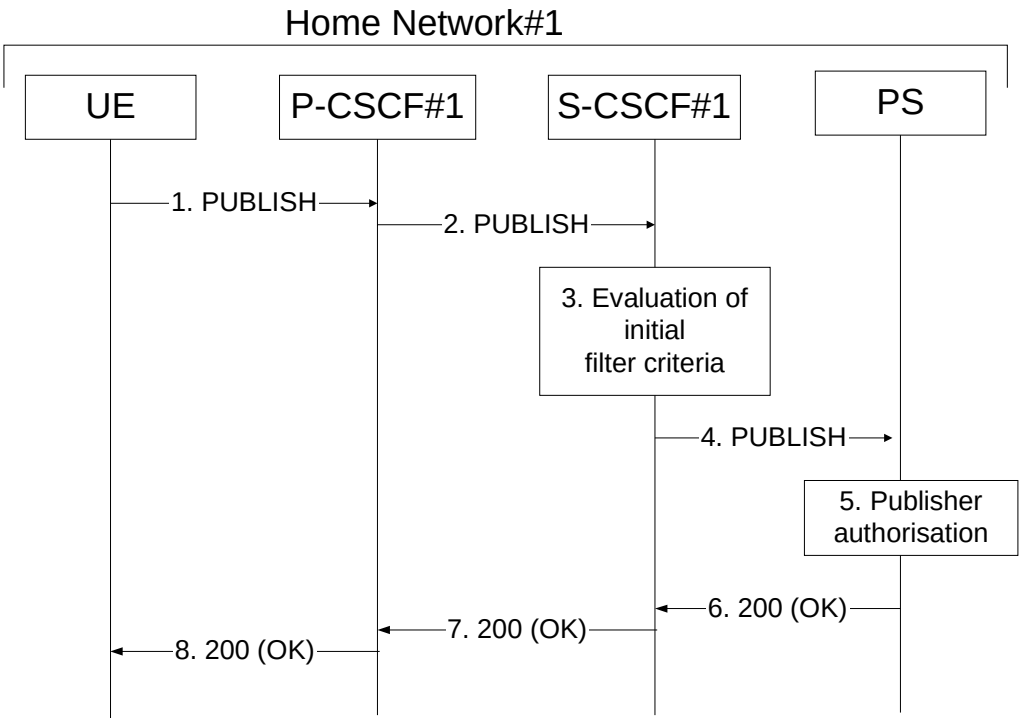


Figure A.4.3.1-1: UE updating presence information

Figure A.4.3.1-1 shows an UE refreshing the presence information about a presentity. The details of the signalling flows are as follows:

1. **PUBLISH request (UE to P-CSCF) – see example in table A.4.3.1-1**

   A PUA in a UE wishes to refresh already existing presence information. To initiate the publication, the UE generates a PUBLISH request according to ~~draft-ietf-sip-publish-03~~RFC 3903 [23].

Table A.4.3.1-1: PUBLISH request (UE to P-CSCF)

```
PUBLISH sip:user1_public1@home1.net SIP/2.0
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
Max-Forwards: 70
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
Route: <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>, <sip:orig@scscf1.home1.net;lr>
P-Preferred-Identity: <sip:user1_public1@home1.net>
Privacy: none
From: <sip:user1_public1@home1.net>;tag=31415
To: <sip:user1_public1@home1.net>
Call-ID: b89rjhnedlrfjflslj40a111
CSeq: 61 PUBLISH
Require: sec-agree
Proxy-Require: sec-agree
Security-Verify: ipsec-3gpp; q=0.1; alg=hmac-sha-1-96; spi-c=98765432; spi-s=87654321; port-
      c=8642; port-s=7531
Event: presence
SIP-If-Match: 123xy
Expires: 7200
Content-Length: 0
```

Request-URI: Public user identity whose presence information the PUA intends to publish.

Event: This field is populated with the value "presence" to specify the use of the presence package.

To: Same as the Request-URI.

SIP-If-Match: This field is populated with the entity-tag earlier provided by the PS in the SIP-ETag header field of the previous 200(OK) response and is used as a versioning precondition to the PUBLISH refresh.

2. **PUBLISH request (P-CSCF to S-CSCF) - see example in table A.4.3.1-2**

P-CSCF looks up the serving network information for the public user identity that was stored during the registration procedure. The PUBLISH request is forwarded to the S-CSCF. A Route header is inserted into PUBLISH request. The information for the Route header is taken from the service route determined during registration.

Table A.4.3.1-2: PUBLISH request (P-CSCF to S-CSCF)

```
PUBLISH sip:user1_public1@home1.net SIP/2.0
Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Access-Network-Info:
Max-Forwards: 69
P-Asserted-Identity: <sip:user1_public1@home1.net>
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"
Privacy:
Route: <sip:orig@scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
SIP-If-Match:
Expires:
Content-Length:
```

P-Charging-Vector: The P-CSCF populates the icid parameter with a globally unique value.

3. **Evaluation of initial filter criteria**

S-CSCF#1 validates the service profile of this publisher and evaluates the initial filter criteria. For user1_public1@home1.net the S-CSCF has originating initial Filter Criteria with Service Points of Interest of Method = PUBLISH AND Event = "presence" AND To = "sip:user1_public1@home1.net" that informs the S-CSCF to route the PUBLISH request to the PS ps.home1.net.

4. **PUBLISH (S-CSCF to PS) – see example in table A.4.3.1-4**

The S-CSCF forwards the PUBLISH request to the PS.

Table A.4.3.1-4: PUBLISH (S-CSCF to PS)

```
PUBLISH sip:user1_public1@home1.net SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Access-Network-Info:
Max-Forwards: 68
P-Asserted-Identity:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Privacy:
Route: <sip:ps.home1.net;lr>, <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Event:
SIP-If-Match:
Expires:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF populates the P-Charging-Function-Addresses header field to be passed to the PS.

5. **Authorization of publisher**

The PS performs the necessary authorization checks on the originator to ensure it is allowed to publish the presentity's presence information. In this example all privacy conditions are met, so the PS sends a 200 (OK) response to the S-CSCF.

6. **200 (OK) response (PS to S-CSCF) - see example in table A.4.3.1-6**

The PS sends the response to S-CSCF.

Table A.4.3.1-6: 200 (OK) response (PS to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP
      pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023555517"; orig-
      ioi=home1.net;term-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
From:
To: <sip:user1_public1@home1.net>;tag=151170
Call-ID:
CSeq:
Expires: 7200
SIP-ETag: 345abc
Content-Length: 0
```

P-Charging-Vector: The PS stores the originating Inter Operator Identifier (IOI) parameter and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The PS stores the P-Charging-Function-Addresses header field and passes this header to the S-CSCF.

SIP-ETag: This field is populated with a new locally unique entity-tag.

7. **200 (OK) response (S-CSCF to P-CSCF) - see example in table A.4.3.1-7**

S-CSCF#1 forwards the response to P-CSCF.

Table A.4.3.1-7: 200 (OK) response (S-CSCF to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.home1.net;branch=z9hG4bK240f34.1, SIP/2.0/UDP
      [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=023555517"
P-Charging-Function-Addresses:
From:
To:
Call-ID:
CSeq:
Expires:
SIP-ETag:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the terminating Inter Operator Identifier (IOI) parameter and removes the orig-ioi and the term-ioi parameters.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the P-CSCF.

8. **200 (OK) response (P-CSCF to UE) - see example in table A.4.3.1-6**

P-CSCF#1 forwards the response to the PUA in the UE.

Table A.4.3.1-8: 200 (OK) response (P-CSCF to UE)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP [5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp;branch=z9hG4bKnashds7
From:
To:
Call-ID:
CSeq:
Expires:
SIP-ETag:
Content-Length:
```

A.5 PS notifying watcher of updates to presence information

A.5.1 Introduction

This subclause covers the signalling flows that show how the PS notifies watchers of updates to presence information.

A.5.2 Watcher and presentity in the different networks, UE in the home network

A.5.2.1 Successful notification

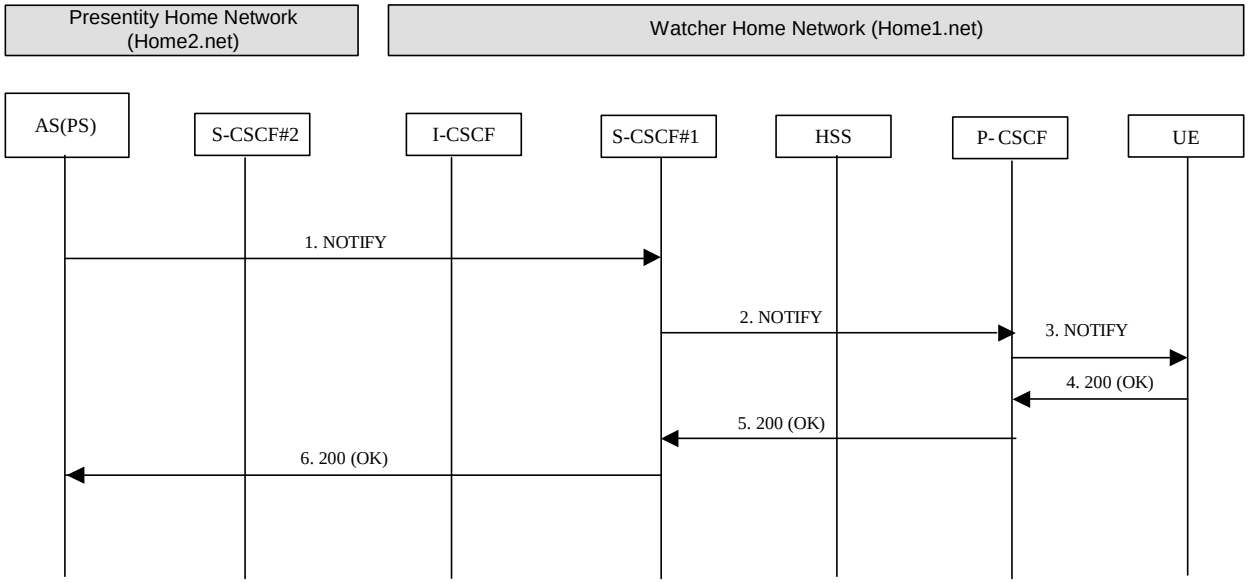


Figure A.5.2.1-1: Notification to watcher in the visited network

Figure A.5.2.1-1 shows how a watcher is notified of updates to a presentity's presence information. The signalling flow is applicable to the case where the watcher and presentity are in the same or in different IM CN subsystems.

1. **NOTIFY request (PS to S-CSCF) – see example in table A.5.2.1-1**

   The PS determines which authorized watchers are entitled to receive the updates of the presence information for this presentity. For each appropriate watcher, the PS sends a NOTIFY request that contains the updated state of presence information. The NOTIFY request may either contain the complete set of presence information, or only the information that has changed since the last notification. In this example, the watcher indicated preference for partial notification in the SUBSCRIBE request, so the NOTIFY request is formulated according to draft-ietf-simple-partial-notify-03~~1~~ [24] and draft-ietf-simple-partial-pidf-format-02~~0~~ [38] by including only the information that has changed since the last notification. (Note that the first NOTIFY request has contained the full state of the presence information.)

Table A.5.2.1-1: NOTIFY request (PS to S-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP ps.home2.net;branch=z9hG4bK240f34.1
Max-Forwards: 70
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=523551024"; orig-ioi=home2.net
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.home1.net;lr>
From: <sip:user2_public1@home2.net>;tag=151170
To: <sip:user1_public1@home1.net>;tag=31415
Call-ID: b89rjhnedlrfjflslj40a222
CSeq: 43 NOTIFY
Subscription-State: active;expires=5000
Event: presence
Contact: <sip:ps.home2.net>
Content-Type: application/pidf-diffpartial+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
   <pidf-diff xmlns="urn:ietf:params:xml:ns:pidf-diff"
             entity="pres:user2_public1@home2.net" version="1">

    <add parent="presence" sel="*[2]">
      <![CDATA[
```

```
      <tuple id="xfjsk">
       <status>
         <basic>open</basic>
         <es:status-icon>http://example.com/~user2/iconABC.gif</es:status-icon>
       </status>
       <et:class>voice</et:class>
       <contact priority="0.2">tel:40302020@home2.net</contact>
       <note xml:lang="en">This is a new tuple inserted as the 2nd tuple.</note>
       <timestamp>2004-11-01T11:49:29Z</timestamp>
      </tuple>
       ]] >
    </add>
    <replace sel="presence/tuple[@id="a8098a.672364762364"]/status/basic/text()">closed
    </replace>

    <remove sel="presence/tuple[@id="a8098a.672364762364"]/status/es:privacy"/>

    <remove sel="presence/tuple[@id="a8098a.672364762364"]/status/rs:activity"/>

    <replace sel="presence/tuple[@id="a8098a.672364762364"]/status/es:status-icon/text()">
    http://example.com/~user2/iconXYZ.gif</replace>

   </pidf-diff>

<pidf-part:presence xmlns="urn:ietf:params:xml:ns:pidf"
          xmlns:part="urn:ietf:params:xml:ns:pidf-partial"
          xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
          xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
          xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
          xmlns:pcp="urn:ietf:params:xml:ns:simple-prescaps-ext"
          entity="pres:user2_public1@home2.net" version="2" state="partial">

     <tuple id="a8098a.672364762364">
       <status>
         <basic>open</basic>
         <es:placetype until="2003-08-27T18:30:00Z">office</es:placetype>
         <es:privacy>public</es:privacy>
         <pcp:prescaps>
           <pcp:video negated="false"></pcp:video>
           <pcp:mobility>mobile</pcp:mobility>
           <pcp:audio negated="true"></pcp:audio>
         </pcp:prescaps>
       </status>
       <et:class>sip</et:class>
       <et:contact-type>service</et:contact-type>
       <contact priority="1.0">sip:user2_public1@home2.net</contact>
       <timestamp>2003-08-27T17:35:29Z</timestamp>
     </tuple>

     <tuple id="sfddsj74.78">
       <status>
         <basic>open</basic>
       </status>
       <et:class>presentity</et:class>
       <et:contact-type>presentity</et:contact-type>
       <ci:homepage>http://example.com/~user2</ci:homepage>
       <ci:icon>http://example.com/~user2/icon.gif</ci:icon>
       <ci:card>http://example.com/~user2/card.vcd</ci:card>
       <note xml:lang="en">I'm in a boring meeting</note>
       <note xml:lang="en">I'll be in Tokyo next week</note>
       <timestamp>2003-08-27T12:00:30Z</timestamp>
     </tuple>

     <pidf-part:removed>
       <pidf-part:t_id>jklhgf9788934774.78</pidf-part:t_id>
     </pidf-part:removed>

</pidf-part:presence>
```

P-Charging-Vector: The PS populates the icid parameter with a globally unique value and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

2. **NOTIFY request (S-CSCF to P-CSCF) - see example in table A.5.2.1-2**

The S-CSCF forwards the NOTIFY request to the P-CSCF.

Table A.5.2.1-2: NOTIFY request (S-CSCF to P-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      scscf2.home2.net;branch=z9hG4bK764z87.1, SIP/2.0/UDP ps.home2.net;branch=z9hG4bK240f34.1
Max-Forwards: 69
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=523551024"
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:pcscf1.home1.net;lr>
Record-Route: <sip:scscf2.home2.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter.

P-Charging-Function-Addresses: The S-CSCF populates the P-Charging-Function-Addresses header field to be passed to the P-CSCF.

3. **NOTIFY request (P-CSCF to UE) - see example in table A.5.2.1-3**

The P-CSCF forwards the NOTIFY request to the UE.

Table A.5.2.1-3: NOTIFY request (P-CSCF to UE)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP pcscf1.home1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK351g45.1, SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 68
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.home1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

4. **200 (OK) response (UE to P-CSCF) - see example in table A.5.2.1-4**

The UE acknowledges the NOTIFY request with a 200 (OK) response to the P-CSCF.

Table A.5.2.1-4: 200 (OK) response (UE to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.home1.net;branch=240f34.1, SIP/2.0/UDP
      scscf2.home2.net;branch=z9hG4bK764z87.1, SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

5. **200 (OK) response (P-CSCF to S-CSCF) - see example in table A.5.2.1-5**

The P-CSCF forwards the 200 (OK) response to the S-CSCF.

Table A.5.2.1-5: 200 (OK) response (P-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      ps.home2.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=523551024"
From:
To:
Call-ID:
CSeq:
Content-Length:
```

6. **200 (OK) response (S-CSCF to PS) - see example in table A.5.2.1-6**

The S-CSCF forwards the 200 (OK) response to the PS.

Table A.5.2.1-6: 200 (OK) response (S-CSCF to PS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP SIP/2.0/UDP ps.home2.net;branch=z9hG4bK240f34.1
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=523551024"; orig-
      ioi=home1.net:term-ioi=home2.net
From:
To:
Call-ID:
CSeq:
Content-Length:
```

P-Charging-Vector: The S-CSCF inserts the originating Inter Operator Identifier (IOI) parameter and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

A.5.3 Notification to resource list in a different network and notification to watcher in the visited network

A.5.3.1 Successful notification

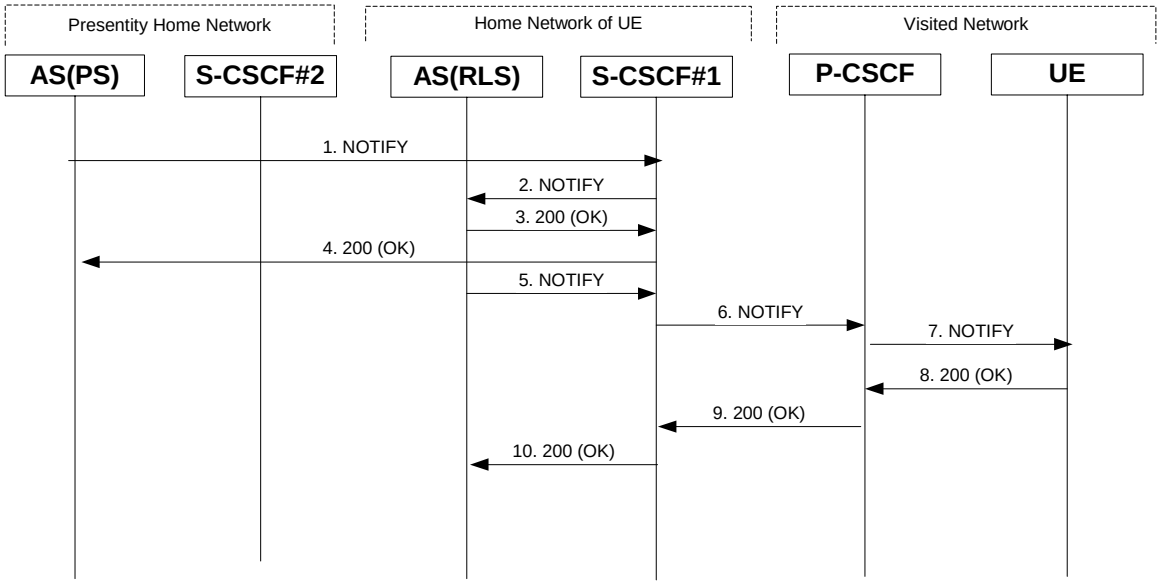


Figure A.5.3.1-1: Notification to resource list in a different network and notification to watcher in the visited network

Figure A.5.3.1-1 shows the PS providing presence event notification about a presentity to a RLS in a different network. This notification triggers the RLS to provide presence event notification to the watcher. The details of the signalling flows are as follows:

1. **NOTIFY request (PS to S-CSCF) - see example in table A.5.3.1-1**

   The PS determines which authorized watchers are entitled to receive presence information. For each appropriate watcher, the PS sends a NOTIFY request that contains the updated state of presence information. In this example the notification is only sent to the RLS.

   The NOTIFY request may either contain the complete set of presence information, or only those presence information~~tuples~~ that have changed since the last notification. For this example, the complete set of presence information is sent.

Table A.5.3.1-1: NOTIFY request (PS to S-CSCF)

```
NOTIFY sip:rls.home1.net SIP/2.0
Via: SIP/2.0/UDP ps.home2.net;branch=z9hG4bK240f34.1
Max-Forwards: 70
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=623551024"; orig-ioi=home12.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:scscf1.home1.net;lr>
From: <sip:user2_public1@home2.net>;tag=151170
To: <sip:user1_public1@home1.net>;tag=31415
Call-ID: gahjt393yhakfh83hfasl98a
CSeq: 43 NOTIFY
Subscription-State: active;expires=5000
Event: presence
Contact: <sip:ps.home2.net>
Content-Type: application/pidf+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
   <presence xmlns="urn:ietf:params:xml:ns:pidf"
             xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
             xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
             xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid-person"
             xmlns:dmp="urn:ietf:params:xml:ns:pidf:person"
             xmlns:pcp="urn:ietf:params:xml:ns:pidf:servcaps"
             xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
         entity="pres:user2_public1@home2.net">

     <tuple id="a8098a.672364762364">
       <status>
         <basic>open</basic>
         <es:privacy><text/></es:privacy>
         <es:status-icon>http://example.com/~user2/icon.gif</es:status-icon>
       </status>
       <et:class>sip</et:class>
       <pcp:video>false</pcp:video>
       <pcp:audio>true</pcp:audio>
       <contact priority="0.8">sip:user2_public1@home2.net</contact>
       <note xml:lang="en">Don't Disturb Please!</note>
       <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
       <timestamp>2003-08-27T11:49:29Z</timestamp>
     </tuple>

     <dmp:person>
       <ep:class>presentity</ep:class>
       <ci:homepage>http://example.com/~user2</ci:homepage>
       <ci:card>http://example.com/~user2/card.vcd</ci:card>
       <dmp:status>
         <ep:activities><ep:meeting/></ep:activities>
         <ep:place-type until="2003-08-27T17:30:00Z">office</ep:place-type>
       </dmp:status>
     </dmp:person>

   </presence>

<?xml version="1.0" encoding="UTF-8"?>
   <presence xmlns="urn:ietf:params:xml:ns:pidf"
             xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
             xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
             xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
             xmlns:pcp="urn:ietf:params:xml:ns:simple-prescaps-ext"
         entity="pres:user2_public1@home2.net ">

     <tuple id="a8098a.672364762364">
       <status>
         <basic>closed</basic>
         <es:placetype until="2003-08-27T18:30:00Z">office</es:placetype>
         <es:privacy>public</es:privacy>
         <pcp:prescaps>
           <pcp:video negated="false"></pcp:video>
           <pcp:mobility>mobile</pcp:mobility>
           <pcp:audio negated="true"></pcp:audio>
         </pcp:prescaps>
       </status>
       <et:class>sip</et:class>
       <et:contact-type>service</et:contact-type>
       <contact priority="1.0">sip:user2_public1@home2.net</contact>
```

```
      <timestamp>2003-08-27T17:35:29Z</timestamp>
    </tuple>

    <tuple id="sfddsj74.78">
      <status>
        <basic>open</basic>
      </status>
      <et:class>presentity</et:class>
      <et:contact-type>presentity</et:contact-type>
      <ci:homepage>http://example.com/~user2</ci:homepage>
      <ci:icon>http://example.com/~user2/icon.gif</ci:icon>
      <ci:card>http://example.com/~user2/card.vcd</ci:card>
      <note xml:lang="en">I'm in a boring meeting</note>
      <note xml:lang="en">I'll be in Tokyo next week</note>
      <timestamp>2003-08-27T12:00:30Z</timestamp>
    </tuple>

  </presence>
```

P-Charging-Vector: The PS populates the icid parameter with a globally unique value and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The PS populates the P-Charging-Function-Addresses header field to be passed to the S-CSCF.

2. **NOTIFY request (S-CSCF to RLS) - see example in table A.5.3.1-2**

The S-CSCF#1 forwards the NOTIFY request to the RLS.

Table A.5.3.1-2: NOTIFY request (S-CSCF to RLS)

```
NOTIFY sip:rls.home1.net SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bKehuehjgt, SIP/2.0/UDP
      scscf2.home2.net;branch=z9hG4bK764z87.1, SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
Max-Forwards: 69
P-Charging-Vector:
P-Charging-Function-Addresses:
Record-Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the RLS.

3. **200 (OK) response (RLS to S-CSCF) - see example in table A.5.3.1-3**

The RLS generates a 200 (OK) response to the NOTIFY request.

Table A.5.3.1-3: 200 (OK) response (RLS to S-CSCF)

```
SIP/2.0 200 OK
Via:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=623551024"; orig-
      ioi=home1.net:term-ioi=home1.net
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

P-Charging-Vector: The RLS stores the terminating Inter Operator Identifier (IOI) parameter and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

4. **200 (OK) response (S-CSCF to PS) - see example in table A.5.3.1-4**

The S-CSCF#1 forwards the 200 (OK) response to the PS.

Table A.5.3.1-4: 200 (OK) response (S-CSCF to PS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP ps.home2.net;branch=z9hG4bK348923.1
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=623551024"; orig-
      ioi=home1.net:term-ioi=home1.net
From:
To:
Call-ID:
CSeq:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the terminating Inter Operator Identifier (IOI) parameter and.populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

5. **NOTIFY request (RLS to S-CSCF#1) - see example in table A.5.3.1-5**

The RLS may decide to wait for other notifications and combine them in a single notification towards the UE or it sends the notification to the UE without any waiting. In this example, the RLS does not wait for other notifications.

Table A.5.3.1-5: NOTIFY request (RLS to S-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 70
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=723551024"; orig-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net;lr>
From: <sip:user1_list1@home1.net>;;tag=151170
To: <sip:user1_public1@home1.net>;tag=31415
Call-ID: gahjt393yhakfh83hfasl98a
CSeq: 90 NOTIFY
Subscription-State: active;expires=4500
Require: eventlist
Event: presence
Contact: <sip:rls.home1.net>
Content-Type: multipart/related;type="application/rlmi+xml";
       start="<njhhsdhj@rls.home1.net>";boundary="70UBfW7L78hjgfgUPe5z"
Content-Length: (...)

--70UBfW7L78hjgfgUPe5z
Content-Transfer-Encoding: binary
Content-ID: <njhhsdhj@rls.home1.net>
Content-Type: application/rlmi+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
  <list xmlns="urn:ietf:params:xml:ns:rmli"
```

```
        uri="sip:user1_list1@home1.net"
                  version="2"
                  fullState="false"
     <resource uri="pres:user2_public1@home2.net" name="Kovacs Janos">
       <instance id="hqzsuxtfyq" state="active" cid="uhjgfd@rls.home1.net"/>
     </resource>
  </list>

--70UBfW7L78hjgfgUPe5z
Content-Transfer-Encoding: binary
Content-ID: <uhjgfd@rls.home1.net>
Content-Type: application/pidf+xml;charset="UTF-8"

<?xml version="1.0" encoding="UTF-8"?>
   <presence xmlns="urn:ietf:params:xml:ns:pidf"
             xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
             xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
             xmlns:ep="urn:ietf:params:xml:ns:pidf:rpid-person"
             xmlns:dmp="urn:ietf:params:xml:ns:pidf:person"
             xmlns:pcp="urn:ietf:params:xml:ns:pidf:servcaps"
             xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
             entity="pres:user2_public1@home2.net">

     <tuple id="a8098a.672364762364">
       <status>
         <basic>open</basic>
         <es:privacy><text/></es:privacy>
         <es:status-icon>http://example.com/~user2/icon.gif</es:status-icon>
       </status>
       <et:class>sip</et:class>
       <pcp:video>false</pcp:video>
       <pcp:audio>true</pcp:audio>
       <contact priority="0.8">sip:user2_public1@home2.net</contact>
       <note xml:lang="en">Don't Disturb Please!</note>
       <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
       <timestamp>2003-08-27T11:49:29Z</timestamp>
     </tuple>

     <dmp:person>
       <ep:class>presentity</ep:class>
       <ci:homepage>http://example.com/~user2</ci:homepage>
       <ci:card>http://example.com/~user2/card.vcd</ci:card>
       <dmp:status>
         <ep:activities><ep:meeting/></ep:activities>
         <ep:place-type until="2003-08-27T17:30:00Z">office</ep:place-type>
       </dmp:status>
     </dmp:person>

   </presence>
<?xml version="1.0" encoding="UTF-8"?>
   <presence xmlns="urn:ietf:params:xml:ns:pidf"
             xmlns:es="urn:ietf:params:xml:ns:pidf:status:rpid-status"
             xmlns:et="urn:ietf:params:xml:ns:pidf:rpid-tuple"
             xmlns:ci="urn:ietf:params:xml:ns:pidf:cipid"
             xmlns:pcp="urn:ietf:params:xml:ns:simple-prescaps-ext"
             entity="pres:user2_public1@home2.net">

     <tuple id="a8098a.672364762364">
       <status>
         <basic>closed</basic>
         <es:placetype until="2003-08-27T18:30:00Z">office</es:placetype>
         <es:privacy>public</es:privacy>
         <pcp:prescaps>
           <pcp:video negated="false"></pcp:video>
           <pcp:mobility>mobile</pcp:mobility>
           <pcp:audio negated="true"></pcp:audio>
         </pcp:prescaps>
       </status>
       <et:class>sip</et:class>
       <et:contact-type>service</et:contact-type>
       <contact priority="1.0">sip:user2_public1@home2.net</contact>
       <timestamp>2003-08-27T17:35:29Z</timestamp>
     </tuple>

     <tuple id="sfddsj74.78">
       <status>
         <basic>open</basic>
       </status>
```

```
        <et:class>presentity</et:class>
        <et:contact-type>presentity</et:contact-type>
        <ci:homepage>http://example.com/~user2</ci:homepage>
        <ci:icon>http://example.com/~user2/icon.gif</ci:icon>
        <ci:card>http://example.com/~user2/card.vcd</ci:card>
        <note xml:lang="en">I'm in a boring meeting</note>
        <note xml:lang="en">I'll be in Tokyo next week</note>
        <timestamp>2003-08-27T12:00:30Z</timestamp>
    </tuple>

</presence>

--70UBfW7L78hjgfgUPe5z
```

P-Charging-Vector: The RLS populates the icid parameter with a globally unique value and populates the identifier of its own network to the originating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The RLS populates the P-Charging-Function-Addresses header field to be passed to the S-CSCF.

6. **NOTIFY request (S-CSCF to P-CSCF) - see example in table A.5.3.6**

The S-CSCF forwards the NOTIFY request to the P-CSCF.

Table A.5.3.1-6: NOTIFY request (S-CSCF to P-CSCF)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 69
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=723551024"
P-Charging-Function-Addresses:
Route: <sip:pcscf1.visited1.net;lr>
Record-Route: <sip:scscf1.home1.net;lr>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Require:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

P-Charging-Vector: The S-CSCF stores the originating Inter Operator Identifier (IOI) parameter.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the P-CSCF.

7. **NOTIFY request (P-CSCF to UE) - see example in table A.5.3.1-7**

The P-CSCF forwards the NOTIFY request to the UE.

Table A.5.3.1-7: NOTIFY request (P-CSCF to UE)

```
NOTIFY sip:[5555::aaa:bbb:ccc:ddd]:1357;comp=sigcomp SIP/2.0
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
Max-Forwards: 68
Record-Route: <sip:scscf1.home1.net;lr>, <sip:pcscf1.visited1.net:7531;lr;comp=sigcomp>
From:
To:
Call-ID:
CSeq:
Subscription-State:
Require:
Event:
Contact:
Content-Type:
Content-Length:

(…)
```

8. **200 (OK) response (UE to P-CSCF) - see example in table A.5.3.1-8**

The UE acknowledges the NOTIFY request with a 200 (OK) response to the P-CSCF.

Table A.5.3.1-8: 200 (OK) response (UE to P-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pcscf1.visited1.net;branch=240f34.1, SIP/2.0/UDP
      scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info: 3GPP-UTRAN-TDD; utran-cell-id-3gpp=234151D0FCE11
From:
To:
Call-ID:
CSeq:
Content-Length: 0
```

9. **200 (OK) response (P-CSCF to S-CSCF) – see example in table A.5.3.1-9**

The P-CSCF forwards the 200 (OK) response to the S-CSCF.

Table A.5.3.1-9: 200 (OK) response (P-CSCF to S-CSCF)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP scscf1.home1.net;branch=z9hG4bK332b23.1, SIP/2.0/UDP
      rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=723551024"
From:
To:
Call-ID:
CSeq:
Content-Length:
```

10. **200 (OK) response (S-CSCF to RLS) – see example in table A.5.3.1-10**

The S-CSCF forwards the response to the RLS in the home network of the presentity.

Table A.5.3.1-10: 200 (OK) response (S-CSCF to RLS)

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP rls.home1.net;branch=z9hG4bK240f34.1
P-Access-Network-Info:
P-Charging-Vector: icid-value="AyretyU0dm+6O2IrT5tAFrbHLso=723551024"; orig-
      ioi=home1.net:term-ioi=home1.net
P-Charging-Function-Addresses: ccf=[5555::b99:c88:d77:e66]; ccf=[5555::a55:b44:c33:d22];
      ecf=[5555::1ff:2ee:3dd:4ee]; ecf=[5555::6aa:7bb:8cc:9dd]
From:
To:
Call-ID:
CSeq:
Content-Length:
```

P-Charging-Vector: The S-CSCF stores the terminating Inter Operator Identifier (IOI) parameter and populates the identifier of its own network to the terminating Inter Operator Identifier (IOI) parameter of this header.

P-Charging-Function-Addresses: The S-CSCF stores the P-Charging-Function-Addresses header field and passes this header to the RLS.